9.3

IBM MQ ' nun Güvenliğinin sağlanması

Not

Bu bilgileri ve desteklediği ürünü kullanmadan önce, “Özel notlar” sayfa 703 bölümündeki bilgileri okuyun.

Bu basım, yeni basımlarda tersi belirtilmedikçe, IBM® MQ sürüm 9 yayın düzeyi 3 ve sonraki tüm yayınlar ve değişiklikler için geçerlidir.

IBM'e bilgi gönderdiğinizde, IBM ' e bu bilgileri size hiçbir sorumluluk yüklemeden uygun gördüğü yöntemlerle kullanması ya da dağıtması için münhasır olmayan bir hak verirsiniz.

İçindekiler

güvenlikIBM MQ

Güvenlik, hem IBM MQ uygulamalarının geliştiricileri hem de IBM MQ sistem yöneticileri için önemli bir konudur. Mutlak olarak, güvenli bölge içindeki ve işletmen iş istasyonlarındaki tüm donanım ve yazılımların destek yaşam çevrimleri içinde olduğundan, zorunlu yazılım güncellemeleriyle güncel olduğundan ve güvenlik güncellemelerinin hemen uygulandığından emin olmanız gerekir.

İlgili başvurular

IBM Güvenlik Açığı Yönetimi

z/OS IBM Z ve LinuxOne Güvenlik Portalı

Güvenliğe genel bakış

Bu konu derlemi, IBM MQ güvenlik kavramlarını tanıtır.

Güvenlik kavramları ve mekanizmaları, herhangi bir bilgisayar sistemi için geçerli olduğu şekilde, önce sunulur, ardından bu güvenlik mekanizmaları IBM MQiçinde uygulandıkça tartışılır.

Güvenliğin yaygın olarak kabul edilen yönleri şunlardır:

- “Tanımlama ve kimlik doğrulama” sayfa 7
- “Yetkilendirme” sayfa 9
- “Denetleme” sayfa 9
- “Gizlilik” sayfa 9
- “Veri bütünlüğü” sayfa 10

Güvenlik mekanizmaları , güvenlik hizmetlerini uygulamak için kullanılan teknik araçlar ve tekniklerdir. Bir mekanizma, belirli bir hizmeti sağlamak için kendi kendine ya da başkalarıyla birlikte çalışabilir. Ortak güvenlik mekanizmalarına örnek olarak şunlar verilebilir:

- “Şifreleme” sayfa 11
- “İleti özetleri ve dijital imzalar” sayfa 12
- “dijital sertifikalar” sayfa 13
- “Genel anahtar altyapısı (PKI)” sayfa 17

Bir IBM MQ uygulaması planlarken, sizin için önemli olan güvenlik yönlerini uygulamak için hangi güvenlik mekanizmalarının gerekli olduğunu göz önünde bulundurun. Bu konuları okuduktan sonra neleri dikkate almanız gerekeceği hakkında bilgi için bkz. “Güvenlik gereksinimlerinizin planlanması” sayfa 84.

Tanımlama ve kimlik doğrulama

Tanımlama , sistemde çalışan bir sistem kullanıcısını ya da uygulamayı benzersiz olarak tanımlama yeteneğidir. *Kimlik Doğrulaması* , bir kullanıcının ya da uygulamanın o kişinin ya da uygulamanın iddia ettiği kişinin gerçekten kim olduğunu kanıtlama yeteneğidir.

Örneğin, bir kullanıcı kimliği ve parola girerek sistemde oturum açan bir kullanıcı düşünün. Sistem, kullanıcıyı tanımlamak için kullanıcı kimliğini kullanır. Sistem, oturum açma sırasında sağlanan parolanın doğru olup olmadığını denetleyerek kullanıcının kimliğini doğrular.

IBM MQ içinde kimlik belirleme ve kimlik doğrulama

Bir uygulama IBM MQ' a bağlandığında, kullanıcı kimliği her zaman bağlantıyla ilişkilendirilir. Kullanıcı kimliği, başlangıçta uygulama işlemiyle ilişkilendirilmiş İşletim Sistemi kullanıcı kimliğidir. Bu kimlik genellikle, kuyruk yöneticisiyle aynı sistemde bulunan yerel olarak bağlı uygulamalar için yeterlidir. Ancak, kuyruk yöneticisi bağlantıyla ilişkili kimliği birkaç şekilde doğrulayabilir ve değiştirebilir. Güvenilir olması

gerekmeyen istemci uygulamaları bir ağ üzerinden bir kuyruk yöneticisine bağlandığında, bağlantıyla ilişkili kimliğin doğrulanması önemlidir.

Bir IBM MQ kuyruk yöneticisine uygulama bağlantısıyla ilişkili kimlik, aşağıdaki mekanizmalardan herhangi biri kullanılarak oluşturulabilir:

- Bir uygulama bir kuyruk yöneticisine bağlandığında, bir kullanıcı kimliği ve parola sağlayabilir. Kuyruk yöneticisi, kimlik bilgilerini yapılandırmasına dayalı olarak doğrular. Örneğin, kullanıcı kimliği ve parola, kimliği doğrulanacak kuyruk yöneticisinin İşletim Sistemine ya da LDAP sunucusuna iletilebilir.
- V 9.3.4 IBM MQ 9.3.4' den bir uygulama, dış kimlik doğrulama sunucusundan edineceği bir kimlik doğrulama belirteci de sağlayabilir. Kimlik doğrulama belirteçleriyle ilgili daha fazla bilgi için bkz. “Kimlik doğrulama belirteçleriyle çalışma” sayfa 344.
- Bir istemci kanalı, geçerli bir dijital sertifikayla yapılandırıldıysa, TLS karşılıklı kimlik doğrulamasını kullanacak şekilde yapılandırılabilir. TLS kimlik doğrulaması, uygun bir kullanıcı kimliğini bağlantıyla ilişkilendirmek için bir kanal kimlik doğrulaması (CHLAUTH) kuralıyla birleştirilebilir. Daha fazla bilgi için bkz. “TLS ' nin tanımlama, kimlik doğrulama, gizlilik ve bütünlüğü nasıl sağladığı” sayfa 20,
- Kanal kimlik doğrulaması (CHLAUTH) kuralları, bağlantıya ilişkin bilgilere dayalı olarak kimliği geçersiz kılabilir. Örneğin, bir kanal kimlik doğrulama kuralı, istemcinin IP adresine dayalı olarak bir bağlantıyla ilişkili kullanıcı kimliğini ayarlayabilir.
- Özel çıkış kodu, seçtiğiniz ölçütlere dayalı olarak bir kimlik ayarlayabilir.

Kimlik ve kimlik doğrulaması, iki kuyruk yöneticisi arasındaki kanallar için de geçerlidir. Bu kanallar mesaj kanalları olarak bilinir. Bir ileti kanalı başlatıldığında, kanalın her ucundaki ileti kanalı aracısı (MCA) ortağının kimliğini doğrulayabilir. Bu teknik, *karşılıklı kimlik doğrulama*olarak bilinir. Gönderen MCA için, mesaj göndermek üzere olduğu ortağın gerçek olduğuna dair güvence sağlar. Benzer şekilde, alan MCA ' nın gerçek bir ortaktan mesaj almak üzere olduğu kesindir.

Bir kimlik oluşturulduğunda ve gerekirse kimliği doğrulandığında, IBM MQ tarafından çeşitli şekillerde kullanılır:

- Önemli olan, varsayılan olarak, sonraki “Yetkilendirme” sayfa 9 denetimleri bu kimlik kullanılarak yapılır. Örneğin, bir uygulama bir iletiyi kuyruğa yerleştirmeyi denerse, kuyruk yöneticisi, uygulamayla ilişkili kimliğin kuyruk nesnesi üzerinde 'koyma' yetkisi olduğunu doğrular.
- Ayrıca, her ileti *ileti bağlamı* bilgilerini içerebilir. Bu bilgiler ileti tanımlayıcısında (MQMD) tutulur. Bir uygulama iletiyi bir kuyruğa koyduğunda, kuyruk yöneticisi ileti bağlamını otomatik olarak oluşturabilir. Diğer bir seçenek olarak, uygulamayla ilişkili kullanıcı kimliği bunu yapma yetkisine sahip ise, uygulama ileti bağlamını sağlayabilir. Bir iletideki bu bağlam bilgisi, iletiyi oluşturan ile ilgili ileti bilgilerini alan uygulamaya bilgi verir. Örneğin, iletiyi koyan uygulamanın adını ve uygulamayla ilişkili kullanıcı kimliğini içerir.

Reddedememe

Reddetmek istemeyen hizmetin genel amacı, belirli bir iletinin belirli bir kişiyle ilişkili olduğunu kanıtlayabilmektir.

inkar etme hizmeti, tanımlama ve kimlik doğrulama hizmetinin bir uzantısı olarak görüntülenebilir. Genel olarak, reddetme, veriler elektronik olarak iletildiğinde geçerlidir; örneğin, hisse senedi satın almak veya satmak için bir borsa aracısına bir sipariş veya bir bankadan bir hesaptan diğerine para transferi için bir sipariş.

Reddedememe hizmeti, her bileşenin farklı bir işlev sağladığı birden çok bileşen içerebilir. Bir iletinin göndereni iletiyi göndermeyi reddediyorsa, *kaynak kanıtı* olan inkar etme hizmeti alıcıya iletinin o kişi tarafından gönderildiğine dair inkar edilemez bir kanıt sağlayabilir. Bir iletinin alıcısı iletiyi almayı reddediyorsa, *teslim kanıtı* olan inkar edilemez hizmet, gönderene iletinin o kişi tarafından alındığına dair inkar edilemez bir kanıt sağlayabilir.

Pratikte, neredeyse %100 kesinliğe sahip kanıt veya inkar edilemez kanıt, zor bir hedeftir. Gerçek dünyada hiçbir şey tamamen güvenli değildir. Güvenliğin yönetilmesi, riskin işletme tarafından kabul edilebilir bir düzeye yönetilmesi ile daha çok ilgilidir. Böyle bir ortamda, inkar etmeyen hizmetin daha gerçekçi bir beklentisi, mahkemede kabul edilebilir ve sizin davanızı destekleyen kanıtlar sunabilmektir.

IBM MQ , verileri elektronik olarak iletme aracı olduğundan, IBM MQ ortamında ilgili bir güvenlik hizmetidir. Örneğin, belirli bir iletinin belirli bir kişiyle ilişkili bir uygulama tarafından gönderildiğine ya da alındığına ilişkin eşzamanlı bir kanıta gereksinim duyabilirsiniz.

IBM MQ with Advanced Message Security , temel işlevinin bir parçası olarak inkar etmeme hizmeti sağlamaz. Ancak bu ürün belgeleri, kendi çıkış programlarınızı yazarak IBM MQ ortamında kendi inkar etme hizmetinizi nasıl sağlayabileceğinize ilişkin öneriler içerir.

Yetkilendirme

Yetkilendirme , erişimi yalnızca yetkili kullanıcılarla ve uygulamalarıyla sınırlayarak bir sistemdeki kritik kaynakları korur. Bir kaynağın yetkisiz olarak kullanılmasını ya da yetkisiz olarak kullanılmasını önler.

IBM MQ içinde yetkilendirme

Belirli kişilerin ya da uygulamaların IBM MQ ortamınızda yapabileceği işlemleri sınırlamak için yetki kullanabilirsiniz.

Aşağıda, IBM MQ ortamındaki yetkilendirmenin bazı örnekleri verilmiştir:

- IBM MQ kaynaklarını yönetmek için yalnızca yetkili bir yöneticinin komut vermesine izin verme.
- Uygulamanın bir kuyruk yöneticisine bağlanmasına izin verilmesi için, uygulamayla ilişkili kullanıcı kimliğinin bu yetkiye sahip olması gerekir.
- Bir uygulamanın yalnızca işlevi için gerekli olan kuyrukları açmasına izin verme.
- Bir uygulamanın yalnızca işlevi için gerekli olan konulara abone olmasına izin verilmesi.
- Bir uygulamanın yalnızca, işlevi için gerekli olan işlemleri bir kuyrukta gerçekleştirmesine izin verme. Örneğin, bir uygulamanın iletileri koymak ya da almak için değil, yalnızca belirli bir kuyruktaki iletilere göz atması gerekebilir.

Yetkilendirmenin nasıl ayarlanacağıyla ilgili daha fazla bilgi için bkz. “Planlama yetkisi” sayfa 88 ve ilişkili alt konular.

Denetleme

Denetleme , beklenmeyen ya da yetkisiz bir etkinliğin gerçekleşip gerçekleşmediğini ya da bu tür bir etkinliği gerçekleştirmek için herhangi bir girişimde bulunulup bulunulmadığını saptamak üzere olayları kaydetme ve denetleme işlemidir.

IBM MQ içinde denetleme

IBM MQ , olağan dışı etkinliğin gerçekleşmiş olduğunu kaydetmek için olay iletileri yayınlayabilir.

Aşağıda, IBM MQ ortamında denetime ilişkin bazı örnekler verilmiştir:

- Uygulama, açma yetkisi olmayan bir kuyruğu açma girişiminde bulunur. Özel işlemden geçirme olayı iletisi yayınlandı. Olay iletisini inceleyerek, bu girişimin gerçekleştirildiğini keşfeder ve hangi işlemin gerekli olduğuna karar verebilirsiniz.
- Bir uygulama bir kanalı açmayı dener, ancak TLS bağlantısına izin verilmediği için girişim başarısız olur. Özel işlemden geçirme olayı iletisi yayınlandı. Olay iletisini inceleyerek, bu girişimin gerçekleştirildiğini keşfeder ve hangi işlemin gerekli olduğuna karar verebilirsiniz.

Gizlilik

Gizlilik hizmeti, hassas bilgileri yetkisiz açıklamadan korur.

Hassas veriler yerel olarak saklandığında, erişim denetimi mekanizmaları, verilere erişilemediğinde verilerin okunamayacağı varsayımıyla verileri korumak için yeterli olabilir. Daha yüksek bir güvenlik düzeyi gerekiyorsa, veriler şifrelenebilir.

Özellikle Internet gibi güvenli olmayan bir ağ üzerinden, bir iletişim ağı üzerinden iletildiğinde hassas verileri şifreleyin. Bir ağ ortamında, erişim denetimi mekanizmaları, dinleme gibi verileri engelleme girişimlerine karşı etkili değildir.

IBM MQ içinde gizlilik

İletileri şifreleyerek IBM MQ içinde gizlilik uygulayabilirsiniz.

IBM MQ ortamında gizlilik aşağıdaki gibi sağlanabilir:

- Gönderen MCA bir iletim kuyruğundan ileti aldıktan sonra, IBM MQ iletiyi alan MCA ' ya ağ üzerinden gönderilmeden önce şifrelemek için TLS kullanır. Kanalın diğer ucunda, alan MCA iletiyi hedef kuyruğuna koymadan önce iletinin şifresi çözülür.
- İletiler yerel bir kuyrukta saklanırken, IBM MQ tarafından sağlanan erişim denetimi mekanizmaları, içeriklerini yetkisiz açıklamaya karşı korumak için yeterli olarak kabul edilebilir. Ancak, daha yüksek güvenlik düzeyi için, kuyruklarda saklanan iletileri şifrelemek için Advanced Message Security kullanabilirsiniz.
- z/OS Yerel kuyruklarda saklanan iletiler, z/OS veri kümesi şifrelemesi kullanılarak atıl olarak şifrelenebilir.

  veri kümesi şifrelemesiyle IBM MQ for z/OS üzerinde atıl durumdaki veriler için gizlilik. başlıklı bölüme bakın. ek bilgi için.

Veri bütünlüğü

Veri bütünlüğü hizmeti, verilerde yetkisiz değişiklik olup olmadığını saptar.

Verilerin değiştirilebileceği iki yol vardır: yanlışlıkla, donanım ve iletim hataları yoluyla ya da kasıtlı bir saldırı nedeniyle. Birçok donanım ürünü ve iletim protokolleri, donanım ve iletim hatalarını saptamada ve düzeltmekte mekanizmalara sahip olur. Veri bütünlüğü hizmetinin amacı, kasıtlı bir saldırı tespit etmektir.

Veri bütünlüğü hizmeti yalnızca verilerin değiştirilip değiştirilmediğini saptamayı amaçlar. Değiştirildiyse, verileri özgün durumuna geri yüklemeyi amaçlamaz.

Erişim reddedilirse veriler değiştirilemeyeceği sürece, erişim denetimi mekanizmaları veri bütünlüğüne katkıda bulunabilir. Ancak, gizlilik konusunda olduğu gibi, erişim denetimi mekanizmaları bir ağ ortamında etkili değildir.

IBM MQ içinde veri bütünlüğü

Veri bütünlüğü IBM MQ ortamında aşağıdaki gibi sağlanabilir:

- Bir ileti bir ağ üzerinden aktarılırken iletinin içeriğinin kasıtlı olarak değiştirilip değiştirilmediğini saptamak için TLS ' yi kullanabilirsiniz. TLS ' de ileti özeti algoritması, geçiş sırasında değiştirilen iletilerin algılanmasını sağlar.

  Tüm IBM MQ CipherSpecs , ileti verileri bütünlüğü sağlamayan TLS_RSA_WITH_NULL_NULL dışında bir ileti özeti algoritması sağlar.

  IBM MQ , değiştirilen iletileri aldıktan sonra saptar; IBM MQ değiştirilmiş bir ileti aldığında, hata günlüğüne bir AMQ9661 hata iletisi yazılır ve kanal durur.
- İletiler yerel bir kuyrukta saklanırken, IBM MQ tarafından sağlanan erişim denetimi mekanizmaları, iletilerin içeriğinin kasıtlı olarak değiştirilmesini önlemek için yeterli kabul edilebilir.

  Ancak, daha yüksek bir güvenlik düzeyi için, iletinin kuyruğa konması ile kuyruktan alınması arasında bir iletinin içeriğinin kasıtlı olarak değiştirilip değiştirilmediğini saptamak için Advanced Message Security kullanabilirsiniz.

  Değiştirilen bir ileti saptanırsa, iletiyi alma girişiminde bulunan uygulama bir MQRC_SECURITY_ERROR (2063) dönüş kodu alır. Uygulama bir MQGET çağrısı kullanıyorsa, ileti SYSTEM.PROTECTION.ERROR.QUEUE (KUYRUK).

Şifreleme kavramları

Bu konular derlemi, IBM MQiçin geçerli olan şifreleme kavramlarını açıklar.

Varlık terimi, bir kuyruk yöneticisine, IBM MQ MQI client, tek bir kullanıcıya ya da ileti alışverişi yapabilen başka bir sisteme başvurmak için kullanılır.

Şifreleme

Şifreleme, *düz metin*adı verilen okunabilir metin ile *şifreli metin*adı verilen okunamayan bir form arasında dönüştürme işlecidir.

Bu durum aşağıdaki gibi oluşur:

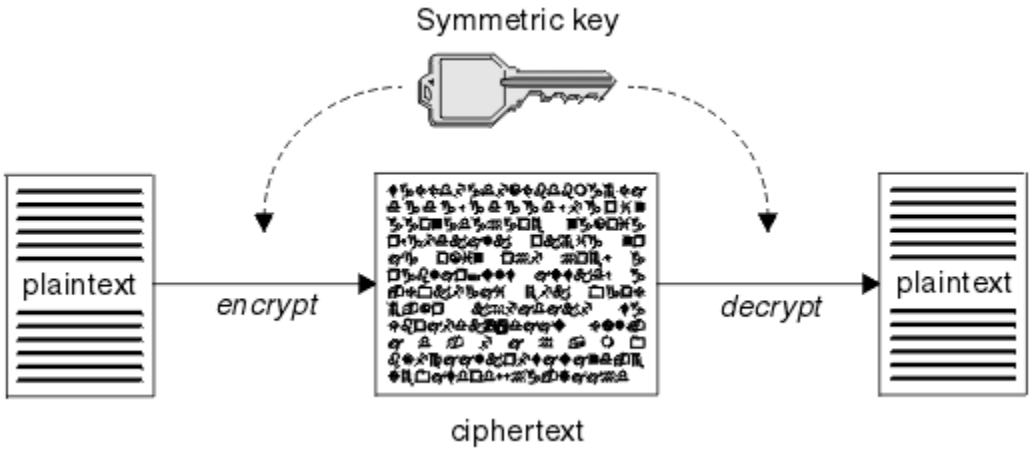
1. Gönderen, düz metin iletisini şifreli metne dönüştürür. Sürecin bu kısmına *şifreleme* denir (bazen *şifreleme*).
2. Şifreli metin alıcıya iletilir.
3. Alıcı, şifreli metin iletisini düz metin biçimine geri çevirir. İşlemin bu bölümüne *şifre çözme* (bazen *şifre çözme*) adı verilir.

Dönüştürme, iletim sırasında iletinin görünümünü değiştiren, ancak içeriği etkilemeyen bir dizi matematiksel işlemi içerir. Şifreli teknikler, şifreli bir ileti anlaşılabilir olmadığı için, gizliliği güvence altına alabilir ve iletileri yetkisiz görüntülemeye (gizlice dinleme) karşı koruyabilir. Mesaj bütünlüğünün güvencesini sağlayan dijital imzalar, şifreleme tekniklerini kullanır. Ek bilgi için bkz. “SSL/TLS ' de dijital imzalar” sayfa 22 .

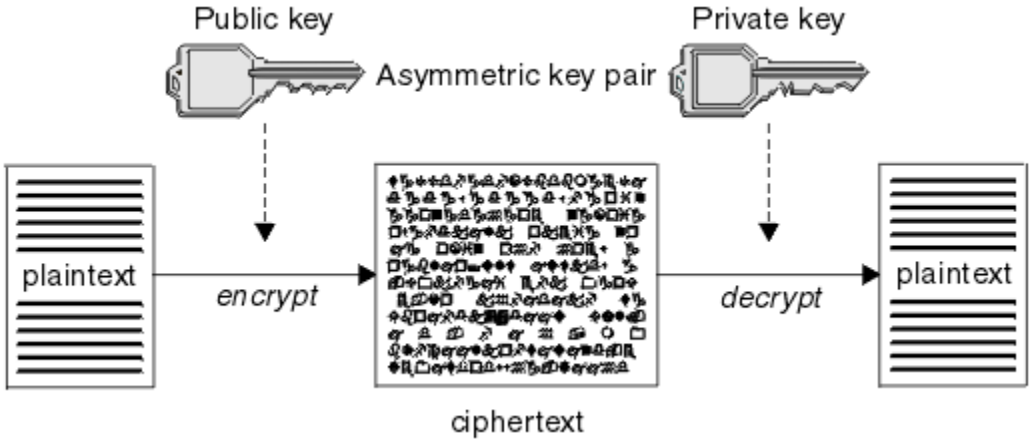
Kriptografik teknikler, anahtarların kullanımıyla özel olarak yapılan genel bir algoritmayı içerir. İki algoritma sınıfı vardır:

- Her iki tarafın da aynı gizli anahtarı kullanmasını gerektirenler. Paylaşılan anahtar kullanan algoritmalar, *simetrik* algoritmalar olarak bilinir. Şekil 1 sayfa 11 içinde simetrik anahtar şifrelemesi gösterilir.
- Şifreleme için bir anahtar ve şifre çözme için farklı bir anahtar kullananlar. Bunlardan biri gizli tutulmalı ama diğeri halka açık olabilir. Genel ve özel anahtar çiftlerini kullanan algoritmalar, *asimetrik* algoritmalar olarak bilinir. Şekil 2 sayfa 12 , *genel anahtar şifrelemesi*olarak da bilinen asimetrik anahtar şifrelemesi gösterir.

Kullanılan şifreleme ve şifre çözme algoritmaları genel olabilir, ancak paylaşılan gizli anahtar ve özel anahtar gizli tutulmalıdır.



Şekil 1. Simetrik anahtar şifrelemesi



Şekil 2. Asimetrik anahtar şifrelemesi

Şekil 2 sayfa 12 , alıcının genel anahtarıyla şifrelenmiş ve alıcının özel anahtarıyla şifresi çözülmüş düz metni gösterir. Şifre metninin şifresini çözmek için yalnızca istenen alıcı özel anahtarı tutar. Gönderenin iletileri özel bir anahtarla da şifreleyebileceğini unutmayın; bu, gönderenin açık anahtarını tutan herkesin iletinin şifresini çözmesine ve iletinin gönderenden gelmiş olması gerektiğine dair güvence verir.

Asimetrik algoritmalarla, iletiler genel ya da özel anahtarla şifrelenir, ancak yalnızca diğer anahtarla şifresi çözülebilir. Yalnızca özel anahtar gizlidir, genel anahtar herkes tarafından bilinebilir. Simetrik algoritmalarla, paylaşılan anahtar yalnızca iki taraf tarafından bilinmelidir. Buna *anahtar dağıtım sorunu*denir. Asimetrik algoritmalar daha yavaştır, ancak anahtar dağılım problemi olmaması avantajına sahiplerdir.

Şifreleme ile ilişkili diğer terminoloji:

Güvenlik düzeyi
Şifrelemenin gücü anahtar boyutuna göre belirlenir. Asimetrik algoritmalar büyük anahtarlar gerektirir, örneğin:

1024 bit	Düşük güçlü asimetrik anahtar
2048 bit	Orta güçlü asimetrik anahtar
4096 bit	Yüksek güçlü asimetrik anahtar

Simetrik anahtarlar daha küçüktür: 256 bit anahtarlar size güçlü şifreleme sağlar.

Blok şifreleme algoritması
Bu algoritmalar verileri bloklara göre şifreler. Örneğin, RSA Data Security Inc. ' in RC2 algoritması 8 bayt uzunluğunda blokları kullanır. Blok algoritmaları genellikle akış algoritmalarından daha yavaştır.

Akış şifresi algoritması
Bu algoritmalar her bir veri baytı üzerinde çalışır. Akış algoritmaları genellikle blok algoritmalarından daha hızlıdır.

İleti özetleri ve dijital imzalar

İleti özeti, bir iletinin içeriğinin değişmez büyüklüklü sayısal gösterimidir. İleti özeti bir özet fonksiyonu ile hesaplanır ve şifrelenebilir ve dijital imza oluşturur.

Bir ileti özetini hesaplamak için kullanılan hash işlevi iki ölçütü karşılamalıdır:

- Bir yolu olmalı. Olası tüm iletilerin sınanması dışında, belirli bir ileti özetine karşılık gelen iletiyi bulmak için işlevin tersine çevrilememesi gerekir.
- Aynı özet için özet değeri olan iki ileti bulmak, hesaplanabilir olarak olanaksız olmalıdır.

İleti özeti iletinin kendisiyle birlikte gönderilir. Alıcı, ileti için bir özet oluşturabilir ve bunu gönderenin özeti ile karşılaştırabilir. İletinin bütünlüğü, iki mesaj özetleri aynı olduğunda doğrulanır. İletim sırasında herhangi bir kurcalama neredeyse kesinlikle farklı bir mesaj sindirme ile sonuçlanır.

Gizli simetrik anahtar kullanılarak oluşturulan bir ileti özeti, iletinin değiştirilmediğini garanti edebileceği için İleti Kimlik Doğrulama Kodu (MAC) olarak bilinir.

Gönderen ayrıca bir ileti özeti oluşturabilir ve daha sonra asimetrik bir anahtar çiftinin özel anahtarını kullanarak bir dijital imza oluşturarak özeti şifreleyebilir. Daha sonra imzanın yerel olarak oluşturulan bir özet ile karşılaştırılmadan önce alıcı tarafından şifresi çözülmelidir.

İlgili kavramlar

"SSL/TLS ' de dijital imzalar" sayfa 22
Dijital imza, bir iletinin gösteriminin şifrelenmesiyle oluşturulur. Şifreleme, imza sahibinin özel anahtarını kullanır ve verimlilik için genellikle mesajın kendisinden ziyade bir ileti özeti üzerinde çalışır.

dijital sertifikalar

Dijital sertifikalar, bir genel anahtarın belirli bir varlığa ait olduğunu onaylayan, kimliğe bürünmeye karşı koruma sunar. Bunlar bir Sertifika Yetkilisi tarafından verilir.

Dijital sertifikalar, kimliğe bürünmeye karşı koruma sağlar; çünkü dijital sertifika, sahibine bir genel anahtarı bağlar; sahip ister bir birey, ister bir kuyruk yöneticisi ister başka bir varlık olsun. Dijital sertifikalar genel anahtar sertifikaları olarak da bilinir, çünkü asimetrik bir anahtar şeması kullandığınızda ortak bir anahtarın sahipliğine ilişkin güvenceler verir. Sayısal sertifika, bir varlığın ortak anahtarını içerir ve genel anahtarın o varlığa ait olduğunu belirtir:

- Sertifika tek bir varlık içinse, sertifikaya *kişisel sertifika* ya da *kullanıcı sertifikası*denir.
- Sertifika bir Sertifika Yetkilisi içinse, sertifikaya *CA sertifikası* ya da *imzalayıcı sertifikası*denir.

Ortak anahtarlar doğrudan sahipleri tarafından başka bir varlığa gönderilirse, iletinin engellenebilme ve genel anahtarın başka bir varlıkla değiştirilme riski vardır. Bu, *ortadaki saldırı adamı*olarak bilinir. Bu sorunun çözümü, açık anahtarın iletişim kurmakta olduğunuz varlığa gerçekten ait olduğuna dair size güçlü bir güvence veren, güvenilir bir üçüncü taraf aracılığıyla açık anahtarları değiştirmektir. Genel anahtarınızı doğrudan göndermek yerine, güvenilir üçüncü kişiden bunu bir dijital sertifikaya dahil etmesini istiyorsunuz. Dijital sertifikaları veren güvenilir üçüncü kişiye Sertifika Yetkilisi (CA) adı verilir (bkz. "Sertifika Yetkilileri" sayfa 14).

Dijital sertifikada ne var?

Sayısal sertifikalar, X.509 standardında belirlendiği şekilde belirli bilgi parçaları içerir.

IBM MQ tarafından kullanılan sayısal sertifikalar, gerekli bilgileri ve bunu göndermek için kullanılacak biçimi belirten X.509 standardına uygundur. X.509 , X.500 serisi standartların Kimlik Doğrulama çerçevesi parçasıdır.

Dijital sertifikalar, sertifikalandırılmakta olan varlıkla ilgili en az aşağıdaki bilgileri içerir:

- Sahibin genel anahtarı
- Sahibin Ayırt Edici Adı
- Sertifikayı veren CA ' nın Ayırt Edici Adı
- Sertifikanın geçerli olduğu tarih
- Sertifikanın süre bitim tarihi
- X.509' da tanımlandığı şekilde, sertifika verileri biçiminin sürüm numarası. X.509 standardının yürürlükteki sürümü Sürüm 3 'tür ve çoğu sertifika bu sürüme uygundur.
- Seri numarası. Bu, sertifikayı veren CA tarafından atanan benzersiz bir tanıtıcıdır. Seri numarası, sertifikayı veren CA içinde benzersizdir: Aynı CA sertifikası tarafından imzalanmış iki sertifika aynı seri numarasına sahip değildir.

X.509 Sürüm 2 sertifikası bir Sertifika Veren Tanıtıcısı ve bir Konu Tanıtıcısı içerir ve X.509 Sürüm 3 sertifikası bir dizi uzantı içerebilir. Temel Kısıt uzantısı gibi bazı sertifika uzantıları *standart*, ancak diğerleri

somutlamaya özgüdür. Bir uzantı *kritik*olabilir; bu durumda, bir sistem alanı tanıyabilir; alanı tanımazsa sertifikayı reddetmelidir. Bir uzantı kritik değilse, sistem bunu tanımazsa yoksayabilir.

Kişisel sertifikadaki dijital imza, bu sertifikayı imzalayan CA ' nın özel anahtarı kullanılarak oluşturulur. Kişisel sertifikayı doğrulaması gereken herkes bunu yapmak için CA ' nın genel anahtarını kullanabilir. CA ' nın sertifikası ortak anahtarını içeriyor.

Dijital sertifikalar özel anahtarınızı içermiyor. Özel anahtarını gizli tutmalısın.

Kişisel sertifikalara ilişkin gereksinimler

IBM MQ , X.509 standardına uygun dijital sertifikaları destekler. İstemci kimlik denetimi seçeneğini gerektirir.

IBM MQ eşler arası bir sistem olduğundan, SSL/TLS terminolojisinde istemci kimlik doğrulaması olarak görüntülenir. Bu nedenle, SSL/TLS kimlik doğrulaması için kullanılan herhangi bir kişisel sertifikanın istemci kimlik doğrulamasının anahtar kullanımına izin vermesi gerekir. Tüm sunucu sertifikalarında bu seçenek etkinleştirilmediğinden, sertifika sağlayıcısının güvenli sertifika için kök sertifika kuruluşunda istemci kimlik doğrulamasını etkinleştirmesi gerekebilir.

Sayısal sertifika için veri biçimini belirten standartlara ek olarak, bir sertifikanın geçerli olup olmadığını belirlemeye yönelik standartlar da vardır. Bu standartlar, belirli güvenlik ihlallerini önlemek için zaman içinde güncellenmiştir. Örneğin, eski X.509 sürüm 1 ve 2 sertifikaları, sertifikanın diğer sertifikaları imzalamak için meşru olarak kullanılıp kullanılamayacağını göstermiyordu. Bu nedenle, kötü amaçlı bir kullanıcının meşru bir kaynaktan kişisel sertifika alması ve diğer kullanıcıların kimliğine bürünmek için tasarlanmış yeni sertifikalar oluşturması mümkündü.

X.509 sürüm 3 sertifikaları kullanılırken, hangi sertifikaların diğer sertifikaları meşru olarak imzalayabileceğini belirtmek için BasicConstraints ve KeyUsage sertifika uzantıları kullanılır. IETF RFC 5280 standardı, taklit saldırılarını önlemek için uyumlu uygulama yazılımının uygulaması gereken bir dizi sertifika doğrulama kuralını belirtir. Sertifika kuralları kümesi, sertifika doğrulama ilkesi olarak bilinir.

IBM MQiçindeki sertifika doğrulama ilkeleri hakkında daha fazla bilgi için bkz. “IBM MQ içindeki sertifika doğrulama ilkeleri” sayfa 44.

Sertifika Yetkilileri

Sertifika Yetkilisi (CA), bir varlığın açık anahtarının gerçekten bu varlığa ait olduğuna dair bir güvence sağlamak için dijital sertifikaları veren güvenilir bir üçüncü taraftır.

Bir CA ' nın rolleri şunlardır:

- Dijital sertifika isteği alındıktan sonra, kişisel sertifikayı oluşturmadan, imzalamadan ve iade etmeden önce istekte bulunanın kimliğini doğrulamak için
- CA sertifikasında CA ' nın kendi genel anahtarını sağlamak için
- Artık bir Sertifika İptal Listesinde (CRL) güvenilmeyen sertifika listelerini yayınlamak için. Daha fazla bilgi için bkz. “İptal edilen sertifikalarla çalışma” sayfa 355
- OCSP yanıt veren sunucusunu çalıştırarak sertifika iptal durumuna erişim sağlamak için

Ayırt Edici Adlar

Ayırt edici ad (DN), X.509 sertifikasında bir varlığı benzersiz olarak tanıtır.

Uyarı: Bir SSLPEER süzgecinde yalnızca aşağıdaki çizelgedeki öznitelikler kullanılabilir. Sertifika DN ' leri başka öznitelikler içerebilir, ancak bu özniteliklerde süzmeye izin verilmez.

Çizelge 1. Bir SSLPEER süzgecinde kullanılabilecek DN ' de bulunan öznitelik tipleri

Öznitelik tipi	**Açıklama**
SERI NUMARASı	Sertifika seri numarası
POSTA	E-posta adresi
Deprecated P	E-posta adresi (MAIL tercihine göre kullanımdan kaldırıldı)

Çizelge 1. Bir SSLPEER süzgecinde kullanılabilecek DN ' de bulunan öznitelik tipleri (devamı var)	
Öznitelik tipi	**Açıklama**
UID ya da USERID	Kullanıcı kimliği
CN	Ortak Ad
T	Unvan
Kuruluş Birimi	Kuruluş Birimi adı
DAĞITIM MERKEZİ	Etki alanı bileşeni
O	Kuruluş adı
Sokak	Açık/İlk adres satırı
L	İlçe adı
ST (ya da SP ya da S)	Eyalet ya da İl adı
PC	Posta kodu/posta kodu
C	Ülke
UNSTRUCTUREDNAME	Anasistem adı
YAPıLMAMış ELBISE	IP adresi
DNQ	Ayırt edici ad niteleyicisi

X.509 standardı, genellikle DN ' nin bir parçasını oluşturmayan, ancak sayısal sertifika için isteğe bağlı uzantılar sağlayabilen diğer öznitelikleri tanımlar.

X.509 standardı, bir dizilim biçiminde belirtilecek bir DN sağlar. Örneğin:

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

Ortak Ad (CN), bir kullanıcıyı ya da başka bir varlığı (örneğin, bir web sunucusu) açıklayabilir.

DN, birden çok kuruluş birimi ve DC özniteliği içerebilir. Diğer özniteliklerin her birinin yalnızca bir eşgörünümüne izin verilir. Kuruluş Birimi girdilerinin sırası önemlidir: sıra, en üst düzey birim önce olacak şekilde Kuruluş Birimi adlarının bir sıradüzenini belirtir. DC girişlerinin sırası da önemlidir.

IBM MQ , bazı bozuk biçimli DN ' leri tolere eder. Daha fazla bilgi için bkz. IBM MQ SSLPEER değerleri kuralları.

İlgili kavramlar

“Dijital sertifikada ne var?” sayfa 13
Sayısal sertifikalar, X.509 standardında belirlendiği şekilde belirli bilgi parçaları içerir.

Sertifika yetkilisinden kişisel sertifika alınması

Güvenilir bir dış sertifika kuruluşundan (CA) sertifika alabilirsiniz.

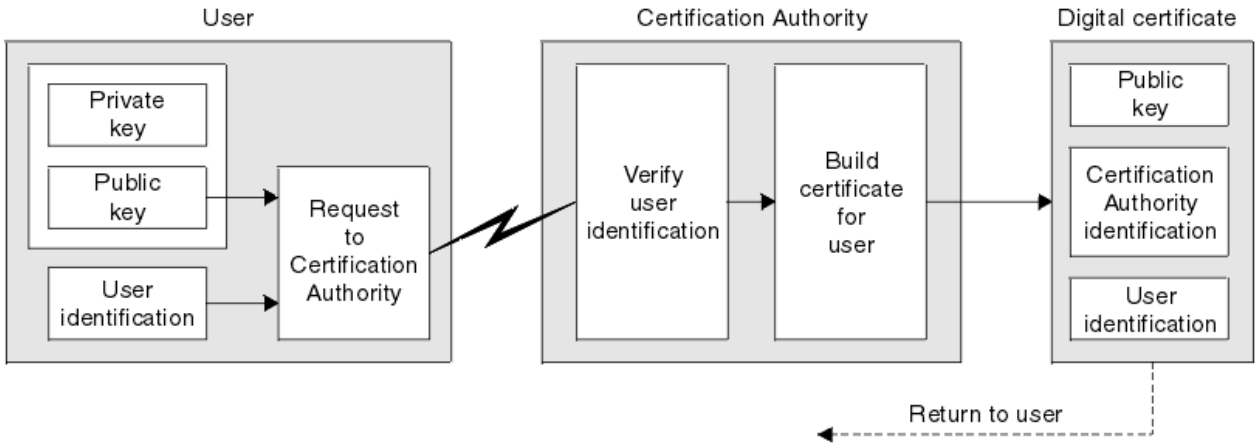
Sertifika isteği biçiminde bir sertifika kuruluşuna bilgi göndererek sayısal sertifika edinebilirsiniz. X.509 standardı, bu bilgiler için bir biçim tanımlar, ancak bazı CA ' ların kendi biçimi vardır. Sertifika istekleri genellikle sisteminizin kullandığı sertifika yönetimi aracı tarafından oluşturulur; örneğin:

- Multi Çoklu platformlarüzerinde **strmqikm** komutu (iKeyman aracı) ve AIX, Linux, and Windowsüzerinde **runmqckm** ve **runmqakm** komutları.
- z/OS RACF açık z/OS.

Bilgiler, Ayırt Edici Adınızı ve ortak anahtarınızı içerir. Sertifika yönetimi aracınız sertifika isteğinizi oluşturduğunda, güvenli tutmanız gereken özel anahtarınızı da oluşturur. Özel anahtarınızı hiçbir zaman dağıtma.

Sertifika yetkilisi isteğinizi aldığında, sertifika oluşturmadan önce kimliğinizi doğrular ve kişisel sertifika olarak size geri verir.

Şekil 3 sayfa 16 içinde bir CA ' dan sayısal sertifika alma işlemi gösterilmektedir.



Şekil 3. Dijital sertifika alınması

Çizgede:

- Kullanıcı kimliği, Konu Ayırt Edici Adınızı içerir.
- Sertifika Yetkilisi tanımlaması, sertifikayı veren CA ' nın Ayırt Edici Adını içerir.

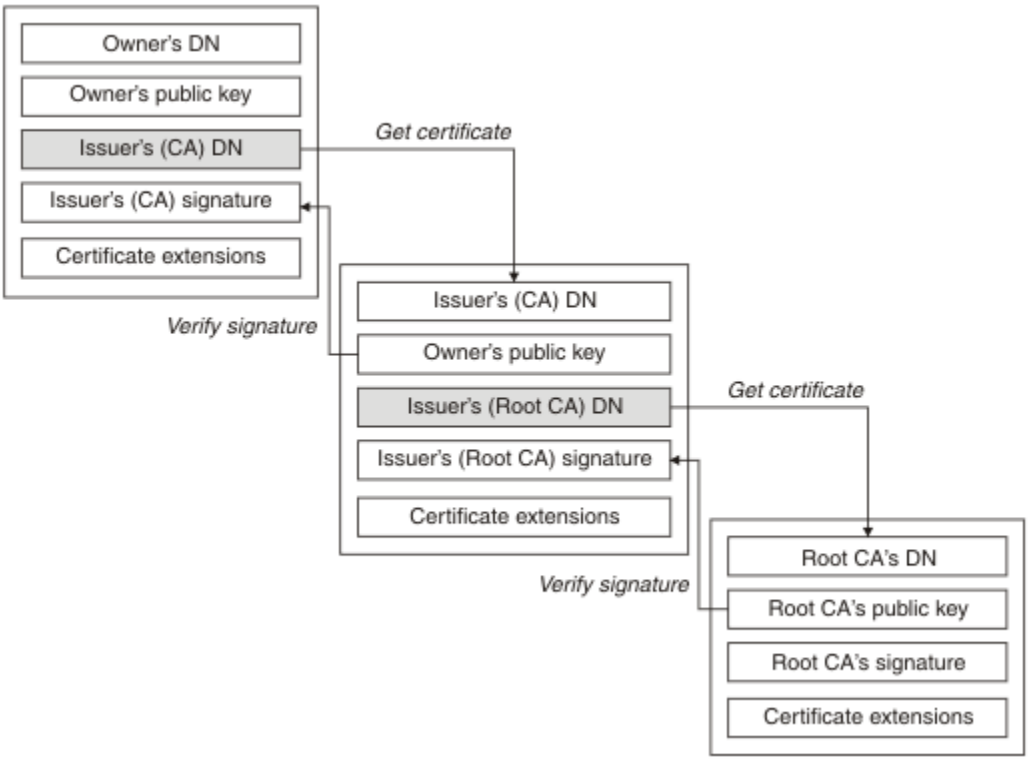
Sayısal sertifikalar, çizgede gösterilenler dışında ek alanlar içerir. Dijital sertifikadaki diğer alanlarla ilgili daha fazla bilgi için bkz. “Dijital sertifikada ne var?” sayfa 13.

Sertifika zincirleri nasıl çalışır?

Başka bir varlığa ilişkin sertifikayı aldığınızda, *kök CA* sertifikasını almak için *sertifika zinciri* kullanmanız gerekebilir.

*Sertifikasyon yolu*olarak da bilinen sertifika zinciri, bir varlığın kimliğini doğrulamak için kullanılan sertifikaların bir listesidir. Zincir ya da yol, o varlığın sertifikasıyla başlar ve zincirdeki her sertifika, zincirdeki bir sonraki sertifika tarafından tanımlanan varlık tarafından imzalanır. Zincir bir kök sertifika kuruluşu (CA) sertifikasıyla sona erer. Kök CA sertifikası her zaman sertifika yetkilisi (CA) tarafından imzalanır. Kök sertifika kuruluşu (CA) sertifikasına ulaşılıncaya kadar, zincirdeki tüm sertifikaların imzaları doğrulanmalıdır.

Şekil 4 sayfa 17 içinde, sertifika sahibinden kök sertifika kuruluşuna kadar olan bir sertifikasyon yolu gösterilmektedir; burada güven zinciri başlar.



Şekil 4. Güven zinciri

Her sertifika bir ya da daha fazla uzantı içerebilir. Bir CA ' ya ait bir sertifika genellikle, isCA işaretinin diğer sertifikaları imzalamasına izin verildiğini belirtmek için ayarlanmış BasicConstraints uzantısını içerir.

Sertifikalar artık geçerli olmadığında

Sayısal sertifikaların süresi sona erebilir ya da iptal edilebilir.

Dijital sertifikalar sabit bir dönem için verilir ve süre bitim tarihinden sonra geçerli değildir.

Sertifikalar, aşağıdakiler de dahil olmak üzere çeşitli nedenlerle iptal edilebilir:

- Sahip farklı bir kuruluşa taşındı.
- Özel anahtar artık gizli değil.

IBM MQ , bir Online Certificate Status Protocol (OCSP) yanıtlayıcısına (yalnızca AIX, Linux, and Windows üzerinde) istek göndererek bir sertifikanın iptal edilip edilmediğini denetleyebilir. Diğer bir seçenek olarak, LDAP sunucusundaki bir Sertifika İptal Listesine (CRL) erişebilirler. OCSP iptal ve CRL bilgileri bir Sertifika Yetkilisi tarafından yayınlanır. Daha fazla bilgi için "İptal edilen sertifikalarla çalışma" sayfa 355 başlıklı konuya bakın.

Genel anahtar altyapısı (PKI)

Açık Anahtar Altyapısı (PKI), bir işlemde yer alan tarafların kimliğini doğrulamak için açık anahtar şifrelemesi kullanımını destekleyen bir tesis, politika ve hizmet sistemidir.

Bir Genel Anahtar Altyapısının bileşenlerini tanımlayan tek bir standart yoktur, ancak PKI genellikle sertifika yetkililerini (CA) ve Kayıt Yetkililerini (RA) içerir. CA ' lar aşağıdaki hizmetleri sağlar:

- Dijital sertifikaları verme
- Dijital sertifikaların geçerliliği denetleniyor
- Dijital sertifikaları iptal etme
- Ortak anahtarların dağıtılması

X.509 standartları, sektör standardı Genel Anahtar Altyapısı için temel sağlar.

Dijital sertifikalar ve sertifika yetkilileriyle (CA) ilgili ek bilgi için bkz. “dijital sertifikalar” sayfa 13 . RNA 'lar, sayısal sertifikalar istendiğinde sağlanan bilgileri doğrulayın. RA bu bilgileri doğruysa, CA istekte bulunana bir sayısal sertifika verebilir.

PKI, dijital sertifikaları ve ortak anahtarları yönetmek için araçlar da sağlayabilir. Bir PKI, dijital sertifikaların yönetilmesi için *güven sıradüzeni* olarak tanımlanır, ancak çoğu tanımlama ek hizmetler içerir. Bazı tanımlar şifreleme ve dijital imza hizmetlerini içerir, ancak bu hizmetler PKI ' nin çalışması için gerekli değildir.

Şifreleme güvenliği iletişim kuralları: TLS

Şifreleme protokolleri güvenli bağlantılar sağlayarak iki tarafın gizlilik ve veri bütünlüğü ile iletişim kurmasını sağlar. TLS (Transport Layer Security; İletim Katmanı Güvenliği) protokolü SSL ' nin (Secure Sockets Layer; Güvenli Yuva Katmanı Katmanı) protokolünden gelişti. IBM MQ TLS ' yi destekler.

Her iki iletişim kuralının birincil amacı, gizlilik (bazen *gizlilik* olarak adlandırılır), veri bütünlüğü, tanımlama ve dijital sertifikaları kullanarak kimlik doğrulama sağlamaktır.

İki iletişim kuralı benzer olsa da, farklılıklar SSL 3.0 ve TLS ' nin çeşitli sürümlerinin birlikte çalışmaması için yeterince önemlidir.

İlgili kavramlar

“IBM MQ içinde TLS güvenlik iletişim kuralları” sayfa 24
IBM MQ , ileti kanalları ve MQI kanalları için bağlantı düzeyinde güvenlik sağlamak üzere Transport Layer Security (TLS) iletişim kuralını destekler.

Aktarım Katmanı Güvenliği (TLS) kavramları

TLS iletişim kuralı, iki tarafın birbirini tanımlamasına ve doğrulamasına ve gizlilik ve veri bütünlüğü ile iletişim kurmasına olanak sağlar. TLS iletişim kuralı Netscape SSL 3.0 iletişim kuralından gelişti, ancak TLS ve SSL birlikte çalışmaz.

TLS iletişim kuralı, internet üzerinden iletişim güvenliği sağlar ve istemci/sunucu uygulamalarının gizli ve güvenilir bir şekilde iletişim kurmasına izin verir. Protokollerin iki katmanı vardır: bir Kayıt Protokolü ve bir Tokalaşma Protokolü ve bunlar, TCP/IP gibi bir taşıma protokolünün üzerinde katmanlıdır. Her ikisi de asimetrik ve simetrik kriptografi tekniklerini kullanır.

TLS bağlantısı, TLS istemcisi olan bir uygulama tarafından başlatılır. Bağlantıyı alan uygulama TLS sunucusu olur. Her yeni oturum TLS protokolleri tarafından tanımlanan bir el sıkışma ile başlar.

IBM MQ tarafından desteklenen CipherSpecs ' in tam listesi “CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432adresinde bulunmaktadır.

SSL iletişim kuralı hakkında daha fazla bilgi için https://developer.mozilla.org/docs/Mozilla/Projects/NSSadresinde sağlanan bilgilere bakın. TLS iletişim kuralı hakkında daha fazla bilgi için https://www.ietf.org adresindeki Internet Engineering Task Force web sitesinde TLS Çalışma Grubu tarafından sağlanan bilgilere bakın.

SSL/TLS el sıkışması hakkında genel bilgiler

SSL/TLS anlaşması, TLS istemcisinin ve sunucusunun iletişim kurdukları gizli anahtarları oluşturmasını sağlar.

Bu bölümde, TLS istemcisinin ve sunucusunun birbiriyle iletişim kurmasını sağlayan adımların bir özeti sağlanır.

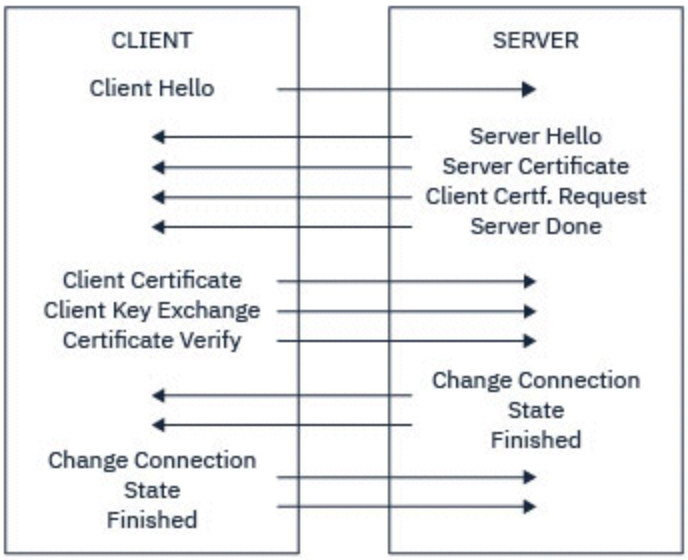
- Kullanılacak protokolün sürümünü kabul edin.
- Şifreleme algoritmalarını seçin.
- Dijital sertifikaları değiş tokuş ederek ve doğrulayarak birbirinizi doğrulayın.
- Anahtar dağıtım sorununu önleyen paylaşılan bir gizli anahtar oluşturmak için asimetrik şifreleme tekniklerini kullanın. Daha sonra TLS, asimetrik şifrelemeden daha hızlı olan iletilerin simetrik şifrelemesi için paylaşılan anahtarı kullanır.

Şifreleme algoritmaları ve dijital sertifikalar hakkında daha fazla bilgi için ilgili bilgilere bakın.

Genel bakışta, TLS el sıkışmasında yer alan adımlar aşağıdaki gibidir:

1. TLS istemcisi, TLS sürümü ve istemcinin tercih sırasına göre istemci tarafından desteklenen CipherSuites gibi şifreleme bilgilerini listeleyen bir "client hello" iletisi gönderir. İleti, sonraki hesaplamalarda kullanılan rasgele bir byte dizgisi de içerir. Bu iletişim kuralı, "client hello" istemcisinin istemci tarafından desteklenen veri sıkıştırma yöntemlerini içermesini sağlar.
2. TLS sunucusu, istemci, oturum tanıtıcısı ve başka bir rasgele bayt dizgisi tarafından sağlanan listeden sunucu tarafından seçilen CipherSuite ' i içeren bir "sunucu merhaba" iletisiyle yanıt verir. Sunucu ayrıca dijital sertifikasını da gönderir. Sunucu, istemci kimlik doğrulaması için sayısal sertifika gerektiriyorsa, desteklenen sertifika tiplerinin ve kabul edilebilir Sertifika Yetkililerinin (CA) Ayırt Edici Adlarını içeren bir "istemci sertifikası isteği" gönderir.
3. TLS istemcisi, sunucunun dijital sertifikasını doğrular. Daha fazla bilgi için bkz “TLS ' nin tanımlama, kimlik doğrulama, gizlilik ve bütünlüğü nasıl sağladığı” sayfa 20.
4. TLS istemcisi, hem istemcinin hem de sunucunun sonraki ileti verilerini şifrelemek için kullanılacak gizli anahtarı hesaplamasını sağlayan rasgele bayt dizgisini gönderir. Rasgele bayt dizgisinin kendisi, sunucunun genel anahtarıyla şifrelenir.
5. TLS sunucusu bir "istemci sertifikası isteği"gönderdiyse, istemci, istemcinin dijital sertifikasıyla birlikte istemcinin özel anahtarıyla şifrelenmiş rasgele bir bayt dizgisi ya da "sayısal sertifika yok uyarısı"gönderir. Bu uyarı yalnızca bir uyarıdır, ancak bazı uygulamalarda istemci kimlik doğrulaması zorunluysa anlaşma başarısız olur.
6. TLS sunucusu, istemcinin sertifikasını doğrular. Daha fazla bilgi için bkz “TLS ' nin tanımlama, kimlik doğrulama, gizlilik ve bütünlüğü nasıl sağladığı” sayfa 20.
7. TLS istemcisi, sunucuya, el sıkışmasının istemci kısmının tamamlandığını belirten gizli anahtarla şifrelenen bir "bitti" iletisi gönderir.
8. TLS sunucusu, istemciye, el sıkışmasının sunucu kısmının tamamlandığını belirten gizli anahtarla şifrelenen bir "bitti" iletisi gönderir.
9. TLS oturumu süresince, sunucu ve istemci artık paylaşılan gizli anahtarla simetrik olarak şifrelenmiş iletileri değiş tokuş edebilir.

Şekil 5 sayfa 19 içinde TLS el sıkışması gösterilmektedir.



Şekil 5. TLS el sıkışması hakkında genel bilgiler

TLS ' nin tanımlama, kimlik doğrulama, gizlilik ve bütünlüğü nasıl sağladığı

Hem istemci hem de sunucu kimlik doğrulaması sırasında, verilerin asimetrik anahtar çiftindeki anahtarlardan biriyle şifrelenmesini ve çiftin diğer anahtarıyla şifresinin çözülmesini gerektiren bir adım vardır. Bütünlük sağlamak için bir ileti özeti kullanılır.

TLS el sıkışmasına dahil olan adımlara genel bakış için bkz. “SSL/TLS el sıkışması hakkında genel bilgiler” sayfa 18.

TLS ' nin kimlik doğrulamasını nasıl sağladığı

Sunucu kimlik doğrulaması için istemci, gizli anahtarı hesaplamak için kullanılan verileri şifrelemek için sunucunun genel anahtarını kullanır. Sunucu, gizli anahtarı yalnızca doğru özel anahtarla bu verilerin şifresini çözebiliyorsa oluşturabilir. Rasgele bayt dizgisinin kendisi, sunucunun genel anahtarıyla (genel bakışta “4” sayfa 19 . adım) şifrelenir.

İstemci kimlik doğrulaması için sunucu, istemci sertifikasındaki ortak anahtarı kullanarak, el sıkışmasının “5” sayfa 19 . adımı sırasında istemcinin gönderdiği verilerin şifresini çözdü. Gizli anahtarla şifrelenmiş tamamlanan iletilerin değişimi (genel bakışta “7” sayfa 19 ve “8” sayfa 19 adımları), kimlik doğrulamasının tamamlandığını doğrular.

Kimlik doğrulama adımlarından herhangi biri başarısız olursa, anlaşma başarısız olur ve oturum sonlandırılır.

TLS el sıkışması sırasında dijital sertifikaların değişimi, kimlik doğrulama sürecinin bir parçasıdır. Sertifikaların kimliğe bürünmeye karşı nasıl koruma sağladığına ilişkin daha fazla bilgi için ilgili bilgilere bakın. Gerekli sertifikalar aşağıdaki gibidir; CA X , TLS istemcisine sertifikayı verir ve CA Y , TLS sunucusuna sertifikayı verir:

Yalnızca sunucu kimlik doğrulaması için TLS sunucusu şunları gerektirir:

- CA Y tarafından sunucuya verilen kişisel sertifika
- Sunucunun özel anahtarı

ve TLS istemcisinin aşağıdaki gereksinimleri karşılar:

- CA sertifikası Y

TLS sunucusu istemci kimlik doğrulaması gerektiriyorsa, sunucu istemcinin dijital sertifikasını istemciye kişisel sertifikayı veren CA ' ya ilişkin genel anahtarla doğrulayarak istemcinin kimliğini doğrular (bu durumda CA X). Sunucu ve istemci kimlik doğrulaması için sunucunun aşağıdakileri yapması gerekir:

- CA Y tarafından sunucuya verilen kişisel sertifika
- Sunucunun özel anahtarı
- CA sertifikası X

ve müşterinin aşağıdakilere gereksinimi vardır:

- CA X tarafından istemciye verilen kişisel sertifika
- İstemcinin özel anahtarı
- CA sertifikası Y

Hem TLS sunucusu hem de istemci, kök sertifika kuruluşu (CA) sertifikasına ilişkin bir sertifika zinciri oluşturmak için diğer CA sertifikalarına gereksinim duyabilirler. Sertifika zincirleriyle ilgili daha fazla bilgi için ilgili bilgilere bakın.

Sertifika doğrulaması sırasında neler oluyor?

Genel bakış “3” sayfa 19 ve “6” sayfa 19 adımlarında belirtildiği gibi, TLS istemcisi sunucunun sertifikasını doğrular ve TLS sunucusu istemcinin sertifikasını doğrular. Bu doğrulamanın dört yönü vardır:

1. Dijital imza denetlenir (bkz. “SSL/TLS ' de dijital imzalar” sayfa 22).

2. Sertifika zinciri denetlenir; ara sertifika yetkilisi sertifikalarınız olmalıdır (bkz. “Sertifika zincirleri nasıl çalışır?” sayfa 16).
3. Süre bitimi ve etkinleştirme tarihleri ve geçerlilik süresi denetlenir.
4. Sertifikanın iptal durumu denetlenir (bkz. “İptal edilen sertifikalarla çalışma” sayfa 355).

Gizli anahtar sıfırlaması

TLS anlaşması sırasında TLS istemcisi ile sunucu arasındaki verileri şifrelemek için bir *gizli anahtar* oluşturulur. Gizli anahtar, düz metni okunamayan şifreli metne ve şifreli metni düz metne dönüştürmek için verilere uygulanan matematiksel formülde kullanılır.

Gizli anahtar, el sıkışmasının bir parçası olarak gönderilen rastgele metinden oluşturulur ve düz metni şifreli metne şifrelemek için kullanılır. Gizli anahtar, bir iletinin değiştirilip değiştirilmediğini belirlemek için kullanılan MAC (İleti Doğrulama Kodu) algoritmasında da kullanılır. Ek bilgi için bkz. “İleti özetleri ve dijital imzalar” sayfa 12 .

Gizli anahtar bulunursa, bir iletinin düz metninin şifresi şifreli metinden çözülebilir ya da ileti özeti, mesajların algılanmadan değiştirilmesine izin vererek hesaplanabilir. Karmaşık bir algoritma için bile, düz metin sonunda şifreli metne olası her matematiksel dönüşüm uygulanarak keşfedilebilir. Gizli anahtar bozulursa çözülebilecek ya da değiştirilebilecek veri miktarını en aza indirmek için gizli anahtar düzenli olarak yeniden görüşülebilir. Gizli anahtar yeniden anlaşıldığında, önceki gizli anahtar artık yeni gizli anahtarla şifrelenmiş verilerin şifresini çözmek için kullanılamaz.

TLS ' nin gizliliği nasıl sağladığı

TLS, ileti gizliliğini sağlamak için simetrik ve asimetrik şifrelemenin bir birleşimini kullanır. TLS anlaşması sırasında TLS istemcisi ve sunucusu, yalnızca bir oturum için kullanılacak bir şifreleme algoritmasını ve paylaşılan bir gizli anahtarı kabul eder. TLS istemcisi ve sunucusu arasında iletilen tüm iletiler, bu algoritma ve anahtar kullanılarak şifrelenir ve mesajın algılansa bile gizli kalmasını sağlar. TLS, paylaşılan gizli anahtarı taşırken asimetrik şifreleme kullandığından, anahtar dağıtım sorunu yoktur. Şifreleme teknikleriyle ilgili daha fazla bilgi için bkz. “Şifreleme” sayfa 11.

TLS ' nin bütünlüğü nasıl sağladığı

TLS, bir ileti özeti hesaplayarak veri bütünlüğü sağlar. Daha fazla bilgi için bkz. “İletilerin veri bütünlüğü” sayfa 489.

TLS kullanımı, kanal tanımlamanızdaki CipherSpec ' in “CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432çizelgesinde açıklandığı gibi bir hash algoritması kullanması koşuluyla, veri bütünlüğünü sağlar.

Özellikle, veri bütünlüğü bir sorunsa, hash algoritması "Yok" olarak listelenen bir CipherSpec seçmekten kaçınmalısınız. MD5 kullanımı da artık çok eski olduğundan ve en pratik amaçlar için artık güvenli olmadığından şiddetle önerilmemektedir.

CipherSpecs ve CipherSuites

Kriptografik güvenlik protokolleri, güvenli bir bağlantı tarafından kullanılan algoritmalar üzerinde anlaşmalıdır. CipherSpecs ve CipherSuites algoritmaların belirli birleşimlerini tanımlar.

CipherSpec , şifreleme algoritması ve İleti Doğrulama Kodu (MAC) algoritmasının birleşimini tanımlar. İletişim kurabilmek için TLS bağlantısının her iki ucu da aynı CipherSpec üzerinde anlaşmalıdır.

IBM MQ , TLS1.3 ve TLS1.2 iletişim kurallarını ve CipherSpecs' i destekler. Ancak, gerekiyorsa, kullanımdan kaldırılan CipherSpecsöğesini etkinleştirebilirsiniz.

Aşağıdakilere ilişkin bilgi için bkz: “CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432

- IBM MQ tarafından desteklenen CipherSpecs (Şifre Belirtimleri)
- Kullanımdan kaldırılan SSL 3.0 ve TLS 1.0 CipherSpecs ' i nasıl etkinleştirdiniz?

Önemli: IBM MQ kanallarıyla uğraşırken CipherSpeckullanılır. Java kanalları, JMS kanalları ya da MQTT kanalları ile ilgilenirken bir CipherSuitebelirtirsiniz.

CipherSpecsile ilgili daha fazla bilgi için bkz. “CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432.

CipherSuite , TLS bağlantısı tarafından kullanılan bir şifreleme algoritması takımıdır. Bir takım üç ayrı algoritmalardan oluşur:

- Tokalaşma sırasında kullanılan anahtar değiş tokuşu ve doğrulama algoritması
- Verileri şifrelemek için kullanılan şifreleme algoritması
- İleti özeti oluşturmak için kullanılan MAC (İleti Doğrulama Kodu) algoritması

Takımın her bileşeni için birkaç seçenek vardır, ancak TLS bağlantısı için belirtildiğinde yalnızca belirli birleşimler geçerlidir. Geçerli bir CipherSuite adı, kullanılan algoritmaların birleşimini tanımlar. Örneğin, CipherSuite TLS_RSA_WITH_AES_128_CBC_SHA aşağıdakileri belirtir:

- RSA anahtar değiş tokuşu ve doğrulama algoritması
- 128 bitlik anahtar ve şifre blok zincirleme (CBC) kipini kullanan AES şifreleme algoritması
- SHA-1 İleti Kimlik Doğrulama Kodu (MAC)

SSL/TLS ' de dijital imzalar

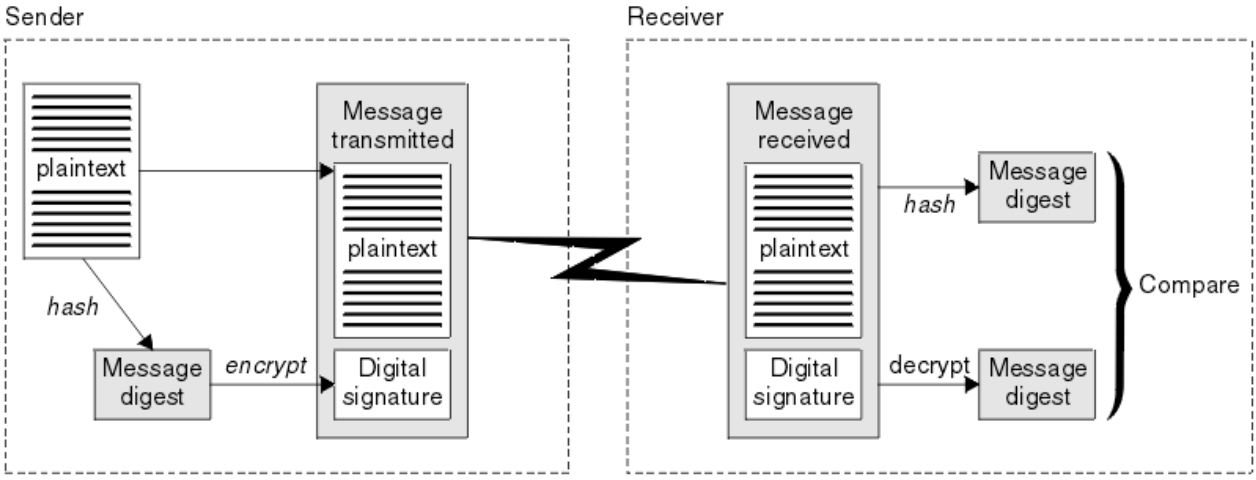
Dijital imza, bir iletinin gösteriminin şifrelenmesiyle oluşturulur. Şifreleme, imza sahibinin özel anahtarını kullanır ve verimlilik için genellikle mesajın kendisinden ziyade bir ileti özeti üzerinde çalışır.

Dijital imzalar, imzalanmakta olan verilerin, imzalanmakta olan belgenin içeriğine bağlı olmayan el yazısı imzalarından farklı olarak, imzalanmasına göre değişir. İki farklı ileti aynı varlık tarafından dijital olarak imzalandıysa, iki imza farklıdır, ancak her iki imza da aynı ortak anahtarla (yani, iletileri imzalayan varlığın genel anahtarıyla) doğrulanabilir.

Dijital imza sürecinin adımları aşağıdaki gibidir:

1. Gönderen, bir ileti özeti hesaplar ve sonra gönderenin özel anahtarını kullanarak dijital imzayı oluşturarak özeti şifreler.
2. Gönderen, dijital imzayı iletir.
3. Alıcı, gönderenin genel anahtarını kullanarak dijital imzanın şifresini çözer ve gönderenin ileti özetini yeniden oluşturur.
4. Alıcı, alınan ileti verilerinden bir ileti özeti hesaplar ve iki özetinin aynı olduğunu doğrular.

Şekil 6 sayfa 22 içinde bu işlem gösterilmektedir.



Şekil 6. Dijital imza süreci

Sayısal imza doğrulanır ve alıcı şunu bilir:

- İleti, iletim sırasında değiştirilmedi.

- İleti, iletiyi gönderdiğini iddia eden varlık tarafından gönderildi.

Dijital imzalar, bütünlük ve kimlik doğrulama hizmetlerinin bir parçasıdır. Dijital imzalar da kaynak kanıtı sağlar. Yalnızca gönderen, gönderenin mesajı gönderen olduğuna dair güçlü kanıtlar sağlayan özel anahtarı bilir.

Not: İletideki bilgilerin gizliliğini koruyan iletinin kendisini de şifreleyebilirsiniz.

Federal Bilgi İşleme Standartları

ABD hükümeti, veri şifreleme de dahil olmak üzere, BT sistemleri ve güvenlik konusunda teknik öneriler üretiyor. Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), BT sistemleri ve güvenliğiyle ilgili önemli bir kurumdur. NIST, Federal Bilgi İşleme Standartları (FIPS) dahil olmak üzere öneriler ve standartlar üretir.

Bu standartlardan önemli bir tanesi, güçlü şifreleme algoritmalarının kullanılmasını gerektiren FIPS 140-2 'dir. FIPS 140-2 ayrıca, paketleri geçiş sırasında değişikliğe karşı korumak için kullanılacak hash algoritmalarına ilişkin gereksinimleri belirler.

Not: AIX, Linux, and Windowsişletim sistemlerinde IBM MQ , IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, IBM Crypto for C (ICC) sertifikasını görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu İşlem listesindeki NIST CMVP modüllerindearanarak görüntülenebilir.

IBM MQ , yapılandırıldığında FIPS 140-2 desteği sağlar.

Zaman içinde analistler, var olan şifreleme ve hash algoritmalarına karşı saldırılar geliştirir. Bu saldırılara karşı koymak için yeni algoritmalar benimsendi. FIPS 140-2, bu değişiklikleri dikkate alacak şekilde düzenli olarak güncellenir.

İlgili kavramlar

“Ulusal Güvenlik Ajansı (NSA) Suite B Şifrelemesi” sayfa 23
Amerika Birleşik Devletleri hükümeti, veri şifrelemesi de dahil olmak üzere BT sistemleri ve güvenliği konusunda teknik öneriler üretiyor. ABD Ulusal Güvenlik Ajansı (NSA), Suite B standardında bir dizi birlikte çalışabilir şifreleme algoritması önermektedir.

Ulusal Güvenlik Ajansı (NSA) Suite B Şifrelemesi

Amerika Birleşik Devletleri hükümeti, veri şifrelemesi de dahil olmak üzere BT sistemleri ve güvenliği konusunda teknik öneriler üretiyor. ABD Ulusal Güvenlik Ajansı (NSA), Suite B standardında bir dizi birlikte çalışabilir şifreleme algoritması önermektedir.

Takım B standardı, yalnızca belirli bir güvenli şifreleme algoritması kümesinin kullanıldığı bir işlem kipini belirtir. Takım B standardı aşağıdakileri belirtir:

- Şifreleme algoritması (AES)
- Anahtar değişim algoritması (Eliptik Eğri Diffie-Hellman, ECDH olarak da bilinir)
- Dijital imza algoritması (ECDSA olarak da bilinen Eliptik Eğri Sayısal İmza Algoritması)
- Hash algoritmaları (SHA-256 ya da SHA-384)

Ayrıca, IETF RFC 6460 standardı, Suite B standardına uymak için gerekli ayrıntılı uygulama yapılandırmasını ve davranışını tanımlayan Suite B uyumlu profilleri belirtir. İki profili tanımlar:

1. TLS 1.2ile kullanılmak üzere Suite B uyumlu bir profil. Suite B uyumlu işlem için yapılandırıldığında, yalnızca listelenen sınırlı şifreleme algoritmaları kümesi kullanılır.
2. TLS 1.0 ya da TLS 1.1ile kullanılmak üzere bir geçiş profili. Bu profil, Suite B uyumlu olmayan sunucularla birlikte çalışabilirlik sağlar. Suite B geçiş işlemi için yapılandırıldığında, ek şifreleme ve hash algoritmaları kullanılabilir.

Suite B standardı, güvenli bir güvenlik düzeyi sağlamak için etkinleştirilmiş şifreleme algoritmaları kümesini kısıtladığı için kavramsal olarak FIPS 140-2 'ye benzer.

AIX, Linux, and Windows sistemlerinde IBM MQ, Suite B uyumlu TLS 1.2 profiline uymak için yapılandırılabilir, ancak Suite B geçiş profilini desteklemez. Daha fazla bilgi için bkz. “IBM MQ içinde NSA Suite B Şifreleme” sayfa 41.

İlgili başvurular

“Federal Bilgi İşleme Standartları” sayfa 23
ABD hükümeti, veri şifreleme de dahil olmak üzere, BT sistemleri ve güvenlik konusunda teknik öneriler üretiyor. Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), BT sistemleri ve güvenliğiyle ilgili önemli bir kurumdur. NIST, Federal Bilgi İşleme Standartları (FIPS) dahil olmak üzere öneriler ve standartlar üretir.

IBM MQ güvenlik mekanizmaları

Bu konu topluluğu, IBM MQ içinde çeşitli güvenlik kavramlarını uygulayan belirli mekanizmaları açıklar.

IBM MQ içinde TLS güvenlik iletişim kuralları

IBM MQ , ileti kanalları ve MQI kanalları için bağlantı düzeyinde güvenlik sağlamak üzere Transport Layer Security (TLS) iletişim kuralını destekler.

İleti kanalları ve MQI kanalları, bağlantı düzeyi güvenliği sağlamak için TLS iletişim kuralını kullanabilir. Çağıran MCA bir TLS istemcisidir ve yanıt veren MCA bir TLS sunucusudur.

IBM MQ , TLS iletişim kuralının 1.2 ve 1.3 sürümlerini destekler. SSL ' nin yanı sıra önceki TLS sürümleri varsayılan olarak etkinleştirilmez, ancak gerekirse etkinleştirilebilir. Kanal tanımlamasının bir parçası olarak bir CipherSpec sağlayarak TLS iletişim kuralı tarafından kullanılan şifreleme algoritmalarını belirtebilirsiniz.

Kullanımdan kaldırılanlar için IBM MQ ve “Kullanımdan kaldırılan CipherSpecs” sayfa 448 tarafından desteklenen CipherSpecs ' in bir listesi için bkz. “CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432 .

SECPROT ve SSLCIPH değiştirgelerini, bir kanalda kullanılan CipherSpec güvenlik protokolünü görüntülemek için kullanabilirsiniz.

Bir ileti kanalının her ucunda ve bir MQI kanalının sunucu sonunda MCA, bağlı olduğu kuyruk yöneticisi adına hareket eder. TLS el sıkışması sırasında MCA, kuyruk yöneticisinin dijital sertifikasını kanalın diğer ucundaki ortağı MCA ' ya gönderir. Bir MQI kanalının istemci ucundaki IBM MQ kodu, IBM MQ istemci uygulamasının kullanıcısı adına hareket eder. TLS anlaşması sırasında IBM MQ kodu, kullanıcının dijital sertifikasını MQI kanalının sunucu ucundaki MCA ' ya gönderir.

Kanalın sunucu tarafında SSLCAUTH (REQUIRED) belirtilmedikçe, kuyruk yöneticilerinin ve IBM MQ istemcisi kullanıcılarının TLS istemcileri olarak hareket ederken kendileriyle ilişkilendirilmiş kişisel dijital sertifikalara sahip olması gerekmez.

Sayısal sertifikalar bir *anahtar havuzunda*saklanır. Kuyruk yöneticisi özniteliği **SSLKeyRepository** , kuyruk yöneticisinin sayısal sertifikasını tutan anahtar havuzunun yerini belirtir. IBM MQ istemci sisteminde MQSSLKEYR ortam değişkeni, kullanıcının sayısal sertifikasını tutan anahtar havuzunun yerini belirtir. Alternatif olarak, bir IBM MQ istemci uygulaması MQCONNX çağrısında TLS yapılandırma seçenekleri yapısının (MQSCO) **KeyRepository** alanında konumunu belirtebilir. Anahtar havuzları hakkında daha fazla bilgi ve bunların nerede bulunacaklarını nasıl belirteceğinizi öğrenmek için ilgili konulara bakın.

TLS desteği

IBM MQ , tüm platformlarda TLS 1.2 ve TLS 1.3 desteği sağlar. TLS iletişim kuralı hakkında daha fazla bilgi için alt konulardaki bilgilere bakın.

Java ve JMS istemcileri
Bu istemciler TLS desteği sağlamak için JVM ' yi kullanır.

AIX, Linux, and Windows
TLS desteği IBM MQile kurulur.

IBM i
TLS desteği, IBM i işletim sisteminin ayrılmaz bir parçasıdır.

z/OS
TLS desteği, z/OS işletim sisteminin ayrılmaz bir parçasıdır. z/OS üzerinde TLS desteği, *Sistem SSL*olarak bilinir.

IBM MQ TLS desteğine ilişkin önkoşullar hakkında bilgi için bkz. IBM MQ için Sistem Gereksinimleri.

İlgili kavramlar
“Şifreleme güvenliği iletişim kuralları: TLS” sayfa 18
Şifreleme protokolleri güvenli bağlantılar sağlayarak iki tarafın gizlilik ve veri bütünlüğü ile iletişim kurmasını sağlar. TLS (Transport Layer Security; İletim Katmanı Güvenliği) protokolü SSL ' nin (Secure Sockets Layer; Güvenli Yuva Katmanı Katmanı) protokolünden gelişti. IBM MQ TLS ' yi destekler.

SSL/TLS anahtar havuzu
Karşılıklı olarak kimliği doğrulanmış bir TLS bağlantısı, bağlantının her sonunda bir anahtar havuzu gerektirir. Anahtar havuzu dijital sertifikaları ve özel anahtarları içerir.

Bu bilgiler, dijital sertifikalara ilişkin depoyu ve bunlarla ilişkili özel anahtarları açıklamak için *anahtar havuzu* genel terimini kullanır. Anahtar havuzuna, TLS ' yi destekleyen farklı platformlarda ve ortamlarda farklı adlar başvurulur:

- IBM iIBM iüzerinde: *sertifika deposu*
- Java ve JMSsistemlerinde: *keystore* ve *truststore*
- ALWAIX, Linux, and Windowsüzerinde: *anahtar veritabanı dosyası*
- z/OSz/OSsistemlerinde: *anahtarlık*

Daha fazla bilgi için bkz. “dijital sertifikalar” sayfa 13 ve “Aktarım Katmanı Güvenliği (TLS) kavramları” sayfa 18.

Karşılıklı olarak kimliği doğrulanmış bir TLS bağlantısı, bağlantının her sonunda bir anahtar havuzu gerektirir. Anahtar havuzu aşağıdaki sertifikaları ve istekleri içerebilir:

- Kuyruk yöneticisinin ya da istemcinin, bağlantının uzak ucundaki ortağından aldığı sertifikaları doğrulamasını sağlayan çeşitli Sertifikasyon Yetkilerinden alınan CA sertifikaları. Tek tek sertifikalar bir sertifika zincirinde olabilir.
- Bir Sertifika Yetkilisinden alınan bir ya da daha fazla kişisel sertifika. Her bir kuyruk yöneticisiyle ya da IBM MQ MQI clientile ayrı bir kişisel sertifika ilişkilendirin. Karşılıklı kimlik doğrulaması gerekiyorsa, TLS istemcisinde kişisel sertifikalar gereklidir. Karşılıklı kimlik doğrulaması gerekmiyorsa, istemcide kişisel sertifikalar gerekmez. Anahtar havuzu, her kişisel sertifikaya karşılık gelen özel anahtarı da içerebilir.
- Güvenilir bir CA sertifikası tarafından imzalanmayı bekleyen sertifika istekleri.

Anahtar havuzunuzu koruma hakkında daha fazla bilgi için bkz. “IBM MQ anahtar havuzlarını koruma” sayfa 26.

Anahtar havuzunun konumu, kullanmakta olduğunuz platforma bağlıdır:

IBM i**IBM i**
Anahtar havuzu bir sertifika depodur. Varsayılan sistem sertifika deposu, tümleşik dosya sisteminde (IFS) `/QIBM/UserData/ICSS/Cert/Server/Default` adresinde bulunur. IBM MQ , sertifika deposunun parolasını bir *parola saklama dosyasında*saklar. Örneğin, `QM1` kuyruk yöneticisine ilişkin saklama dosyası `/QIBM/UserData/mqm/qmgrs/QM1/ssl/Stash.sth`.

Diğer bir seçenek olarak, bunun yerine IBM i sistem sertifikası deposunun kullanılacağını belirtebilirsiniz. Bunu yapmak için, kuyruk yöneticisi **SSLKEYR** özniteliğinin değerini *SYSTEMolarak değiştirin. Bu değer, kuyruk yöneticisinin sistem sertifika deposunu kullanması gerektiğini ve kuyruk yöneticisinin Digital Certificate Manager (DCM) ile uygulama olarak kullanılmak üzere kaydedildiğini gösterir.

Sertifika deposu, kuyruk yöneticisine ilişkin özel anahtarı da içerir.

AIX, Linux, and Windows sistemleri

Anahtar havuzu bir anahtar veritabanı dosyasıdır. Örneğin, AIX and Linux' da, kuyruk yöneticisi `QM1` için varsayılan anahtar veritabanı dosyası şudur: `/var/mqm/qmgrs/QM1/ssl/key.kdb`. IBM MQ varsayılan konuma kurulduysa, Windows üzerindeki eşdeğer yol `C:\ProgramData\IBM\MQ\Qmgrs\QM1\ssl\key.kdb`olur.

V 9.3.0 V 9.3.0 IBM MQ anahtar veritabanı kütüğüne erişmek için anahtar veritabanına ilişkin parola belirtilmelidir. Bu, doğrudan ya da bir parola saklama dosyası aracılığıyla yapılabilir. Bir parola saklama kütüğü kullanılıyorsa, bu kütüğün aynı dizinde bulunması ve anahtar veritabanıyla aynı kütük köküne sahip olması ve `.sth`sonekiyle bitmesi gerekir; örneğin, `/var/mqm/qmgrs/QM1/ssl/key.sth`.

Not: PKCS #11 şifreleme donanım kartları, bir anahtar veritabanı dosyasında tutulan sertifikaları ve anahtarları içerebilir. PKCS #11 kartlarında sertifikalar ve anahtarlar tutulduğunda, IBM MQ için hem anahtar veritabanı dosyasına hem de parola saklama dosyasına erişim gerekir.

AIX, Linux, and Windows sistemlerinde anahtar veritabanı, kuyruk yöneticisiyle ya da IBM MQ MQI clientile ilişkili kişisel sertifikaya ilişkin özel anahtarı da içerir.

z/OS

Sertifikalar z/OSiçinde bir anahtarlık içinde tutulur.

Diğer dış güvenlik yöneticileri (ESM ' ler) de sertifikaları depolamak için anahtarlık kullanır.

Özel anahtarlar RACFtarafından yönetilir.

IBM MQ anahtar havuzlarını koruma

IBM MQ anahtar havuzu bir dosyadır. Anahtar havuzu dosyasına yalnızca istenen kullanıcının erişebildiğinden emin olun. Bu, izinsiz giriş yapan ya da başka bir yetkisiz kullanıcının anahtar havuzu dosyasını başka bir sisteme kopyalamasını ve daha sonra, istenen kullanıcının kimliğine bürünmesi için sistemde aynı kullanıcı kimliğini ayarlamasını önler.

Dosyalar üzerindeki izinler kullanıcının umask değerine ve hangi aracın kullanıldığına bağlıdır. Windowssistemlerinde IBM MQ hesaplar izin gerektirir `BypassTraverseChecking` ; bu, dosya yolundaki klasörlerin izinlerinin etkili olmadığı anlamına gelir.

Anahtar havuzu dosyalarının dosya izinlerini denetleyin ve dosyaların ve içeren klasörün okunabilir olmadığından, tercihen grup okunabilir olmadığından emin olun.

Anahtar deposunun salt okunur kılınması, hangi sistemi kullanırsanız kullanın, yalnızca yöneticinin bakım gerçekleştirmek için yazma işlemlerini etkinleştirmesine izin verilir.

Uygulamada, tüm anahtar depolarını, konumu ne olursa olsun ve parola korumalı olup olmadıklarını korumanız; anahtar havuzlarını korumanız gerekir.

Dijital sertifika etiketleri, gereksinimlerin anlaşılması

Dijital sertifikaları kullanmak için TLS ' yi ayarlarken, kullanılan platforma ve bağlanmak için kullandığınız yönteme bağlı olarak izlemeniz gereken belirli etiket gereksinimleri olabilir.

Sertifika etiketi nedir?

Sertifika etiketi, bir anahtar havuzunda saklanan sayısal bir sertifikayı gösteren benzersiz bir tanıtıcıdır ve anahtar yönetimi işlevlerini gerçekleştirirken belirli bir sertifikaya başvurmak için uygun bir kullanıcı tarafından okunabilir ad sağlar. Bir anahtar havuzuna ilk kez sertifika eklerken sertifika etiketini atarsanız.

Sertifika etiketi, sertifikanın **`Subject Distinguished Name`** ya da **`Subject Common Name`** alanlarından ayrıdır. **`Subject Distinguished Name`** ve **`Subject Common Name`** ' in sertifikanın kendisindeki alanlar olduğunu unutmayın. Bunlar, sertifika yaratıldığında tanımlanır ve değiştirilemez. Ancak gerekirse, bir dijital sertifikayla ilişkili etiketi değiştirebilirsiniz.

Sertifika etiketi sözdizimi

Bir sertifika etiketi, aşağıdaki koşullara sahip harfler, sayılar ve noktalama işaretleri içerebilir:

- Multi Sertifika etiketi en çok 64 karakter içerebilir.
- z/OS Sertifika etiketi en çok 32 karakter içerebilir.
- Sertifika etiketi boşluk içerebilir.
- Etiketler büyük ve küçük harfe duyarlıdır.
- EBCDIC katakana kullanan sistemlerde küçük harfli karakterler kullanamazsınız.

Sertifika etiketi değerlerine ilişkin ek gereksinimler aşağıdaki bölümlerde belirtilmiştir.

Sertifika etiketi nasıl kullanılıyor?

IBM MQ , TLS anlaşması sırasında gönderilen kişisel sertifikayı bulmak için sertifika etiketlerini kullanır. Bu, anahtar havuzunda birden çok kişisel sertifika olduğunda belirsizliği ortadan kaldırır.

Sertifika etiketini seçtiğiniz bir değere ayarlayabilirsiniz. Bir değer ayarlamazsanız, kullanmakta olduğunuz platforma bağlı olarak adlandırma kuralını izleyen bir varsayılan etiket kullanılır. Ayrıntılar için belirli platformlar hakkında takip eden bölümlere bakın.

Notlar:

1. Sertifika etiketini Java ya da JMS sistemlerinde kendiniz ayarlayamazsınız.
2. Kanal otomatik tanımlama (CHAD) çıkışı tarafından oluşturulan otomatik tanımlı kanallar, kanal oluşturulduğunda TLS anlaşması gerçekleştiği için sertifika etiketini ayarlayamaz. Gelen kanallar için CHAD çıkışında sertifika etiketinin ayarlanması etkili olmaz.

Bu bağlamda TLS istemcisi, bir IBM MQ istemcisi ya da başka bir kuyruk yöneticisi olabilecek el sıkışmayı başlatan bağlantı ortağını ifade eder.

TLS anlaşması sırasında TLS istemcisi her zaman sunucudan bir dijital sertifika alır ve doğrular. IBM MQ uygulamasıyla TLS sunucusu her zaman istemciden bir sertifika ister ve istemci bulunursa her zaman sunucuya bir sertifika sağlar. İstemci kişisel sertifika bulamazsa, sunucuya `no certificate` yanıtı gönderir.

TLS sunucusu, gönderildiyse, istemci sertifikasını her zaman doğrular. İstemci bir sertifika göndermezse, TLS sunucusu olarak hareket eden kanalın sonu, **SSLCAUTH** parametresi *REQUIRED* olarak ya da bir **SSLPEER** parametre değeri kümesi olarak ayarlandığında kimlik doğrulaması başarısız olur.

Uzak eşin IBM MQ sürümü sertifika etiketi yapılanışını tam olarak destekliyorsa ve kanal TLS CipherSpeckullanıyorsa, gelen kanalların (alıcı, istekçi, küme alıcı, nitelenmemiş sunucu ve sunucu bağlantısı kanalları da içinde olmak üzere) yalnızca yapılandırılan sertifikayı gönderdiğini unutmayın.

Nitelenmemiş sunucu kanalı, CONNAME alanı ayarlanmamış bir kanaldır.

Diğer tüm durumlarda, gönderilen sertifikayı kuyruk yöneticisi **CERTLABL** parametresi belirler. Özellikle, aşağıdaki öğeler, kanala özgü etiket ayarından bağımsız olarak, kuyruk yöneticisinin **CERTLABL** parametresi tarafından yapılandırılan sertifikayı alır:

- Java ve JMS istemcileri, kanal temelinde kanal temelinde sunucu adı göstergesini (SNI) destekler.
- IBM MQ 8.0sürümünden önceki IBM MQ sürümleri.
- Yönetilen .NET istemcileri

Ayrıca, bir kanal tarafından kullanılan sertifika CipherSpec kanalı için uygun olmalıdır-ek bilgi için bkz. “IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu” sayfa 46 .

IBM MQ 8.0 ve daha sonra, kanal tanımlamasında **CERTLABL** özniteliği kullanılarak belirtilen kanal başına sertifika etiketi kullanılarak aynı kuyruk yöneticisinde birden çok sertifika kullanılmasını destekler. Kuyruk yöneticisine gelen kanallar (örneğin, sunucu bağlantısı ya da alıcı), kuyruk yöneticisinden doğru sertifikayı sunmak için TLS Sunucu Adı Göstergesi 'ni (SNI) kullanarak kanal adının saptanmasına güvenir. Bir

kuyruk yöneticisinde birden çok sertifika kullanma hakkında daha fazla bilgi için bkz. “IBM MQ birden çok sertifika yeteneğini nasıl sağlar?” sayfa 29.

Bir kanal IBM MQ Internet Pass-Thru (MQIPT) aracılığıyla hedef kuyruk yöneticisine bağlanıyorsa ve MQIPT rotasında hem **SSLServer** hem de **SSLClient** ayarlanmışsa, uç noktalar arasında iki ayrı TLS oturumu vardır. IBM MQ 9.2.5' dan önceki sürümlerde, SNI verileri oturum molasında akmaz. Bu, MQIPT ile kuyruk yöneticisi arasındaki TLS bağlantısı için hedef kuyruk yöneticisinde kanal başına bir sertifikanın kullanılmasını önler. IBM MQ 9.2.5'den MQIPT , SNI' yi kanal adına ayarlayarak ya da rotaya gelen bağlantıda alınan SNI ' den geçerek hedef kuyruk yöneticisi tarafından birden çok sertifikayı kullanılmasına izin verecek şekilde yapılandırılabilir. Birden çok sertifika desteği ve MQIPTile ilgili daha fazla bilgi için bkz. IBM MQ birden çok sertifika desteği MQIPT.

Tek yönlü kimlik doğrulamasını kullanarak bir kuyruk yöneticisine bağlanma hakkında daha fazla bilgi için, yani TLS istemcisi bir sertifika göndermediğinde, Tek yönlü kimlik doğrulamasını kullanarak iki kuyruk yöneticisinin bağlanmasıbaşlıklı konuya bakın.

Çoklu platform sistemleri

Multi

Çoklu platformlarüzerinde TLS sunucusu istemciye bir sertifika gönderir.

Sırasıyla kuyruk yöneticileri ve istemciler için, boş olmayan bir değer için sırayla aşağıdaki kaynaklar aranır. Boş olmayan ilk değer sertifika etiketini belirler. Sertifika etiketi anahtar havuzunda var olmalıdır. Doğru büyük/küçük harf ve biçimde bir etiketle eşleşen bir sertifika bulunamazsa, bir hata oluşur ve TLS anlaşması başarısız olur.

Kuyruk yöneticileri

1. Kanal sertifikası etiketi özniteliği **CERTLABL**.
2. Kuyruk yöneticisi sertifika etiketi özniteliği **CERTLABL**.
3. `ibmwebspheremq` biçiminde olan ve sonuna kuyruk yöneticisinin adı eklenmiş olan bir varsayılan değer, tümü küçük harfli. Örneğin, `QM1`adlı bir kuyruk yöneticisi için varsayılan sertifika etiketi şudur: `ibmwebspheremqqm1`.

IBM MQ müşterileri

1. CLNTCONN kanal tanımlamasında **CERTLABL** sertifika etiketi özniteliği.
2. MQSCO yapısı **CertificateLabel** özniteliği.
3. Ortam değişkeni **MQCERTLABL**.
4. İstemci `.ini` dosyası (SSL bölümünde) **CertificateLabel** özniteliği
5. İstemci uygulamasının sonuna eklenmiş olarak çalıştırdığı kullanıcı kimliğiyle `ibmwebspheremq` biçimindeki bir varsayılan değer. Örneğin, `USER1`kullanıcı kimliği için varsayılan sertifika etiketi şudur: `ibmwebspheremquser1`.

z/OS sistemleri

z/OS

IBM MQ İstemcileri z/OSüzerinde desteklenmez. Ancak bir z/OS kuyruk yöneticisi, bir bağlantı başlatılırken TLS istemcisi ya da bir bağlantı isteğini kabul ederken TLS sunucusu rolünde işlem yapabilir. z/OS kuyruk yöneticilerine ilişkin sertifika etiketi gereksinimleri bu rollerin her ikisinde de geçerlidir ve Çoklu platformlarüzerindeki gereksinimlerden farklıdır.

Sırasıyla kuyruk yöneticileri ve istemciler için, boş olmayan bir değer için sırayla aşağıdaki kaynaklar aranır. Boş olmayan ilk değer sertifika etiketini belirler. Sertifika etiketi anahtar havuzunda var olmalıdır. Doğru büyük/küçük harf ve biçimde bir etiketle eşleşen bir sertifika bulunamazsa, bir hata oluşur ve TLS anlaşması başarısız olur.

1. Kanal sertifikası etiketi özniteliği, **CERTLABL**.
2. Paylaşılırsa, kuyruk paylaşım grubu sertifika etiketi özniteliği, **CERTQSGL**.

Paylaşılmazsa, kuyruk yöneticisi sertifika etiketi özniteliği, **CERTLABL**.

3. Varsayılan değer: Kuyruk yöneticisinin ya da kuyruk paylaşım grubunun adının eklendiği `ibmWebSphereMQ` biçimindedir. Bu dizginin büyük ve küçük harfe duyarlı olduğunu ve gösterildiği gibi yazılması gerektiğini unutmayın. Örneğin, `QM1`adlı bir kuyruk yöneticisi için varsayılan sertifika etiketi şudur: `ibmWebSphereMQQM1`.
4. "3" sayfa 29seçeneğinde biçimiyle bulunan bir sertifika yoksa, IBM MQ anahtar halkasında varsayılan olarak işaretlenen sertifikayı kullanmayı dener.

Anahtar havuzunun nasıl görüntüleneceğine ilişkin bilgi için bkz. "z/OS üzerinde bir kuyruk yöneticisine ilişkin anahtar havuzunun bulunması" sayfa 329.

IBM MQ Java ve IBM MQ JMS istemcileri

IBM MQ Java ve IBM MQ JMS istemcileri, TLS anlaşması sırasında kişisel sertifika seçmek için Java Secure Socket Extension (JSSE) sağlayıcısının olanaklarını kullanır ve bu nedenle sertifika etiketi gereksinimlerine tabi değildir.

Varsayılan davranış, JSSE istemcisinin anahtar havuzundaki sertifikalar aracılığıyla yinelenmesidir ve bulunan ilk kabul edilebilir kişisel sertifikayı seçmesidir. Ancak, bu davranış yalnızca bir varsayılandır ve JSSE sağlayıcısının somutlamasına bağlıdır.

Ayrıca, JSSE arabirimi, uygulama tarafından çalıştırma zamanında yapılandırma ve doğrudan erişim aracılığıyla yüksek düzeyde özelleştirilebilir. Belirli ayrıntılar için JSSE sağlayıcınız tarafından sağlanan belgelere bakın.

Sorun gidermek ya da IBM MQ Java istemci uygulaması tarafından belirli JSSE sağlayıcınızla birlikte gerçekleştirilen tokalaşmayı daha iyi anlamak için JVM ortamında `javax.net.debug=ssl` ayarını yaparak hata ayıklamayı etkinleştirebilirsiniz.

Değişkeni uygulama içinde, yapılandırma aracılığıyla ya da komut satırına `-Djavax.net.debug=ssl` girerek ayarlayabilirsiniz.

Linux *IBM MQ birden çok sertifika yeteneğini nasıl sağlar?*

Sunucu Adı Göstergesi (SNI), istemcinin hangi hizmeti gerektirdiğini belirtmesini sağlayan TLS iletişim kuralının bir uzantısıdır. IBM MQ terminolojisinde bu, bir kanala eşittir.

SNI uzantısı, kanal tanımında CERTLABL parametresi kullanılarak farklı kanallarda birden çok sertifika belirtilmesine izin vermek için IBM MQ tarafından kullanılır.

IBM MQ tarafından kullanılan SNI adresi, istenmekte olan kanal adına ve ardından `.chl.mq.ibm.com`sonekine bağlıdır.

IBM MQ kanal adları, aşağıdaki gibi geçerli SNI adları olacak şekilde eşlenir:

- A to Z büyük harfleri küçük harfe katlanır
- `0` - `9` arası rakamlar değiştirilmeden bırakılmış
- a - zarasındaki küçük harfler de içinde olmak üzere diğer tüm karakterler, iki basamaklı onaltılı ASCII karakter koduna (küçük harfli) dönüştürülür ve ardından bir kısa çizgi gelir.
  - Küçük harfler a ile z sırasıyla onaltılı `61-` ile `7a-` eşleşir
  - yüzde (%) onaltılı `25-` ile eşlenir
  - tire (-), onaltılı `2d-` ile eşlenir
  - nokta (`.`) onaltılı `2e-` ile eşlenir
  - eğik çizgi (/) onaltılı `2f-` ile eşlenir
  - altçizgi (_) onaltılı `5f-` ile eşlenir

EBCDIC altyapılarında, bu eşleme uygulanmadan önce kanal adı ASCII ' ye dönüştürülür.

Örneğin, `TO.QMGR1` kanal adı, `to2e-qmgr1.chl.mq.ibm.com`SNI adresiyle eşlenir.

Buna karşılık, `to.qmgr1` küçük harfli kanal adı, `74-6f-2e-71-6d-67-72-1.chl.mq.ibm.com`SNI adresiyle eşlenir.

Not: Oluşturulan SNI URL adresinin URL biçimlendirme belirtimlerine uyması gereken ortamlarda (örneğin, bir istemci Red Hat® OpenShift® Rotası boyunca Red Hat OpenShift içinde çalışan bir kuyruk yöneticisine bağlanırken kanal adının küçük harfle bitmemesi gerekir).

SSL 'nin **OutboundSNI** özelliği, bir TLS bağlantısı başlatılırken SNI' nin hedef IBM MQ kanal adına mı, yoksa anasistem adına mı ayarlanması gerektiğini seçmenizi sağlar. **OutboundSNI** özelliği hakkında daha fazla bilgi için bkz. qm.ini dosyasının SSL kısmı ve istemci yapılandırma dosyasının SSL kısmı.

Birden çok sertifika, SNI ' nin IBM MQ kanal adına ayarlanmasını gerektirir. Bir sertifika etiketi yapılandırılmış bir IBM MQ kanalına bağlanmak için anasistem adı, özel ya da SNI kullanılmazsa, bağlanan uygulama bir MQRC_SSL_INITIALIZATION_ERROR ile reddedilir ve uzak kuyruk yöneticisi hata günlüklerine AMQ9673 iletisi yazdırılır.

V 9.3.0 Bir kanal hedef kuyruk yöneticisine IBM MQ Internet Pass-Thru (MQIPT) aracılığıyla bağlanıyorsa, MQIPT , SNI 'yi kanal adına ayarlayacak ya da hedef kuyruk yöneticisi tarafından birden çok sertifikayı kullanmasına izin vermek için rotaya gelen bağlantıda alınan SNI' den geçecek şekilde yapılandırılmalıdır. Birden çok sertifika desteği ve MQIPTile ilgili daha fazla bilgi için bkz. IBM MQ birden çok sertifika desteği MQIPT.

Bu özelliğin nasıl kullanıldığına ilişkin ek bilgi için Red Hat OpenShift kümesinde konuşlandırılan bir kuyruk yöneticisine bağlanmabaşlıklı konuya bakın.

Kuyruk yöneticisinin anahtar havuzu yenileniyor
Bir anahtar havuzunun içeriğini değiştirdiğinizde, bir REFRESH SECURITY TYPE (SSL) komutu verilinceye ya da kuyruk yöneticisi yeniden başlatılıncaya kadar, var olan kuyruk yöneticisi işlemleri yeni içeriği almaz.

REFRESH SECURITY TYPE (SSL) komutuna ilişkin ek bilgi için REFRESH SECURITYbaşlıklı konuya bakın.

Kuyruk yöneticisi anahtar deposunun içeriğini değiştirdikten sonra yeni bir kanal işlemi (`amqrmppa` ya da **runmqchl**kullanılarak) yaratırsa, yeni işlem yeni sertifikaları kullanmaya hemen başlar, ancak var olan işlemler anahtar deposunun önbelleğe alınmış kopyasını kullanmaya devam eder. Daha fazla ayrıntı için bkz. “Sertifikalarda ya da sertifika deposunda yapılan değişiklikler AIX, Linux, and Windows üzerinde yürürlüğe girdiğinde” sayfa 303 .

Çalışan birden çok kanal, bir REFRESH SECURITY TYPE (SSL) komutu verilinceye kadar anahtar havuzunun farklı sürümlerini kullanıyor olabilir.

Bir anahtar havuzunu PCF komutlarını ya da IBM MQ Explorerkomutunu kullanarak da yenileyebilirsiniz. Daha fazla bilgi için, bu ürün belgelerinin IBM MQ Explorer bölümünde MQCMD_REFRESH_SECURITY komutu ve *TLS Güvenliğinin Yenilenmesi* konusuna bakın.

İlgili kavramlar
“İstemcinin SSL/TLS anahtar havuzu içeriğini ve SSl/TLS ayarlarını görünümünün yenilenmesi” sayfa 30
İstemci uygulamasını anahtar havuzunun yenilenen içeriğiyle güncellemek için istemci uygulamasını durdurmalı ve yeniden başlatmalısınız.

İstemcinin SSL/TLS anahtar havuzu içeriğini ve SSl/TLS ayarlarını görünümünün yenilenmesi
İstemci uygulamasını anahtar havuzunun yenilenen içeriğiyle güncellemek için istemci uygulamasını durdurmalı ve yeniden başlatmalısınız.

IBM MQ istemcisinde güvenliği yenileyemezsiniz; istemciler için REFRESH SECURITY TYPE (SSL) komutunun eşdeğeri yoktur (bkz. REFRESH SECURITY) ek bilgi için.

İstemci uygulamasını anahtar havuzunun yenilenmiş içeriğiyle güncellemek için, güvenlik sertifikasını her değiştirdiğinizde uygulamayı durdurup yeniden başlatmanız gerekir.

Kanal yeniden başlatılırsa, yapılanışlar yenilenir ve uygulamanızda yeniden bağlanma mantığı varsa, STOP CHL STATUS (INACTIVE) komutunu vererek istemcide güvenliği yenileyebilirsiniz.

İlgili kavramlar
“Kuyruk yöneticisinin anahtar havuzu yenileniyor” sayfa 30

Bir anahtar havuzunun içeriğini değiştirdiğinizde, bir REFRESH SECURITY TYPE (SSL) komutu verilinceye ya da kuyruk yöneticisi yeniden başlatılıncaya kadar, var olan kuyruk yöneticisi işlemleri yeni içeriği almaz.

MQCSP parola koruması

MQCSP yapısında belirtilen kimlik doğrulama kimlik bilgileri, IBM MQ MQCSP parola koruma özelliği kullanılarak korunabilir ya da TLS şifrelemesi kullanılarak şifrelenebilir.

IBM MQ client uygulamaları, bir kuyruk yöneticisine bağlandıklarında kullanıcı kimliği ve parola sağlayabilir. V 9.3.4 IBM MQ 9.3.4' dan uygulamalar, alternatif bir kimlik doğrulama yöntemi olarak bir kimlik doğrulama belirteci de sağlayabilir. Bu kimlik bilgileri bir MQCSP yapısında kuyruk yöneticisine gönderilir.

Kanal TLS şifrelemesi kullanıyorsa, MQCSP ' deki kimlik bilgileri TLS şifreleme belirtimine göre şifrelenir. IBM MQ 8.0' den kanal TLS şifrelemesini kullanmıyorsa IBM MQ , ağ üzerinden gönderilmeden önce bu kimlik bilgilerini koruyabilir; bu, kimlik bilgilerinin ağ üzerinden düz metin olarak gönderilmesini önler. Bu kimlik bilgilerini koruyan IBM MQ özelliğine MQCSP parola koruması adı verilir.

MQCSP parola koruması kullanılırsa, MQCSP yapısında aşağıdaki veriler korunur:

- `MQCSP.AuthenticationType` alanı `MQCSP_AUTH_USER_ID_AND_PW`olarak ayarlanırsa, parola.
- V 9.3.4 `MQCSP.AuthenticationType` alanı `MQCSP_AUTH_ID_TOKEN`olarak ayarlanırsa, kimlik doğrulama simgesi.

Önemli: MQCSP parola koruması, test ve geliştirme amaçlarıyla yararlıdır; MQCSP parola korumasının kullanılması TLS şifrelemesi ayarlanmasından daha basittir, ancak güvenli değildir. Üretim amacıyla, TLS şifrelemesi daha güvenli olduğundan, özellikle istemci ile kuyruk yöneticisi arasındaki ağ güvenilmediğinde IBM MQ parola koruması yerine TLS şifrelemesini kullanın.

Hangi şifrelemenin kullanıldığına ve ne kadar koruma sunacağına ilişkin endişeleriniz varsa, tam TLS şifrelemesini kullanmanız gerekir. TLS ile algoritmalar genel olarak bilinir ve **SSLCIPH** kanal özniteliğini kullanarak kuruluşunuz için uygun olanı seçebilirsiniz.

MQCSP yapısıyla ilgili daha fazla bilgi için bakınız: MQCSP structure.

Aşağıdaki koşulların tümü karşılandıysa, MQCSP yapısındaki kimlik bilgileri IBM MQ parola koruması kullanılarak korunur:

- Bağlantının her iki ucu da IBM MQ 8.0ya da daha sonraki bir sürümü kullanıyor.
- Kanal TLS şifrelemesini kullanmıyor. Kanal boş bir **SSLCIPH** özniteliğine sahipse ya da **SSLCIPH** özniteliği şifreleme sağlamayan bir şifreleme belirtimine ayarlıysa, kanal TLS şifrelemesini kullanmıyor. Boş değerli şifrelemeler (örneğin, NULL_SHA) şifreleme sağlamaz.
- `MQCSP.AuthenticationType` alanı `MQCSP_AUTH_USER_ID_AND_PWD` ya da `MQCSP_AUTH_ID_TOKEN`olarak ayarlanır. `MQCSP.AuthenticationType` alanıyla ilgili daha fazla bilgi için bkz. **`AuthenticationType`**.
- İstemci IBM MQ Explorer ise ve kullanıcı kimliği uyumluluğu kipi etkinleştirilmediyse. Bu kip, IBM MQ Explorer tarafından kullanıcı kimliği ve parola göndermek için kullanılan varsayılan kip değildir. Bu koşul yalnızca IBM MQ Exploreriçin geçerlidir.

Bu koşullardan herhangi biri karşılanmazsa, kimlik bilgileri MQCSP parola korumasıyla korunmaz. **`PasswordProtection`** özniteliğinin değeri kimlik bilgilerinin düz metin olarak gönderilmesine izin vermezse ve kanal TLS şifrelemesi kullanmıyorsa, bağlantı başarısız olur ve bir MQRC_PASSWORD_PROTECTION_ERROR (2594) neden kodu döndürülür.

`PasswordProtection` yapılandırma ayarı

İstemci ve kuyruk yöneticisi yapılanış dosyalarının **`Channels`** kısmına ilişkin **`PasswordProtection`** özniteliği, kimlik bilgilerinin düz metin olarak gönderilmesini engelleyebilir.

Not: Bu öznitelik yalnızca TLS şifrelemesi kullanmayan bağlantılar için geçerlidir. Bağlantı TLS şifrelemesi kullanıyorsa, kimlik bilgileri MQCSP parola korumasıyla korunmak yerine TLS kullanılarak şifrelenir.

Öznitelik aşağıdaki değerlerden birine ayarlanabilir. Varsayılan değer `compatible`değeridir.

Uyumlu
Kuyruk yöneticisi ya da istemci IBM MQ 8.0' den önceki bir IBM MQ sürümünü çalıştırıyorsa, kimlik bilgileri düz metin olarak gönderilir. Yani, kimlik bilgileri, MQCSP parola korumasını desteklemeyen IBM MQ sürümleriyle uyumluluk için bir ağ üzerinden düz metin olarak gönderilebilir.

Hem kuyruk yöneticisi hem de istemci IBM MQ 8.0 ya da daha sonraki bir IBM MQ sürümünü çalıştırıyorsa, kimlik bilgileri MQCSP parola korumasıyla korunur.

Hem kuyruk yöneticisi hem de istemci IBM MQ 8.0 ya da sonraki bir IBM MQ sürümünü çalıştırıyorsa ve `MQCSP.AuthenticationType` alanı `MQCSP_AUTH_USER_ID_AND_PW` ya da `MQCSP_AUTH_ID_TOKEN`olarak ayarlanmamışsa, kimlik bilgileri gönderilmeden önce bağlantı başarısız olur.

always
Kimlik bilgileri korumasız bir ağ üzerinden gönderilmemelidir.

Hem kuyruk yöneticisi hem de istemci IBM MQ 8.0 ya da daha sonraki bir IBM MQ sürümünü çalıştırıyorsa, kimlik bilgileri MQCSP parola korumasıyla korunur.

Kimlik bilgileri aşağıdaki durumlarda gönderilmeden önce bağlantı başarısız olur:

- `MQCSP.AuthenticationType` alanı `MQCSP_AUTH_USER_ID_AND_PW` ya da `MQCSP_AUTH_ID_TOKEN`olarak ayarlanmadı.
- Kuyruk yöneticisi ya da istemci, IBM MQ 8.0sürümünden önceki bir IBM MQ sürümünü çalıştırıyor.

isteğe bağlı
Hem kuyruk yöneticisi hem de istemci IBM MQ 8.0 ya da daha sonraki bir IBM MQ sürümünü çalıştırıyorsa ve `MQCSP.AuthenticationType` alanı `MQCSP_AUTH_USER_ID_AND_PW` ya da `MQCSP_AUTH_ID_TOKEN`olarak ayarlıysa, kimlik bilgileri MQCSP parola korumasıyla korunur. Tersi durumda, kimlik bilgileri düz metin olarak gönderilir.

uyarı
Herhangi bir istemcinin düz metin kimlik bilgileri göndermesine izin verilir. Düz metin kimlik bilgileri alınırsa, kuyruk yöneticisi hata günlüklerine AMQ9297W uyarı iletisi yazılır.

Bu seçenek yalnızca kuyruk yöneticisi yapılanış dosyasında belirtilebilir.

Java ve JMS istemcileri için, **`PasswordProtection`** özniteliğinin davranışı istemcinin uyumluluk kipini mi, yoksa MQCSP kipini mi kullandığına bağlı olarak değişir:

- Java ve JMS istemcileri uyumluluk kipinde çalışıyorsa, istemci bağlandığında kullanıcı kimliğini ve parolayı göndermek için MQCSP yapısı kullanılmaz. Bu nedenle, **`PasswordProtection`** özniteliğinin davranışı, IBM MQ 8.0sürümünden önceki bir IBM MQ sürümünü çalıştıran istemciler için açıklanan davranışla aynıdır.
- Java ve JMS istemcileri MQCSP kipinde çalışıyorsa, **`PasswordProtection`** özniteliğinin davranışı açıklanan davranıştır.

Java ve JMS istemcileriyle bağlantı kimlik doğrulaması hakkında daha fazla bilgi için bkz. “Java istemcisiyle bağlantı kimlik doğrulaması” sayfa 81.

MQCSP parola koruması ve MQIPT

V 9.3.1

Bir istemci IBM MQ Internet Pass-Thru (MQIPT) aracılığıyla bir kuyruk yöneticisine bağlanıyorsa, MQIPT rotası TLS şifrelemesi ekleyecek ya da kaldıracak şekilde yapılandırılabilir. Diğer bir deyişle, MQIPT rotası `SSLServer=true` ve `SSLClient=false`ya da `SSLServer=true` ve `SSLClient=false`ile yapılandırılabilir. Bu durumda, istemci ve kuyruk yöneticisi, kanalın bir ucu TLS şifrelemesi kullanırken diğeri kullanmadığı için bir parola koruma algoritmasını kabul edemeyebilir. Bu, MQRC_PASSWORD_PROTECTION_ERROR (2594) neden koduyla bağlantının başarısız olmasına neden olur.

IBM MQ 9.3.1' den MQIPT , TLS şifrelemesi ekleyen ya da kaldırabilen MQIPT rotaları için istemci ile kuyruk yöneticisi arasındaki uyumluluğu korumak amacıyla MQCSP yapılarına kimlik bilgileri için

koruma ekleyebilir ya da var olan kimlik bilgilerini kaldırabilir. MQIPT içindeki MQCSP parola koruması, **PasswordProtection** rota özelliği kullanılarak yapılandırılır.

PasswordProtection özelliğinin varsayılan değeri `gereklidir`. Bu değer, MQIPT ' in MQCSP parola korumasını ekleyebileceği, ancak kaldıramayabileceği anlamına gelir. TLS şifrelemesi ekleyen bir MQIPT rotasına yönelik bağlantılar, **PasswordProtection**değeriyle MQRC_PASSWORD_PROTECTION_ERROR (2594) neden koduyla başarısız olabilir. Bu sorunu çözmek için, MQIPT rota yapılandırmasında **PasswordProtection** özelliğinin değerini `uyumlu` olarak ayarlayın.

MQIPTiçindeki **PasswordProtection** özelliği hakkında daha fazla bilgi için bkz. PasswordProtection.

Dijital Certificate Manager (DCM)

IBM iüzerinde dijital sertifikaları ve özel anahtarları yönetmek için DCM ' yi kullanın.

Digital Certificate Manager (DCM), dijital sertifikaları yönetmenizi ve bunları IBM i sunucusundaki güvenli uygulamalarda kullanmanızı sağlar. Digital Certificate Managerile, Sertifika Yetkilileri 'nden (CA) ya da diğer üçüncü kişilerden sayısal sertifikaları isteyebilir ve işleyebilirsiniz. Kullanıcılarınız için dijital sertifikalar yaratmak ve yönetmek üzere yerel bir Sertifika Yetkilisi olarak da görev yapabilirsiniz.

DCM, daha güçlü bir sertifika ve uygulama doğrulama süreci sağlamak için Sertifika İptal Listelerinin (CRL) kullanılmasını da destekler. IBM MQ 'un belirli bir sertifikanın iptal edilmediğini doğrulayabilmesi için belirli bir Sertifika Yetkilisi CRL 'sinin bir LDAP sunucusunda bulunduğu konumu tanımlamak üzere DCM' yi kullanabilirsiniz.

DCM, çeşitli biçimlerdeki sertifikaları destekler ve otomatik olarak algılayabilir. DCM, şifrelenmiş verileri içeren bir PKCS #12 kodlanmış sertifika ya da PKCS #7 sertifikası algıladığında, otomatik olarak kullanıcıdan sertifikayı şifrelemek için kullanılan parolayı girmesini ister. DCM, şifrelenmiş veri içermeyen PKCS #7 sertifikaları için bilgi isteminde bulunmaz.

DCM, uygulamalarınız ve kullanıcılarınız için dijital sertifikaları yönetmek üzere kullanabileceğiniz tarayıcı tabanlı bir kullanıcı arabirimi sağlar. Kullanıcı arabirimi iki ana çerçeveye ayrılır: bir gezinme çerçevesi ve bir görev çerçevesi.

Sertifikaları ya da bunları kullanan uygulamaları yönetmek üzere görevleri seçmek için gezinme çerçevesini kullanabilirsiniz. Bazı görevler doğrudan ana gezinme çerçevesinde gösterilir, ancak gezinme çerçevesindeki görevlerin çoğu kategoriler halinde düzenlenir. Örneğin, Sertifikaları Yönet, sertifikayı görüntüle, sertifikayı yenile ve sertifikayı içe aktar gibi çeşitli kılavuzlu görevleri içeren bir görev kategorisidir. Gezinme çerçevesindeki bir öğe birden çok görev içeren bir kategoriyse, sol tarafta bir ok görüntülenir. Ok, kategori bağlantısını seçtiğinizde, hangi görevin gerçekleştirileceğini seçmenizi sağlayan genişletilmiş bir görev listesi görüntülendiğini gösterir.

DCM ile ilgili önemli bilgiler için aşağıdaki IBM Redbooks yayınlarına bakın:

- *IBM i Kablolu Ağ Güvenliği: OS/400 V5R1 DCM ve Şifreleme Geliştirmeleri*, SG24-6168. Özellikle, IBM i sisteminizi yerel bir CA olarak ayarlamaya ilişkin temel bilgiler için eklere bakın.
- *AS/400 Internet Security: Bir Dijital Sertifika Altyapısı Geliştirilmesi*, SG24-5659. Özellikle, bkz. Bölüm 5. *Digital Certificate Manager for AS/400* , AS/400 DCM ' yi açıklar.

Federal Bilgi İşleme Standartları (FIPS)

Bu konuda, ABD Ulusal Standartlar ve Teknoloji Enstitüsü 'nün Federal Bilgi İşleme Standartları (FIPS) Şifreleme Modülü Doğrulama Programı ve TLS kanallarında kullanılabilen şifreleme işlevleri tanıtılmaktadır.

Not: AIX, Linux, and Windowsişletim sistemlerinde IBM MQ , IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, IBM Crypto for C (ICC) sertifikasını görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu İşlem listesindeki NIST CMVP modüllerindearanarak görüntülenebilir.

Bu bilgiler aşağıdaki altyapılar için geçerlidir:

- ALW AIX, Linux, and Windows

- z/OS z/OS

ALW AIX, Linux, and Windowsüzerinde bir IBM MQ TLS bağlantısının FIPS 140-2 uyumluluğuyla ilgili daha fazla bilgi için bkz. “AIX, Linux, and Windows için Federal Bilgi İşleme Standartları (FIPS)” sayfa 34.

z/OS z/OSüzerinde bir IBM MQ TLS bağlantısının FIPS 140-2 uyumluluğuyla ilgili daha fazla bilgi için bkz. “z/OS için Federal Bilgi İşleme Standartları (FIPS)” sayfa 37.

Şifreleme donanımı varsa, IBM MQ tarafından kullanılan şifreleme modülleri donanım üreticisi tarafından sağlanacak şekilde yapılandırılabilir. Bu yapıldıysa, yapılandırma yalnızca bu şifreleme modüllerinin FIPS onaylı olması durumunda FIPS uyumludur.

Zaman içinde Federal Bilgi İşleme Standartları, şifreleme algoritmalarına ve protokollerine karşı yeni saldırıları yansıtacak şekilde güncellenir. Örneğin, bazı CipherSpecs FIPS onaylı olmayabilir. Bu tür değişiklikler gerçekleştiğinde, IBM MQ da en son standardı uygulayacak şekilde güncellenir. Sonuç olarak, bakım uygulandıktan sonra davranıştaki değişiklikleri görebilirsiniz.

İlgili kavramlar

“MQI istemcisinde çalıştırma zamanında yalnızca FIPS onaylı CipherSpecs kullanılmasının belirtilmesi” sayfa 265
FIPS uyumlu yazılımı kullanarak anahtar havuzlarınızı oluşturun ve daha sonra, kanalın FIPS onaylı CipherSpecsözelliğini kullanması gerektiğini belirtin.

“Dijital sertifikaları yönetmek için runmqckm, runmqakm ve strmqikm kullanılıyor” sayfa 289
AIX, Linux, and Windows sistemlerinde, anahtarları ve dijital sertifikaları **strmqikm** (iKeyman) ile yönetin GUI ya da **runmqckm** (iKeycmd) ya da **runmqakm** (GSKCapiCmd) komutunu kullanarak komut satırından.

İlgili görevler

IBM MQ classes for Java içinde TLS ' yi etkinleştirme

IBM MQ classes for JMS ile TLS (Transport Layer Security; İletim Katmanı Güvenliği) kullanılması

İlgili başvurular

JMS nesnelerinin TLS özellikleri

“Federal Bilgi İşleme Standartları” sayfa 23
ABD hükümeti, veri şifreleme de dahil olmak üzere, BT sistemleri ve güvenlik konusunda teknik öneriler üretiyor. Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), BT sistemleri ve güvenliğiyle ilgili önemli bir kurumdur. NIST, Federal Bilgi İşleme Standartları (FIPS) dahil olmak üzere öneriler ve standartlar üretir.

ALW *AIX, Linux, and Windows için Federal Bilgi İşleme Standartları (FIPS)*

AIX, Linux, and Windows sistemlerinde bir SSL/TLS kanalında şifreleme gerektiğinde, IBM MQ IBM Crypto for C (ICC)adlı bir şifreleme paketi kullanır. AIX, Linux, and Windows platformlarında ICC yazılımı, ABD Ulusal Standartlar ve Teknoloji Enstitüsü 'nün Federal Bilgi İşleme Standartları (FIPS) Şifreleme Modülü Doğrulama Programı 'nı 140-2 düzeyinde geçti.

Not: AIX, Linux, and Windowsişletim sistemlerinde IBM MQ , IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, IBM Crypto for C (ICC) sertifikasını görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu İşlem listesindeki NIST CMVP modüllerindearanarak görüntülenebilir.

AIX, Linux, and Windows sistemlerinde IBM MQ TLS bağlantısının FIPS 140-2 uyumluluğu aşağıdaki gibidir:

- Aşağıdaki koşullar karşılandıysa, tüm IBM MQ ileti kanalları (CLNTCONN kanal tipleri dışında) için bağlantı FIPS uyumludur:
  - Kurulu IBM Global Security Kit (GSKit) ICC sürümü, kurulu işletim sistemi sürümü ve donanım mimarisinde FIPS 140-2 uyumlu olarak onaylanmıştır.
  - Kuyruk yöneticisinin SSLFIPS özniteliği YES olarak ayarlandı.

  - Tüm anahtar havuzları, `-fips` seçeneğiyle **runmqakm** gibi yalnızca FIPS uyumlu yazılımlar kullanılarak oluşturulmuştur ve işlenmiştir.
  - Tüm anahtar havuzlarına erişim, kuyruk yöneticisinin **KEYRPWD** özniteliği değil, bir zula dosyası kullanılarak sağlanır.
- Tüm IBM MQ MQI client uygulamaları için bağlantı GSKit kullanır ve aşağıdaki koşullar karşılandığında FIPS uyumludur:
  - Kurulu GSKit ICC sürümü, kurulu işletim sistemi sürümü ve donanım mimarisinde FIPS 140-2 uyumlu olarak onaylanmıştır.
  - MQI istemcisine ilişkin ilgili konuda açıklandığı gibi, yalnızca FIPS sertifikalı şifrelemenin kullanılacağını belirttiniz.
  - Tüm anahtar havuzları, `-fips` seçeneğiyle **runmqakm** gibi yalnızca FIPS uyumlu yazılımlar kullanılarak oluşturulmuştur ve işlenmiştir.
  - Tüm anahtar havuzlarına erişim, anahtar havuzu parola düzeneği değil, bir parola saklama dosyası kullanılarak sağlanır.
- İstemci kipini kullanan IBM MQ classes for Java uygulamaları için bağlantı, JRE ' nin TLS uygulamalarını kullanır ve aşağıdaki koşullar karşılandığında FIPS uyumludur:
  - Uygulamayı çalıştırmak için kullanılan Java Runtime Environment, kurulu işletim sistemi sürümü ve donanım mimarisinde FIPS uyumludur.
  - Java istemcisi için ilgili konuda açıklandığı gibi, yalnızca FIPS sertifikalı şifrelemenin kullanılacağını belirttiniz.
  - Tüm anahtar havuzları, `-fips` seçeneğiyle **runmqakm** gibi yalnızca FIPS uyumlu yazılımlar kullanılarak oluşturulmuştur ve işlenmiştir.
- İstemci kipini kullanan IBM MQ classes for JMS uygulamaları için bağlantı, JRE ' nin TLS uygulamalarını kullanır ve aşağıdaki koşullar karşılandığında FIPS uyumludur:
  - Uygulamayı çalıştırmak için kullanılan Java Runtime Environment, kurulu işletim sistemi sürümü ve donanım mimarisinde FIPS uyumludur.
  - JMS istemcisi için ilgili konuda açıklandığı gibi, yalnızca FIPS sertifikalı şifrelemenin kullanılacağını belirttiniz.
  - Tüm anahtar havuzları, `-fips` seçeneğiyle **runmqakm** gibi yalnızca FIPS uyumlu yazılımlar kullanılarak oluşturulmuştur ve işlenmiştir.
- Yönetilmeyen .NET istemci uygulamalarında, aşağıdaki koşullar karşılandığında bağlantı GSKit komutunu kullanır ve FIPS uyumludur:
  - Kurulu GSKit ICC sürümü, kurulu işletim sistemi sürümü ve donanım mimarisinde FIPS 140-2 uyumlu olarak onaylanmıştır.
  - .NET istemcisi için ilgili konuda açıklandığı gibi, yalnızca FIPS sertifikalı şifrelemenin kullanılacağını belirttiniz.
  - Tüm anahtar havuzları, `-fips` seçeneğiyle **runmqakm** gibi yalnızca FIPS uyumlu yazılımlar kullanılarak oluşturulmuştur ve işlenmiştir.
  - Tüm anahtar havuzlarına erişim, anahtar havuzu parola düzeneği değil, bir parola saklama dosyası kullanılarak sağlanır.
- Yönetilmeyen XMS .NET istemci uygulamaları için bağlantı GSKit kullanır ve aşağıdaki koşullar karşılandığında FIPS uyumludur:
  - Kurulu GSKit ICC sürümü, kurulu işletim sistemi sürümü ve donanım mimarisinde FIPS 140-2 uyumlu olarak onaylanmıştır.
  - XMS .NET belgelerinde açıklandığı gibi yalnızca FIPS sertifikalı şifrelemenin kullanılacağını belirttiniz.
  - Tüm anahtar havuzları, `-fips` seçeneğiyle **runmqakm** gibi yalnızca FIPS uyumlu yazılımlar kullanılarak oluşturulmuştur ve işlenmiştir.
  - Tüm anahtar havuzlarına erişim, anahtar havuzu parola düzeneği değil, bir parola saklama dosyası kullanılarak sağlanır.

Desteklenen tüm platformlar, her düzeltme paketinde ya da yenileme paketinde yer alan benioku dosyasında belirtilenler dışında, FIPS 140-2 sertifikalıdır.

GSKitkullanan TLS bağlantıları için, FIPS 140-2 sertifikalı bileşen *ICC*olarak adlandırılır. Belirli bir platformda GSKit FIPS uyumluluğunu belirleyen bu bileşenin sürümüdür. Kurulu olan ICC sürümünü belirlemek için **dspmqver -p 64 -v** komutunu çalıştırın.

Aşağıda, ICCile ilgili **dspmqver -p 64 -v** çıkışının bir örneği verilmiştir:

```
icc
============
@ (#)CompanyName: IBM Corporation
@(#)LegalTrademarks: IBM
@ (#)FileDescription: IBM Crypto
@(#)FileVersion:   8.0.0.0
@ (#)LegalCopyright: Lisanslı Malzeme- IBM
@ (#) ICC
@ (#) (C) Copyright IBM Corp. 2002, 2024.
@ (#) Her Hakkı Saklıdır. ABD Hükümeti Kullanıcıları
@ (#) Sınırlı Haklar-Kullanım, çoğaltma ya da açıklama
@ (#), IBM Corp. ile yapılan GSA ADP Schedule Contract adlı sözleşmeyle sınırlanmıştır.
@ (#)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:
```

GSKit ICC 8 için NIST sertifikasyon bildirimi (GSKit 8içinde bulunur) şu adreste bulunabilir: Cryptographic Module Validation Program.

Şifreleme donanımı varsa, IBM MQ tarafından kullanılan şifreleme modülleri donanım üreticisi tarafından sağlanacak şekilde yapılandırılabilir. Bu yapıldıysa, yapılandırma yalnızca bu şifreleme modüllerinin FIPS onaylı olması durumunda FIPS uyumludur.

FIPS 140-2 ile uyumlu olarak çalışırken uygulanan üçlü DES kısıtlamaları

IBM MQ , FIPS 140-2 ile uyumlu çalışacak şekilde yapılandırıldığında, Triple DES (3DES) CipherSpecsile ilgili olarak ek kısıtlamalar uygulanır. Bu kısıtlamalar, ABD NIST SP800-67 önerisiyle uyumluluğu sağlar.

1. Triple DES anahtarının tüm parçaları benzersiz olmalıdır.
2. NIST SP800-67içindeki tanımlara göre Üçlü DES anahtarının hiçbir parçası Zayıf, Yarı Zayıf ya da Zayıf olamaz.
3. Gizli bir anahtar sıfırlaması gerçekleşmeden önce bağlantı üzerinden en fazla 32 GB veri iletilebilir. Varsayılan olarak IBM MQ , gizli oturum anahtarını sıfırlamaz, bu nedenle bu sıfırlama yapılandırılmalıdır. Üçlü DES CipherSpec ve FIPS 140-2 uyumluluğu kullanılırken gizli anahtar sıfırlama etkinleştirilmemesi, bayt sayısı üst sınırı aşıldıktan sonra AMQ9288 hatasıyla bağlantının kapanmasına neden olur. Gizli anahtar sıfırlamasını yapılandırma hakkında bilgi için bkz. “SSL ve TLS gizli anahtarlarını sıfırlama” sayfa 478.

IBM MQ , 1 ve 2 numaralı kurallara zaten uyan Üçlü DES oturum anahtarları oluşturur. Ancak, üçüncü kısıtlamayı karşılamak için FIPS 140-2 yapılandırmasında Üçlü DES CipherSpecs kullanırken gizli anahtar sıfırlamasını etkinleştirmeniz gerekir. Alternatif olarak, Üçlü DES kullanmaktan kaçınabilirsiniz.

İlgili kavramlar

“MQI istemcisinde çalıştırma zamanında yalnızca FIPS onaylı CipherSpecs kullanılmasının belirtilmesi” sayfa 265
FIPS uyumlu yazılımı kullanarak anahtar havuzlarınızı oluşturun ve daha sonra, kanalın FIPS onaylı CipherSpecsözelliğini kullanması gerektiğini belirtin.

“Dijital sertifikaları yönetmek için runmqckm, runmqakm ve strmqikm kullanılıyor” sayfa 289
AIX, Linux, and Windows sistemlerinde, anahtarları ve dijital sertifikaları **strmqikm** (iKeyman) ile yönetin GUI ya da **runmqckm** (iKeycmd) ya da **runmqakm** (GSKCapiCmd) komutunu kullanarak komut satırından.

İlgili görevler

IBM MQ classes for Java içinde TLS ' yi etkinleştirme

IBM MQ classes for JMS ile TLS (Transport Layer Security; İletim Katmanı Güvenliği) kullanılması

İlgili başvurular

JMS nesnelerinin TLS özellikleri

“Federal Bilgi İşleme Standartları” sayfa 23

ABD hükümeti, veri şifreleme de dahil olmak üzere, BT sistemleri ve güvenlik konusunda teknik öneriler üretiyor. Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), BT sistemleri ve güvenliğiyle ilgili önemli bir kurumdur. NIST, Federal Bilgi İşleme Standartları (FIPS) dahil olmak üzere öneriler ve standartlar üretir.

z/OS *z/OS için Federal Bilgi İşleme Standartları (FIPS)*

z/OS üzerindeki bir SSL/TLS kanalında şifreleme gerektiğinde, IBM MQ Sistem SSL adı verilen bir hizmeti kullanır. Sistem SSL ' nin amacı, ABD Ulusal Standartlar ve Teknoloji Enstitüsü 'nün Federal Bilgi İşleme Standartları (FIPS) Şifreleme Modülü Doğrulama Programını 140-2 düzeyinde uygulamak üzere tasarlanmış bir modda güvenli bir şekilde yürütme yeteneği sağlamaktır.

IBM MQ TLS bağlantılarıyla FIPS 140-2 uyumlu bağlantılar uygularken göz önünde bulundurulması gereken birkaç nokta vardır:

- FIPS-uyumluluğu için IBM MQ ileti kanallarını etkinleştirmek üzere aşağıdaki koşulların karşılandığından emin olun:
  - Sistem SSL Güvenlik Düzeyi 3 FMID kurulu ve yapılandırılmış olmalıdır (bkz. Kuruluşunun planlanması IBM MQ).
  - Sistem SSL modüllerinin geçerliliği denetlenir.
  - Kuyruk yöneticisinin SSLFIPS özniteliği **YES**olarak ayarlandı.

Sistem SSL, FIPS kipinde çalıştırılırken, kullanılabilir olduğunda CP Assist for Cryptographic Function (CPACF) olanağından yararlanılır. Yazılımda gerçekleştirilmesi gereken RSA imza oluşturma dışında, FIPS kipinde çalışırken ICSF destekli donanım tarafından gerçekleştirilen şifreleme işlevleri kullanılmaya devam eder.

Çizelge 2. FIPS kipi ile FIPS kipi olmayan algoritma desteği arasındaki farklar.

	FIPS Olmayan		**FIPS**	
Algoritma	**Anahtar boyutları**	**<**	**Anahtar boyutları**	**<**
RC2	40 ve 128			
RC4	40 ve 128			
DES	56	x		
TDES	168	x	168	x
AES.	128 ve 256	x	128 ve 256	x
MD5	48			
SHA-1	160	x	160	x
SHA-2	224, 256, 384 ve 512	x	224, 256, 384 ve 512	x
RSA	512-4096	x	1024-4096	x
DSA.	512-1024		1024	
DH	512-2048		2048	

FIPS kipinde, Sistem SSL yalnızca Tablo 1 'de gösterilen algoritmaları ve anahtar boyutlarını kullanan sertifikaları kullanabilir. X.509 sertifika doğrulaması sırasında, FIPS kipiyle uyumsuz bir algoritma saptanırsa, sertifika kullanılamaz ve geçersiz olarak kabul edilir.

WebSphere Application Server içinde istemci kipini kullanan IBM MQ sınıf uygulamaları için bkz. Federal Information Processing Standard support.

Sistem SSL modülü yapılandırmasına ilişkin bilgi için Sistem SSL Modülü Doğrulama Ayarı başlıklı konuya bakın.

İlgili başvurular

"Federal Bilgi İşleme Standartları" sayfa 23
ABD hükümeti, veri şifreleme de dahil olmak üzere, BT sistemleri ve güvenlik konusunda teknik öneriler üretiyor. Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), BT sistemleri ve güvenliğiyle ilgili önemli bir kurumdur. NIST, Federal Bilgi İşleme Standartları (FIPS) dahil olmak üzere öneriler ve standartlar üretir.

Multi

mqcertck ile kuyruk yöneticinizin TLS yapılandırmasını doğrulama

MQCERTCK komutu, kuyruk yöneticinizin TLS yapılandırmasında sık kullanılan hataları aramak için kullanılan bir araçtır ve sorunların çözümüne ilişkin bazı öneriler sağlar.

Giriş

mqcertck komutu aşağıdakileri denetler:

- Kuyruk yöneticisi **SSLKEYR** özniteliğinde gönderme yapılan kuyruk yöneticisinin anahtar havuzunun varlığı ve izinleri.
- Kuyruk yöneticisi **CERTLABL** özniteliğinde başvurulan kuyruk yöneticisi sertifikasına ilişkin sertifikanın varlığı ve geçerliliği.
- TLS etkin kanalın **CERTLABL** özniteliklerinde başvurulan sertifikaların varlığı ve geçerliliği.
- Sertifikaların denetlenmesi de içinde olmak üzere, istemci uygulamalarının anahtar havuzu ve sertifikalarının kuyruk yöneticisiyle yetkisi vardır.

Not: mqcertck komutu z/OS ya da IBM üzerinde kullanılamaz.

Kullanım

mqcertck komutunu kullanmak için mqcertckkomutunu, gerekli parametreleriyle birlikte ve bir komut satırından isteğe bağlı parametrelerle birlikte çalıştırın.

Komutun ve komutun aldığı değiştirgelerin açıklaması için mqcertck kısmına bakın.

Örnek

Kuyruk yöneticinizin SVRCONN kanalına bağlanan istemcilerden TLS bağlantılarına izin vermek için QM1 kuyruk yöneticinizi ayarlamayı yeni bitirdiniz.

Birden çok sertifika özelliğini kullanıyorsunuz ve bu nedenle hem kuyruk yöneticinizin hem de kanalınızın **CERTLABL** özniteliklerinde belirtilmiş bir sertifika etiketi var. Kanalın **CERTLABL** özniteliğinde bir hata yaptınız, bu nedenle bir istemci bağlanmaya çalıştığında kuyruk yöneticisi 2393 dönüş kodunu MQRC_SSL_INITIALIZATION_ERROR dönüş kodunu döndürür.

Kuyruk yöneticisini etkinleştirmeden önce, kuyruk yöneticisinin TLS yapılandırmasını doğrulamak için **mqcertck** komutunu kullanın.

mqcertck QM1 komutunu çalıştırıp aşağıdaki çıkışı alırsınız:

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

Bu çıkış, MQCERTCK.CHANNEL. Burada, yaptığınız hatayı görürsünüz ve sorunu çözümlediğinizi doğrulamak için `mqcertck` komutunu yeniden çalıştırmadan önce hatayı düzeltebilirsiniz.

İstemci bağlantılarının doğrulanması

mqcertck komutu, kuyruk yöneticisinin TLS yapılandırmasının yanı sıra istemci anahtar havuzlarını doğrulama yeteneğine sahiptir. Bunu yapmak için **mqcertck** ' in kuyruk yöneticisini çalıştıran makineden istemcinin anahtar havuzuna erişebilmesi gerekir.

mqcertck komutunu çalıştırırken, **-clientkeyr** değiştirgesini istemci anahtar havuzunun yeriyle birlikte sağlarsanız (uzantı dışında) **mqcertck** , bu anahtar havuzunu kuyruk yöneticisiyle karşılaştırılarak denetler.

İstemcinin kuyruk yöneticisine bağlanmak için kullanacağı kanalı biliyorsanız, bunu **-clientchannel** işaretiyle belirtebilirsiniz.

İstemci kuyruk yöneticisine bağlanmak için karşılıklı kimlik doğrulaması kullanıyorsa, istemci anahtar havuzunda hangi sertifikanın kullanılacağını **mqcertck** komutuna söylemek için **-clientusername** ya da **-clientlabel** parametresini kullanabilirsiniz.

Varsayılan sertifikayı kullanıyorsanız ve istemci uygulamasına sertifika etiketi sağlamıyorsanız, bu uygulamayı çalıştıran **-clientusername** ve **username** değiştirgelerini kullanabilirsiniz.

mqcertck komutunun çalışması sırasında komut, `ibmwebspheremq`XXXX sertifika etiketini oluşturur; burada XXXX , **-clientusername** değiştirgesinde geçirilen değerdir.

İstemci anahtar havuzunu tam olarak doğrulamak için **mqcertck** komutu, IBM Global Security Kit (GSKit)komutunu kullanarak kukla bir bağlantı yaratır. Bunu yapmak için, komutun istemci sınamaları sırasında bağlanabileceği bir kapısı olması gerekir. Kullanılan varsayılan kapı şudur: 5857; ancak, bu kapı kullanımdaysa, istemci sınamaları sırasında kullanılmak üzere farklı bir kapı belirtebilirsiniz.

Not: mqcertck komutu bir kapıya bağlansa da, **mqcertck**tarafından dış iletişim kullanılmaz ve tüm sınamalar yerel olarak gerçekleştirilir.

IBM MQ MQI client üzerinde SSL/TLS

IBM MQ , istemcilerde TLS ' yi destekler. TLS kullanımını çeşitli şekillerde uyarlayabilirsiniz.

IBM MQ , AIX, Linux, and Windows sistemlerinde IBM MQ MQI clients için TLS desteği sağlar. IBM MQ classes for Javakullanıyorsanız, bkz. IBM MQ classes for Java ' yi kullanma ve IBM MQ classes for JMSkullanıyorsanız, bkz. IBM MQ classes for JMS ' yi kullanma. Bu bölümün geri kalanı Java ya da JMS ortamları için geçerli değildir.

Bir IBM MQ MQI client için anahtar havuzunu, IBM MQ istemci yapılandırma dosyanızdaki MQSSLKEYR değeriyle ya da uygulamanız bir MQCONNX çağrısı yaptığında belirtebilirsiniz. Bir kanalın TLS kullandığını belirtmek için üç seçeneğiniz vardır:

- Kanal tanımlama çizelgesinin kullanılması
- MQCONNX çağrısında SSL yapılandırma seçenekleri yapısının (MQSCO) kullanılması
- Active Directory ' yi kullanma (Windows sistemlerinde)

Bir kanalın TLS kullandığını belirtmek için MQSERVER ortam değişkenini kullanamazsınız.

TLS, kanalın diğer ucunda belirtilmediği sürece var olan IBM MQ MQI client uygulamalarınızı TLS olmadan çalıştırmaya devam edebilirsiniz.

Bir istemci makinesinde TLS Anahtar Havuzu, TLS Anahtar Havuzu, Kimlik Doğrulama Bilgileri ya da Şifreleme donanım parametrelerinde değişiklik yapıldıysa, uygulamanın kuyruk yöneticisine bağlanmak için kullandığı istemci bağlantısı kanallarında bu değişiklikleri yansıtmak için tüm TLS bağlantılarını sonlandırmanız gerekir. Tüm bağlantılar sona erdikten sonra TLS kanallarını yeniden başlatın. Tüm yeni TLS ayarları kullanılır. Bu ayarlar, kuyruk yöneticisi sistemlerinde REFRESH SECURITY TYPE (SSL) komutuyla yenilenenlere benzer.

IBM MQ MQI client ürününüz şifreleme donanımına sahip bir AIX, Linux, and Windows sisteminde çalıştığında, o donanımı MQSSLCRYP ortam değişkeniyle yapılandırırsınız. Bu değişken, ALTER QMGR MQSC komutundaki SSLCRYP değiştirgesine eşdeğerdir. ALTER QMGR MQSC komutundaki SSLCRYP değiştirgesinin tanımı için ALTER QMGR belgesine bakın. SSLCRYP parametresinin GSK_PCS11 sürümünü kullanırsanız, PKCS #11 simge etiketi tamamen küçük harfli olarak belirtilmelidir.

TLS gizli anahtar sıfırlaması ve FIPS IBM MQ MQI clientsüzerinde desteklenir. Daha fazla bilgi için bkz. “SSL ve TLS gizli anahtarlarını sıfırlama” sayfa 478 ve “AIX, Linux, and Windows için Federal Bilgi İşleme Standartları (FIPS)” sayfa 34.

IBM MQ MQI clientsTLS desteği hakkında daha fazla bilgi için bkz. “IBM MQ MQI client güvenliğinin ayarlanması” sayfa 264 .

İlgili görevler

IBM MQ MQI client yapılandırma dosyası, `mqclient.ini`

Bir MQI kanalının SSL/TLS kullandığını belirtme

Bir MQI kanalının TLS kullanabilmesi için, istemci bağlantısı kanalının *SSLCipherSpec* özniteliğinin değeri, istemci altyapısında IBM MQ tarafından desteklenen bir CipherSpec adı olmalıdır.

Aşağıdaki yöntemlerle, bu öznitelik için bir değer içeren bir istemci-bağlantı kanalı tanımlayabilirsiniz. Bunlar, azalan öncelik sırasına göre listelenir.

1. PreConnect çıkışı, kullanılacak bir kanal tanımlama yapısı sağladığında.

   PreConnect çıkışı, bir kanal tanımlaması yapısının (MQCD) *SSLCipherSpec* alanında CipherSpec adını sağlayabilir. Bu yapı, PreConnect çıkışı tarafından kullanılan MQNXP çıkış değiştirgesi yapısının **ppMQCDArrayPtr** alanında döndürülür.
2. Bir IBM MQ MQI client uygulaması bir MQCONNX çağrısı yayınladığında.

   Uygulama, bir kanal tanımlaması yapısının (MQCD) *SSLCipherSpec* alanında CipherSpec adını belirtebilir.Bu yapıya, MQCONNX çağrısındaki bir değiştirge olan bağlanma seçenekleri yapısı (MQCNO) gönderme yapıyor.
3. İstemci kanal tanımlama çizelgesinin (CCDT) kullanılması.

   Bir istemci kanal tanımlama çizelgesindeki bir ya da daha çok giriş CipherSpecadını belirtebilir.Örneğin, DEFINE CHANNEL MQSC komutunu kullanarak bir giriş yaratırsanız, bir CipherSpecadını belirtmek için komuttaki SSLCIPH parametresini kullanabilirsiniz.
4. Windowsüzerinde Active Directory kullanılıyor.

   Windows sistemlerinde, istemci bağlantısı kanal tanımlarını Active Directory' de yayınlamak için **setmqscp** denetim komutunu kullanabilirsiniz. Bu tanımlamalardan biri ya da daha fazlası CipherSpecadını belirtebilir.

Örneğin, bir istemci uygulaması MQCONNX çağrısındaki bir MQCD yapısında bir istemci bağlantısı kanal tanımlaması sağlıyorsa, bu tanımlama, IBM MQ istemcisi tarafından erişilebilen bir istemci kanal tanımlama çizelgesindeki girişlerin yerine kullanılır.

TLS kullanan bir MQI kanalının istemci ucunda kanal tanımlamasını sağlamak için MQSERVER ortam değişkenini kullanamazsınız.

Bir istemci sertifikasının akıp akmadığını denetlemek için, bir eş ad parametresi değerinin varlığına ilişkin kanal durumunu bir kanalın sunucu ucunda görüntüleyin.

İlgili kavramlar

"IBM MQ MQI client için CipherSpec belirtilmesi" sayfa 456
IBM MQ MQI clientiçin CipherSpec belirtmek üzere üç seçeneğiniz vardır.

IBM MQ içinde CipherSpecs (Şifre Belirtimleri) ve CipherSuites (CipherSuites)

IBM MQ , TLS1.3 ve TLS 1.2 CipherSpecsve RSA ve Diffie-Hellman algoritmalarını destekler. Ancak, gerekiyorsa, kullanımdan kaldırılan CipherSpecsöğesini etkinleştirebilirsiniz.

Aşağıdakilere ilişkin bilgi için bkz: "CipherSpecs Özelliğinin Etkinleştirilmesi" sayfa 432

- IBM MQtarafından desteklenen CipherSpecs (Şifre Belirtimleri).
- Kullanımdan kaldırılan SSL 3.0 ve TLS 1.0 CipherSpecs' i nasıl etkinleştirdiğiniz.

IBM MQ , RSA ve Diffie-Hellman anahtar değiş tokuşu ve kimlik doğrulama algoritmalarını destekler. TLS el sıkışması sırasında kullanılan anahtarın boyutu, kullandığınız dijital sertifikaya bağlı olabilir, ancak bazı CipherSpecs tokalaşma anahtarı boyutu belirtimini içerir. Daha büyük tokalaşma anahtarı boyutları daha güçlü kimlik doğrulaması sağlar. Daha küçük anahtar boyutlarıyla, tokalaşma daha hızlıdır.

İlgili kavramlar

"CipherSpecs ve CipherSuites" sayfa 21
Kriptografik güvenlik protokolleri, güvenli bir bağlantı tarafından kullanılan algoritmalar üzerinde anlaşmalıdır. CipherSpecs ve CipherSuites algoritmaların belirli birleşimlerini tanımlar.

IBM MQ içinde NSA Suite B Şifreleme

Bu konu, Suite B uyumlu TLS 1.2 profiline uymak için IBM MQ for AIX, Linux, and Windows ' in nasıl yapılandırılacağı hakkında bilgi sağlar.

Zamanla, NSA Şifreleme Takımı B Standardı, şifreleme algoritmalarına ve protokollerine karşı yeni saldırıları yansıtacak şekilde güncellenir. Örneğin, bazı CipherSpecs Ürün Grubu B sertifikalı olmayabilir. Bu tür değişiklikler gerçekleştiğinde, IBM MQ da en son standardı uygulayacak şekilde güncellenir. Sonuç olarak, bakım uygulandıktan sonra davranıştaki değişiklikleri görebilirsiniz. IBM MQ benioku dosyası, her ürün bakım düzeyi tarafından uygulanan Suite B sürümünü listeler. IBM MQ ürününü Suite B uyumluluğunu zorlayacak şekilde yapılandırırsanız, bakım uygulamayı planlarken her zaman benioku dosyasına bakın. Bkz. IBM MQ, WebSphere MQve MQSeries ürün benioku bilgileri.

AIX, Linux, and Windows sistemlerinde IBM MQ , Tablo 1 'de gösterilen güvenlik düzeylerinde Suite B uyumlu TLS 1.2 profiline uymak için yapılandırılabilir.

Çizelge 3. İzin verilen CipherSpecs ve dijital imza algoritmalarıyla Suite B güvenlik düzeyleri

Güvenlik Düzeyi	**İzin Verilen CipherSpecs**	**İzin verilen dijital imza algoritmaları**
128 bit	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	SHA-256 ile ECDSA SHA-384 ile ECDSA
192 bit	ECDHE_ECDSA_AES_256_GCM_SHA384	SHA-384 ile ECDSA
Her ikisi de [1]	ECDHE_ECDSA_AES_128_GCM_SHA256 ECDHE_ECDSA_AES_256_GCM_SHA384	SHA-256 ile ECDSA SHA-384 ile ECDSA

1. 128 bit ve 192 bit güvenlik düzeylerini eşzamanlı olarak yapılandırmak mümkündür. Suite B yapılandırması kabul edilebilir en düşük şifreleme algoritmalarını belirlediğinden, her iki güvenlik düzeyinin yapılandırılması yalnızca 128 bit güvenlik düzeyinin yapılandırılmasıyla eşdeğerdir. 192 bitlik güvenlik düzeyinin şifreleme algoritmaları, 128 bitlik güvenlik düzeyi için gerekli alt sınırdan daha güçlüdür, bu nedenle 192 bitlik güvenlik düzeyi etkinleştirilmese bile 128 bitlik güvenlik düzeyi için bunlara izin verilir.

Not: Güvenlik düzeyi için kullanılan adlandırma kuralları, eliptik eğri boyutunu ya da AES şifreleme algoritmasının anahtar boyutunu göstermez.

CipherSpec Suite B ' ye uyumluluk

IBM MQ varsayılan davranışı Suite B standardına uymamakla birlikte, IBM MQ AIX, Linux, and Windows sistemlerinde güvenlik düzeylerinden birine ya da her ikisine de uyacak şekilde yapılandırılabilir. Takım B 'yi kullanmak için IBM MQ yapılandırmasının başarılı bir şekilde gerçekleştirilmesinin ardından, CipherSpec kullanılarak bir giden kanal başlatma girişimi, Suite B' ye uymayan AMQ9282hatasıyla sonuçlanır. Bu etkinlik, MQI istemcisinin MQRC_CIPHER_SPEC_NOT_SUITE_Bneden kodunu döndürmesi ile de sonuçlanır. Benzer şekilde, Suite B yapılandırmasına uymayan bir CipherSpec kullanılarak bir gelen kanal başlatılmaya çalışılması AMQ9616hatasıyla sonuçlanır.

IBM MQ CipherSpecshakkında daha fazla bilgi için bkz. “CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432

Suite B ve dijital sertifikalar

Suite B, dijital sertifikaları imzalamak için kullanılabilecek dijital imza algoritmalarını sınırlandırır. Takım B, sertifikaların içerebileceği ortak anahtar tipini de sınırlar. Bu nedenle IBM MQ , dijital imza algoritmasına ve genel anahtar tipine, uzak iş ortağının yapılandırılmış Suite B güvenlik düzeyi tarafından izin verilen sertifikaları kullanacak şekilde yapılandırılmalıdır. Güvenlik düzeyi gereksinimleriyle uyumlu olmayan sayısal sertifikalar reddedilir ve bağlantı AMQ9633 ya da AMQ9285hatasıyla başarısız olur.

128 bit Suite B güvenlik düzeyi için sertifika öznesinin açık anahtarının NIST P-256 eliptik eğrisini ya da NIST P-384 eliptik eğrisini kullanması ve NIST P-256 eliptik eğrisiyle ya da NIST P-384 eliptik eğrisiyle imzalanması gerekir. 192 bitlik Suite B güvenlik düzeyinde, sertifika konusunun açık anahtarının NIST P-384 eliptik eğrisini kullanması ve NIST P-384 eliptik eğrisiyle imzalanması gerekir.

Takım B uyumlu işleme uygun bir sertifika edinmek için **runmqakm** komutunu kullanın ve uygun bir sayısal imza algoritması istemek için **-sig_alg** parametresini belirtin. EC_ecdsa_with_SHA256 ve EC_ecdsa_with_SHA384 **-sig_alg** parametre değerleri, izin verilen Suite B dijital imza algoritmaları tarafından imzalanmış eliptik eğri anahtarlarına karşılık gelir.

runmqakm komutuyla ilgili daha fazla bilgi için bkz. runmqckm ve runmqakm seçenekleri.

Not: runmqckm ve **strmqikm** komutları, Suite B uyumlu işlem için dijital sertifikaların oluşturulmasını desteklemez.

Dijital sertifikaların oluşturulması ve istenmesi

Suite B testi için kendinden onaylı bir dijital sertifika oluşturmak üzere bkz. “AIX, Linux, and Windows üzerinde kendinden onaylı kişisel sertifika oluşturma” sayfa 303

Suite B üretim kullanımı için CA imzalı bir dijital sertifika istemek üzere bkz. “AIX, Linux, and Windows üzerinde kişisel sertifika isteme” sayfa 306.

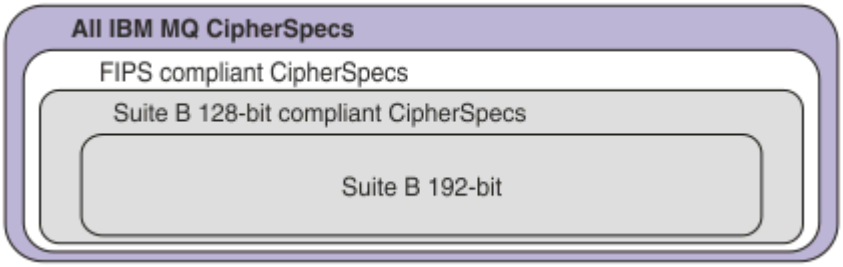
Not: Kullanılan sertifika yetkilisi, IETF RFC 6460 'da açıklanan gereksinimleri karşılayan sayısal sertifikalar oluşturmalıdır.

FIPS 140-2 ve Suite B

Not: AIX, Linux, and Windowsişletim sistemlerinde IBM MQ , IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, IBM Crypto for C (ICC) sertifikasını görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu İşlem listesindeki NIST CMVP modüllerindearanarak görüntülenebilir.

Suite B standardı, güvenli bir güvenlik düzeyi sağlamak için etkinleştirilmiş şifreleme algoritmaları kümesini kısıtladığı için kavramsal olarak FIPS 140-2 ile benzerdir. IBM MQ FIPS 140-2 uyumlu işlem için yapılandırıldığında, şu anda desteklenen Suite B CipherSpecs kullanılabilir. Bu nedenle IBM MQ hem FIPS hem de Suite B uyumluluğu için eşzamanlı olarak yapılandırılabilir; bu durumda her iki kısıtlama kümesi de geçerli olur.

Aşağıdaki çizgede bu altkümeler arasındaki ilişki gösterilir:



IBM MQ ürününü Suite B uyumlu işlem için yapılandırma

Suite B uyumlu işlem için AIX, Linux, and Windows üzerinde IBM MQ ' nin nasıl yapılandırılacağı hakkında bilgi için bkz. “IBM MQ ürününü Suite B için yapılandırma” sayfa 43.

IBM MQ , IBM i ve z/OS platformlarında Suite B uyumlu işlemi desteklemez. IBM MQ Java ve JMS istemcileri de Suite B uyumlu işlemi desteklemez.

İlgili kavramlar

“MQI istemcisinde çalıştırma zamanında yalnızca FIPS onaylı CipherSpecs kullanılmasının belirtilmesi” sayfa 265

FIPS uyumlu yazılımı kullanarak anahtar havuzlarınızı oluşturun ve daha sonra, kanalın FIPS onaylı CipherSpecsözelliğini kullanması gerektiğini belirtin.

ALW *IBM MQ ürününü Suite B için yapılandırma*

IBM MQ , AIX, Linux, and Windows platformlarında NSA Suite B standardına uygun olarak çalışacak şekilde yapılandırılabilir.

Suite B, güvenli bir güvenlik düzeyi sağlamak için etkinleştirilmiş şifreleme algoritmaları kümesini sınırlar. IBM MQ , geliştirilmiş bir güvenlik düzeyi sağlamak için Suite B ile uyumlu çalışacak şekilde yapılandırılabilir. Suite B hakkında daha fazla bilgi için bkz. “Ulusal Güvenlik Ajansı (NSA) Suite B Şifrelemesi” sayfa 23. Suite B yapılandırması ve TLS kanalları üzerindeki etkisi hakkında daha fazla bilgi için bkz. “IBM MQ içinde NSA Suite B Şifreleme” sayfa 41.

Kuyruk yöneticisi

Bir kuyruk yöneticisinde, gerekli güvenlik düzeyinize uygun değerleri ayarlamak için **SUITEB** parametresiyle birlikte **ALTER QMGR** komutunu kullanın. Ek bilgi için bkz. ALTER QMGR.

Kuyruk yöneticisini Suite B uyumlu işlem için yapılandırmak üzere **MQIA_SUITE_B_STRENGTH** parametresiyle birlikte PCF **MQCMD_CHANGE_Q_MGR** komutunu da kullanabilirsiniz.

Not: Bir kuyruk yöneticisinin Suite B ayarlarını değiştirirseniz, bu ayarların yürürlüğe girmesi için MQXR hizmetini yeniden başlatmanız gerekir.

MQI istemcisi

Varsayılan olarak, MQI istemcileri Suite B uyumluluğunu uygulamaz. Aşağıdaki seçeneklerden birini yürüterek MQI istemcisini Suite B uyumluluğu için etkinleştirebilirsiniz:

1. MQCONNX çağrısında MQSCO yapısında EncryptionPolicySuiteB alanını aşağıdaki değerlerden birine ya da daha fazlasına ayarlayarak:
   - `MQ_SUITE_B_NONE`
   - `MQ_SUITE_B_128_BIT`
   - `MQ_SUITE_B_192_BIT`

   `MQ_SUITE_B_NONE` ' in başka bir değerle kullanılması geçersizdir.

   MQSCO yapısıyla ilgili daha fazla bilgi için MQSCO-SSL yapılandırma seçenekleribaşlıklı konuya bakın.

2. **MQSUITEB** ortam değişkenini aşağıdaki değerlerden birine ya da birkaçına ayarlayarak:
   - YOK
   - 128_BIT
   - 192_BIT

   Virgülle ayrılmış bir liste kullanarak birden çok değer belirtebilirsiniz. NONE değerinin başka bir değerle kullanılması geçersizdir.
3. İstemci yapılanış kütüğünün SSL kısmına ilişkin **EncryptionPolicySuiteB** özniteliğini aşağıdaki değerlerden birine ya da daha fazlasına ayarlayarak:
   - YOK
   - 128_BIT
   - 192_BIT

   Virgülle ayrılmış bir liste kullanarak birden çok değer belirtebilirsiniz. NONE değerinin başka bir değerle kullanılması geçersizdir.

Not: MQI istemcisi ayarları öncelik sırasına göre listelenir. MQCONNX çağrısındaki MSCO yapısı, SSL kıstasındaki özniteliği geçersiz kılan **MQSUITEB** ortam değişkenindeki ayarı geçersiz kılar.

.NET

.NET yönetilmeyen istemciler için **MQC.ENCRYPTION_POLICY_SUITE_B** özelliği, gerekli Suite B güvenliğinin tipini gösterir.

IBM MQ classes for .NETiçinde Suite B kullanımı hakkında bilgi için bkz. MQEnvironment .NET sınıfı.

AMQP

Bir kuyruk yöneticisine ilişkin Suite B öznitelik ayarları, o kuyruk yöneticisindeki AMQP kanallarına uygulanır. Kuyruk yöneticisi Suite B ayarlarını değiştirirseniz, değişikliklerin yürürlüğe girmesi için AMQP hizmetini yeniden başlatmanız gerekir.

IBM MQ içindeki sertifika doğrulama ilkeleri

Sertifika doğrulama ilkesi, sertifika zinciri doğrulamasının sektör güvenlik standartlarına tam olarak nasıl uyduğunu belirler.

Sertifika geçerlilik denetimi ilkesi, aşağıdaki gibi platforma ve ortama bağlıdır:

- Tüm platformlardaki Java ve JMS uygulamaları için sertifika geçerlilik denetimi ilkesi, Java yürütme ortamının JSSE bileşenine bağlıdır. Sertifika geçerlilik denetimi ilkesine ilişkin ek bilgi için JRE ' ye ilişkin belgelere bakın.
- ALW AIX, Linux, and Windows sistemleri için, sertifika doğrulama ilkesi IBM Global Security Kit (GSKit) tarafından sağlanır ve yapılandırılabilir. İki farklı sertifika doğrulama ilkesi desteklenir:
  - Geçerli IETF sertifikası doğrulama standartlarına uymayan eski dijital sertifikalarla en üst düzeyde geriye dönük uyumluluk ve birlikte çalışabilirlik için kullanılan eski bir sertifika doğrulama ilkesi. Bu ilke Temel ilke olarak bilinir.
  - RFC 5280 standardını uygulayan katı, standartlara uygun bir sertifika doğrulama ilkesi. Bu ilke Standart ilke olarak bilinir.
- IBM i IBM i sistemleri için sertifika doğrulama ilkesi, işletim sistemi tarafından sağlanan güvenli yuva kitaplığına bağlıdır. Sertifika geçerlilik denetimi ilkesine ilişkin ek bilgi için işletim sistemine ilişkin belgelere bakın.
- z/OS z/OS sistemleri için sertifika doğrulama ilkesi, işletim sistemi tarafından sağlanan Sistem SSL bileşenine bağlıdır. Sertifika geçerlilik denetimi ilkesine ilişkin ek bilgi için işletim sistemine ilişkin belgelere bakın.

Sertifika doğrulama ilkesinin nasıl yapılandırılacağı hakkında bilgi için bkz. “IBM MQ içinde sertifika doğrulama ilkelerini yapılandırma” sayfa 45. Temel ve Standart sertifika doğrulama ilkeleri arasındaki farklar hakkında daha fazla bilgi için bkz. AIX, Linux, and Windowsüzerinde sertifika doğrulama ve güven ilkesi tasarımı.

IBM MQ içinde sertifika doğrulama ilkelerini yapılandırma

Uzak iş ortağı sistemlerinden alınan dijital sertifikaları doğrulamak için hangi TLS sertifika doğrulama ilkesinin kullanılacağını belirtmenin birkaç farklı yolu vardır.

Bu görev hakkında

Sertifika doğrulama ilkesi, sertifika zinciri doğrulamasının sektör güvenlik standartlarına tam olarak nasıl uyduğunu belirler. Sertifika geçerlilik denetimi ilkesi, platforma ve ortama bağlıdır. Sertifika doğrulama ilkeleri hakkında daha fazla bilgi için bkz. “IBM MQ içindeki sertifika doğrulama ilkeleri” sayfa 44.

Yordam

- Kuyruk yöneticisinde sertifika geçerlilik denetimi ilkesini ayarlamak için **CERTVPOL**kuyruk yöneticisi özniteliğini kullanın.

  Bu özniteliğin ayarlanmasıyla ilgili ek bilgi için ALTER QMGR (kuyruk yöneticisi ayarlarının değiştirilmesi)başlıklı konuya bakın.
- İstemcide sertifika geçerlilik denetimi ilkesini ayarlamak için aşağıdaki yöntemleri kullanın.

  İlkeyi ayarlamak için birden çok yöntem kullanılırsa, istemci ayarları aşağıdaki öncelik sırasıyla kullanır:

  1. İstemci MQSCO yapısındaki `CertificateValPolicy` alanını kullanın. Alanı aşağıdaki değerlerden birine ayarlayın:

     MQ_CERT_VAL_POLICY_ANY
     Güvenli yuva kitaplığı tarafından desteklenen sertifika geçerlilik denetimi ilkelerinin her birini uygulayın. İlkelerden herhangi biri sertifika zincirini geçerli sayarsa sertifika zincirini kabul edin.

     MQ_CERT_VAL_POLICY_RFC5280
     Yalnızca RFC5280 uyumlu sertifika doğrulama ilkesini uygulayın. Bu ayar, HERHANGİ BİRİ ayarından daha sıkı doğrulama sağlar, ancak bazı eski dijital sertifikaları reddeder.

     Bu alanı kullanma hakkında daha fazla bilgi için bkz. MQSCO-SSL yapılandırma seçenekleri.
  2. **MQCERTVPOL**istemci ortam değişkenini kullanın. Bu ortam değişkenini ayarlamak için aşağıdaki komutlardan birini kullanın:
     – Linux AIX AIX and Linux sistemleri için:

     ```
     export MQCERTVPOL= value
     ```

     – Windows Windows sistemleri için:

     ```
     SET MQCERTVPOL= value
     ```

     – IBM i IBM i sistemleri için:

     ```
     ADDENVVAR ENVVAR(MQCERTVPOL) VALUE(value)
     ```
  3. İstemci yapılanış kütüğündeki SSL ' nin **CertificateValPolicy** özniteliğini kullanın. Bu özniteliği aşağıdaki değerlerden birine ayarlayın:

     Fark Etmez
     Temeldeki güvenli yuva kitaplığı tarafından desteklenen herhangi bir sertifika geçerlilik denetimi ilkesini kullanın. Bu ayar varsayılan ayardır.

RFC5280
Yalnızca RFC 5280 standardına uyan sertifika geçerlilik denetimini kullanın.

Bu özniteliğin kullanılmasıyla ilgili ek bilgi için İstemci yapılanış kütüğünün SSL kısmıbaşlıklı konuya bakın.

IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu

Bu konuda, IBM MQiçinde CipherSpecs ile dijital sertifikalar arasındaki ilişkiyi özetleyerek güvenlik ilkeniz için uygun CipherSpecs ve dijital sertifikaların nasıl seçileceğine ilişkin bilgiler sağlanır.

Desteklenen tüm sayısal sertifika tipleriyle yalnızca desteklenen CipherSpecs alt kümesi kullanılabilir. Bu nedenle, sayısal sertifikanız için uygun bir CipherSpec seçmeniz gerekir. Benzer şekilde, kuruluşunuzun güvenlik ilkesi belirli bir CipherSpec ' i kullanmanızı gerektiriyorsa, bu CipherSpeciçin uygun bir sayısal sertifika edinmeniz gerekir.

MD5 dijital imza algoritması ve TLS 1.2

MD5 algoritması kullanılarak imzalanan sayısal sertifikalar, TLS 1.2 protokolü kullanıldığında reddedilir. Bunun nedeni, MD5 algoritmasının artık birçok şifreleme analisti tarafından zayıf kabul edilmesi ve kullanımının genellikle önerilmez olmasıdır. TLS 1.2 iletişim kuralına dayalı daha yeni CipherSpecs kullanmak için dijital sertifikaların dijital imzalarında MD5 algoritmasını kullanmadığından emin olun. TLS 1.0 iletişim kurallarını kullanan daha eski CipherSpecs bu kısıtlamaya tabi değildir ve MD5 dijital imzaları olan sertifikaları kullanmaya devam edebilir.

Belirli bir sertifikaya ilişkin sayısal imza algoritmasını görüntülemek için **runmqakm** komutunu kullanabilirsiniz:

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

Burada *cert_label* , görüntülenecek dijital imza algoritmasının sertifika etiketidir. Ayrıntılar için bkz. Dijital sertifika etiketleri .

Not: runmqckm (iKeycmd) ve **strmqikm** (iKeyman) GUI 'si bir dizi dijital imza algoritmasını görüntülemek için kullanılabilse de, **runmqakm** aracı daha geniş bir aralık sağlar.

runmqakm komutunun çalıştırılması, belirtilen imza algoritmasının kullanımını görüntüleyen bir çıkış üretir:

```
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
```

`Signature Algorithm` satırı, `MD5WithRSASignature` algoritmasının kullanıldığını gösterir. Bu algoritma MD5 ' i temel alır ve bu dijital sertifika TLS 1.2 CipherSpecsile kullanılamaz.

Eliptik Eğri ve RSA CipherSpecs Birlikte Çalışabilirlik

Tüm CipherSpecs dijital sertifikalarla birlikte kullanılamaz. CipherSpecs , CipherSpec ad önekiyle gösterilir. Her CipherSpec tipi, kullanılabilecek sayısal sertifika tipi üzerinde farklı kısıtlamalar uygular. Bu kısıtlamalar tüm IBM MQ TLS bağlantıları için geçerlidir, ancak özellikle Eliptik Eğri şifreleme kullanıcıları için geçerlidir.

Aşağıdaki tablo, CipherSpecs ile dijital sertifikalar arasındaki ilişkileri özetler:

Çizelge 4. CipherSpecs ile dijital sertifikalar arasındaki ilişkiler

Tip	CipherSpec Ad Öneki	Açıklama	Gerekli genel anahtar tipi	Dijital imza şifreleme algoritması	Gizli anahtar oluşturma yöntemi
1	ECDHE_ECDSA_	Eliptik Eğri ve genel anahtarları, Eliptik Eğri gizli tuşları ve Eliptik Eğri sayısal imza algoritmalarını kullanan CipherSpecs (Şifre Belirtimleri).	Eliptik Eğri	ECDSA	ECDHE
2	ECDHE_RSA_	RSA ortak anahtarlarını, Eliptik Eğri gizli anahtarlarını ve RSA dijital imza algoritmalarını kullanan CipherSpecs (Şifre Belirtimleri).	RSA	RSA	ECDHE
3	(Tüm TLS 1.3 CipherSpecs)	Eliptik Eğri ya da RSA genel anahtarlarını, Eliptik Eğri gizli anahtarlarını ve Eliptik Eğri ya da RSA sayısal imza algoritmalarını kullanan CipherSpecs .	Eliptik Eğri ya da RSA	ECDSA ya da RSA	ECDHE ya da RSA
4	(Diğerleri)	RSA ortak anahtarlarını ve RSA dijital imza algoritmalarını kullanan CipherSpecs (Şifre Belirtimleri).	RSA	RSA	RSA

Not: Tip 1 ve 2 CipherSpecs , IBM i altyapısında IBM MQ kuyruk yöneticileri ve MQI istemcileri tarafından desteklenmez.

Gerekli genel anahtar tipi kolonu, her CipherSpectipi kullanılırken kişisel sertifikanın sahip olması gereken genel anahtarın tipini gösterir. Kişisel sertifika, kuyruk yöneticisini ya da istemciyi uzak ortağına tanıtan son varlık sertifikasıdır.

Sertifika etiketinde adı belirtilen sertifikanın CipherSpeckanalı için uygun olduğundan emin olmanız gerekir. Yani, Elliptic Curve (EC) sertifikası gerektiren bir CipherSpec ile bir kanal yapılandırırsanız,

sertifika etiketinde bir RSA sertifikası adlanamaz. RSA sertifikası gerektiren bir CipherSpec ile bir kanal yapılandırırsanız, sertifika etiketinde bir EC sertifikasını adlayamazsınız.

IBM MQ' in doğru şekilde yapılandırıldığını varsayarak şunları yapabilirsiniz:

- RSA ve EC sertifikalarının karışımını içeren tek bir kuyruk yöneticisi.
- Aynı kuyruk yöneticisinde RSA ya da EC sertifikası kullanan farklı kanallar.

Dijital imza şifreleme algoritması, eşin geçerliliğini denetlemek için kullanılan şifreleme algoritmasına başvurur. Şifreleme algoritması, sayısal imzayı hesaplamak için MD5, SHA-1 ya da SHA-256 gibi bir hash algoritmasıyla birlikte kullanılır. Çeşitli sayısal imza algoritmaları kullanılabilir; örneğin, MD5 ile RSA ya da SHA-256ile ECDSA. Tabloda ECDSA, ECDSA kullanan dijital imza algoritmaları kümesini ifade eder; RSA, RSA kullanan dijital imza algoritmaları kümesini ifade eder. Belirtilen şifreleme algoritmasına dayalı olması koşuluyla, sette desteklenen herhangi bir dijital imza algoritması kullanılabilir.

Tip 1 CipherSpecs , kişisel sertifikanın bir Eliptik Eğri genel anahtarına sahip olmasını gerektirir. Bu CipherSpecs kullanıldığında, bağlantıya ilişkin gizli anahtarı oluşturmak için Elliptic Curve Diffie Hellman Ephemeral anahtar sözleşmesi kullanılır.

Tip 2 CipherSpecs , kişisel sertifikanın bir RSA genel anahtarına sahip olmasını gerektirir. Bu CipherSpecs kullanıldığında, bağlantıya ilişkin gizli anahtarı oluşturmak için Elliptic Curve Diffie Hellman Ephemeral anahtar sözleşmesi kullanılır.

Tip 3 CipherSpecs , kişisel sertifikanın bir RSA genel anahtarına sahip olmasını gerektirir. Bu CipherSpecs kullanıldığında, bağlantıya ilişkin gizli anahtarı oluşturmak için RSA anahtar değişimi kullanılır.

Bu kısıtlama listesi ayrıntılı değildir: Yapılandırmaya bağlı olarak, birlikte çalışma yeteneğini daha fazla etkileyebilecek ek kısıtlamalar olabilir. Örneğin, IBM MQ FIPS 140-2 ya da NSA Suite B standartlarına uyacak şekilde yapılandırıldıysa, bu işlem izin verilen yapılandırma aralığını da sınırlar. Daha fazla bilgi için aşağıdaki bölüme bakın.

Aynı kuyruk yöneticisinde ya da istemci uygulamasında farklı tiplerde CipherSpec kullanmanız gerekirse, istemci tanımlamasında uygun bir sertifika etiketi ve CipherSpec birleşimi yapılandırın.

Üç CipherSpec tipi doğrudan birlikte çalışmaz: Bu, yürürlükteki TLS standartlarının bir sınırlamasıdır. Örneğin, QM1adlı bir kuyruk yöneticisinde TO.QM1 adlı bir alıcı kanalı için ECDHE_ECDSA_AES_128_CBC_SHA256 CipherSpec ' ı kullanmayı seçtiğinizi varsayın; bu durumda, alıcının Eliptik Eğri anahtarı ve ECDSA tabanlı sayısal imzası olan kişisel bir sertifikası olmalıdır. Alıcı kanal bu gereksinimleri karşılamıyorsa, kanal başlatılamıyor.

QM1 kuyruk yöneticisine bağlanan diğer kanallar, her bir kanalın CipherSpec için doğru tipte bir sertifika kullanması koşuluyla diğer CipherSpecskanallarını kullanabilir. Örneğin, QM1 ' in QM2adlı başka bir kuyruk yöneticisine ileti göndermek için TO.QM2 adlı bir gönderen kanalı kullandığını varsayın. Kanal TO.QM2 Tip 3 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA256 , RSA genel anahtarlarını içeren kanal kullanım sertifikalarının her iki ucunu da içermesi koşuluyla kullanabilir. Sertifika etiketi kanal özniteliği, her kanal için farklı bir sertifika yapılandırmak için kullanılabilir.

IBM MQ ağlarınızı planlarken hangi kanalların TLS gerektirdiğini dikkatle göz önünde bulundurun ve her kanal için kullanılan sertifika tipinin o kanaldaki CipherSpec ile kullanılmaya uygun olduğundan emin olun.

Dijital sertifikaya ilişkin sayısal imza algoritmasını ve genel anahtar tipini görüntülemek için **runmqakm** komutunu kullanabilirsiniz:

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

Burada *cert_label* , dijital imza algoritmasını görüntülemeniz gereken sertifikanın etiketidir. Ayrıntılar için bkz. Dijital sertifika etiketleri .

runmqakm komutunun çalıştırılması, Genel Anahtar Tipini görüntüleyen bir çıkış üretecektir:

```
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
```

Bu durumda Genel Anahtar Tipi satırı, sertifikanın bir Eliptik Eğri ortak anahtarı olduğunu gösterir. Bu durumda İmza Algoritması satırı, EC_ecdsa_with_SHA384 algoritmasının kullanımda olduğunu gösterir: Bu, ECDSA algoritmasına dayalıdır. Bu nedenle bu sertifika yalnızca Tip 1 CipherSpecsile kullanıma uygundur.

runmqckm komutunu aynı parametrelerle de kullanabilirsiniz. Anahtar havuzunu açar ve sertifikanın etiketini çift tıklatırsanız, dijital imza algoritmalarını görüntülemek için **strmqikm** GUI de kullanılabilir. Ancak, daha geniş bir algoritma yelpazesini desteklediğinden dijital sertifikaları görüntülemek için **runmqakm** aracını kullanmalısınız.

TLS 1.3 CipherSpecs

TLS 1.3 CipherSpecs , hem ECDSA hem de RSA sertifikalarını destekler.

Eliptik Eğri CipherSpecs ve NSA Suite B

IBM MQ , Suite B uyumlu TLS 1.2 profiline uymak üzere yapılandırıldığında, izin verilen CipherSpecs ve dijital imza algoritmaları “IBM MQ içinde NSA Suite B Şifreleme” sayfa 41içinde açıklandığı şekilde kısıtlanır. Ayrıca, kabul edilebilir Eliptik Eğri tuşlarının aralığı, yapılandırılan güvenlik düzeylerine göre azaltılır.

128 bit Suite B güvenlik düzeyinde, sertifika konusunun açık anahtarının NIST P-256 ya da NIST P-384 eliptik eğrisini kullanması ve NIST P-256 eliptik eğrisiyle ya da NIST P-384 eliptik eğrisiyle imzalanması gerekir. **runmqakm** komutu, EC_ecdsa_with_SHA256ya da EC_ecdsa_with_SHA384-sig_alg değiştirgesini kullanarak bu güvenlik düzeyine ilişkin sayısal sertifikaları istemek için kullanılabilir.

192 bitlik Suite B güvenlik düzeyinde, sertifika konusunun açık anahtarının NIST P-384 eliptik eğrisini kullanması ve NIST P-384 eliptik eğrisiyle imzalanması gerekir. **runmqakm** komutu, EC_ecdsa_with_SHA384-sig_alg parametresi kullanılarak bu güvenlik düzeyine ilişkin sayısal sertifikaları istemek için kullanılabilir.

Desteklenen NIST eliptik eğrileri şunlardır:

Çizelge 5. Desteklenen NIST eliptik eğrileri		
NIST FIPS 186-3 eğri adı	**RFC 4492 eğri adı**	**Eliptik Eğri anahtar boyutu (bit)**
P-256	secp256r1	256
P-384	secp384r1	384
P-521	secp521r1	521

Not: NIST P-521 eliptik eğrisi, Suite B uyumlu işlem için kullanılamaz.

İlgili kavramlar

"CipherSpecs Özelliğinin Etkinleştirilmesi" sayfa 432
DEFINE CHANNEL ya da **ALTER CHANNEL** MQSC komutunda **SSLCIPH** değiştirgesini kullanarak CipherSpec ' i etkinleştirin.

"MQI istemcisinde çalıştırma zamanında yalnızca FIPS onaylı CipherSpecs kullanılmasının belirtilmesi" sayfa 265
FIPS uyumlu yazılımı kullanarak anahtar havuzlarınızı oluşturun ve daha sonra, kanalın FIPS onaylı CipherSpecsözelliğini kullanması gerektiğini belirtin.

"IBM MQ içinde NSA Suite B Şifreleme" sayfa 41
Bu konu, Suite B uyumlu TLS 1.2 profiline uymak için IBM MQ for AIX, Linux, and Windows ' in nasıl yapılandırılacağı hakkında bilgi sağlar.

"Ulusal Güvenlik Ajansı (NSA) Suite B Şifrelemesi" sayfa 23
Amerika Birleşik Devletleri hükümeti, veri şifrelemesi de dahil olmak üzere BT sistemleri ve güvenliği konusunda teknik öneriler üretiyor. ABD Ulusal Güvenlik Ajansı (NSA), Suite B standardında bir dizi birlikte çalışabilir şifreleme algoritması önermektedir.

Kanal kimlik doğrulama kayıtları

Kanal düzeyinde bağlanan sistemlere verilen erişim üzerinde daha kesin denetim sağlamak için kanal kimlik doğrulama kayıtlarını kullanabilirsiniz.

İstemcilerin kuyruk yöneticinizle boş bir kullanıcı kimliği ya da istemcinin istenmeyen işlemler gerçekleştirmesine olanak sağlayacak üst düzey bir kullanıcı kimliği kullanarak bağlantı kurmaya çalıştıklarından haberiniz olabilir. Kanal kimlik doğrulama kayıtlarını kullanarak bu istemcilere erişimi engelleyebilirsiniz. Diğer bir seçenek olarak, bir istemci istemci altyapısında geçerli olan ancak bilinmeyen ya da sunucu altyapısında geçersiz bir biçimi olan bir kullanıcı kimliğini belirleyebilir. Bildirili kullanıcı kimliğini geçerli bir kullanıcı kimliğiyle eşlemek için bir kanal kimlik doğrulama kaydı kullanabilirsiniz.

Kuyruk yöneticinizle bağlantı kuran ve bir şekilde kötü davranan bir istemci uygulaması bulabilirsiniz. Sunucuyu, bu uygulamanın neden olduğu sorunlardan korumak için, güvenlik duvarı kuralları güncelleninceye ya da istemci uygulaması düzeltilinceye kadar istemci uygulamasının açık olduğu IP adresi kullanılarak geçici olarak engellenmesi gerekir. İstemci uygulamasının bağlandığı IP adresini engellemek için bir kanal kimlik doğrulama kaydı kullanabilirsiniz.

IBM MQ Explorergibi bir yönetim aracı ve bu belirli kullanım için bir kanal oluşturdıysanız, yalnızca belirli istemci bilgisayarların bu aracı kullanabildiğinden emin olmak isteyebilirsiniz. Kanalın yalnızca belirli IP adreslerinden kullanılmasına izin vermek için bir kanal doğrulama kaydı kullanabilirsiniz.

İstemci olarak çalışan bazı örnek uygulamaları kullanmaya yeni başlıyorsanız, kanal kimlik doğrulama kayıtlarını kullanarak kuyruk yöneticisini güvenli bir şekilde ayarlamaya ilişkin bir örnek için Örnek programları hazırlama ve çalıştırma başlıklı konuya bakın.

Gelen kanalları denetlemek üzere kanal kimlik doğrulama kayıtlarını almak için **ALTER QMGR CHLAUTH(ENABLED)** MQSC komutunu kullanın.

CHLAUTH kuralları, yeni gelen bağlantıya yanıt olarak oluşturulan bir kanal MCA için uygulanır. Yerel olarak başlatılan kanala yanıt olarak oluşturulan bir kanal MCA için **CHLAUTH** kuralları uygulanmaz.

Çizelge 6. CHLAUTH kurallarının farklı kanal çiftleri için uygulandığı yer	
Kanal tipi	**CHLAUTH kurallarının uygulandığı MCA**
SDR-RCVR	RCVR
RQSTR-SVR (SVR ' de başlatıldı)	RQSTR
RQSTR-SVR (RQSTR ' de başlatıldı)	SVR
RQSTR-SDR (SDR ' de başlatıldı)	RQSTR
RQSTR-SDR (RQSTR ' de başlatıldı)	İlk bağlantı için SDR. Geri çağırma bağlantısı için RQSTR.

Aşağıdaki işlevleri gerçekleştirmek için kanal doğrulama kayıtları oluşturulabilir:

- Belirli IP adreslerinden bağlantıları engellemek için.
- Belirli kullanıcı kimliklerinden gelen bağlantıları engellemek için.
- Belirli bir IP adresinden bağlanan herhangi bir kanal için kullanılacak bir MCAUSER değeri ayarlamak için.
- Belirli bir kullanıcı kimliğini belirlemeye yönelik herhangi bir kanal için kullanılacak bir MCAUSER değeri ayarlamak için.
- Belirli bir SSL ya da TLS Ayırt Edici Adı (DN) olan herhangi bir kanal için kullanılacak bir MCAUSER değeri ayarlamak için.
- Belirli bir kuyruk yöneticisinden bağlantı kuran herhangi bir kanal için kullanılacak bir MCAUSER değeri ayarlamak için.
- Bağlantı belirli bir IP adresinden olmadığı sürece, belirli bir kuyruk yöneticisinden olduğunu iddia eden bağlantıları engellemek için.
- Bağlantı belirli bir IP adresinden olmadığı sürece, belirli bir SSL ya da TLS sertifikası sunan bağlantıları engellemek için.

Bu kullanımlar aşağıdaki bölümlerde daha ayrıntılı olarak açıklanmıştır.

SET CHLAUTH ya da PCF komutunu **Set Channel Authentication Record**kullanarak kanal kimlik doğrulama kayıtları yaratabilir, bunları değiştirebilir ya da kaldırabilirsiniz.

Not: Çok sayıda kanal kimlik doğrulama kaydı, kuyruk yöneticisinin başarımını olumsuz etkileyebilir.

Engelleyici IP adresleri

Normalde belirli IP adreslerinden erişimi önlemek için bir güvenlik duvarının rolüdür. Ancak, IBM MQ sisteminize erişimi olmaması gereken bir IP adresinden bağlantı girişimleriyle karşılaşabileceğiniz ve güvenlik duvarının güncellenebilmesi için adresi geçici olarak engellemeniz gereken durumlar olabilir. Bu bağlantı girişimleri IBM MQ kanallarından gelmeyebilir; bu bağlantı girişimleri, IBM MQ dinleyicinizi hedeflemek için yanlış yapılandırılan diğer yuva uygulamalarından geliyor olabilir. BLOCKADDR tipinde bir kanal doğrulama kaydı ayarlayarak IP adreslerini engelleyin. Bir ya da daha çok tek adres, adres aralığı ya da genel arama karakterleri de içinde olmak üzere kalıp belirtebilirsiniz.

IP adresi bu şekilde engellendiği için gelen bir bağlantı reddedildiğinde, kanal olayları etkinleştirildiyse ve kuyruk yöneticisi çalışıyorsa, MQRQ_CHANNEL_BLOCKED olay iletisi ve neden niteleyicisi MQRQ_CHANNEL_BLOCKED_ADDRESS yayınlanır. Ayrıca, dinleyicinin engellenen bağlantı için yinelenen girişimler nedeniyle su basmadığından emin olmak için, bağlantı, hata döndürmeden önce 30 saniye boyunca açık tutulur.

Yalnızca belirli kanallardaki IP adreslerini engellemek ya da hata bildirilmeden önceki gecikmeyi önlemek için, USERSRC (NOACCESS) parametresiyle ADDRESSMAP tipinde bir kanal kimlik doğrulama kaydı ayarlayın.

Bu nedenle gelen bağlantı reddedildiğinde, kanal olayları etkinleştirildiyse ve kuyruk yöneticisi çalışıyorsa, neden niteleyicisi MQRQ_CHANNEL_BLOCKED_NOACCESS olan bir olay iletisi yayınlanır.

Bir örnek için bkz. “Belirli IP adreslerini engelleme” sayfa 398 .

Engelleyici kullanıcı kimlikleri

Belirli kullanıcı kimliklerinin bir istemci kanalı üzerinden bağlanmasını önlemek için, BLOCKUSER tipinde bir kanal kimlik doğrulama kaydı ayarlayın. Bu tip kanal kimlik doğrulama kaydı, ileti kanallarına değil, yalnızca istemci kanallarına uygulanır. Engellenecek bir ya da daha çok kullanıcı kimliği belirtebilirsiniz, ancak genel arama karakterleri kullanamazsınız.

Bu nedenle gelen bağlantı reddedildiğinde, kanal olaylarının etkinleştirildiği durumlarda, neden niteleyicisi MQRQ_CHANNEL_BLOCKED_USERID olan bir olay iletisi yayınlanır.

Bir örnek için bkz. “Belirli kullanıcı kimliklerini engelleme” sayfa 400 .

Belirli kanallarda belirtilen kullanıcı kimliklerine ilişkin erişimi engellemek için, USERSRC (NOACCESS) parametresiyle USERMAP tipinde bir kanal kimlik doğrulama kaydı da ayarlayabilir.

Bu nedenle gelen bağlantı reddedildiğinde, kanal olayları etkinleştirildiyse ve kuyruk yöneticisi çalışıyorsa, neden niteleyicisi MQRQ_CHANNEL_BLOCKED_NOACCESS olan bir olay iletisi yayınlanır.

Bir örnek için bkz. “İstemci kullanıcı kimliği için erişimin engellenmesi” sayfa 403 .

Engelleyici kuyruk yöneticisi adları

Belirtilen bir kuyruk yöneticisinden bağlantı kuran herhangi bir kanalın erişimi olmadığını belirtmek için, USERSRC (NOACCESS) parametresiyle QMGRMAP tipinde bir kanal kimlik doğrulama kaydı ayarlayın. Tek bir kuyruk yöneticisi adı ya da joker karakter içeren bir kalıp belirtebilirsiniz. Kuyruk yöneticilerinden erişimi engellemek için BLOCKUSER işlevinin eşdeğeri yoktur.

Bu nedenle gelen bağlantı reddedildiğinde, kanal olayları etkinleştirildiyse ve kuyruk yöneticisi çalışıyorsa, neden niteleyicisi MQRQ_CHANNEL_BLOCKED_NOACCESS olan bir olay iletisi yayınlanır.

Bir örnek için bkz. “Uzak kuyruk yöneticisinden erişimin engellenmesi” sayfa 402 .

SSL ya da TLS DN ' leri engelleme

Belirtilen bir DN içeren bir SSL ya da TLS kişisel sertifikası sunan herhangi bir kullanıcının erişimi olmadığını belirtmek için, USERSRC (NOACCESS) parametresiyle SSLPEERMAP tipinde bir kanal kimlik doğrulama kaydı ayarlayın. Genel arama karakterleri de içinde olmak üzere tek bir ayırt edici ad ya da bir kalıp belirtebilirsiniz. DN ' lere erişimi engellemek için BLOCKUSER işlevinin eşdeğeri yoktur.

Bu nedenle gelen bağlantı reddedildiğinde, kanal olayları etkinleştirildiyse ve kuyruk yöneticisi çalışıyorsa, neden niteleyicisi MQRQ_CHANNEL_BLOCKED_NOACCESS olan bir olay iletisi yayınlanır.

Bir örnek için bkz. “SSL ya da TLS Ayırt Edici Adı için erişimi engelleme” sayfa 403 .

Kullanılacak IP adreslerinin kullanıcı kimlikleriyle eşlenmesi

Belirtilen bir IP adresinden bağlanan herhangi bir kanalın belirli bir MCAUSER ' i kullanacağı belirtmek için, ADDRESSMAP tipinde bir kanal kimlik doğrulama kaydı ayarlayın. Tek bir adres, adres aralığı ya da joker karakter içeren bir kalıp belirleyebilirsiniz için mı var?

Bir kapı ileticisi, DMZ oturumu kesmesi ya da kuyruk yöneticisine sunulan IP adresini değiştiren başka bir ayar kullanıyorsanız, eşleme IP adresleri kullanmanız için uygun olmayabilir.

Bir örnek için bkz. “Bir IP adresinin MCAUSER kullanıcı kimliğiyle eşlenmesi” sayfa 404 .

Kuyruk yöneticisi adlarının kullanılacak kullanıcı kimlikleriyle eşlenmesi

Belirtilen bir kuyruk yöneticisinden bağlantı kuran herhangi bir kanalın belirli bir MCAUSER ' i kullanacağı belirtmek için, QMGRMAP tipinde bir kanal kimlik doğrulama kaydı ayarlayın. Tek bir kuyruk yöneticisi adı ya da joker karakter içeren bir kalıp belirtebilirsiniz.

Bir örnek için bkz. “Uzak kuyruk yöneticisinin MCAUSER kullanıcı kimliğiyle eşlenmesi” sayfa 400 .

Bir istemci tarafından bildirilen kullanıcı kimliklerinin kullanılacak kullanıcı kimlikleriyle eşlenmesi

Belirli bir kullanıcı kimliği IBM MQ MQI istemcisinden gelen bir bağlantı tarafından kullanılıyorsa, farklı, belirtilen bir MCAUSER kullanılacaksa, USERMAP tipinde bir kanal kimlik doğrulama kaydı ayarlayın. Kullanıcı kimliği eşlemesi joker karakter kullanmaz.

Bir örnek için bkz. "Bir istemci kullanıcı kimliğinin MCAUSER kullanıcı kimliğiyle eşlenmesi" sayfa 401 .

SSL ya da TLS DN ' lerini kullanılacak kullanıcı kimlikleriyle eşleme

Belirli bir DN içeren bir SSL/TLS kişisel sertifikası sunan herhangi bir kullanıcının belirli bir MCAUSER ' i kullanacağını belirtmek için, SSLPEERMAP tipinde bir kanal kimlik doğrulama kaydı ayarlayın. Genel arama karakterleri de içinde olmak üzere tek bir ayırt edici ad ya da bir kalıp belirtebilirsiniz.

Bir örnek için bkz. "Bir SSL ya da TLS Ayırt Edici Adının MCAUSER Kullanıcı Kimliğiyle Eşlenmesi" sayfa 402 .

Kuyruk yöneticilerini, istemcileri ya da SSL ya da TLS DN ' lerini IP adresine göre eşleme

Bazı durumlarda, üçüncü bir tarafın bir kuyruk yöneticisi adını sahtecisi olması mümkün olabilir. SSL ya da TLS sertifikası ya da anahtar veritabanı dosyası da çalınabilir ve yeniden kullanılabilir. Bu tehditlere karşı korumak için, belirli bir kuyruk yöneticisinden ya da istemciden gelen bir bağlantının ya da belirli bir DN ' nin belirli bir IP adresinden bağlanması gerektiğini belirtebilirsiniz. USERMAP, QMGRMAP ya da SSLPEERMAP tipinde bir kanal kimlik doğrulama kaydı ayarlayın ve ADDRESS parametresini kullanarak izin verilen IP adresini ya da IP adresleri kalıbını belirtin.

Bir örnek için bkz. "Uzak kuyruk yöneticisinin MCAUSER kullanıcı kimliğiyle eşlenmesi" sayfa 400 .

Kanal kimlik doğrulama kayıtları arasındaki etkileşim

Bir kanal bağlantı kurmaya çalışırken birden fazla kanal kimlik doğrulama kaydıyla eşleşiyor olabilir ve bunların çelişkili etkileri olabilir. Örneğin, bir kanal, BLOCKUSER kanal kimlik doğrulama kaydı tarafından engellenen, ancak farklı bir kullanıcı kimliği ayarlayan bir SSLPEERMAP kaydıyla eşleşen bir SSL ya da TLS sertifikasıyla bir kullanıcı kimliğini doğrulayabilir. Ayrıca, kanal kimlik doğrulama kayıtları joker karakter kullanıyorsa, tek bir IP adresi, kuyruk yöneticisi adı ya da SSL ya da TLS DN birden çok kalıpla eşleşebilir. Örneğin, 192.0.2.6 IP adresi 192.0.2.0-24, 192.0.2. *, ve 192.0. * .6. Yapılan işlem aşağıdaki gibi belirlenir.

- Kullanılan kanal kimlik doğrulama kaydı aşağıdaki gibi seçilir:
  - Kanal adıyla açıkça eşleşen bir kanal kimlik doğrulama kaydı, genel arama karakteri kullanarak kanal adıyla eşleşen bir kanal kimlik doğrulama kaydından önceliklidir.
  - SSL ya da TLS DN kullanan bir kanal kimlik doğrulama kaydı, kullanıcı kimliği, kuyruk yöneticisi adı ya da IP adresi kullanan bir kayıttan önceliklidir.
  - Kullanıcı kimliği ya da kuyruk yöneticisi adı kullanan bir kanal kimlik doğrulama kaydı, IP adresi kullanan bir kayıttan önceliklidir.
- Eşleşen bir kanal kimlik doğrulama kaydı bulunursa ve bir MCAUSER belirtirse, bu MCAUSER kanala atanır.
- Eşleşen bir kanal kimlik doğrulama kaydı bulunursa ve kanalın erişimi olmadığını belirtirse, kanala *NOACCESS MCAUSER değeri atanır. Bu değer daha sonra bir güvenlik çıkış programı tarafından değiştirilebilir.
- Eşleşen bir kanal kimlik doğrulama kaydı bulunamazsa ya da eşleşen bir kanal kimlik doğrulama kaydı bulunursa ve kanalın kullanıcı kimliğinin kullanılacağını belirtirse, MCAUSER alanı incelenir.
  - MCAUSER alanı boşsa, istemci kullanıcı kimliği kanala atanır.
  - MCAUSER alanı boş değilse, kanala atanır.
- Güvenlik çıkış programı çalıştırılır. Bu çıkış programı kanal kullanıcı kimliğini ayarlayabilir ya da erişimin engelleneceğini belirleyebilir.

- Bağlantı engellenirse ya da MCAUSER *NOACCESS olarak ayarlanırsa, kanal sona erer.
- Bağlantı engellenmezse, istemci kanalı dışında herhangi bir kanal için, önceki adımlarda belirlenen kanal kullanıcı kimliği, engellenen kullanıcılar listesine göre denetlenir.
  - Kullanıcı kimliği engellenen kullanıcılar listesinde yer alıyorsa, kanal sona erer.
  - Kullanıcı kimliği engellenen kullanıcılar listesinde yoksa, kanal çalışır.

Kanal kimlik doğrulama kayıtlarının sayısı, bir kanal adı, IP adresi, anasistem adı, kuyruk yöneticisi adı ya da SSL ya da TLS ile eşleştiği durumlarda, en özel eşleşme kullanılır. Eşleşbilmek (Not ı gibi gibi):

- En çok genel arama karakteri içermeyen bir ad kullanılır; örneğin:
  - A.B.C
  - 192.0.2.6 IP adresi
  - `hursley.ibm.com` anasistem adı
  - 192.0.2.6 kuyruk yöneticisi adı
- En soysal olan, aşağıdaki gibi eşleşen tek bir yıldız işaretidir:
  - Tüm kanal adları
  - Tüm IP adresleri
  - Tüm anasistem adları
  - Tüm kuyruk yöneticisi adları
- Bir dizginin başında yıldız işareti olan bir kalıp, bir dizginin başında tanımlı bir değerden daha soysaldır:
  - Kanallar için, *.B.C , A ' dan daha genel. *
  - IP adresleri için, *.0.2.6 192 'den daha soysaldır. *
  - Anasistem adları için `*.ibm.com` , `hursley.*` değerinden daha soysal
  - Kuyruk yöneticisi adları için, *QUEUEMANAGER değeri QUEUEMANAGER* değerinden daha soysal
- Bir dizginin belirli bir yerinde yıldız işareti olan bir kalıp, bir dizginin aynı yerinde tanımlı bir değerden daha soysaldır ve benzer şekilde, bir dizginin sonraki her yeri için de geçerlidir:
  - Kanallar için A. * .C, A.B.*
  - IP adresleri için 192. *.2.6 , 192.0. * değerinden daha soysaldır.
  - Anasistem adları için `hursley.*.com` , `hursley.ibm.*` değerinden daha soysal
  - Kuyruk yöneticisi adları için, Q* MANAGER, QUEUE* ' den daha soysal
- İki ya da daha çok örüntüde bir dizginin belirli bir yerinde yıldız işareti varsa, yıldız işaretini izleyen daha az düğüm sayısı daha soysaldır:
  - Kanallar için, A. * A * .C ' den daha soysal
  - IP adresleri için, 192. * 192. * .2. * değerinden daha soymıştım de ve ve
  - Anasistem adları için `hurlsey.*` , `hursley.*.com` değerinden daha soysal
  - Kuyruk yöneticisi adları için Q*, Q* MGR ' den daha soysal
- Ayrıca, bir IP adresi için:
  - Tire (-) ile gösterilen bir aralık, yıldız işaretinden daha belirlidir. Bu nedenle 192.0.2.0-24 , 192.0.2. * ' den daha özgüdür.
  - Başka bir alt küme olan bir aralık, daha büyük aralıktan daha özeldir. Bu nedenle 192.0.2.5-15 , 192.0.2.0-24' ten daha özeldir.
  - Çakışan aralıklara izin verilmez. Örneğin, 192.0.2.0-15 ve 192.0.2.10-20için kanal kimlik doğrulama kayıtlarına sahip olamazsınız.
  - Örüntü, tek bir sondaki yıldız işaretiyle bitmedikçe, bir örüntünün istenen parça sayısından daha az sayıda olamaz. Örneğin 192.0.2 geçersiz, ancak 192.0.2.* geçerli.

– Sondaki yıldız işareti, adresin geri kalan kısmından uygun parça ayırıcısıyla (IPv4için nokta (.), IPv6için iki nokta (:) ayrılmalıdır. Örneğin, 192.0* geçerli değildir; yıldız işareti kendi içinde değildir.
– Sondaki yıldız işaretinin yanında yıldız işareti olmaması koşuluyla, bir kalıp ek yıldız işaretleri içerebilir. Örneğin, 192. * .2. * geçerli, ancak 192.0. * .* (çizelge adı) geçersiz.
– IPv6 adres kalıbı çift iki nokta üst üste ve sondaki yıldız işareti içeremez; sonuçta elde edilen adres belirsiz olur. Örneğin, 2001:: *, 2 0 0 1: 0 0 0: *, 2001:0000:0000: * ve bu şekilde genişletilebilir

- Bir SSL ya da TLS Ayırt Edici Adı (DN) için, alt dizgilerin öncelik sırası aşağıdaki gibidir:

Çizelge 7. Alt dizgilerin öncelik sırası

Sıralama	**DN alt dizgisi**	**Ad**
1	SERI NUMARASı=	Sertifika seri numarası
2	MAIL=	E-posta adresi
3	Deprecated E=	E-posta adresi (MAIL tercihine göre kullanımdan kaldırıldı)
4	UID=, USERID=	Kullanıcı kimliği
5	CN=	Ortak ad
6	T =	Başlık
7	OU=	Kuruluş Birimi
8	DC=	Etki alanı bileşeni
9	O=	Kuruluş
10	SOKAK =	Sokak lbay lması hakkında ama ilk satırı-adres
11	L=	İlçe
12	ST =, SP=, S=	Eyalet ya da bölge adı
13	PC=	Posta kodu/posta kodu
14	C=	Ülke
15.000	YAPıLANDıRıLMAMış AD	Anasistem adı
16	YAPıLANDıRMA DAMDARı=	IP adresi
17	DNQ=	Ayırt edici ad niteleyicisi

Bu nedenle, bir SSL ya da TLS sertifikası O=IBM ve C=UK alt dizgilerini içeren bir DN ile sunulursa, IBM MQ , her ikisi de varsa, C=UK için bir yerine O=IBM için bir kanal kimlik doğrulama kaydı kullanır.

Bir DN, büyük kuruluş birimleri önce belirtilen sıradüzenli olarak belirtilmesi gereken birden çok kuruluş birimi içerebilir. İki DN, kuruluş birimi değerleri dışında her bakımdan eşitse, daha spesifik DN aşağıdaki gibi belirlenir:

1. Farklı sayıda kuruluş birimi özniteliklerine sahiplerse, en çok kuruluş birimi değerine sahip DN daha belirdıktan bunu masını masını sağlar. Bunun nedeni, daha fazla Kuruluş Birimine sahip DN 'in DN' yi daha ayrıntılı olarak niteleyip daha fazla eşleşme ölçütü sağlamış olmasıdır. Üst düzey kuruluş birimi bir genel arama karakteri (OU = *) olsa da, daha fazla kuruluş birimine sahip DN genel olarak daha özel kabul edilir.
2. Aynı sayıda Kuruluş Birimi özniteliğine sahiplerse, karşılık gelen Kuruluş Birimi değerleri çiftleri soldan sağa doğru sırayla karşılaştırılır; burada en soldaki Kuruluş Birimi, aşağıdaki kurallara göre en yüksek düzeydir (en az belirli ... ya ...).

a. Genel arama karakteri değeri olmayan bir kuruluş birimi en özelidir; yalnızca tek bir dizgiyle eşleşebilir.
b. Başında ya da sonunda tek bir genel arama karakteri bulunan bir kuruluş birimi (örneğin, OU=ABC* ya da OU= *ABC) en özel seçenekyoruz " bu updemek.
c. İki genel arama karakteri içeren bir kuruluş birimi (OU= *ABC*) sonraki sonra sonra iyi Ayırt Edici (OU= *ABC*) olur.
d. Yalnızca yıldız işaretinden (OU = *) oluşan bir kuruluş birimi en az özeldir.

3. Dizgi karşılaştırması aynı belirliğe sahip iki öznitelik değeri arasında bağlıysa, hangi öznitelik dizgisi daha uzunsa, o değer daha belirgindir.
4. Dizgi karşılaştırması aynı özgüllük ve uzunluğa sahip iki öznitelik değeri arasında bağlıysa, sonuç, DN ' nin herhangi bir genel arama karakteri dışında olan kısmının büyük ve küçük harfe duyarsız bir dizgi karşılaştırmasıyla saptanır.

İki DN, DC değerleri dışında her bakımdan eşitse, DC değerlerinde en soldaki DC 'nin en düşük düzey (en spesifik) olması ve karşılaştırma sıralamasının buna göre farklılık göstermesi dışında, aynı eşleşen kurallar OU' lar için de geçerlidir.

Kanal kimlik doğrulama kayıtlarının görüntülenmesi

Kanal kimlik doğrulama kayıtlarını görüntülemek için **DISPLAY CHLAUTH** ya da PCF komutunu **Inquire Channel Authentication Records**kullanın. Sağlanan kanal adıyla eşleşen tüm kayıtları döndürmeyi seçebilir ya da açık bir eşleşme seçebilirsiniz. Belirtik eşleşme, bir kanal belirli bir kuyruk yöneticisinden ya da belirli bir kullanıcı kimliğinden bağlantı kurmaya çalıştıysa ve isteğe bağlı olarak, belirli bir DN içeren bir SSL/TLS kişisel sertifikası sunuyorsa, hangi kanal kimlik doğrulama kaydının kullanılacağını belirtir.

İlgili kavramlar

“Uzak ileti sistemi güvenliği” sayfa 99
Bu bölümde, güvenliğin uzak ileti sistemi özellikleri ele alınmıştır.

CHLAUTH ve CONNAUTH Etkileşimi

Kanal kimlik doğrulama kayıtları (CHLAUTH) ve bağlantı kimlik doğrulaması (CONNAUTH), bir kanaldaki tek bir etkileşim durumunda IBM MQiçinde nasıl etkileşimde bulunur.

Farklı bağ tanımı tipleri

IBM MQ , bir uygulamanın bağlanması için iki yöntemi destekler:

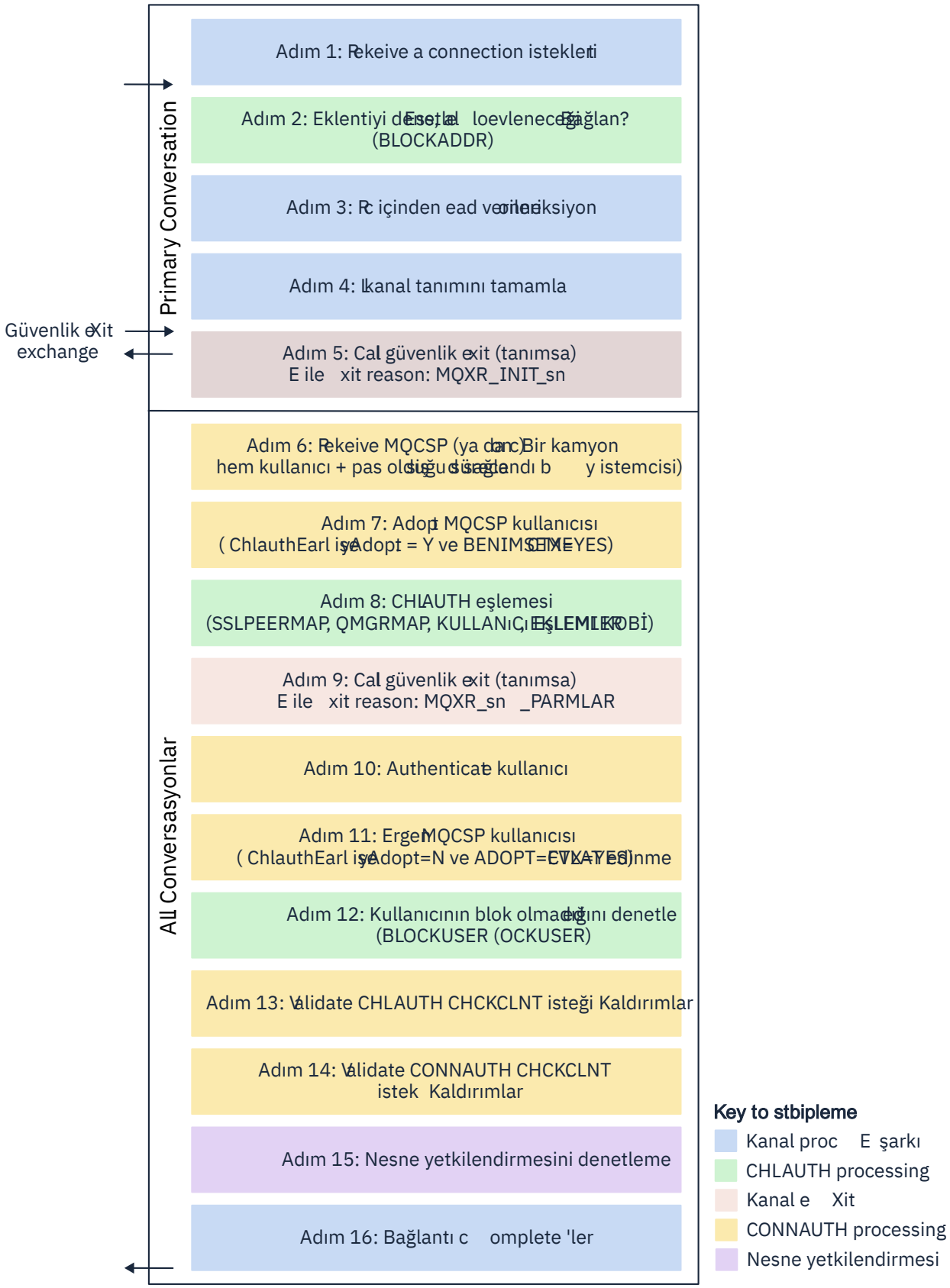
Yerel bağ tanımları
Uygulama ve kuyruk yöneticisi aynı işletim görüntüsünde olduğunda geçerlidir. CHLAUTH, bu uygulama bağlantısı tipiyle ilgili değil.

İstemci bağ tanımları
Uygulama ve kuyruk yöneticisi iletişim kurmak için ağı kullandığında geçerlidir. Uygulama ve kuyruk yöneticisi aynı makinede ya da farklı makinelerde çalışabilir. IBM MQiçinde, bir istemci bağlantısı (SVRCONN) kanalı biçiminde işlenir ve bu durumda hem CONNAUTH hem de CHLAUTH geçerlidir.

Bir kanalın alıcı ucunun bağlama adımları

Bir uygulama bir kuyruk yöneticisine bağlandığında, kanalın her iki ucunun da diğer ucunun nelerin desteklendiğini anladığından emin olmak için önemli miktarda denetim gerçekleştirilir. Kanalın giriş ucu, istemcinin bağlanmasına izin verildiğinden emin olmak için CHLAUTH ve CONNAUTH dahil olmak üzere ek bir denetim yapar ve bu işlem sonucu etkileyebileceği için bir güvenlik çıkışı da içerebilir. Bu kanal bağlantı aşamasına *bağ tanımlama aşaması*da denir.

Aşağıdaki çizge, sunucu sona erdiğinde (kuyruk yöneticisinde) SVRCONN kanalının geçeceği adımları listeler:



Adım 1: Bağlantı isteği alınması
Kanal başlatıcı ya da dinleyici, ağ üzerinde bir yerden bağlantı isteği alır.

Adım 2: Adresin bağlanmasına izin veriliyor mu?
Herhangi bir veri okunmadan önce IBM MQ , adresin *BLOCKADDR* kuralında olup olmadığını görmek için ortağın IP adresini CHLAUTH kurallarına göre denetler. Adres bulunamazsa ve engellenmezse, akış sonraki adıma ilerler.

Adım 3: Kanaldan verileri okuyun
IBM MQ şimdi verileri bir arabelleğe okur ve gönderilen bilgileri işlemeye başlar.

Adım 4: Kanal tanımını arayın
İlk veri akışında IBM MQ , diğer şeylerin yanı sıra, gönderme ucunun başlatmaya çalıştığı kanalın adını gönderir. Alıcı kuyruk yöneticisi, kanal için belirtilen tüm ayarlara sahip kanal tanımını arayabilir.

Adım 5: Güvenlik çıkışını çağır (tanımlandıysa)
Kanalda tanımlı bir güvenlik çıkışı (SCYEXIT) varsa, bu, çıkış nedeniyle (MQCXP.**ExitReason**) çağrılır. MQXR_INIT_SEC olarak ayarlayın.

Adım 6: MQCSP ' yi Al
Gerekirse, istemci tarafından sağlanan kimlik doğrulama kimlik bilgilerini oluşturun.

İstemci uyumluluk kipinde çalışan bir Java ya da JMS uygulamasıysa, kuyruk yöneticisine MQCSP yapısı geçirmez. Bunun yerine, uygulama bir kullanıcı kimliği ve parola sağladıysa, burada bir MQCSP yapısı oluşturulur.

Adım 7: MQCSP kullanıcısını benimseyin (`ChlauthEarlyAdopt` *Y* ve ADOPTCTX=YES ise)
İstemci tarafından sağlanan kimlik bilgilerinin kimliği doğrulanır.

CONNAUTH, bildirilir bir ayırt edici adı kısa bir kullanıcı kimliğiyle eşlemek için LDAP kullanıyorsa, eşleme bu adımda gerçekleşir.

Kimlik doğrulaması başarılı olursa, kullanıcı kimliği kanal tarafından kullanılır ve CHLAUTH eşleme adımı tarafından kullanılır.

Not: IBM MQ 9.0.4 ' den **`ChlauthEarlyAdopt=`** *Y* parametresi, yeni kuyruk yöneticileri için qm.ini dosyasının kanal kısmına otomatik olarak eklenir.

Adım 8: CHLAUTH eşlemesi
CHLAUTH önbelleği, *SSLPEERMAP*, *USERMAP*, *QMGRMAP*ve *ADDRESSMAP*eşleme kurallarını aramak için yeniden incelenir.

Gelen kanalda en özel olarak eşleşen kural kullanılır. Kuralda **USERSRC***(CHANNEL)* ya da *(MAP)*varsa, kanal bağlama üzerinde devam eder.

CHLAUTH kuralları **USERSRC***(NOACCESS)*içeren bir kuralı değerlendirirse, kimlik bilgileri daha sonra 9numaralı adımda geçerli bir kimlik bilgileriyle geçersiz kılınmadıkça, uygulamanın kanala bağlanması engellenir.

Adım 9: Güvenlik çıkışını çağır (tanımlandıysa)
Kanalda tanımlı bir güvenlik çıkışı (SCYEXIT) varsa, bu, çıkış nedeniyle (MQCXP.**ExitReason**) çağrılır. MQXR_SEC_PARMS olarak ayarlayın.

MQCXP yapısının **SecurityParms** alanında MQCSP göstergesi bulunur.

MQCSP yapısının kullanıcı kimliğine ilişkin göstergeleri var (MQCSP.**CSPUserIdPtr**) ve parola (MQCSP.**CSPPasswordPtr**). V 9.3.4 IBM MQ 9.3.4' den MQCSP yapısı, kimlik doğrulama simgesine (MQCSP.**TokenPtr**) ilişkin bir gösterge de içerir.

Çıkışta kullanıcı kimliği ve parolave kimlik doğrulama simgesi değiştirilir. Aşağıdaki örnekte, bir güvenlik çıkışının kullanıcı kimliği ve parola değerlerini denetim günlüğüne nasıl yazdıracağı gösterilmektedir:

```
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);
```

Çıkış, MQCXP 'de *MQXCC_CLOSE_CHANNEL* döndürülerek IBM MQ ' a kanalı kapatmasını söyleyebilir.**Exitresponse** alanı. Tersi durumda, kanal işleme bağlantı doğrulama aşamasına devam eder.

Not: Bildirilen kullanıcı güvenlik çıkışı tarafından değiştirilirse, CHLAUTH eşleme kuralları yeni kullanıcıya yeniden uygulanmaz.

Adım 10: Kullanıcının kimliğini doğrula

Kuyruk yöneticisinde CONNAUTH etkinleştirilirse kimlik doğrulama aşaması oluşur.

Bunu denetlemek için 'DISPLAY QMGR CONNAUTH' MQSC komutunu verin.

z/OS Aşağıdaki örnek, IBM MQ for z/OSüzerinde çalışan bir kuyruk yöneticisinden **DISPLAY QMGR CONNAUTH** komutunun çıkışını göstermektedir.

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

Multi Aşağıdaki örnek, IBM MQ for Multiplatformsüzerinde çalışan bir kuyruk yöneticisinden '**DISPLAY QMGR CONNAUTH**' komutunun çıkışını göstermektedir.

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

CONNAUTH değeri, bir **AUTHINFO** IBM MQ nesnesinin adıdır.

İşletim sistemi kimlik doğrulaması (**AUTHTYPE**(*IDPWOS*)) IBM MQ for Multiplatforms ve IBM MQ for z/OSsistemlerinde geçerli olduğundan, örnekler işletim sistemi kimlik doğrulamasını kullanır.

z/OS Aşağıdaki örnekte, IBM MQ for z/OSüzerinde çalışan bir kuyruk yöneticisinden **AUTHTYPE**(*IDPWOS*) ile birlikte varsayılan AUTHINFO nesnesi gösterilmektedir.

```
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 Aşağıdaki örnekte, IBM MQ for Multiplatformsüzerinde çalışan bir kuyruk yöneticisinden **AUTHTYPE**(*IDPWOS*) ile birlikte varsayılan AUTHINFO nesnesi gösterilmektedir.

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

AUTHINFO TYPE (IDPWOS) nesnesi, CHCKCLNTadlı bir özniteliğe sahiptir. Değer *REQUIRED* olarak değiştirilirse, tüm istemci uygulamalarının geçerli kimlik bilgileri sağlaması gerekir.

Kullanıcının kimliği Adım 7' de doğrulandıysa, aşağıda belirtilenler dışında başka bir kimlik doğrulama denetimi gerçekleştirilmez:

- MQCXP yapısının SecurityParms alanındaki kullanıcı kimliği, parola ya da kimlik doğrulama simgesi Adım 9' da bir güvenlik çıkışıyla değiştirildi.
- İstemci uygulaması, yeniden bağlanabilir işlevsellik isteyen seçeneklerle bağlantı kurdu.

Adım 11: MQCSP kullanıcısının bağlamını benimseyin (`ChlauthEarlyAdopt=`*N* ve ADOPTCTX=YES ise)

Kanalın MCAUSER altında mı, yoksa uygulamanın sağladığı kullanıcı kimliği altında mı çalıştığını denetleyen ADOPTCTX özniteliğini ayarlayabilirsiniz.

MQCSP ya da MQXCP yapısının **`SecurityParms`** alanında bildirilen kullanıcı kimliğinin kimliği başarıyla doğrulandıysa ve **ADOPTCTX** *EVET*.ise, 7 ve 8 adımlarından kaynaklanan kullanıcı bağlamı, MQCXP yapısının **`SecurityParms`** alanındaki kullanıcı kimliği, parola ya da kimlik doğrulama simgesi , 9adımındaki bir güvenlik çıkışıyla değiştirilmediyse, bu uygulama için kullanılacak bağlam olarak benimsenir.

Bu bildiri kullanıcı kimliği, IBM MQ kaynaklarını kullanma yetkisi için denetlenen kullanıcı kimliğidir.

Örneğin, SVRCONN kanalında ayarlanmış bir MCAUSER 'iniz yok ve istemciniz Linux makinenizde'`johndoe`' altında çalışıyor. Uygulamanız MQCSP 'de'`fred`'kullanıcısını belirtiyor, bu nedenle kanal etkin MCAUSER olarak'`johndoe`' ile çalışmaya başlıyor. CONNAUTH denetiminden sonra, kullanıcı '`fred`' benimsenir ve kanal etkin MCAUSER olarak '`fred`' ile çalışır.

Adım 12: Kullanıcının engellenmediğini denetleyin (BLOCKUSER)

CONNAUTH denetimi başarılı olursa, etkin MCAUSER ' in bir *BLOCKUSER* kuralı tarafından engellenip engellenmediğini denetlemek için CHLAUTH önbelleği yeniden incelenir. Kullanıcı engellenirse, kanal sona erer.

Adım 13: CHLAUTH CHCKCLNT gereksinimlerinin geçerliliğini denetle

8 . adımda seçilen CHLAUTH kuralı ayrıca REQUIRED ya da REQDADM için bir CHCKCLNT değeri belirtiyorsa, gereksinimi karşılamak için geçerli bir CONNAUTH kullanıcı kimliğinin sağlandığından emin olmak için doğrulama gerçekleştirilir.

- CHCKCLNT (GEREKLİ) değeri belirlendiyse, 7 ya da 10numaralı adımlarda bir kullanıcının kimliği doğrulanmış olmalıdır. Tersi durumda, bağlantı reddedilir.
- CHCKCLNT (REQDADM) ayarlanırsa, bu bağlantının ayrıcalıklı olduğu belirlenirse, 7 ya da 10 adımlarında bir kullanıcının kimliği doğrulanmış olmalıdır. Tersi durumda, bağlantı reddedilir.
- CHCKCLNT (ASQMGR) ayarlanırsa, bu adım atlanır.

Notlar:

1. CHCKCLNT (REQUIRED) ya da CHCKCLNT (REQDADM) ayarlandıysa, ancak kuyruk yöneticisinde CONNAUTH etkinleştirilmediyse, yapılandırmadaki çakışma nedeniyle bağlantı bir MQRC_SECURITY_ERROR (2063) dönüş koduyla başarısız olur.
2. Bu adımda kullanıcının kimliği yeniden doğrulanmadı.

Adım 14: CONNAUTH CHCKCLNT gereksinimlerini doğrulayın.

Kuyruk yöneticisinde CONNAUTH etkinleştirilirse kimlik doğrulama aşaması oluşur.

Gelen bağlantılar için hangi gereksinimlerin ayarlandığını belirlemek üzere CONNAUTH CHCKCLNT değeri denetlenir:

- CHCKCLNT (NONE) ayarlanırsa bu adım atlanır
- CHCKCLNT (İSTEğE BAğLı) ayarlıysa, bu adım atlanır.
- CHCKCLNT (REQUIRED) ayarlanırsa, 7 ya da 10adımında bir kullanıcının kimliği doğrulanmış olmalıdır. Tersi durumda, bağlantı reddedilir.
- CHCKCLNT (REQDADM) ayarlanırsa, bu bağlantının ayrıcalıklı olduğu belirlenirse, 7 ya da 10 adımlarında bir kullanıcının kimliği doğrulanmış olmalıdır. Tersi durumda, bağlantı reddedilir.

Not: Bu adımda kullanıcının kimliği yeniden doğrulanmadı.

Multi **Adım 15: Nesne yetkilendirmesini denetleme**
Etkin MCAUSER ' in kuyruk yöneticisine bağlanmak için uygun yetkiye sahip olduğundan emin olmak için bir denetim yapılır.

ALW Ek bilgi için Nesne Yetkilisi Yöneticisikonusuna bakın.

IBM i Daha fazla bilgi için bkz. “IBM i üzerinde nesne yetkisi yöneticisi” sayfa 155.

Adım 16: Bağlantı tamamlanır
Önceki adımlar başarıyla tamamlanırsa, bağlantı tamamlanır.

İlgili kavramlar
CONNAUTH
Bir kuyruk yöneticisi, bağlandığında bir uygulama tarafından sağlanan kimlik bilgilerini doğrulayacak şekilde yapılandırılabilir.

İlgili başvurular
CHLAUTH AYARLA
AUTHINFO DEğIşTIR

CHLAUTH erişim sorunlarını çözme
Kanal kimlik doğrulama kayıtlarını (CHLAUTH) kullanırken belirli erişim sorunlarını çözmek için adımlar ve örnekler.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. MQSC komutlarını kullanarak IBM MQ yönetme.

Bu görev hakkında

CHLAUTH işlemesi için üç varsayılan kural vardır:

- `MQ-admin*` kullanıcıları tarafından tüm kanallara ERIşIM YOK
- TüM SISTEMLERE ERIşIM YOK. * tüm kullanıcılar tarafından kanallar
- SYSTEM.ADMIN.SVRCONN kanalı (`MQ-admin` dışı kullanıcılar)

İlk iki kural tüm kanallara erişimi engeller. Üçüncü kural daha spesifiktir ve bu nedenle kanal SYSTEM.ADMIN.SVRCONN kanalı, bu kanalda erişime izin verir.

CHLAUTH kuralları, bir kanalın başlatılıp başlatılamayacağını belirlemek için kullanılır ve MCAUSER aracılığıyla başka bir kullanıcı kimliğiyle eşlemeye izin verir. Kanal başlatılamazsa, genellikle aşağıdaki hatalar oluşur:

- `RC 2035 MQRC_NOT_AUTHORIZED`
- `RC 2059 MQRC_Q_MGR_NOT_KULLANILAMIYOR`
- `AMQ4036 Erişimi yok`
- `AMQ9776: Kanal kullanıcı kimliği tarafından engellendi`
- `AMQ9777: Kanal engellendi`
- `MQJE001: Bir MQException oluştu: Tamamlanma Kodu 2, Neden 2035`
- `MQJE036: Kuyruk yöneticisi bağlantı girişimini reddetti`

Erişimi kesin olarak engellemeli ve kanallara kimlerin erişebileceğini ve kanalları başlatabileceğini denetlemek için daha fazla CHLAUTH kuralı eklemelisiniz.

Geçici bir önlem olarak ve listelenen hataları gidermek için aşağıdaki adımlardan herhangi birini tamamlayın.

Yordam

- **CHLAUTH kurallarını devre dışı bırak**

  Geçici bir önlem olarak ve yukarıdaki hataları gidermek için CHLAUTH kurallarını devre dışı bırakabilirsiniz. Kurallar herhangi bir zamanda yeniden etkinleştirilebilir ve CHLAUTH kurallarının devre dışı bırakılması bağlantı sorununu çözerse, bunun nedeni olduğunu bilirsiniz.

  CHLAUTH kurallarını devre dışı bırakmak için aşağıdaki MQSC komutunu çalıştırın:

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

  CHLAUTH ' yi *WARN*olarak ayarlayabileceğinizi unutmayın; bu, kuralın erişimine izin verir ve bunun sonucunu günlüğe kaydeder.

- **CHLAUTH kurallarını değiştirme ya da kaldırma**

  CHLAUTH kuralını ya da kurallarını silebilir ya da değiştirebilirsiniz; bu da sorununuza neden olur.

  Bir CHLAUTH kuralını değiştirmek için, SET CHLAUTH komutunu ACTION (REPLACE) ile kullanın. Örneğin, engellenmek yerine, herhangi bir `MQ-admin` kullanıcısının UYARI için tüm kanallara erişmesine neden olmayan varsayılan kuralı değiştirmek için aşağıdaki MQSC komutunu çalıştırın:

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

  Bir CHLAUTH kuralını silmek için, SET CHLAUTH komutunu ACTION (REMOVE) ile birlikte kullanırsınız. Örneğin, herhangi bir `MQ-admin` kullanıcısının tüm kanallara erişmesine neden olmayan varsayılan kuralı silmek için aşağıdaki MQSC komutunu çalıştırın:

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

- **MATCH kullanarak erişimi test edin (RUNCHECK)**

  CHLAUTH kurallarınızın sonucunu, CHLAUTH kuralının MATCH (*RUNCHECK*) seçeneğini kullanarak sınayabilirsiniz. **MATCH** (*RUNCHECK*) seçeneği, bu kanal bu kuyruk yöneticisine bağlanıyorsa, yürütme sırasında belirli bir gelen kanal tarafından eşleştirilen kaydı döndürür. Aşağıdakileri sağlamanız gerekir:

  – Kanal adı
  – ADDRESS özniteliği
  – SSLPEER özniteliği, yalnızca gelen kanal SSL ya da TLS kullanıyorsa
  – QMNAME, gelen kanal bir kuyruk yöneticisi kanalıysa, ya da
  – CLNTUSER özniteliği, gelen kanal bir istemci kanalıysa

  Aşağıdaki örnek, hangi CHLAUTH kuralının yerinde olduğunu denetlemek için bir MQSC komutu çalıştırır; varsayılan kurallar, `CHAN1`adlı bir kanala erişen bir `MQ-admin` kullanıcı `johndoe` ile sonuçlanır:

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

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

  `johndoe`kullanıcısı için kanal çalışmaz, *MQADMIN kullanıcıları için BLOCKUSER kuralı nedeniyle kullanıcı engellenir.

Aşağıdaki örnek, hangi CHLAUTH kuralının yerinde olduğunu denetlemek için bir MQSC komutunu çalıştırır; varsayılan kurallar, kullanıcının `alice` `MQ-admin` kullanıcısı olmayan, `CHAN1`adlı bir kanala erişmesiyle sonuçlanır:

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

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

`alice`kullanıcısı için, kanal çalışır ve kanal `alice` MCAUSER olarak geçer. MCAUSER, IBM MQ nesne yetkilerini denetlemek için kullanılan kullanıcı kimliğidir.

İlgili başvurular
CHLAUTH AYARLA
DISPLAYCHLAUTH

Kullanıcılar için yeni CHLAUTH kuralları yaratılması
Kullanıcılar için bazı genel senaryolar ve bunların gerçekleştirilmesine ilişkin CHLAUTH kuralları.

Başlamadan önce

Not: Bu görevdeki adımlar, MQSC komutlarını çalıştırmanızı gerektirir. Bunu nasıl yapacağınız platforma göre değişir. Bkz. MQSC komutlarını kullanarak IBM MQ yönetme.

Bu görev hakkında

CHLAUTH işlemesi için üç varsayılan kural vardır:

- `MQ-admin*` kullanıcıları tarafından tüm kanallara ERIşIM YOK
- TüM SISTEMLERE ERIşIM YOK. * tüm kullanıcılar tarafından kanallar
- SYSTEM.ADMIN.SVRCONN kanalı (`MQ-admin` dışı kullanıcılar)

İlk iki kural tüm kanallara erişimi engeller. Üçüncü kural daha spesifiktir ve bu nedenle kanal SYSTEM.ADMIN.SVRCONN kanalı, bu kanalda erişime izin verir.

Kullanıcılar için yeni CHLAUTH kuralları oluşturmak üzere aşağıdaki senaryolardan birini ya da birkaçını yapılandırın.

Yordam

- **Belirli `MQ-admin` kullanıcıları için erişimi denetleme**
  a) IBM MQ Explorer' den bağlanmak için, yönetim perspektifi için özel olarak kullanılacak bir sunucu bağlantısı kanalı ayarlayın.

  Bu kullanıma ilişkin belirli bir kanalınız ve bağlantıların kabul edilmesini istediğiniz IP adresi ya da adresleriniz var ve bağlantı belirtilen IP adreslerinden birinden değilse `'mqm'` kimliği için erişim engellendi.

  b) IBM MQ Explorer ve `MQ-admin` kullanıcıları için `ADMIN.CHAN`adlı bir SVRCONN kanalı oluşturun.

  Aşağıdaki MQSC komutunu çalıştırın:

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

  c) Sınama için, `MQ-admin` grubunda tanımlı bir kullanıcı olduğundan ve tanımlı olmadığından emin olun.

  Bu senaryoda, `mqadm` `MQ-admin` grubunda yer alsa da `alice` almıyor.

  d) Varsayılan CHLAUTH kurallarının yerinde olduğunu doğrulayın.

  e) Belirli bir kullanıcının belirli IP adreslerinden ADMIN.CHAN ' e `MQ-admin` olarak erişmesine izin vermek için üç kural ekleyin:

– NOACCESS değerini herhangi bir adresten ayarla
– Bu kanal için BLOCKUSER değerini yalnızca nobodykullanıcısını engelleyecek şekilde ayarlayın; bu, *MQADMIN BLOCKUSER değerini geçersiz kılar.
– Belirli bir adres alt ağındaki mqadm kullanıcısına erişime izin ver ve mqadm kullanıcı yetkisine ilişkin MAP ' ye izin ver

Bunu yapmak için aşağıdaki MQSC komutlarını çalıştırın:

```
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)
```

Bu noktada, kullanıcı mqadm ADMIN.CHAN kanalı.

f) İsteğe bağlı: Bu komutların her birinin sonuçlarını görmek için istediğiniz zaman MQSC komutunu MATCH (RUNCHECK) çalıştırabilirsiniz:

```
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)
```

Bu noktada, yalnızca CHLAUTH kaydı olan kullanıcıların ADMIN.CHAN.

- **Belirli bir kullanıcı ve IBM MQ istemci uygulaması için erişimi denetleme**

  Bu senaryoda, doğru IBM MQ yetkisini (setmqautkullanılarak) sağlamak için belirli bir kullanıcı için IBM MQ yetkisinin ayarlanması gerektiği varsayılarak, varsayılan CHLAUTH kuralları yeterlidir.

  Bu senaryoda, MQ-admin kullanıcısı olmayan mqapp1kullanıcısı için yetkiler ayarlanır.

  a) SVRCONN kanalı APP1.CHAN.

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

b) Varsayılan CHLAUTH kuralları uygulanarak, kullanıcı mqapp1 APP1.CHAN kanalı.

IBM MQ istemci uygulamasından gelen kullanıcı kimliği, IBM MQ nesne yetkisi denetimi için kullanılır. Bu durumda, mqapp1 kullanıcısının IBM MQ istemci uygulamasını çalıştırdığı varsayılarak, bu IBM MQ nesne yetkisi denetimi için kullanılır. Bu nedenle, mqapp1 ' in uygulamanın gereksinim duyduğu IBM MQ nesnelerine erişimi varsa, sorun yoktur; yoksa yetki hataları elde edeceksiniz.

mqapp1 kullanıcı kimliği için belirli CHLAUTH kuralları oluşturarak güvenliği daha da artırabilirsiniz, ancak varsayılan kurallar altında MQ-admin grubunun hiçbir üyesi bu kanala erişemiyor.

Aşağıdaki MQSC komutlarını çalıştırın:

```
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)
```

- **Belirli bir kullanıcıya ilişkin erişimi, o kullanıcının sertifika ayırt edici adını (DN) kullanarak denetleyebilirsiniz**

  Bu senaryoda, kullanıcının kuyruk yöneticisine akıtılan bir sertifikası olmalıdır. Ayırt edici ad, CHLAUTH kuralının SSLPEER ayarıyla eşleştirilir ve SSLPEER genel arama karakterlerini kullanabilir.

  Eşleştirilirse, kullanıcı IBM MQ nesne yetkilerini denetlemek amacıyla farklı bir MCAUSER ile de eşlenebilir. MCAUSER ' in eşlenmesi, IBM MQ nesne yetki yöneticisinde (OAM) yönetilmesi gereken kullanıcı sayısını en aza indirebilir.

  a) Sertifikaları olan bir TLS kanalınız var ve aşağıdakiler için kurallar gereklidir:

  – Belirli bir kanal için tüm kullanıcıları engelle
  – IBM MQ OAM erişimi için yalnızca o kullanıcının istemcisini kullanan belirli bir SSLPEER ' e sahip kullanıcılara izin verin.

  Aşağıdaki MQSC komutlarını çalıştırın:

```
.
# 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)
```

  Kanalda bağlanan istemci kullanıcı kimliği, IBM MQ nesnelerinin IBM MQ OAM yetkisi için kullanılır; bu nedenle, kullanıcı kimliğinin uygun IBM MQ yetkileri olmalıdır.

  b) İsteğe bağlı: Farklı bir IBM MQ kullanıcı kimliğiyle eşleyin.

  USERSRC(CHANNEL)için USERSRC(MAP) MCAUSER('mquser1') yerine önceki MQSC komutunu kullanarak yeniden çalıştırın.

- **Belirli bir kullanıcıyı `mqm` kullanıcısıyla eşleme**

  Bu, Belirli MQ-admin kullanıcıları için erişimi denetleseçeneğine ilişkin bir ekleme ya da değişikliktir.

  Belirli kullanıcıları IBM MQ OAM ' da IBM MQ nesne yetkisi ayarına sahip `mqm` kullanıcısıyla ya da bir `MQ-admin` kullanıcı kimliğiyle eşlemek üzere aşağıdaki CHLAUTH kuralını eklemek için MQSC komutlarını kullanın.

```
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)
```

  Bu, `johndoe` kullanıcısını belirli bir kanal için `mqm` kullanıcısıyla eşler ADMIN.CHAN.

İlgili kavramlar

“Kanallar için yeni CHLAUTH kuralları yaratılması” sayfa 66
Kendi CHLAUTH kurallarınızı oluşturmanıza yardımcı olmak için, burada kanallar için bazı yaygın senaryolar ve bunları gerçekleştirmek için CHLAUTH kuralları bulunmaktadır.

İlgili görevler

“CHLAUTH erişim sorunlarını çözme” sayfa 61
Kanal kimlik doğrulama kayıtlarını (CHLAUTH) kullanırken belirli erişim sorunlarını çözmek için adımlar ve örnekler.

İlgili başvurular

CHLAUTH AYARLA
DISPLAYCHLAUTH

Kanallar için yeni CHLAUTH kuralları yaratılması
Kendi CHLAUTH kurallarınızı oluşturmanıza yardımcı olmak için, burada kanallar için bazı yaygın senaryolar ve bunları gerçekleştirmek için CHLAUTH kuralları bulunmaktadır.

Bu konu aşağıdaki senaryoları içerir:

- "Yalnızca belirli bir IP adresi aralığından belirli bir kanala erişime izin verin." sayfa 66
- "Belirli bir kanal için, tüm kullanıcıları engelleyin, ancak belirli kullanıcıların bağlanmasına izin verin." sayfa 66
- "Alıcı ve gönderen kanalları için CHLAUTH kullanılması" sayfa 67

Yalnızca belirli bir IP adresi aralığından belirli bir kanala erişime izin verin.

Bu senaryo için şunları yapmak istiyorsunuz:

- Herhangi bir yerden kanala erişim yok olarak ayarlayın
- Belirli bir IP adresi ya da adres aralığından erişime izin ver

```
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)
```

Bu, yalnızca APP2.CHAN kanalı, bağlantı belirlenen IP adresi aralığından geldiğinde başlatılır.

MCAUSER olarak bağlanan kullanıcı `mqapp2`ile eşlenir ve bu nedenle bu kullanıcı için IBM MQ OAM yetkisini alır.

Belirli bir kanal için, tüm kullanıcıları engelleyin, ancak belirli kullanıcıların bağlanmasına izin verin.

CHLAUTH işlemesi için üç varsayılan kural vardır:

- `MQ-admin*` kullanıcıları tarafından tüm kanallara ERIşIM YOK
- TüM SISTEMLERE ERIşIM YOK. * tüm kullanıcılar tarafından kanallar
- SYSTEM.ADMIN.SVRCONN kanalı (`MQ-admin` dışı kullanıcılar)

İlk iki kural tüm kanallara erişimi engeller. Üçüncü kural daha spesifiktir ve bu nedenle kanal SYSTEM.ADMIN.SVRCONN kanalı, bu kanalda erişime izin verir.

Bu senaryoda, MY.SVRCONN kanalına erişim varsayılan CHLAUTH kurallarına sahiptir.

Aşağıdakileri eklemeniz gerekir:

```
# 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)
```

Kodun bu ilk bölümü, MY.SVRCONN' da herkesin bağlanmasını engeller, ardından kod, bağlantı belirli bir kullanıcı kimliğinden geldiğinde yalnızca MY.SVRCONN kanalının başlatılmasına izin verir `johndoe`.

`johndoe` kanalına bağlanan kullanıcı, IBM MQ nesnelerinin IBM MQ OAM yetkisi için kullanılır. Bu nedenle, kullanıcı kimliğinin uygun IBM MQ yetkileri olmalıdır.

Aşağıdaki işlemleri kullanarak, isterseniz farklı bir IBM MQ kullanıcı kimliğiyle eşleyebilirsiniz:

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

USERSRC(CHANNEL)yerine.

Alıcı ve gönderen kanalları için CHLAUTH kullanılması

Alıcı ve gönderen kanallarına ek güvenlik eklemek, alıcı kanalına erişimi kısıtlamak için CHLAUTH kurallarını kullanabilirsiniz. CHLAUTH kurallarına ekleme ya da değişiklik yapıyorsanız, güncellenen CHLAUTH kuralları yalnızca kanal başlatılırken geçerlidir; bu nedenle kanallar zaten çalışıyorsa, CHLAUTH güncellemelerinin uygulanması için bunları durdurmanız ve yeniden başlatmanız gerekir.

CHLAUTH kuralları herhangi bir kanalda kullanılabilir, ancak bazı kısıtlamalar vardır. Örneğin, USERMAP kuralları yalnızca SVRCONN kanalları için geçerlidir.

Bu örnek, TO.MYSVR1 kanalı:

```
# 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)
```

Bu örnek, yalnızca belirli bir kuyruk yöneticisinden gelen bağlantıya izin verir:

```
# 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)
```

İlgili görevler

“CHLAUTH erişim sorunlarını çözme” sayfa 61
Kanal kimlik doğrulama kayıtlarını (CHLAUTH) kullanırken belirli erişim sorunlarını çözmek için adımlar ve örnekler.

“Kullanıcılar için yeni CHLAUTH kuralları yaratılması” sayfa 63
Kullanıcılar için bazı genel senaryolar ve bunların gerçekleştirilmesine ilişkin CHLAUTH kuralları.

İlgili başvurular

CHLAUTH AYARLA

DISPLAYCHLAUTH

CHLAUTH geri durdurma kuralı oluşturma

Kuyruk yöneticinize gelen bağlantıların denetimini düşünürken iki seçeneğiniz vardır. İzin verilmeyen tüm bağlantıları listelemeye çalışabilir ya da tüm bağlantılara izin verilmediğini söyleyerek başlayabilir ve daha sonra, izin verilen tüm bağlantıları listelemeye çalışabilirsiniz. Bu ikinci seçenek burada açıklanmıştır.

Bu görev hakkında

İkinci seçeneği kullanmanın nedeni, izin verilmeyen tüm bağlantıları listelemeye çalışırsanız ve bu nedenle listede yer almayan her şeye izin verilirse, listede bir bağlantının eksik olmasının sonucu, izin verilmemesi gereken bir bağlantının bağlanamaması olur ve olası bir güvenlik ihlaline neden olur.

Tersine, bunun yerine, her bağlantıya izin verilmediğini söyleyerek başlarsınız ve sonra bu listedeki bağlantılardan birini kaybetmenin sonucu bir güvenlik ihlali değildir. Kuruluşunuz ek bağlantıların eklenmesini gerektiriyorsa, bu nispeten basit bir görevdir, ancak olası bir güvenlik ihlali yoktur.

Yapılacak ilk iş, daha belirli kurallarla eşleşmeyen bağlantıları yakalayan bir kural olan *back-stop* kuralı yaratmaktır. Bu kural, uzak bağlantıların kuyruk yöneticinizle bağlantı kurmasını engellemenizi sağlar.

Ancak, bu yaklaşımla ilgili endişeleriniz varsa, uyarı kipinde *back-stop* kuralını ayarlayabilirsiniz; bkz. adım “2” sayfa 68

Yordam

1. Kuyruk yöneticinize bağlı uzak bağlantıları durduran bir geri durdurma kuralı oluşturmak için aşağıdaki komutu verin:

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

   Artık tüm uzak bağlantılarda kapıyı kapattığınıza göre, belirli bağlantılara izin vermek için daha özel kurallar koymaya başlayabilirsiniz. Örneğin:

   ```
   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. Geri durdurma kuralını uyarı kipinde yaratmak istiyorsanız şu komutu verin:

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

   Şimdi devam edebilir ve tüm olumlu kurallarınızı yapabilirsiniz. Gereksinim duyduğunuz tüm kuralları oluşturduğuna inandığınızda, aşağıdaki komutu vererek kanal olaylarını açın:

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

   and monitor the SYSTEM.ADMIN.CHANNEL.EVENT queue for events with **Reason** set to MQRC_CHANNEL_BLOCKED_WARNING.

   Bu olaylar, arka durdurma kuralınızla eşleşen bağlantıları ayrıntılı olarak gösterir, ancak komut uyarı kipinde çalıştığından, şu an için engellenmedi.

   Bu olayların her birini gözden geçirin ve bu bağlantının izin vermek için yerinde pozitif bir kural olup olmadığını ya da *back-stop* kuralıyla doğru bir şekilde eşleşip eşleşmediğini belirleyin. Tüm gelen kanalları gördüğünüzden memnun oluncaya ve tümü için uygun pozitif kurallara sahip oluncaya kadar olayları oluşturuldukları gibi gözden geçirerek bu modda çalıştırabilirsiniz.

   Bu noktada, *back-stop* kuralını, aşağıdaki komutu vererek eşleştiği bağlantıları gerçekten engellemeye başlayacak şekilde değiştirebilirsiniz:

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

Ayrıcalıklı olmayan bir IBM MQ yöneticisi oluşturma
CHLAUTH kullanarak ayrıcalıklı olmayan bir IBM MQ yöneticisi oluşturma.

Bu görev hakkında

Bu görev bağlamında, terimler:

ayrıcalıklı kullanıcı
Bu işlemi gerçekleştirmek için belirtik olarak erişim verilmeden işlem gerçekleştirme yetkisi olan bir kullanıcı anlamına gelir. mqm grubundaki kullanıcılar, bu ayrıcalıklı kullanıcıların örnekleridir.

IBM MQ yöneticisi
IBM MQiçin **DEFINE QLOCAL** ya da **START CHANNEL**gibi yönetimle ilgili komutlar verme gereksinimi olan bir kullanıcı anlamına gelir.

Aşağıdaki adımlar ayrıcalıklı olmayan bir IBM MQ yöneticisi yaratır.

Yordam

1. İşletmenizin kullandığı platforma ya da platformlara ilişkin uygun komutları kullanarak kuyruk yöneticisi makinesinde bir kullanıcı kimliği yaratın.

   Bu örnekte `alice` kullanıcı adı kullanılmıştır.
2. Aşağıdaki yordamı gerçekleştirerek tüm IBM MQ yönetim komutlarını yayınlamak için bu yeni kullanıcıya yetki verin:
   a) Ayrıcalıklı bir kullanıcı kullanarak IBM MQ Explorer programını başlatın.
   b) *Rol Tabanlı Sihirbaz* ' a gitmek için uygun kuyruk yöneticisini seçin, ardından `Nesne Yetkilileri` ve `Rol Tabanlı Yetkiler Ekle`öğelerini seçin.
   c) Açılan sihirbaz panosunda, ilk adımda yarattığınız kullanıcı kimliğini girin ya da gruplarla çalışmayı tercih ediyorsanız, ayrıcalıklı olmayan IBM MQ denetimcilerine yapmak istediğiniz kullanıcı ya da kullanıcı kümesini girin.
   d) Sihirbazı tam denetim erişimi için ayarlayın.
   e) Ayrıcalıklı olmayan IBM MQ yöneticinizin kuyruklardaki iletilere göz atabilmesini istiyorsanız, bu onay kutusunu da seçin.
   f) Sihirbazın alt kısmındaki önizleme panosunda bulunan komutları gözden geçirin.

      Kendi komut dosyalarınızı oluşturmak için bu komutları kesip yapıştırabilirsiniz.

      Bunu kendi komut dosyanızla yapmak istemenizin bir nedeni, bu kullanıcıya verdiğiniz erişim miktarını azaltmaktır. Tüm nesnelere erişim vermek yerine, yalnızca belirli bir nesne grubuna erişim vermek isteyebilirsiniz.

      Sihirbazda **OK** (Tamam) düğmesine basılması, komutları gösterildiği gibi verir.
   g) Ayrıcalıklı olmayan IBM MQ yöneticisi için de uzaktan erişim gerekmesi durumunda, bu kullanıcı kimliği için uzaktan erişime izin vermek üzere bazı CHLAUTH kuralları ayarlamanız gerekir.

      İşletmenizin “CHLAUTH geri durdurma kuralı oluşturma” sayfa 67içindeki kılavuzu kullandığını varsayarak, yapmanız gereken tek şey bir etkinleştirme kuralı eklemektir.

      Oluşturduğunuz kural, uzak IBM MQ yöneticilerinizin kimliğini nasıl doğrulamayı seçtiğinize bağlıdır.

      Zayıf TCP/IP kimlik doğrulaması kullanıyorsanız, aşağıdaki gibi görünen bir CHLAUTH kuralı ayarlayabilirsiniz:

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

      9. TLS kimlik doğrulaması kullanıyorsanız, aşağıdaki gibi görünen bir CHLAUTH kuralı ayarlayabilirsiniz:

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

      Şimdi, bir kullanıcı `admin-channel-name` ' a bağlandığında (ve CHLAUTH kurallarıyla eşleştiğinde), kuyruk yöneticisinde `alice` kullanıcı kimliği altında komut yayınlayabildikleri için ayrıcalıklı uzaktan erişim gerekmez.

Bağlantı kimlik doğrulaması

Bağlantı kimlik doğrulaması, uygulamaların bir kuyruk yöneticisine bağlandıklarında kimlik doğrulama kimlik bilgilerini sağlamalarına olanak sağlar. Kuyruk yöneticisi kimlik bilgilerini doğrular. Kimlik bilgilerinde sağlanan kullanıcı kimliği, uygulamanın eriştiği kaynaklar için yetkilendirme denetimlerinde kullanılmak üzere de benimsenebilir.

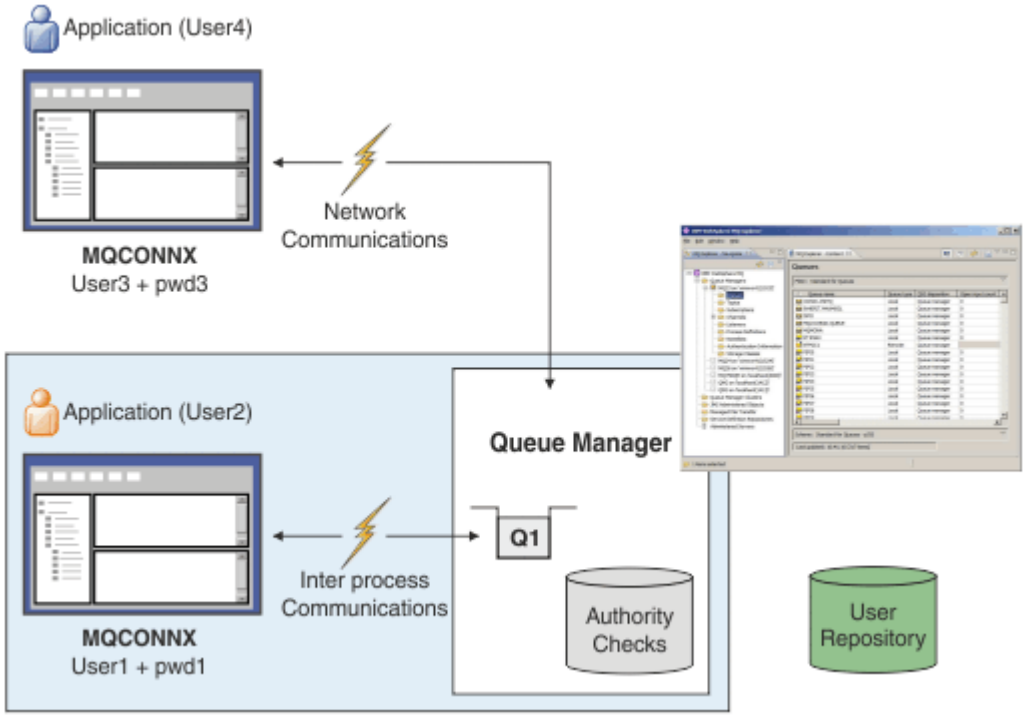
Uygulamalar, bir kuyruk yöneticisine bağlandıklarında kimlik doğrulaması için bir kullanıcı kimliği ve parola sağlayabilir.

V 9.3.4 IBM MQ 9.3.4' den IBM MQ client uygulamaları, alternatif bir kimlik doğrulama yöntemi olarak bir kimlik doğrulama belirteci de sağlayabilir.

Kuyruk yöneticisi, uygulama tarafından sağlanan kimlik bilgilerini doğrulayacak şekilde yapılandırılabilir.

Bir uygulama tarafından sağlanan bir kullanıcı kimliği ve parola, kuyruk yöneticisi yapılandırmasındaki kullanıcı havuzu kullanılarak denetlenir. Kullanıcı kimliklerini ve parolaları denetlemek için kullanılan havuzla ilgili daha fazla bilgi için bkz. Kullanıcı havuzları.

V 9.3.4 Kimlik doğrulama belirteçleri, belirtecin imzasını doğrulamak için kuyruk yöneticisinin belirteç kimlik doğrulama anahtar deposundaki sertifikalar ve simetrik anahtarlar kullanılarak doğrulanır. Kimlik doğrulama belirteçleriyle kullanıcıların kimliğini doğrulama hakkında daha fazla bilgi için bkz. “Kimlik doğrulama belirteçleriyle çalışma” sayfa 344.



Çizgede, iki uygulama bir kuyruk yöneticisiyle, bir uygulama istemci olarak ve bir uygulama da yerel bağ tanımlarını kullanarak bağlantı kuruyor. Uygulamalar kuyruk yöneticisine bağlanmak için çeşitli API ' leri kullanabilir, ancak tümü bir kullanıcı kimliği ve parola sağlama yeteneğine sahiptir. IBM MQ' e sunulan olağan işletim sistemi kullanıcı kimliği olan şemada, uygulamanın altında çalıştığı kullanıcı kimliği `User2` ve `User4` , uygulama tarafından sağlanan kullanıcı kimliğinden (`User1` ve `User3`) farklı olabilir.

Kuyruk yöneticisi yapılanış komutlarını alır (çizgede IBM MQ Explorer kullanılıyor) ve kaynakların açılmasını yönetir ve bu kaynaklara erişim yetkisini denetler. IBM MQ içinde bir uygulamanın erişim yetkisi gerektirebileceği birçok farklı kaynak vardır. Çizge, çıkış için bir kuyruk açılmasını gösterir, ancak aynı ilkeler diğer kaynaklar için de geçerlidir.

İlgili kavramlar

“Bağlantı kimlik doğrulaması: Yapılandırma” sayfa 71
Bir kuyruk yöneticisi, bağlandığında bir uygulama tarafından sağlanan kimlik bilgilerini doğrulayacak şekilde yapılandırılabilir.

“Bağlantı kimlik doğrulaması: Uygulama değişiklikleri” sayfa 75

“Bağlantı kimlik doğrulaması: Kullanıcı havuzları” sayfa 76

Kuyruk yöneticilerinizin her biri için, kullanıcı kimliklerini ve parolaları doğrulamak üzere farklı tiplerde kimlik doğrulama bilgileri nesnesi seçebilirsiniz.

Bağlantı kimlik doğrulaması: Yapılandırma

Bir kuyruk yöneticisi, bağlandığında bir uygulama tarafından sağlanan kimlik bilgilerini doğrulayacak şekilde yapılandırılabilir.

Kuyruk yöneticisinde bağlantı kimlik doğrulamasını açma

Bir kuyruk yöneticisi nesnesinde, **CONNAUTH** özniteliği bir kimlik doğrulama bilgileri (AUTHINFO) nesnesinin adına ayarlanabilir. AUTHINFO nesnesinin **AUTHTYPE** özniteliği, nesnenin tipini belirtir. Bağlantı doğrulaması için kullanılan AUTHINFO nesneleri aşağıdaki iki tipten biri olabilir:

IDPWOS
Kuyruk yöneticisi, bağlanan bir uygulama tarafından sağlanan kullanıcı kimliğini ve parolayı doğrulamak için yerel işletim sistemini kullanır.

Linux AIX V 9.3.4 IBM MQ 9.3.4' den bu tip AUTHINFO nesnesi, AIX ya da Linux üzerinde çalışan bir kuyruk yöneticisinin kimlik doğrulama belirteçlerini doğrulamasını da sağlar. Bağlantı kimlik doğrulamasını yapılandırmak için kullanılan AUTHINFO nesnesine ek olarak, kuyruk yöneticisi `qm.ini` dosyasının **`AuthInfo`** kısmı ile kimlik doğrulama belirteçlerini kabul edecek şekilde yapılandırılmalıdır. Bir kuyruk yöneticisinin kimlik doğrulama belirteçlerini kabul edecek şekilde yapılandırılmasıyla ilgili daha fazla bilgi için bkz. “Bir kuyruk yöneticisinin kimlik doğrulama belirteçlerini kabul edecek şekilde yapılandırılması” sayfa 349.

IDPWLDAP
Kuyruk yöneticisi, bağlanan bir uygulama tarafından sağlanan kullanıcı kimliğini ve parolayı doğrulamak için bir LDAP sunucusu kullanır.

Not: Kuyruk yöneticisinin **CONNAUTH** özniteliğinde başka bir kimlik doğrulama bilgisi nesnesi tipi belirtemezsiniz.

`IDPWOS` ve `IDPWLDAP` tipindeki AUTHINFO nesneleri, bazı özniteliklerinde benzerdir. Burada açıklanan öznitelikler her iki nesne tipi için de ortaktır.

Aşağıdaki örnek MQSC komutları, aşağıdaki işlemlerle bağlantı kimlik doğrulamasını açar:

1. `USE.PW`adlı bir AUTHINFO nesnesi tanımlayın.
2. Kuyruk yöneticisi **CONNAUTH** özniteliğini, bu AUTHINFO nesnesine gönderme yapmak üzere değiştirin.
3. Kuyruk yöneticisinin bağlantı kimlik doğrulama yapılanışını yenilemek için **REFRESH SECURITY** komutunu verin. Kuyruk yöneticisi, bağlantı kimlik doğrulama yapılandırmasında yapılan değişiklikleri tanımadan önce **REFRESH SECURITY** komutu verilmelidir.

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

ALTER QMGR CONNAUTH(USE.PW)

REFRESH SECURITY TYPE(CONNAUTH)
```

Yerel olarak bağlı uygulamalar tarafından yapılan bağlantılar için kimlik bilgilerinin denetlenip denetlenmediğini denetlemek için **CHCKLOCL** AUTHINFO özniteliğini kullanın (yerel bağlantıları denetleyin). İstemci uygulamaları tarafından yapılan bağlantılar için kimlik bilgilerinin denetlenip denetlenmediğini denetlemek için **CHCKCLNT** AUTHINFO özniteliğini kullanın (istemci bağlantılarını denetleyin).

CHCKLOCL , `NONE` ve `OPTIONAL`değerlerini kabul eder ve **CHCKCLNT** , kimlik doğrulama gereksinimlerinin yapılandırılması için `NONE` değerini verir:

YOK
Uygulamalar tarafından sağlanan kimlik doğrulama kimlik bilgileri denetlenmez.

İsteğe Bağlı
Bir uygulama tarafından sağlanan kimlik bilgilerinin geçerli olmasını sağlar. Ancak, uygulamaların kimlik doğrulama kimlik bilgileri sağlaması zorunlu değildir. Bu seçenek geçiş sırasında yararlı olabilir; örneğin.

Aşağıdakileri yaparsanız:

- Kullanıcı adı ve parolayı sağlayın, bunların kimliği doğrulandı.
- Kullanıcı adı ve parola girmeyin, bağlantıya izin verilir.
- Kullanıcı adını belirtin, ancak bir hata aldığınız parolayı sağlamayın.

Önemli: İSTEğE BAğLı , kanal kimlik doğrulaması (CHLAUTH) kurallarında daha kısıtlayıcı bir seçenek belirlemek istiyorsanız ayarlayabileceğiniz en düşük değerdir.

NONE seçeneğini belirlerseniz ve istemci bağlantısı **CHCKCLNT** ayarı REQUIRED (ya da z/OSdışındaki platformlarda REQDADM) olan bir CHLAUTH kaydıyla eşleşirse, bağlantı başarısız olur. z/OSdışındaki platformlarda AMQ9793 iletisini ve z/OSüzerinde CSQX793E iletisini alırsınız.

Bazı istemci bağlantıları için daha kısıtlayıcı **CHCKCLNT** seçenekleri ayarlamak üzere kanal kimlik doğrulama kurallarını kullanma hakkında daha fazla bilgi için bkz. “Yapılandırma ayrıntı düzeyi” sayfa 72.

ZORUNLU
Tüm uygulamaların geçerli kimlik bilgileri sağlamasını gerektirir. Aşağıdaki nota da bakın.

REQDADM
Ayrıcalıklı kullanıcılar geçerli kimlik bilgileri sağlamalıdır, ancak ayrıcalıklı olmayan kullanıcılar İSTEğE BAğLı ayarında işlem görür. Aşağıdaki nota da bakın. z/OS (Bu ayara z/OS sistemlerinde izin verilmez.)

Not:

CHCKLOCL değerinin REQUIRED ya da REQDADM olarak ayarlanması, kullanıcı **runmqsc** komutunda kullanıcı kimliğini belirtmek için **-u** parametresini belirtmedikçe **runmqsc** (hata AMQ8135: Yetkili değil) komutunu kullanarak kuyruk yöneticisini yerel olarak yönetemeyeceğiniz anlamına gelir. Bu parametre ayarlandığında, **runmqsc** konsolda kullanıcının parolasını ister.

Benzer şekilde, yerel sistemde IBM MQ Explorer komutunu çalıştıran bir kullanıcı, kuyruk yöneticisine bağlanmaya çalışırken AMQ4036 hatasını görür. Bir kullanıcı kimliği ve parola belirtmek için, yerel kuyruk yöneticisi nesnesini sağ tıklatın ve **Bağlantı Ayrıntıları** > **Özellikler ...** seçeneğini belirleyin. menüden. **Kullanıcı kimliği** bölümünde, kullanılacak kullanıcı kimliğini ve parolayı girin ve **Tamam**düğmesini tıklatın.

CHCKCLNTile yapılan uzak bağlantılar için de benzer noktalar geçerlidir.

Kuyruk yöneticisi **CONNAUTH** özniteliği, IBM MQ 8.0' den önceki sürümlerden geçirilen, ancak *SYSTEM.DEFAULT.AUTHINFO.IDPWOS* . Bu varsayılan **AUTHINFO** tanımı, **CHCKCLNT** varsayılan olarak REQDADM değerine ayarlıdır.

Bu nedenle, bağlanmak için ayrıcalıklı bir kullanıcı kimliği kullanan var olan istemciler geçerli kimlik bilgileri sağlamalıdır.

Uyarı: Bir istemci uygulamasına ilişkin MQCSP yapısındaki kimlik bilgileri bazen ağ üzerinden düz metin olarak gönderilir. İstemci kimlik bilgilerinin korunduğundan emin olmak için bkz. “MQCSP parola koruması” sayfa 31.

Yapılandırma ayrıntı düzeyi

AUTHINFO nesnesinin **CHCKLOCL** ve **CHCKCLNT** öznitelikleri, kuyruk yöneticisine yönelik tüm bağlantılar için kimlik doğrulama gereksinimlerini ayarlar. Bu özniteliklere ek olarak, kanal kimlik doğrulaması (CHLAUTH) kurallarındaki **CHCKCLNT** özniteliği, CHLAUTH kuralıyla eşleşen belirli istemci bağlantıları için daha sıkı kimlik doğrulama gereksinimlerinin ayarlanmasına izin verir.

CHLAUTH kuralında **CHCKCLNT** değerini REQUIRED ya da REQDADM olarak ayarlayarak, genel **CHCKCLNT** değerini örneğin, AUTHINFO nesnesinde OPTIONALolarak ayarlayabilir ve daha sonra belirli kanallar için

daha sıkı olacak şekilde yükseltebilirsiniz. Varsayılan olarak CHLAUTH kuralları **CHCKCLNT(ASQMGR)** ile tanımlanır, bu nedenle bu ayrıntı düzeylerinin kullanılması gerekmez. Örneğin, bu MQSC komutları, AUTHINFO nesnesinin **CHCKCLNT** özniteliğini geçersiz kılan bir CHLAUTH kuralı ve aşağıdakileri yapmayan bir CHLAUTH kuralı tanımlar:

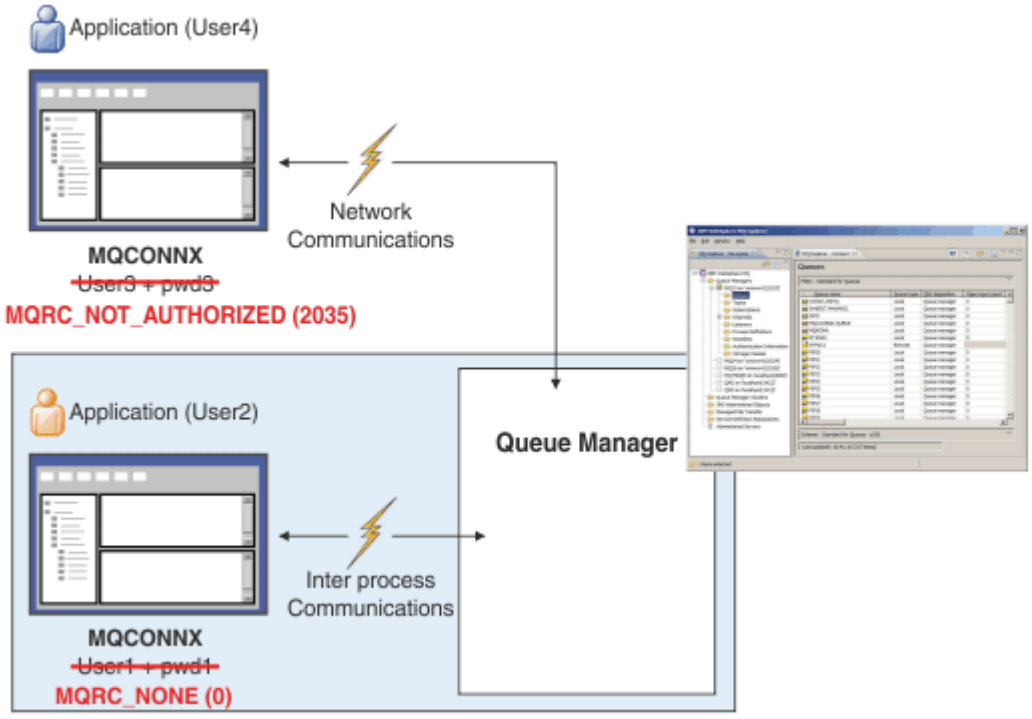
```
DEFINE AUTHINFO(USE.PW) AUTHTYPE(xxxxxx) +
CHCKCLNT(OPTIONAL)

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

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

CHLAUTH kuralları hakkında daha fazla bilgi için bkz. “Kanal kimlik doğrulama kayıtları” sayfa 50.

Hata bildirimi



Aşağıdaki durumlarda bir hata kaydedilir:

- Bir uygulama gerektiğinde kimlik doğrulama kimlik bilgilerini sağlamaz.
- Bir uygulama geçersiz kimlik doğrulama kimlik bilgileri sağlıyor. Yapılandırma, uygulamaların kimlik bilgilerini sağlamasının isteğe bağlı olduğunu belirtse de, bu durum bir hata olarak kabul edilir.

Not: CHCKLOCL ya da **CHCKCLNT** NONEolarak ayarlandığında, uygulamalar tarafından sağlanan geçersiz kimlik bilgileri algılanmaz.

Başarısız kimlik doğrulamaları, hata uygulamaya döndürülmeden önce **FAILDLAY** özniteliği tarafından belirtilen saniye sayısı kadar tutulur. Bu gecikme, sürekli olarak bağlanmaya çalışan bir uygulamaya karşı koruma sağlar.

Hata birkaç şekilde kaydedilir:

Uygulama
Uygulamaya bir MQRC_NOT_AUTHORIZED (2035) neden kodu döndürüldü.

Yönetici
IBM MQ yöneticisi, hata günlüğünde bildirilen olayı görür. Hata iletisi, kullanıcının bağlantı yetkisi olmadığı için değil, kimlik bilgileri geçersiz olduğu için bağlantının reddedildiğini gösterir.

İzleme aracı

Yetki olaylarını açmanız durumunda, SYSTEM.ADMIN.QMGR.EVENT kuyruğundaki bir olay iletisiyle hata bir izleme aracına da bildirilebilir. Yetki olaylarını açmak için aşağıdaki MQSC komutunu verin:

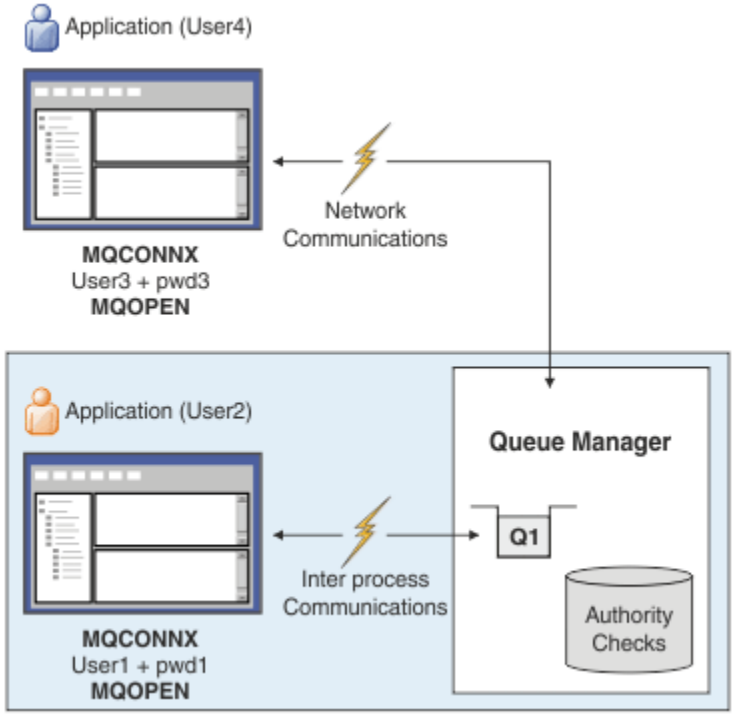
```
ALTER QMGR AUTHOREV(ENABLED)
```

Bu "Not Authorized" (Yetkili Değil) olayı bir Tip 1 bağlanma olayıdır ve sağlanan ek bir alan olan MQCSP kullanıcı kimliği ile diğer Tip 1 olaylarıyla aynı alanları sağlar. Uygulama bir parola sağladıysa, bu parola olay iletisine eklenmez. Bu, olay iletisinde iki kullanıcı kimliği olduğu anlamına gelir:

- Uygulamanın altında çalıştığı kullanıcı kimliği.
- Uygulamanın sunduğu kimlik bilgilerindeki kullanıcı kimliği.

Bu olay iletisiyle ilgili daha fazla bilgi için bkz. Not Authorized (type 1).

Yetkilendirme için kullanıcıları benimseyen



Kuyruk yöneticisini, uygulama tarafından bağlantı bağlamı olarak sunulan kimlik bilgilerini kabul edecek şekilde yapılandırabilirsiniz. Kimlik bilgilerinin benimsenmesi, kimlik doğrulama kimlik bilgilerinde sağlanan kullanıcı kimliğinin, yönetim görüntülerinde gösterilen yetkilendirme denetimleri için kullanıldığı ve iletilerde görüntülendiği anlamına gelir. AUTHINFO nesnesindeki **ADOPTCTX** özniteliği, kimlik bilgilerinin uygulama bağlamı olarak kullanılıp kullanılmayacağını denetler. Örneğin, aşağıdaki MQSC komutları, bağlantı kimlik doğrulaması için kullanılan USE.PWD adlı bir AUTHINFO nesnesini tanımlar ve **ADOPTCTX** özniteliğini YESolarak ayarlar:

```
DEFINE AUTHINFO(USE.PWD) +
AUTHTYPE(xxxxxx) +
CHCKLOCL(OPTIONAL) +
CHCKCLNT(REQUIRED) +
ADOPTCTX(YES)

ALTER QMGR CONNAUTH(USE.PWD)
```

ADOPTCTX özniteliği için aşağıdaki değerler belirtilebilir:

ADOPTCTX (EVET)
Uygulama tarafından sağlanan kimlik bilgileri, bağlantı süresi boyunca uygulama bağlamı olarak kabul edilir. Bir uygulamaya ilişkin tüm yetki denetimleri, kimliği doğrulanan kimlik bilgilerinde kullanıcı kimliği ile yapılır.

Uyarı: ADOPTCTX(YES) ve yerel işletim sistemi kullanıcı kimliklerini kullanırken, benimsenmekte olan kullanıcı kimliğinin IBM MQiçindeki kullanıcı kimliklerine ilişkin gereksinimleri karşıladığından emin olmanız gerekir. Daha fazla bilgi için bkz "Kullanıcı Kimlikleri" sayfa 87.

ADOPTCTX (HAYıR)
Bir uygulama tarafından sağlanan kimlik bilgileri yalnızca bağlantı sırasında kimlik doğrulaması için kullanılır. Uygulamanın altında çalıştığı kullanıcı kimliği, ileride yapılacak yetki denetimleri için kullanılmaya devam eder. Bu seçeneği geçiş sırasında ya da ileti kanalı aracısı kullanıcı kimliğini (MCAUSER)atamak için kanal kimlik doğrulama kayıtları gibi diğer mekanizmaları kullanmayı planlıyorsanız yararlı bulabilirsiniz.

Kanal Kimlik Doğrulamasıyla Etkileşim

Kanal kimlik doğrulama kuralları, istemciden alınan kullanıcı kimliğine dayalı olarak, bir uygulama bağlantısının bağlamı olarak kullanılan kullanıcı kimliğini değiştirmek için kullanılabilir. Bir bağlantıyla ilişkili kullanıcı kimliğini değiştirmek için kanal kimlik doğrulama kuralı kullanma örneği için bkz. "Bir istemci kullanıcı kimliğinin MCAUSER kullanıcı kimliğiyle eşlenmesi" sayfa 401.

Bağlantı kimlik doğrulaması ve kanal kimlik doğrulama kurallarının işlendiği sıra, IBM MQ istemci uygulaması bağlantıları için güvenlik bağlamının belirlenmesinde önemli bir etmendir. `qm.ini` dosyasının **`channels`** kısmındaki **`ChlauthEarlyAdopt`** parametresi, kuyruk yöneticisinin uygulama tarafından sağlanan kimlik bilgilerinden bağlamı benimseme sırasını denetler ve kanal kimlik doğrulama kurallarını uygular. **`ChlauthEarlyAdopt`**ile ilgili daha fazla bilgi için bkz. Kanalların öznitelikleri.

Uyarı: Kimlik doğrulama bilgileri nesnesinde **`ADOPTCTX(YES)`** parametresini kullandığınızda, uygulama tarafından sağlanan kimlik bilgilerinden benimsenen bağlam yalnızca **`ChlauthEarlyAdopt`** parametresi Yolarak ayarlanırsa kanal kimlik doğrulama kuralları tarafından değiştirilebilir.

Bağlantı kimlik doğrulaması ve kanal kimlik doğrulamasının etkileşimi ve istemci uygulamasının bir kuyruk yöneticisine bağlanması sırasında oluşan denetim sırası hakkında daha fazla bilgi için bkz. "CHLAUTH ve CONNAUTH Etkileşimi" sayfa 56.

İlgili kavramlar
"Bağlantı kimlik doğrulaması" sayfa 69
Bağlantı kimlik doğrulaması, uygulamaların bir kuyruk yöneticisine bağlandıklarında kimlik doğrulama kimlik bilgilerini sağlamalarına olanak sağlar. Kuyruk yöneticisi kimlik bilgilerini doğrular. Kimlik bilgilerinde sağlanan kullanıcı kimliği, uygulamanın eriştiği kaynaklar için yetkilendirme denetimlerinde kullanılmak üzere de benimsenebilir.

"Bağlantı kimlik doğrulaması: Uygulama değişiklikleri" sayfa 75

"Bağlantı kimlik doğrulaması: Kullanıcı havuzları" sayfa 76
Kuyruk yöneticilerinizin her biri için, kullanıcı kimliklerini ve parolaları doğrulamak üzere farklı tiplerde kimlik doğrulama bilgileri nesnesi seçebilirsiniz.

Bağlantı kimlik doğrulaması: Uygulama değişiklikleri

İleti kuyruğu arabirimini (MQI) kullanan bir uygulama, MQCONNX çağrıldığında bağlantı güvenliği değiştirgelerinde (MQCSP) bir kullanıcı kimliği ve parola sağlayabilir. Diğer uygulama programlama arabirimlerinde, MQCSP yapısı genellikle uygulama adına IBM MQ kitaplıkları tarafından oluşturulur.

V 9.3.4 IBM MQ 9.3.4' den, AIX ya da Linux sistemlerinde çalışan bir kuyruk yöneticisine bağlanan istemci uygulamaları, MQCSP yapısında bir kimlik doğrulama simgesini diğer bir tanımlama aracı olarak da gönderebilir.

Kullanıcı kimliği ve parolaya da kimlik doğrulama simgesi , kuyruk yöneticisiyle birlikte sağlanan nesne yetkisi yöneticisine (OAM) ya da z/OS sistemlerinde kuyruk yöneticisiyle birlikte sağlanan yetkilendirme hizmeti bileşenine denetleme için iletilir. Kendi özel arabiriminizi yazmanız gerekmez.

Uygulama bir istemci, kullanıcı kimliği ve parolaya da kimlik doğrulama simgesi olarak çalışıyorsa, istemci tarafı ve sunucu tarafı güvenlik çıkışlarına işlenmek üzere geçirilir. Bunlar, bir kanal eşgörünümünün ileti kanalı aracısı kullanıcı kimliği (MCAUSER) özniteliğini ayarlamak için de kullanılabilir.

Uyarı: Bir istemci uygulamasına ilişkin MQCSP yapısındaki kimlik bilgileri bazen ağ üzerinden düz metin olarak gönderilir. İstemci uygulaması kimlik bilgilerinin korunduğundan emin olmak için bkz. “MQCSP parola koruması” sayfa 31.

Kullanıcı kimliği ve parola sağlamak için XAOPEN dizgisini kullanarak uygulama kodunu değiştirmek zorunda kalmaktan kaçınabilirsiniz.

Not:

IBM WebSphere MQ 6.0olanağından, güvenlik çıkışı MQCSP ' nin ayarlanmasına izin verir. Bu nedenle, bu düzeydeki ya da daha sonraki düzeydeki istemcilerin yükseltilmesi gerekmez.

Ancak, IBM MQ 8.0öncesi IBM MQ sürümlerinde MQCSP, uygulama tarafından sağlanan kullanıcı kimliği ve parolayla ilgili herhangi bir kısıtlama getirmedi. Bu değerleri IBM MQ tarafından sağlanan özelliklerle kullanırken, bu özelliklerin kullanımı için geçerli olan sınırlar vardır, ancak bunları yalnızca kendi çıkışlarınıza geçirirseniz, bu sınırlar geçerli olmaz.

İlgili kavramlar

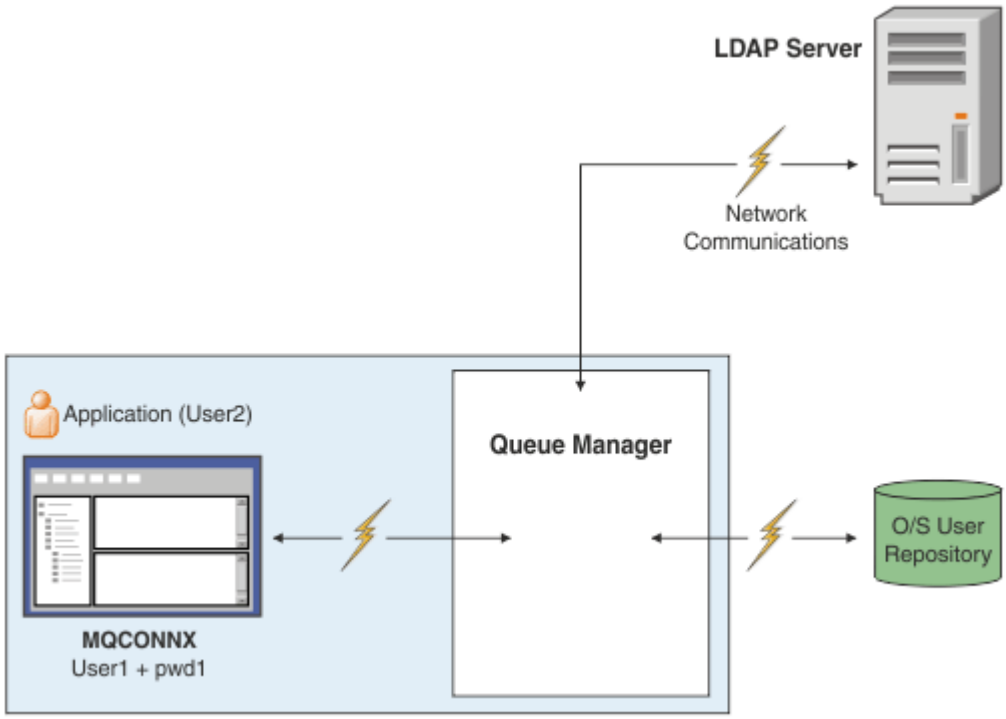
“Bağlantı kimlik doğrulaması” sayfa 69
Bağlantı kimlik doğrulaması, uygulamaların bir kuyruk yöneticisine bağlandıklarında kimlik doğrulama kimlik bilgilerini sağlamalarına olanak sağlar. Kuyruk yöneticisi kimlik bilgilerini doğrular. Kimlik bilgilerinde sağlanan kullanıcı kimliği, uygulamanın eriştiği kaynaklar için yetkilendirme denetimlerinde kullanılmak üzere de benimsenebilir.

“Bağlantı kimlik doğrulaması: Yapılandırma” sayfa 71
Bir kuyruk yöneticisi, bağlandığında bir uygulama tarafından sağlanan kimlik bilgilerini doğrulayacak şekilde yapılandırılabilir.

“Bağlantı kimlik doğrulaması: Kullanıcı havuzları” sayfa 76
Kuyruk yöneticilerinizin her biri için, kullanıcı kimliklerini ve parolaları doğrulamak üzere farklı tiplerde kimlik doğrulama bilgileri nesnesi seçebilirsiniz.

Bağlantı kimlik doğrulaması: Kullanıcı havuzları

Kuyruk yöneticilerinizin her biri için, kullanıcı kimliklerini ve parolaları doğrulamak üzere farklı tiplerde kimlik doğrulama bilgileri nesnesi seçebilirsiniz.



Şekil 7. Kimlik doğrulama bilgisi nesnelerinin tipleri

```
DEFINE AUTHINFO(USE.OS) AUTHTYPE(IDPWOS)
DEFINE AUTHINFO(USE.LDAP) +
AUTHTYPE(IDPWLDAP) +
CONNAME('ldap1(389),ldap2(389)') +
LDAPUSER('CN=QMGR1') +
LDAPPWD('passw0rd') SECCOMM(YES)
```

Çizgede gösterildiği gibi, iki tip kimlik doğrulama bilgisi nesnesi vardır:

- `IDPWOS` , kuyruk yöneticisinin kullanıcı kimliğini ve parolayı doğrulamak için yerel işletim sistemini kullandığını belirtmek için kullanılır. Yerel işletim sistemini kullanmayı seçerseniz, önceki konularda açıklandığı gibi ortak öznitelikleri ayarlamanız gerekir.
- `IDPWLDAP` , kuyruk yöneticisinin kullanıcı kimliğini ve parolayı doğrulamak için bir LDAP sunucusu kullandığını belirtmek için kullanılır. LDAP sunucusu kullanmayı seçerseniz, bu konuda daha fazla bilgi sağlanır.

Kuyruk yöneticisinin **CONNAUTH** özniteliğinde uygun nesne adlandırılarak, her kuyruk yöneticisinin kullanması için tek bir kimlik doğrulama bilgisi nesnesi seçilebilir.

Kimlik doğrulaması için LDAP sunucusu kullanılıyor.

CONNAME alanını, kuyruk yöneticisine ilişkin LDAP sunucusunun adresine ayarlayın. LDAP sunucusu için virgülle ayrılmış bir listede daha fazla adres sağlayabilirsiniz; bu, LDAP sunucusu bu olanağı sağlamazsa yedekliliğinize yardımcı olabilir.

Kuyruk yöneticisinin LDAP sunucusuna erişebilmesi ve kullanıcı kayıtlarıyla ilgili bilgileri arayabilmesi için **LDAPUSER** ve **LDAPPWD** alanlarında gerekli LDAP sunucusu kimliğini ve parolasını ayarlayın.

LDAP Sunucusuna Güvenli Bağlantı

Kanallardan farklı olarak, LDAP sunucusuyla iletişim için TLS kullanımını açmak için **SSLCIPH** parametresi yoktur. Bu durumda IBM MQ , LDAP sunucusu için istemci işlevi görür ve yapılandırmanın çoğu LDAP sunucusunda yapılır. IBM MQ içinde var olan bazı parametreler, bağlantının nasıl çalıştığını yapılandırmak için kullanılır.

LDAP sunucusuna bağlanırlığın TLS kullanıp kullanmayacağını denetlemek için **SECCOMM** alanını ayarlayın.

Bu özniteliğe ek olarak, kuyruk yöneticisi öznitelikleri **SSLFIPS** ve **SUITEB** , seçilen şifreleme belirtimleri kümesini kısıtlar. Kuyruk yöneticisini LDAP sunucusuna tanıtmak için kullanılan sertifika, `ibmwebspheremq` *qmgr-name* kuyruk yöneticisi sertifikasıdır ya da **CERTLABL** özniteliğinin değeridir. Ayrıntılar için bkz. Dijital sertifika etiketleri .

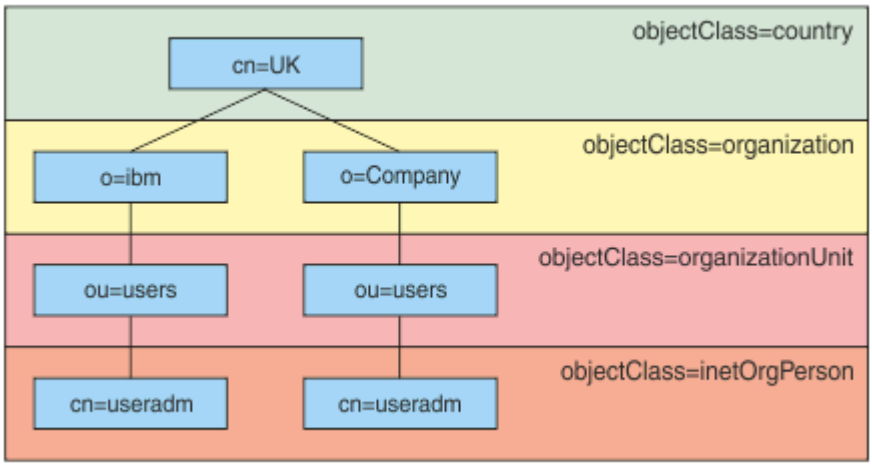
LDAP Kullanıcı Havuzu

Bir LDAP kullanıcı havuzu kullanılırken, kuyruk yöneticisine LDAP sunucusunu nerede bulacağını söylemek dışında, kuyruk yöneticisinde yapılması gereken başka bir yapılanış vardır.

LDAP sunucusunda tanımlanan kullanıcı kimlikleri, bunları benzersiz olarak tanımlayan sıradüzenli bir yapıya sahiptir. Bu nedenle, bir uygulama kuyruk yöneticisine bağlanabilir ve kullanıcı kimliğini tam olarak nitelenmiş sıradüzenli kullanıcı kimliği olarak sunabilir.

Ancak, bir uygulamanın sağlaması gereken bilgileri basitleştirmek için, kuyruk yöneticisini, sıradüzenin ilk kısmının tüm tanıtıcılar için ortak olduğunu varsaymak ve uygulama tarafından sağlanan kısaltılmış tanıtıcıdan önce bunu otomatik olarak eklemek üzere yapılandırmak mümkündür. Kuyruk yöneticisi daha sonra LDAP sunucusuna tam bir kimlik sunabilir.

BASEDNU değerini, LDAP aramasının LDAP sıradüzeninde kimliği aradığı ilk noktaya ayarlayın. BASEDNU ' yu ayarladığınızda, LDAP sıradüzeninde kimliği ararken yalnızca bir sonuç döndürüldüğünden emin olmanız gerekir.



Şekil 8. Örnek bir LDAP sıradüzeni

Örneğin, Şekil 8 sayfa 78 BASEDNU içinde "ou=users, o=ibm, c = UK" ya da ", o=ibm, c = UK" olarak ayarlanabilir. Ancak, hem "o = ibm" dalında hem de "o=Company" dalında "cn = useradm" içeren bir ayırt edici ad bulunduğundan, BASEDNU "c = UK" olarak ayarlanamaz. Performans ve güvenlik nedenleriyle, LDAP sıradüzeninizdeki, gereksinim duyduğunuz tüm kullanıcı kimliklerine başvurabileceğiniz en yüksek noktayı kullanın. Bu örnekte "ou=users, o=ibm, c = UK".

Uygulamanız, LDAP öznitelik adını (örneğin, CN=) sağlamadan kuyruk yöneticisine kullanıcı kimliğini sunabilir. USRFIELD değerini LDAP öznitelik adına ayarlarsanız, bu değer uygulamadan gelen kullanıcı kimliğine önek olarak eklenir. Bu, işletim sistemi kullanıcı kimliklerinden LDAP kullanıcı kimliklerine geçtiğinizde yararlı bir geçiş yardımı olabilir; böylece uygulama her iki durumda da aynı dizgiyi sunabilir ve uygulamayı değiştirmekten kaçınabilirsiniz.

Bu nedenle, LDAP sunucusuna sunulan tam kullanıcı kimliği şöyle görünür:

```
USRFIELD = ID_from_application BASEDNU
```

İlgili kavramlar

"Bağlantı kimlik doğrulaması" sayfa 69
Bağlantı kimlik doğrulaması, uygulamaların bir kuyruk yöneticisine bağlandıklarında kimlik doğrulama kimlik bilgilerini sağlamalarına olanak sağlar. Kuyruk yöneticisi kimlik bilgilerini doğrular. Kimlik bilgilerinde sağlanan kullanıcı kimliği, uygulamanın eriştiği kaynaklar için yetkilendirme denetimlerinde kullanılmak üzere de benimsenebilir.

"Bağlantı kimlik doğrulaması: Yapılandırma" sayfa 71
Bir kuyruk yöneticisi, bağlandığında bir uygulama tarafından sağlanan kimlik bilgilerini doğrulayacak şekilde yapılandırılabilir.

"Bağlantı kimlik doğrulaması: Uygulama değişiklikleri" sayfa 75

Kullanıcı kimliği ve parola eklemek için istemci tarafı güvenlik çıkışı (`mqccred`)

Kullanıcı kimliği ya da parola göndermek için gerekli istemci uygulamalarınız varsa, ancak henüz kaynağı değiştiremiyorsanız, kullanabileceğiniz IBM MQ 8.0 ile birlikte gönderilen **mqccred** adlı bir güvenlik çıkışı vardır. **mqccred** , bir `.ini` dosyasından istemci uygulaması adına bir kullanıcı kimliği ve parola sağlar. Bu kullanıcı kimliği ve parola, kuyruk yöneticisine gönderilir; bu kullanıcı kimliği ve parola, kuyruk yöneticisine gönderilecek şekilde yapılandırıldıysa, bunların kimliklerini doğrular.

Genel Bakış

mqccred , istemci uygulamanızla aynı makinede çalışan bir güvenlik çıkışıdır. Kullanıcı kimliği ve parola bilgilerinin istemci uygulaması adına sağlanmasına olanak sağlar; burada, bu bilgiler uygulamanın kendisi tarafından sağlanmaz. Kullanıcı kimliği ve parola bilgileri, Bağlantı Güvenlik Parametreleri (MQCSP) olarak bilinen bir yapıda sağlanır ve bağlantı kimlik doğrulaması yapılandırıldıysa kuyruk yöneticisi tarafından doğrulanır.

Kullanıcı kimliği ve parola bilgileri, istemci makinedeki bir `.ini` dosyasından alınır. Dosyadaki parolalar, **runmqccred** komutu kullanılarak gizlenerek ve `.ini` dosyasındaki dosya izinlerinin yalnızca istemci uygulamasını çalıştıran kullanıcı kimliği (ve dolayısıyla çıkış) okuyabilecek şekilde ayarlandığından emin olunarak korunur.

Yer

mqccred kurulu:

Windows Platformlar
installation_directory\Tools\c\Samples\mqccred\ dizininde

AIX and Linux Platformlar
installation_directory/samp/mqccred dizininde

Notlar: Çıkış:

1. Tamamen bir güvenlik kanalı çıkışı görevi görür ve bir kanalda tanımlanan tek çıkış olması gerekir.
2. Genellikle İstemci Kanal Tanımlama Çizelgesi (CCDT) aracılığıyla adlandırılır, ancak bir Java istemcisinin JNDI nesnelerinde doğrudan belirtilen çıkışı olabilir ya da çıkış, MQCD yapısını el ile oluşturan uygulamalar için yapılandırılabilir.
3. **mqccred** ve **mqccred_r** programlarını `var/mqm/exits` dizinine kopyalamanız gerekir.

   Örneğin, 64 bit AIX ya da Linux sisteminde şu komutu verin:

   ```
   cp installation_directory/samp/mqccred/lib64/* /var/mqm/exits
   ```

   Daha fazla bilgi için mqccred 'in nasıl test edileceğine ilişkin adım adım bir örnek başlıklı konuya bakın.
4. IBM WebSphere MQ 7.0.1' e kadar önceki IBM MQsürümlerinde çalışabilir.

Kullanıcı kimliklerini ve parolaları ayarlama

`.ini` dosyası, belirlenmemiş kuyruk yöneticileri için genel bir ayarla birlikte her kuyruk yöneticisi için bölüm içerir. Her bölüm, kuyruk yöneticisinin adını, bir kullanıcı kimliğini ve düz metin ya da gizlenmiş bir parola içerir.

İstediğiniz düzenleyiciyi kullanarak `.ini` dosyasını el ile düzenlemeniz ve stanzas 'a düz metin parolası özniteliğini eklemeniz gerekir. `.ini` dosyasını alan ve **Password** özniteliğini parolanın gizlenmiş bir biçimi olan **OPW** özniteliğiyle değiştiren **runmqccred** programını çalıştırın.

Komutun ve değiştirgelerinin açıklaması için runmqccred komutuna bakın.

`mqccred.ini` dosyası, kullanıcı kimliği ve parola bilgilerinizi içerir.

Kuruluşunuz için bir başlangıç noktası sağlamak üzere çıkışla aynı dizinde bir şablon `.ini` dosyası sağlanır.

Varsayılan olarak, bu dosya $HOME/`.mqs/mqccred.ini`içinde aranacaktır. Yerini başka bir yerde bulmak istiyorsanız, *MQCCRED* ortam değişkenini kullanarak bunu gösterebilirsiniz:

```
MQCCRED=C:\mydir\mqccred.ini
```

*MQCCRED*kullanıyorsanız, değişken, herhangi bir `.ini` dosya tipi de içinde olmak üzere yapılandırma dosyasının tam adını içermelidir. Bu dosya parolalar içerdiğinden (gizlenmiş olsa bile), yetkisiz kişilerin dosyayı okuyamamasını sağlamak için işletim sistemi ayrıcalıklarını kullanarak dosyayı korumanız beklenir. Doğru dosya iznine sahip değilseniz, çıkış başarılı bir şekilde çalışmaz.

Uygulama zaten bir MQCSP yapısı sağladıysa, çıkış normalde bu yapıya uygun olur ve `.ini` kütüğünden herhangi bir bilgi eklemez. Ancak, kıtadaki **Force** özniteliğini kullanarak bunu geçersiz kılabilirsiniz.

Force değeri *TRUE* olarak ayarlandığında, uygulama tarafından sağlanan kullanıcı kimliği ve parola kaldırılır ve ini dosya sürümüyle değiştirilir.

Dosyanın genel bölümündeki **Force** özniteliğini, o dosyanın varsayılan değerini ayarlamak için de ayarlayabilirsiniz.

Force için varsayılan değer *FALSE*değeridir.

Tüm kuyruk yöneticileri için ya da her bir kuyruk yöneticisi için bir kullanıcı kimliği ve parola sağlayabilirsiniz. Bu bir `mqccred.ini` dosyası örneğidir:

```
# 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
```

Notlar:

1. Tek tek kuyruk yöneticisi tanımlamaları, genel ayardan önceliklidir.
2. Öznitelikler büyük ve küçük harfe duyarlı değildir.

Kısıtlamalar

Bu çıkış kullanımdayken, uygulamayı çalıştıran kişinin yerel kullanıcı kimliği istemciden sunucuya akmaz. Kullanılabilir kimlik bilgileri yalnızca ini dosyası içeriğinden elde edilebilir.

Bu nedenle, kuyruk yöneticisini **ADOPTCTX(YES)** kullanacak şekilde yapılandırmanız ya da gelen bağlantı isteğini, kullanılabilir düzeneklerden (örneğin, "Kanal kimlik doğrulama kayıtları" sayfa 50) biri aracılığıyla uygun bir kullanıcı kimliğiyle eşlemeniz gerekir.

Önemli: Yeni parolalar eklerseniz ya da eskileri güncellerseniz, **runmqccred** komutu yalnızca düz metin parolalarını işler ve gizlenmiş parolalarınıza dokunulmaz.

Hata ayıklama

Çıkış, etkinleştirildiğinde standart IBM MQ izlemesine yazar.

Yapılandırma sorunlarında hata ayıklamaya yardımcı olmak için çıkış doğrudan stdout 'a da yazabilir.

Kanal güvenliği çıkış verisi yok (**SCYDATA**) Normalde kanal için yapılandırma gereklidir. Ancak, aşağıdakileri belirtebilirsiniz:

HATA
Yalnızca, yapılandırma dosyasını bulamamak gibi, hata durumlarını içeren bilgileri yazdırın.

HATA AYIKLAMA
Bu hata koşullarını ve bazı ek izleme deyimlerini görüntüler.

NOKS İşARETLERI
Dosya izinleriyle ilgili kısıtlamaları atlar ve `.ini` dosyasının korunmayan parolalar içermemesi gerektiğini belirtir.

Bu öğelerden birini ya da birkaçını, virgülle ayrılmış olarak herhangi bir sırada **SCYDATA** alanına koyabilirsiniz. Örneğin, `SCYDATA=(NOCHECKS,DEBUG)`.

Öğelerin büyük ve küçük harfe duyarlı olduğunu ve büyük harfle girilmesi gerektiğini unutmayın.

Kullanılan `mqccred`

Dosyanızı ayarladıktan sonra, istemci-bağlantı kanalı tanımlamanızı `SCYEXIT('mqccred(ChlExit)')` özniteliğini içerecek şekilde güncelleyerek kanal çıkışını başlatabilirsiniz:

```
DEFINE CHANNEL(channelname) CHLTYPE(clntconn) +
CONNAME(remote machine) +
QMNAME(remote qmgr) +
SCYEXIT('mqccred(ChlExit)') +
REPLACE
```

İlgili başvurular
SCYDATA
SCYEXIT
mqccred

Java istemcisiyle bağlantı kimlik doğrulaması

Bağlantı kimlik doğrulaması, kuyruk yöneticisinin sağlanan bir kullanıcı kimliğini ve parolayı kullanarak uygulamaları doğrulayabilmesi için kuyruk yöneticilerini yapılandırmanızı sağlayan bir özelliktir. IBM MQ Uygulama, istemci iletimi kullanan bir Java uygulamasıyken, bağlantı kimlik doğrulaması uyumluluk kipinde ya da MQCSP kimlik doğrulama kipinde çalıştırılabilir.

Kimliği doğrulanacak kullanıcı kimliği ve parola, uygulama tarafından aşağıdaki yöntemlerden biri kullanılarak belirtilir:

- Bir IBM MQ classes for Java uygulamasında, `MQEnvironment` sınıfında ya da `com.ibm.mq.MQQueueManager` oluşturucusuna geçirilen Hashtable özelliklerinde.
- Bir IBM MQ classes for JMS uygulamasında, `createConnection(String username, String Password)` ya da `createContext(String username, String password)` yöntemine ilişkin bağımsız değişkenler olarak.

MQCSP kimlik doğrulama kipi

Bu kipte, uygulamanın altında çalıştığı istemci tarafı kullanıcı kimliği kuyruk yöneticisine ve kimliği doğrulanacak kullanıcı kimliği ve parolaya gönderilir. IBM MQ classes for Java ve IBM MQ classes for JMS , bir MQCSP yapısında kuyruk yöneticisine kimliği doğrulanacak kullanıcı kimliğini ve parolayı gönderir.

Kullanıcı kimliği ve parola, MQCSP yapısındaki bir sunucu bağlantısı güvenlik çıkışında kullanılabilir. MQCSP yapı adresi, kanala ilişkin MQCXP yapısının **SecurityParms** alanında bulunabilir.

MQCSP kimlik doğrulama kipinin yararları şunlardır:

- Kimliği doğrulanacak kullanıcı kimliği uzunluğu üst sınırı 1024 karakterdir.
- Kimlik doğrulaması için parola uzunluğu üst sınırı 256 karakterdir.
- IBM MQ kaynaklarını kullanmak için yetki denetimleri, kuyruk yöneticisinde bağlantı kimlik doğrulamasını denetlemek için kullanılan kimlik doğrulama bilgileri nesnesi ADOPTCTX (NO) ile yapılandırıldığında, uygulamanın altında çalıştığı istemci tarafı kullanıcı kimliği kullanılarak gerçekleştirilebilir.

Uyumluluk kipi

IBM MQ 8.0' den önce Java istemcisi, istemci-bağlantı kanalından sunucu bağlantısı kanalına bir kullanıcı kimliği ve parola gönderebilir ve bunları MQCD yapısının **RemoteUserIdentifier** ve **RemotePassword** alanlarında bir güvenlik çıkışına gönderebilir. Uyumluluk kipinde bu davranış korunur.

Bu kipi, bağlantı kimlik doğrulamasıyla birlikte kullanabilir ve daha önce aynı işi yapmak için kullanılan güvenlik çıkışlarından uzaklaşabilirsiniz.

Bu kip aşağıdaki kısıtlamaları içerir:

- Kullanıcı kimliği ve parola uzunluğu 12 karakter ya da daha az olmalıdır. 12 karakterden uzun kullanıcı kimlikleri 12 karaktere kesilir. Bu, bağlantının MQRC_NOT_AUTHORIZED neden koduyla başarısız olmasına neden olabilir.
- Uygulamanın çalıştığı istemci tarafı kullanıcı kimliği kuyruk yöneticisine gönderilmez. IBM MQ kaynaklarını kullanma yetkisi için denetlenen kanal MCA kullanıcı kimliğini ayarlamak için, kuyruk yöneticisinde bağlantı kimlik doğrulamasını denetlemek için kullanılan kimlik doğrulama bilgileri nesnesinde ADOPTCTX (YES) değerini ayarlamanız ya da TLS sertifikasını temel alan bir kanal kimlik doğrulama kuralı gibi başka bir yöntem kullanmanız gerekir.

Varsayılan kimlik doğrulama kipi

Bir IBM MQ classes for Java ya da IBM MQ classes for JMS istemci uygulaması tarafından kullanılan varsayılan kimlik doğrulama kipi, uygulamanın bir kullanıcı kimliği ve parola belirtip belirtmediğine bağlı olarak değişir.

- V 9.3.0 IBM MQ 9.2.1' den bir kullanıcı kimliği ve parola belirtilirse, varsayılan olarak MQCSP kimlik doğrulaması kullanılır.
- IBM MQ 9.2.1öncesi sürümlerde, bir kullanıcı kimliği ve parola belirtilirse, varsayılan kip aşağıdaki gibidir:
  - MQCSP kimlik doğrulaması, varsayılan olarak IBM MQ classes for Javakullanan uygulamalar tarafından kullanılır.
  - Uyumluluk kipi varsayılan olarak IBM MQ classes for JMSkullanan uygulamalar tarafından kullanılır.
- Bir kullanıcı kimliği belirtilirse, ancak parola belirtilmezse, varsayılan olarak uyumluluk kipi kullanılır.
- Kullanıcı kimliği belirtilmezse, uyumluluk kipi her zaman kullanılır.

Kullanıcı kimliğinin belirtilmesi durumunda, uygulama tarafından her bir bağlantı için belirli bir kimlik doğrulama kipi seçilebilir ya da uygulama başlatılmadan önce genel olarak ayarlanabilir (açıklamalar için bkz. “Kimlik doğrulama kipinin seçilmesi” sayfa 83).

Not: V 9.3.0 IBM MQ classes for JMS kullanan uygulamalar, IBM MQ 9.3.0içinde varsayılan kimlik doğrulama kipindeki değişiklikten etkilenebilir. IBM MQ classes for JMS ürününü IBM MQ 9.3.0düzeyine yükselttikten sonra, varsayılan olarak daha önce uyumluluk kipini kullanan uygulamalar bunun yerine MQCSP kimlik doğrulamasını kullanır. Bu, daha önce bir kuyruk yöneticisine başarıyla bağlanan uygulamaların, 2035 neden kodunu (MQRC_NOT_AUTHORIZED) içeren bir `JMSException` ile bağlantı kuramamasına neden olabilir. Bu durumda, uygulamanın uyumluluk kipini kullandığını belirtmek için "Kimlik doğrulama kipinin seçilmesi" sayfa 83 içinde açıklanan yöntemlerden birini kullanın.

Yerel bağ tanımlarını kullanarak kuyruk yöneticisine bağlanan Java uygulamaları her zaman MQCSP kimlik doğrulama kipini kullanır.

Kimlik doğrulama kipinin seçilmesi

Kuyruk yöneticisine bağlanırken kullanıcı kimliği belirten Java istemci uygulamaları tarafından kullanılan kimlik doğrulama kipi, aşağıdaki yöntemlerden biri kullanılarak belirtilebilir. Bu yöntemler öncelik sırasını azaltarak listelenir. Kimlik doğrulama kipi bu yöntemlerden herhangi biri kullanılarak belirtilmezse, varsayılan kimlik doğrulama kipi kullanılır.

Not: V 9.3.0 Kimlik doğrulama kipini seçmek için bu yöntemlerin kullanımı IBM MQ 9.3.0içinde açıklığa kavuşturulmuştur. Bazı durumlarda, bir Java istemci uygulaması tarafından kullanılan kimlik doğrulama kipi IBM MQ classes for Java ya da IBM MQ classes for JMS , IBM MQ 9.3.0olarak yükseltildiğinde değişebilir. Bu, daha önce bir kuyruk yöneticisine başarıyla bağlanan uygulamaların, 2035 neden kodunu (MQRC_NOT_AUTHORIZED) içeren bir `JMSException` ile bağlantı kuramamasına neden olabilir. Bu durumda, gerekli kimlik doğrulama kipini seçmek için aşağıdaki yöntemlerden birini kullanın.

- Kuyruk yöneticisine bağlanmadan önce uygulamada uygun özelliği ayarlayarak her bir bağlantı için kimlik doğrulama kipini belirtin.
  - IBM MQ classes for Javakullanırken, `com.ibm.mq.MQQueueManager` oluşturucusuna geçirilen Hashtable özelliklerinde *MQConstants.USE_MQCSP_AUTHENTICATION_PROPERTY* özelliğini ayarlayın.
  - IBM MQ classes for JMSkullanırken *JmsConstantsözelliğini ayarlayın.Bağlantıyı yaratmadan önce uygun bağlantı üreticisinde USER_AUTHENTICATION_MQCSP* .

  Bu özelliklerin değerini aşağıdaki değerlerden birine ayarlayın:

  doğru
  Bir kuyruk yöneticisiyle kimlik doğrulaması yaparken MQCSP kimlik doğrulama kipini kullanın.

  yanlış
  Bir kuyruk yöneticisiyle kimlik doğrulaması yaparken uyumluluk kipini kullanın.
- Uygulamayı başlatırken *com.ibm.mq.cfg.jmqi.useMQCSPauthentication* Java sistem özelliğini ayarlayarak bir uygulama tarafından yapılan tüm istemci bağlantıları için kimlik doğrulama kipini belirtin. Özelliğin değerini aşağıdaki değerlerden birine ayarlayın:

  Y
  Bir kuyruk yöneticisiyle kimlik doğrulaması yaparken MQCSP kimlik doğrulama kipini kullanın.

  N
  Bir kuyruk yöneticisiyle kimlik doğrulaması yaparken uyumluluk kipini kullanın.

  Örneğin, aşağıdaki komut özelliği uyumluluk kipini seçecek şekilde ayarlar ve bir Java uygulamasını başlatır:

  ```
  java -Dcom.ibm.mq.cfg.jmqi.useMQCSPauthentication=N application_name
  ```
- Uygulamanın başlatıldığı ortamda *com.ibm.mq.jmqi.useMQCSPauthentication* ortam değişkenini ayarlayarak, aynı ortamda başlatılan uygulamalar tarafından yapılan tüm istemci bağlantıları için kimlik doğrulama kipini belirtin. Ortam değişkeninin değerini aşağıdaki değerlerden birine ayarlayın:

  Y
  Bir kuyruk yöneticisiyle kimlik doğrulaması yaparken MQCSP kimlik doğrulama kipini kullanın.

N
Bir kuyruk yöneticisiyle kimlik doğrulaması yaparken uyumluluk kipini kullanın.

- İstemci yapılanış kütüğünün JMQI kısmına **useMQCSPauthentication** özniteliğini belirterek, belirli bir IBM MQ MQI client istemcisi yapılanış kütüğünü kullanan tüm uygulamalar için kimlik doğrulama kipini belirtin. Özniteliğin değerini aşağıdaki değerlerden birine ayarlayın:

  EVET
  Bir kuyruk yöneticisiyle kimlik doğrulaması yaparken MQCSP kimlik doğrulama kipini kullanın.

  HAYIR
  Bir kuyruk yöneticisiyle kimlik doğrulaması yaparken uyumluluk kipini kullanın.

  useMQCSPauthentication özniteliğiyle ilgili daha fazla bilgi için bakınız: JMQI stanza of the client configuration file.

IBM MQ Explorer içinde kimlik doğrulama kipinin seçilmesi

IBM MQ Explorer bir Java uygulamasıdır, bu nedenle uyumluluk kipi ve MQCSP kimlik doğrulama kipi olmak üzere bu iki kip de bu uygulamaya uygulanabilir.

IBM MQ 9.1.0' den MQCSP kimlik doğrulama kipi varsayılan kiptir. IBM MQ 9.1' den önce uyumluluk kipi varsayılan kiptir.

Kullanıcı kimliğinin sağlandığı panolarda, uyumluluk kipini etkinleştirmek ya da devre dışı bırakmak için bir onay kutusu vardır:

- IBM MQ 9.1.0' dan varsayılan olarak bu onay kutusu seçilmez. Uyumluluk kipini kullanmak için bu onay kutusunu işaretleyin.
- IBM MQ 9.1.0' den önce, varsayılan olarak bu onay kutusu etkindir. MQCSP kimlik doğrulamasını kullanmak için onay kutusunu temizleyin.

İlgili kavramlar
“Bağlantı kimlik doğrulaması” sayfa 69
Bağlantı kimlik doğrulaması, uygulamaların bir kuyruk yöneticisine bağlandıklarında kimlik doğrulama kimlik bilgilerini sağlamalarına olanak sağlar. Kuyruk yöneticisi kimlik bilgilerini doğrular. Kimlik bilgilerinde sağlanan kullanıcı kimliği, uygulamanın eriştiği kaynaklar için yetkilendirme denetimlerinde kullanılmak üzere de benimsenebilir.

“Bağlantı kimlik doğrulaması: Uygulama değişiklikleri” sayfa 75

“Bağlantı kimlik doğrulaması: Kullanıcı havuzları” sayfa 76
Kuyruk yöneticilerinizin her biri için, kullanıcı kimliklerini ve parolaları doğrulamak üzere farklı tiplerde kimlik doğrulama bilgileri nesnesi seçebilirsiniz.

IBM MQ içinde ileti güvenliği

IBM MQ altyapısında ileti güvenliği Advanced Message Securitytarafından sağlanır.

Advanced Message Security (AMS) İleti düzeyinde veri imzalama ve şifreleme sağlamak için IBM MQ güvenlik hizmetlerini genişletir. Genişletilmiş hizmetler, ileti verilerinin başlangıçta bir kuyruğa yerleştirildiği zaman ile alındığı zaman arasında değiştirilmediğini garanti eder. Buna ek olarak AMS , ileti verilerini gönderen bir kullanıcının imzalı iletileri hedef kuyruğa yerleştirme yetkisinin olduğunu doğrular.

İlgili kavramlar
“Advanced Message Security” sayfa 609
Advanced Message Security (AMS), son uygulamaları etkilemediği halde IBM MQ ağı üzerinden akan hassas veriler için yüksek düzeyde koruma sağlayan bir IBM MQ bileşenidir.

Güvenlik gereksinimlerinizin planlanması

Bu konu grubu, IBM MQ ortamında güvenliği planlarken nelerin dikkate alınması gerektiğini açıklar.

IBM MQ çeşitli platformlar üzerinde çok çeşitli uygulamalar için kullanılabilir. Güvenlik gereksinimlerinin her uygulama için farklı olması olasıdır. Bazıları için, güvenlik kritik önem taşıyacak.

IBM MQ , Transport Layer Security (TLS) desteği de dahil olmak üzere bir dizi bağlantı düzeyinde güvenlik hizmeti sağlar.

IBM MQürününü kurmayı planlarken güvenliğin belirli yönlerini göz önünde bulundurmanız gerekir:

- Multi Çoklu platformlarüzerinde, bu yönleri yoksayar ve hiçbir şey yapmazsanız, IBM MQ' yi kullanamazsınız.
- z/OS z/OSüzerinde, bu yönleri yoksaymanın etkisi, IBM MQ kaynaklarınızın korunmamasıdır. Yani, tüm kullanıcılar tüm IBM MQ kaynaklarına erişebilir ve bu kaynakları değiştirebilir.

IBM MQ ürününü denetleme yetkisi

IBM MQ yöneticilerinin aşağıdakileri gerçekleştirmeleri için yetkilerinin olması gerekir:

- IBM MQ yönetimi için komutlar yayınlayın
- Şunu kullanın:IBM MQ Explorer
- IBM i IBM i yönetim panolarını ve komutlarını kullanın.
- z/OS z/OS üzerindeki işlemleri ve denetim panolarını kullanma
- z/OS z/OS üzerinde IBM MQ yardımcı programını (CSQUTIL) kullanın
- z/OS z/OS üzerindeki kuyruk yöneticisi veri kümelerine erişin

Daha fazla bilgi için bkz.

- ALW “AIX, Linux, and Windows üzerinde IBM MQ yönetimi yetkisi” sayfa 414
- IBM i “IBM i üzerinde IBM MQ yönetimi yetkisi” sayfa 89
- z/OS “z/OS üzerinde IBM MQ yönetimi yetkisi” sayfa 90

IBM MQ nesneleriyle çalışma yetkisi

Uygulamalar, MQI çağrıları yayınlayarak aşağıdaki IBM MQ nesnelerine erişebilir:

- Kuyruk yöneticileri
- Kuyruklar
- Süreçler
- Namelistler
- Konular

Uygulamalar, bu IBM MQ nesnelerine erişmek ve kanallara ve kimlik doğrulama bilgileri nesnelerine erişmek için de Programlanır Komut Biçimi (PCF) komutlarını kullanabilir. Bu nesneler IBM MQ tarafından korunabilir; böylece, uygulamalarla ilişkili kullanıcı kimliklerinin bunlara erişmek için gereken yetkiye sahip olması gerekir.

Daha fazla bilgi için “Uygulamaların IBM MQ ' yi kullanması için yetki” sayfa 92 başlıklı konuya bakın.

Kanal güvenliği

İleti kanalı aracılarıyla (MCA ' lar) ilişkili kullanıcı kimliklerinin çeşitli IBM MQ kaynaklarına erişmek için yetkisi olmalıdır. Örneğin, bir MCA bir kuyruk yöneticisine bağlanabilmelidir. Gönderen bir MCA ise, kanal için iletim kuyruğunu açabilmelidir. Alan bir MCA ise, hedef kuyrukları açabilmelidir. Kanalları, kanal başlatıcılarını ve dinleyicileri denetlemesi gereken uygulamalarla ilişkili kullanıcı kimliklerinin ilgili PCF komutlarını kullanması için yetki gerekir. Ancak, çoğu uygulamanın böyle bir erişime ihtiyacı yoktur.

Daha fazla bilgi için “Kanal yetkilendirmesi” sayfa 112 başlıklı konuya bakın.

Dikkat edilmesi gereken ek noktalar

Yalnızca belirli IBM MQ işlevini ya da temel ürün uzantılarını kullanıyorsanız, güvenliğin aşağıdaki yönlerini göz önünde bulundurmanız gerekir:

- “Kuyruk yöneticisi kümeleri için güvenlik” sayfa 124
- “IBM MQ Yayınlama/Abone Olma Güvenliği” sayfa 124
- “IBM MQ Internet Pass-Thru için güvenlik” sayfa 126

Planlama tanımlaması ve kimlik doğrulaması

Hangi kullanıcı kimliklerini kullanacağınıza ve kimlik doğrulama denetimlerini hangi düzeylerde ve nasıl uygulayacağınıza karar verin.

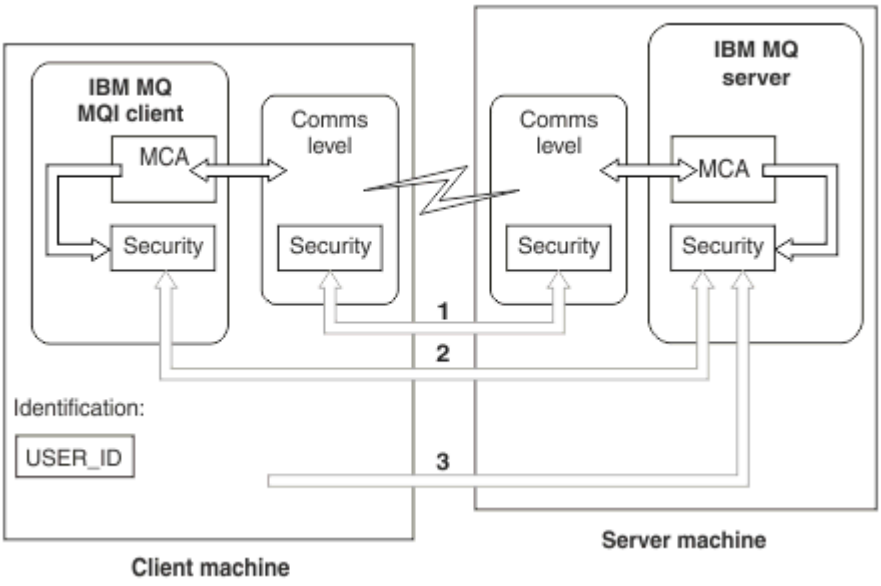
Farklı işletim sistemlerinin farklı uzunluktaki kullanıcı kimliklerini desteklediğini göz önünde bulunduracak şekilde, IBM MQ uygulamalarınızın kullanıcılarını nasıl tanımlayacağınıza karar vermeniz gerekir. Bir kullanıcı kimliğinden diğerine eşlemek ya da bağlantının bazı özniteliğine dayalı bir kullanıcı kimliği belirtmek için kanal kimlik doğrulama kayıtlarını kullanabilirsiniz. TLS kullanan IBM MQ kanalları, kimlik belirleme ve kimlik doğrulama mekanizması olarak dijital sertifikaları kullanır. Her dijital sertifikanın, kanal kimlik doğrulama kayıtları kullanılarak belirli kimliklerle eşlenebilen bir konu ayırt edici adı vardır. Ayrıca, anahtar havuzundaki CA sertifikaları, IBM MQkimlik doğrulaması için hangi dijital sertifikaların kullanılabileceğini belirler. Daha fazla bilgi için bakınız:

- “Uzak kuyruk yöneticisinin MCAUSER kullanıcı kimliğiyle eşlenmesi” sayfa 400
- “Bir istemci kullanıcı kimliğinin MCAUSER kullanıcı kimliğiyle eşlenmesi” sayfa 401
- “Bir SSL ya da TLS Ayırt Edici Adının MCAUSER Kullanıcı Kimliğiyle Eşlenmesi” sayfa 402
- “Bir IP adresinin MCAUSER kullanıcı kimliğiyle eşlenmesi” sayfa 404

Bir istemci uygulaması için kimlik doğrulamasını planlama

Kimlik doğrulama denetimlerini dört düzeyde uygulayabilirsiniz: iletişim düzeyinde, güvenlik çıkışlarında, kanal kimlik doğrulama kayıtlarıyla ve güvenlik çıkışına geçirilen kimlik açısından.

Göz önünde bulundurulması gereken dört güvenlik seviyesi var. Çizge, bir sunucuya bağlı bir IBM MQ MQI client ' i gösterir. Güvenlik, aşağıdaki metinde açıklandığı gibi dört düzeyde uygulanır. MCA bir İleti Kanal Aracısı 'dır.



Şekil 9. İstemci/sunucu bağlantısında güvenlik

1. İletişim düzeyi

   Bkz. ok 1. İletişim düzeyinde güvenlik uygulamak için TLS kullanın. Daha fazla bilgi için bkz. “Şifreleme güvenliği iletişim kuralları: TLS” sayfa 18

2. Kanal kimlik doğrulama kayıtları

   Bkz. oklar 2 ve 3. Kimlik doğrulama, güvenlik düzeyinde IP adresi ya da TLS ayırt edici adları kullanılarak denetlenebilir. Kullanıcı kimliği engellenebilir ya da bildiri içeren bir kullanıcı kimliği geçerli bir kullanıcı kimliğiyle eşlenebilir. “Kanal kimlik doğrulama kayıtları” sayfa 50içinde tam bir açıklama verilir.

3. Bağlantı kimlik doğrulaması

   Bkz. ok 3. İstemci bir kullanıcı kimliği ve parola gönderirya da bir kimlik doğrulama simgesi. Daha fazla bilgi için bkz “Bağlantı kimlik doğrulaması: Yapılandırma” sayfa 71.

4. Kanal güvenliği çıkışları

   Bkz. ok 2. İstemci-sunucu iletişimi için kanal güvenliği çıkışları, sunucudan sunucuya iletişim için olduğu gibi çalışabilir. İstemcinin ve sunucunun karşılıklı kimlik doğrulamasını sağlamak için iletişim kuralından bağımsız bir çıkış çifti yazılabilir. Kanal güvenliği çıkış programlarındatam bir açıklama verilir.

5. Kanal güvenlik çıkışına geçirilen tanıtıcı

   Bkz. ok 3. İstemci-sunucu iletişiminde, kanal güvenliği çıkışlarının çift olarak çalışması gerekmez. IBM MQ istemci tarafındaki çıkış atlanabilir. Bu durumda, kullanıcı kimliği kanal tanımlayıcısına (MQCD) yerleştirilir ve gerekirse sunucu tarafındaki güvenlik çıkışı bunu değiştirebilir.

   IBM MQ MQI clients , tanımlamaya yardımcı olmak için ek bilgi de gönderir.

   - Sunucuya geçirilen kullanıcı kimliği, istemcide oturum açmış olan kullanıcı kimliğidir.
   - Şu anda oturum açmış olan kullanıcının güvenlik kimliği.

   Kullanıcı kimliği ve varsa, güvenlik kimliği değerleri, IBM MQ MQI clientkimliğini oluşturmak için sunucu güvenlik çıkışı tarafından kullanılabilir.

IBM MQ 8.0olanağından, MQCSP yapısının içerdiği parolaları gönderebilirsiniz.

Linux AIX V 9.3.4 IBM MQ 9.3.4' den AIX ya da Linux sistemlerinde çalışan IBM MQ kuyruk yöneticilerine IBM MQ MQI clients bağlanılması, MQCSP yapısında kimlik doğrulama belirteçleri de gönderebilir.

Uyarı: Bazı durumlarda, bir istemci uygulamasına ilişkin MQCSP yapısındaki ya da kimlik doğrulama simgesi parolası ağ üzerinden düz metin olarak gönderilir. İstemci uygulama parolalarının ve kimlik doğrulama simgelerinin uygun şekilde korunduğundan emin olmak için bkz. “MQCSP parola koruması” sayfa 31.

Kullanıcı Kimlikleri

İstemci uygulamaları için kullanıcı kimlikleri yarattığınızda, kullanıcı kimlikleri izin verilen uzunluk üst sınırını aşmamalıdır. UNKNOWN ve UNKNOWN ayrılmış kullanıcı kimliklerini kullanmamalısınız. İstemcinin bağlandığı sunucu bir IBM MQ for Windows sunucusuysa, @ işareti kullanımından kurtulmanız gerekir. Kullanıcı kimliklerinin izin verilen uzunluğu, sunucu için kullanılan platforma bağlıdır:

- z/OS Linux AIX z/OS, AIX and Linuxüzerinde, bir kullanıcı kimliği uzunluğu üst sınırı 12 karakterdir.
- IBM i IBM i' da, kullanıcı kimliği uzunluğu üst sınırı 10 karakterdir.
- Windows Windowssistemlerinde hem IBM MQ MQI client hem de IBM MQ sunucusu Windowsüzerindeyse ve sunucu, istemci kullanıcı kimliğinin tanımlandığı etki alanına erişime sahipse, kullanıcı kimliği uzunluğu üst sınırı 20 karakterdir. Ancak, IBM MQ sunucusu bir Windows sunucusu değilse, kullanıcı kimliği 12 karaktere kesilir.

- Kimlik bilgilerini iletmek için MQCSP yapısını kullanırsanız, kullanıcı kimliği uzunluğu üst sınırı 1024 karakterdir. MQCSP yapısı kullanıcı kimliği, yetki için IBM MQ tarafından kullanılan kullanıcı kimliği uzunluğu üst sınırını aşmak için kullanılamaz. MQCSP yapısıyla ilgili daha fazla bilgi için bkz. "MQCSP yapısını kullanarak kullanıcıların belirlenmesi ve doğrulanmaları" sayfa 341.

AIX and Linux sistemlerinde varsayılan değer, kimlik doğrulaması için kullanıcı kimliklerinin kullanılmasıdır ve gruplar yetkilendirme için kullanılır. Ancak, bu sistemlerin konfigürasyonunu kullanıcı kimliğine karşı yetki vermek üzere tanımlayabilirsiniz. Daha fazla bilgi için bkz "AIX and Linux üzerinde OAM kullanıcı tabanlı izinler" sayfa 368. Windows sistemleri, hem kimlik doğrulama hem de yetkilendirme için gruplar için her iki kullanıcı kimliğini de kullanabilir.

Gruplara dikkat etmeden hizmet hesapları oluşturur ve tüm kullanıcı kimliklerini farklı şekilde yetkilendirirseniz, her kullanıcı diğer kullanıcıların bilgilerine erişebilir.

Kısıtlı kullanıcı kimlikleri

UNKNOWN ve grup NOBODY kullanıcı kimliklerinin IBM MQiçin özel anlamları vardır. İşletim sisteminde UNKNOWN adlı bir kullanıcı kimliği ya da NOBODY adlı bir grup yaratılması istenmeyen sonuçlara neden olabilir.

IBM MQ for Windows sunucusuna bağlanırken kullanıcı kimlikleri

Windows

İstemci @ karakterini içeren bir kullanıcı kimliği altında çalışıyorsa, IBM MQ for Windows sunucusu IBM MQ MQI client bağlantısını desteklemez; örneğin, `abc@d`. İstemcide MQCONN çağrısına dönüş kodu MQRC_NOT_AUTHORIZED.

Ancak, kullanıcı kimliğini iki @ karakteri kullanarak belirtebilirsiniz; örneğin, `abc@@d`. `id@domain` biçiminin kullanılması, kullanıcı kimliğinin tutarlı olarak doğru etki alanında çözümlendiğinden emin olmak için tercih edilen uygulamadır; bu nedenle `abc@@d@domain`.

Planlama yetkisi

Yönetim yetkisine sahip olacak kullanıcıları planlayın ve uygulama kullanıcılarına, IBM MQ MQI clientiçinden bağlantı kuranlar da dahil olmak üzere IBM MQ nesnelerini uygun şekilde kullanma yetkisini nasıl vereceklerini planlayın.

IBM MQkullanabilmek için kişilere ya da uygulamalara erişim verilmelidir. Hangi erişimin gerekli olduğu, üstlendikleri rollere ve gerçekleştirmeleri gereken görevlere bağlıdır. IBM MQ içindeki yetkilendirme iki ana kategoriye bölünebilir:

- Yönetim işlemleri gerçekleştirme yetkisi
- Uygulamaların IBM MQ ' yi kullanması için yetki

Her iki işlem sınıfı da aynı bileşen tarafından denetlenir ve bir kişiye her iki işlem kategorisini gerçekleştirme yetkisi verilebilir.

Aşağıdaki konularda, göz önünde bulundurmanız gereken belirli yetki alanları hakkında daha fazla bilgi verilmektedir:

IBM MQ ürününü denetleme yetkisi

IBM MQ yöneticileri çeşitli işlevleri gerçekleştirmek için yetkiye gereksinim duyar. Bu yetki farklı platformlarda farklı şekillerde elde edilir.

IBM MQ yöneticilerinin aşağıdakileri gerçekleştirmeleri için yetkilerinin olması gerekir:

- IBM MQ' ni yönetmek için komutlar verin.
- Linux Windows IBM MQ Explorerkullanın.
- z/OS z/OSüzerindeki işlemleri ve denetim panolarını kullanın.

- z/OS z/OSüzerinde IBM MQ yardımcı programını (CSQUTIL) kullanın.
- z/OS z/OSüzerindeki kuyruk yöneticisi veri kümelerine erişin.

Daha fazla bilgi için işletim sisteminize uygun konuya bakın.

ALW *AIX, Linux, and Windows sistemlerinde IBM MQ ' yi yönetme yetkisi*

IBM MQ yöneticisi, mqm grubunun bir üyesidir. Bu grubun tüm IBM MQ kaynaklarına erişimi vardır ve IBM MQ denetim komutlarını verebilir. Bir yönetici, diğer kullanıcılara belirli yetkiler verebilir.

AIX, Linux, and Windows sistemlerinde IBM MQ yöneticisi olmak için bir kullanıcının *mqm grubunun*üyesi olması gerekir. Bu grup, IBM MQürününü kurduğunuzda otomatik olarak oluşturulur. Kullanıcıların denetim komutları yayınlamalarına izin vermek için bunları mqm grubuna eklemelisiniz. Bu, AIX and Linuxüzerindeki kök kullanıcıyı içerir.

mqm grubunun üyesi olmayan kullanıcılara denetim ayrıcalıkları verilebilir, ancak bunlar IBM MQ denetim komutlarını veremez ve bu kullanıcılara yalnızca erişim yetkisi verilen komutları yürütme yetkisi verilir.

Ayrıca, Windows sistemlerinde SYSTEM ve Administrator hesaplarının IBM MQ kaynaklarına tam erişimi vardır.

Mqm grubunun tüm üyeleri, sistemde çalışan herhangi bir kuyruk yöneticisini denetleyebilmek de içinde olmak üzere, sistemdeki tüm IBM MQ kaynaklarına erişebilir. Bu erişim yalnızca mqm grubundan bir kullanıcı kaldırılarak iptal edilebilir. Windows sistemlerinde Administrators (Yöneticiler) grubunun üyeleri de tüm IBM MQ kaynaklarına erişebilirler.

Yöneticiler, IBM MQ Script (MQSC) komutlarını yayınlamak için **runmqsc** denetim komutunu kullanabilir. MQSC komutlarını uzak bir kuyruk yöneticisine göndermek için dolaylı kipte **runmqsc** kullanıldığında, her MQSC komutu Escape PCF komutunda kapsüllenir. Uzak kuyruk yöneticisi tarafından işlenecek MQSC komutları için denetimcilerin gerekli yetkileri olmalıdır.

IBM MQ Explorer , denetim görevlerini gerçekleştirmek için PCF komutları verir. Denetimciler, yerel sistemde bir kuyruk yöneticisini denetlemek için IBM MQ Explorer ' yi kullanmak üzere ek yetkilere gereksinim duymaz. IBM MQ Explorer başka bir sistemdeki bir kuyruk yöneticisini denetlemek için kullanıldığında, denetimcilerin PCF komutlarının uzak kuyruk yöneticisi tarafından işlenmesi için gerekli yetkileri olmalıdır.

PCF ve MQSC komutları işlenirken gerçekleştirilen yetki denetimlerine ilişkin ek bilgi için aşağıdaki konulara bakın:

- Kuyruk yöneticileri, kuyruklar, kanallar, süreçler, ad ve kimlik doğrulama bilgileri nesnelerinde çalışan komutlar için bkz. “Uygulamaların IBM MQ ' yi kullanması için yetki” sayfa 92.
- Kanallar, kanal başlatıcılar, dinleyiciler ve kümeler üzerinde çalışan komutlar için bkz. Kanal güvenliği.
- z/OS IBM MQ for z/OSüzerinde komut sunucusu tarafından işlenen MQSC komutları için bkz. “z/OS üzerinde komut güvenliği ve komut kaynağı güvenliği” sayfa 91.

IBM MQ for AIX, Linux, and Windows sistemlerini denetlemek için gereken yetkiye ilişkin ek bilgi için ilgili bilgilere bakın.

IBM i *IBM i üzerinde IBM MQ yönetimi yetkisi*

IBM iüzerinde IBM MQ yöneticisi olmak için *QMQMADM grubunun*bir üyesi olmanız gerekir. Bu grup, AIX, Linux, and Windows sistemlerindeki mqm grubuna benzer özelliklere sahip. Özellikle, IBM MQ for IBM ikurulurken QMQMADM grubu yaratılır ve QMQMADM grubunun üyeleri sistemdeki tüm IBM MQ kaynaklarına erişebilir. *ALLOBJ yetkiniz varsa, tüm IBM MQ kaynaklarına da erişiminiz vardır.

Yöneticiler IBM MQ' i yönetmek için CL komutlarını kullanabilir.Bu komutlardan biri, diğer kullanıcılara yetki vermek için kullanılan GRTMQMAUT komutudur. STRMQMMQSC başka bir komut, bir yöneticinin yerel bir kuyruk yöneticisine MQSC komutları vermesine olanak sağlar.

IBM MQ for IBM itarafından sağlanan iki CL komutu grubu vardır:

Grup 1
Bu kategoride bir komut yayınlamak için kullanıcının QMQMADM grubunun üyesi olması ya da *ALLOBJ yetkisi olması gerekir. GRTMQMAUT ve STRMQMMQSC bu kategoriye aittir; örneğin.

Grup 2
Bu kategoride bir komut yayınlamak için kullanıcının QMQMADM grubunun üyesi olması ya da *ALLOBJ yetkisi olması gerekmez. Bunun yerine, iki yetki düzeyi gereklidir:

- Kullanıcının komutu kullanabilmesi için IBM i yetkisi gerekir. Bu yetki, GRTOBJAUT komutu kullanılarak verilir.
- Kullanıcı, komutla ilişkili herhangi bir IBM MQ nesnesine erişmek için IBM MQ yetkisi gerektirir. Bu yetki, GRTMQMAUT komutu kullanılarak verilir.

Aşağıdaki örneklerde bu gruptaki komutlar gösterilmektedir:

- CRTMQMQ, MQM Kuyruğu Yarat
- CHGMQMPRC, MQM Sürecini Değiştir
- DLTMQMNL, MQM ad çizelgesinin silinmesi
- DSPMQMAUTI, MQM Kimlik Doğrulama Bilgilerini Görüntüle
- CRTMQMCHL, MQM kanalı yarat

Bu komut grubu hakkında daha fazla bilgi için bkz. “Uygulamaların IBM MQ ' yi kullanması için yetki” sayfa 92.

Grup 1 ve grup 2 komutlarının tam listesi için bkz. “IBM i üzerindeki IBM MQ nesnelerine ilişkin erişim yetkileri” sayfa 156

IBM iüzerinde IBM MQ yönetmeniz gereken yetki hakkında daha fazla bilgi için bkz. Yönetme IBM i .

z/OS *z/OS üzerinde IBM MQ yönetimi yetkisi*

Bu konu grubu, IBM MQ for z/OS' i yönetmek için gereksinim duyduğunuz yetkinin çeşitli yönlerini açıklar.

z/OS *z/OS üzerinde yetki denetimleri*

IBM MQ for z/OS , yetki denetimlerine ilişkin istekleri z/OS Security Server Resource Access Control Facility (RACF) gibi bir dış güvenlik yöneticisine (ESM) yönlendirmek için System Authorization Facility (SAF) olanağını kullanır. IBM MQ , kendi yetki denetimlerini yapmaz.

ESM olarak RACF kullandığınız varsayılır. Farklı bir ESM kullanıyorsanız, RACF için sağlanan bilgileri ESM ile ilgili olacak şekilde yorumlamanız gerekebilir.

Her kuyruk yöneticisi için ayrı ayrı mı, yoksa bir kuyruk paylaşım grubundaki her kuyruk yöneticisi için mi yetki denetimlerinin açılmasını ya da kapatılmasını istediğinizi belirtebilirsiniz.Bu denetim düzeyine *altsistem güvenliği*adı verilir. Belirli bir kuyruk yöneticisi için altsistem güvenliğini kapatırsanız, o kuyruk yöneticisi için yetki denetimi gerçekleştirilmez.

Belirli bir kuyruk yöneticisi için altsistem güvenliğini açmanız durumunda, yetki denetimleri iki düzeyde gerçekleştirilebilir:

Kuyruk paylaşımı grup düzeyi güvenliği
Yetki denetimleri, kuyruk paylaşım grubundaki tüm kuyruk yöneticileri tarafından paylaşılan RACF tanıtımlarını kullanır. Bu, tanımlanacak ve korunacak daha az profil olduğu anlamına gelir ve güvenlik yönetimini kolaylaştırır.

Kuyruk yöneticisi düzeyinde güvenlik
Yetki denetimleri, kuyruk yöneticisine özgü RACF tanıtımlarını kullanır.

Kuyruk paylaşım grubu ve kuyruk yöneticisi düzeyinde güvenlik birleşimi kullanabilirsiniz. Örneğin, bir kuyruk yöneticisine özgü tanıtımları, ait olduğu kuyruk paylaşım grubunun tanıtımlarını geçersiz kılacak şekilde düzenleyebilirsiniz.

*Anahtar tanıtımları*tanımlanarak altsistem güvenliği, kuyruk paylaşımı grup düzeyi güvenliği ve kuyruk yöneticisi düzeyinde güvenlik açılır ya da kapanır. Anahtar profili, IBM MQiçin özel bir anlamı olan normal bir RACF profilidir.

z/OS *z/OS üzerinde komut güvenliği ve komut kaynağı güvenliği*

Komut güvenliği, bir komut verme yetkisiyle ilgilidir; komut kaynağı yetkisi, bir kaynak üzerinde işlem gerçekleştirme yetkisiyle ilgilidir. Her ikisi de RACF sınıfları kullanılarak gerçekleştirilir.

IBM MQ yöneticisi bir MQSC komutu yayınladığında yetki denetimleri gerçekleştirilir. Buna *komut güvenliği*denir.

Komut güvenliğini uygulamak için, belirli RACF tanıtımlarını tanımlamalı ve gerekli düzeylerde bu tanıtımlara gereken gruplara ve kullanıcı kimliklerine erişim vermelisiniz. Komut güvenliğine ilişkin bir tanıtımın adı MQSC komutunun adını içerir.

Bazı MQSC komutları, yerel kuyruk yaratmak için DEFINE QLOCAL komutu gibi bir IBM MQ kaynağı üzerinde işlem gerçekleştirir. Bir denetimci bir MQSC komutu verdiğinde, komutta belirtilen kaynak üzerinde istenen işlemin gerçekleştirilip gerçekleştirilemeyeceğini saptamak için yetki denetimleri gerçekleştirilir. Buna *komut kaynak güvenliği*denir.

Komut kaynağı güvenliğini uygulamak için, belirli RACF tanıtımlarını tanımlamalı ve gerekli düzeylerde bu tanıtımlara gereken gruplara ve kullanıcı kimliklerine erişim vermelisiniz. Komut kaynağı güvenliğine ilişkin bir tanıtımın adı, bir IBM MQ kaynağının adını ve tipini (QUEUE, PROCESS, NAMELIST, TOPIC, AUTHINFO ya da CHANNEL) içerir.

Komut güvenliği ve komut kaynağı güvenliği bağımsızdır. Örneğin, bir yönetici şu komutu yayınladığında:

```
DEFINE QLOCAL(MOON.EUROPA)
```

Aşağıdaki yetki denetimleri gerçekleştirilir:

- Komut güvenliği, denetimcinin DEFINE QLOCAL komutunu verme yetkisi olup olmadığını denetler.
- Komut kaynağı güvenliği, denetimcinin MOON.EUROPA.

Anahtar tanıtımları tanımlanarak komut güvenliği ve komut kaynağı güvenliği açılabilir ya da kapatılabilir.

z/OS *z/OS üzerinde MQSC komutları ve sistem komutu giriş kuyruğu*

Komut sunucusunun, z/OSüzerindeki sistem komut giriş kuyruğuna yönlendirilen MQSC komutlarını nasıl işlediğini anlamak için bu konuyu kullanın.

Komut güvenliği ve komut kaynağı güvenliği, komut sunucusu sistem komut giriş kuyruğundan MQSC komutu içeren bir ileti aldığında da kullanılır.Yetki denetimleri için kullanılan kullanıcı kimliği, MQSC komutunu içeren iletinin ileti tanımlayıcısındaki *UserIdentifier* alanında bulunan kullanıcı kimliğidir. Bu kullanıcı kimliğinin, komutun işlendiği kuyruk yöneticisinde gerekli yetkileri olmalıdır. *UserIdentifier* alanıyla ve nasıl ayarlandığıyla ilgili daha fazla bilgi için İleti bağlamıbaşlıklı konuya bakın.

MQSC komutlarını içeren iletiler, aşağıdaki durumlarda sistem komutu giriş kuyruğuna gönderilir:

- İşlemler ve denetim panoları, MQSC komutlarını hedef kuyruk yöneticisinin sistem komut giriş kuyruğuna gönderir. MQSC komutları, panolarda seçtiğiniz işlemlere karşılık gelir. Her iletideki *UserIdentifier* alanı, yöneticinin TSO kullanıcı kimliğine ayarlanır.
- IBM MQ yardımcı programının COMMAND işlevi olan CSQUTIL, giriş verileri kümesindeki MQSC komutlarını hedef kuyruk yöneticisinin sistem komutu giriş kuyruğuna gönderir. COPY ve EMPTY işlevleri, DISPLAY QUEUE ve DISPLAY STGCLASS komutlarını gönderir. Her iletideki *UserIdentifier* alanı, iş kullanıcı kimliğine ayarlanır.
- CSQINPX veri kümelerindeki MQSC komutları, kanal başlatıcısının bağlı olduğu kuyruk yöneticisinin sistem komut giriş kuyruğuna gönderilir. Her iletideki *UserIdentifier* alanı, kanal başlatıcı adres alanı kullanıcı kimliğine ayarlanır.

  CSQINP1 ve CSQINP2 veri kümelerinden MQSC komutları verildiğinde yetki denetimi gerçekleştirilmez. RACF veri kümesi korumasını kullanarak bu veri kümelerini kimlerin güncellemesine izin verilmesini denetleyebilirsiniz.
- Bir kuyruk paylaşım grubu içinde, kanal başlatıcı, bağlı olduğu kuyruk yöneticisinin sistem komut giriş kuyruğuna START CHANNEL komutları gönderebilir. Komut, paylaşılan iletim kuyruğunu kullanan bir

giden kanal tetiklenerek başlatıldığında gönderilir. Her iletideki *UserIdentifier* alanı, kanal başlatıcı adres alanı kullanıcı kimliğine ayarlanır.

- Bir uygulama, bir sistem komutu giriş kuyruğuna MQSC komutları gönderebilir. Varsayılan olarak, her iletideki *UserIdentifier* alanı, uygulamayla ilişkili kullanıcı kimliğine ayarlanır.
- AIX, Linux, and Windows sistemlerinde **runmqsc** denetim komutu, MQSC komutlarını z/OSüzerindeki bir kuyruk yöneticisinin sistem komut giriş kuyruğuna göndermek için dolaylı kipte kullanılabilir. Her iletideki *UserIdentifier* alanı, **runmqsc** komutunu yayınlayan yöneticinin kullanıcı kimliğine ayarlanır.

z/OS *z/OS üzerindeki kuyruk yöneticisi veri kümelerine erişim*

IBM MQ for z/OS yöneticilerinin kuyruk yöneticisi veri kümelerine erişmek için yetkilerine sahip olması gerekir. Hangi veri kümelerinin RACF koruması gerektiğini anlamak için bu konuyu kullanın.

Bu veri kümeleri şunları içerir:

- Kuyruk yöneticisinin başlatılan görev yordamında CSQINP1, CSQINP2ve CSQINPT tarafından gönderme yapılan veri kümeleri.
- Kuyruk yöneticisinin sayfa kümeleri, etkin günlük veri kümeleri, arşiv günlüğü veri kümeleri ve önyükleme veri kümeleri (BSDSs)
- Kanal başlatıcının başlatılan görev yordamında CSQXLIB ve CSQINPX tarafından başvurulan veri kümeleri

Yetkisiz bir kullanıcının kuyruk yöneticisini başlatamaması ya da herhangi bir kuyruk yöneticisi verisine erişmesi için veri kümelerini korumalısınız. Bunu yapmak için RACF veri kümesi korumasını kullanın.

Uygulamaların IBM MQ ' yi kullanması için yetki

Uygulamalar nesnelere eriştiğinde, uygulamalarla ilişkili kullanıcı kimlikleri için uygun yetki gerekir.

Uygulamalar, MQI çağrıları yayınlayarak aşağıdaki IBM MQ nesnelerine erişebilir:

- Kuyruk yöneticileri
- Kuyruklar
- Süreçler
- Namelistler
- Konular

Uygulamalar, IBM MQ nesnelerini denetlemek için PCF komutlarını da kullanabilir. PCF komutu işlendiğinde, PCF iletisini koyan kullanıcı kimliğinin yetki bağlamını kullanır.

Bu bağlamda uygulamalar, kullanıcılar ve satıcı firmalar tarafından yazılanları ve IBM MQ for z/OSile birlikte sağlananları içerir. IBM MQ for z/OS ile sağlanan uygulamalar şunlardır:

- İşlemler ve denetim panoları
- IBM MQ yardımcı programı, CSQUTIL
- Ölü harf kuyruğu işleyici yardımcı programı, CSQUDLQH

IBM MQ classes for Java, IBM MQ classes for JMS, IBM MQ classes for .NETya da Message Service Clients for C/C++ and .NET kullanan uygulamalar MQI ' ı dolaylı olarak kullanır.

MCA 'lar MQI çağrıları da yayınlıyor ve MCA' larla ilişkili kullanıcı kimliklerinin bu IBM MQ nesnelerine erişmek için yetkisi gerekiyor. Bu kullanıcı kimlikleri ve gerekli yetkiler hakkında daha fazla bilgi için bkz. “Kanal yetkilendirmesi” sayfa 112.

z/OSüzerinde, uygulamalar bu IBM MQ nesnelerine erişmek için MQSC komutlarını kullanabilir, ancak komut güvenliği ve komut kaynağı güvenliği bu koşullarda yetki denetimlerini sağlar. z/OS Daha fazla bilgi için bkz. “z/OS üzerinde komut güvenliği ve komut kaynağı güvenliği” sayfa 91 ve “z/OS üzerinde MQSC komutları ve sistem komutu giriş kuyruğu” sayfa 91.

IBM iüzerinde, Grup 2 'de CL komutu veren bir kullanıcı, komutla ilişkili bir IBM MQ nesnesine erişmek için yetki gerektirebilir. Daha fazla bilgi için “Yetki denetimleri gerçekleştirildiğinde” sayfa 93 başlıklı konuya bakın.

Yetki denetimleri gerçekleştirildiğinde

Bir uygulama bir kuyruk yöneticisine, kuyruğa, işleme ya da ad belirlemeye erişmeye çalıştığında yetki denetimleri gerçekleştirilir.

IBM iüzerinde, bir kullanıcı Grup 2 'de bu IBM MQ nesnelerinden herhangi birine erişen bir CL komutu verdiğinde de yetki denetimleri gerçekleştirilebilir. Denetimler aşağıdaki koşullarda gerçekleştirilir:

Bir uygulama `MQCONN` ya da `MQCONNX` çağrısıyla bir kuyruk yöneticisine bağlandığında
Kuyruk yöneticisi, işletim sisteminden uygulamayla ilişkili kullanıcı kimliğini ister. Kuyruk yöneticisi daha sonra, kullanıcı kimliğinin bu kullanıcı kimliğine bağlanma yetkisi olup olmadığını denetler ve ileride yapılacak denetimler için kullanıcı kimliğini korur.

Kullanıcıların IBM MQ' da oturum açmalarına gerek yoktur. IBM MQ , kullanıcıların temel işletim sisteminde oturum açtılarını ve bu işletim sistemi tarafından doğrulandığını varsayar.

Bir uygulama `MQOPEN` ya da `MQPUT1` çağrısı kullanarak bir IBM MQ nesnesini açtığında
Bir nesne açıldığında, daha sonra erişildiğinde değil, tüm yetki denetimleri gerçekleştirilir. Örneğin, uygulama bir kuyruğu açtığında yetki denetimleri gerçekleştirilir. Bu iletiler, uygulama kuyruğa ileti koyduğunda ya da kuyruktan ileti aldığında gerçekleştirilmez.

Bir uygulama bir nesneyi açtığında, nesne üzerinde gerçekleştirmesi gereken işlem tiplerini belirtir. Örneğin, bir uygulama üzerindeki iletilere göz atmak, ondan ileti almak, ancak üzerine ileti koymamak için bir kuyruğu açabilir. Kuyruk yöneticisi, her işlem tipi için, uygulamayla ilişkili kullanıcı kimliğinin o işlemi gerçekleştirme yetkisi olup olmadığını denetler.

Bir uygulama bir kuyruğu açtığında, nesne tanımlayıcısının `ObjectName` alanında adı belirtilen nesne için yetki denetimleri gerçekleştirilir. `ObjectName` alanı, `MQOPEN` ya da `MQPUT1` çağrılarında kullanılır. Nesne bir diğer ad kuyruğu ya da uzak kuyruk tanımlamasıysa, yetki denetimleri nesnenin kendisine karşı gerçekleştirilir. Bunlar, diğer ad kuyruğunun ya da uzak kuyruk tanımlamasının çözüldüğü kuyrukta gerçekleştirilmez. Bu, kullanıcının buna erişmek için izne ihtiyacı olmadığı anlamına gelir. Ayrıcalıklı kullanıcılar için kuyruk yaratma yetkisini sınırlayın. Bunu yapmazsanız, kullanıcılar bir diğer ad yaratarak normal erişim denetimini atlayabilir.

Bir uygulama uzak bir kuyruğa belirtik olarak gönderme yapabilir. Nesne tanımlayıcıdaki `ObjectName` ve `ObjectQMgrAd` alanlarını uzak kuyruk ve uzak kuyruk yöneticisinin adlarına ayarlar. Uzak kuyruk yöneticisiyle aynı adı taşıyan iletim kuyruğuna yönelik yetki denetimleri gerçekleştirilir:

- z/OS z/OSişletim sistemi üzerinde, RACF kuyruk tanıtımında uzak kuyruk yöneticisi adıyla eşleşen bir denetim yapılır ve bu iletim kuyruğunun yerel olarak tanımlanıp tanımlanmadığı denetlenir.
- Multi Çoklu platformlarüzerinde, kümeleme kullanılıyorsa, uzak kuyruk yöneticisi adıyla eşleşen RQMNAME tanıtımıyla ilgili bir denetim yapılır.

Bir uygulama, nesne tanımlayıcıdaki `ObjectName` alanını küme kuyruğunun adına ayarlayarak küme kuyruğuna belirtik olarak gönderme yapabilir. Yetki denetimleri küme iletim kuyruğuna (`SYSTEM.CLUSTER.TRANSMIT.QUEUE`) göre gerçekleştirilir.

Dinamik bir kuyruk için yetki, türetildiği model kuyruğuna dayalıdır, ancak mutlaka aynı değildir; bkz. not 1.

Kuyruk yöneticisinin yetki denetimleri için kullandığı kullanıcı kimliği, işletim sisteminden alınır. Kullanıcı kimliği, uygulama kuyruk yöneticisine bağlandığında alınır. Uygun yetkili bir uygulama, alternatif bir kullanıcı kimliği belirterek bir `MQOPEN` çağrısı yapabilir; daha sonra diğer kullanıcı kimliğiyle erişim denetimi denetimleri yapılır. Alternatif bir kullanıcı kimliği kullanılması, uygulamayla ilişkili kullanıcı kimliğini değiştirmez, yalnızca erişim denetimi denetimleri için kullanılan kullanıcı kimliğini değiştirir.

Bir uygulama MQSUB çağrısıyla bir konuya abone olduğunda
Bir uygulama bir konuya abone olduğunda, gerçekleştirmesi gereken işlem tipini belirtir. Bir abonelik oluşturuyor, var olan bir aboneliği değiştiriyor ya da var olan bir aboneliği değiştirmeden sürdürüyor. Kuyruk yöneticisi, her işlem tipi için, uygulamayla ilişkili kullanıcı kimliğinin işlemi gerçekleştirme yetkisi olup olmadığını denetler.

Bir uygulama bir konuya abone olduğunda, konu ağacında bulunan konu nesnelerine ilişkin yetki denetimleri gerçekleştirilir. Konu nesneleri, uygulamanın abone olduğu konu ağacında ya da üstünde yer alır. Yetki denetimleri birden çok konu nesnesi üzerinde denetimler içerebilir. Kuyruk yöneticisinin yetki denetimleri için kullandığı kullanıcı kimliği, işletim sisteminden alınır. Kullanıcı kimliği, uygulama kuyruk yöneticisine bağlandığında alınır.

Kuyruk yöneticisi, yönetilen kuyruklarda değil, abone kuyruklarında yetki denetimi gerçekleştirir.

Bir uygulama MQCLOSE çağrısıyla kalıcı bir dinamik kuyruğu sildiğinde
MQCLOSE çağrısında belirtilen nesne tanıtıcısı, kalıcı dinamik kuyruğu yaratan MQOPEN çağrısıyla döndürülen nesne tanıtıcısıyla aynı olmayabilir. Farklıysa, kuyruk yöneticisi MQCLOSE çağrısını yayınlayan uygulamayla ilişkili kullanıcı kimliğini denetler. Kullanıcı kimliğinin kuyruğu silme yetkisi olup olmadığını denetler.

Kaldırmak üzere bir aboneliği kapatan bir uygulama yaratmadığında, kaldırmak için uygun yetkinin kullanılması gerekir.

Bir IBM MQ nesnesi üzerinde çalışan bir PCF komutu komut sunucusu tarafından işlendiğinde
Bu kural, bir PCF komutunun bir kimlik doğrulama bilgi nesnesi üzerinde çalıştığı durumu içerir.

Yetki denetimleri için kullanılan kullanıcı kimliği, PCF komutunun ileti tanımlayıcısındaki UserIdentifier alanında bulunan kullanıcı kimliğidir. Bu kullanıcı kimliğinin, komutun işlendiği kuyruk yöneticisinde gerekli yetkileri olmalıdır. Bir Escape PCF komutunda kapsüllenmiş eşdeğer MQSC komutu da aynı şekilde işlenir. UserIdentifier alanı ve nasıl ayarlandığı hakkında daha fazla bilgi için bkz. “İleti bağlamı” sayfa 95.

IBM i **IBM iüzerinde, bir kullanıcı IBM MQ nesnesi üzerinde çalışan Grup 2 'de bir CL komutu yayınladığında**
Bu kural, Grup 2 'deki bir CL komutunun kimlik doğrulama bilgileri nesnesi üzerinde çalıştığı durumu içerir.

Kullanıcının komutla ilişkili bir IBM MQ nesnesi üzerinde çalışma yetkisi olup olmadığını belirlemek için denetimler gerçekleştirilir. Denetimler, kullanıcı QMQMADM grubunun bir üyesi değilse ya da *ALLOBJ yetkisi yoksa gerçekleştirilir. Gereken yetki, komutun nesne üzerinde gerçekleştirdiği işlemin tipine bağlıdır. Örneğin, **CHGMQMQ**(MQM kuyruğunu değiştir) komutu, komutun belirttiği kuyruğun özniteliklerini değiştirme yetkisini gerektirir. Buna karşılık, **DSPMQMQ**(MQM Kuyruğunu Görüntüle) komutu, komutun belirttiği kuyruğun özniteliklerini görüntüleme yetkisini gerektirir.

Birçok komut birden çok nesne üzerinde çalışır. Örneğin, **DLTMQMQ**(MQM Kuyruğunu Sil) komutunu vermek için aşağıdaki yetkiler gereklidir:

- Komutun belirlediği kuyruk yöneticisine bağlanma yetkisi
- Komutun belirlediği kuyruğu silme yetkisi

Bazı komutlar hiçbir nesne üzerinde çalışmaz. Bu durumda, kullanıcının bu komutlardan birini yayınlamak için yalnızca IBM i yetkisi gerekir. **STRMQMLSR**, MQM Dinleyicisini Başlat, böyle bir komutun bir örneğidir.

Diğer kullanıcı yetkisi

Bir uygulama bir nesneyi açtığında ya da bir konuya abone olduğunda, uygulama MQOPEN, MQPUT1ya da MQSUB çağrısında bir kullanıcı kimliği sağlayabilir. Kuyruk yöneticisinden, uygulamayla ilişkili yetki denetimleri yerine bu kullanıcı kimliğini kullanmasını isteyebilir.

Uygulama, yalnızca aşağıdaki koşulların her ikisi de karşılandıysa nesneyi açmayı başarır:

- Uygulamayla ilişkilendirilen kullanıcı kimliği, yetki denetimleri için farklı bir kullanıcı kimliği sağlama yetkisine sahiptir. Uygulamanın *diğer kullanıcı yetkisine*sahip olduğu söyilir.

- Uygulama tarafından sağlanan kullanıcı kimliği, istenen işlem tipleri için nesneyi açma ya da konuya abone olma yetkisine sahiptir.

İleti bağlamı

İleti bağlamı bilgileri, bir iletiyi alan uygulamanın iletiyi başlatan hakkında bilgi edinmesini sağlar. Bilgi, ileti tanımlayıcısındaki alanlarda tutulur ve alanlar üç mantıksal bölüme ayrılır.

Bu parçalar aşağıdaki gibidir:

kimlik bağlamı
Bu alanlar, iletiyi kuyruğa koyan uygulama kullanıcısına ilişkin bilgileri içerir.

kaynak bağlam
Bu alanlar, uygulamanın kendisiyle ve iletinin kuyruğa ne zaman konduğuyla ilgili bilgi içerir.

kullanıcı bağlamı
Bu alanlar, uygulamaların kuyruk yöneticisinin teslim etmesi gereken iletileri seçmek için kullanabileceği ileti özelliklerini içerir.

Bir uygulama bir iletiyi kuyruğa koyduğunda, uygulama kuyruk yöneticisinden iletideki bağlam bilgilerini oluşturmasını isteyebilir. Varsayılan işlem budur. Alternatif olarak, bağlam alanlarının bilgi içermediğini belirtebilir. Bir uygulamayla ilişkilendirilen kullanıcı kimliği, bunların ikisini de yapmak için özel bir yetki gerektirmez.

Bir uygulama, bir iletideki kimlik bağlamı alanlarını ayarlayarak kuyruk yöneticisinin kaynak bağlamı oluşturmasına izin verebilir ya da tüm bağlam alanlarını ayarlayabilir. Bir uygulama ayrıca, kimlik bağlamı alanlarını, alındığı bir iletiden bir kuyruğa koyuyor olduğu bir iletiye geçirebilir ya da tüm bağlam alanlarını geçirebilir. Ancak, bir uygulamayla ilişkili kullanıcı kimliği, bağlam bilgilerini ayarlama ya da iletme yetkisi gerektirir. Bir uygulama, iletileri koymak üzere olduğu kuyruğu açtığında bağlam bilgilerini ayarlamak ya da iletmek istediğini belirtir ve şu anda yetkisi denetlenir.

Aşağıda, bağlam alanlarının her birinin kısa bir açıklaması yer almaktadır:

Kimlik bağlamı

UserIdentifier
İletiyi koyan uygulamayla ilişkili kullanıcı kimliği. Kuyruk yöneticisi bu alanı ayarlarsa, uygulama kuyruk yöneticisine bağlandığında işletim sisteminden alınan kullanıcı kimliğine ayarlanır.

AccountingToken
İletinin sonucu olarak yapılan iş için ücret almak üzere kullanılabilecek bilgiler.

ApplIdentityVerileri
Bir uygulamayla ilişkili kullanıcı kimliğinin kimlik bağlamı alanlarını ayarlama ya da tüm bağlam alanlarını ayarlama yetkisi varsa, uygulama bu alanı kimlikle ilgili herhangi bir değere ayarlayabilir. Kuyruk yöneticisi bu alanı ayarlarsa, bu alan boş olarak ayarlanır.

Kaynak bağlam

PutApplTipi
İletiyi koyan uygulamanın tipi; örneğin, CICS işlemi.

PutApplAdı
İletiyi koyan uygulamanın adı.

PutDate
İletinin konma tarihi.

PutTime
Mesajın gönderildiği zaman.

ApplOriginVerileri
Bir uygulamayla ilişkili kullanıcı kimliğinin tüm bağlam alanlarını ayarlama yetkisi varsa, uygulama bu alanı kaynakla ilgili herhangi bir değere ayarlayabilir. Kuyruk yöneticisi bu alanı ayarlarsa, bu alan boş olarak ayarlanır.

Kullanıcı bağlamı

MQINQMP ya da **MQSETMP**için aşağıdaki değerler desteklenir:

MQPD_KULLANıCı_BAğLAMı

Özellik, kullanıcı bağlamıyla ilişkilendirildi.

MQSETMP çağrısını kullanarak kullanıcı bağlamıyla ilişkili bir özellik ayarlayabilmek için özel bir yetkilendirme gerekmez.

Bir V7.0 ya da sonraki kuyruk yöneticisinde, kullanıcı bağlamıyla ilişkili bir özellik MQOO_SAVE_ALL_CONTEXT için açıklandığı gibi saklanır. MQOO_PASS_ALL_CONTEXT belirtilmiş bir MQPUT, özelliğin kaydedilen bağlamdan yeni iletiye kopyalanmasına neden oluyor.

MQPD_NO_CONTEXT

Özellik bir ileti bağlamıyla ilişkilendirilmemiş.

Tanınmayan bir değer MQRC_PD_ERROR ile reddedildi. Bu alanın ilk değeri şudur: **MQPD_NO_CONTEXT**.

Bağlam alanlarının her birine ilişkin ayrıntılı açıklama için MQMD-Message descriptorbaşlıklı konuya bakın. İleti bağlamının nasıl kullanılacağı hakkında daha fazla bilgi için bkz. İleti bağlamı.

IBM i ALW IBM i *IBM i , AIX, Linux, and Windows sistemlerinde IBM MQ nesneleriyle çalışma yetkisi*

IBM MQ ile sağlanan yetkilendirme hizmeti bileşenine *nesne yetki yöneticisi* (OAM) adı verilir. Kimlik doğrulama ve yetkilendirme denetimleri aracılığıyla erişim denetimi sağlar.

Kimlik doğrulama.

IBM MQ ile birlikte sağlanan OAM tarafından gerçekleştirilen kimlik doğrulama denetimi temeldir ve yalnızca belirli koşullarda gerçekleştirilir. Yüksek güvenlikli bir ortamda beklenen katı gereksinimlerin karşılanması amaçlanmamıştır.

OAM, bir uygulama bir kuyruk yöneticisine bağlandığında kimlik doğrulama denetimini gerçekleştirir ve aşağıdaki koşullar geçerlidir:

- Bağlantı kuran uygulama tarafından bir MQCSP yapısı sağlandıysa, ve
- MQCSP yapısındaki *AuthenticationType* özniteliğine MQCSP_AUTH_USER_ID_AND_PWD değeri verilir ve
- Konfigürasyonu tanımlanmış AUTHINFO nesnesindeki CHCKLOCL ya da CHKCCLNT değeri 'NONE' değil

OAM ' daki kimlik doğrulama adımları, kullanıcı adının çok fazla yanlış parola testi denemesi olmadığından emin olmak gibi ek denetimler gerçekleştirecek şekilde yapılandırılmış olan işletim sistemi hizmetlerini kullanarak parolayı doğrular.

Yeni bir yetkilendirme hizmeti bileşeni yazarsanız ya da bir satıcıdan bir bileşen edinirseniz, alternatif kimlik doğrulama mekanizmaları kullanılabilir.

Yetki belgesi.

Yetki denetimleri kapsamlidir ve en normal gereksinimleri karşılaması amaçlanmıştır.

Bir uygulama bir kuyruk yöneticisine, kuyruğa, işleme, konuya ya da ad çizelgelerine erişmek için MQI çağrısı verdiğinde yetkilendirme denetimleri gerçekleştirilir. Bunlar, örneğin, bir komut Komut Sunucusu tarafından gerçekleştirilirken başka zamanlarda da gerçekleştirilir.

IBM i IBM i , AIX, Linux, and Windows sistemlerinde *yetkilendirme hizmeti* , bir uygulama kuyruk yöneticisi, kuyruk, süreç, konu ya da ad melist olan bir IBM MQ nesnesine erişmek için MQI çağrısı yaptığında erişim denetimi sağlar. Bu, alternatif kullanıcı yetkisi denetimlerini ve bağlam bilgilerini belirleme ya da iletme yetkisini içerir.

Windows Windows üzerinde OAM, Yöneticiler grubunun üyelerine UAC etkinleştirildiğinde bile tüm IBM MQ nesnelerine erişim yetkisi verir. Ayrıca, Windows sistemlerinde SYSTEM hesabının IBM MQ kaynaklarına tam erişimi vardır.

Yetki hizmeti, bir PCF komutu bu IBM MQ nesnelerinden birinde ya da bir kimlik doğrulama bilgileri nesnesinde çalışırken de yetki denetimi sağlar. Bir Escape PCF komutunda kapsüllenmiş eşdeğer MQSC komutu da aynı şekilde işlenir.

IBM i IBM i ' de, kullanıcı QMQMADM grubunun bir üyesi olmadığı ya da *ALLOBJ yetkisi olmadığı sürece, yetkilendirme hizmeti, bir kullanıcının Grup 2 'de bu IBM MQ nesnelerinin ya da kimlik doğrulama bilgileri nesnelerinin herhangi birinde çalışan bir CL komutu verdiğinde de yetki denetimi sağlar.

Yetkilendirme hizmeti, *kurulabilir bir hizmettir*; bu, hizmetin bir ya da daha çok *kurulabilir hizmet bileşeni*tarafından gerçekleştirildiği anlamına gelir. Her bileşen belgelenmiş bir arabirim kullanılarak çağrılır. Bu, kullanıcıların ve satıcı firmaların IBM MQ ürünleri tarafından sağlananları genişletmek ya da değiştirmek için bileşenler sağlamalarına olanak sağlar.

IBM MQ ile sağlanan yetkilendirme hizmeti bileşeni, nesne yetki yöneticisi (OAM) olarak adlandırılır. OAM, yarattığınız her kuyruk yöneticisi için otomatik olarak etkinleştirilir.

OAM, erişimi denetlediği her IBM MQ nesnesi için bir erişim denetimi listesi (ACL) sağlar. AIX and Linux sistemlerinde yalnızca grup tanıtıcıları bir EDL ' de görüntülenebilir. Bu, bir grubun tüm üyelerinin aynı yetkilere sahip olduğu anlamına gelir. IBM i IBM i ve Windows sistemlerinde, kullanıcı kimlikleri ve grup kimlikleri bir EDL ' de görüntülenebilir. Bu, yetkilerin bireysel kullanıcılara ve gruplara verilebileceği anlamına gelir.

12 karakterlik bir sınırlama hem grup hem de kullanıcı kimliği için geçerlidir. UNIX altyapıları genellikle kullanıcı kimliğinin uzunluğunu 12 karakterle sınırlar. AIX ve Linux bu sınırı artırdı, ancak IBM MQ tüm UNIX platformlarında 12 karakterlik bir kısıtlama gözlemlemeye devam ediyor. 12 karakterden uzun bir kullanıcı kimliği kullanırsanız, IBM MQ bunu "UNKNOWN"(Bilinmiyor) değeriyle değiştirir. "UNKNOWN"(Bilinmiyor) değeriyle bir kullanıcı kimliği tanımlamayın.

OAM, bir kullanıcının kimliğini doğrulayabilir ve uygun kimlik bağlamı alanlarını değiştirebilir. Bunu etkinleştirmek için, MQCONNX çağrısında bir bağlantı güvenliği değiştirgeleri yapısı (MQCSP) belirtin. Yapı, uygun kimlik bağlamı alanlarını ayarlayan OAM Kullanıcı Kimliği Doğrula işlevine (MQZ_AUTHENTICATE_USER) aktarılır. IBM MQ istemcisinden bir MQCONNX bağlantısı varsa, MQCSP ' deki bilgiler istemcinin istemci bağlantısı ve sunucu bağlantısı kanalı üzerinden bağlandığı kuyruk yöneticisine gönderilir. Bu kanalda güvenlik çıkışları tanımlanırsa, MQCSP her güvenlik çıkışına geçirilir ve çıkış tarafından değiştirilebilir. Güvenlik çıkışları MQCSP ' yi de yaratabilir. Bu bağlamda güvenlik çıkışlarının kullanımına ilişkin daha fazla ayrıntı için bkz. Kanal güvenliği çıkış programları.

Uyarı: Bazı durumlarda, bir istemci uygulamasına ilişkin MQCSP yapısındaki parola ağ üzerinden düz metin olarak gönderilir. İstemci uygulaması parolalarının uygun şekilde korunduğundan emin olmak için bkz. IBM MQCSP parola koruması.

AIX, Linux, and Windows sistemlerinde, **setmqaut** denetim komutu yetkileri verir ve geri alır ve EDL ' leri korumak için kullanılır. Örneğin, komut:

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER +browse +get
```

VOYAGER grubunun üyelerinin MOON.EUROPA . Bu, üyelerin kuyruktan da ileti almalarını sağlar. Bu yetkileri daha sonra iptal etmek için aşağıdaki komutu girin:

```
setmqaut -m JUPITER -t queue -n MOON.EUROPA -g VOYAGER -browse -get
```

Komut:

```
setmqaut -m JUPITER -t queue -n MOON.* -g VOYAGER +put
```

VOYAGER grubunun üyelerinin, `MOON.` karakterleriyle başlayan bir adla herhangi bir kuyruğa ileti koymalarını sağlar. MOON.* soysal bir tanıtımın adıdır.*Soysal tanıtım* , tek bir **setmqaut** komutunu kullanarak bir nesne kümesi için yetki vermenizi sağlar.

dspmqaut denetim komutu, bir kullanıcının ya da grubun belirli bir nesne üzerindeki yürürlükteki yetkilerini görüntülemek için kullanılabilir.**dmpmqaut** denetim komutu, soysal tanıtımlarla ilişkili yürürlükteki yetkileri görüntülemek için de kullanılabilir.

IBM i IBM i' ta, bir denetimci yetkilere yetki vermek için GRTMQMAUT Denetim dili (CL) komutunu ve yetkilerini iptal etmek için RVKMQMAUT Denetim dili (CL) komutunu kullanır. Soysal tanıtımlar da kullanılabilir.Örneğin, CL komutu:

```
GRTMQMAUT MQMNAME(JUPITER) OBJTYPE(*Q) OBJ('MOON.*') USER(VOYAGER) AUT(*PUT)
```

setmqaut komutunun önceki örneğiyle aynı işlevi sağlar; bu işlev, VOYAGER grubunun üyelerinin `MOON.` karakterleriyle başlayan bir adla herhangi bir kuyruğa ileti koymalarını sağlar.

IBM i CL komutu DSPMQMAUT, kullanıcının ya da grubun belirtilen bir nesne için sahip olduğu yürürlükteki yetkileri görüntüler. WRKMQMAUT ve WRKMQMAUTD CL komutları, nesnelerle ve soysal tanıtımlarla ilişkili yürürlükteki yetkililerle çalışmak için de kullanılabilir.

Örneğin, bir test ortamında herhangi bir yetki denetimi istemiyorsanız, OAM ' yi devre dışı bırakabilirsiniz.

Multi *OAM komutlarına erişmek için PCF kullanılması*
IBM i, AIX, Linux, and Windows sistemlerinde, OAM yönetim komutlarına erişmek için PCF komutlarını kullanabilirsiniz.

PCF komutları ve eşdeğer OAM komutları aşağıdaki gibidir:

Çizelge 8. PCF komutları ve eşdeğer OAM komutları

PCF komutu	**OAM komutu**
Yetki Kayıtlarını Sor	dmpmqaut
Varlık Yetkilisi 'ni sorgularken	mqaut
Yetki Kaydını Ayarla	setmqaut
Yetki Kaydını Sil	-remove seçeneğiyle setmqaut

setmqaut ve **dmpmqaut** komutları, mqm grubunun üyeleriyle sınırlıdır. Eşdeğer PCF komutları, kuyruk yöneticisinde dsp ve chg yetkileri verilmiş olan herhangi bir gruptaki kullanıcılar tarafından yürütülebilir.

Bu komutların kullanılmasıyla ilgili ek bilgi için Introduction to Programmable Command Formatsbaşlıklı konuya bakın.

z/OS *z/OS üzerindeki IBM MQ nesneleriyle çalışma yetkisi*

z/OSüzerinde, MQI 'a yapılan çağrılarla ilişkili yedi yetki denetimi kategorisi vardır. Belirli RACF profillerini tanımlamalı ve bu profillere uygun erişim vermelisiniz. Kaç kullanıcı kimliği denetleneceğini denetlemek için *RESLEVEL* tanıtımını kullanın.

MQI çağrılarıyla ilişkili yedi yetki denetimi kategorisi:

Bağlantı güvenliği
Uygulama bir kuyruk yöneticisine bağlandığında gerçekleştirilen yetki denetimleri

Kuyruk güvenliği
Bir uygulama bir kuyruğu açtığında ya da kalıcı bir dinamik kuyruğu sildiğinde gerçekleştirilen yetki denetimleri

Süreç güvenliği
Uygulama bir süreç nesnesini açtığında gerçekleştirilen yetki denetimleri

Namelist güvenliği
Uygulama bir ad listesi nesnesini açtığında gerçekleştirilen yetki denetimleri

Diğer kullanıcı güvenliği
Bir uygulama bir nesneyi açarken diğer kullanıcı yetkisi istediğinde gerçekleştirilen yetki denetimleri

Bağlam güvenliği
Yetki, bir uygulama bir kuyruğu açtığında gerçekleştirilen işlemleri denetler ve kuyruğa koyduğu iletilerde bağlam bilgilerini ayarlayıp aktarmayı amaçladığını belirtir.

Konu güvenliği
Uygulama bir konuyu açtığında gerçekleştirilen yetki denetimleri

Her yetki denetimi kategorisi, komut güvenliği ve komut kaynağı güvenliğinin uygulandığı şekilde gerçekleştirilir. Belirli RACF tanıtımlarını tanımlamalı ve gerekli düzeylerde bu tanıtımlara erişim yetkisi vermelisiniz. Kuyruk güvenliği için erişim düzeyi, uygulamanın bir kuyrukta gerçekleştirebileceği işlem tiplerini belirler. Bağlam güvenliği için erişim düzeyi, uygulamanın aşağıdakileri yapıp yapabileceğini belirler:

- Tüm bağlam alanlarını geçir
- Tüm bağlam alanlarını geçir ve kimlik bağlamı alanlarını ayarla
- Tüm bağlam alanlarını geçir ve ayarla

Her yetki denetimi kategorisi, anahtar tanıtımları tanımlanarak açılabilir ya da kapatılabilir.

Bağlantı güvenliği dışında tüm kategoriler toplu olarak *API-kaynak güvenliği*olarak bilinir.

Varsayılan olarak, toplu bağlantı kullanan bir uygulamadan gelen bir MQI çağrısının sonucu olarak bir API kaynağı güvenlik denetimi gerçekleştirildiğinde, yalnızca bir kullanıcı kimliği denetlenir. Bir CICS ya da IMS uygulamasından ya da kanal başlatıcısından MQI çağrısının sonucu olarak bir denetim gerçekleştirildiğinde, iki kullanıcı kimliği denetlenir.

Ancak bir *RESLEVEL tanıtımı*tanımlayarak, sıfır, bir ya da iki kullanıcı kimliğinin denetlenip denetlenmeyeceğini denetleyebilirsiniz. Denetlenen kullanıcı kimliklerinin sayısı, uygulama kuyruk yöneticisine ve kullanıcı kimliğinin RESLEVEL tanıtımına bağlandığında bağlantı tipiyle ilişkili kullanıcı kimliği tarafından belirlenir. Her bağlantı tipiyle ilişkili kullanıcı kimliği:

- Toplu bağlantılar için bağlanan görevin kullanıcı kimliği
- CICS bağlantıları için CICS adres alanı kullanıcı kimliği
- IMS bağlantıları için IMS bölge adresi alanı kullanıcı kimliği
- Kanal başlatıcı bağlantıları için kanal başlatıcı adres alanı kullanıcı kimliği

z/OSüzerindeki IBM MQ nesneleriyle çalışma yetkisi hakkında daha fazla bilgi için bkz. “z/OS üzerinde IBM MQ yönetimi yetkisi” sayfa 90.

Uzak ileti sistemi güvenliği

Bu bölümde, güvenliğin uzak ileti sistemi özellikleri ele alınmıştır.

Kullanıcılara IBM MQ olanaklarını kullanma yetkisi vermeniz gerekir. Bu, nesneler ve tanımlamalarla ilgili olarak yapılacak işlemlere göre düzenlenir. Örneğin:

- Kuyruk yöneticileri yetkili kullanıcılar tarafından başlatılabilir ve durdurulabilir
- Uygulamalar kuyruk yöneticisine bağlanmalı ve kuyrukları kullanma yetkisine sahip olmalıdır
- İleti kanallarının yetkili kullanıcılar tarafından oluşturulması ve denetlenmesi gerekir
- Nesneler kitaplıklarda tutulur ve bu kitaplıklara erişim kısıtlanabilir

Uzak bir sitedeki ileti kanalı aracısı, teslim edilmekte olan iletinin bu uzak sitede bunu yapma yetkisi olan bir kullanıcıdan kaynaklandığını denetlemelidir. Buna ek olarak, MCA 'lar uzaktan başlatılabildiğinden, MCA' larınızı başlatmaya çalışan uzak işlemlerin bunu yapma yetkisi olduğunu doğrulamak gerekebilir. Bununla başa çıkmanın dört yolu vardır:

1. RCVR, RQSTR ya da CLUSRCVR kanal tanımınızın PutAuthority özniteliğini kullanarak, gelen iletiler kuyruklarınıza konduğu sırada yetki denetimlerinde hangi kullanıcının kullanılacağını denetleyin. MQSC Command Reference 'da DEFINE CHANNEL komut tanımına bakın.

2. İstenmeyen bağlantı girişimlerini reddetmek için kanal kimlik doğrulama kayıtlarını uygulayın ya da aşağıdakilere dayalı bir MCAUSER değeri ayarlayın: Uzak IP adresi, uzak kullanıcı kimliği, sağlanan TLS Konu Ayırt Edici Adı (DN) ya da uzak kuyruk yöneticisi adı.
3. İlgili ileti kanalının yetkili olduğundan emin olmak için *kullanıcı çıkışı* güvenlik denetimini uygulayın. İlgili kanalı barındıran kuruluşun güvenliği, tek tek iletileri denetlemeniz gerekmemesi için tüm kullanıcıların düzgün bir şekilde yetkilendirilmelerini sağlar.
4. Tek tek iletilerin yetki için incelendiğinden emin olmak için *kullanıcı çıkışı* ileti işlemesini uygulayın.

IBM i *IBM MQ for IBM i nesnelerinin güvenliği*

Bu bölümde, güvenliğin uzak ileti sistemi özellikleri ele alınmıştır.

IBM MQ for IBM i olanaklarından yararlanmak için kullanıcılara yetki vermeniz gerekir. Bu yetki, nesneler ve tanımlarla ilgili olarak yapılacak işlemlere göre düzenlenir. Örneğin:

- Kuyruk yöneticileri yetkili kullanıcılar tarafından başlatılabilir ve durdurulabilir
- Uygulamaların kuyruk yöneticisine bağlanması ve kuyruklardan yararlanma yetkisi olması gerekir
- İleti kanallarının yetkili kullanıcılar tarafından oluşturulması ve denetlenmesi gerekir

Uzak bir sitedeki ileti kanalı aracısı, teslim edilmekte olan iletinin, bu uzak sitede iletiyi isue yetkisi olan bir kullanıcıdan türetilmiş olup olmadığını denetlemelidir. Buna ek olarak, MCA 'lar uzaktan başlatılabildiğinden, MCA' larınızı başlatmaya çalışan uzak işlemlerin bunu yapma yetkisi olduğunu doğrulamak gerekebilir. Bununla başa çıkmanın dört yolu vardır:

- Kanal tanımında iletilerin kabul edilebilir *bağlam* yetkisi içermesi gerektiğini, aksi takdirde atılacaklarını belirtin.
- İstenmeyen bağlantı girişimlerini reddetmek için kanal kimlik doğrulama kayıtlarını uygulayın ya da aşağıdakilerden birine dayalı olarak bir MCAUSER değeri ayarlayın: Uzak IP adresi, uzak kullanıcı kimliği, sağlanan TLS Distinguished Name (DN) ya da uzak kuyruk yöneticisi adı.
- İlgili ileti kanalının yetkili olduğundan emin olmak için kullanıcı çıkışı güvenlik denetimini uygulayın. İlgili kanalı barındıran kuruluşun güvenliği, tek tek iletileri denetlemeniz gerekmemesi için tüm kullanıcıların düzgün bir şekilde yetkilendirilmelerini sağlar.
- Bireysel iletilerin yetkilendirme için incelendiğinden emin olmak için kullanıcı çıkışı ileti işlemesini uygulayın.

IBM MQ for IBM i ' in güvenliği nasıl işlediği hakkında bazı bilgiler aşağıda verilmiştir:

- Kullanıcıların kimlikleri IBM itarafından belirlenir ve doğrulanır.
- Uygulamalar tarafından çağrılan kuyruk yöneticisi hizmetleri, kuyruk yöneticisi kullanıcı tanıtımının yetkisiyle, ancak kullanıcının işleminde çalıştırılır.
- Kullanıcı komutları tarafından çağrılan kuyruk yöneticisi hizmetleri, kuyruk yöneticisi kullanıcı tanıtımının yetkisiyle çalıştırılır.

Linux AIX *AIX and Linux üzerindeki nesnelerin güvenliği*

Bu kimlik IBM MQ denetim komutlarını kullanacaksa, denetim kullanıcıları sisteminizdeki mqm grubunun bir parçası olmalıdır (kök de içinde olmak üzere).

amqcrsta 'yı her zaman "mqm" kullanıcı kimliği olarak çalıştırmalısınız.

AIX and Linux üzerindeki kullanıcı kimlikleri

Kuyruk yöneticisi, tüm büyük harf ya da büyük harf karışık kullanıcı tanıtıcılarını küçük harfe dönüştürür. Kuyruk yöneticisi daha sonra kullanıcı tanıtıcılarını bir iletinin bağlam bölümüne ekler ya da yetkilerini denetler. Bu nedenle yetkiler yalnızca küçük harfli tanıtıcılara dayalıdır.

Windows *Windows sistemlerinde nesnelerin güvenliği*

Bu kimlik IBM MQ yönetim komutlarını kullanacaksa, yönetim kullanıcıları Windows sistemlerinde hem mqm grubunun hem de denetimciler grubunun bir parçası olmalıdır.

Windows sistemlerinde kullanıcı kimlikleri

Windows sistemlerinde *kurulu ileti çıkışı yoksa*, kuyruk yöneticisi büyük ya da karışık büyük harfli kullanıcı tanıtıcılarını küçük harfe dönüştürür. Kuyruk yöneticisi daha sonra kullanıcı tanıtıcılarını bir iletinin bağlam bölümüne ekler ya da yetkilerini denetler. Bu nedenle yetkiler yalnızca küçük harfli tanıtıcılara dayalıdır.

Sistemler arası kullanıcı kimlikleri

AIX, Linux, and Windows sistemleri dışındaki platformlar, iletilerde kullanıcı kimlikleri için büyük harfli karakterler kullanır. AIX, Linux, and Windows sistemlerinin iletilerde küçük harfli kullanıcı kimlikleri kullanmasına izin vermek için, ileti kanalı aracısı (MCA) alfabetik karakterlerin uygun dönüşümlerini gerçekleştirmelidir.

AIX, Linux, and Windows sistemlerinin iletilerde küçük harfli kullanıcı kimlikleri kullanmasına izin vermek için, bu altyapılarda ileti kanalı aracısı (MCA) tarafından aşağıdaki dönüştürmeler gerçekleştirilir:

Gönderme sonunda
Tüm kullanıcı kimliklerindeki alfabetik karakterler, ileti çıkışı kurulu değilse, büyük harfli karakterlere dönüştürülür.

Alma sonunda
Kurulu ileti çıkışı yoksa, tüm kullanıcı kimliklerindeki alfabetik karakterler küçük harfli karakterlere dönüştürülür.

Başka bir nedenle AIX, Linux, and Windows üzerinde bir ileti çıkışı sağlarsanız, otomatik dönüştürmeler gerçekleştirilmez.

Özel yetkilendirme hizmetinin kullanılması

IBM MQ , kurulabilir bir yetkilendirme hizmeti sağlar. Alternatif bir hizmet kurmayı seçebilirsiniz.

IBM MQ ile sağlanan yetkilendirme hizmeti bileşeni, Nesne Yetkilisi Yöneticisi (OAM) olarak adlandırılır. OAM, gereksinim duyduğunuz yetkilendirme olanaklarını sağlamazsa, kendi yetkilendirme hizmeti bileşeninizi yazabilirsiniz. Bir yetkilendirme hizmeti bileşeni tarafından gerçekleştirilmesi gereken kurulabilir hizmet işlevleri Kurulabilir hizmetler arabirimi başvuru bilgileribaşlığı altında açıklanmıştır.

İstemciler için erişim denetimi

Erişim denetimi kullanıcı kimliklerine dayalıdır. Denetlenecek birçok kullanıcı kimliği olabilir ve kullanıcı kimlikleri farklı biçimlerde olabilir. MCAUSER sunucu bağlantısı kanal özelliğini, istemciler tarafından kullanılmak üzere özel bir kullanıcı kimliği değerine ayarlayabilirsiniz.

IBM MQ içindeki erişim denetimi kullanıcı kimliklerine dayalıdır. MQI çağrıları yapan sürecin kullanıcı kimliği olağan durumda kullanılır. MQ MQI istemcileri için sunucu bağlantısı MCA, MQ MQI istemcileri adına MQI çağrıları yapar. Sunucu bağlantısı MCA ' nın MQI çağrıları yapmak için kullanacağı alternatif bir kullanıcı kimliği seçebilirsiniz. Diğer kullanıcı kimliği, istemci iş istasyonuyla ya da istemcilerin erişimini düzenlemek ve denetlemek için seçtiğiniz herhangi bir öğeyle ilişkilendirilebilir. Kullanıcı kimliğinin, MQI çağrıları yayınlamak için sunucuda bu kullanıcı için ayrılmış gerekli yetkilere sahip olması gerekir. Alternatif bir kullanıcı kimliği seçilmesi, istemcilerin sunucu bağlantısı MCA ' nın yetkisiyle MQI çağrıları yapmalarına izin verilmesini tercih eder.

Çizelge 9. Sunucu bağlantısı kanalı tarafından kullanılan kullanıcı kimliği

Kullanıcı Kimliği	Kullanıldığında
Bir güvenlik çıkışı tarafından ayarlanan kullanıcı kimliği	Bir **CHLAUTH TYPE(BLOCKUSER)** kuralı tarafından engellenmedikçe kullanılır. Daha fazla bilgi için aşağıdaki “Güvenlik çıkışında kullanıcı kimliğinin ayarlanması” sayfa 102 bölümüne bakın.
CHLAUTH kuralı tarafından ayarlanan kullanıcı kimliği	Bir güvenlik çıkışı tarafından aşırı binilmedikçe kullanılır. Daha fazla bilgi için bkz. Kanal Kimlik Doğrulama Kayıtları .

Çizelge 9. Sunucu bağlantısı kanalı tarafından kullanılan kullanıcı kimliği (devamı var)	
Kullanıcı Kimliği	**Kullanıldığında**
SVRCONN kanal tanımındaki **MCAUSER** özniteliğinde tanımlanan kullanıcı kimliği	Bir güvenlik çıkışı ya da CHLAUTH kuralı tarafından geçersiz kılınmadıkça kullanılır.
İstemci makinesinden akan kullanıcı kimliği	Başka bir yolla kullanıcı kimliği ayarlanmadığı zaman kullanılır.
Sunucu bağlantısı kanalını başlatan kullanıcı kimliği	Başka bir yolla kullanıcı kimliği ayarlanmadığı ve istemci kullanıcı kimliği akmadığı zaman kullanılır. Daha fazla bilgi için aşağıdaki “Kanal programını çalıştıran kullanıcı kimliği” sayfa 103 bölümüne bakın.

Sunucu bağlantısı MCA, uzak kullanıcılar adına MQI çağrıları yaptığı için, uzak istemciler adına MQI çağrıları yayınlayan sunucu bağlantısı MCA ' nın güvenlik etkilerini ve çok sayıda kullanıcının erişiminin nasıl yönetileceğini göz önünde bulundurmak önemlidir.

- Bir yaklaşım, sunucu bağlantısı MCA ' nın MQI çağrılarını kendi yetkisiyle yayınlamasını sağlar. Ancak, istemci kullanıcıları adına MQI çağrıları yayınlamak, güçlü erişim yetenekleriyle sunucu bağlantısı MCA için normalde istenmeyen bir durum.
- Başka bir yaklaşım, istemciden akan kullanıcı kimliğini kullanmaktır. Sunucu bağlantısı MCA, istemci kullanıcı kimliğinin erişim yeteneklerini kullanarak MQI çağrıları verebilir. Bu yaklaşım göz önünde bulundurulması gereken bir dizi soru sunar:
  1. Farklı platformlarda kullanıcı kimliği için farklı biçimler vardır. Bu durum, istemcideki kullanıcı kimliğinin biçimi sunucudaki kabul edilebilir biçimlerden farklıysa sorunlara neden olabilir.
  2. Farklı kullanıcı kimliklerine sahip ve kullanıcı kimliklerini değiştiren potansiyel olarak çok sayıda istemci var. Kimliklerin sunucuda tanımlanması ve yönetilmesi gerekir.
  3. Kullanıcı kimliği güvenilir mi? Oturum açmış kullanıcının kimliği değil, herhangi bir kullanıcı kimliği istemciden akıtılabilir. Örneğin, istemci, güvenlik nedenleriyle yalnızca sunucuda kasıtlı olarak tanımlanmış tam `mqm` yetkisine sahip bir kimlik akıtabilir.
- Tercih edilen yaklaşım, sunucuda istemci tanıtıcı belirteçlerinin tanımlanması ve bu nedenle istemci bağlantılı uygulamaların yeteneklerinin sınırlandırılmasıdır. Bu genellikle, sunucu bağlantısı kanal özelliği MCAUSER, istemciler tarafından kullanılacak özel bir kullanıcı kimliği değerine ayarlanarak ve sunucuda farklı yetki düzeyine sahip istemciler tarafından kullanılmak üzere birkaç kimlik tanımlanarak yapılır.

Güvenlik çıkışında kullanıcı kimliğinin ayarlanması

IBM MQ MQI clientsiçin, MQI çağrılarını veren işlem sunucu bağlantısı MCA 'dır. Sunucu bağlantısı MCA tarafından kullanılan kullanıcı kimliği, MQCD ' nin `MCAUserIdentifier` ya da `LongMCAUserIdentifier` alanlarında bulunur. Bu alanların içeriği aşağıdaki şekilde belirlenir:

- Güvenlik çıkışları tarafından ayarlanan herhangi bir değer
- İstemcideki kullanıcı kimliği
- MCAUSER (sunucu bağlantısı kanal tanımında)

Güvenlik çıkışı, çağrıldığında görünür olan değerleri geçersiz kılabilir.

- Sunucu-bağlantı kanalı MCAUSER özniteliği boş değil olarak ayarlanırsa, MCAUSER değeri kullanılır.
- Sunucu bağlantısı kanalı MCAUSER özniteliği boşsa, istemciden alınan kullanıcı kimliği kullanılır.
- Sunucu bağlantısı kanalı MCAUSER özniteliği boşsa ve istemciden kullanıcı kimliği alınmazsa, sunucu bağlantısı kanalını başlatan kullanıcı kimliği kullanılır.

İstemci tarafı güvenlik çıkışı kullanımdayken, IBM MQ istemcisi bildirilir kullanıcı kimliğini sunucuya akıtmaz.

Kanal programını çalıştıran kullanıcı kimliği

Kullanıcı kimliği alanları, sunucu bağlantısı kanalını başlatan kullanıcı kimliğinden türetildiğinde aşağıdaki değer kullanılır:

- z/OS z/OSiçin, kanal başlatıcısına atanan kullanıcı kimliği, z/OS tarafından başlatılan yordamlar çizelgesi tarafından başlatıldı.
- TCP/IP (z/OS dışı) için, `inetd.conf` girişindeki kullanıcı kimliği ya da dinleyiciyi başlatan kullanıcı kimliği.
- SNA (z/OS dışı) için, SNA Server girişindeki kullanıcı kimliği ya da (yoksa) gelen bağlanma isteği ya da dinleyiciyi başlatan kullanıcı kimliği.
- NetBIOS ya da SPX için, dinleyiciyi başlatan kullanıcı kimliği.

MCAUSER özniteliği boş olarak ayarlanmış bir sunucu bağlantısı kanal tanımlaması varsa, istemciler bu kanal tanımlamasını, istemci tarafından sağlanan kullanıcı kimliği tarafından belirlenen erişim yetkisiyle kuyruk yöneticisine bağlanmak için kullanabilirler. Kuyruk yöneticisinin çalıştığı sistem yetkisiz ağ bağlantılarına izin veriyorsa, bu güvenlik açığı olabilir. IBM MQ varsayılan sunucu bağlantısı kanalı (SYSTEM.DEF.SVRCONN) için MCAUSER özniteliği boş olarak ayarlanmış. Yetkisiz erişimi önlemek için, varsayılan tanımlamanın MCAUSER özniteliğini, IBM MQ MQ nesnelerine erişimi olmayan bir kullanıcı kimliğiyle güncelleyin.

Kullanıcı kimlikleri büyük/küçük harf durumu

`runmqsc`ile bir kanal tanımladığınızda, kullanıcı kimliği tek tırnak işareti içinde bulunmadığı sürece MCAUSER özniteliği büyük harfe çevrilir.

ALW AIX, Linux, and Windowsüzerindeki sunucular için, istemciden alınan `MCAUserIdentifier` alanının içeriği küçük harfe çevrilir.

IBM i IBM iüzerindeki sunucular için, istemciden alınan `LongMCAUserIdentifier` alanının içeriği büyük harfe çevrilir.

Linux AIX AIX and Linux sistemlerindeki sunucular için, istemciden alınan `LongMCAUserIdentifier` alanının içeriği küçük harfe çevrilir.

Varsayılan olarak, bir IBM MQ JMS bağ tanımlama uygulaması kullanıldığında geçirilen kullanıcı kimliği, uygulamanın çalıştığı JVM ' nin kullanıcı kimliğidir.

`createQueueConnection` yöntemiyle bir kullanıcı kimliği de geçirilebilmektedir.

Planlama gizliliği

Verilerinizi nasıl gizli tutacağınızı planlayın.

Uygulama düzeyinde ya da bağlantı düzeyinde gizlilik uygulayabilirsiniz. TLS kullanmayı seçebilirsiniz; bu durumda, dijital sertifika kullanımınızı planlamanız gerekir. Standart olanaklar gereksinimlerinizi karşılamıyorsa, kanal çıkış programlarını da kullanabilirsiniz.

İlgili kavramlar

“Bağlantı düzeyinde güvenlik ve uygulama düzeyinde güvenlik karşılaştırması” sayfa 104
Bu konu, bağlantı düzeyi güvenlik ve uygulama düzeyi güvenliğin çeşitli yönleriyle ilgili bilgileri içerir ve iki güvenlik düzeyini karşılaştırır.

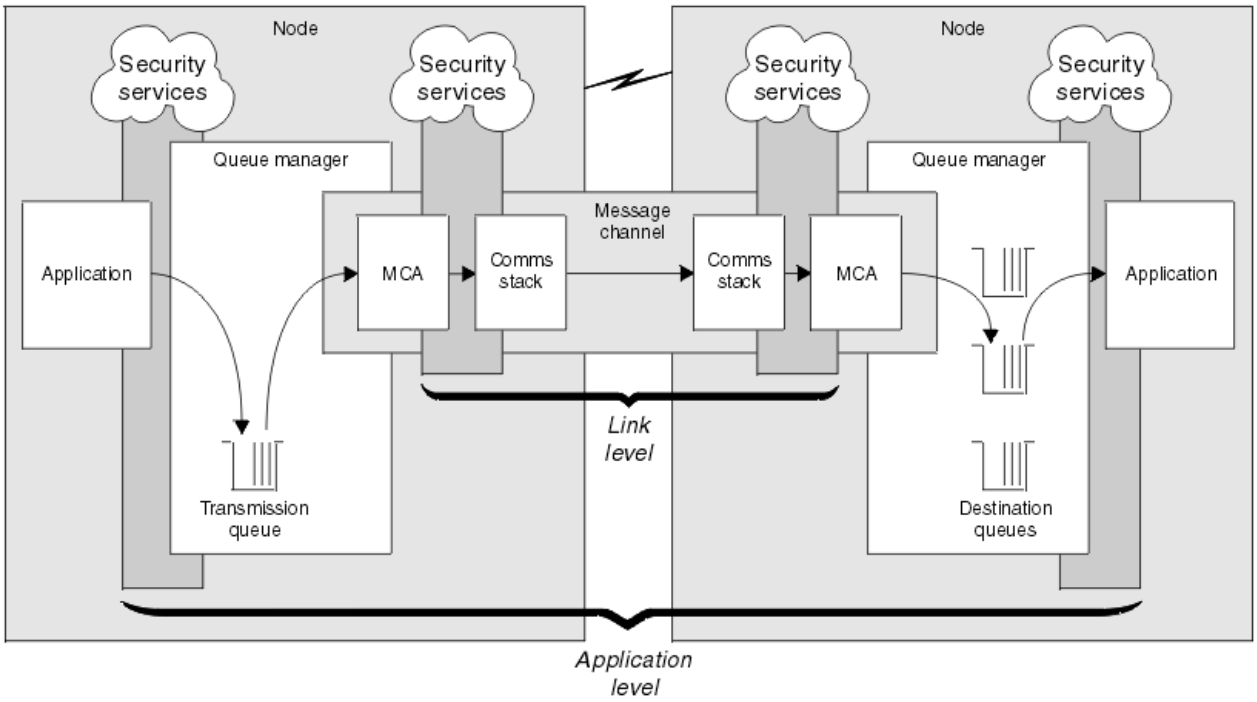
“Kanal çıkış programları” sayfa 108
Kanal çıkış programları , MCA ' nın işleme sırasındaki tanımlı yerlerde çağrılan programlardır. Kullanıcılar ve satıcılar kendi kanal çıkış programlarını yazabilirler. Bazıları IBMtarafından sağlanır.

“SSL/TLS ile kanalları koruma” sayfa 114
IBM MQ içindeki TLS desteği, kuyruk yöneticisi kimlik doğrulama bilgileri nesnesini ve çeşitli MQSC komutlarını kullanır. Dijital sertifika kullanımınızı da göz önünde bulundurmanız gerekir.

Bağlantı düzeyinde güvenlik ve uygulama düzeyinde güvenlik karşılaştırması

Bu konu, bağlantı düzeyi güvenlik ve uygulama düzeyi güvenliğin çeşitli yönleriyle ilgili bilgileri içerir ve iki güvenlik düzeyini karşılaştırır.

Bağlantı düzeyi ve uygulama düzeyi güvenliği Şekil 10 sayfa 104içinde gösterilmiştir.



Şekil 10. Bağlantı düzeyinde güvenlik ve uygulama düzeyinde güvenlik

Kuyruklardaki iletileri koruma

Bağ düzeyi güvenlik, iletiler bir kuyruk yöneticisinden diğerine aktarılırken iletileri koruyabilir. İletiler güvenli olmayan bir ağ üzerinden iletildiğinde bu özellikle önemlidir. Ancak, iletiler bir kaynak kuyruk yöneticisinde, hedef kuyruk yöneticisinde ya da bir ara kuyruk yöneticisinde saklanırken iletileri koruyamaz.

z/OS z/OS veri kümesi şifrelemesi, yalnızca yerel bir kuyruk yöneticisinde atıl durumdaki veriler için, kuyruklarda saklanan iletilerin bazı korunmasını sağlayabilir. veri kümesi şifrelemesiyle IBM MQ for z/OS üzerinde atıl durumdaki veriler için gizlilik. başlıklı bölüme bakın. ek bilgi için.

Uygulama düzeyinde güvenlik, iletilerin kuyruklarda saklanması sırasında bunları koruyabilir ve dağıtılmış kuyruğa alma kullanılmadığında bile geçerli olur. Bu, bağlantı düzeyi güvenlik ile uygulama düzeyi güvenlik arasındaki başlıca farktır ve Şekil 10 sayfa 104içinde gösterilmiştir.

Denetimli ve güvenilir ortamlarda çalışmayan kuyruk yöneticileri

Bir kuyruk yöneticisi denetimli ve güvenilir bir ortamda çalışıyorsa, IBM MQ tarafından sağlanan erişim denetimi mekanizmaları, kuyruklarında saklanan iletileri korumak için yeterli kabul edilebilir. Bu, özellikle yalnızca yerel kuyruğa alma ile ilgiliyse ve iletiler kuyruk yöneticisinden hiçbir zaman ayrılmamışsa geçerlidir. Bu durumda uygulama düzeyinde güvenlik gereksiz olarak kabul edilebilir.

İletiler, denetimli ve güvenilir bir ortamda da çalışan başka bir kuyruk yöneticisine aktarılırsa ya da böyle bir kuyruk yöneticisinden alınırsa, uygulama düzeyinde güvenlik gereksiz olarak da kabul edilebilir. Denetimli ve güvenilir bir ortamda çalışmayan bir kuyruk yöneticisine ileti aktarıldığında ya da bu yöneticiden ileti alındığında, uygulama düzeyinde güvenlik gereksinimi daha fazla olur.

Maliyet farkları

Uygulama düzeyinde güvenlik, yönetim ve performans açısından bağlantı düzeyinde güvenlikten daha fazlasına mal olabilir.

Yapılandırılacak ve korunacak daha fazla kısıtlama olabileceğinden, yönetim maliyeti daha yüksek olabilir. Örneğin, belirli bir kullanıcının yalnızca belirli ileti tiplerini gönderdiğinden ve yalnızca belirli hedeflere ileti gönderdiğinden emin olmanız gerekebilir. Tersi durumda, belirli bir kullanıcının yalnızca belirli tipte ileti aldığından ve yalnızca belirli kaynaklardan ileti aldığından emin olmanız gerekebilir. Tek bir ileti kanalında bağlantı düzeyi güvenlik hizmetlerini yönetmek yerine, o kanalda ileti alışverişi yapan her kullanıcı çifti için kuralları yapılandırmanız ve sürdürmeniz gerekebilir.

Bir uygulama her ileti gönderdiğinde ya da aldığında güvenlik hizmetleri çağrılırsa, performans üzerinde bir etkisi olabilir.

Kuruluşlar, uygulanması daha kolay olabileceğinden önce bağlantı düzeyinde güvenliği göz önünde bulundurmaya eğilimlidir. Bağlantı düzeyi güvenliğin tüm gereksinimlerini karşılamadığını keşfederlerse, uygulama düzeyinde güvenliği göz önünde bulundurarlar.

Bileşenlerin kullanılabilirliği

Genellikle, dağıtılmış bir ortamda, bir güvenlik hizmeti en az iki sistemde bir bileşen gerektirir. Örneğin, bir ileti bir sistemde şifrelenebilir ve başka bir sistemde şifresi çözülebilir. Bu, hem bağlantı düzeyinde güvenlik hem de uygulama düzeyinde güvenlik için geçerlidir.

Farklı platformların kullanıldığı, her biri farklı güvenlik işlevlerine sahip türdeş olmayan bir ortamda, güvenlik hizmetinin gerekli bileşenleri, ihtiyaç duydukları her platform için ve kullanımı kolay olan bir biçimde kullanılamayabilir. Bu, özellikle çeşitli kaynaklardan bileşenler satın alarak kendi uygulama düzeyi güvenliğinizi sağlamayı amaçlıyorsanız, uygulama düzeyinde güvenlik için bağlantı düzeyinde güvenlikten daha büyük bir sorundur.

Gönderilmeyen iletiler kuyruğundaki iletiler

Bir ileti uygulama düzeyi güvenliğiyle korunuyorsa, herhangi bir nedenle ileti hedefine ulaşmazsa ve gitmeyen iletiler kuyruğuna konursa bir sorun olabilir. İleti tanımlayıcıdaki ve gitmeyen harf üstbilgisindeki bilgilerden iletinin nasıl işleneceğini çözemezseniz, uygulama verilerinin içeriğini incelemeniz gerekebilir. Uygulama verileri şifrelenmişse ve yalnızca istenen alıcı şifresini çözebiliyorsa bunu yapamazsınız.

Uygulama düzeyinde güvenliğin yapamayamayacağı

Uygulama düzeyinde güvenlik tam bir çözüm değildir. Uygulama düzeyinde güvenlik uygulasanız bile, bağlantı düzeyinde bazı güvenlik hizmetlerine gereksinim duyabilirsiniz. Örneğin:

- Bir kanal başlatıldığında, iki MCA ' nin karşılıklı kimlik doğrulaması yine de bir gereksinim olabilir. Bu yalnızca bağlantı düzeyinde bir güvenlik hizmeti tarafından yapılabilir.
- Uygulama düzeyi güvenlik, yerleşik ileti tanımlayıcısını içeren iletim kuyruğu üstbilgisi olan MQXQH ' yi koruyamıyor. Ayrıca, ileti verileri dışındaki IBM MQ kanal iletişim kuralı akışlarındaki verileri de koruyabilir. Bu korumayı yalnızca bağlantı düzeyinde güvenlik sağlayabilir.
- Bir MQI kanalının sunucu ucunda uygulama düzeyinde güvenlik hizmetleri çağrılırsa, hizmetler kanal üzerinden gönderilen MQI çağrılarının değiştirgelerini koruyamaz. Özellikle bir MQPUT, MQPUT1ya da MQGET çağrısındaki uygulama verileri korunmuyor. Bu durumda yalnızca bağlantı düzeyinde güvenlik koruma sağlayabilir.

Bağlantı düzeyi güvenlik

Bağlantı düzeyi güvenlik , MCA, iletişim altsistemi ya da birlikte çalışan ikisinin birleşimi tarafından doğrudan ya da dolaylı olarak çağrılan güvenlik hizmetlerini ifade eder.

Bağlantı düzeyi güvenliği Şekil 10 sayfa 104içinde gösterilmektedir.

Aşağıda bağlantı düzeyi güvenlik hizmetlerine ilişkin bazı örnekler verilmiştir:

- Bir ileti kanalının her ucundaki MCA, ortağının kimliğini doğrulayabilir. Bu, kanal başlatıldığında ve bir iletişim bağlantısı kurulduğunda, ancak herhangi bir ileti akmaya başlamadan önce yapılır. Her iki uçta da kimlik doğrulaması başarısız olursa, kanal kapanır ve hiçbir ileti aktarılmaz. Bu, bir tanımlama ve kimlik doğrulama hizmeti örneğidir.
- Bir ileti, bir kanalın gönderme ucunda şifrelenebilir ve alıcı ucunda şifresi çözülebilir. Bu bir gizlilik hizmeti örneğidir.
- Bir ileti, ağ üzerinden iletildiği sırada içeriğinin kasıtlı olarak değiştirilip değiştirilmediğini belirlemek için bir kanalın alıcı ucunda kontrol edilebilir. Bu, bir veri bütünlüğü hizmeti örneğidir.

IBM MQ tarafından sağlanan bağlantı düzeyi güvenlik

IBM MQ ' de gizlilik ve veri bütünlüğünün sağlanmasının birincil yolu TLS kullanımıyla ilgilidir. IBM MQiçinde TLS kullanımı hakkında daha fazla bilgi için bkz. “IBM MQ içinde TLS güvenlik iletişim kuralları” sayfa 24. IBM MQ , kimlik doğrulaması için kanal kimlik doğrulama kayıtlarını kullanma olanağını sağlar. Kanal kimlik doğrulama kayıtları, tek tek kanallar veya kanal grupları düzeyinde, bağlanan sistemlere verilen erişim üzerinde kesin denetim sağlar. Daha fazla bilgi için “Kanal kimlik doğrulama kayıtları” sayfa 50 başlıklı konuya bakın.

Kendi bağlantı düzeyi güvenliğinizi sağlama
Kendi bağlantı düzeyi güvenlik hizmetlerinizi sağlayabilirsiniz. Kendi kanal çıkış programlarınızı yazmak, kendi bağlantı düzeyinde güvenlik hizmetlerinizi sağlamanın ana yoludur.

Kanal çıkış programları “Kanal çıkış programları” sayfa 108içinde kullanıma sunulmuştur. Aynı konu, IBM MQ for Windows (SSPI kanal çıkış programı) ile sağlanan kanal çıkış programını da açıklar. Kaynak kodu gereksinimlerinize uyacak şekilde değiştirebilmeniz için bu kanal çıkış programı kaynak biçiminde sağlanır. Bu kanal çıkış programı ya da diğer satıcı firmaların kullanabileceği kanal çıkış programları gereksinimlerinizi karşılamıyorsa, kendi kanal çıkış programınızı tasarlayabilir ve yazabilirsiniz. Bu konuda, kanal çıkış programlarının güvenlik hizmetleri sağlayabileceği yollar açıklanmaktadır. Kanal çıkış programının nasıl yazılacağına ilişkin bilgi için Kanal çıkış programlarının yazılmasıbaşlıklı konuya bakın.

Güvenlik çıkışı kullanan bağlantı düzeyi güvenlik
Güvenlik çıkışları normal olarak çift olarak çalışır; bir kanalın her bir ucunda bir adet. Bunlar, kanal başlatıldığında ilk veri anlaşması tamamlandıktan hemen sonra çağrılır.

Kimlik belirleme ve kimlik doğrulama, erişim denetimi ve gizlilik sağlamak için güvenlik çıkışları kullanılabilir.

İleti çıkışı kullanan bağlantı düzeyi güvenlik
İleti çıkışı, MQI kanalında değil, yalnızca ileti kanalında kullanılabilir. Hem yerleşik ileti tanımlayıcısını, hem de bir iletideki uygulama verilerini içeren iletim kuyruğu üstbilgisi olan MQXQH ' ye erişimi vardır. İletinin içeriğini değiştirebilir ve uzunluğunu değiştirebilir.

Bir ileti çıkışı, bir kısmı yerine tüm iletiye erişim gerektiren herhangi bir amaçla kullanılabilir.

İleti çıkışları, kimlik belirleme ve kimlik doğrulama, erişim denetimi, gizlilik, veri bütünlüğü ve inkar edilmeme gibi güvenlik dışındaki nedenlerle kullanılabilir.

Gönderme ve alma çıkışlarını kullanarak bağlantı düzeyinde güvenlik
Gönderme ve alma çıkışları hem ileti kanallarında hem de MQI kanallarında kullanılabilir. Bunlar, bir kanalda akan tüm veri tipleri ve her iki yönde akan akışlar için çağrılır.

Gönderme ve alma çıkışlarının her iletim bölümüne erişimi vardır. İçeriğini değiştirebilir ve uzunluğunu değiştirebilir.

Bir ileti kanalında, bir MCA ' nın bir iletiyi bölmesi ve birden fazla iletim bölümünde göndermesi gerekirse, iletinin bir bölümünü içeren her iletim bölümü için bir gönderme çıkışı çağrılır ve alıcı ucunda, her iletim bölümü için bir alma çıkışı çağrılır. MQI çağrısının giriş ya da çıkış değiştirgeleri tek bir iletim kesiminde gönderilemeyecek kadar büyükse, MQI kanalında da aynı durum oluşur.

Bir MQI kanalında, bir iletim parçasının 10 byte 'ı MQI çağrısını tanımlar ve iletim parçasının çağrıya ilişkin giriş ya da çıkış değiştirgelerini içerip içermediğini gösterir. Gönderme ve alma çıkışları, MQI çağrılarının korunması gerekebilecek uygulama verileri içerip içermediğini saptamak için bu byte 'ı inceleyebilir.

Bir gönderme çıkışı ilk kez çağrıldığında, ihtiyacı olan kaynakları elde etmek ve kullanıma hazırlamak için MCA ' dan bir iletim kesimini tutan arabellekte belirli bir miktarda yer ayırmasını isteyebilir. Daha sonra bir iletim bölümünü işlemek için çağrıldığında, örneğin, şifrelenmiş bir anahtar ya da dijital imza eklemek için bu alanı kullanabilir. Kanalın diğer ucundaki karşılık gelen alma çıkışı, gönderme çıkışı tarafından eklenen verileri kaldırabilir ve iletim bölümünü işlemek için bunu kullanabilir.

Gönderme ve alma çıkışları, işledikleri verilerin yapısını anlamaları gerekmeyen amaçlar için en uygun yoldur ve bu nedenle her iletim kesimini ikili nesne olarak ele alabilir.

Gönderme ve alma çıkışları, gizlilik ve veri bütünlüğü sağlamak için ve güvenlik dışında kullanımlar için kullanılabilir.

İlgili görevler

Bir gönderme ya da alma çıkış programında API çağrısını tanımlama

Uygulama düzeyinde güvenlik

Uygulama düzeyi güvenlik , bir uygulama ile bağlı olduğu bir kuyruk yöneticisi arasındaki arabirimde çağrılan güvenlik hizmetlerini ifade eder.

Bu hizmetler, uygulama kuyruk yöneticisine MQI çağrıları verdiğinde çağrılır. Hizmetler doğrudan ya da dolaylı olarak uygulama, kuyruk yöneticisi, IBM MQ' u destekleyen başka bir ürün ya da bunların birlikte çalışan herhangi bir bileşimi tarafından çağrılabilir. Uygulama düzeyinde güvenlik, Şekil 10 sayfa 104içinde gösterilmektedir.

Uygulama düzeyinde güvenlik, *uçtan uca güvenlik* ya da *ileti düzeyinde güvenlik*olarak da bilinir.

Aşağıda, uygulama düzeyinde güvenlik hizmetlerine ilişkin bazı örnekler verilmiştir:

- Bir uygulama bir iletiyi kuyruğa koyduğunda, ileti tanımlayıcısı uygulamayla ilişkilendirilmiş bir kullanıcı kimliği içerir. Ancak, kullanıcı kimliğini doğrulamak için kullanılabilecek şifrelenmiş parola gibi bir veri yoktur. Bir güvenlik hizmeti bu verileri ekleyebilir. İleti en sonunda alıcı uygulama tarafından alındığında, hizmetin başka bir bileşeni, iletiyle birlikte gönderilen verileri kullanarak kullanıcı kimliğini doğrulayabilir. Bu, bir tanımlama ve kimlik doğrulama hizmeti örneğidir.
- Bir ileti, bir uygulama tarafından kuyruğa konduğunda ve alan uygulama tarafından alındığında şifresi çözüldüğünde şifrelenebilir.Bu bir gizlilik hizmeti örneğidir.
- Bir ileti, alan uygulama tarafından alındığında denetlenir. Bu denetim, içeriğin gönderen uygulama tarafından kuyruğa ilk konmasından bu yana kasıtlı olarak değiştirilip değiştirilmediğini belirler.Bu, bir veri bütünlüğü hizmeti örneğidir.

Advanced Message Security planlaması

Advanced Message Security (AMS), son uygulamaları etkilemediği halde IBM MQ ağı üzerinden akan hassas veriler için yüksek düzeyde koruma sağlayan bir IBM MQ bileşenidir.

Özellikle hasta kayıtları veya kredi kartı ayrıntıları gibi gizli ya da ödemeyle ilgili bilgileri, yüksek düzeyde hassas ya da değerli bilgileri taşıyorsanız, bilgi güvenliğine özel olarak dikkat etmeniz gerekir. Bilgilerin kuruluş içinde hareket etmesini ve yetkisiz erişimden korunmasını sağlamak sürekli bir zorluk ve sorumluluktur. Ayrıca, uyumluluğa uyulmaması durumunda, güvenlik düzenlemelerine uymanız gerekebilir.

IBM MQiçin kendi güvenlik uzantılarınızı geliştirebilirsiniz. Ancak bu tür çözümler uzman becerileri gerektirir ve bakımı karmaşık ve pahalı olabilir. Advanced Message Security , bilgilerin hemen hemen her tür ticari BT sistemi arasında kuruluş çapında taşınması sırasında bu zorlukların aşılmasına yardımcı olur.

Advanced Message Security , IBM MQ ürününün güvenlik özelliklerini aşağıdaki şekillerde genişletir:

- İletilerin şifrelenmesi ya da dijital imzası kullanılarak noktadan noktaya ileti sistemi altyapınız için uygulama düzeyinde, uçtan uca veri koruması sağlar.
- Karmaşık güvenlik kodu yazmadan ya da var olan uygulamaları değiştirmeden ya da yeniden derlemeden kapsamlı güvenlik sağlar.
- İletiler için kimlik doğrulama, yetkilendirme, gizlilik ve veri bütünlüğü hizmetleri sağlamak için Genel Anahtar Altyapısı (PKI) teknolojisini kullanır.

- Ana bilgisayar ve dağıtılmış sunucular için güvenlik ilkelerinin yönetimini sağlar.
- Hem IBM MQ sunucularını hem de istemcilerini destekler.
- Uçtan uca güvenli bir ileti sistemi çözümü sağlamak için Managed File Transfer ile bütünleşir.

Daha fazla bilgi için “Advanced Message Security” sayfa 609 başlıklı konuya bakın.

Kendi uygulama düzeyinde güvenliğinizi sağlama
Kendi uygulama düzeyinde güvenlik hizmetlerinizi sağlayabilirsiniz. Uygulama düzeyi güvenliği uygulamanıza yardımcı olmak için IBM MQ , API çıkışı ve API geçişi çıkışı olmak üzere iki çıkış sağlar.

API çıkışı ve API geçiş çıkışı, kimlik belirleme ve kimlik doğrulama, erişim denetimi, gizlilik, veri bütünlüğü ve inkar etmeme hizmetleri ve güvenlikle ilgili olmayan diğer işlevler sağlayabilir.

Sistem ortamınızda API çıkışı ya da API geçişi çıkışı desteklenmiyorsa, kendi uygulama düzeyi güvenliğinizi sağlamanın başka yollarını da göz önünde bulundurmak isteyebilirsiniz. Bir yol, MQI ' ı çevreleyen daha üst düzey bir API geliştirmektir. Programcılar daha sonra IBM MQ uygulamalarını yazmak için MQI yerine bu API ' yi kullanır.

Daha yüksek düzeyli bir API kullanmak için en yaygın nedenler şunlardır:

- MQI 'ın daha gelişmiş özelliklerini programcılardan gizlemek için.
- MQI kullanımında standartları uygulamak için.
- MQI 'a işlev eklemek için. Bu ek işlev güvenlik hizmetleri olabilir.

Bazı satıcı firma ürünleri, IBM MQiçin uygulama düzeyinde güvenlik sağlamak üzere bu tekniği kullanır.

Güvenlik hizmetlerini bu şekilde sağlamayı planlıyorsanız, veri dönüştürmeyle ilgili olarak aşağıdakilere dikkat edin:

- Bir iletideki uygulama verilerine sayısal imza gibi bir güvenlik simgesi eklendiyse, veri dönüştürmeyi gerçekleştiren herhangi bir kod, bu simgenin varlığından haberdar olmalıdır.
- Bir güvenlik simgesi, uygulama verilerinin ikili görüntüsünden türetilmiş olabilir. Bu nedenle, veriler dönüştürülmeden önce belirteç denetimi yapılmalıdır.
- Bir iletideki uygulama verileri şifrelenmişse, veri dönüştürmeden önce şifresinin çözülmesi gerekir.

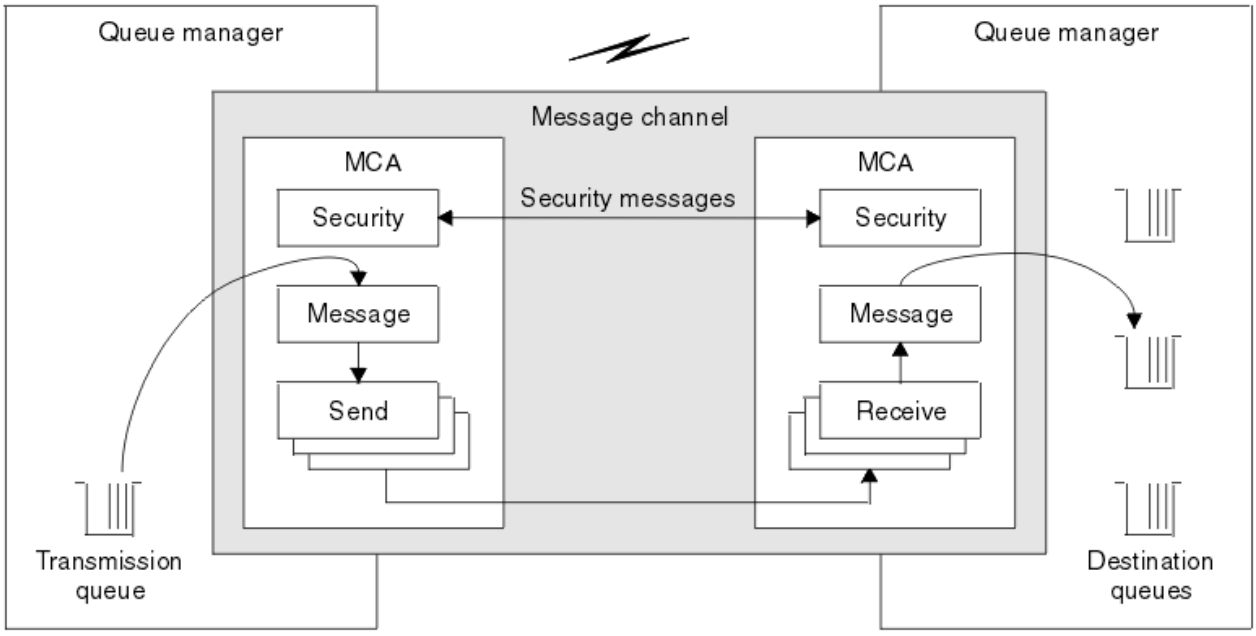
Kanal çıkış programları

Kanal çıkış programları , MCA ' nın işleme sırasındaki tanımlı yerlerde çağrılan programlardır. Kullanıcılar ve satıcılar kendi kanal çıkış programlarını yazabilirler. Bazıları IBMtarafından sağlanır.

Kanal çıkış programının birkaç tipi vardır, ancak bağlantı düzeyinde güvenlik sağlanmasında yalnızca dördü rol oynar:

- Güvenlik Çıkışı
- İleti çıkışı
- Çıkış gönder
- Alma çıkışı

Bu dört kanal çıkış programı tipi Şekil 11 sayfa 109 içinde gösterilmiştir ve aşağıdaki konularda açıklanmıştır.



Şekil 11. Bir ileti kanalındaki güvenlik, ileti, gönderme ve alma çıkışları

İlgili kavramlar

İleti sistemi kanalları için kanal çıkış programları

Güvenlik çıkışlarına genel bakış

Güvenlik çıkışları genellikle çiftler halinde çalışır. Bunlar ileti akışından önce çağrılır ve amaçları bir MCA ' nın ortağını doğrulamasını sağlamaktır.

Güvenlik çıkışları genellikle çift olarak çalışır; bir kanalın her bir ucunda bir tane bulunur. Bunlar, kanal başlangıcında ilk veri anlaşması tamamlandıktan hemen sonra, ancak herhangi bir ileti akmaya başlamadan önce çağrılır. Güvenlik çıkışının birincil amacı, bir kanalın her ucunda MCA ' nın ortağının kimliğini doğrulamasını sağlamaktır. Ancak, güvenlikle ilgisi olmayan bir işlevin bile, güvenlik çıkışının başka bir işlevi gerçekleştirmesini önleyecek bir şey yoktur.

Güvenlik çıkışları, *güvenlik iletileri*göndererek birbiriyle iletişim kurabilir. Bir güvenlik iletisinin biçimi tanımlanmaz ve kullanıcı tarafından belirlenir. Güvenlik iletilerinin değiş tokuş edilmesinin olası bir sonucu, güvenlik çıkışlarından birinin daha fazla devam etmemeye karar vermemesi olabilir. Bu durumda, kanal kapalıdır ve iletiler akmaz. Bir kanalın yalnızca bir ucunda bir güvenlik çıkışı varsa, çıkış yine de çağrılır ve devam etmeyi ya da kanalı kapatmayı seçebilir.

Güvenlik çıkışları hem ileti kanallarında hem de MQI kanallarında çağrılabilir. Bir güvenlik çıkışının adı, kanalın her bir ucundaki kanal tanımında bir parametre olarak belirlenir.

Güvenlik çıkışları hakkında daha fazla bilgi için bkz. “Güvenlik çıkışı kullanan bağlantı düzeyi güvenlik” sayfa 106.

İleti çıkışı

İleti çıkışları yalnızca ileti kanallarında çalışır ve genellikle çiftler halinde çalışır. Bir ileti çıkışı tüm iletide çalışabilir ve iletide çeşitli değişiklikler yapabilir.

Bir kanalın gönderme ve alma uçlarındaki *ileti çıkışları* olağan durumda çiftler halinde çalışır. MCA iletim kuyruğundan bir ileti aldıktan sonra, bir kanalın gönderme sonundaki bir ileti çıkışı çağrılır. Bir kanalın alıcı ucunda, MCA hedef kuyruğuna bir ileti koymadan önce bir ileti çıkışı çağrılır.

İleti çıkışı, hem yerleşik ileti tanımlayıcısını, hem de bir iletideki uygulama verilerini içeren iletim kuyruğu üstbilgisi olan MQXQH ' ye erişebilir. Bir ileti çıkışı, iletinin içeriğini değiştirebilir ve uzunluğunu değiştirebilir. Uzunluk değişikliği, iletinin sıkıştırılmasının, sıkıştırılmasının, şifrelenmesinin ya da şifresinin çözülmesinin sonucu olabilir. Bu, iletiye veri eklenmesinin ya da iletiden veri kaldırılmasının sonucu da olabilir.

İleti çıkışları, güvenlik için değil, tüm iletiye erişim gerektiren herhangi bir amaç için kullanılabilir.

Bir ileti çıkışı, işlenmekte olduğu iletinin hedefine daha fazla devam etmemesi gerektiğini saptayabilir. MCA daha sonra iletiyi ölü mektup kuyruğuna koyar. Bir ileti çıkışı kanalı da kapatabilir.

İleti çıkışları MQI kanallarında değil, yalnızca ileti kanallarında çağrılabilir. Bunun nedeni, bir MQI kanalının amacının, MQI çağrılarının giriş ve çıkış değiştirgelerinin IBM MQ MQI client uygulaması ile kuyruk yöneticisi arasında akmasını sağlamak olmasıdır.

Bir ileti çıkışının adı, bir kanalın her sonunda kanal tanımında bir parametre olarak belirtilir. Art arda çalıştırılacak ileti çıkışlarının listesini de belirtebilirsiniz.

İleti çıkışları hakkında daha fazla bilgi için bkz. “İleti çıkışı kullanan bağlantı düzeyi güvenlik” sayfa 106.

Gönderme ve alma çıkışları

Gönderme ve alma çıkışları genellikle çiftler halinde çalışır. İletim segmentleri üzerinde çalışır ve en iyi şekilde, işledikleri verilerin yapısının ilgili olmadığı durumlarda kullanılır.

Bir kanalın bir ucundaki *gönderme çıkışı* ve diğer ucundaki *alma çıkışı* olağan koşullarda çiftler halinde çalışır. MCA, bir iletişim bağlantısı üzerinden veri göndermek için bir iletişim göndermeden hemen önce bir gönderme çıkışı çağrılır. MCA, bir iletişim alma işleminden sonra denetimi yeniden ele geçirdikten ve iletişim bağlantısından veri aldıktan hemen sonra bir alma çıkışı çağrılır. Paylaşım etkileşimleri kullanıyorsa, bir MQI kanalı üzerinden, her etkileşim için farklı bir gönderme ve alma çıkışı eşgörünümü çağrılır.

Bir ileti kanalındaki iki MCA arasında IBM MQ kanal iletişim kuralı akışı, ileti verilerinin yanı sıra denetim bilgilerini de içerir. Benzer şekilde, bir MQI kanalında akışlar, MQI çağrılarının değiştirgelerinin yanı sıra denetim bilgilerini de içerir. Tüm veri tipleri için gönderme ve alma çıkışları çağrılır.

İleti verileri bir ileti kanalında yalnızca bir yönde akar, ancak bir MQI kanalında bir MQI çağrısı akışının giriş parametreleri bir yönde, çıkış parametreleri ise diğer yönde akar. Hem ileti kanallarında hem de MQI kanallarında, denetim bilgileri her iki yönde de akar. Sonuç olarak, bir kanalın her iki ucunda da gönderme ve alma çıkışları çağrılabilir.

İki MCA arasında tek bir akışta iletilen veri birimine *iletim kesimi*denir. Gönderme ve alma çıkışlarının her iletim bölümüne erişimi vardır. İçeriğini değiştirebilir ve uzunluğunu değiştirebilir. Ancak, gönderme çıkışı, bir iletim kesiminin ilk 8 baytını değiştirmemelidir. Bu 8 bayt, IBM MQ kanal iletişim kuralı üstbilgisinin bir bölümünü oluşturur. Ayrıca, bir gönderme çıkışının iletim kesiminin uzunluğunu ne kadar artırabileceğine ilişkin kısıtlamalar da vardır. Özellikle, bir gönderme çıkışı, kanal başlangıcında iki MCA arasında kararlaştırılan uzunluk üst sınırını aşamaz.

Bir ileti kanalında, bir ileti tek bir iletim bölümünde gönderilemeyecek kadar büyükse, gönderen MCA iletiyi böler ve birden fazla iletim bölümünde gönderir. Sonuç olarak, iletinin bir bölümünü içeren her iletim kesimi için bir gönderme çıkışı çağrılır ve alıcı ucunda, her iletim kesimi için bir alma çıkışı çağrılır. Alan MCA, alma çıkışı tarafından işlendikten sonra iletim bölümlerinden gelen iletiyi yeniden oluşturur.

Benzer şekilde, bir MQI kanalında, MQI çağrısının giriş ya da çıkış parametreleri çok büyükse birden çok iletim bölümünde gönderilir. Bu durum, örneğin, bir MQPUT, MQPUT1ya da uygulama verileri yeterince büyükse MQGET çağrısında oluşabilir.

Bu hususları göz önünde bulundurarak, gönderme ve alma çıkışlarını, işledikleri verilerin yapısını anlamaları gerekmeyen amaçlarla kullanmak daha uygundur ve bu nedenle her iletim kesimini ikili nesne olarak ele alabilir.

Bir gönderme ya da alma çıkışı bir kanalı kapatabilir.

Bir gönderme çıkışının ve alma çıkışının adları, kanalın her bir ucundaki kanal tanımında parametre olarak belirlenir. Art arda çalıştırılacak gönderme çıkışlarının listesini de belirtebilirsiniz. Benzer şekilde, bir alma çıkışları listesi belirtebilirsiniz.

Gönderme ve alma çıkışları hakkında daha fazla bilgi için bkz. “Gönderme ve alma çıkışlarını kullanarak bağlantı düzeyinde güvenlik” sayfa 106.

Planlama verileri bütünlüğü

Verilerinizin bütünlüğünü nasıl koruyacağınızı planlayın.

Veri bütünlüğünü uygulama düzeyinde ya da bağlantı düzeyinde uygulayabilirsiniz.

Uygulama düzeyinde, standart olanaklar gereksinimlerinizi karşılamıyorsa API çıkış programlarını kullanabilirsiniz. Yetkisiz değişikliklere karşı koruma sağlamak üzere iletileri dijital olarak imzalamak için Advanced Message Security (AMS) kullanmayı seçebilirsiniz.

Bağlantı düzeyinde TLS kullanmayı seçebilirsiniz; bu durumda dijital sertifika kullanımınızı planlamanız gerekir. Standart olanaklar gereksinimlerinizi karşılamıyorsa, kanal çıkış programlarını da kullanabilirsiniz.

İlgili kavramlar

"SSL/TLS ile kanalları koruma" sayfa 114
IBM MQ içindeki TLS desteği, kuyruk yöneticisi kimlik doğrulama bilgileri nesnesini ve çeşitli MQSC komutlarını kullanır. Dijital sertifika kullanımınızı da göz önünde bulundurmanız gerekir.

"Veri bütünlüğü" sayfa 10
Veri bütünlüğü hizmeti, verilerde yetkisiz değişiklik olup olmadığını saptar.

"Advanced Message Security planlaması" sayfa 107
Advanced Message Security (AMS), son uygulamaları etkilemediği halde IBM MQ ağı üzerinden akan hassas veriler için yüksek düzeyde koruma sağlayan bir IBM MQ bileşenidir.

İlgili başvurular

API çıkış başvurusu

Kanal çıkışı çağrıları ve veri yapıları

Planlama denetimi

Denetlemeniz gereken verileri ve denetim bilgilerini nasıl yakalayacağınıza ve işleyeceğinize karar verin. Sisteminizin konfigürasyonunun doğru tanımlanıp tanımlanmadığını denetlemenizi göz önünde bulundurun.

Etkinlik izlemenin çeşitli yönleri vardır. Göz önünde bulundurmanız gereken hususlar genellikle denetçi gereksinimleri tarafından tanımlanır ve bu gereksinimler genellikle HIPAA (Health Insurance Portability and Accountability Act; Sağlık Sigortası Taşınabilirlik ve Sorumluluk Yasası) veya SOX (Sarbanes-Oxley) gibi yasal düzenleme standartlarına göre belirlenir. IBM MQ , bu tür standartlara uyulmasına yardımcı olmak için tasarlanmış özellikler sağlar.

Yalnızca istisnalarla ilgilenip ilgilenmediğinizi ya da tüm sistem davranışlarıyla ilgilenip ilgilenmediğinizi göz önünde bulundurun.

Denetimin bazı yönleri de operasyonel izleme olarak kabul edilebilir; denetim için bir fark, yalnızca gerçek zamanlı uyarılara değil, genellikle tarihi verilere bakmaktır. İzleme, İzleme ve performansbölümünde ele alınmıştır.

Denetlenecek veriler

Aşağıdaki bölümlerde açıklandığı gibi denetlemeniz gereken veri ya da etkinlik tiplerini göz önünde bulundurun:

IBM MQ arabirimlerini kullanarak IBM MQ üzerinde yapılan değişiklikler
Özel işlemden geçirme olaylarını, özellikle komut olaylarını ve yapılandırma olaylarını yayınlamak için IBM MQ ' yi yapılandırın.

IBM MQ üzerinde denetiminin dışında yapılan değişiklikler
Bazı değişiklikler IBM MQ ' in davranışını etkileyebilir, ancak IBM MQtarafından doğrudan izlenemez. Bu tür değişikliklere örnek olarak, `mqs.ini`, `qm.ini`ve `mqclient.ini`yapılandırma dosyalarında yapılan değişiklikler, kuyruk yöneticilerinin yaratılması ve silinmesi, kullanıcı çıkış programları gibi ikili dosyaların kurulması ve dosya izinlerinde yapılan değişiklikler verilebilir. Bu etkinlikleri izlemek için işletim sistemi düzeyinde çalışan araçları kullanmanız gerekir. Farklı araçlar kullanılabilir ve farklı işletim sistemleri için uygundur. *sudo*gibi ilişkili araçlar tarafından oluşturulan günlükleriniz de olabilir.

IBM MQ işletim denetimi
Kuyruk yöneticilerinin başlatılması ve durdurulması gibi etkinlikleri denetlemek için işletim sistemi araçlarını kullanmanız gerekebilir. Bazı durumlarda, IBM MQ özel işlemden geçirme olayları yayınlamak üzere yapılandırılabilir.

IBM MQ içindeki uygulama etkinliği
Kuyruk açma ve ileti koyma ve alma gibi uygulamaların işlemlerini denetlemek için IBM MQ ' i uygun olayları yayınlamak üzere yapılandırın.

Davetsiz misafir uyarıları
Güvenlik ihlallerini denetlemek için sisteminizi yetkilendirme olayları yayınlamak üzere yapılandırın. Kanal olayları, özellikle bir kanal beklenmedik bir şekilde sona ererse, etkinliği göstermek için de yararlı olabilir.

Denetim verilerinin yakalanmasının, görüntülenmesinin ve arşivlenmesinin planlanması

Gereksinim duyduğunuz öğelerin çoğu IBM MQ olay iletisi olarak raporlanır. Bu iletileri okuyabilen ve biçimleyebilen araçları seçmeniz gerekir. Uzun vadeli depolama ve çözümlemelerle ilgileniyorsanız, bunları veritabanı gibi yardımcı bir depolama mekanizmasına taşımanız gerekir. Bu iletileri işlemezseniz, bunlar olay kuyruğunda kalır ve büyük olasılıkla kuyruğu doldurur. Bazı olaylara dayalı olarak otomatik olarak işlem gerçekleştiren bir araç uygulamaya karar verebilirsiniz; örneğin, bir güvenlik hatası oluştuğunda uyarı yayınlamak.

Sisteminizin doğru şekilde yapılandırıldığının doğrulanması

IBM MQ Explorerile birlikte bir test kümesi sağlanır. Nesne tanımlamalarınızda sorun olup olmadığını denetlemek için bunları kullanın.

Ayrıca, sistem yapılandırmasının beklediğiniz gibi olup olmadığını da düzenli aralıklarla denetleyin. Komut ve yapılandırma olayları bir şey değiştiğinde rapor verebilse de, yapılandırmanın dökümünü almak ve bunu bilinen iyi bir kopyayla karşılaştırmak da yararlıdır.

Topolojiye göre planlama güvenliği

Bu bölüm, kanallar, kuyruk yöneticisi kümeleri, yayınlama/abone olma ve çok hedefli uygulamalar ve güvenlik duvarı kullanıldığında belirli durumlarda güvenliği kapsar.

Ek bilgi için aşağıdaki alt konulara bakın:

Kanal yetkilendirmesi

Bir kanal aracılığıyla ileti gönderdiğinizde ya da aldığınızda, çeşitli IBM MQ kaynaklarına erişim sağlamanız gerekir. Message Channel Agents (MCA), kuyruk yöneticileri arasında ileti taşıyan ve bu nedenle doğru çalışması için çeşitli IBM MQ kaynaklarına erişim gerektiren IBM MQ uygulamalarıdır.

MCA ' lara ilişkin PUT zamanında ileti almak için, MCA ile ilişkili kullanıcı kimliğini ya da iletiyle ilişkili kullanıcı kimliğini kullanabilirsiniz.

CONNECT sırasında, **CHLAUTH** kanal kimlik doğrulama kayıtlarını kullanarak, bildirilen kullanıcı kimliğini diğer bir kullanıcıyla eşleyebilirsiniz.

IBM MQiçinde kanallar TLS desteğiyle korunabilir.

MCAUSER özniteliğinin kullanılmadığı gönderen kanalı dışında, gönderme ve alma kanallarıyla ilişkili kullanıcı kimlikleri için aşağıdaki kaynaklara erişim gerekir:

- Gönderen kanalla ilişkilendirilen kullanıcı kimliği, kuyruk yöneticisine, iletim kuyruğuna, gelmeyen iletiler kuyruğuna ve kanal çıkışlarının gerektirdiği diğer kaynaklara erişim gerektirir.
- Alıcı kanalının MCAUSER kullanıcı kimliği için *+ setall* yetkisi gerekir. Bunun nedeni, alıcı kanalının uzak gönderen kanalından aldığı verileri kullanarak, tüm bağlam alanları da içinde olmak üzere, tüm MQMD

' yi yaratması gerekmesi olabilir. Bu nedenle kuyruk yöneticisi, bu etkinliği gerçekleştiren kullanıcının *+ setall* yetkisine sahip olmasını gerektirir. Bu *+ setall* yetkisi kullanıcıya aşağıdakiler için verilmelidir:

  - Alıcı kanalının iletileri geçerli bir şekilde koyduğu tüm kuyruklar.
  - Kuyruk yöneticisi nesnesi. Daha fazla bilgi için bkz. Bağlam için yetkilendirmeler.

- Kaynak kullanıcının COA rapor iletisi istediği bir alıcı kanalın MCAUSER kullanıcı kimliği, rapor iletisini döndüren iletim kuyruğunda *+ passid* yetkisi gerektirir. Bu yetki olmadan AMQ8077 hata iletileri günlüğe kaydedilir.
- Alan kanalla ilişkilendirilen kullanıcı kimliğiyle, kuyruklara ileti koymak için hedef kuyrukları açabilirsiniz. Bu, İleti Kuyruklama Arabirimi 'ni (MQI) içerir; bu nedenle, IBM MQ Nesne Yetkilisi Yöneticisi 'ni (OAM) kullanmıyorsanız ek erişim denetimi denetimleri yapmanız gerekebilir. Yetki denetimlerinin MCA ile ilişkili kullanıcı kimliği (bu konuda açıklandığı gibi) için mi, yoksa iletiyle ilişkili kullanıcı kimliği (MQMD UserIdentifier alanından) için mi gerçekleştirileceğini belirtebilirsiniz.

  Geçerli olduğu kanal tipleri için, kanal tanımının **PUTAUT** parametresi, bu denetimler için hangi kullanıcı kimliğinin kullanılacağını belirtir.

  - Kanal varsayılan olarak, tam denetim haklarına sahip olan ve özel yetki gerektirmeyen kuyruk yöneticisinin hizmet hesabını kullanır.
  - Sunucu bağlantısı kanalları söz konusu olduğunda, yönetim bağlantıları varsayılan olarak CHLAUTH kuralları tarafından engellenir ve belirtik yetkilendirme gerektirir.
  - Alıcı, istekte bulunan ve küme alıcısı tipindeki kanallar, denetimci bu erişimi sınırlamak için gereken adımları atmadıkça, bitişik herhangi bir kuyruk yöneticisi tarafından yerel denetime izin verir.
  - Bir alıcı kanalının MCAUSER kullanıcı kimliği için *dsp* ve *ctrlx* yetkisi verilmesi gerekmez.
  - IBM MQ 8.0.0 Fix Pack 4' den önce, IBM MQ yönetim ayrıcalıklarına sahip olmayan bir kullanıcı kimliği kullanırsanız, kanalın çalışması için o kullanıcı kimliği için kanala **dsp** ve **ctrlx** yetkisi vermeniz gerekir.

    IBM MQ 8.0.0 Fix Pack 4' den bir kanal kendisini yeniden eşzamanladığında ve sıra numaralarını düzelttiğinde yetki denetimi olmaz.

    Ancak, el ile RESET CHANNEL komutu verilmesi, tüm yayınlarda **+dsp** ve **+ctrlx** komutlarını gerektirir.

    Uyarı: İleti toplu iş doğrulaması için bir kanal sıfırlaması gerektiğinde, IBM MQ kanalı sorgulamaya çalışır; bu da **+dsp** yetkisi gerektirir.
  - MCAUSER özniteliği SDR kanal tipi için kullanılmıyor.
  - İletiyle ilişkili kullanıcı kimliğini kullanırsanız, kullanıcı kimliği uzak bir sistemden olabilir. Bu uzak sistem kullanıcı kimliği hedef sistem tarafından tanınmalıdır. Aşağıdaki komutlar, uzak sistemden bir kullanıcı kimliğine yetki vermek için verebileceğiniz komut tipine örnektir:

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect +inq +setall
```

```
setmqaut -m QMgrName -t chl -n Profile -g GroupName +dsp +ctrlx
```

Burada *Profil* bir kanaldır.

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

Burada *Profil* , ayarlanmışsa, bir gönderilmeyen ileti kuyruğudur.

```
setmqaut -m QMgrName -t q -n Profile -g GroupName +put +setall
```

Burada *Tanıtım* , yetkili kuyrukların bir listesidir.

Uyarı: Komut Kuyruğuna ya da diğer hassas sistem kuyruklarına ileti yerleştirmek için bir kullanıcı kimliği yetkilendirirken dikkatli olun.

MCA ile ilişkili kullanıcı kimliği MCA tipine bağlıdır. İki tip MCA vardır:

Arayan MCA
Kanal başlatan MCA ' lar. Çağıran MCA ' lar tek tek işlemler olarak, kanal başlatıcısının iş parçacıkları olarak ya da bir süreç havuzunun iş parçacıkları olarak başlatılabilir. Kullanılan kullanıcı kimliği, üst işlemle (kanal başlatıcısı) ilişkili kullanıcı kimliği ya da MCA ' yı başlatan işlemle ilişkilendirilmiş kullanıcı kimliğidir.

Yanıt Veren MCA
Yanıt veren MCA 'lar, çağıran MCA tarafından bir isteğin sonucu olarak başlatılan MCA' lardır. Yanıt veren MCA ' ları tek tek işlemler olarak, dinleyicinin iş parçacıkları olarak ya da bir süreç havuzunun iş parçacıkları olarak başlatılabilir. Kullanıcı kimliği aşağıdaki tiplerden herhangi biri olabilir (bu tercih sırasıyla):

1. APPC ' de çağıran MCA, yanıt veren MCA için kullanılacak kullanıcı kimliğini belirtebilir. Bu, ağ kullanıcı kimliği olarak adlandırılır ve yalnızca tek tek işlemler olarak başlatılan kanallar için geçerlidir. Kanal tanımının USERID parametresini kullanarak ağ kullanıcı kimliğini ayarlayın.
2. **USERID** parametresi kullanılmazsa, yanıt veren MCA 'nın kanal tanımı, MCA' nın kullanması gereken kullanıcı kimliğini belirtebilir. Kanal tanımlamasının **MCAUSER** değiştirgesini kullanarak kullanıcı kimliğini ayarlayın.
3. Kullanıcı kimliği önceki (iki) yöntemlerden biri tarafından ayarlanmamışsa, MCA ' yı başlatan işlemin kullanıcı kimliği ya da üst işlemin (dinleyici) kullanıcı kimliği kullanılır.

İlgili kavramlar
“Kanal kimlik doğrulama kayıtları” sayfa 50
Kanal düzeyinde bağlanan sistemlere verilen erişim üzerinde daha kesin denetim sağlamak için kanal kimlik doğrulama kayıtlarını kullanabilirsiniz.

İlgili başvurular
Kanal kimlik doğrulama kaydı özellikleri

Kanal başlatıcı tanımlarının korunması
Yalnızca mqm grubunun üyeleri kanal başlatıcılarını işleyebilirler.

IBM MQ kanal başlatıcıları IBM MQ nesne değildir; bunlara erişim OAM tarafından denetlenmez. IBM MQ , kullanıcı kimlikleri mqm grubunun bir üyesi değilse, kullanıcıların ya da uygulamaların bu nesneleri işlemesine izin vermez. **StartChannelInitiator**PCF komutunu veren bir uygulamanız varsa, PCF iletisinin ileti tanımlayıcısında belirtilen kullanıcı kimliği, hedef kuyruk yöneticisindeki mqm grubunun bir üyesi olmalıdır.

Bir kullanıcı kimliği, Escape PCF komutuyla eşdeğer MQSC komutlarını yayınlamak ya da `runmqsc` komutunu dolaylı kipte kullanmak için hedef makinedeki mqm grubunun bir üyesi de olmalıdır.

İletim kuyrukları
Kuyruk yöneticileri uzak iletileri otomatik olarak bir iletim kuyruğuna yerleştirir; bu işlem için özel bir yetki gerekmez.

Ancak, bir iletiyi doğrudan bir iletim kuyruğuna koymanız gerekirse, bu özel yetki gerektirir; bkz. Çizelge 12 sayfa 131.

Kanal çıkışları
Kanal kimlik doğrulama kayıtları uygun değilse, ek güvenlik için kanal çıkışlarını kullanabilirsiniz. Güvenlik çıkışı, iki güvenlik çıkış programı arasında güvenli bir bağlantı oluşturur. Bir program gönderen ileti kanalı aracısı (MCA) için, diğeri ise alan MCA içindir.

Kanal çıkışlarına ilişkin ek bilgi için bkz. “Kanal çıkış programları” sayfa 108 .

SSL/TLS ile kanalları koruma
IBM MQ içindeki TLS desteği, kuyruk yöneticisi kimlik doğrulama bilgileri nesnesini ve çeşitli MQSC komutlarını kullanır. Dijital sertifika kullanımınızı da göz önünde bulundurmanız gerekir.

Dijital sertifikalar ve anahtar havuzları

Kuyruk yöneticisi sertifika etiketi özniteliğini ayarlamak iyi bir uygulamadır (**CERTLABL**) kanalların çoğunluğu için kullanılacak kişisel sertifikanın adını girin ve farklı sertifikalar gerektiren kanallarda sertifika etiketini ayarlayarak, kural dışı durumlar için sertifikayı geçersiz kılın.

Kuyruk yöneticisinde varsayılan sertifika kümesinden farklı sertifikalara sahip birçok kanala gereksinim duyarsanız, kanalları birkaç kuyruk yöneticisi arasında bölmeyi ya da farklı bir sertifika sunmak için kuyruk yöneticisinin önünde bir MQIPT yetkili sunucusu kullanmayı düşünmelisiniz.

Her kanal için farklı bir sertifika kullanabilirsiniz, ancak bir anahtar havuzunda çok fazla sertifika saklıyorsanız, TLS kanalları başlatılırken performansın etkilenmesini bekleyebilirsiniz. Bir anahtar havuzundaki sertifika sayısını 50 'den az tutmayı deneyin ve IBM Global Security Kit (GSKit) performansı daha büyük anahtar havuzlarıyla keskin bir şekilde düştükçe 100 'ü maksimum olarak değerlendirin.

Aynı kuyruk yöneticisinde birden çok sertifikayı kullanmanıza izin verilmesi, aynı kuyruk yöneticisinde birden çok CA sertifikası kullanılma olasılığını artırır. Bu, ayrı sertifika yetkilileri tarafından verilen sertifikalar için sertifika Konusu Ayırt Edici Adı ad alanı çakışmaları olasılığını artırır.

Profesyonel sertifika yetkilileri daha dikkatli olmalarına karşın, kurum içi sertifika yetkilileri genellikle net adlandırma kurallarından yoksundur ve bir CA ile diğeri arasında istenmeyen eşleşmeler olabilir.

Konu Ayırt Edici Adına ek olarak sertifika veren ayırt edici adını da kontrol etmelisiniz. Bunu yapmak için bir kanal kimlik doğrulaması SSLPEERMAP kaydı kullanın ve **SSLPEER** ve **SSLCERTI** alanlarını sırasıyla Subject DN ve Issuer DN ile eşleşecek şekilde ayarlayın.

Kendinden imzalı ve CA imzalı sertifikalar

Hem uygulamanızı geliştirip test ederken hem de üretimde kullanımı için dijital sertifika kullanımınızı planlamanız önemlidir. Kuyruk yöneticilerinizin ve istemci uygulamalarınızın kullanımına bağlı olarak CA imzalı sertifikaları ya da kendinden imzalı sertifikaları kullanabilirsiniz.

CA imzalı sertifikalar
Üretim sistemleri için sertifikalarınızı güvenilir bir sertifika kuruluşundan (CA) edinin. Bir dış sertifika kuruluşundan sertifika aldığınızda, hizmet için ödeme yapınız.

kendinden imzalı sertifikalar
Uygulamanızı geliştirirken, platforma bağlı olarak, yerel bir CA tarafından verilen kendinden imzalı sertifikaları ya da sertifikaları kullanabilirsiniz:

ALW AIX, Linux, and Windows sistemlerinde kendinden imzalı sertifikaları kullanabilirsiniz. Yönergeler için bkz. “AIX, Linux, and Windows üzerinde kendinden onaylı kişisel sertifika oluşturma” sayfa 303.

IBM i IBM i sistemlerinde, yerel CA tarafından imzalanan sertifikaları kullanabilirsiniz. Yönergeler için bkz. “IBM i üzerinde sunucu sertifikası isteme” sayfa 281 .

z/OS z/OSüzerinde, kendinden imzalı ya da yerel CA imzalı sertifikaları kullanabilirsiniz. Yönergeler için bkz. “z/OS üzerinde kendinden onaylı kişisel sertifika oluşturma” sayfa 331 ya da “z/OS üzerinde kişisel sertifika isteme” sayfa 331 .

Kendinden imzalı sertifikalar, aşağıdaki nedenlerden ötürü üretim kullanımı için uygun değildir:

- Kendinden imzalı sertifikalar geri alınamaz; bu, bir saldırganın özel bir anahtar açığa çıktıktan sonra kimlik sahteciliği yapmasına izin verebilir. Sertifika yetkilisi, ihlal edilen bir sertifikayı iptal edebilir ve bu da sertifikanın daha fazla kullanılmasını önler. Bu nedenle CA imzalı sertifikaların üretim ortamında kullanılması daha güvenlidir, ancak kendinden imzalı sertifikalar test sistemi için daha uygundur.
- Kendinden imzalı sertifikaların süresi hiçbir zaman dolmaz. Bu, bir test ortamında hem kullanışlı hem de güvenlidir, ancak bir üretim ortamında onları nihai güvenlik ihlallerine açık bırakır. Risk, kendinden imzalı sertifikaların iptal edilemediği gerçeğiyle birleştirilmiştir.
- Kendinden onaylı sertifika hem kişisel sertifika olarak hem de kök (ya da güven çıpası) CA sertifikası olarak kullanılır. Kendinden onaylı bir kişisel sertifikaya sahip bir kullanıcı, bunu diğer kişisel sertifikaları

imzalamak için kullanabilir. Genel olarak bu, bir CA tarafından verilen kişisel sertifikalar için geçerli değildir ve önemli bir güvenlik açığını temsil eder.

CipherSpecs ve dijital sertifikalar

Desteklenen tüm sayısal sertifika tipleriyle yalnızca desteklenen CipherSpecs alt kümesi kullanılabilir. Bu nedenle, dijital sertifikalarınız için uygun bir CipherSpec seçmeniz gerekir. Benzer şekilde, kuruluşunuzun güvenlik ilkesi belirli bir CipherSpec kullanılmasını gerektiriyorsa, uygun dijital sertifikaları edinmeniz gerekir.

CipherSpecs ile sayısal sertifikalar arasındaki ilişki hakkında daha fazla bilgi için bkz. “IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu” sayfa 46

Sertifika doğrulama ilkeleri

IETF RFC 5280 standardı, taklit saldırılarını önlemek için uyumlu uygulama yazılımının uygulaması gereken bir dizi sertifika doğrulama kuralını belirtir. Sertifika doğrulama kuralları kümesi, sertifika doğrulama ilkesi olarak bilinir. IBM MQiçindeki sertifika doğrulama ilkeleri hakkında daha fazla bilgi için bkz. “IBM MQ içindeki sertifika doğrulama ilkeleri” sayfa 44.

Sertifika iptal denetimine ilişkin planlama

Farklı sertifika yetkililerinin birden çok sertifikaya izin verilmesi, gereksiz ek sertifika iptali denetimine neden olabilir.

Özellikle, belirli bir CA 'dan (örneğin, bir AUTHINFO nesnesi ya da kimlik doğrulama bilgi kaydı (MQAIR) yapısı kullanarak bir iptal sunucusu kullanımını belirtik olarak yapılandırdıysanız, farklı bir CA' dan bir sertifika sunulduğunda iptal denetimi başarısız olur.

Belirtik sertifika iptal sunucusu yapılandırmasını önleyin. Bunun yerine, her sertifikanın bir sertifika uzantısında kendi iptal sunucusu konumunu (örneğin, CRL Dağıtım Noktası ya da OCSP AuthorityInfoErişimi) içerdiği örtük denetlemeyi etkinleştirmeniz gerekir.

Daha fazla bilgi için bkz. OCSPCheckExtensions ve CDPCheckExtensions.

TLS desteği için komutlar ve öznitelikler

TLS (Transport Layer Security; İletim Katmanı Güvenliği) protokolü, gizlice dinleme, kurcalama ve kimliğe bürünmeye karşı koruma ile kanal güvenliği sağlar. TLS için IBM MQ desteği, kanal tanımında belirli bir kanalın TLS güvenliğini kullandığını belirtmenize olanak sağlar. Kullanmak istediğiniz şifreleme algoritması gibi, istediğiniz güvenlik tipinin ayrıntılarını da belirtebilirsiniz.

- Aşağıdaki MQSC komutları TLS ' yi destekler:

  AUTHINFO DEğIşTIR
  Bir kimlik doğrulama bilgi nesnesinin özniteliklerini değiştirir.

  AUTHINFO TANıMLAYıN
  Bir kimlik doğrulama bilgileri nesnesi oluşturur.

  AUTHINFO ÖĞESINI SIL
  Kimlik doğrulama bilgileri nesnesini siler.

  AUTHINFO BILGILERINI GöRüNTüLE
  Belirli bir kimlik doğrulama bilgileri nesnesine ilişkin öznitelikleri görüntüler.

- Aşağıdaki kuyruk yöneticisi parametreleri TLS ' yi destekler:

  CERTLABL
  Kullanılacak kişisel sertifika etiketini tanımlar.

  V 9.3.0 V 9.3.0 **KEYRPWD**
  AIX, Linux, and Windows sistemlerinde, IBM MQ ' in anahtar havuzuna erişmek için kullandığı parolayı tanımlar. Bu alan, parola koruma sistemi kullanılarak şifrelenir.

SSLCRLNL
SSLCRLNL özniteliği, gelişmiş TLS sertifika denetimine izin vermek üzere sertifika iptal konumlarını sağlamak için kullanılan kimlik doğrulama bilgileri nesnelerinin ad bilgilerini belirtir.

SSLCRYP
AIX, Linux, and Windows sistemlerinde **SSLCryptoHardware** kuyruk yöneticisi özniteliğini ayarlar. Bu öznitelik, sisteminizde bulunan şifreleme donanımını yapılandırmak için kullanabileceğiniz parametre dizgisinin adıdır.

SLEV
TLS kullanan bir kanal TLS bağlantısı kuramazsa TLS olay iletisinin raporlanıp raporlanmayacağını belirler.

SSLFIPS
Şifreleme, kriptografik donanım yerine IBM MQ içinde gerçekleştirilirse yalnızca FIPS onaylı algoritmaların kullanılıp kullanılmayacağını belirtir. Şifreleme donanımı yapılandırıldıysa, donanım ürünü tarafından sağlanan şifreleme modülleri kullanılır ve bunlar belirli bir düzeyde FIPS onaylı olabilir. Bu, kullanılmakta olan donanım ürününe bağlıdır.

SSLKEYR
AIX, Linux, and Windows sistemlerinde, bir anahtar havuzunu bir kuyruk yöneticisiyle ilişkilendirir. GSKit , AIX, Linux, and Windows sistemlerinde TLS güvenliğini kullanmanızı sağlar.

SSLRKEYC
Gizli anahtar yeniden anlaşılmadan önce TLS etkileşimi içinde gönderilecek ve alınacak bayt sayısı. Bayt sayısı, MCA tarafından gönderilen denetim bilgilerini içerir.

- Aşağıdaki kanal parametreleri TLS ' yi destekler:

CERTLABL
Kullanılacak kişisel sertifika etiketini tanımlar.

SSLCAUTH
IBM MQ ' in TLS istemcisinden bir sertifikayı gerektirip doğrulayıp gerektirmediğini tanımlar.

SSLCIPH
Şifreleme gücünü ve işlevini (CipherSpec) belirtir; örneğin, TLS_RSA_WITH_AES_128_CBC_SHA. CipherSpec , kanalın her iki ucunda da eşleşmelidir.

SSLPEER
İzin verilen iş ortaklarının ayırt edici adını (benzersiz tanıtıcı) belirtir.

Bu bölümde, kimlik doğrulama bilgileri nesnesini desteklemek için **setmqaut**, **dspmqaut**, **dmpmqaut**, **rcrmqobj**, **rcdmqimg**ve **dspmqfls** komutları açıklanmaktadır. Ayrıca, AIX, Linux, and Windowsüzerinde sertifikaları yönetmeye ilişkin **runmqckm** (iKeycmd) ve **runmqakm** komutlarını da açıklar. Aşağıdaki bölümlere bakın:

- setmqaut
- dspmqaut
- dmpmqaut
- rcrmqobj
- rcdmqimg
- dspmqfls
- Anahtarların ve sertifikaların yönetilmesi

TLS kullanan kanal güvenliğine genel bakış için bkz.

- “IBM MQ içinde TLS güvenlik iletişim kuralları” sayfa 24

TLS ile ilişkili MQSC komutlarının ayrıntıları için bkz.

- ALTER AUTHINFO
- AUTHINFO TANıMLAYıN
- YETKI BILGISINI SIL

- AUTHINFO GöRüNTüLE

TLS ile ilişkili PCF komutlarının ayrıntıları için bkz.

- Kimlik Doğrulama Bilgileri Nesnesini Değiştir, Kopyala ve Oluştur
- Kimlik Doğrulama Bilgileri Nesnesini Sil
- Kimlik Doğrulama Bilgileri Nesnesi

z/OS *IBM MQ for z/OS sunucu bağlantısı kanalı*

IBM MQ for z/OS SVRCONN kanalı, kanal kimlik doğrulaması uygulanmadan ya da TLS kullanarak bir güvenlik çıkışı eklenmeden güvenli değildir. SVRCONN kanallarının varsayılan olarak tanımlanmış bir güvenlik çıkışı yoktur.

Güvenlik endişeleri

SVRCONN kanalları başlangıçta tanımlandığı gibi güvenli değildir, SYSTEM.DEF.SVRCONN . Bir SVRCONN kanalının güvenliğini sağlamak için SET CHLAUTH komutunu kullanarak kanal kimlik doğrulamasını ayarlamanız ya da bir güvenlik çıkışı kurmanız ve TLS ' yi uygulamanız gerekir.

Genel kullanıma açık bir örnek güvenlik çıkışı kullanmalı, kendiniz bir güvenlik çıkışı yazmalı ya da bir güvenlik çıkışı satın almalısınız.

Kendi SVRCONN kanal güvenlik çıkışınızı yazmak için iyi bir başlangıç noktası olarak kullanabileceğiniz birkaç örnek vardır.

IBM MQ for z/OSiçinde, hlq.SCSQC37S kitaplığınızdaki CSQ4BCX3 üyesi, C dilinde yazılmış bir güvenlik çıkışı örneğidir. Örnek CSQ4BCX3 , hlq.SCSQAUTH kitaplığınızda önceden derlenmiş olarak gönderilir.

CSQ4BCX3 örnek çıkışını, derlenmiş hlq.SCSQAUTH(CSQ4BCX3) üyesini CHIN Proc içindeki CSQXLIB DD ' ye ayrılmış bir yükleme kitaplığına kopyalayarak uygulayabilirsiniz. CHIN ' in yükleme kitaplığının "Program Denetimli" olarak ayarlanmasını gerektirdiğini unutmayın.

SVRCONN kanalınızı, güvenlik çıkışı olarak CSQ4BCX3 değerini belirecek şekilde değiştirin.

Bir istemci bu SVRCONN kanalını kullanarak bağlandığında, CSQ4BCX3 , MQCD 'deki **RemoteUserIdentifier** ve **RemotePassword** çiftini ya da IBM MQ for z/OS 9.1.4' den MQCSP ' deki **CSPUserIdPtr** ve **CSPPasswordPtr** çiftini kullanarak kimlik doğrulaması gerçekleştirir. Kimlik doğrulama başarılı olursa, iş parçacığının kimlik bağlamını değiştirerek **RemoteUserIdentifier** dosyasını **MCAUserIdentifier**içine kopyalar.

Long Term Support ve Continuous Delivery öncesinde IBM MQ for z/OS 9.1.4için, bir istemci bu SVRCONN kanalını kullanarak bağlandığında, CSQ4BCX3 MQCD ' deki **RemoteUserIdentifier** ve **RemotePassword** çiftini kullanarak kimlik doğrulaması gerçekleştirir. Kimlik doğrulama başarılı olursa, iş parçacığının kimlik bağlamını değiştirerek **RemoteUserIdentifier** dosyasını **MCAUserIdentifier**içine kopyalar.

Bir IBM MQ Java istemcisi yazıyorsanız, kullanıcıyı sorgulamak ve MQEnvironment.userID ve MQEnvironment.passwordöğelerini ayarlamak için açılır pencereleri kullanabilirsiniz. Bağlantı kurulduğunda bu değerler geçirilir.

İşlevsel bir güvenlik çıkışa sahip olduğunuzda, kullanıcı kimliği ve parolanın, sonraki IBM MQ iletilerinin içeriği gibi, bağlantı kurulduğunda ağ üzerinden düz metin olarak iletilmesi de ilginçtir. IBM MQ iletilerinin içeriğinin yanı sıra bu ilk bağlantı bilgilerini şifrelemek için TLS ' yi kullanabilirsiniz.

Örnek

IBM MQ Explorer SVRCONN kanalı SYSTEM.ADMIN.SVRCONN aşağıdaki adımları tamamlayın:

1. hlq.SCSQAUTH(CSQ4BCX3) dosyasını, CHINIT Proc içindeki CSQXLIB DD ' ye ayrılmış bir yükleme kitaplığına kopyalayın.
2. Yükleme kitaplığının Program Denetimli olduğunu doğrulayın.
3. SYSTEM ADMIN.SVRCONN dosyasını CSQ4BCX3güvenlik çıkışını kullanacak şekilde değiştirin.

4. IBM MQ Explorericinde z/OS Kuyruk Yöneticisi adını sağ tıklatın, **Bağlantı Ayrıntıları** > **Özellikler** > **Kullanıcı kimliği** seçeneğini belirleyin ve z/OS kullanıcı kimliğinizi girin.
5. Bir parola girerek z/OS Kuyruk Yöneticisine bağlanın.

Ek bilgiler

CSQ4BCX3 çıkışının Program Denetimli bir ortamda çalışması için, CHIN adres alanına yüklenen her şeyin Program Denetimli bir kitaplıktan yüklenmesi gerekir; örneğin, STEPLIB içindeki tüm kitaplıklar ve CSQXLIB DD ' deki tüm kitaplıklar. Bir yükleme kitaplığını Program Denetimli sorun RACF komutları olarak ayarlamak için. Aşağıdaki örnekte yükleme kitaplığı adı MY.TEST.LOADLIB.

```
RALTER PROGRAM * ADDMEM('MY.TEST.LOADLIB'//NOPADCHK)
SETROPTS WHEN(PROGRAM)REFRESH
```

SVRCONN kanalını CSQ4BCX3gerçekleştirecek şekilde değiştirmek için aşağıdaki IBM MQ komutunu verin:

```
ALTER CHANNEL(SYSTEM ADMIN.SVRCONN) CHLTYPE(SVRCONN) SCYEXIT(CSQ4BCX3)
```

Yukarıdaki örnekte, kullanılmakta olan SVRCONN kanal adı SYSTEM ADMIN.SVRCONN' dur.

Kanal çıkışlarına ilişkin ek bilgi için bkz. “Kanal çıkış programları” sayfa 108 .

İlgili görevler

z/OS üzerinde kanal çıkış programları yazılıyor

SNA LU 6.2 güvenlik hizmetleri

SNA LU 6.2 , oturum düzeyinde şifreleme, oturum düzeyinde kimlik denetimi ve etkileşim düzeyinde kimlik doğrulaması sunar.

Not: Bu konu derlemi, SNA (Sistem Ağ Mimarisi) ile ilgili temel bir anlayışa sahip olduğunuz varsayılmıştır. Bu bölümde atıfta bulunulan diğer belgeler, ilgili kavramlara ve terminolojiye kısa bir giriş içerir. SNA ' ya daha kapsamlı bir teknik giriş gerekiyorsa, bkz. *Systems Network Architecture Technical Overview*, GC30-3073.

SNA LU 6.2 üç güvenlik hizmeti sağlar:

- Oturum düzeyinde şifreleme
- Oturum düzeyi kimlik doğrulaması
- Etkileşim düzeyi kimlik doğrulaması

Oturum düzeyinde şifreleme ve oturum düzeyinde kimlik doğrulaması için, SNA *Veri Şifreleme Standardı (DES)* algoritmasını kullanır. DES algoritması, verileri şifrelemek ve verilerin şifresini çözmek için simetrik bir anahtar kullanan bir blok şifreleme algoritmasıdır. Hem blok hem de anahtar 8 bayt uzunluğundadır.

Oturum düzeyinde şifreleme

Oturum düzeyinde şifreleme , DES algoritmasını kullanarak oturum verilerini şifreler ve şifresini çözer. Bu nedenle, SNA LU 6.2 kanallarında bağlantı düzeyinde gizlilik hizmeti sağlamak için kullanılabilir.

Mantıksal birimler (LU), zorunlu (ya da gerekli) veri şifrelemesi, seçmeli veri şifrelemesi ya da veri şifrelemesi sağlamaz.

Zorunlu bir şifreleme oturumunda, bir LU giden tüm veri isteği birimlerini şifreler ve tüm gelen veri isteği birimlerinin şifresini çözer.

Bir *seçmeli şifreleme oturumunda*, bir LU yalnızca gönderen hareket programı (TP) tarafından belirtilen veri isteği birimlerini şifreler. Gönderen LU, istek üstbilgisinde bir gösterge ayarlanarak verilerin şifrelendiğini gösterir. Bu göstergeyi denetleyerek, alan LU hangi istek birimlerinin şifresini çözülebileceğini, bunları alan TP ' ye aktarmadan önce belirleyebilir.

Bir SNA ağında, IBM MQ MCA ' ları hareket programlarıdır. MCA ' lar gönderdikleri veriler için şifreleme istemezler. Bu nedenle seçici veri şifreleme bir seçenek değildir; bir oturumda yalnızca zorunlu veri şifrelemesi ya da veri şifrelemesi yapılamaz.

Zorunlu veri şifrelemenin nasıl uygulanacağına ilişkin bilgi için SNA altsisteminize ilişkin belgelere bakın. z/OSüzerinde Üçlü DES 24 baytlık şifreleme gibi platformunuzda kullanılabilecek daha güçlü şifreleme biçimlerine ilişkin bilgi için aynı belgelere bakın.

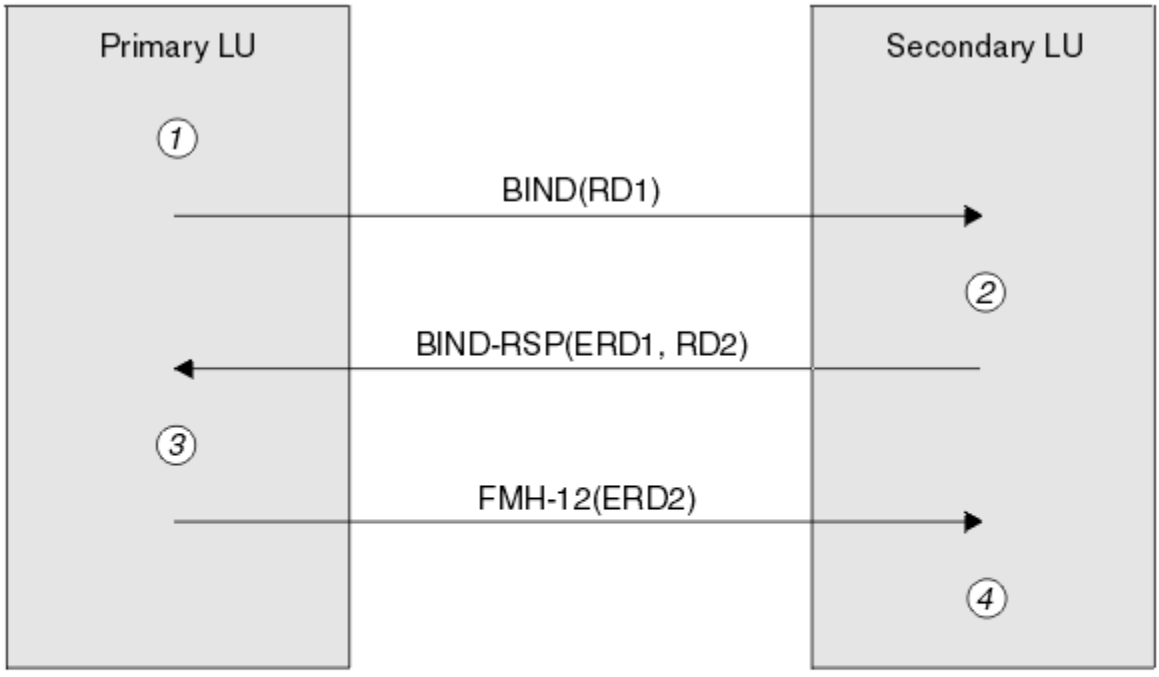
Oturum düzeyinde şifrelemeyle ilgili ek bilgi için bkz. *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808.

Oturum düzeyi kimlik doğrulaması
Oturum düzeyinde kimlik doğrulama , iki LU 'nun bir oturumu etkinleştirirken birbirlerinin kimliğini doğrulamasını sağlayan oturum düzeyinde bir güvenlik protokolüdür. *LU-LU doğrulaması*olarak da bilinir.

Bir LU, ağdaki bir sisteme ilişkin "ağ geçidi" olduğu için, belirli koşullarda bu kimlik doğrulama düzeyinin yeterli olduğunu düşünebilirsiniz. Örneğin, kuyruk yöneticinizin denetimli ve güvenilir bir ortamda çalışan uzak bir kuyruk yöneticisiyle ileti değiş tokuşu yapması gerekiyorsa, LU kimliği doğrulandıktan sonra uzak sistemin geri kalan bileşenlerinin kimliklerine güvenmeye hazır olabilirsiniz.

Oturum düzeyinde kimlik doğrulaması, ortak parolasını doğrulayan her LU tarafından gerçekleştirilir. Her LU çifti arasında bir parola oluşturulduğundan, parolaya *LU-LU parolası* adı verilir. LU-LU parolasının oluşturulma şekli, uygulamaya bağlıdır ve SNA kapsamı dışındadır.

Şekil 12 sayfa 120 , oturum düzeyinde kimlik doğrulamaya ilişkin akışları gösterir.



Legend:

BIND	=	BIND request unit
BIND-RSP	=	BIND response unit
ERD	=	Encrypted random data
FMH-12	=	Function Management Header 12
RD	=	Random data

Şekil 12. Oturum düzeyinde kimlik doğrulaması için akışlar

Oturum düzeyinde kimlik doğrulamaya ilişkin protokol aşağıdaki gibidir. Yordamdaki numaralar, Şekil 12 sayfa 120içindeki numaralara karşılık gelir.

1. Birincil LU rasgele bir veri değeri (RD1) oluşturur ve BIND isteğinde ikincil LU ' ya gönderir.
2. İkincil LU rasgele verilerle BIND isteğini aldığında, anahtar olarak LU-LU parolasının kopyasıyla DES algoritmasını kullanarak verileri şifreler. İkincil LU daha sonra ikinci bir rasgele veri değeri (RD2) oluşturur ve şifrelenmiş verilerle (ERD1) BIND yanıtındaki birincil LU ' ya gönderir.
3. Birincil LU BIND yanıtını aldığında, başlangıçta oluşturduğu rasgele verilerden şifrelenmiş verilerin kendi sürümünü hesaplar. Bunu, anahtar olarak LU-LU parolasının kopyasıyla birlikte DES algoritmasını kullanarak yapar. Daha sonra, sürümünü BIND yanıtında aldığı şifrelenmiş verilerle karşılaştırır. İki değer aynıysa, birincil LU, ikincil LU 'nun parolasının aynı olduğunu ve ikincil LU' nun kimliğinin doğrulandığını bilir. İki değer eşleşmezse, birincil LU oturumu sonlandırır.

   Birincil LU, BIND yanıtında aldığı rasgele verileri şifreler ve şifrelenmiş verileri (ERD2) İşlev Yönetimi Üstbilgisi 12 'deki (FMH-12) ikincil LU ' ya gönderir.
4. İkincil LU FMH-12' yi aldığında, oluşturduğu rasgele verilerden şifrelenmiş verilerin kendi sürümünü hesaplar. Daha sonra, sürümünü FMH-12içinde aldığı şifrelenmiş verilerle karşılaştırır. İki değer aynıysa, birincil LU ' nun kimliği doğrulanır. İki değer eşleşmezse, ikincil LU oturumu sonlandırır.

İkincil LU, ortadaki saldırılarda insana karşı daha iyi koruma sağlayan protokolün geliştirilmiş bir sürümünde, RD1, RD2'den bir DES İleti Kimlik Doğrulama Kodu (MAC) ve ikincil LU' nun tam olarak nitelenmiş adını anahtar olarak LU-LU parolasının kopyasını kullanarak hesaplar. İkincil LU, MAC 'i ERD1yerine BIND yanıtında birincil LU' ya gönderir.

Birincil LU, ikincil LU ' nun kimliğini, BIND yanıtında alınan MAC ile karşılaştırdığı MAC sürümünü hesaplayarak doğrular. Birincil LU daha sonra RD1 ve RD2'den ikinci bir MAC hesaplar ve MAC' i ERD2yerine FMH-12 'deki ikincil LU' ya gönderir.

İkincil LU, FMH-12'de alınan MAC ile karşılaştırdığı ikinci MAC sürümünü hesaplayarak birincil LU' nun kimliğini doğrular.

Oturum düzeyinde kimlik doğrulamasını nasıl yapılandıracağınıza ilişkin bilgi için SNA altsisteminize ilişkin belgelere bakın. Oturum düzeyi kimlik doğrulamasıyla ilgili daha fazla bilgi için bkz. *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808.

Etkileşim düzeyi kimlik doğrulaması
Yerel bir TP ortak TP ile etkileşim ayırmayı denediğinde, yerel LU ortak LU 'ya ortak TP' yi bağlamasını isteyen bir ekleme isteği gönderir. Belirli koşullar altında, ekleme isteği, ortak LU 'nun yerel TP' yi doğrulamak için kullanabileceği güvenlik bilgilerini içerebilir. Bu, *etkileşim düzeyi kimlik doğrulaması*ya da *son kullanıcı doğrulaması*olarak bilinir.

Aşağıdaki konularda, IBM MQ ' in etkileşim düzeyinde kimlik doğrulaması için nasıl destek sağladığı açıklanmaktadır.

Etkileşim düzeyi kimlik doğrulamasıyla ilgili ek bilgi için bkz. *Systems Network Architecture LU 6.2 Reference: Peer Protocols*, SC31-6808.

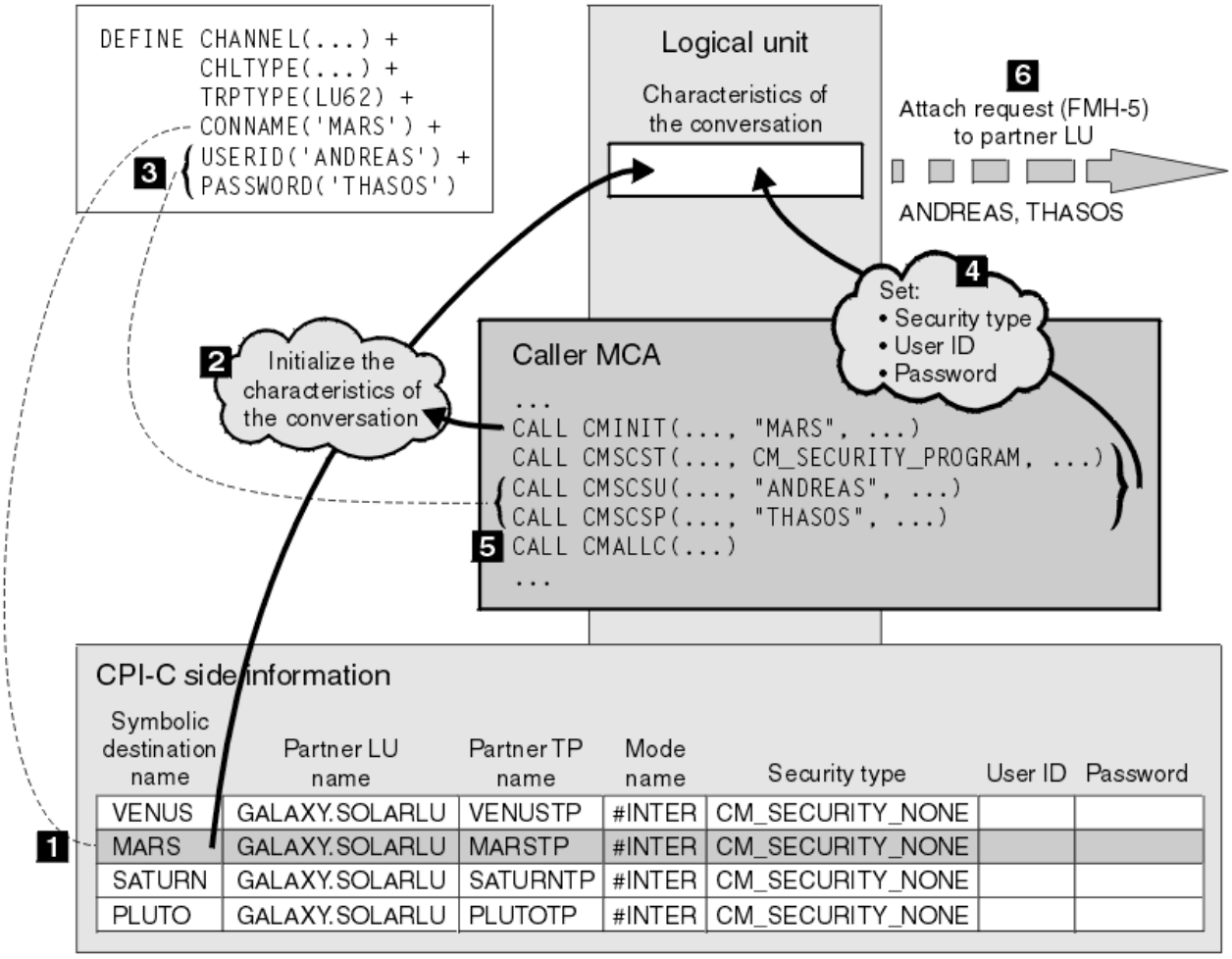
z/OS' e özgü bilgi için bkz. z/OS MVS Planning: APPC/MVS Management.

CPI-C ile ilgili ek bilgi için CPI İletişimlerinin Kullanılmasıbaşlıklı konuya bakın.

APPC/MVS TP Conversation Callable Services ile ilgili ek bilgi için APPC/MVS TP Conversation Callable Servicesbaşlıklı konuya bakın.

Multi *Multiplatforms üzerinde etkileşim düzeyi kimlik doğrulaması için destek*
Çoklu platformlarda etkileşim düzeyi kimlik doğrulamasının nasıl çalıştığına ilişkin bir genel bakış elde etmek için bu konuyu kullanın.

Multiplatforms üzerinde etkileşim düzeyi kimlik doğrulaması desteği Şekil 13 sayfa 122içinde gösterilmiştir. Çizgedeki sayılar, aşağıdaki açıklamadaki sayılara karşılık gelir.



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		

Şekil 13. Etkileşim düzeyi kimlik doğrulaması için IBM MQ desteği

Multiplatforms üzerinde, bir MCA, SNA ağı üzerinden ortak MCA ile iletişim kurmak için Common Programming Interface Communications (CPI-C) çağrılarını kullanır. Bir kanalın çağıran ucundaki kanal tanımında, CONNAME parametresinin değeri, bir CPI-C yan bilgi girişini (1) tanıtan sembolik bir hedef adıdır. Bu giriş şunları belirtir:

- Ortak LU ' nun adı
- Yanıt veren MCA olan ortak TP ' nin adı
- Etkileşim için kullanılacak kipin adı

Bir yan bilgi girişi aşağıdaki güvenlik bilgilerini de belirtebilir:

- Bir güvenlik tipi.

  Yaygın olarak uygulanan güvenlik tipleri şunlardır: CM_SECURITY_NONE, CM_SECURITY_PROGRAM ve CM_SECURITY_SAME, ancak diğerleri CPI-C belirtiminde tanımlanır.
- Bir kullanıcı kimliği.
- Bir parola.

Çağıran bir MCA, çağrıdaki parametrelerden biri olarak CONNAME değerini kullanarak CPI-C çağrısı CMINIT yayınlayarak yanıt veren MCA ile bir etkileşim ayırmaya hazırlanır. CMINIT çağrısı, yerel LU yararına, MCA ' nın etkileşim için kullanmayı planladığı yan bilgi girişini tanımlar. Yerel LU, etkileşimin (2) özelliklerini kullanıma hazırlamak için bu girişteki değerleri kullanır.

Daha sonra çağıran MCA, kanal tanımında (3) USERID ve PASSWORD parametrelerinin değerlerini denetler. USERID ayarlanırsa, çağıran MCA aşağıdaki CPI-C çağrılarını verir (4):

- CMSCST, etkileşimin güvenlik tipini CM_SECURITY_PROGRAM olarak ayarlamak için.
- CMSCSU, etkileşimin kullanıcı kimliğini USERID değerine ayarlamak için kullanılır.
- Etkileşimin parolasını PASSWORD değerine ayarlamak için CMSCSP. PASSWORD belirlenmedikçe, CMSCSP çağrılmaz.

Bu çağrılar tarafından ayarlanan güvenlik tipi, kullanıcı kimliği ve parola, yan bilgi girişinden daha önce elde edilen değerleri geçersiz kılar.

Daha sonra arayan MCA, sohbeti ayırmak için CPI-C çağrısı CMALLC ' yi yayınlar (5). Bu çağrıya yanıt olarak, yerel LU ortak LU ' ya (6) bir ekleme isteği (İşlev Yönetimi Üstbilgisi 5 ya da FMH-5) gönderir.

Ortak LU bir kullanıcı kimliğini ve parolayı kabul ederse, ekleme isteğine USERID ve PASSWORD değerleri eklenir. Ortak LU bir kullanıcı kimliğini ve parolayı kabul etmezse, ekleme isteğine değerler dahil edilmez. Yerel LU, LU 'lar oturum oluşturmak için bağ tanımladığında, ortak LU' nun bilgi değiştokuşunun bir parçası olarak bir kullanıcı kimliğini ve parolayı kabul edip etmeyeceğini saptanır.

Ekleme isteğinin daha sonraki bir sürümünde, parola yerine koyma değeri, net bir parola yerine LU ' lar arasında akabilir. Parola yerine koyma değeri, paroladan oluşturulan bir DES İleti Kimlik Doğrulama Kodu (MAC) ya da SHA-1 ileti özetidir. Parola yerine koyma değerleri yalnızca her iki LU da bunları destekliyorsa kullanılabilir.

Ortak LU, bir kullanıcı kimliği ve parola içeren bir gelen ekleme isteği aldığında, kullanıcı kimliğini ve parolayı tanımlama ve kimlik denetimi amacıyla kullanabilir. Ortak LU, erişim denetim listelerine başvurarak, kullanıcı kimliğinin bir etkileşim ayırma ve yanıt veren MCA ' yı ekleme yetkisine sahip olup olmadığını da belirleyebilir.

Ayrıca, yanıt veren MCA, ekleme isteğinin içerdiği kullanıcı kimliği altında çalışabilir. Bu durumda, kullanıcı kimliği yanıt veren MCA için varsayılan kullanıcı kimliği olur ve MCA kuyruk yöneticisine bağlanmayı denediğinde yetki denetimleri için kullanılır. MCA kuyruk yöneticisinin kaynaklarına erişmeye çalıştığında daha sonra yetki denetimleri için de kullanılabilir.

Bir ekleme isteğindeki kullanıcı kimliği ve parolanın tanımlama, kimlik doğrulama ve erişim denetimi için nasıl kullanılabileceği uygulamaya bağlıdır. SNA altsisteminize özgü bilgiler için uygun belgelere bakın.

USERID ayarlanmazsa, çağıran MCA CMSCST, CMSCSU ve CMSCSP ' yi çağırmaz. Bu durumda, bir ekleme isteğinde akan güvenlik bilgileri yalnızca yan bilgi girişinde belirtilenlere ve ortak LU ' nun kabul edeceği bilgilere göre belirlenir.

Etkileşim düzeyi kimlik doğrulaması ve IBM MQ for z/OS
z/OSüzerinde etkileşim düzeyi kimlik doğrulamasının nasıl çalıştığına ilişkin bir genel bakış elde etmek için bu konuyu kullanın.

IBM MQ for z/OSişletim tarihinde, MCA ' lar CPI-C kullanmaz. Bunun yerine, bazı CPI-C özelliklerine sahip Advanced Program-to-Program Communication (APPC) uygulamasının bir uygulaması olan APPC/MVS TP Conversation Callable Services 'i kullanırlar. Bir çağıran MCA bir etkileşim ayırdığında, çağrıda SAME güvenlik tipi belirtilir. Bu nedenle, bir APPC/MVS LU giden etkileşimler için değil, yalnızca gelen etkileşimler için kalıcı doğrulamayı desteklediğinden, iki olasılık vardır:

- Ortak LU, APPC/MVS LU ' ya güvenirse ve önceden doğrulanmış bir kullanıcı kimliğini kabul ederse, APPC/MVS LU, aşağıdakileri içeren bir bağlantı isteği gönderir:
  - Kanal başlatıcı adres alanı kullanıcı kimliği
  - RACF kullanılırsa, kanal başlatıcısı adres alanı kullanıcı kimliğinin yürürlükteki bağlantı grubunun adı olan bir güvenlik profili adı
  - Önceden doğrulanmış bir gösterge
- Ortak LU, APPC/MVS LU ' ya güvenmiyorsa ve önceden doğrulanmış bir kullanıcı kimliğini kabul etmezse, APPC/MVS LU, güvenlik bilgisi içermeyen bir ekleme isteği gönderir.

IBM MQ for z/OS' ta, DEFINE CHANNEL komutundaki USERID ve PASSWORD parametreleri bir ileti kanalı için kullanılamaz ve yalnızca bir MQI kanalının istemci bağlantısı sonunda geçerlidir. Bu nedenle, bir APPC/MVS LU ' nun ekleme isteği hiçbir zaman bu değiştirgelerle belirtilen değerleri içermez.

Kuyruk yöneticisi kümeleri için güvenlik

Kuyruk yöneticisi kümelerinin kullanımı kolay olsa da, bunların güvenliğine özel olarak dikkat etmeniz gerekir.

Kuyruk yöneticisi kümesi , bir şekilde mantıksal olarak ilişkilendirilmiş kuyruk yöneticilerinden oluşan bir ağdır. Bir kümenin üyesi olan bir kuyruk yöneticisine *küme kuyruğu yöneticisi*denir.

Küme kuyruğu yöneticisine ait bir kuyruk, kümedeki diğer kuyruk yöneticileri tarafından tanınabilir. Böyle bir kuyruğa *küme kuyruğu*denir. Bir kümedeki herhangi bir kuyruk yöneticisi, aşağıdakilere gerek kalmadan küme kuyruklarına ileti gönderebilir:

- Her küme kuyruğu için belirtik bir uzak kuyruk tanımlaması
- Her uzak kuyruk yöneticisine/yöneticisinden açık olarak tanımlanmış kanallar
- Her giden kanal için ayrı bir iletim kuyruğu

İki ya da daha çok kuyruk yöneticisinin klon olduğu bir küme yaratabilirsiniz. Bu, küme kuyruğu olarak bildirilen yerel kuyruklar da içinde olmak üzere, aynı yerel kuyrukların eşgörünümlerine sahip oldukları ve aynı sunucu uygulamalarının eşgörünümlerini destekleyebildikleri anlamına gelir.

Küme kuyruk yöneticisine bağlı bir uygulama, eşkopyalanan kuyruk yöneticilerinin her birinde yönetim ortamı olan bir küme kuyruğuna ileti gönderdiğinde, IBM MQ hangi kuyruk yöneticisine gönderileceğine karar verir. Birçok uygulama küme kuyruğuna ileti gönderdiğinde IBM MQ , iş yükünü kuyruğun eşgörünümü olan kuyruk yöneticilerinin her birinde dengeler. Klonlanmış bir kuyruk yöneticisini barındıran sistemlerden biri başarısız olursa, IBM MQ , başarısız olan sistem yeniden başlatılıncaya kadar kalan kuyruk yöneticilerindeki iş yükünü dengelemeye devam eder.

Kuyruk yöneticisi kümelerini kullanıyorsanız, aşağıdaki güvenlik sorunlarını göz önünde bulundurmanız gerekir:

- Yalnızca seçilen kuyruk yöneticilerinin kuyruk yöneticinize ileti göndermesine izin verme
- Yalnızca uzak kuyruk yöneticisinin seçilen kullanıcılarının kuyruk yöneticinizdeki bir kuyruğa ileti göndermesine izin verilmesi
- Kuyruk yöneticinize bağlı uygulamaların yalnızca seçilen uzak kuyruklara ileti göndermesine izin verme

Kümeleri kullanmıyor olsanız da bu konular önemlidir, ancak kümeleri kullanıyorsanız bunlar daha önemli hale gelir.

Bir uygulama bir küme kuyruğuna ileti gönderebiliyorsa, ek uzak kuyruk tanımlarına, iletim kuyruklarına ya da kanallara gerek kalmadan başka bir küme kuyruğuna ileti gönderebilir. Bu nedenle, kuyruk yöneticinizdeki küme kuyruklarına erişimi kısıtlamak ve uygulamalarınızın ileti gönderebileceği küme kuyruklarını sınırlamak gerekip gerekmediğini göz önünde bulundurmanız daha önemli hale gelir.

Yalnızca kuyruk yöneticisi kümelerini kullanıyorsanız ilgili olan bazı ek güvenlik konuları vardır:

- Yalnızca seçilen kuyruk yöneticilerinin bir kümeye katılmasına izin verilmesi
- İstenmeyen kuyruk yöneticilerini kümeden ayrılmaya zorlama

Tüm bu önemli noktalar hakkında daha fazla bilgi için bkz. Kümelerin güvenliğini sağlama. z/OS IBM MQ for z/OSile ilgili önemli noktalar için bkz. “z/OS üzerindeki kuyruk yöneticisi kümelerinde güvenlik” sayfa 259.

İlgili görevler

“Kuyruk yöneticilerinin ileti almasını engelleme” sayfa 494
Bir küme kuyruk yöneticisinin, alma yetkisi olmayan iletileri çıkış programlarını kullanarak almasını önleyebilirsiniz.

IBM MQ Yayınlama/Abone Olma Güvenliği

IBM MQ Yayınlama/Abone Olma özelliğini kullanıyorsanız, güvenlikle ilgili dikkat edilmesi gereken ek noktalar vardır.

Bir yayınlama/abone olma sisteminde iki tip uygulama vardır: yayınlayıcı ve abone. *Yayınlayıcılar* , IBM MQ iletileri biçiminde bilgi sağlar. Bir yayınlayıcı bir ileti yayınladığında, ileti içindeki bilgilerin konusunu tanımlayan bir *konu*belirtir.

Aboneler , yayınlanan bilgilerin tüketicileridir. Bir abone, abone olarak ilgilendiği konuları belirtir.

Kuyruk yöneticisi , IBM MQ Yayınla/Abone Ol ile verilen bir uygulamadır. Abonelerden yayıncılardan ve abonelik isteklerinden yayınlanan iletileri alır ve yayınlanan iletileri abonelere yönlendirir. Bir aboneye yalnızca abone olduğu konularda ileti gönderilir.

Daha fazla bilgi için bkz. Güvenliğin yayınlanması/abone olunması.

Çok hedefli güvenlik

IBM MQ Multicast ile güvenlik süreçlerinin neden gerekli olabileceğini anlamak için bu bilgileri kullanın.

IBM MQ Birden çok hedefin yerleşik güvenliği yoktur. Güvenlik denetimleri MQOPEN zamanında kuyruk yöneticisinde işlenir ve MQMD alan ayarı istemci tarafından işlenir. Ağdaki bazı uygulamalar IBM MQ uygulamaları olmayabilir (örneğin, LLM uygulamaları, daha fazla bilgi için IBM MQ Düşük Gecikmeli İleti Sistemi ile çok hedefli birlikte çalışabilirlik konusuna bakın), bu nedenle uygulamaları almak bağlam alanlarının geçerliliğine emin olamadığından kendi güvenlik yordamlarınızı uygulamanız gerekebilir.

Göz önünde bulundurulması gereken üç güvenlik süreci vardır:

Erişim denetimi

IBM MQ içindeki erişim denetimi kullanıcı kimliklerine dayalıdır. Bu konuda daha fazla bilgi için bkz. “İstemciler için erişim denetimi” sayfa 101.

Ağ güvenliği

Yalıtılmış bir ağ, sahte mesajları önlemek için uygulanabilir bir güvenlik seçeneği olabilir. Çok hedefli grup adresindeki bir uygulamanın, aynı çok hedefli grup adresindeki bir uygulamadan geldiği için MQ iletilerinden ayırt edilemeyen yerel iletişim işlevlerini kullanarak kötü amaçlı iletiler yayınlaması mümkündür.

Çok hedefli grup adresindeki bir istemcinin, aynı çok hedefli grup adresindeki diğer istemciler için tasarlanmış iletileri alması da mümkündür.

Çoklu yayın ağının yalıtılması, yalnızca geçerli istemcilerin ve uygulamaların erişiminin olmasını sağlar. Bu güvenlik önlemi, kötü amaçlı iletilerin gelmesini ve gizli bilgilerin dışarı çıkmasını önleyebilir.

Çok hedefli grup ağ adresleri hakkında bilgi için bkz: Çok hedefli trafik için uygun ağı ayarlama

Dijital imzalar

Dijital imza, bir iletinin gösteriminin şifrelenmesiyle oluşturulur. Şifreleme, imza sahibinin özel anahtarını kullanır ve verimlilik için genellikle mesajın kendisinden ziyade bir ileti özeti üzerinde çalışır. MQPUT ' dan önce bir iletinin dijital olarak imzalanması iyi bir güvenlik önlemindir, ancak bu işlem büyük hacimli iletiler varsa performans üzerinde zararlı bir etkiye sahip olabilir.

Dijital imzalar, imzalanmakta olan verilere göre değişir. İki farklı ileti aynı varlık tarafından dijital olarak imzalandıysa, iki imza farklıdır, ancak her iki imza da aynı ortak anahtarla (yani, iletileri imzalayan varlığın genel anahtarıyla) doğrulanabilir.

Bu bölümde daha önce de belirtildiği gibi, çok hedefli grup adresindeki bir uygulamanın MQ iletilerinden ayırt edilemeyen yerel iletişim işlevlerini kullanarak kötü amaçlı iletiler yayınlaması mümkün olabilir. Dijital imzalar kaynak kanıtı sağlar ve gönderenin mesajı oluşturan kişi olduğuna dair güçlü bir kanıt sağlayan özel anahtarı yalnızca gönderen bilir.

Bu konuda daha fazla bilgi için bkz. “Şifreleme kavramları” sayfa 11.

Güvenlik duvarları ve İnternet üzerinden geçiş

Normalde düşman IP adreslerinden (örneğin, Hizmet Dışı Bırakma saldırısında) erişimi önlemek için bir güvenlik duvarı kullanırsınız. Ancak, güvenlik denetimcisinin güvenlik duvarı kurallarını güncellemesini beklerken IBM MQiçindeki IP adreslerini geçici olarak engellemeniz gerekebilir.

Bir ya da daha çok IP adresini engellemek için, BLOCKADDR ya da ADDRESSMAP tipinde bir kanal kimlik doğrulama kaydı oluşturun. Daha fazla bilgi için, bkz. “Belirli IP adreslerini engelleme” sayfa 398.

IBM MQ Internet Pass-Thru için güvenlik

IBM MQ Internet Pass-Thru , bir güvenlik duvarı üzerinden iletişimi basitleştirebilir, ancak bunun güvenlik etkileri vardır.

IBM MQ Internet Pass-Thru (MQIPT), İnternet üzerinden uzak siteler arasında ileti sistemi çözümlerini uygulamak için kullanılabilecek isteğe bağlı bir IBM MQ bileşenidir.

MQIPT , iki kuyruk yöneticisinin ya da bir IBM MQ istemci uygulamasının doğrudan TCP/IP bağlantısı gerektirmeden Internet üzerinden bir kuyruk yöneticisine bağlanmasını sağlar. Bu, bir güvenlik duvarı iki sistem arasında doğrudan TCP/IP bağlantısını engellerse kullanışlıdır. IBM MQ kanal iletişim kuralının bir güvenlik duvarına geçişini, HTTP içindeki akışları tünelleyerek ya da bir yetkili sunucu olarak hareket ederek daha basit ve daha yönetilebilir hale getirir. TLS (Transport Layer Security; İletim Katmanı Güvenliği) kullanılarak, Internet üzerinden gönderilen iletileri şifrelemek ve iletilerin şifresini çözmek için de kullanılabilir.

IBM MQ sisteminiz MQIPTile iletişim kurarken, MQIPT' de SSL yetkili sunucu kipini kullanmıyorsan, IBM MQ tarafından kullanılan CipherSpec öğesinin MQIPTtarafından kullanılan CipherSuite ile eşleştiğinden emin olun:

- MQIPT TLS sunucusu olarak hareket ederken ve IBM MQ TLS istemcisi olarak bağlanırken, IBM MQ tarafından kullanılan CipherSpec , ilgili MQIPT anahtar halkasında etkinleştirilen bir CipherSuite ' e karşılık gelmelidir.
- MQIPT TLS istemcisi olarak hareket ederken ve bir IBM MQ TLS sunucusuna bağlanırken, MQIPT CipherSuite alıcı IBM MQ kanalında tanımlanan CipherSpec ile eşleşmelidir.

MQIPT ' den tümleşik IBM MQ TLS desteğine geçiş yaparsanız, dijital sertifikaları **mqiptKeyman** ya da **mqiptKeycmd**kullanarak MQIPT anahtarlığından aktarın.

Daha fazla bilgi için bkz. IBM MQ Internet Pass-Thru.

z/OS IBM MQ for z/OS güvenlik uygulaması denetim listesi

Bu kısımda, IBM MQ kuyruk yöneticilerinizin her biri için güvenlik somutlaması üzerinde çalışmak ve tanımlamak için kullanabileceğiniz adım adım bir yordam verilmiştir.

RACF , IBM MQ güvenlik sınıflarına ilişkin tanımlamaları, verilen durağan sınıf tanımlayıcı çizelgesinde (CDT) sağlar. Denetim listesinde çalışırken, bu sınıflardan hangilerinin kuruluş için gerekli olduğunu saptayabilirsiniz. Bunların “RACF güvenlik sınıfları” sayfa 182içinde açıklandığı şekilde etkinleştirildiğinden emin olmanız gerekir.

Ayrıntılar için diğer bölümlere bakın (özellikle “IBM MQ kaynaklarına erişimi denetlemek için kullanılan profiller” sayfa 192).

Güvenlik denetimine gerek duyarsanız, bunu gerçekleştirmek için bu denetim listesini izleyin:

1. RACF MQADMIN (büyük harf profilleri) ya da MXADMIN (karışık büyük harf profilleri) sınıfını etkinleştirin.
   - Kuyruk paylaşım grubu düzeyinde, kuyruk yöneticisi düzeyinde ya da her ikisinin birleşiminde güvenlik istiyor musunuz?

     Bkz. “Kuyruk paylaşım grubunu ya da kuyruk yöneticisi düzeyinde güvenliği denetlemek için profiller” sayfa 187.
2. Bağlantı güvenliğine ihtiyacınız var mı?
   - **Evet**: MQCONN sınıfını etkinleştirin. MQCONN sınıfında kuyruk yöneticisi düzeyinde ya da kuyruk paylaşım grubu düzeyinde uygun bağlantı tanıtımlarını tanımlayın. Daha sonra, uygun kullanıcıların ya da grupların bu profillere erişmesine izin verin.

Not: Yalnızca MQCONN API isteği ya da CICS ya da IMS adres alanı kullanıcı kimliklerinin ilgili bağlantı tanıtımına erişimi olmalıdır.

- **Hayır**: Bir hlq.NO.CONNECT.CHECKS profili.

3. Komutlarda güvenlik denetimine ihtiyacınız var mı?
   - **Evet**: MQCMDS sınıfını etkinleştirin. MQCMDS sınıfında kuyruk yöneticisi düzeyinde ya da kuyruk paylaşım grubu düzeyinde uygun komut profillerini tanımlayın. Daha sonra, uygun kullanıcıların ya da grupların bu profillere erişmesine izin verin.

     Bir kuyruk paylaşım grubu kullanıyorsanız, kuyruk yöneticisinin kendisi ve kanal başlatıcısı tarafından kullanılan kullanıcı kimliklerini eklemeniz gerekebilir. Bkz. "IBM MQ for z/OS kaynak güvenliğini ayarlama" sayfa 251.
   - **Hayır**: Bir hlq.NO.CMD.CHECKS profili.
4. Komutlarda kullanılan kaynaklarda güvenliğe gerek var mı?
   - **Evet**: MQADMIN ya da MXADMIN sınıfının etkin olduğundan emin olun. MQADMIN ya da MXADMIN sınıfında kuyruk yöneticisi düzeyinde ya da kuyruk paylaşım grubu düzeyinde komutlardaki kaynakları korumak için uygun profilleri tanımlayın. Daha sonra, uygun kullanıcıların ya da grupların bu profillere erişmesine izin verin. CSQ6SYSP ' deki CMDUSER parametresini, komut güvenliği denetimleri için kullanılacak varsayılan kullanıcı kimliğine ayarlayın.

     Bir kuyruk paylaşım grubu kullanıyorsanız, kuyruk yöneticisinin kendisi ve kanal başlatıcısı tarafından kullanılan kullanıcı kimliklerini eklemeniz gerekebilir. Bkz. "IBM MQ for z/OS kaynak güvenliğini ayarlama" sayfa 251.
   - **Hayır**: Bir hlq.NO.CMD.RESC.CHECKS profili.
5. Kuyruk güvenliğine ihtiyacınız var mı?
   - **Evet**: MQQUEUE ya da MXQUEUE sınıfını etkinleştirin. MQQUEUE ya da MXQUEUEclass içindeki gerekli kuyruk yöneticisi ya da kuyruk paylaşım grubu için uygun kuyruk profillerini tanımlayın. Daha sonra, uygun kullanıcıların ya da grupların bu profillere erişmesine izin verin.
   - **Hayır**: Bir hlq.NO.QUEUE.CHECKS profili.
6. Süreç güvenliğine ihtiyacınız var mı?
   - **Evet**: MQPROC ya da MXPROC sınıfını etkinleştirin. Kuyruk yöneticisi ya da kuyruk paylaşım grubu düzeyinde uygun süreç profillerini tanımlayın ve uygun kullanıcıların ya da grupların bu profillere erişmesine izin verin.
   - **Hayır**: Bir hlq.NO.PROCESS.CHECKS profili.
7. Namelist güvenliğine ihtiyacın var mı?
   - **Evet**: MQNLIST ya da MXNLISTclass sınıfını etkinleştirin. MQNLIST ya da MXNLIST sınıfında kuyruk yöneticisi düzeyinde ya da kuyruk paylaşım grubu düzeyinde uygun ad listesi profillerini tanımlayın. Daha sonra, uygun kullanıcıların ya da grupların bu profillere erişmesine izin verin.
   - **Hayır**: Bir hlq.NO.NLIST.CHECKS profili.
8. Konu güvenliğine ihtiyacınız var mı?
   - **Evet**: MXTOPIC sınıfını etkinleştirin. MXTOPIC sınıfında kuyruk yöneticisi düzeyinde ya da kuyruk paylaşım grubu düzeyinde uygun konu profillerini tanımlayın. Daha sonra, uygun kullanıcıların ya da grupların bu profillere erişmesine izin verin.
   - **Hayır**: Bir hlq.NO.TOPIC.CHECKS profili.
9. Herhangi bir kullanıcının bağlam kullanımıyla ilgili MQOPEN ya da MQPUT1 seçeneklerini korumasına gerek var mı?
   - **Evet**: MQADMIN ya da MXADMIN sınıfının etkin olduğundan emin olun. MQADMIN ya da MXADMIN sınıfında kuyruk, kuyruk yöneticisi ya da kuyruk paylaşım grubu düzeyinde hlq.CONTEXT.queuename profillerini tanımlayın. Daha sonra, uygun kullanıcıların ya da grupların bu profillere erişmesine izin verin.
   - **Hayır**: Bir hlq.NO.CONTEXT.CHECKS profili.

10. Alternatif kullanıcı kimliklerinin kullanımını korumanız gerekiyor mu?
    - **Evet**: MQADMIN ya da MXADMIN sınıfının etkin olduğundan emin olun. Uygun hlq.ALTERNATE.USER. Gerekli kuyruk yöneticisi ya da kuyruk paylaşım grubuna ilişkin *alternateuserid* tanıtımları ve gerekli kullanıcıların ya da grupların bu tanıtımlara erişmesine izin verir.
    - **Hayır**: hlq.NO.ALTERNATE.USER.CHECKS .
11. RESLEVEL aracılığıyla kaynak güvenliği denetimleri için kullanılacak kullanıcı kimliklerini uyarlamanız gerekiyor mu?
    - **Evet**: MQADMIN ya da MXADMIN sınıfının etkin olduğundan emin olun. MQADMIN ya da MXADMIN sınıfında kuyruk yöneticisi düzeyinde ya da kuyruk paylaşım grubu düzeyinde bir hlq.RESLEVEL profili tanımlayın. Daha sonra, gerekli kullanıcıların ya da grupların profile erişmesine izin verin.
    - **Hayır**: hlq.RESLEVELiçin geçerli olabilen MQADMIN ya da MXADMIN sınıfında herhangi bir genel profil olmadığından emin olun. Gerekli kuyruk yöneticisi ya da kuyruk paylaşım grubu için bir hlq.RESLEVEL tanıtımı tanımlayın ve hiçbir kullanıcının ya da grubun bu tanıtıma erişimi olmadığını doğrulayın.
12. IBM MQ 'den kullanılmayan kullanıcı kimliklerini' zamanaşımına ' uğramanız gerekiyor mu?
    - **Evet**: Hangi zamanaşımı değerlerini kullanmak istediğini belirleyin ve TIMEOUT ve INTERVAL değiştirgelerini değiştirmek için MQSC ALTER SECURITY komutunu verin.
    - **Hayır**: INTERVAL değerini sıfıra ayarlamak için MQSC ALTER SECURITY komutunu verin.

    Not: Altsisteminiz tarafından kullanılan CSQINP1 kullanıma hazırlama giriş veri kümesini, kuyruk yöneticisi başlatıldığında MQSC ALTER SECURITY komutunun otomatik olarak verileceği şekilde güncelleyin.
13. Dağıtılmış kuyruğa alma kullanıyor musunuz?
    - **Evet**: Kanal kimlik doğrulama kayıtlarını kullanın. Daha fazla bilgi için bkz “Kanal kimlik doğrulama kayıtları” sayfa 50.
    - Her bir kanal için uygun MCAUSER öznitelik değerini belirleyebilir ya da uygun kanal güvenlik çıkışları sağlayabilirsiniz.
14. TLS (Transport Layer Security; İletim Katmanı Güvenliği) kullanmak istiyor musunuz?
    - **Evet**: Belirli bir ayırt edici adı (DN) içeren bir TLS kişisel sertifikasını sunan herhangi bir kullanıcının belirli bir MCAUSER ' i kullanacağını belirtmek için, SSLPEERMAP tipinde bir kanal kimlik doğrulama kaydı ayarlayın. Genel arama karakterleri de içinde olmak üzere tek bir ayırt edici ad ya da örüntü belirtebilirsiniz.
    - TLS altyapınızı planlayın. z/OSSistem SSL özelliğini kurun. RACFiçinde, sertifika adı süzgeçlerinizi (CNF ' ler) ve bunları kullanıyorsanız dijital sertifikalarınızı ayarlayın. SSL anahtarlığı ayarlarınızı yapın. SSLKEYR kuyruk yöneticisi özniteliğinin boş olmadığından ve SSL anahtar halkasına işaret ettiğinden emin olun. Ayrıca SSLTASKS değerinin en az 2 olduğundan emin olun.
    - **Hayır**: SSLKEYR 'nin boş olduğundan ve SSLTASKS' nin sıfır olduğundan emin olun.

    TLS hakkında daha fazla ayrıntı için bkz. “IBM MQ içinde TLS güvenlik iletişim kuralları” sayfa 24.
15. Müşteri kullanıyor musunuz?
    - **Evet**: Kanal kimlik doğrulama kayıtlarını kullanın.
    - Ayrıca, her bir sunucu bağlantısı kanalı için uygun MCAUSER öznitelik değerini belirleyebilir ya da gerekirse uygun kanal güvenlik çıkışları sağlayabilirsiniz.
16. Anahtar ayarlarınızı denetleyin.

    IBM MQ , kuyruk yöneticisi başlatıldığında güvenlik ayarlarınızı görüntüleyen iletiler yayınlar. Anahtarlarınızın doğru ayarlanıp ayarlanmadığını belirlemek için bu iletileri kullanın.
17. İstemci uygulamalarından parola gönderebilir misiniz?
    - **Evet**: En iyi koruma için z/OS özelliğinin kurulu olduğundan ve ICSF ' nin (Integrated Cryptographic Service Facility; Tümleşik Şifreleme Hizmeti Olanağı) başlatıldığından emin olun.

- **Hayır**: ICSF ' nin başlatılmadığını bildiren hata iletisini yoksayabilirsiniz.

ICSF hakkında daha fazla bilgi için bkz. “Integrated Cryptographic Service Facility (ICSF) olanağının kullanılması” sayfa 259

Güvenliğin ayarlanması

Bu konular derlemi, farklı işletim sistemlerine özgü bilgileri ve istemcilerin kullanımını içerir.

ALW AIX, Linux, and Windows üzerinde güvenliğin ayarlanması

AIX, Linux, and Windows sistemlerine özgü güvenlikle ilgili önemli noktalar.

IBM MQ kuyruk yöneticileri, değerli olabilecek bilgileri aktardığından, yetkisiz kullanıcıların kuyruk yöneticilerinize erişemediğinden emin olmak için bir yetki sistemi kullanmanız gerekir. Aşağıdaki güvenlik denetimi tiplerini göz önünde bulundurun:

IBM MQ ürününü yönetebilenler
IBM MQ' i yönetmek için komut yayınlayabilecek kullanıcılar kümesini tanımlayabilirsiniz.

IBM MQ nesnelerini kimler kullanabilir
Aşağıdakileri gerçekleştirmek için hangi kullanıcıların (genellikle uygulamalar) MQI çağrılarını ve PCF komutlarını kullanabileceğini tanımlayabilirsiniz:

- Bir kuyruk yöneticisine bağlanabilen.
- Nesnelere (kuyruklar, süreç tanımlamaları, ad bilgileri, kanallar, istemci bağlantısı kanalları, dinleyiciler, hizmetler ve kimlik doğrulama bilgileri nesneleri) ve bu nesnelere erişebilecek erişim tipleri.
- IBM MQ iletilerine kimler erişebilir.
- Bir iletiyle ilişkili bağlam bilgilerine kimler erişebilir.

Kanal güvenliği
Uzak sistemlere ileti göndermek için kullanılan kanalların gerekli kaynaklara erişebildiğinden emin olmanız gerekir.

Program kitaplıklarına, MQI bağlantı kitaplıklarına ve komutlara erişim vermek için standart işletim olanaklarını kullanabilirsiniz. Ancak, kuyrukları ve diğer kuyruk yöneticisi verilerini içeren dizin IBM MQiçin özeldir; MQI kaynaklarına yetki vermek ya da MQI yetkilerini geri almak için standart işletim sistemi komutlarını kullanmayın.

ALW AIX, Linux, and Windows üzerinde yetkilendirmeler nasıl çalışır?

Bu bölümdeki konulardaki yetkilendirme belirtimi tabloları, yetkilendirmelerin nasıl çalıştığını ve geçerli kısıtlamaları tam olarak tanımlar.

Çizelgeler aşağıdaki durumlar için geçerlidir:

- MQI çağrıları veren uygulamalar
- Kaçış PCF ' leri olarak MQSC komutlarını veren denetim programları
- PCF komutlarını veren denetim programları

Bu bölümde, bilgiler aşağıdakileri belirten bir tablo kümesi olarak sunulur:

Gerçekleştirilecek işlem
MQI seçeneği, MQSC komutu ya da PCF komutu.

Erişim denetimi nesnesi
Kuyruk, işlem, kuyruk yöneticisi, ad listesi, kimlik doğrulama bilgileri, kanal, istemci bağlantı kanalı, dinleyici ya da hizmet.

Yetki gerekli
MQZAO_ sabiti olarak ifade edilir.

Çizelgelerde, öneki MQZAO_ olan değişmezler, belirli bir varlığa ilişkin `setmqaut` komutuna ilişkin yetki listesindeki anahtar sözcüklere karşılık gelir. Örneğin, MQZAO_BROWSE `+browse`anahtar sözcüğüne karşılık gelir, MQZAO_SET_ALL_CONTEXT `+setall`anahtar sözcüğüne karşılık gelir, vb. Bu sabitler, ürünle birlikte verilen cmqzc.hüstbilgi kütüğünde tanımlanır.

ALW *MQI çağrılarına ilişkin yetkiler*

MQCONN, **MQOPEN**, **MQPUT1**ve **MQCLOSE** yetki denetimi gerektirebilir. Bu konudaki tablolar, her çağrı için gerekli yetkileri özetler.

Bir uygulamaya belirli MQI çağrıları ve seçenekleri verme izni verilir; bunun için, uygulamanın çalıştırıldığı kullanıcı kimliğine (ya da yetkilendirdiği kullanıcıya) ilgili yetki verilmiş olmalıdır.

Dört MQI çağrısı için yetki denetimi gerekebilir: **MQCONN**, **MQOPEN**, **MQPUT1**ve **MQCLOSE**.

MQOPEN ve **MQPUT1**için, bir ad çözüldükten sonra, ad ya da adlar üzerinde değil, açılmakta olan nesnenin adı üzerinde yetki denetimi yapılır. Örneğin, bir uygulamaya, diğer adın çözüldüğü temel kuyruğu açma yetkisi olmadan diğer ad kuyruğunu açma yetkisi verilebilir. Bu kural, kuyruk yöneticisi diğer adı tanımlaması doğrudan açılmadıkça, kuyruk yöneticisi diğer adı olmayan bir adı çözme işlemi sırasında karşılaşılan ilk tanımlama üzerinde denetimin yapılmasıdır; yani, adı nesne tanımlayıcısının *ObjectName* alanında görüntülenir. Açılmakta olan nesne için her zaman yetki gerekir. Bazı durumlarda, kuyruk yöneticisi nesnesi için yetki yoluyla elde edilen kuyruktan bağımsız ek yetki gereklidir.

Çizelge 10 sayfa 130, Çizelge 11 sayfa 130, Çizelge 12 sayfa 131ve Çizelge 13 sayfa 132 her çağrı için gereken yetkileri özetler. *Uygulanamaz* çizelgelerinde, yetki denetiminin bu işlemle ilgili olmadığı anlamına gelir; *Denetim yok* , yetki denetiminin gerçekleştirilmediği anlamına gelir.

Not: Bu çizelgelerde ad listesi, kanal, istemci bağlantısı kanalları, dinleyiciler, hizmetler ya da kimlik doğrulama bilgileri nesnelerinden herhangi birini bulamazsınız. Bunun nedeni, diğer nesnelerle aynı yetkilerin geçerli olduğu MQOO_INQUIRE dışında, bu nesnelere ilişkin yetkilerin hiçbiri geçerli değildir.

Özel yetki MQZAO_ALL_MQI, denetim yetkileri olarak sınıflanan MQZAO_DELETE ve MQZAO_DISPLAY dışında, nesne tipiyle ilgili çizelgelerdeki tüm yetkileri içerir.

İleti bağlamı seçeneklerinden herhangi birini değiştirmek için, çağrıyı yayınlamak için gereken yetkilere sahip olmanız gerekir. Örneğin, MQOO_SET_IDENTITY_CONTEXT ya da MQPMO_SET_IDENTITY_CONTEXT kullanabilmek için `+setid` iznine sahip olmanız gerekir.

Çizelge 10. MQCONN çağrıları için güvenlik yetkilendirmesi gerekiyor

Bu öğe için yetki gerekiyor:	**Kuyruk nesnesi ("1" sayfa 132)**	**Süreç nesnesi**	**Kuyruk yöneticisi nesnesi**
MQCONN	Geçerli değildir	Geçerli değildir	MQZAO_CONNECT

Çizelge 11. MQOPEN çağrıları için güvenlik yetkilendirmesi gerekiyor

Bu öğe için yetki gerekiyor:	**Kuyruk nesnesi ("1" sayfa 132)**	**Süreç nesnesi**	**Kuyruk yöneticisi nesnesi**
MQOO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE	MQZAO_INQUIRE
MQOO_GÖZ AT	MQZAO_GöZ AT	Geçerli değildir	Çek yok
MQOO_INPUT_ *	MQZAO_INPUT	Geçerli değildir	Çek yok
MQOO_SAVE_ ALL_CONTEXT **("2" sayfa 132)**	MQZAO_INPUT	Geçerli değildir	Geçerli değildir
MQOO_OUTPUT (Normal kuyruk) **("3" sayfa 132)**	MQZAO_OUTPUT	Geçerli değildir	Geçerli değildir

Çizelge 11. MQOPEN çağrıları için güvenlik yetkilendirmesi gerekiyor (devamı var)			
Bu öğe için yetki gerekiyor:	**Kuyruk nesnesi ("1" sayfa 132)**	**Süreç nesnesi**	**Kuyruk yöneticisi nesnesi**
MQOO_PASS_ IDENTITY_CONTEXT **("4" sayfa 132)**	MQZAO_PASS_ KIMLIK_BAĞLAMI	Geçerli değildir	Çek yok
MQOO_PASS_ALL_ BAĞLAMı **("4" sayfa 132, "5" sayfa 132)**	MQZAO_PASS _ALL_CONTEXT	Geçerli değildir	Çek yok
MQOO_SET_ IDENTITY_CONTEXT **("4" sayfa 132, "5" sayfa 132)**	MQZAO_SET_ KIMLIK_XX_ENCODE_CA SE_CAPS_LOCK_OFF BAĞLAMI	Geçerli değildir	MQZAO_SET_ IDENTITY_CONTEXT **("6" sayfa 132)**
MQOO_SET_ ALL_CONTEXT **("4" sayfa 132, "7" sayfa 132)**	MQZAO_SET_ ALL_CONTEXT	Geçerli değildir	MQZAO_SET_ ALL_CONTEXT **("6" sayfa 132)**
MQOO_OUTPUT (İletim kuyruğu) **("8" sayfa 132)**	MQZAO_SET_ ALL_CONTEXT	Geçerli değildir	MQZAO_SET_ ALL_CONTEXT **("6" sayfa 132)**
MQOO_SET	MQZAO_KüMESI	Geçerli değildir	Çek yok
MQOO_ALTERNATE_ KULLANıCı_XX_ENCODE_ CASE_ONE yetkisi	**("9" sayfa 132)**	**("9" sayfa 132)**	MQZAO_ALTERNATE_ USER_AUTHORITY **("9" sayfa 132, "10" sayfa 132)**

Çizelge 12. MQPUT1 çağrıları için gereken güvenlik yetkisi			
Bu öğe için yetki gerekiyor:	**Kuyruk nesnesi ("1" sayfa 132)**	**Süreç nesnesi**	**Kuyruk yöneticisi nesnesi**
MQPMO_PASS_ KIMLIK_BAĞLAMI	MQZAO_PASS_ IDENTITY_CONTEXT **("11" sayfa 132)**	Geçerli değildir	Çek yok
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **("11" sayfa 132)**	Geçerli değildir	Çek yok
MQPMO_SET_ KIMLIK_XX_ENCODE_CA SE_CAPS_LOCK_OFF BAĞLAMI	MQZAO_SET_ IDENTITY_CONTEXT **("11" sayfa 132)**	Geçerli değildir	MQZAO_SET_ IDENTITY_CONTEXT **("6" sayfa 132)**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **("11" sayfa 132)**	Geçerli değildir	MQZAO_SET_ ALL_CONTEXT **("6" sayfa 132)**
(İletim kuyruğu) **("8" sayfa 132)**	MQZAO_SET_ ALL_CONTEXT	Geçerli değildir	MQZAO_SET_ ALL_CONTEXT **("6" sayfa 132)**
MQPMO_ALTERNATE_ KULLANıCı_XX_ENCODE_ CASE_ONE yetkisi	**("12" sayfa 132)**	Geçerli değildir	MQZAO_ALTERNATE_ USER_AUTHORITY **("10" sayfa 132)**

Çizelge 13. MQCLOSE çağrıları için güvenlik yetkilendirmesi gerekiyor			
Bu öğe için yetki gerekiyor:	**Kuyruk nesnesi ("1" sayfa 132)**	**Süreç nesnesi**	**Kuyruk yöneticisi nesnesi**
MQCO_DELETE	MQZAO_DELETE **("13" sayfa 132)**	Geçerli değildir	Geçerli değildir
MQCO_DELETE _PURGE	MQZAO_DELETE **("13" sayfa 132)**	Geçerli değildir	Geçerli değildir

Tablolara ilişkin notlar:

1. Bir model kuyruğu açılırken:
   - Model kuyruğu için, açmakta olduğunuz erişim tipine ilişkin model kuyruğunu açma yetkisinin yanı sıra, MQZAO_DISPLAY yetkisi de gereklidir.
   - Dinamik kuyruğu yaratmak için MQZAO_CREATE yetkisi gerekmez.
   - Model kuyruğunu açmak için kullanılan kullanıcı kimliğine, yaratılan dinamik kuyruk için kuyruğa özgü tüm yetkiler (MQZAO_ALL ile eşdeğer) otomatik olarak verilir.
2. MQOO_INPUT_ * da belirtilmelidir. Bu, yerel, model ya da diğer ad kuyruğu için geçerlidir.
3. Bu denetim, iletim kuyrukları dışında tüm çıkış durumları için gerçekleştirilir (bkz. not "8" sayfa 132).
4. MQOO_OUTPUT da belirtilmelidir.
5. MQOO_PASS_IDENTITY_CONTEXT de bu seçenek tarafından örtük olarak belirtilmiştir.
6. Bu yetki, hem kuyruk yöneticisi nesnesi hem de belirli bir kuyruk için gereklidir.
7. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT ve MQOO_SET_IDENTITY_CONTEXT de bu seçenek tarafından örtük olarak belirtilmiştir.
8. Bu denetim, MQUS_ILETIM kuyruğunun *Kullanım* kuyruk özniteliğine sahip yerel ya da model kuyruğu için gerçekleştirilir ve çıkış için doğrudan açılır. Uzak kuyruk açıldığında (uzak kuyruk yöneticisi ve uzak kuyruk adları belirtilerek ya da uzak kuyruğun yerel tanımının adı belirtilerek) bu işlem uygulanmaz.
9. En az bir MQOO_INQUIRE (herhangi bir nesne tipi için) ya da MQOO_BROWSE, MQOO_INPUT_ *, MQOO_OUTPUT ya da MQOO_SET (kuyruklar için) belirtilmelidir. Gerçekleştirilen denetim, belirli bir nesne yetkisi için sağlanan diğer kullanıcı kimliği ve MQZAO_ALTERNATE_USER_IDENTIFIER denetimi için yürürlükteki uygulama yetkisi kullanılarak, belirtilen diğer seçenekler içindir.
10. Bu yetki, herhangi bir *AlternateUserkimliği* belirtilmesine izin verir.
11. Kuyrukta MQUS_ILETIM kuyruğunun *Kullanım* kuyruk özniteliği yoksa, bir MQZAO_OUTPUT denetimi de gerçekleştirilir.
12. Gerçekleştirilen denetim, belirtilen diğer seçenekler için, belirli bir adı belirtilen kuyruk yetkisi için sağlanan diğer kullanıcı kimliği ve MQZAO_ALTERNATE_USER_IDENTIFIER denetimi için yürürlükteki uygulama yetkisi kullanılarak gerçekleştirilir.
13. Denetim yalnızca aşağıdaki deyimlerin her ikisi de doğruysa gerçekleştirilir:
    - Kalıcı dinamik kuyruk kapatılıyor ve siliniyor.
    - Kuyruk, kullanılmakta olan nesne tanıtıcısını döndüren MQOPEN çağrısıyla yaratılmadı.

    Aksi takdirde, çek olmaz.

ALW *Kaçış PCF ' lerindeki MQSC komutlarına ilişkin yetkiler*

Bu bilgiler, Escape PCF ' de bulunan her MQSC komutu için gereken yetkileri özetler.

Uygulanamaz , bu işlemin bu nesne tipiyle ilgili olmadığı anlamına gelir.

Komutu gönderen programın çalıştırıldığı kullanıcı kimliğinin de aşağıdaki yetkileri olmalıdır:

- Kuyruk yöneticisi için MQZAO_CONNECT yetkisi

- PCF komutlarını gerçekleştirmek için kuyruk yöneticisi üzerinde MQZAO_DISPLAY yetkisi
- MQSC komutunu Escape PCF komutunun metni içinde verme yetkisi

ALTER *nesnesi*

Nesne	**Yetki gerekli**
Kuyruk	MQZAO_CHANGE
Konu	MQZAO_CHANGE
Süreç	MQZAO_CHANGE
Kuyruk yöneticisi	MQZAO_CHANGE
Ad listesi	MQZAO_CHANGE
Kimlik doğrulama bilgileri	MQZAO_CHANGE
Kanal	MQZAO_CHANGE
İstemci bağlantı kanalı	MQZAO_CHANGE
Dinleyici	MQZAO_CHANGE
Hizmet	MQZAO_CHANGE
İletişim bilgileri	MQZAO_CHANGE

CLEAR *nesne*

Nesne	**Yetki gerekli**
Kuyruk	MQZAO_CLEAR
Konu	MQZAO_CLEAR
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	Geçerli değildir
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir
Hizmet	Geçerli değildir
İletişim bilgileri	Geçerli değildir

DEFINE *object* NOREPLACE (“1” sayfa 137)

Nesne	**Yetki gerekli**
Kuyruk	MQZAO_CREATE **(“2” sayfa 137)**
Konu	MQZAO_CREATE **(“2” sayfa 137)**
Süreç	MQZAO_CREATE **(“2” sayfa 137)**
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	MQZAO_CREATE **(“2” sayfa 137)**
Kimlik doğrulama bilgileri	MQZAO_CREATE **(“2” sayfa 137)**

Nesne	Yetki gerekli
Kanal	MQZAO_CREATE **(“2” sayfa 137)**
İstemci bağlantı kanalı	MQZAO_CREATE **(“2” sayfa 137)**
Dinleyici	MQZAO_CREATE **(“2” sayfa 137)**
Hizmet	MQZAO_CREATE **(“2” sayfa 137)**
İletişim bilgileri	MQZAO_CREATE **(“2” sayfa 137)**

DEFINE *object* REPLACE (“1” sayfa 137, “3” sayfa 137)

Nesne	Yetki gerekli
Kuyruk	MQZAO_CHANGE
Konu	MQZAO_CHANGE
Süreç	MQZAO_CHANGE
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	MQZAO_CHANGE
Kimlik doğrulama bilgileri	MQZAO_CHANGE
Kanal	MQZAO_CHANGE
İstemci bağlantı kanalı	MQZAO_CHANGE
Dinleyici	MQZAO_CHANGE
Hizmet	MQZAO_CHANGE
İletişim bilgileri	MQZAO_CHANGE

***nesneyi* SIL**

Nesne	Yetki gerekli
Kuyruk	MQZAO_DELETE
Konu	MQZAO_DELETE
Süreç	MQZAO_DELETE
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	MQZAO_DELETE
Kimlik doğrulama bilgileri	MQZAO_DELETE
Kanal	MQZAO_DELETE
İstemci bağlantı kanalı	MQZAO_DELETE
Dinleyici	MQZAO_DELETE
Hizmet	MQZAO_DELETE
İletişim bilgileri	MQZAO_DELETE

***NESNEYI* GöRüNTüLE**

Nesne	Yetki gerekli
Kuyruk	MQZAO_DISPLAY

Nesne	Yetki gerekli
Konu	MQZAO_DISPLAY
Süreç	MQZAO_DISPLAY
Kuyruk yöneticisi	MQZAO_DISPLAY
Ad listesi	MQZAO_DISPLAY
Kimlik doğrulama bilgileri	MQZAO_DISPLAY
Kanal	MQZAO_DISPLAY
İstemci bağlantı kanalı	MQZAO_DISPLAY
Dinleyici	MQZAO_DISPLAY
Hizmet	MQZAO_DISPLAY
İletişim bilgileri	MQZAO_DISPLAY

START *nesne*

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	MQZAO_CONTROL
Hizmet	MQZAO_CONTROL
İletişim bilgileri	Geçerli değildir

STOP *nesne*

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	MQZAO_CONTROL

Nesne	Yetki gerekli
Hizmet	MQZAO_CONTROL
İletişim bilgileri	Geçerli değildir

Kanal Komutları

Komut	Nesne	Yetki gerekli
PING KANALı	Kanal	MQZAO_CONTROL
KANALI ILK DURUMUNA GETIR	Kanal	MQZAO_CONTROL_EXTENDED
KANAL ÇöZüMLE	Kanal	MQZAO_CONTROL_EXTENDED

Abonelik Komutları

Komut	Nesne	Yetki gerekli
ALTER SUB	Konu	MQZAO_CONTROL
ALT öğEYI TAN	Konu	MQZAO_CONTROL
ALT ÖĞEYI SIL	Konu	MQZAO_CONTROL
ALT öğEYI Gö	Konu	MQZAO_DISPLAY

Güvenlik Komutları

Komut	Nesne	Yetki gerekli
AUTHREC DEğERINI AYARLA	Kuyruk yöneticisi	MQZAO_CHANGE
AUTHREC ' YI SIL	Kuyruk yöneticisi	MQZAO_CHANGE
AUTHREC 'I GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY
AUTHSERV 'I GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY
ENTAUTH 'U GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY
CHLAUTH AYARLA	Kuyruk yöneticisi	MQZAO_CHANGE
CHLAUTH 'U GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY
Güvenliği yenileme	Kuyruk yöneticisi	MQZAO_CHANGE

Durum Görüntüleri

Komut	Nesne	Yetki gerekli
CHSTATUS DURUMUNU GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY Kanal tipi CLUSSDR ise, iletim kuyruğunda `+inq` yetkisinin (ya da eşdeğer olarak MQZAO_INQUIRE) gerekli olduğunu unutmayın.
LSSTATUS DURUMUNU GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY
PUBSUB 'I GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY
SBSTATUS DURUMUNU GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY

Komut	Nesne	Yetki gerekli
SVSTATUS DURUMUNU GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY
TPSTATüYü GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY

Küme Komutları

Komut	Nesne	Yetki gerekli
CLUSQMGR ' YI GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY
KüMEYI YENILE	'mqm' grup üyeliği gerekiyor	
KüMEYI SıFı	'mqm' grup üyeliği gerekiyor	
QMGR ' YI ASKıYA AL	'mqm' grup üyeliği gerekiyor	
QMGR ' YI SüRDüR	'mqm' grup üyeliği gerekiyor	

Diğer Yönetim Komutları

Komut	Nesne	Yetki gerekli
PING QMGR	Kuyruk yöneticisi	MQZAO_DISPLAY
QMGR ' YI YENILE	Kuyruk yöneticisi	MQZAO_CHANGE
QMGR ' YI SıFıR	Kuyruk yöneticisi	MQZAO_CHANGE
KONN GöRüNTüLE	Kuyruk yöneticisi	MQZAO_DISPLAY
DURDUR	Kuyruk yöneticisi	MQZAO_CHANGE

Not:

1. DEFINE komutları için, LIKE nesnesi için ya da uygun SYSTEM.DEFAULT.xxx nesnesi.
2. MQZAO_CREATE yetkisi belirli bir nesneye ya da nesne tipine özgü değil. `setmqaut` komutunda QMGR nesne tipi belirtilerek, belirtilen bir kuyruk yöneticisine ilişkin tüm nesneler için yaratma yetkisi verilir.
3. Bu, değiştirilecek nesne zaten varsa geçerlidir. Yoksa, denetim DEFINE *object* NOREPLACE ile ilgili olur.

İlgili bilgiler

Kümeleme: REFRESH CLUSTER en iyi uygulamalarını kullanma

ALW *PCF komutlarına ilişkin yetkiler*

Bu bölümde, her PCF komutu için gereken yetkiler özetlenmektedir.

Denetim yok , yetki denetimi yapılmadığı anlamına gelir; *Uygulanamaz* , bu işlemin bu nesne tipiyle ilgili olmadığı anlamına gelir.

Komutu gönderen programın çalıştırıldığı kullanıcı kimliğinin de aşağıdaki yetkileri olmalıdır:

- Kuyruk yöneticisi için MQZAO_CONNECT yetkisi
- PCF komutlarını gerçekleştirmek için kuyruk yöneticisi üzerinde MQZAO_DISPLAY yetkisi

MQZAO_ALL_ADMIN özel yetkilendirmesi, belirli bir nesne ya da nesne tipine özgü olmayan MQZAO_CREATE dışında, nesne tipiyle ilgili olan tüm yetkileri içerir.

***Nesneyi* Değiştir**

Nesne	Yetki gerekli
Kuyruk	MQZAO_CHANGE

Nesne	Yetki gerekli
Konu	MQZAO_CHANGE
Süreç	MQZAO_CHANGE
kuyruk yöneticisi	MQZAO_CHANGE
Namelist	MQZAO_CHANGE
Kimlik doğrulama bilgileri	MQZAO_CHANGE
Kanal	MQZAO_CHANGE
İstemci bağlantı kanalı	MQZAO_CHANGE
Dinleyici	MQZAO_CHANGE
Hizmet	MQZAO_CHANGE
İletişim bilgileri	MQZAO_CHANGE

***nesneyi* temizle**

Nesne	Yetki gerekli
Kuyruk	MQZAO_CLEAR
Konu	MQZAO_CLEAR
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	Geçerli değildir
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir
Hizmet	Geçerli değildir
İletişim bilgileri	Geçerli değildir

***nesneyi kopyala* (değiştirmeden) (1)**

Nesne	Yetki gerekli
Kuyruk	MQZAO_CREATE **(2)**
Konu	MQZAO_CREATE **(2)**
Süreç	MQZAO_CREATE **(2)**
Kuyruk yöneticisi	Geçerli değildir
Namelist	MQZAO_CREATE **(2)**
Kimlik doğrulama bilgileri	MQZAO_CREATE **(2)**
Kanal	MQZAO_CREATE **(2)**
İstemci bağlantı kanalı	MQZAO_CREATE **(2)**
Dinleyici	MQZAO_CREATE **(2)**

Nesne	Yetki gerekli
Hizmet	MQZAO_CREATE **(2)**
İletişim bilgileri	MQZAO_CREATE **("2" sayfa 143)**

nesneyi kopyala **(başkasıyla değiştir) (1, 4)**

Nesne	Yetki gerekli
Kuyruk	MQZAO_CHANGE
Konu	MQZAO_CHANGE
Süreç	MQZAO_CHANGE
Kuyruk yöneticisi	Geçerli değildir
Namelist	MQZAO_CHANGE
Kimlik doğrulama bilgileri	MQZAO_CHANGE
Kanal	MQZAO_CHANGE
İstemci bağlantı kanalı	MQZAO_CHANGE
Dinleyici	MQZAO_CHANGE
Hizmet	MQZAO_CHANGE
İletişim bilgileri	MQZAO_CHANGE

Nesne **yarat (değiştirmeden) (3)**

Nesne	Yetki gerekli
Kuyruk	MQZAO_CREATE **(2)**
Konu	MQZAO_CREATE **(2)**
Süreç	MQZAO_CREATE **(2)**
Kuyruk yöneticisi	Geçerli değildir
Namelist	MQZAO_CREATE **(2)**
Kimlik doğrulama bilgileri	MQZAO_CREATE **(2)**
Kanal	MQZAO_CREATE **(2)**
İstemci bağlantı kanalı	MQZAO_CREATE **(2)**
Dinleyici	MQZAO_CREATE **(2)**
Hizmet	MQZAO_CREATE **(2)**
İletişim bilgileri	MQZAO_CREATE **(2)**

nesne **yarat (başkasıyla değiştir) (3, 4)**

Nesne	Yetki gerekli
Kuyruk	MQZAO_CHANGE
Konu	MQZAO_CHANGE
Süreç	MQZAO_CHANGE
Kuyruk yöneticisi	Geçerli değildir

Nesne	Yetki gerekli
Namelist	MQZAO_CHANGE
Kimlik doğrulama bilgileri	MQZAO_CHANGE
Kanal	MQZAO_CHANGE
İstemci bağlantı kanalı	MQZAO_CHANGE
Dinleyici	MQZAO_CHANGE
Hizmet	MQZAO_CHANGE
İletişim bilgileri	MQZAO_CHANGE

***Nesne*' ni sil**

Nesne	Yetki gerekli
Kuyruk	MQZAO_DELETE
Konu	MQZAO_DELETE
Süreç	MQZAO_DELETE
Kuyruk yöneticisi	Geçerli değildir
Namelist	MQZAO_DELETE
Kimlik doğrulama bilgileri	MQZAO_DELETE
Kanal	MQZAO_DELETE
İstemci bağlantı kanalı	MQZAO_DELETE
Dinleyici	MQZAO_DELETE
Hizmet	MQZAO_DELETE
İletişim bilgileri	MQZAO_DELETE

***nesne* nesnesini sorgularken**

Nesne	Yetki gerekli
Kuyruk	MQZAO_DISPLAY
Konu	MQZAO_DISPLAY
Süreç	MQZAO_DISPLAY
kuyruk yöneticisi	MQZAO_DISPLAY
Namelist	MQZAO_DISPLAY
Kimlik doğrulama bilgileri	MQZAO_DISPLAY
Kanal	MQZAO_DISPLAY
İstemci bağlantı kanalı	MQZAO_DISPLAY
Dinleyici	MQZAO_DISPLAY
Hizmet	MQZAO_DISPLAY
İletişim bilgileri	MQZAO_DISPLAY

***Nesne* adlarını sorma**

Nesne	**Yetki gerekli**
Kuyruk	Çek yok
Konu	Çek yok
Süreç	Çek yok
Kuyruk yöneticisi	Çek yok
Ad listesi	Çek yok
Kimlik doğrulama bilgileri	Çek yok
Kanal	Çek yok
İstemci bağlantı kanalı	Çek yok
Dinleyici	Çek yok
Hizmet	Çek yok
İletişim bilgileri	Çek yok

***Nesne* ' ni Başlat**

Nesne	**Yetki gerekli**
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	MQZAO_CONTROL
Hizmet	MQZAO_CONTROL
İletişim bilgileri	Geçerli değildir

***nesneyi* durdur**

Nesne	**Yetki gerekli**
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL

Nesne	Yetki gerekli
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	MQZAO_CONTROL
Hizmet	MQZAO_CONTROL
İletişim bilgileri	Geçerli değildir

Kanal Komutları

Komut	Nesne	Yetki gerekli
Ping Kanalı	Kanal	MQZAO_CONTROL
Kanalı İlk Durumuna Getir	Kanal	MQZAO_CONTROL_EXTENDED
Kanalı Çözümle	Kanal	MQZAO_CONTROL_EXTENDED

Abonelik Komutları

Komut	Nesne	Yetki gerekli
Aboneliği Değiştir	Konu	MQZAO_CONTROL
Abonelik Yarat	Konu	MQZAO_CONTROL
Aboneliği Sil	Konu	MQZAO_CONTROL
Aboneliği sorgulamak	Konu	MQZAO_DISPLAY

Güvenlik Komutları

Komut	Nesne	Yetki gerekli
Yetki Kaydını Ayarla	Kuyruk yöneticisi	MQZAO_CHANGE
Yetki Kaydını Sil	Kuyruk yöneticisi	MQZAO_CHANGE
Yetki Kayıtlarını Sor	Kuyruk yöneticisi	MQZAO_DISPLAY
Yetki Hizmetini Sorsana	Kuyruk yöneticisi	MQZAO_DISPLAY
Varlık Yetkisini Sor	Kuyruk yöneticisi	MQZAO_DISPLAY
Kanal Kimlik Doğrulama Kaydını Ayarla	Kuyruk yöneticisi	MQZAO_CHANGE
Kanal Kimlik Doğrulama Kayıtlarını Sor	Kuyruk yöneticisi	MQZAO_DISPLAY
Güvenliği Yenile	Kuyruk yöneticisi	MQZAO_CHANGE

Durum Görüntüleri

Komut	Nesne	Yetki gerekli
Kanal Durumunu Sor	Kuyruk yöneticisi	MQZAO_DISPLAY Kanal tipi CLUSSDR ise, iletim kuyruğunda `+inq` yetkisinin (ya da eşdeğer olarak MQZAO_INQUIRE) gerekli olduğunu unutmayın.
Kanal Dinleyici Durumunu sor	Kuyruk yöneticisi	MQZAO_DISPLAY

Komut	Nesne	Yetki gerekli
Pub/Alt Durum Sorusu	Kuyruk yöneticisi	MQZAO_DISPLAY
Abonelik Durumunu Sor	Kuyruk yöneticisi	MQZAO_DISPLAY
Hizmet Durumunu Sor	Kuyruk yöneticisi	MQZAO_DISPLAY
Konu Durumunu Sor	Kuyruk yöneticisi	MQZAO_DISPLAY

Küme Komutları

Komut	Nesne	Yetki gerekli
Küme Kuyruğu Yöneticisini Sor	Kuyruk yöneticisi	MQZAO_DISPLAY
Kümeyi Yenile	'mqm' grup üyeliği gerekiyor	'mqm' grup üyeliği gerekiyor
Kümeyi İlk Durumuna Getir	'mqm' grup üyeliği gerekiyor	'mqm' grup üyeliği gerekiyor
Kuyruk Yöneticisi Kümesini Askıya Al	'mqm' grup üyeliği gerekiyor	'mqm' grup üyeliği gerekiyor
Kuyruk Yöneticisi Kümesini Sürdür	'mqm' grup üyeliği gerekiyor	'mqm' grup üyeliği gerekiyor

Diğer Yönetim Komutları

Komut	Nesne	Yetki gerekli
Ping Kuyruğu Yöneticisi	Kuyruk yöneticisi	MQZAO_DISPLAY
Kuyruk Yöneticisini Yenile	Kuyruk yöneticisi	MQZAO_CHANGE
Kuyruk Yöneticisini İlk Durumuna Getir	Kuyruk yöneticisi	MQZAO_CHANGE
Kuyruk İstatistiklerini İlk Durumuna Getir	Kuyruk	MQZAO_DISPLAY ve MQZAO_CHANGE
Bağlantıyı Sorgulamak	Kuyruk yöneticisi	MQZAO_DISPLAY
Bağlantıyı Durdur	Kuyruk yöneticisi	MQZAO_CHANGE

Not:

1. Kopyalama komutları için, Kaynak nesne için MQZAO_DISPLAY yetkisi de gereklidir.
2. MQZAO_CREATE yetkisi belirli bir nesneye ya da nesne tipine özgü değil. `setmqaut` komutunda QMGR nesne tipi belirtilerek, belirtilen bir kuyruk yöneticisine ilişkin tüm nesneler için yaratma yetkisi verilir.
3. Yaratma komutları için, uygun SYSTEM.DEFAULT.* nesne.
4. Bu, değiştirilecek nesne zaten varsa geçerlidir. Tersi durumda, denetim Copy (Kopyalama) ya da Create (Değiştirmeden Yarat) içindir.

AIX AIX üzerinde grup oluşturma ve yönetme

AIXişletim sistemi üzerinde, NIS ya da NIS + kullanmadığınızı belirterek, gruplarla çalışmak için SMITTY komutunu kullanın.

Bu görev hakkında

AIXüzerinde, bir grup oluşturmak, bir gruba kullanıcı eklemek, gruptaki kullanıcıların bir listesini görüntülemek ve bir kullanıcıyı gruptan kaldırmak için SMITTY özelliğini kullanabilirsiniz.

Yordam

1. SMITTY ' den **Security and Users** (Güvenlik ve Kullanıcılar) seçeneğini belirleyin ve Enter tuşuna basın.
2. **Groups** (Gruplar) seçeneğini belirleyin ve Enter tuşuna basın.
3. Bir grup oluşturmak için aşağıdaki adımları tamamlayın:
   a) **Add a Group** (Grup Ekle) seçeneğini belirleyin ve Enter tuşuna basın.
   b) Virgüllerle ayrılmış olarak gruba eklemek istediğiniz kullanıcıların adlarını ve grubun adını girin.
   c) Grubu oluşturmak için Enter tuşuna basın.
4. Bir gruba kullanıcı eklemek için aşağıdaki adımları tamamlayın:
   a) **Grupların Özelliklerini Değiştir/Göster** seçeneğini belirleyin ve Enter tuşuna basın.
   b) Grubun üyelerinin bir listesini göstermek için grubun adını girin.
   c) Gruba eklemek istediğiniz kullanıcıların adlarını virgülle ayrılmış olarak ekleyin.
   d) Adları gruba eklemek için Enter tuşuna basın.
5. Bir grupta kimlerin olduğunu görüntülemek için aşağıdaki adımları tamamlayın:
   a) **Grupların Özelliklerini Değiştir/Göster** seçeneğini belirleyin ve Enter tuşuna basın.
   b) Grubun üyelerinin bir listesini göstermek için grubun adını girin.
6. Bir kullanıcıyı gruptan kaldırmak için aşağıdaki adımları tamamlayın:
   a) **Grupların Özelliklerini Değiştir/Göster** seçeneğini belirleyin ve Enter tuşuna basın.
   b) Grubun üyelerinin bir listesini göstermek için grubun adını girin.
   c) Gruptan kaldırmak istediğiniz kullanıcının adını silin.
   d) Adı gruptan kaldırmak için Enter tuşuna basın.

Linux Linux üzerinde grup oluşturma ve yönetme

NIS ya da NIS + kullanmıdığınızı gösteren Linuxişletim sistemi üzerinde, gruplarla çalışmak için `/etc/group` dosyasını kullanın.

Bu görev hakkında

Linux' da, grup bilgileri `/etc/group` dosyasında tutulur. Grup yaratmak, gruba kullanıcı eklemek, gruptaki kullanıcıların listesini görüntülemek ve gruptan kullanıcı kaldırmak için komutları kullanabilirsiniz.

Yordam

1. Yeni bir grup oluşturmak için **groupadd** komutunu kullanın.
   Aşağıdaki komutu yazın:

   ```
   groupadd -g group-ID group-name
   ```

   Burada *grup-tnt* , grubun sayısal tanıtıcısıdır ve *grup-adı* , grubun adıdır.
2. Bir tamamlayıcı gruba üye eklemek için, kullanıcının şu anda üyesi olduğu tamamlayıcı grupları ve kullanıcının üyesi olacağı tamamlayıcı grupları listelemek için **usermod** komutunu kullanın.
   Örneğin, kullanıcı zaten `groupa`grubunun üyesiyse ve `groupb`grubunun üyesi olacaksa aşağıdaki komutu kullanın:

   ```
   usermod -G groupa,groupb user-name
   ```

   Burada *kullanıcı-adı* kullanıcı adıdır.
3. Bir grubun üyesi olan kişiyi görüntülemek için **getent** komutunu kullanın.
   Aşağıdaki komutu yazın:

   ```
   getent group group-name
   ```

Burada *grup-adı* , grubun adıdır.

4. Tamamlayıcı gruptan bir üyeyi kaldırmak için kullanıcının üyesi olarak kalmasını istediğiniz tamamlayıcı grupları listelemek üzere **usermod** komutunu kullanın.
   Örneğin, kullanıcının birincil grubu `users` ise ve kullanıcı aynı zamanda `mqm`, `groupa` ve `groupb`gruplarının üyesiyse, kullanıcıyı `mqm` grubundan kaldırmak için aşağıdaki komutu kullanın:

   ```
   usermod -G groupa,groupb user-name
   ```

   Burada *kullanıcı-adı* kullanıcı adıdır.

Windows Windows üzerinde grup oluşturma ve yönetme

Windowsüzerinde, bir iş istasyonundaki ya da üye sunucu makinesindeki grupları yönetmek için Bilgisayar Yönetimi özelliğini kullanırsınız.

Bu görev hakkında

Etki alanı denetleyicileri için, kullanıcılar ve gruplar Active Directoryaracılığıyla yönetilir. Active Directory ' yi kullanma hakkında daha fazla bilgi için uygun işletim sistemi yönergelerine bakın.

Bir birincil kullanıcının grup üyeliğinde yaptığınız değişiklikler, kuyruk yöneticisi yeniden başlatılıncaya ya da **REFRESH SECURITY** (ya da PCF eşdeğeri) MQSC komutunu verinceye kadar tanınmaz.

Kullanıcı ve gruplarla çalışmak için Windows Bilgisayar Yönetimi panosunu kullanın. Oturum açmış olan kullanıcıda yapılan değişiklikler, kullanıcı yeniden oturum açıncaya kadar geçerli olmayabilir.

Windows *Windows üzerinde grup oluşturma*

Denetim panosunu kullanarak bir grup oluşturun.

Yordam

1. Denetim panosunu aç
2. **Administrative Tools**(Yönetim Araçları) öğesini çift tıklatın.
   Administrative Tools (Yönetim Araçları) panosu açılır.
3. **Computer Management**(Bilgisayar Yönetimi) öğesini çift tıklatın.
   Computer Management (Bilgisayar Yönetimi) panosu açılır.
4. **Yerel Kullanıcılar ve Gruplar**nesnesini açın.
5. **Gruplar**seçeneğini sağ tıklatın ve **Yeni Grup ...**seçeneğini belirleyin.
   Yeni Grup panosu görüntülenir.
6. Grup adı alanına uygun bir ad yazın ve **Oluştur**' u tıklatın.
7. **Kapat**'ı tıklatın.

Windows *Windows üzerinde bir gruba kullanıcı eklenmesi*

Denetim panosunu kullanarak bir kullanıcıyı gruba ekleyin.

Yordam

1. Denetim panosunu aç
2. **Administrative Tools**(Yönetim Araçları) öğesini çift tıklatın.
   Administrative Tools (Yönetim Araçları) panosu açılır.
3. **Computer Management**(Bilgisayar Yönetimi) öğesini çift tıklatın.
   Computer Management (Bilgisayar Yönetimi) panosu açılır.
4. Bilgisayar Yönetimi panosundan **Yerel Kullanıcılar ve Gruplar**nesnesini açın.
5. **Kullanıcılar** seçeneğini belirleyin.

6. Gruba eklemek istediğiniz kullanıcıyı çift tıklatın.
   Kullanıcı özellikleri panosu görüntülenir.
7. **Üye** sekmesini seçin.
8. Kullanıcıyı eklemek istediğiniz grubu seçin. İstediğiniz grup görünmüyorsa:
   a) **Ekle ...**düğmesini tıklatın.
      Select Groups (Grup Seç) panosu görüntülenir.
   b) **Locations ...**(Konumlar ...) öğesini tıklatın.
      Locations (Konumlar) panosu görüntülenir.
   c) Kullanıcıyı eklemek istediğiniz grubun yerini listeden seçin ve **Tamam**düğmesini tıklatın.
   d) Sağlanan alana grup adını yazın.

      Diğer bir seçenek olarak, **Gelişmiş ...** düğmesini tıklatın. ve seçili konumda bulunan grupları listelemek için **Şimdi Bul** seçeneğini belirleyin. Buradan kullanıcıyı eklemek istediğiniz grubu seçin ve **Tamam**' ı tıklatın.
   e) **Tamam**'ı tıklatın.
      Eklediğiniz grubu gösteren kullanıcı özellikleri panosu görüntülenir.
   f) Grubu seçin.
9. **Tamam**'ı tıklatın.
   Computer Management (Bilgisayar Yönetimi) panosu görüntülenir.

Windows *Windows üzerinde bir grupta kimlerin olduğunu görüntüleme*

Denetim panosunu kullanarak bir grubun üyelerini görüntüleyin.

Yordam

1. Denetim panosunu aç
2. **Administrative Tools**(Yönetim Araçları) öğesini çift tıklatın.
   Administrative Tools (Yönetim Araçları) panosu açılır.
3. **Computer Management**(Bilgisayar Yönetimi) öğesini çift tıklatın.
   Computer Management (Bilgisayar Yönetimi) panosu açılır.
4. Bilgisayar Yönetimi panosundan **Yerel Kullanıcılar ve Gruplar**nesnesini açın.
5. **Gruplar**seçeneğini belirleyin.
6. Bir grubu çift tıklatın. Grup özellikleri panosu görüntülenir.
   Grup özellikleri panosu görüntülenir.

Sonuçlar

Grup üyeleri görüntülenir.

Windows *Windows üzerindeki bir gruptan kullanıcı kaldırılması*

Denetim panosunu kullanarak bir kullanıcıyı gruptan kaldırın.

Yordam

1. Denetim panosunu aç
2. **Administrative Tools**(Yönetim Araçları) öğesini çift tıklatın.
   Administrative Tools (Yönetim Araçları) panosu açılır.
3. **Computer Management**(Bilgisayar Yönetimi) öğesini çift tıklatın.
   Computer Management (Bilgisayar Yönetimi) panosu açılır.
4. Bilgisayar Yönetimi panosundan **Yerel Kullanıcılar ve Gruplar**nesnesini açın.

5. **Kullanıcılar**seçeneğini belirleyin.
6. Gruba eklemek istediğiniz kullanıcıyı çift tıklatın.
   Kullanıcı özellikleri panosu görüntülenir.
7. **Üye** sekmesini seçin.
8. Kullanıcıyı kaldırmak istediğiniz grubu seçin ve **Kaldır**düğmesini tıklatın.
9. **Tamam**'ı tıklatın.
   Computer Management (Bilgisayar Yönetimi) panosu görüntülenir.

Sonuçlar
Kullanıcıyı gruptan kaldırdınız.

Windows Windows üzerinde güvenlikle ilgili dikkat edilmesi gereken özel noktalar

Bazı güvenlik işlevleri, Windows' un farklı sürümlerinde farklı davranır.

IBM MQ güvenliği, kullanıcı yetkileri ve grup üyelikleri hakkında bilgi için işletim sistemi API çağrılarına dayanır. Bazı işlevler Windows sistemlerinde aynı şekilde davranmaz. Bu konu derlemi, Windows ortamında IBM MQ çalıştırılırken bu farklılıkların IBM MQ güvenliğini nasıl etkileyebileceğine ilişkin açıklamaları içerir.

Windows *IBM MQ Windows hizmeti için yerel ve etki alanı kullanıcı hesapları*
IBM MQ çalışırken, kuyruk yöneticilerine ya da kuyruklarına yalnızca yetkili kullanıcıların erişebileceğini denetlemesi gerekir. Bu, IBM MQ ' in bu tür bir erişimi deneyen herhangi bir kullanıcıya ilişkin bilgileri sorgulamak için kullanabileceği özel bir kullanıcı hesabı gerektirir.

- “Prepare IBM MQ Wizard ile özel kullanıcı hesaplarını yapılandırma” sayfa 147
- “IBM MQ ' yi Active Directory ile kullanma” sayfa 148
- “Bir IBM MQ Windows hizmeti için gereken kullanıcı hakları” sayfa 148

Prepare IBM MQ Wizard ile özel kullanıcı hesaplarını yapılandırma

Prepare IBM MQ Wizard , Windows hizmetinin kullanılması gereken süreçler tarafından paylaşılabilmesi için özel bir kullanıcı hesabı yaratır (bkz. IBM MQ ' nun PPrepare IBM MQ Wizardile yapılandırılması).

Bir Windows hizmeti, IBM MQ kuruluşu için istemci işlemleri arasında paylaşılır. Her kuruluş için bir hizmet yaratılır. Her hizmet `MQ_`*`InstallationName`*olarak adlandırılır ve görüntü adı `IBM MQ(`*`InstallationName`*`)` olur.

Her hizmetin etkileşimli olmayan ve etkileşimli oturum açma oturumları arasında paylaşılması gerektiğinden, her birini özel bir kullanıcı hesabı altında başlatmanız gerekir. Tüm hizmetler için tek bir özel kullanıcı hesabı kullanabilir ya da farklı özel kullanıcı hesapları oluşturabilirsiniz. Daha fazla bilgi için bkz. Çizelge 14 sayfa 148. Her özel kullanıcı hesabının `Hizmet olarak oturum aç`kullanıcı hakkına sahip olması gerekir. Kullanıcı kimliğinin hizmeti çalıştırma yetkisi yoksa, hizmet başlamaz ve Windows sistem olay günlüğünde bir hata döndürür. Genellikle, Prepare IBM MQ Wizardprogramını çalıştırır ve kullanıcı kimliğini doğru şekilde ayarlarsanız. Ancak, kullanıcı kimliğini el ile yapılandırdıysanız, çözmeniz gereken bir sorun olabilir mi?

IBM MQ 'ı kurduğunuzda ve Prepare IBM MQ Wizard ' ı ilk kez çalıştırdığınızda, `Hizmet olarak oturum` açda dahil olmak üzere gerekli ayarlar ve izinlerle MUSR_MQADMIN adlı hizmet için yerel bir kullanıcı hesabı oluşturur.

Sonraki kuruluşlar için Prepare IBM MQ Wizard , MUSR_MQADMIN*x*adlı bir kullanıcı hesabı oluşturur; burada *x* , var olmayan bir kullanıcı kimliğini gösteren bir sonraki kullanılabilir sayıdır. MUSR_MQADMIN*x* parolası, hesap yaratıldığında rasgele oluşturulur ve hizmet için oturum açma ortamını yapılandırmak için kullanılır. Oluşturulan parolanın süresi dolmaz.

Bu IBM MQ hesabı, hesap parolalarının belirli bir süre sonra değiştirilmesini zorunlu kılmak için sistemde ayarlanan hesap ilkelerinden etkilenmez.

Parola, bu bir kerelik işlemin dışında bilinmez ve Windows işletim sistemi tarafından kaydın güvenli bir bölümünde depolanır.

IBM MQ ' yi Active Directory ile kullanma

Active Directory dizin hizmetini kullanan etki alanı denetleyicilerinde kullanıcı hesaplarının tanımlandığı bazı ağ yapılandırmalarında, IBM MQ ' in altında çalıştığı yerel kullanıcı hesabı, diğer etki alanı kullanıcı hesaplarının grup üyeliğini sorgulama yetkisine sahip olmayabilir. IBM MQürününü kurduğunuzda Prepare IBM MQ Wizard , sınamalar gerçekleştirerek ve ağ yapılandırmasıyla ilgili sorular sorarak bunun olup olmadığını tanımlar.

IBM MQ ' in çalıştığı yerel kullanıcı hesabının gerekli yetkisi yoksa, Prepare IBM MQ Wizard sizden belirli kullanıcı haklarına sahip bir etki alanı kullanıcı hesabının hesap ayrıntılarını ister. Windows etki alanı hesabı oluşturma ve ayarlama hakkında bilgi için bkz. IBM MQ. Etki alanı kullanıcı hesabının gerektirdiği kullanıcı hakları için bkz. Çizelge 14 sayfa 148.

Prepare IBM MQ Wizardiçine etki alanı kullanıcı hesabı için geçerli hesap ayrıntılarını girdiğinizde sihirbaz, yeni hesap altında çalışacak bir IBM MQ Windows hizmeti yapılandırır. Hesap ayrıntıları, Kayıt Defteri 'nin güvenli bölümünde tutulur ve kullanıcılar tarafından okunamaz.

Hizmet çalışırken, bir IBM MQ Windows hizmeti başlatılır ve hizmet çalıştığı sürece çalışmaya devam eder. Windows hizmeti başlatıldıktan sonra sunucuda oturum açan bir IBM MQ yöneticisi, sunucudaki kuyruk yöneticilerini yönetmek için IBM MQ Explorer ' yi kullanabilir. Bu, IBM MQ Explorer öğesini var olan Windows hizmet sürecine bağlar. Bu iki işlem çalışmadan önce farklı izin düzeyleri gerekir:

- Başlatma işlemi için başlatma izni gerekiyor.
- IBM MQ yöneticisi Erişim izni gerektirir.

Bir IBM MQ Windows hizmeti için gereken kullanıcı hakları

Aşağıdaki tabloda, IBM MQ kuruluşuna ilişkin Windows hizmetinin çalıştırıldığı yerel ve etki alanı kullanıcı hesapları için gereken kullanıcı hakları listelenmektedir.

Çizelge 14. IBM MQ Windows hizmeti için gereken kullanıcı hakları

İzin	**Açıklama**
Toplu iş olarak oturum aç	Bir IBM MQ Windows hizmetinin bu kullanıcı hesabı altında çalışmasını sağlar.
Hizmet olarak oturum aç	Kullanıcıların, yapılandırılan hesabı kullanarak oturum açmaları için IBM MQ Windows hizmetini ayarlamalarını sağlar.
Sistemi kapat	IBM MQ Windows hizmetinin, bir hizmet kurtarma işlemi başarısız olduğunda sunucuyu yeniden başlatmasını sağlar.
Kotayı büyüt	İşletim sistemi `CreateProcessAsUser` çağrısı için gereklidir.
İşletim sisteminin bir parçası olarak davran	İşletim sistemi `LogonUser` çağrısı için gereklidir.
Geçiş denetimini atla	İşletim sistemi `LogonUser` çağrısı için gereklidir.
Bir işlem düzeyi anahtarını değiştir	İşletim sistemi `LogonUser` çağrısı için gereklidir.

Not: ASP ve IIS uygulamalarını çalıştıran ortamlarda hata ayıklama programı hakları gerekebilir.

Etki alanı kullanıcı hesabınızın bu Windows kullanıcı hakları, Yerel Güvenlik İlkesi uygulamasında listelenen etkin kullanıcı hakları olarak ayarlanmış olmalıdır. Değilse, bunları yerel olarak sunucuda

Yerel Güvenlik İlkesi uygulamasını kullanarak ya da Etki Alanı Güvenlik Uygulaması etki alanı genişliğini kullanarak ayarlayın.

Windows *Windows Sunucu güvenliği izinleri*
IBM MQ kuruluşu, yerel bir kullanıcının ya da etki alanı kullanıcısının kuruluşu gerçekleştirip gerçekleştirmediğine bağlı olarak Windows Server 'da farklı davranır.

Yerel bir kullanıcı IBM MQkurarsa, Prepare IBM MQ Wizard , IBM MQ Windows hizmeti için oluşturulan yerel kullanıcının, kuruluş kullanıcısının grup üyeliği bilgilerini alabildiğini algılar. Prepare IBM MQ Wizard , kullanıcıya Windows 2000 üzerinde ya da daha sonra çalışan etki alanı denetleyicilerinde tanımlı başka kullanıcı hesapları olup olmadığını belirlemek için ağ yapılandırmasıyla ilgili sorular sorar. Bu durumda, IBM MQ Windows hizmetinin belirli ayarlar ve yetkilere sahip bir etki alanı kullanıcı hesabı altında çalışması gerekir. Prepare IBM MQ Wizard , IBM MQ ürününün Prepare IBM MQ Wizard ile yapılandırılmasıiçinde açıklandığı gibi kullanıcıdan bu kullanıcının hesap ayrıntılarını ister.

Bir *etki alanı* kullanıcısı IBM MQkurarsa, Prepare IBM MQ Wizard , IBM MQ Windows hizmeti için oluşturulan yerel kullanıcının, kuruluş kullanıcısının grup üyeliği bilgilerini alamadığını algılar. Bu durumda, Prepare IBM MQ Wizard her zaman IBM MQ Windows hizmetinin kullanacağı etki alanı kullanıcı hesabının hesap ayrıntılarını kullanıcıdan ister.

IBM MQ Windows hizmetinin bir etki alanı kullanıcı hesabı kullanması gerektiğinde, Prepare IBM MQ Wizardkullanılarak yapılandırılıncaya kadar IBM MQ düzgün çalışamaz. Prepare IBM MQ Wizard , Windows hizmeti uygun bir hesapla yapılandırılıncaya kadar kullanıcının diğer görevlerle devam etmesine izin vermez.

Daha fazla bilgi için bkz. IBM MQiçin etki alanı hesapları oluşturma ve ayarlama.

Windows *IBM MQ hizmetiyle ilişkili kullanıcı adının değiştirilmesi*
Yeni bir hesap oluşturarak ve Prepare IBM MQ Wizardkullanarak ayrıntılarını girerek IBM MQ hizmetiyle ilişkili kullanıcı adını değiştirebilirsiniz.

Bu görev hakkında

IBM MQ ürününü kurduğunuzda ve Prepare IBM MQ Wizard ürününü ilk kez çalıştırdığınızda, bu, MUSR_MQADMIN adlı hizmet için yerel bir kullanıcı hesabı oluşturur. Sonraki kuruluşlar için Prepare IBM MQ Wizard , MUSR_MQADMIN*x*adlı bir kullanıcı hesabı oluşturur; burada *x* , var olmayan bir kullanıcı kimliğini gösteren bir sonraki kullanılabilir sayıdır.

IBM MQ hizmetiyle ilişkili kullanıcı adını MUSR_MQADMIN ya da MUSR_MQADMIN*x* değiştirmeniz gerekebilir. Örneğin, kuyruk yöneticiniz 8 karakterden uzun kullanıcı adlarını kabul etmeyen Db2ile ilişkiliyse bunu yapmanız gerekebilir.

Yordam

1. Yeni bir kullanıcı hesabı yaratır (örneğin, **NEW_NAME**)
2. Yeni kullanıcı hesabının ayrıntılarını girmek için Prepare IBM MQ Wizard komutunu kullanın.

İlgili görevler
IBM MQ ' nun Prepare IBM MQ Wizard ile yapılandırılması

Windows *IBM MQ Windows hizmeti yerel kullanıcı hesabının parolasını değiştirme*
IBM MQ Windows hizmet yerel kullanıcı hesabının parolasını, Bilgisayar Yönetimi panosunu kullanarak değiştirebilirsiniz.

Bu görev hakkında

IBM MQ Windows hizmet yerel kullanıcı hesabının parolasını değiştirmek için aşağıdaki adımları gerçekleştirin:

Yordam

1. Hizmetin altında çalıştığı kullanıcıyı tanımlayın.
2. IBM MQ hizmetini Bilgisayar Yönetimi panosundan durdurun.
3. Gerekli parolayı, bir kişinin parolasını değiştirdiğiniz gibi değiştirin.
4. Bilgisayar Yönetimi panosundan IBM MQ hizmetine ilişkin özelliklere gidin.
5. **Oturum Aç** sayfasını seçin.
6. Belirtilen hesap adının, parolanın değiştirildiği kullanıcıyla eşleştiğini doğrulayın.
7. Parolayı **Parola** ve **Parolayı onayla** alanlarına yazın ve **Tamam**düğmesini tıklatın.

Windows *Etki alanı kullanıcı hesabı altında çalışan bir kuruluş için IBM MQ Windows hizmetinin parolasını değiştirme*

Etki alanı kullanıcı hesabının hesap ayrıntılarını girmek için Prepare IBM MQ Wizard ' i kullanmanın bir alternatifi olarak, kuruluşa özgü IBM MQ Service için **Oturum Açma** ayrıntılarını değiştirmek için Computer Management (Bilgisayar Yönetimi) panosunu kullanabilirsiniz.

Bu görev hakkında

Bir kuruluşa ilişkin IBM MQ Windows hizmeti bir etki alanı kullanıcı hesabı altında çalışıyorsa, hesabın parolasını aşağıdaki gibi değiştirebilirsiniz:

Yordam

1. Etki alanı denetleyicisindeki etki alanı hesabının parolasını değiştirin. Etki alanı yöneticinizden bunu sizin için yapmasını istemeniz gerekebilir.
2. IBM MQ hizmetine ilişkin **Oturum Açma** sayfasını değiştirmek için aşağıdaki adımları tamamlayın.
   a) Hizmetin altında çalıştığı kullanıcıyı tanımlayın.
   b) IBM MQ hizmetini Bilgisayar Yönetimi panosundan durdurun.
   c) Gerekli parolayı, bir kişinin parolasını değiştirdiğiniz gibi değiştirin.
   d) Bilgisayar Yönetimi panosundan IBM MQ hizmetine ilişkin özelliklere gidin.
   e) **Oturum Aç** sayfasını seçin.
   f) Belirtilen hesap adının, parolanın değiştirildiği kullanıcıyla eşleştiğini doğrulayın.
   g) Parolayı **Parola** ve **Parolayı onayla** alanlarına yazın ve **Tamam**düğmesini tıklatın.

   IBM MQ Windows hizmetinin çalıştığı kullanıcı hesabı, kullanıcı arabirimi uygulamaları tarafından verilen ya da sistem başlatma, kapatma ya da hizmet kurtarma sırasında otomatik olarak gerçekleştirilen MQSC komutlarını yürütür. Bu nedenle, bu kullanıcı hesabının IBM MQ yönetim haklarına sahip olması gerekir. Varsayılan olarak, sunucudaki yerel mqm grubuna eklenir. Bu üyelik kaldırılırsa, IBM MQ Windows hizmeti çalışmaz. Kullanıcı haklarıyla ilgili daha fazla bilgi için bkz. “Bir IBM MQ Windows hizmeti için gereken kullanıcı hakları” sayfa 148.

   IBM MQ Windows hizmetinin altında çalıştığı kullanıcı hesabıyla ilgili bir güvenlik sorunu ortaya çıkarsa, sistem olay günlüğünde hata iletileri ve açıklamalar görüntülenir.

İlgili görevler

IBM MQ ' nun Prepare IBM MQ Wizard ile yapılandırılması

Windows *Windows sunucuları etki alanı denetleyicilerine yükseltilirken dikkat edilecek noktalar*

Bir Windows sunucusunu etki alanı denetleyicisine yükseltilirken, kullanıcı ve grup izinleriyle ilgili güvenlik ayarının uygun olup olmadığını göz önünde bulundurmanız gerekir. Sunucu ve etki alanı denetleyicisi arasında bir Windows makinesinin durumunu değiştirirken, IBM MQ yerel olarak tanımlı bir mqm grubu kullandığından, bunun IBM MQ ' un çalışmasını etkileyebileceğini göz önünde bulundurmalısınız.

Etki alanı kullanıcı ve grup izinleriyle ilgili güvenlik ayarları

IBM MQ , güvenlik ilkesini uygulamak için grup üyeliği bilgilerine güvenir; bu, IBM MQ işlemlerini gerçekleştiren kullanıcı kimliğinin diğer kullanıcıların grup üyeliklerini belirleyebileceği anlamına gelir.

Bir Windows sunucusunu bir etki alanı denetleyicisine yükselttiğinizde, size kullanıcı ve grup izinleriyle ilgili güvenlik ayarı için bir seçenek sunulur. Bu seçenek, isteğe bağlı kullanıcıların etkin dizinden grup üyeliklerini alıp alamayacağını denetler. Bir etki alanı denetleyicisi, yerel hesapların etki alanı kullanıcı hesaplarının grup üyeliğini sorgulama yetkisine sahip olacak şekilde ayarlanırsa, kuruluş işlemi sırasında IBM MQ tarafından oluşturulan varsayılan kullanıcı kimliği, diğer kullanıcılar için gerektiğinde grup üyeliklerini alabilir. Ancak, bir etki alanı denetleyicisi, yerel hesapların etki alanı kullanıcı hesaplarının grup üyeliğini sorgulama yetkisi olmayacak şekilde ayarlanırsa, bu, IBM MQ ' un etki alanında tanımlanan kullanıcıların kuyruk yöneticilerine ya da kuyruklarına erişim yetkisi olup olmadığını denetlemelerini tamamlamasını önler ve erişim başarısız olur. Bu şekilde ayarlanmış bir etki alanı denetleyicisinde Windows kullanıyorsanız, gerekli izinlere sahip özel bir etki alanı kullanıcı hesabı kullanılmalıdır.

Bu durumda şunu bilmeniz gerekir:

- Windows sürümünüze ilişkin güvenlik izinlerinin davranışı.
- Etki alanı mqm grubu üyelerinin grup üyeliğini okumasına izin verme.
- IBM MQ Windows hizmetini bir etki alanı kullanıcısı altında çalışacak şekilde yapılandırma.

Daha fazla bilgi için IBM MQbaşlıklı konuya bakın.

Yerel mqm grubuna IBM MQ erişimi

Windows sunucuları etki alanı denetleyicilerine yükseltildiğinde ya da etki alanı denetleyicilerinden indirgendiğinde, IBM MQ yerel mqm grubuna erişimi kaybeder.

Bir sunucu etki alanı denetleyicisi olarak yükseltildiğinde, kapsam yerel etki alanından yerel etki alanına değişir. Makine sunucuya indirgendiğinde, tüm etki alanı yerel grupları kaldırılır. Bu, bir makinenin sunucudan etki alanı denetleyicisine ve sunucuya geri dönmesinin yerel bir mqm grubuna erişimi kaybettiği anlamına gelir. Belirti, yerel bir mqm grubunun eksikliğini gösteren bir hatadır; örneğin:

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

Bu sorunu çözmek için, standart Windows yönetim araçlarını kullanarak yerel mqm grubunu yeniden yaratın. Tüm grup üyeliği bilgileri kaybolduğundan, yeni oluşturulan yerel mqm grubundaki ayrıcalıklı IBM MQ kullanıcılarını yeniden yürürlüğe almanız gerekir. Makine bir etki alanı üyesiyse, ayrıcalıklı etki alanı IBM MQ kullanıcı kimliklerine gereken yetki düzeyini vermek için yerel mqm grubuna etki alanı mqm grubunu da eklemelisiniz.

Windows *Windows üzerinde iç içe geçmiş gruplarla ilgili kısıtlamalar*

İç içe geçmiş grupların kullanımına ilişkin kısıtlamalar vardır. Bu, kısmen etki alanı işlevsel düzeyinden ve kısmen IBM MQ kısıtlamalarından kaynaklanır.

Active Directory , Etki Alanı işlevsel düzeyine bağlı olarak bir Etki Alanı bağlamında farklı grup tiplerini destekleyebilir. Varsayılan olarak, Windows 2003 etki alanları " Windows 2000 karışık " işlevsel düzey. (Windows Server 2008 ve Windows Server 2012, Windows 2003 etki alanı modelini izler.) Etki alanı işlevsel düzeyi, bir etki alanı ortamında kullanıcı kimlikleri yapılandırılırken izin verilen desteklenen grup tiplerini ve içiçe yerleştirme düzeyini belirler. Grup Kapsamı ve içerme ölçütlerine ilişkin ayrıntılar için Active Directory belgelerine bakın.

Active Directory gereksinimlerine ek olarak, IBM MQtarafından kullanılan tanıtıcılar için de ek kısıtlamalar uygulanır. IBM MQ tarafından kullanılan ağ API ' leri, etki alanı işlev düzeyi tarafından desteklenen tüm yapılandırmaları desteklemez. Sonuç olarak IBM MQ , daha sonra yerel bir gruba yerleştirilen bir Etki Alanı Yerel grubunda bulunan Etki Alanı Tanıtıcılarının grup üyeliklerini sorgulayamaz. Ayrıca, küresel ve evrensel grupların birden çok yuvalanması desteklenmez. Ancak, hemen içiçe yerleştirilmiş genel ya da evrensel gruplar desteklenir.

Windows *Kullanıcılara IBM MQ ürününü uzaktan kullanma yetkisi verilmesi*

IBM MQ ' e uzaktan bağlandığınızda kuyruk yöneticilerini yaratmanız ve başlatmanız gerekiyorsa, `Genel nesneler yarat` kullanıcı erişiminizin olması gerekir.

Bu görev hakkında

Not: Denetimcilerin varsayılan olarak `Genel nesneler yarat` kullanıcı erişimi vardır; denetimciyseniz, kullanıcı haklarınızı değiştirmeden uzaktan bağlandığınızda kuyruk yöneticilerini yaratabilir ve başlatabilirsiniz.

Bir Windows makinesine Uçbirim Hizmetleri ya da Uzak Masaüstü Bağlantısı kullanarak bağlanıyorsanız ve bir kuyruk yöneticisi yaratma, başlatma ya da silme sorunlarıyla karşılaşıyorsanız, bunun nedeni `Genel nesne yaratma`kullanıcı erişiminizin olmaması olabilir.

`Genel nesne yarat` kullanıcı erişimi, genel ad alanında nesne yaratma yetkisi olan kullanıcıları sınırlar. Bir uygulamanın genel nesne yaratabilmesi için, uygulamanın genel ad alanında çalışıyor olması ya da uygulamanın çalıştığı kullanıcının `Genel nesneler yarat` kullanıcı erişimine sahip olması gerekir.

Uçbirim Hizmetleri ya da Uzak Masaüstü Bağlantısı kullanarak bir Windows makinesine uzaktan bağlandığınızda, uygulamalar kendi yerel ad alanında çalışır. IBM MQ Explorer ya da **crtmqm** ya da **dltmqm** komutunu kullanarak bir kuyruk yöneticisi yaratmayı ya da silmeyi ya da **strmqm** komutunu kullanarak bir kuyruk yöneticisi başlatmayı denerseniz, bu bir yetkilendirme hatasıyla sonuçlanır. Bu, araştırıcı tanıtıcısı XY132002olan bir IBM MQ FDC yaratır.

IBM MQ Explorerya da **amqmdain qmgr start** komutu kullanılarak bir kuyruk yöneticisinin başlatılması, bu komutların kuyruk yöneticisini doğrudan başlatmaması nedeniyle doğru çalışır. Komutlar, kuyruk yöneticisini genel ad alanında çalışan ayrı bir sürece başlatma isteğini gönderir.

Uçbirim hizmetlerini kullanırken IBM MQ yönetim yöntemlerinin çeşitli yöntemleri çalışmazsa, `Genel nesneler yarat` kullanıcısını doğru ayarlamayı deneyin.

Yordam

1. Yönetim Araçları panosunu açın:

   Windows Server 2008 ve Windows Server 2012
   Denetim Masası > **Sistem ve Bakım** > **Yönetim Araçları**' nı kullanarak bu panoya erişin.

   Windows 8.1
   Administrative Tools > **Computer Management** (Yönetim Araçları Bilgisayar Yönetimi) olanağını kullanarak bu panele erişin
2. **Yerel Güvenlik İlkesi**' ni çift tıklatın.
3. **Yerel İlkeler**nesnesini açın.
4. **Kullanıcı Hakları Ataması**' nı tıklatın.
5. Yeni kullanıcıyı ya da grubu `Genel nesneler yarat` ilkesine ekleyin.

Windows *Windows üzerinde SSPI kanal çıkış programı*

IBM MQ for Windows , hem ileti kanallarında hem de MQI kanallarında kullanılabilecek bir güvenlik çıkış programı sağlar. Çıkış, kaynak ve nesne kodu olarak sağlanır ve tek yönlü ve iki yönlü kimlik doğrulaması sağlar.

Güvenlik çıkışı, Windows platformlarının tümleşik güvenlik olanaklarını sağlayan Güvenlik Destek Sağlayıcı Arabirimi 'ni (SSPI) kullanır.

Güvenlik çıkışı aşağıdaki tanımlama ve kimlik doğrulama hizmetlerini sağlar:

Tek yönlü kimlik doğrulama
Bu, Windows NT LAN Manager (NTLM) kimlik doğrulama desteğini kullanır. NTLM, sunucuların istemcilerini doğrulamasını sağlar. Bu, bir istemcinin bir sunucuyu doğrulamasını ya da bir sunucunun başka bir sunucuyu doğrulamasını sağlamaz. NTLM, sunucuların gerçek olduğu kabul edilen bir

ağ ortamı için tasarlanmıştır. NTLM, IBM WebSphere MQ 7.0tarafından desteklenen tüm Windows platformlarında desteklenir.

Bu hizmet genellikle bir sunucu kuyruk yöneticisinin bir IBM MQ MQI client uygulamasının kimliğini doğrulamasını sağlamak için MQI kanalında kullanılır. Bir istemci uygulaması, çalışmakta olan süreçle ilişkilendirilmiş kullanıcı kimliğiyle tanıtılır.

Kimlik doğrulamasını gerçekleştirmek için, bir kanalın istemci ucundaki güvenlik çıkışı NTLM ' den bir kimlik doğrulama belirteci alır ve bir güvenlik iletisinde belirteci kanalın diğer ucundaki ortağına gönderir. Ortak güvenlik çıkışı, belirteci NTLM ' ye geçirir ve belirtecin gerçek olup olmadığını denetler. Ortak güvenlik çıkışı, belirtecin gerçekliğinden memnun değilse, MCA ' ya kanalı kapatmasını bildirir.

İki yönlü ya da karşılıklı kimlik doğrulaması

Bu, Kerberos kimlik doğrulama hizmetlerini kullanır. Kerberos iletişim kuralı, bir ağ ortamındaki sunucuların gerçek olduğunu varsaymaz. Sunucular, istemcilerin ve diğer sunucuların kimliğini doğrulayabilir ve istemciler sunucuların kimliğini doğrulayabilir. Kerberos , IBM WebSphere MQ 7.0tarafından desteklenen tüm Windows platformlarında desteklenir.

Bu hizmet hem ileti kanallarında hem de MQI kanallarında kullanılabilir. Bir ileti kanalında, iki kuyruk yöneticisinin karşılıklı kimlik doğrulamasını sağlar. Bir MQI kanalında, sunucu kuyruk yöneticisinin ve IBM MQ MQI client uygulamasının birbirlerinin kimliğini doğrulamasını sağlar. Bir kuyruk yöneticisi, adının öneki `ibmMQSeries`/dizgisiyle belirlenir. Bir istemci uygulaması, çalışmakta olan süreçle ilişkilendirilmiş kullanıcı kimliğiyle tanıtılır.

Karşılıklı kimlik doğrulamasını gerçekleştirmek için, başlatan güvenlik çıkışı Kerberos güvenlik sunucusundan bir kimlik doğrulama simgesi alır ve güvenlik iletisindeki simgeyi iş ortağına gönderir. Ortak güvenlik çıkışı, simgeyi Kerberos sunucusuna geçirir ve bu sunucu, simgenin gerçek olup olmadığını denetler. Kerberos güvenlik sunucusu, iş ortağının başlatıcı güvenlik çıkışına bir güvenlik iletisi gönderdiği ikinci bir simge oluşturur. Başlatıcı güvenlik çıkışı, Kerberos sunucusundan ikinci simgenin gerçek olup olmadığını denetlemesini ister. Bu değiş tokuş sırasında, güvenlik çıkışlarından herhangi biri, diğeri tarafından gönderilen belirtecin gerçekliğinden memnun değilse, MCA ' ya kanalı kapatmasını bildirir.

Güvenlik çıkışı hem kaynak hem de nesne biçiminde sağlanır. Kaynak kodu, kendi kanal çıkış programlarınızı yazmak için başlangıç noktası olarak kullanabilir ya da nesne modülünü sağlanan şekilde kullanabilirsiniz. Nesne modülünde, biri NTLM kimlik doğrulama desteğini kullanarak kimlik doğrulaması için, diğeri Kerberos kimlik doğrulama hizmetlerini kullanarak iki yönlü kimlik doğrulaması için olmak üzere iki giriş noktası vardır.

SSPI kanal çıkış programının nasıl çalıştığına ve bunun nasıl uygulanacağına ilişkin yönergeler için Windows sistemlerinde SSPI güvenlik çıkışının kullanılmasıbaşlıklı konuya bakın.

Windows *Windows üzerinde güvenlik şablonu dosyaları uygulanıyor*

Bir şablonun uygulanması, IBM MQ dosyalarına ve dizinlerine uygulanan güvenlik ayarlarını etkileyebilir. Yüksek güvenlikli şablonu kullanıyorsanız, IBM MQürününü kurmadan önce şablonu uygulayın.

Windows , Güvenlik Yapılandırması ve Çözümlemesi MMC ile bir ya da daha fazla bilgisayara tek tip güvenlik ayarları uygulamak için kullanabileceğiniz metin tabanlı güvenlik şablonu dosyalarını destekler. Özellikle Windows , belirli güvenlik düzeylerini sağlamak amacıyla bir dizi güvenlik ayarı içeren birkaç şablon sağlar. Bu şablonlar Compatible, Secure ve Highly Secure içerir.

Bu şablonlardan birinin uygulanması, IBM MQ dosyalarına ve dizinlerine uygulanan güvenlik ayarlarını etkileyebilir. Yüksek Güvenli şablonu kullanmak istiyorsanız, IBM MQürününü kurmadan önce makinenizi yapılandırın.

Yüksek güvenlikli şablonu IBM MQ ' un kurulu olduğu bir makineye uygularsanız, IBM MQ dosya ve dizinlerinde ayarladığınız tüm izinler kaldırılır. Bu izinler kaldırıldığından, hata dizinlerinden *Yönetici*, *mqm*ve varsa *Herkes* grup erişimini kaybedersiniz.

Windows *IBM MQ ' e bağlanan Windows uygulamaları için ek yetkinin yapılandırılması*

IBM MQ işlemlerinin çalıştırıldığı hesap, uygulama süreçlerine EŞITLEME erişimi verilmeden önce ek yetki gerektirebilir.

Bu görev hakkında

You might experience problems if you have Windows applications, for example ASP pages, connecting to IBM MQ that are configured to run at a security level higher than usual.

IBM MQ , belirli eylemleri koordine etmek için uygulama süreçlerine EŞITLEME erişimi gerektirir. Bir sunucu uygulaması bir kuyruk yöneticisine ilk kez bağlanmayı denediğinde IBM MQ , IBM MQ denetimcileri için UYUMLULAŞTıRMA yetkisi verme işlemini değiştirir. Ancak, istenen erişimin verilebilmesi için IBM MQ işlemlerinin çalıştırıldığı hesaba ek yetki verilmesi gerekebilir.

IBM MQ işlemlerinin çalıştığı kullanıcı kimliği için ek yetki yapılandırmak üzere aşağıdaki adımları izleyin:

Yordam

1. Yerel Güvenlik İlkesi aracını başlatın, **Güvenlik Ayarları**->**Yerel İlkeler**->**Kullanıcı Sağ Atamaları**seçeneklerini tıklatın ve **Hata Ayıkla Programları**seçeneğini tıklatın.
2. **Hata Ayıklama Programları**' nı çift tıklatın ve IBM MQ kullanıcı kimliğinizi listeye ekleyin

   Sistem bir Windows etki alanindaysa ve geçerli ilke ayarı ayarlanmamışsa, yerel ilke ayarı ayarlanmış olsa da, kullanıcı kimliği Etki Alanı Güvenlik İlkesi aracı kullanılarak etki alanı düzeyinde aynı şekilde yetkilendirilmelidir.

IBM i IBM i üzerinde güvenliğin ayarlanması

IBM i üzerinde güvenlik, IBM MQ Object Authority Manager (OAM) ve IBM i nesne düzeyi güvenliği kullanılarak uygulanır.

IBM MQ nesnelerine erişim yetkisi saptanırken dikkat edilmesi gereken güvenlik konuları.

Kuruluşunuzda kullanıcılar için yetkiler ayarlarken aşağıdaki noktaları göz önünde bulundurmanız gerekir:

1. IBM i GRTOBJAUT ve RVKOBJAUT komutlarını kullanarak IBM MQ for IBM i komutlarına yetki verin ve yetkileri iptal edin.

   QMQM kitaplığında, bazı noncommand (* cmd) nesneleri, ***USE**için ***PUBLIC** yetkisine sahip olacak şekilde ayarlanır. Bu nesnelerin yetkilerini değiştirmeyin ya da yetki sağlamak için bir yetki listesi kullanın. Yanlış bir yetki IBM MQ işlevselliğini tehlikeye atabilir.
2. IBM MQ for IBM ikuruluşu sırasında aşağıdaki özel kullanıcı profilleri oluşturulur:

   QMQM
   Birincil olarak yalnızca iç ürün işlevleri için kullanılır. Ancak, güvenilir uygulamaları MQCNO_FASTPATH_BINDINGS kullanılarak çalıştırmak için kullanılabilir. MQCONNX çağrısı kullanılarak kuyruk yöneticisiyle bağlantı kurulmasıbaşlıklı konuya bakın.

   QMQMADM
   IBM MQyöneticilerinin grup profili olarak kullanılır. Grup tanıtımı, CL komutlarına ve IBM MQ kaynaklarına erişim sağlar.

   IBM MQ komutlarını çağıtan programları göndermek için SBMJOB kullanılırken, USER belirtik olarak QMQMADM olarak ayarlanmamalıdır. Bunun yerine, USER değerini QMQM olarak ya da QMQMADM ' nin grup olarak belirtilmiş olduğu başka bir kullanıcı tanıtımına ayarlayın.
3. Uzak kuyruk yöneticilerine kanal komutları gönderiyorsanız, kullanıcı tanıtımınızın hedef sistemdeki QMQMADM grubunun üyesi olduğundan emin olun. PCF ve MQSC kanal komutlarının bir listesi için bkz. IBM MQ for IBM i CL komutları.
4. Grup yetkileri OAM tarafından hesaplandığında, bir kullanıcıyla ilişkili grup kümesi önbelleğe alınır.

Grup kümesi önbelleğe alındıktan sonra kullanıcının grup üyeliklerinde yapılan değişiklikler, kuyruk yöneticisi yeniden başlatılıncaya ya da güvenliği yenilemek için RFRMQMAUT yürütülünceye kadar tanınmaz.

5. Özellikle duyarlı komutlarla çalışma yetkisi olan kullanıcıların sayısını sınırlayın. Bu komutlar şunlardır:
   - İleti Kuyruğu Yöneticisi Yarat (CRTMQM)
   - İleti Kuyruğu Yöneticisini Sil (DLTMQM)
   - İleti Kuyruğu Yöneticisini Başlat (STRMQM)
   - İleti Kuyruğu Yöneticisini Sona Erdir (ENDMQM)
   - Komut Sunucusunu Başlat (STRMQMCSVR)
   - Komut Sunucusunu Sona Erdir (ENDMQMCSVR)
6. Kanal tanımları bir güvenlik çıkış programı belirtimi içerir. Kanal oluşturma ve değiştirme özel konular gerektirir. Güvenlik çıkışlarının ayrıntıları “Güvenlik çıkışlarına genel bakış” sayfa 109içinde verilmiştir.
7. Kanal çıkışı ve tetikleyici programları yerine konabilir. Bu tür yedeklerin güvenliği programcının sorumluluğundadır.

IBM i IBM i üzerinde nesne yetkisi yöneticisi

Nesne yetki yöneticisi (OAM), kullanıcıların kuyruklar ve süreç tanımlamaları da içinde olmak üzere IBM MQ nesnelerini kullanma yetkilerini yönetir. Ayrıca, belirli bir kullanıcı grubu için bir nesneye erişim yetkisi verebileceğiniz ya da nesneyi geri alabileceğiniz bir komut arabirimi de sağlar. Bir kaynağa erişime izin verme kararı OAM tarafından yapılır ve kuyruk yöneticisi bu kararı izler. OAM bir karar veremezse, kuyruk yöneticisi o kaynağa erişimi engeller.

OAM aracılığıyla aşağıdakileri denetleyebilirsiniz:

- MQI aracılığıyla IBM MQ nesnelerine erişim. Bir uygulama programı bir nesneye erişmeye çalıştığında, OAM, isteği yapan kullanıcı tanıtımının istenen işleme ilişkin yetkiye sahip olup olmadığını denetler.

  Bu, özellikle kuyrukların ve kuyruklardaki iletilerin yetkisiz erişimden korunabileceği anlamına gelir.
- PCF ve MQSC komutlarını kullanma izni.

Farklı kullanıcı grupları aynı nesne için farklı erişim yetkisine sahip olabilir. Örneğin, belirli bir kuyruk için, bir grup hem koyma işlemlerini, hem de alma işlemlerini gerçekleştirebilir; başka bir gruba yalnızca kuyruğa göz atma izni verilebilir (MQGET ve göz atma seçeneği). Benzer şekilde, bazı grupların alma ve kuyruğa koyma yetkisi olabilir, ancak kuyruğu değiştirme ya da silme izni yoktur.

IBM MQ for IBM i komutları ve IBM MQ for IBM i nesneleri üzerinde işlem gerçekleştirme

IBM i IBM i üzerindeki IBM MQ yetkileri

IBM MQ nesnelerine erişmek için komutu verme ve başvurulan nesneye erişme yetkiniz olması gerekir. Yöneticilerin tüm IBM MQ kaynaklarına erişimi vardır.

IBM MQ nesnelerine erişim, aşağıdakilere ilişkin yetkiler tarafından denetlenir:

1. IBM MQ komutunu girin
2. Komutun başvurduğu IBM MQ nesnelerine erişim

Tüm IBM MQ for IBM i CL komutları QMQM ' nin sahibiyle birlikte verilir ve QMQMADM (denetim tanıtımı) *PUBLIC erişimi *EXCLUDE olarak ayarlanmış *USE haklarına sahiptir.

Not: QSRDUPER programı, IBM MQ for IBM i lisanslı program kuruluş programı tarafından QSYS içindeki Command (*CMD) nesnelerini çoğaltmak için kullanılır. IBM i V5R4 ve sonraki yayın düzeylerinde, QSRDUPER programı değiştirildi; böylece varsayılan davranış, özgün komutun yinelemesi yerine bir yetkili sunucu komutu yaratmaktır. Yetkili sunucu komutu, komut yürütmeyi başka bir komuta yönlendirir ve PRX özniteliği vardır. Kopyalanmakta olan komutla aynı adı taşıyan bir yetkili sunucu komutu QSYS kitaplığında varsa, ürün kitaplığındaki komuta yetkili sunucu komutu için özel yetkiler verilmez. QSYS ' de yetkili sunucu komutunu sorma ya da çalıştırmayı dener. Ürün kitaplığında hedef komutun yetkisini denetleyin.

Bu nedenle, *CMD nesnelerine ilişkin yetki değişikliklerinin ürün kitaplığında (QMQM) yapılması ve QSYS ' deki nesnelerin değiştirilmesi gerekmez. Örneğin:

```
GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
```

Bu değişiklikleri yapmak için IBM MQ nesneleri üzerinde OAM yetkiniz varsa, ürünün CL komutlarının bazılarının yetki yapısında yapılan değişiklikler bu komutların genel kullanımına izin verir.

IBM iüzerinde IBM MQ yöneticisi olmak için *QMQMADM grubunun*bir üyesi olmanız gerekir. Bu grup, AIX, Linux, and Windows sistemlerinde mqm grubunun özellikleri gibi özelliklere sahiptir. Özellikle, IBM MQ for IBM ikurulurken QMQMADM grubu yaratılır ve QMQMADM grubunun üyeleri sistemdeki tüm IBM MQ kaynaklarına erişebilir. *ALLOBJ yetkiniz varsa, tüm IBM MQ kaynaklarına da erişiminiz vardır.

Yöneticiler IBM MQ' i yönetmek için CL komutlarını kullanabilir. Bu komutlardan biri, diğer kullanıcılara yetki vermek için kullanılan GRTMQMAUT komutudur. STRMQMMQSC başka bir komut, bir yöneticinin yerel bir kuyruk yöneticisine MQSC komutları vermesine olanak sağlar.

İlgili kavramlar

“IBM i üzerinde IBM MQ yönetimi yetkisi” sayfa 89

IBM i *IBM i üzerindeki IBM MQ nesnelerine ilişkin erişim yetkileri*

IBM MQ CL komutlarını çalıştırmak için gereken erişim yetkileri.

IBM MQ for IBM i , ürünün CL komutlarını iki gruba ayırır:

Grup 1

Bu komutları işlemek için kullanıcıların QMQMADM kullanıcı grubunda olmaları ya da *ALLOBJ yetkileri olması gerekir. Bu yetkililerden herhangi birine sahip olan kullanıcılar, ek yetki gerektirmeden tüm kategorilerdeki tüm komutları işleyebilir.

Not: Bu otoriteler herhangi bir OAM yetkisini geçersiz kılıyor.

Bu komutlar aşağıdaki gibi gruplanabilir:

- Komut Sunucusu Komutları
  - ENDMQMCSVR, IBM MQ Komut Sunucusunu Sona Erdir
  - STRMQMCSVR, IBM MQ Komut Sunucusunu Başlat
- Teslim Edilmeyen İleti Kuyruğu İşleyici Komutu
  - STRMQMDLQ, IBM MQ ' U Başlatan İleti Kuyruğu İşleyicisi
- Dinleyici Komutu
  - ENDMQMLSR, IBM MQ dinleyicisini sona erdir
  - STRMQMLSR, nesne olmayan dinleyiciyi başlat
- Ortam Kurtarma Komutları
  - RCDMQMIMG, Kayıt IBM MQ Nesne Görüntüsü
  - RCRMQMOBJ, IBM MQ Nesnesi Yeniden Oluştur
  - WRKMQMTRN, IBM MQ Q İşlemleriyle Çalışma
- Kuyruk Yöneticisi Komutları
  - CRTMQM, İleti Kuyruğu Yöneticisi Yarat
  - DLTMQM, İleti Kuyruğu Yöneticisini Sil
  - ENDMQM, İleti Kuyruğu Yöneticisini Sona Erdir
  - STRMQM, İleti Kuyruk Yöneticisini Başlat
- Güvenlik Komutları
  - GRTMQMAUT, IBM MQ Nesne Yetkisi Ver

- RVKMQMAUT, IBM MQ Nesne Yetkisini Geri Al

- İzleme Komutu
  - TRCMQM, İzleme IBM MQ İşi
- Hareket Komutları
  - RSVMQMTRN, IBM MQ İşlemini Çözümle
- İzleme Programı Komutlarını Tetikle
  - STRMQMTRM, Tetikleyici İzleyiciyi Başlat
- IBM MQSC Komutları
  - RUNMQSC, IBM MQSC Komutlarını Çalıştır
  - STRMQMMQSC, Start IBM MQSC Komutları

Grup 2

İki yetki düzeyi gerektiren komutların geri kalanı:

1. Komutu çalıştırmak için IBM i yetkisi. IBM MQ denetimcisi bunu, bir kullanıcı ya da kullanıcı grubuna ilişkin *PUBLIC (*EXCLUDE) kısıtlamasını geçersiz kılmak için **GRTOBJAUT** komutunu kullanarak belirler.

   Örneğin:

   ```
   GRTOBJAUT OBJ(QMQM/DSPMQMQ) OBJTYPE(*CMD) USER(MQUSER) AUT(*USE)
   ```

2. Adım 1 'de doğru IBM i yetkisi verildiğinde, komutla ya da komutlarla ilişkili IBM MQ nesnelerini işlemek için IBM MQ yetkisi.

   Bu yetki, **GRTMQMAUT** komutunu kullanarak IBM MQ yöneticisi tarafından ayarlanan, gerekli işlem için uygun OAM yetkisine sahip kullanıcı tarafından denetlenir.

   Örneğin:

   ```
   GRTMQMAUT *connect authority to the queue manager + *admchg authority to
           the queue
   ```

Komutlar aşağıdaki gibi gruplanabilir:

- Kanal Komutları
  - CHGMQMCHL, IBM MQ Kanalını Değiştir

    Bu, * kuyruk yöneticisine bağlanma yetkisi ve * kanal için admchg yetkisi gerektirir.
  - CPYMQMCHL, IBM MQ Kanalı Kopyala

    Bu, kuyruk yöneticisine * bağlanma ve * admcrt yetkisi, kopyalanacak varsayılan kanal tipine * admdsp yetkisi ve kanal nesne sınıfına * admcrt yetkisi gerektirir.

    Örneğin, bir Gönderen kanalının kopyalanması için SYSTEM.DEF.SENDER kanalı
  - CRTMQMCHL, IBM MQ Kanalı Oluştur

    Bu, kuyruk yöneticisi için * connect ve * admcrt yetkisi, yaratılacak varsayılan kanal tipi için * admdsp yetkisi ve kanal nesne sınıfı için * admcrt yetkisi gerektirir.

    Örneğin, Gönderen kanalı yaratılması için SYSTEM.DEF.SENDER kanalı
  - DLTMQMCHL, IBM MQ Kanalını Sil

    Bu, kuyruk yöneticisine * bağlanma yetkisi ve kanala * admdlt yetkisi gerektirir.
  - RSVMQMCHL, IBM MQ Kanalını Çözümle

    Bu, kuyruk yöneticisine * bağlanma yetkisi ve kanala * ctrlx yetkisi gerektirir.
- Komutları görüntüle

DSP komutlarını işlemek için, kullanıcıya kuyruk yöneticisi için *connect* ve *admdsp* yetkisi vermeniz gerekir. Bu yetki, listelenen belirli bir seçenekle birlikte kullanılır:

- DSPMQM, İleti Kuyruk Yöneticisini Görüntüle
- DSPMQMAUT, IBM MQ Nesne Yetkisini Görüntüle
- DSPMQMAUTI, Kimlik doğrulama bilgileri nesnesinde IBM MQ Kimlik Doğrulama Bilgileri- *admdsp* öğesini görüntüle
- DSPMQMCHL, Kanala IBM MQ Kanal *admdsp* ' ı görüntüle
- DSPMQMCSVR, IBM MQ Komut Sunucusunu Görüntüle
- DSPMQMNL, Namelist için IBM MQ Namelist- *admdsp* öğesini görüntüle
- DSPMQMOBJN, IBM MQ Nesne Adlarını Görüntüle
- DSPMQMPRC, Sürece ilişkin IBM MQ Süreç- *admdsp*
- DSPMQMQ, IBM MQ Kuyruk- *admdsp* kuyruğunu kuyruğa görüntüle
- DSPMQMTOP, Konuyla ilgili IBM MQ Konu- *admdsp* öğesini görüntüle

- Komutlarla Çalışma

  WRK komutlarını işlemek ve seçenek panosunu görüntülemek için kullanıcıya *connect* ve *admdsp* yetki vermeniz gerekir. Bu yetki, listelenen belirli bir seçenekle birlikte kullanılır:

  - WRKMQM, İleti Kuyruğu Yöneticileriyle Çalışma
  - WRKMQMAUT, IBM MQ Nesne Yetkisi ile Çalışma
  - WRKMQMAUTD, IBM MQ Nesne Yetkisi Verileriyle Çalışma
  - WRKMQMAUTI, IBM MQ Kimlik Doğrulama Bilgileriyle Çalışma
    - IBM MQ Kimlik Doğrulama Bilgileri Nesnesini Değiştir komutu için *admchg* .
    - IBM MQ Kimlik Doğrulama Bilgileri Nesnesi Oluştur ve Kopyala komutu için *admcrt* .
    - IBM MQ Kimlik Doğrulama Bilgileri Nesnesini Sil komutu için *admdlt* .
    - IBM MQ Kimlik Doğrulama Bilgileri Nesnesini Görüntüle komutu için *admdsp* .
  - WRKMQMCHL, IBM MQ Kanalı ile Çalışma

    Bu, aşağıdaki yetkileri gerektirir:

    - IBM MQ Kanal Değiştir komutu için *admchg* .
    - Clear IBM MQ Channel komutu için *admclr* .
    - IBM MQ Kanal Oluştur ve Kopyala komutu için *admcrt* .
    - Delete IBM MQ Channel komutu için *admdlt* .
    - IBM MQ Kanalını Görüntüle komutu için *admdsp* .
    - Başlat IBM MQ Kanal komutu için *ctrl* .
    - IBM MQ Kanalını Sona Erdir komutu için *ctrl* .
    - Ping IBM MQ Channel komutu için *ctrl* .
    - IBM MQ Kanal İlk Durumuna Getir komutu için *ctrlx* .
    - IBM MQ Kanal Çözümle komutu için *ctrlx* .
  - WRKMQMCHST, IBM MQ Kanal Durumuyla Çalışma

    Bu, kanal için *admdsp* yetkisi gerektirir.
  - WRKMQMCL, IBM MQ Kümeleriyle Çalışma
  - WRKMQMCLQ, IBM MQ Küme Kuyruklarıyla Çalışma
  - WRKMQMCLQM, IBM MQ Küme Kuyruk Yöneticisiyle Çalışma
  - WRKMQMLSR, IBM MQ Dinleyicisiyle Çalışma
  - WRKMQMMSG, IBM MQ İletileriyle Çalışma

Bu, kuyruk için *browse* yetkisi gerektirir

– WRKMQMNL, IBM MQ Namelists ile çalışma

  Bu, aşağıdaki yetkileri gerektirir:

  - IBM MQ Namelist komutunu değiştir için *admchg* .
  - Oluştur ve Kopyala IBM MQ Namelist komutu için *admcrt* .
  - Sil IBM MQ Namelist komutu için *admdlt* .
  - IBM MQ Namelist komutunu görüntüle için *admdsp* .

– WRKMQMPRC, IBM MQ Süreçleri ile Çalışma

  Bu, aşağıdaki yetkileri gerektirir:

  - IBM MQ Değiştir komutu için *admchg* .
  - Oluştur ve Kopyala IBM MQ işlemi komutu için *admcrt* .
  - Sil IBM MQ İşlem komutu için *admdlt* .
  - *admdsp* (Görüntü Birimi IBM MQ İşlemi komutu için).

– WRKMQMQ, IBM MQ kuyruklarıyla çalışma

  Bu, aşağıdaki yetkileri gerektirir:

  - IBM MQ Kuyruğunu Değiştir komutu için *admchg* .
  - IBM MQ Kuyruğunu Temizle komutu için *admclr* .
  - IBM MQ Kuyruğunu Yarat ve Kopyala komutu için *admcrt* .
  - IBM MQ Kuyruğunu Sil komutu için *admdlt* .
  - IBM MQ Kuyruğunu Görüntüle komutu için *admdsp* .

– WRKMQMQSTS, IBM MQ Kuyruk Durumuyla Çalışma

– WRKMQMTOP, IBM MQ Konuları ile Çalışma

  Bu, aşağıdaki yetkileri gerektirir:

  - IBM MQ Konuyu Değiştir komutu için *admchg* .
  - IBM MQ Konu Oluştur ve Kopyala komutu için *admcrt* .
  - Sil IBM MQ Konu komutu için *admdlt* .
  - IBM MQ Konu komutunu görüntüle için *admdsp* .

– WRKMQMSUB, IBM MQ Abonelikleriyle Çalışma

• Diğer Kanal komutları

  Kanal komutlarını işlemek için kullanıcıya listelenen belirli yetkileri vermeniz gerekir:

  – ENDMQMCHL, IBM MQ Kanalını Sona Erdir

    Bu, kuyruk yöneticisi için *connect* yetkisi ve kanalla ilişkili iletim kuyruğu için *allmqi* yetkisi gerektirir.

  – ENDMQMLSR, IBM MQ Dinleyicisini Sona Erdir

    Bu, kuyruk yöneticisi için *connect* yetkisi ve adı belirtilen dinleyici nesnesi için *ctrl* yetkisi gerektirir.

  – PNGMQMCHL, Ping IBM MQ Kanalı

    Bu, kuyruk yöneticisi için *connect* ve *inq* yetkisi ve kanal nesnesi için *ctrl* yetkisi gerektirir.

  – RSTMQMCHL, IBM MQ Kanalını Sıfırla

    Bu, kuyruk yöneticisi için *connect* yetkisi gerektirir.

  – STRMQMCHL, IBM MQ Kanalını Başlat

    Bu, kuyruk yöneticisi için *connect* yetkisi ve kanal nesnesi için *ctrl* yetkisi gerektirir.

– STRMQMCHLI, IBM MQ Kanal Başlatıcısını Başlat

  Bu, kuyruk yöneticisi için `*connect` ve `*inq` yetkisi ve kanalın iletim kuyruğuyla ilişkili başlatma kuyruğu için `*allmqi` yetkisi gerektirir.

– STRMQMLSR, IBM MQ Dinleyicisini Başlat

  Bu, kuyruk yöneticisine * bağlanma yetkisi ve adı belirtilen dinleyici nesnesine * ctrl yetkisi gerektirir.

- Diğer komutlar:

  Aşağıdaki komutları işlemek için kullanıcıya listelenen belirli yetkileri vermeniz gerekir:

  – CCTMQM, İleti Kuyruğu Yöneticisine Bağlan

    Bu, IBM MQ nesne yetkisi gerektirmez.

  – CHGMQM, İleti Kuyruğu Yöneticisini Değiştir

    Bu, kuyruk yöneticisi için `*connect` ve `*admchg` yetkisi gerektirir.

  – CHGMQMAUTI, IBM MQ Kimlik Doğrulama Bilgilerini Değiştir

    Bu, kuyruk yöneticisi ve `*admchg` ve `*admdsp` kimlik doğrulama bilgileri nesnesi için `*connect` yetkisi gerektirir.

  – CHGMQMNL, IBM MQ Namelist öğesini Değiştir

    Bu, kuyruk yöneticisi için `*connect` yetkisi ve ad listesi için `*admchg` yetkisi gerektirir.

  – CHGMQMPRC, Değişiklik IBM MQ Süreci

    Bu, kuyruk yöneticisi için `*connect` yetkisi ve işlem için `*admchg` yetkisi gerektirir.

  – CHGMQMQ, IBM MQ Kuyruğunu Değiştir

    Bu, kuyruk yöneticisi için `*connect` yetkisi ve kuyruk için `*admchg` yetkisi gerektirir.

  – CLRMQMQ, IBM MQ Kuyruğunu Temizle

    Bu, kuyruk yöneticisi için `*connect` yetkisi ve kuyruk için `*admclr` yetkisi gerektirir.

  – CPYMQMAUTI, IBM MQ Kimlik Doğrulama Bilgilerini Kopyala

    Bu, kuyruk yöneticisi için `*connect` yetkisi ve kimlik doğrulama bilgileri nesnesi için `*admdsp` yetkisi ve kimlik doğrulama bilgileri nesne sınıfı için `*admcrt` yetkisi gerektirir.

  – CPYMQMNL, Kopya IBM MQ Namelist

    Bu, kuyruk yöneticisi için `*connect` ve `*admcrt` yetkisi gerektirir.

  – CPYMQMPRC, IBM MQ Sürecini Kopyala

    Bu, kuyruk yöneticisi için `*connect` ve `*admcrt` yetkisi gerektirir.

  – CPYMQMQ, IBM MQ Kuyruğunu Kopyala

    Bu, kuyruk yöneticisi için `*connect` ve `*admcrt` yetkisi gerektirir.

  – CRTMQMAUTI, IBM MQ Kimlik Doğrulama Bilgileri Yarat

    Bu, kuyruk yöneticisi için `*connect` yetkisi ve kimlik doğrulama bilgileri nesnesi için `*admdsp` yetkisi ve kimlik doğrulama bilgileri nesne sınıfı için `*admcrt` yetkisi gerektirir.

  – CRTMQMNL, IBM MQ Namelist Oluştur

    Bu, kuyruk yöneticisi için `*connect` ve `*admcrt` yetkisi ve varsayılan ad listesi için `*admdsp` yetkisi gerektirir.

  – CRTMQMPRC, IBM MQ Süreci Oluştur

    Bu, kuyruk yöneticisi için `*connect` ve `*admcrt` yetkisi ve varsayılan işlem için `*admdsp` yetkisi gerektirir.

  – CRTMQMQ, IBM MQ Kuyruğu Yarat

Bu, kuyruk yöneticisi için *connect ve *admcrt yetkisi ve varsayılan kuyruk için *admdsp yetkisi gerektirir.

- CVTMQMDTA, Dönüştürme IBM MQ Veri Tipi Komutu

  Bu, IBM MQ nesne yetkisi gerektirmez.
- DLTMQMAUTI, IBM MQ Kimlik Doğrulama Bilgilerini Sil

  Bu, kuyruk yöneticisi için *connect yetkisi ve kimlik doğrulama bilgileri nesnesi için *ctrlx yetkisi gerektirir.
- DLTMQMNL, IBM MQ Namelist öğesini Sil

  Bu, kuyruk yöneticisi için *connect yetkisi ve ad listesi için *admdlt yetkisi gerektirir.
- DLTMQMPRC, IBM MQ Süreci Sil

  Bu, kuyruk yöneticisi için *connect yetkisi ve işlem için *admdlt yetkisi gerektirir.
- DLTMQMQ, IBM MQ Kuyruğunu Sil

  Bu, kuyruk yöneticisi için *connect yetkisi ve kuyruk için *admdlt yetkisi gerektirir.
- DSCMQM, İleti Kuyruğu Yöneticisi ile Bağlantıyı Kes

  Bu, IBM MQ nesne yetkisi gerektirmez.
- RFRMQMAUT, Güvenliği Yenile

  Bu, kuyruk yöneticisi için *connect yetkisi gerektirir.
- RFRMQMCL, Küme Yenile

  Bu, kuyruk yöneticisi için *connect yetkisi gerektirir.
- RSMMQMCLQM, Küme Kuyruk Yöneticisini Sürdür

  Bu, kuyruk yöneticisi için *connect yetkisi gerektirir.
- RSTMQMCL, Kümeyi Sıfırla

  Bu, kuyruk yöneticisi için *connect yetkisi gerektirir.
- SPDMQMCLQM, Küme Kuyruğu Yöneticisini Askıya Al

  Bu, kuyruk yöneticisi için *connect yetkisi gerektirir.

IBM i *IBM i üzerindeki erişim yetkileri*

Erişim yetkisi komutlarını anlamak için bu bilgileri kullanın.

GRTMQMAUT ve RVKMQMAUT komutlarında AUT anahtar sözcüğü tarafından tanımlanan yetkilendirmeler aşağıdaki gibi kategorilere ayrılabilir:

- MQI çağrılarıyla ilgili yetkiler
- Yetkilendirmeyle ilgili yönetim komutları
- Bağlam yetkileri
- MQI çağrıları, komutlar ya da her ikisi için genel yetkiler

Aşağıdaki çizelgeler, MQI çağrıları, Bağlam çağrıları, MQSC ve PCF komutları ve soysal işlemler için AUT değiştirgesini kullanarak farklı yetkileri listeler.

Çizelge 15. MQI çağrılarına ilişkin yetkiler

AUT.	**Açıklama**
*ALTUSR	MQOPEN ve MQPUT1 çağrıları için başka bir kullanıcının yetkisinin kullanılmasına izin verin.
*GöZ AT	BROWSE seçeneğiyle bir MQGET çağrısı yayınlayarak kuyruktan ileti alın.

Çizelge 15. MQI çağrılarına ilişkin yetkiler (devamı var)	
AUT.	**Açıklama**
*BAğLANTı	Bir MQCONN çağrısı yayınlayarak uygulamayı belirtilen kuyruk yöneticisine bağlayın.
*GET	Bir MQGET çağrısı yayınlayarak kuyruktan ileti alın.
*INQ	MQINQ çağrısı yayınlayarak belirli bir kuyrukta sorgu yürütebilirsiniz.
*PUB	MQPUT çağrısıyla ileti yayınlamak için bir konu açın.
*PUT	Bir MQPUT çağrısı yayınlayarak belirli bir kuyruğa ileti koyun.
*SüRDüR	MQSUB çağrısıyla aboneliği sürdürün.
*SET	MQSET çağrısı yayınlayarak, MQI 'dan bir kuyruktaki öznitelikleri ayarlayın. Bir kuyruğu birden çok seçenek için açarsanız, bunların her biri için yetkili olmanız gerekir.
*SUB	MQSUB çağrısıyla bir konu aboneliği yaratın, değiştirin ya da aboneliğe devam edin.

Çizelge 16. Bağlam çağrılarına ilişkin yetkiler	
AUT.	**Açıklama**
*GEçIş	Belirtilen kuyruktaki tüm bağlamı geçirin. Tüm bağlam alanları özgün istekten kopyalanır.
*PASSID	Belirtilen kuyrukta kimlik bağlamını geçirin. Kimlik bağlamı, isteğin bağlamıyla aynı.
*SETALL	Belirtilen kuyruktaki tüm bağlamı ayarlayın. Bu, özel sistem yardımcı programları tarafından kullanılır.
*SETID	Belirtilen kuyrukta kimlik bağlamını ayarlayın. Bu, özel sistem yardımcı programları tarafından kullanılır.

Çizelge 17. MQSC ve PCF çağrılarına ilişkin yetkiler	
AUT.	**Açıklama**
*ADMCHG	Belirtilen nesnenin özniteliklerini değiştirin.
*ADMCLR	Belirlenen nesneyi temizle (yalnızca PCF nesneyi temizle komutu).
*ADMCRT	Belirtilen tipte nesneler yaratın.
*ADMDLT	Belirtilen nesneyi siler.
*ADMDSP	Belirtilen nesnenin özniteliklerini görüntüler.

Çizelge 18. Soysal işlemlere ilişkin yetkiler	
AUT.	**Açıklama**
*ALL	Nesne için geçerli olan tüm işlemleri kullanın. `all` yetkisi, nesne tipine uygun yetkilerin `alladm`, `allmqive` `system` birleşmesiyle eşdeğerdir.
*ALLADM	Nesne için geçerli olan tüm denetim işlemlerini gerçekleştirin.
*ALLMQI	Nesne için geçerli olan tüm MQI çağrılarını kullanın.
*CTRL	Kanalların, dinleyicilerin ve hizmetlerin başlatılmasını ve kapatılmasını denetler.
*CTRLX	Sıra numarasını sıfırlayın ve belirsiz kanalları çözün.

IBM i *IBM i üzerinde erişim yetkisi komutlarının kullanılması*

Erişim yetkisi komutlarına ilişkin bilgi edinmek ve komut örneklerini kullanmak için bu bilgileri kullanın.

GRTMQMAUT komutunun kullanılması

Gerekli yetkiniz varsa, belirli bir nesneye erişmek üzere bir kullanıcı tanıtımı ya da kullanıcı grubu için yetki vermek üzere GRTMQMAUT komutunu kullanabilirsiniz. Aşağıdaki örnekler, GRTMQMAUT komutunun nasıl kullanıldığını göstermektedir:

1. ```
   GRTMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
             AUT(*BROWSE *PUT) MQMNAME('saturn.queue.manager')
   ```

   Bu örnekte:

   - RED.LOCAL.QUEUE , nesne adıdır.
   - *LCLQ (yerel kuyruk) nesne tipidir.
   - GROUPA , sistemdeki yetkilendirmelerin değiştirileceği kullanıcı profilinin adıdır. Bu profil, diğer kullanıcılar için bir grup profili olarak kullanılabilir.
   - *BROWSE ve *PUT , belirtilen kuyruğa verilen yetkilerdir.

     *BROWSE , kuyruktaki iletilere göz atma yetkisi ekler (MQGET ' i göz atma seçeneğiyle yayınlamak için).

     *PUT , kuyruğa MQPUT iletileri koymak için yetki ekler.
   - saturn.queue.manager , kuyruk yöneticisi adıdır.
2. Aşağıdaki komut, varsayılan kuyruk yöneticisi için tüm süreç tanımlamalarına JACK ve JILL geçerli tüm yetkilerini verir.

   ```
   GRTMQMAUT OBJ(*ALL) OBJTYPE(*PRC) USER(JACK JILL) AUT(*ALL)
   ```
3. Aşağıdaki komut, kullanıcıya kuyruk yöneticisine (TRENT) bir ileti koymak için GEORGE yetkisi verir ORDERS.

   ```
   GRTMQMAUT OBJ(TRENT) OBJTYPE(*MQM) USER(GEORGE) AUT(*CONNECT) MQMNAME (TRENT)
   GRTMQMAUT OBJ(ORDERS) OBJTYPE(*Q) USER(GEORGE) AUT(*PUT) MQMNAME (TRENT)
   ```

RVKMQMAUT komutunun kullanılması

Gerekli yetkiniz varsa, RVKMQMAUT komutunu kullanarak, belirli bir nesneye erişmek için önceden verilen bir kullanıcı tanıtımı ya da kullanıcı grubu yetkisini kaldırabilirsiniz. Aşağıdaki örnekler, RVKMQMAUT komutunun nasıl kullanıldığını göstermektedir:

1. ```
   RVKMQMAUT OBJ(RED.LOCAL.QUEUE) OBJTYPE(*LCLQ) USER(GROUPA) +
   AUT(*PUT) MQMNAME('saturn.queue.manager')
   ```

   Önceki örnekte verilen iletileri belirtilen kuyruğa koyma yetkisi, GROUPAiçin kaldırılır.
2. ```
   RVKMQMAUT OBJ(PAY*) OBJTYPE(*Q) USER(*PUBLIC) AUT(*GET) +
   MQMNAME(PAYROLLQM)
   ```

   PAYKuyruk yöneticisinin PAYROLLQMiyeliğindeki karakterleriyle başlayan iletileri herhangi bir kuyruktan alma yetkisi, ayrı olarak yetkilendirilmediği sürece, sistemin tüm kullanıcılarından kaldırılır.
   ```

DSPMQMAUT komutunun kullanılması

Görüntüleme MQM yetkisi (DSPMQMAUT) Komut, belirtilen nesne ve kullanıcı için, kullanıcının nesne için sahip olduğu yetkilerin listesini gösterir. Aşağıdaki örnek, komutun nasıl kullanıldığını gösterir:

```
DSPMQMAUT OBJ(ADMINNL) OBJTYPE(*NMLIST) USER(JOE) OUTPUT(*PRINT) +
MQMNAME(ADMINQM)
```

RFRMQMAUT komutunun kullanılması

MQM güvenliğini yenileme (RFRMQMAUT) komut, kuyruk yöneticisini durdurup yeniden başlatmanıza gerek kalmadan, işletim sistemi düzeyinde yapılan değişiklikleri yansıtan OAM yetki grubu bilgilerini hemen güncellemenizi sağlar. Aşağıdaki örnek, komutun nasıl kullanıldığını gösterir:

```
RFRMQMAUT MQMNAME(ADMINQM)
```

IBM i IBM i üzerinde yetkilendirme belirtimi çizelgeleri

Belirli API çağrılarını ve bu çağrıların belirli seçeneklerini (kuyruk nesneleri, işlem nesneleri ve kuyruk yöneticisi nesneleri) kullanmak için hangi yetkinin gerekli olduğunu saptamak için bu bilgileri kullanın.

Çizelge 19 sayfa 165 içinde başlayan yetkilendirme belirtimi tabloları, yetkilerin nasıl çalıştığını ve geçerli kısıtlamaları tam olarak tanımlar. Çizelgeler aşağıdaki durumlar için geçerlidir:

- MQI çağrıları veren uygulamalar
- Kaçış PCF ' leri olarak MQSC komutlarını veren denetim programları
- PCF komutlarını veren denetim programları

Bu bölümde, bilgiler aşağıdaki verileri belirten bir tablo kümesi olarak sunulur:

Gerçekleştirilecek işlem
MQI seçeneği, MQSC komutu ya da PCF komutu.

Erişim denetimi nesnesi
Kuyruk, süreç tanımlaması, kuyruk yöneticisi, ad listesi, kanal, istemci bağlantısı kanalı, dinleyici, hizmet ya da kimlik doğrulama bilgileri nesnesi.

Yetki gerekli
MQZAO_ sabiti olarak ifade edilir.

Çizelgelerde, öneki MQZAO_ olan değişmezler, belirli bir varlığa ilişkin **GRTMQMAUT** ve **RVKMQMAUT** komutlarına ilişkin yetki listesindeki anahtar sözcüklere karşılık gelir. Örneğin, MQZAO_BROWSE *BROWSE anahtar sözcüğüne karşılık gelir; Benzer şekilde, MQZAO_SET_ALL_CONTEXT anahtar sözcüğü *SETALLanahtar sözcüğüne karşılık gelir. Bu sabitler, ürünle birlikte verilen cmqzc.hüstbilgi kütüğünde tanımlanır.

MQI yetkileri

Bir uygulamaya belirli MQI çağrıları ve seçenekleri verme izni verilir; bunun için, uygulamanın çalıştırıldığı kullanıcı kimliğine (ya da yetkilendirdiği kullanıcıya) ilgili yetki verilmiş olmalıdır.

Dört MQI çağrısı yetkilendirme denetimi gerektirir: MQCONN, MQOPEN, MQPUT1ve MQCLOSE.

MQOPEN ve MQPUT1için, yetki denetimi, bir ad çözüldükten sonra sonuçlanan, adı ya da adları değil, açılmakta olan nesnenin adı üzerinde yapılır. Örneğin, bir uygulamaya, diğer adın çözüldüğü temel kuyruğu açma yetkisi olmadan diğer ad kuyruğunu açma yetkisi verilebilir. Kural, kuyruk yöneticisi diğer adı tanımı doğrudan açılmadıkça, ad çözme işlemi sırasında karşılaşılan ilk tanımlama üzerinde denetimin gerçekleştirilmesidir; yani, adı nesne tanımlayıcısının *ObjectName* alanında görüntülenir. Açılmakta olan nesne için her zaman yetki gerekir; bazı durumlarda, kuyruk yöneticisi nesnesi için bir yetki yoluyla elde edilen kuyruktan bağımsız ek yetki gerekir.

Çizelge 19 sayfa 165, Çizelge 20 sayfa 165, Çizelge 21 sayfa 166ve Çizelge 22 sayfa 166 her çağrı için gereken yetkileri özetler.

Not: Bu çizelgelerde ad bilgileri, kanallar, istemci bağlantısı kanalları, dinleyiciler, hizmetler ya da kimlik doğrulama bilgileri nesneleri bulunmaz. Bunun nedeni, diğer nesnelerle aynı yetkilerin geçerli olduğu MQOO_INQUIRE dışında, bu nesnelere ilişkin yetkilerin hiçbiri geçerli değildir.

Çizelge 19. MQCONN çağrıları için güvenlik yetkilendirmesi gerekiyor

Bu öğe için yetki gerekiyor:	**Kuyruk nesnesi ("1" sayfa 166)**	**Süreç nesnesi**	**Kuyruk yöneticisi nesnesi**
MQCONN seçeneği	Geçerli değildir	Geçerli değildir	MQZAO_CONNECT

Çizelge 20. MQOPEN çağrıları için güvenlik yetkilendirmesi gerekiyor

Bu öğe için yetki gerekiyor:	**Kuyruk nesnesi ("1" sayfa 166)**	**Süreç nesnesi**	**Kuyruk yöneticisi nesnesi**
MQOO_INQUIRE	MQZAO_INQUIRE **("2" sayfa 166)**	MQZAO_INQUIRE **("2" sayfa 166)**	MQZAO_INQUIRE **("2" sayfa 166)**
MQOO_GÖZ AT	MQZAO_GöZ AT	Geçerli değildir	Çek yok
MQOO_INPUT_ *	MQZAO_INPUT	Geçerli değildir	Çek yok
MQOO_SAVE_ ALL_CONTEXT **("3" sayfa 166)**	MQZAO_INPUT	Geçerli değildir	Geçerli değildir
MQOO_OUTPUT (Normal kuyruk) **("4" sayfa 166)**	MQZAO_OUTPUT	Geçerli değildir	Geçerli değildir
MQOO_PASS_ IDENTITY_CONTEXT **("5" sayfa 166)**	MQZAO_PASS_ KIMLIK_BAĞLAMI	Geçerli değildir	Çek yok
MQOO_PASS_ALL_ BAğLAMı **("5" sayfa 166, "6" sayfa 166)**	MQZAO_PASS _ALL_CONTEXT	Geçerli değildir	Çek yok
MQOO_SET_ IDENTITY_CONTEXT **("5" sayfa 166, "6" sayfa 166)**	MQZAO_SET_ KIMLIK_XX_ENCODE_CA SE_CAPS_LOCK_OFF BAĞLAMI	Geçerli değildir	MQZAO_SET_ IDENTITY_CONTEXT **("7" sayfa 166)**
MQOO_SET_ ALL_CONTEXT **("5" sayfa 166, "8" sayfa 166)**	MQZAO_SET_ ALL_CONTEXT	Geçerli değildir	MQZAO_SET_ ALL_CONTEXT **("7" sayfa 166)**
MQOO_OUTPUT (İletim kuyruğu) **("9" sayfa 167)**	MQZAO_SET_ ALL_CONTEXT	Geçerli değildir	MQZAO_SET_ ALL_CONTEXT **("7" sayfa 166)**
MQOO_SET	MQZAO_KüMESI	Geçerli değildir	Çek yok
MQOO_ALTERNATE_ KULLANıCı_XX_ENCODE_ CASE_ONE yetkisi	**("10" sayfa 167)**	**("10" sayfa 167)**	MQZAO_ALTERNATE_ USER_AUTHORITY **("10" sayfa 167, "11" sayfa 167)**

Çizelge 21. MQPUT1 çağrıları için gereken güvenlik yetkisi			
Bu öğe için yetki gerekiyor:	**Kuyruk nesnesi ("1" sayfa 166)**	**Süreç nesnesi**	**Kuyruk yöneticisi nesnesi**
MQPMO_PASS_ KIMLIK_BAĞLAMI	MQZAO_PASS_ IDENTITY_CONTEXT **("12" sayfa 167)**	Geçerli değildir	Çek yok
MQPMO_PASS_ALL _CONTEXT	MQZAO_PASS_ ALL_CONTEXT **("12" sayfa 167)**	Geçerli değildir	Çek yok
MQPMO_SET_ KIMLIK_XX_ENCODE_CA SE_CAPS_LOCK_OFF BAĞLAMI	MQZAO_SET_ IDENTITY_CONTEXT **("12" sayfa 167)**	Geçerli değildir	MQZAO_SET_ IDENTITY_CONTEXT **("7" sayfa 166)**
MQPMO_SET_ ALL_CONTEXT	MQZAO_SET_ ALL_CONTEXT **("12" sayfa 167)**	Geçerli değildir	MQZAO_SET_ ALL_CONTEXT **("7" sayfa 166)**
(İletim kuyruğu) **("9" sayfa 167)**	MQZAO_SET_ ALL_CONTEXT	Geçerli değildir	MQZAO_SET_ ALL_CONTEXT **("7" sayfa 166)**
MQPMO_ALTERNATE_ KULLANıCı_XX_ENCODE_ CASE_ONE yetkisi	**("13" sayfa 167)**	Geçerli değildir	MQZAO_ALTERNATE_ USER_AUTHORITY **("11" sayfa 167)**

Çizelge 22. MQCLOSE çağrıları için güvenlik yetkilendirmesi gerekiyor			
Bu öğe için yetki gerekiyor:	**Kuyruk nesnesi ("1" sayfa 166)**	**Süreç nesnesi**	**Kuyruk yöneticisi nesnesi**
MQCO_DELETE	MQZAO_DELETE **("14" sayfa 167)**	Geçerli değildir	Geçerli değildir
MQCO_DELETE _PURGE	MQZAO_DELETE **("14" sayfa 167)**	Geçerli değildir	Geçerli değildir

Tablolara ilişkin notlar:

1. Bir model kuyruğu açılırsa:
   - Model kuyruğu için, açmakta olduğunuz erişim tipine ilişkin model kuyruğunu açma yetkisinin yanı sıra, MQZAO_DISPLAY yetkisi de gereklidir.
   - Dinamik kuyruğu yaratmak için MQZAO_CREATE yetkisi gerekmez.
   - Model kuyruğunu açmak için kullanılan kullanıcı kimliğine, yaratılan dinamik kuyruk için kuyruğa özgü tüm yetkiler (MQZAO_ALL ile eşdeğer) otomatik olarak verilir.
2. Açılmakta olan nesnenin tipine bağlı olarak, kuyruk, işlem, ad listesi ya da kuyruk yöneticisi nesnesi denetlenir.
3. MQOO_INPUT_ * da belirtilmelidir. Bu seçenek, yerel, model ya da diğer ad kuyruğu için geçerlidir.
4. Bu denetim, "9" sayfa 167notunda belirtilen vaka dışında tüm çıkış vakaları için gerçekleştirilir.
5. MQOO_OUTPUT da belirtilmelidir.
6. MQOO_PASS_IDENTITY_CONTEXT de bu seçenek tarafından örtük olarak belirtilmiştir.
7. Bu yetki, hem kuyruk yöneticisi nesnesi hem de belirli bir kuyruk için gereklidir.
8. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT ve MQOO_SET_IDENTITY_CONTEXT de bu seçenek tarafından örtük olarak belirtilmiştir.

9. Bu denetim, MQUS_ILETIM kuyruğunun *Kullanım* kuyruk özniteliğine sahip yerel ya da model kuyruğu için gerçekleştirilir ve çıkış için doğrudan açılır. Uzak kuyruk açıldığında (uzak kuyruk yöneticisi ve uzak kuyruk adları belirtilerek ya da uzak kuyruğun yerel tanımının adı belirtilerek) bu işlem uygulanmaz.
10. En az bir MQOO_INQUIRE (herhangi bir nesne tipi için) ya da (kuyruklar için) MQOO_BROWSE, MQOO_INPUT_ *, MQOO_OUTPUT ya da MQOO_SET de belirtilmelidir. Gerçekleştirilen denetim, belirli bir nesne yetkisi için sağlanan diğer kullanıcı kimliği ve MQZAO_ALTERNATE_USER_IDENTIFIER denetimi için yürürlükteki uygulama yetkisi kullanılarak, belirtilen diğer seçenekler içindir.
11. Bu yetki, herhangi bir *AlternateUserkimliği* belirtilmesine izin verir.
12. Kuyrukta MQUS_ILETIM kuyruğunun *Kullanım* kuyruk özniteliği yoksa, bir MQZAO_OUTPUT denetimi de gerçekleştirilir.
13. Yürütülen denetim, belirtilen kuyruk yetkisi için sağlanan diğer kullanıcı kimliği ve MQZAO_ALTERNATE_USER_IDENTIFIER denetimi için yürürlükteki uygulama yetkisi kullanılarak, belirtilen diğer seçenekler içindir.
14. Denetim yalnızca aşağıdaki deyimlerin her ikisi de doğruysa gerçekleştirilir:
    - Kalıcı dinamik kuyruk kapatılıyor ve siliniyor.
    - Kuyruk, kullanılmakta olan nesne tanıtıcısını döndüren MQOPEN tarafından yaratılmadı.

    Aksi takdirde, çek olmaz.

Genel notlar:

1. Özel yetki MQZAO_ALL_MQI, nesne tipiyle ilgili tüm yetkileri içerir:
   - MQZAO_CONNECT
   - MQZAO_INQUIRE
   - MQZAO_KüMESI
   - MQZAO_GöZ AT
   - 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 (bkz. not “14” sayfa 167) ve MQZAO_DISPLAY, denetim yetkileri olarak sınıflandırılır. Bu nedenle MQZAO_ALL_MQI 'a dahil edilmezler.
3. *Denetim yok* , yetki denetiminin gerçekleştirilmediği anlamına gelir.
4. *Uygulanamaz* , yetki denetiminin bu işlemle ilgili olmadığı anlamına gelir. Örneğin, bir süreç nesnesine MQPUT çağrısı veremezsiniz.

IBM i *IBM i üzerindeki kaçış PCF ' lerinde MQSC komutlarına ilişkin yetkiler*

Bu yetkiler, kullanıcının denetim komutlarını kaçış PCF iletisi olarak vermesine olanak sağlar. Bu yöntemler, bir programın bir kuyruk yöneticisine ileti olarak bir denetim komutu göndermesini ve o kullanıcı adına yürütülmesini sağlar.

Bu kısım, Escape PCF ' de bulunan her MQSC komutu için gereken yetkileri özetler.

Uygulanamaz , yetki denetiminin bu işlemle ilgili olmadığı anlamına gelir.

Komutu gönderen programın çalıştırıldığı kullanıcı kimliğinin de aşağıdaki yetkileri olmalıdır:

- Kuyruk yöneticisi için MQZAO_CONNECT yetkisi
- PCF komutlarını gerçekleştirmek için kuyruk yöneticisinde DISPLAY yetkisi

- Escape PCF komutunun metni içinde MQSC komutlarını verme yetkisi

ALTER *nesnesi*

Nesne	Yetki gerekli
Kuyruk	MQZAO_CHANGE
Konu	MQZAO_CHANGE
Süreç	MQZAO_CHANGE
Kuyruk yöneticisi	MQZAO_CHANGE
Ad listesi	MQZAO_CHANGE
Kimlik doğrulama bilgileri	MQZAO_CHANGE
Kanal	MQZAO_CHANGE
İstemci bağlantı kanalı	MQZAO_CHANGE
Dinleyici	MQZAO_CHANGE
Hizmet	MQZAO_CHANGE

CLEAR *nesne*

Nesne	Yetki gerekli
Kuyruk	MQZAO_CLEAR
Konu	MQZAO_CLEAR
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	Geçerli değildir
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir
Hizmet	Geçerli değildir

DEFINE *object* NOREPLACE (“1” sayfa 171)

Nesne	Yetki gerekli
Kuyruk	MQZAO_CREATE **(“2” sayfa 171)**
Konu	MQZAO_CREATE **(“2” sayfa 171)**
Süreç	MQZAO_CREATE **(“2” sayfa 171)**
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	MQZAO_CREATE **(“2” sayfa 171)**
Kimlik doğrulama bilgileri	MQZAO_CREATE **(“2” sayfa 171)**
Kanal	MQZAO_CREATE **(“2” sayfa 171)**
İstemci bağlantı kanalı	MQZAO_CREATE **(“2” sayfa 171)**

Nesne	Yetki gerekli
Dinleyici	MQZAO_CREATE **(“2” sayfa 171)**
Hizmet	MQZAO_CREATE **(“2” sayfa 171)**

DEFINE *object* REPLACE (“1” sayfa 171, “3” sayfa 172)

Nesne	Yetki gerekli
Kuyruk	MQZAO_CHANGE
Konu	MQZAO_CHANGE
Süreç	MQZAO_CHANGE
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	MQZAO_CHANGE
Kimlik doğrulama bilgileri	MQZAO_CHANGE
Kanal	MQZAO_CHANGE
İstemci bağlantı kanalı	MQZAO_CHANGE
Dinleyici	MQZAO_CHANGE
Hizmet	MQZAO_CHANGE

***nesneyi* SIL**

Nesne	Yetki gerekli
Kuyruk	MQZAO_DELETE
Konu	MQZAO_DELETE
Süreç	MQZAO_DELETE
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	MQZAO_DELETE
Kimlik doğrulama bilgileri	MQZAO_DELETE
Kanal	MQZAO_DELETE
İstemci bağlantı kanalı	MQZAO_DELETE
Dinleyici	MQZAO_DELETE
Hizmet	MQZAO_DELETE

***NESNEYI* GöRüNTüLE**

Nesne	Yetki gerekli
Kuyruk	MQZAO_DISPLAY
Konu	MQZAO_DISPLAY
Süreç	MQZAO_DISPLAY
Kuyruk yöneticisi	MQZAO_DISPLAY
Ad listesi	MQZAO_DISPLAY
Kimlik doğrulama bilgileri	MQZAO_DISPLAY

Nesne	Yetki gerekli
Kanal	MQZAO_DISPLAY
İstemci bağlantı kanalı	MQZAO_DISPLAY
Dinleyici	
Hizmet	

PING KANALı

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir
Hizmet	Geçerli değildir

KANALI ILK DURUMUNA GETIR

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL_EXTENDED
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir
Hizmet	Geçerli değildir

KANAL ÇöZüMLE

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir

Nesne	Yetki gerekli
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL_EXTENDED
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir
Hizmet	Geçerli değildir

START *nesne*

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	MQZAO_CONTROL
Hizmet	MQZAO_CONTROL

STOP *nesne*

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	MQZAO_CONTROL
Hizmet	MQZAO_CONTROL

Not:

1. DEFINE komutları için, LIKE nesnesi için ya da uygun SYSTEM.DEFAULT.xxx nesnesi.
2. MQZAO_CREATE yetkisi belirli bir nesneye ya da nesne tipine özgü değil. `GRTMQMAUT` komutunda QMGR nesne tipi belirlenerek, belirtilen bir kuyruk yöneticisine ilişkin tüm nesneler için yaratma yetkisi verilir.

3. Bu seçenek, değiştirilecek nesne zaten varsa geçerlidir. Yoksa, denetim DEFINE *object* NOREPLACE ile ilgili olur.

IBM i *IBM i üzerinde PCF komutlarına ilişkin yetkiler*

Bu yetkiler, kullanıcının denetim komutlarını PCF komutları olarak vermesine olanak sağlar. Bu yöntemler, bir programın bir kuyruk yöneticisine ileti olarak bir denetim komutu göndermesini ve o kullanıcı adına yürütülmesini sağlar.

Bu bölümde, her PCF komutu için gereken yetkiler özetlenmektedir.

Denetim yok , yetki denetiminin gerçekleştirilmediği anlamına gelir; *Uygulanamaz* , yetki denetiminin bu işlemle ilgili olmadığı anlamına gelir.

Komutu gönderen programın çalıştırıldığı kullanıcı kimliğinin de aşağıdaki yetkileri olmalıdır:

- Kuyruk yöneticisi için MQZAO_CONNECT yetkisi
- PCF komutlarını gerçekleştirmek için kuyruk yöneticisinde DISPLAY yetkisi

MQZAO_ALL_ADMIN özel yetkilendirmesi aşağıdaki yetkileri içerir:

- MQZAO_CHANGE
- MQZAO_CLEAR
- MQZAO_DELETE
- MQZAO_DISPLAY
- MQZAO_CONTROL
- MQZAO_CONTROL_EXTENDED

MQZAO_CREATE, belirli bir nesneye ya da nesne tipine özgü olmadığı için dahil edilmedi

***Nesneyi* Değiştir**

Nesne	**Yetki gerekli**
Kuyruk	MQZAO_CHANGE
Konu	MQZAO_CHANGE
Süreç	MQZAO_CHANGE
Kuyruk yöneticisi	MQZAO_CHANGE
Ad listesi	MQZAO_CHANGE
Kimlik doğrulama bilgileri	MQZAO_CHANGE
Kanal	MQZAO_CHANGE
İstemci bağlantı kanalı	MQZAO_CHANGE
Dinleyici	MQZAO_CHANGE
Hizmet	MQZAO_CHANGE

***nesneyi* temizle**

Nesne	**Yetki gerekli**
Kuyruk	MQZAO_CLEAR
Konu	MQZAO_CLEAR
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir

Nesne	Yetki gerekli
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	Geçerli değildir
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir
Hizmet	Geçerli değildir

nesneyi kopyala **(değiştirmeden) (“1” sayfa 177)**

Nesne	Yetki gerekli
Kuyruk	MQZAO_CREATE **(“2” sayfa 177)**
Konu	MQZAO_CREATE **(“2” sayfa 177)**
Süreç	MQZAO_CREATE **(“2” sayfa 177)**
Kuyruk yöneticisi	Geçerli değildir
NamelistMQZAO_CREATE	MQZAO_CREATE **(“2” sayfa 177)**
Kimlik doğrulama bilgileri	MQZAO_CREATE **(“2” sayfa 177)**
Kanal	MQZAO_CREATE **(“2” sayfa 177)**
İstemci bağlantı kanalı	MQZAO_CREATE **(“2” sayfa 177)**
Dinleyici	MQZAO_CREATE **(“2” sayfa 177)**
Hizmet	MQZAO_CREATE **(“2” sayfa 177)**

nesneyi kopyala **(başkasıyla değiştir) (“1” sayfa 177, “4” sayfa 177)**

Nesne	Yetki gerekli
Kuyruk	MQZAO_CHANGE
Konu	MQZAO_CHANGE
Süreç	MQZAO_CHANGE
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	MQZAO_CHANGE
Kimlik doğrulama bilgileri	MQZAO_CHANGE
Kanal	MQZAO_CHANGE
İstemci bağlantı kanalı	MQZAO_CHANGE
Dinleyici	MQZAO_CHANGE
Hizmet	MQZAO_CHANGE

Nesne **yarat (değiştirmeden) (“3” sayfa 177)**

Nesne	Yetki gerekli
Kuyruk	MQZAO_CREATE **(“2” sayfa 177)**
Konu	MQZAO_CREATE **(“2” sayfa 177)**
Süreç	MQZAO_CREATE **(“2” sayfa 177)**

Nesne	Yetki gerekli
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	MQZAO_CREATE **(“2” sayfa 177)**
Kimlik doğrulama bilgileri	MQZAO_CREATE **(“2” sayfa 177)**
Kanal	MQZAO_CREATE **(“2” sayfa 177)**
İstemci bağlantı kanalı	MQZAO_CREATE **(“2” sayfa 177)**
Dinleyici	MQZAO_CHANGE
Hizmet	MQZAO_CHANGE

***Nesne* yarat (başkasıyla değiştir) (“3” sayfa 177, “4” sayfa 177)**

Nesne	Yetki gerekli
Kuyruk	MQZAO_CHANGE
Konu	MQZAO_CHANGE
Süreç	MQZAO_CHANGE
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	MQZAO_CHANGE
Kimlik doğrulama bilgileri	MQZAO_CHANGE
Kanal	MQZAO_CHANGE
İstemci bağlantı kanalı	MQZAO_CHANGE
Dinleyici	MQZAO_CHANGE
Hizmet	MQZAO_CHANGE

***Nesne* ' ni sil**

Nesne	Yetki gerekli
Kuyruk	MQZAO_DELETE
Konu	MQZAO_DELETE
Süreç	MQZAO_DELETE
Kuyruk yöneticisi	MQZAO_DELETE
Ad listesi	MQZAO_DELETE
Kimlik doğrulama bilgileri	MQZAO_DELETE
Kanal	MQZAO_DELETE
İstemci bağlantı kanalı	MQZAO_DELETE
Dinleyici	MQZAO_DELETE
Hizmet	MQZAO_DELETE

***nesne* nesnesini sorgularken**

Nesne	Yetki gerekli
Kuyruk	MQZAO_DISPLAY

Nesne	Yetki gerekli
Konu	MQZAO_DISPLAY
Süreç	MQZAO_DISPLAY
Kuyruk yöneticisi	MQZAO_DISPLAY
Ad listesi	MQZAO_DISPLAY
Kimlik doğrulama bilgileri	MQZAO_DISPLAY
Kanal	MQZAO_DISPLAY
İstemci bağlantı kanalı	MQZAO_DISPLAY
Dinleyici	MQZAO_DISPLAY
Hizmet	MQZAO_DISPLAY

***Nesne* adlarını sorma**

Nesne	Yetki gerekli
Kuyruk	Çek yok
Konu	Çek yok
Süreç	Çek yok
Kuyruk yöneticisi	Çek yok
Ad listesi	Çek yok
Kimlik doğrulama bilgileri	Çek yok
Kanal	Çek yok
İstemci bağlantı kanalı	Çek yok
Dinleyici	Çek yok
Hizmet	Çek yok

Ping Kanalı

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir
Hizmet	Geçerli değildir

Kanalı Sıfırla

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL_EXTENDED
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir
Hizmet	Geçerli değildir

Kuyruk İstatistiklerini Sıfırla

Nesne	Yetki gerekli
Kuyruk	MQZAO_DISPLAY ve MQZAO_CHANGE
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	Geçerli değildir
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	
Hizmet	

Kanalı Çözümle

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL_EXTENDED
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir

Nesne	Yetki gerekli
Hizmet	Geçerli değildir

Kanalı Başlat

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir
Hizmet	Geçerli değildir

Kanalı Durdur

Nesne	Yetki gerekli
Kuyruk	Geçerli değildir
Konu	Geçerli değildir
Süreç	Geçerli değildir
Kuyruk yöneticisi	Geçerli değildir
Ad listesi	Geçerli değildir
Kimlik doğrulama bilgileri	Geçerli değildir
Kanal	MQZAO_CONTROL
İstemci bağlantı kanalı	Geçerli değildir
Dinleyici	Geçerli değildir
Hizmet	Geçerli değildir

Not:

1. Kopyalama komutları için, Kaynak nesne için MQZAO_DISPLAY yetkisi de gereklidir.
2. MQZAO_CREATE yetkisi belirli bir nesneye ya da nesne tipine özgü değil. GRTMQMAUT komutunda QMGR nesne tipi belirlenerek, belirtilen bir kuyruk yöneticisine ilişkin tüm nesneler için yaratma yetkisi verilir.
3. Yaratma komutları için, uygun SYSTEM.DEFAULT.* nesne.
4. Bu seçenek, değiştirilecek nesne zaten varsa geçerlidir. Tersi durumda, denetim Copy (Kopyalama) ya da Create (Değiştirmeden Yarat) içindir.

IBM i IBM i üzerinde soysal OAM profilleri

Nesne yetkisi yöneticisi (OAM) soysal tanıtımları, yaratıldığında her nesne için ayrı **GRTMQMAUT** komutları yayınlamak yerine, bir kullanıcının aynı anda birden çok nesne için sahip olduğu yetkiyi ayarlamanızı

sağlar. **GRTMQMAUT** komutunda soysal tanıtımların kullanılması, o tanıtıma uyan, ileride yaratılacak tüm nesneler için soysal bir yetki belirlemenizi sağlar.

Bu bölümün geri kalanında, soysal profillerin kullanımı daha ayrıntılı olarak açıklanmaktadır:

- "Joker Karakterlerin Kullanılması" sayfa 178
- "Profil öncelikleri" sayfa 178

Joker Karakterlerin Kullanılması

Bir tanıtımı soysal yapan şey, tanıtım adında özel karakterlerin (genel arama karakterleri) kullanılmasıdır. Örneğin, soru işareti (?) genel arama karakteri, bir addaki herhangi bir tek karakterle eşleşir. Bu nedenle, `ABC.?EF`değerini belirlerseniz, o tanıtıma ilişkin yetki `ABC.DEF`, `ABC.CEF`, `ABC.BEF`gibi adlarla yaratılan nesneler için geçerlidir.

Kullanılabilecek genel arama karakterleri şunlardır:

?

Herhangi bir tek karakter yerine soru işaretini (?) kullanın. Örneğin, `AB.?D` , `AB.CD`, `AB.ED`ve `AB.FD`nesnelerine uygulanır.

*****

Yıldız işaretini (*) aşağıdaki gibi kullanın:

- Bir profil adında, nesne adındaki herhangi bir niteleyiciyle eşleşecek *niteleyici* . Niteleyici, bir nokta ile sınırlanmış bir nesne adının parçasıdır. Örneğin, `ABC.DEF.GHI`içinde niteleyiciler şunlardır: `ABC`, `DEF`ve `GHI`.

  Örneğin, `ABC.*.JKL` , `ABC.DEF.JKL`ve `ABC.GHI.JKL`nesnelerine uygulanır. (`ABC.JKL` için geçerli **olmayacağına** dikkat edin; * bu bağlamda kullanılan her zaman bir niteleyiciyi gösterir.)
- Bir profil adındaki niteleyici içindeki, bir nesne adındaki niteleyici içindeki sıfır ya da daha fazla karakterle eşleşecek karakter.

  Örneğin, `ABC.DE*.JKL` , `ABC.DE.JKL`, `ABC.DEF.JKL`ve `ABC.DEGH.JKL`nesnelerine uygulanır.

******

Profil adında çift yıldız işaretini (**) ***bir kez*** kullanın:

- Tüm nesne adlarıyla eşleşecek tüm profil adı. Örneğin, süreçleri tanımlamak için `OBJTYPE (*PRC)` anahtar sözcüğünü kullanırsanız, tanıtım adı olarak ** değerini kullanırsanız, tüm süreçlere ilişkin yetkileri değiştirirsiniz.
- Bir tanıtım adında, nesne adında sıfır ya da daha fazla niteleyiciyle eşleşecek başlangıç, ikinci ya da bitiş niteleyicisi olarak. Örneğin, `**.ABC` son niteleyicisi ABC olan tüm nesneleri tanıtır.

Profil öncelikleri

Soysal tanıtımlar kullanılırken anlaşılması gereken önemli bir nokta, yaratılmakta olan bir nesneye uygulanacak yetkiler belirlenirken tanıtımların verildiği önceliktir. Örneğin, şu komutları yayınladığınızı varsayalım:

```
GRTMQMAUT OBJ(AB.*) OBJTYPE(*Q) USER(FRED) AUT(*PUT) MQMNAME(MYQMGR)
GRTMQMAUT OBJ(AB.C*) OBJTYPE(*Q) USER(FRED) AUT(*GET) MQMNAME(MYQMGR)
```

İlki, FRED birincil kullanıcısına ilişkin tüm kuyruklara ilişkin, AB. * tanıtımıyla eşleşen adlarla koyma yetkisi verir; İkincisi, AB.C*.

Şimdi AB.CD. Joker karakter eşleştirmeye ilişkin kurallara göre, bu kuyruğa GRTMQMAUT uygulanabilir. Yani, otoriteye sahip mi?

Yanıtı bulmak için, bir nesneye birden çok profil uygulanabildiğinde **yalnızca en özel profilin geçerli olduğu**kuralını uygulayabilirsiniz. Bu kuralı uygulama şekliniz, profil adlarını soldan sağa karşılaştırmaktır. Farklı oldukları her yerde, soysal olmayan bir karakter soysal bir karakterden daha özgüdür. Bu nedenle, önceki örnekte, kuyruk AB.CD ' nin **get** yetkisi (AB.C*, AB. * ' den daha özeldir).

Soysal karakterleri karşılaştırırken *özgüllük* sırası şöyledir:

1. ?
2. *
3. **

IBM i IBM i üzerinde kurulu yetkilendirme hizmetinin belirtilmesi

Hangi yetkilendirme hizmeti bileşeninin kullanılacağını belirtebilirsiniz.

GRTMQMAUT ve **RVKMQMAUT** üzerindeki **Service Component name** değiştirgesi, kurulu yetki hizmeti bileşeninin adını belirtmenizi sağlar.

Her iki komutun sonraki panosunda **F24** , ardından **F9=All parametreleri** seçildiğinde, kurulu yetki bileşenini (*DFT) ya da kuyruk yöneticisinin qm.ini dosyasının Hizmet alanında belirtilen yetki hizmeti bileşeninin adını belirlemenizi sağlar.

DSPMQMAUT ' in bu ek parametresi de vardır. Bu parametre, belirlenen nesne adı, nesne tipi ve kullanıcı için kurulu tüm yetki bileşenlerini (*DFT) ya da belirlenen yetki hizmeti bileşen adını aramanızı sağlar.

IBM i IBM i üzerinde yetki tanıtımlarıyla ve yetki olmadan çalışma

Yetki tanıtımlarıyla nasıl çalışacağınızı ve yetki tanıtımları olmadan nasıl çalışacağınızı öğrenmek için bu bilgileri kullanın.

Burada açıklandığı gibi, “Yetki tanıtımlarıyla çalışma” sayfa 179içinde açıklandığı gibi ya da bunlar olmadan yetki profilleriyle çalışabilirsiniz:

Yetki tanıtımları olmadan çalışmak için, yetki olmadan tanıtım yaratmak üzere **GRTMQMAUT** üzerinde Yetki değiştirgesi olarak *NONE değiştirgesini kullanın. Bu, var olan profilleri değişmeden bırakır.

RVKMQMAUT' ta, var olan bir yetki tanıtımını kaldırmak için Yetki parametresi olarak *REMOVE değerini kullanın.

Yetki tanıtımlarıyla çalışma

Yetki profili oluşturmayla ilişkili iki komut vardır:

- **WRKMQMAUT**
- **WRKMQMAUTD**

Bu komutlara doğrudan komut satırından ya da WRKMQM panosundan aşağıdaki işlemleri yaparak erişebilirsiniz:

1. Kuyruk yöneticisi adını yazın ve **WRKMQM** sonuç panosuna erişmek için Enter tuşuna basın.
2. Bu panoda F23=More options seçiliyor.

Seçenek 24, **WRKMQMAUT** komut için sonuç panosunu seçer ve seçenek 25, SSL bağ tanımları katmanıyla kullanılan **WRKMQMAUTI** komutunu seçer.

WRKMQMAUT

Bu komut, yetki kuyruğunda tutulan yetki verileriyle çalışmanızı sağlar.

Not: Bu komutu çalıştırmak için kuyruk yöneticisi üzerinde *connect ve *admdsp yetkinizin olması gerekir. Ancak, tanıtım yaratmak ya da silmek için QMQMADM yetkinizin olması gerekir.

Bilgileri ekrana verdiğinizde, tipleriyle birlikte bir yetki tanıtımı adları listesi görüntülenir. Çıktıyı yazdırırsanız, tüm yetki verilerinin, kayıtlı kullanıcıların ve yetkilerinin ayrıntılı bir listesini alırsınız.

Bu panoda bir nesne ya da profil adı girilmesi ve ENTER tuşuna basılması, **WRKMQMAUT** için sonuçlar panosuna geçmenizi sağlar.

4=Deleteseçeneğini belirlerseniz, belirlediğiniz soysal yetki tanıtımı adına kayıtlı tüm kullanıcı adlarını silmek istediğinizi onaylayabileceğiniz yeni bir panoya gidersiniz. Bu seçenek, tüm kullanıcılar için *REMOVE seçeneğiyle **RVKMQMAUT** komutunu çalıştırır ve soysal tanıtım adlarına **yalnızca** uygular.

12=Work with profile seçeneğini belirlerseniz, “WRKMQMAUTD” sayfa 180içinde açıklandığı gibi **WRKMQMAUTD** komut sonuçları panosuna gidersiniz.

WRKMQMAUTD

Bu komut, belirli bir yetki tanıtımı adı ve nesne tipiyle kayıtlı tüm kullanıcıları görüntülemenizi sağlar. Bu komutu çalıştırmak için kuyruk yöneticisi üzerinde *connect ve *admdsp yetkinizin olması gerekir. Ancak, bir tanıtım vermek, çalıştırmak, yaratmak ya da silmek için QMQMADM yetkinizin olması gerekir.

İlk giriş panosundan F24=More keys seçeneğinin ardından F9=All Parameters seçeneği seçildiğinde, **GRTMQMAUT** ve **RVKMQMAUT**için Service Component Name (Hizmet Bileşeni Adı) görüntülenir.

Not: F11=Display Object Authorizations tuşu, aşağıdaki yetki tipleri arasında geçiş yapar:

- Nesne yetkileri
- Bağlam yetkileri
- MQI yetkileri

Ekrandaki seçenekler şunlardır:

2=Grant
Geçerli yetkilere eklemek için sizi **GRTMQMAUT** panosuna götürür.

3=Revoke
Yürürlükteki tanımlamalardan bazılarını kaldırmak için **RVKMQMAUT** panosuna gidin

4=Delete
Belirlenen kullanıcılara ilişkin yetki verilerini silmenizi sağlayan bir panoya götürür. Bu, *REMOVE seçeneğiyle **RVKMQMAUT** komutunu çalıştırır.

5=Display
Sizi var olan **DSPMQMAUT** komutuna götürür

F6=Create
Bir tanıtım yetki kaydı yaratmanızı sağlayan **GRTMQMAUT** panosuna gidin.

IBM i ile ilgili Nesne Yetkilisi Yöneticisi yönergeleri

Nesne yetki yöneticisini (OAM) kullanmaya ilişkin ek ipuçları

Hassas işlemlere erişimi sınırla

Bazı işlemler hassastır; ayrıcalıklı kullanıcılarla sınırlandırın. Örneğin,

- İletim kuyrukları ya da komut kuyruğu gibi bazı özel kuyruklara erişilmesi SYSTEM.ADMIN.COMMAND.QUEUE
- Tam MQI bağlamı seçeneklerini kullanan programların çalıştırılması
- Uygulama kuyruklarının yaratılması ve kopyalanması

Kuyruk yöneticisi dizinleri

Kuyruk ve diğer kuyruk yöneticisi verilerini içeren dizinler ve kitaplıklar ürün için özeldir. MQI kaynaklarına yetki vermek ya da yetkileri iptal etmek için standart işletim sistemi komutlarını kullanmayın.

Kuyruklar

Dinamik bir kuyruk için yetki, türetildiği model kuyruğuyla aynı olmasına karşın mutlaka aynı değildir.

Diğer ad kuyrukları ve uzak kuyruklar için yetki, diğer adın ya da uzak kuyruğun çözüldüğü kuyruk değil, nesnenin kendisi içindir. Bir kullanıcı tanıtımına, kullanıcı tanıtımının erişim izni olmayan bir yerel kuyruğa çözülen bir diğer ad kuyruğuna erişim yetkisi verilebilir.

Ayrıcalıklı kullanıcılar için kuyruk yaratma yetkisini sınırlayın. Bunu yapmazsanız, kullanıcılar bir diğer ad yaratarak normal erişim denetimini atlayabilir.

Diğer kullanıcı yetkisi

Diğer kullanıcı yetkisi, bir kullanıcı tanıtımının bir IBM MQ nesnesine erişirken başka bir kullanıcı tanıtımının yetkisini kullanıp kullanamayacağını denetler. Bu teknik, bir sunucu bir programdan istek alırsa ve sunucu, programın istek için gerekli yetkiye sahip olduğundan emin olmak isterse gereklidir. Sunucu gerekli yetkiye sahip olabilir, ancak programın istediği işlemlere ilişkin yetkiye sahip olup olmadığını bilmesi gerekir.

Örneğin:

- PAYSERV kullanıcı profili altında çalışan bir sunucu programı, USER1kullanıcı profili tarafından kuyruğa konan bir kuyruktan bir istek iletisi alır.
- Sunucu programı istek iletisini aldığında, isteği işler ve yanıtı istek iletisiyle belirlenen yanıt kuyruğuna geri koyar.
- Sunucu, yanıt kuyruğunun açılmasını yetkilendirmek için kendi kullanıcı profilini (PAYSERV) kullanmak yerine, USER1adlı başka bir kullanıcı profili belirtebilir. Bu örnekte, PAYSERV ' nin yanıt kuyruğunu açtığında alternatif kullanıcı profili olarak USER1 belirtip belirtmeyeceğini denetlemek için alternatif kullanıcı yetkisini kullanabilirsiniz.

Diğer kullanıcı tanıtımı, nesne tanımlayıcısının *AlternateUserId* alanında belirtilir.

Not: Herhangi bir IBM MQ nesnesinde diğer kullanıcı tanıtımlarını kullanabilirsiniz. Diğer kullanıcı tanıtımının kullanılması, diğer kaynak yöneticileri tarafından kullanılan kullanıcı tanıtımını etkilemez.

Bağlam yetkisi

Bağlam, belirli bir ileti için geçerli olan ve iletinin bir parçası olan ileti tanımlayıcısı MQMD ' de bulunan bilgilerdir.

Bağlamla ilgili ileti tanımlayıcı alanlarının açıklamaları için bkz. MQMD-Message descriptor.

Bağlam seçenekleri hakkında bilgi için bkz. İleti bağlamı.

Uzak güvenlikle ilgili dikkat edilmesi gereken noktalar

Uzak güvenlik için şunları göz önünde bulundurun:

Yetki koy
Kuyruk yöneticileri arasında güvenlik için, bir kanal başka bir kuyruk yöneticisinden gönderilen bir iletiyi aldığında kullanılacak koyma yetkisini belirtebilirsiniz.

Bu parametre yalnızca RCVR, RQSTR ya da CLUSRCVR kanal tipleri için geçerlidir. PUTAUT kanal özniteliğini aşağıdaki gibi belirtin:

DEF
Varsayılan kullanıcı profili. Bu, ileti kanalı aracısının altında çalıştığı QMQM kullanıcı profilidir.

CTX
İleti bağlamındaki kullanıcı profili.

İletim kuyrukları
Kuyruk yöneticileri uzak iletileri otomatik olarak bir iletim kuyruğuna yerleştirir; özel bir yetki gerekmez. Ancak, iletiyi doğrudan bir iletim kuyruğuna koymak için özel yetki gerekir.

Kanal çıkışları
Kanal çıkışları, ek güvenlik için kullanılabilir.

Kanal kimlik doğrulama kayıtları
Kanal düzeyinde bağlanan sistemlere verilen erişim üzerinde daha kesin denetim sağlamak için kullanılır.

Uzak güvenlikle ilgili daha fazla bilgi için bkz. “Kanal yetkilendirmesi” sayfa 112.

SSL/TLS ile kanalları koruma

TLS (Transport Layer Security; İletim Katmanı Güvenliği) protokolü, gizlice dinleme, kurcalama ve taklit edilmeye karşı koruma ile kanal güvenliği sağlar. TLS için IBM MQ desteği, kanal tanımında belirli bir kanalın TLS güvenliğini kullandığını belirtmenize olanak sağlar. Kullanmak istediğiniz şifreleme algoritması gibi, istediğiniz güvenliğin ayrıntılarını da belirleyebilirsiniz.

IBM MQ içindeki TLS desteği, kuyruk yöneticisi *kimlik doğrulama bilgileri nesnesini* ve çeşitli CL ve MQSC komutlarını ve kuyruk yöneticisi ve kanal parametrelerini kullanır.

Aşağıdaki CL komutları TLS ' yi destekler:

WRKMQMAUTI
Bir kimlik doğrulama bilgileri nesnesinin öznitelikleriyle çalışabilmenizi sağlar.

CHGMQMAUTI
Bir kimlik doğrulama bilgileri nesnesinin özniteliklerini değiştirin.

CRTMQMAUTI
Bir kimlik doğrulama bilgileri nesnesi oluşturun.

CPYMQMAUTI
Var olan bir kimlik doğrulama bilgisi nesnesini kopyalayarak bir kimlik doğrulama bilgisi nesnesi oluşturun.

DLTMQMAUTI
Bir kimlik doğrulama bilgileri nesnesini silin.

DSPMQMAUTI
Belirli bir kimlik doğrulama bilgileri nesnesine ilişkin öznitelikleri görüntüler.

TLS kullanan kanal güvenliğine genel bakış için bkz.

- TLS ile kanalları koruma

TLS ile ilişkili PCF komutlarının ayrıntıları için bkz.

- Kimlik Doğrulama Bilgileri Nesnesini Değiştir, Kopyala ve Oluştur
- Kimlik Doğrulama Bilgileri Nesnesini Sil
- Kimlik Doğrulama Bilgileri Nesnesi

z/OS z/OS üzerinde güvenliğin ayarlanması

z/OS' e özgü güvenlikle ilgili önemli noktalar.

IBM MQ for z/OS içindeki güvenlik, RACF ya da eşdeğer bir dış güvenlik yöneticisi (ESM) kullanılarak denetlenir.

Aşağıdaki yönergelerde RACFkullandığınız varsayılmıştır.

İlgili kavramlar
Güvenlik senaryosu: z/OS üzerinde iki kuyruk yöneticisi
Güvenlik senaryosu: z/OS üzerinde kuyruk paylaşım grubu

z/OS RACF güvenlik sınıfları

RACF sınıfları, IBM MQ güvenlik denetimi için gereken tanıtımları bulundurmak için kullanılır. Üye sınıflarının çoğunun eşdeğer grup sınıfları vardır. Sınıfları etkinleştirmeniz ve soysal tanıtımları kabul etmelerini sağlamanız gerekir.

Her RACF sınıfı, Çizelge 23 sayfa 183içinde gösterildiği gibi, denetim sırasının bir noktasında kullanılan bir ya da daha çok tanıtımı içerir.

Çizelge 23. IBM MQ tarafından kullanılan RACF sınıfları

Üye sınıfı	Grup sınıfı	İçerik
MQADMIN	GMQADMIN	Temel olarak yönetim işlevleri için kullanılan profiller. Örneğin: • IBM MQ güvenlik anahtarlarına ilişkin profiller. • RESLEVEL güvenlik profili. • Alternatif kullanıcı güvenliğine ilişkin profiller. • Bağlam güvenliğine ilişkin profiller. • Komut kaynağı güvenliğine ilişkin profiller. Bu sınıf yalnızca büyük harf RACF profillerini tutabilir.
MXADMIN	GMXADMIN	Temel olarak yönetim işlevleri için kullanılan profiller. Örneğin: • IBM MQ güvenlik anahtarlarına ilişkin profiller. • RESLEVEL güvenlik profili. • Alternatif kullanıcı güvenliğine ilişkin profiller. • Bağlam güvenliğine ilişkin profiller. • Komut kaynağı güvenliğine ilişkin profiller. Bu sınıf hem büyük harf, hem de büyük harf karışık RACF tanıtımlarını bulundurabilir.
MQCONN		Bağlantı güvenliği için kullanılan tanıtımlar.
MQCMDS		Komut güvenliği için kullanılan profiller.
MQQUEUE	GMQKUYRUğU	Kuyruk kaynağı güvenliğinde kullanılan büyük harfli profiller.
MXQUEUE	GMXQUEUE	Kuyruk kaynak güvenliğinde kullanılan büyük ve küçük harf karışık profiller.
MQPROC	GMQPROC	Süreç kaynak güvenliğinde kullanılan büyük harfli profiller.
MXPROC	GMXPROC	Süreç kaynak güvenliğinde kullanılan büyük ve küçük harf karışık profiller.
MQNLIST	GMQNLIST	Namelist kaynak güvenliğinde kullanılan büyük harfli profiller.
MXNLIST	GMXNLIST	Ad listesi kaynak güvenliğinde kullanılan büyük ve büyük harfli profiller.
MXTOPIC	GMXTOPIC	Konu güvenliğinde kullanılan büyük ve küçük harf karışık profiller.

Bazı sınıfların, benzer erişim gereksinimlerine sahip kaynak gruplarını bir araya getirmenizi sağlayan ilgili bir *grup sınıfı* vardır. Üye ve grup sınıfları arasındaki fark ve bir üye ya da grup sınıfının ne zaman kullanılacağı ile ilgili ayrıntılar için z/OS Security Server RACF Security Administrator's Guideadlı yayına bakın.

Güvenlik denetimlerinin yapılabilmesi için sınıfların etkinleştirilmesi gerekir. Tüm IBM MQ sınıflarını etkinleştirmek için şu RACF komutunu kullanabilirsiniz:

```
SETROPTS CLASSACT(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                  MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

Sınıfları, soysal tanıtımları kabul edebilmeleri için ayarladığınızdan da emin olmanız gerekir. Bunu RACF komutu **SETROPTS**ile de yaparsınız; örneğin:

```
SETROPTS GENERIC(MQADMIN,MXADMIN,MQQUEUE,MXQUEUE,MQPROC,MXPROC,
                 MQNLIST,MXNLIST,MXTOPIC,MQCONN,MQCMDS)
```

z/OS RACF profiller

IBM MQ tarafından kullanılan tüm RACF tanıtımları, kuyruk yöneticisi adı ya da kuyruk paylaşım grubu adı olan bir önek içerir. Joker karakter olarak yüzde işaretini kullanırken dikkatli olun.

IBM MQ tarafından kullanılan tüm RACF tanıtımları bir önek içerir. Kuyruk paylaşım grubu düzeyi güvenliği için bu, kuyruk paylaşım grubu adıdır. Kuyruk yöneticisi düzeyinde güvenlik için önek, kuyruk yöneticisi adıdır. Kuyruk yöneticisi ve kuyruk paylaşım grubu düzeyi güvenliğinin bir karışımını kullanıyorsanız, her iki tip önek içeren tanıtımları kullanırsınız. Kuyruk paylaşım grubu ve kuyruk yöneticisi düzeyi güvenliği IBM MQ for z/OS içindeki güvenlik denetimleri ve seçenekleriiçinde açıklanmıştır.

Örneğin, kuyruk paylaşım grubu düzeyinde QSG1 kuyruk paylaşım grubunda QUEUE_FOR_SUBSCRIBER_LIST adlı bir kuyruğu korumak istiyorsanız, uygun profil RACF olarak tanımlanabilir:

```
RDEFINE MQQUEUE QSG1.QUEUE_FOR_SUBSCRIBER_LIST
```

Kuyruk yöneticisi düzeyinde kuyruk yöneticisi STCD 'sine ait olan QUEUE_FOR_LOST_CARD_LIST adlı bir kuyruğu korumak istiyorsanız, uygun profil RACF olarak tanımlanır:

```
RDEFINE MQQUEUE STCD.QUEUE_FOR_LOST_CARD_LIST
```

Bu, farklı kuyruk yöneticilerinin ve kuyruk paylaşım gruplarının aynı RACF veritabanını paylaşabileceği ve farklı güvenlik seçeneklerine sahip olduğu anlamına gelir.

Beklenmeyen kullanıcı erişimini önlemek için profillerde soysal kuyruk yöneticisi adlarını kullanmayın.

IBM MQ , nesne adlarında yüzde işaretinin (%) kullanılmasına izin verir. Ancak RACF , tek karakterli genel arama karakteri olarak% karakterini kullanır. Bu, adında% karakteri olan bir nesne adı tanımladığınızda, ilgili profili tanımlarken bunu göz önünde bulundurmanız gerektiği anlamına gelir.

Örneğin, kuyruk yöneticisi CRDP ' de CREDIT_CARD_ %_RATE_SORGUSU kuyruğu için profil RACF olarak tanımlanır:

```
RDEFINE MQQUEUE CRDP.CREDIT_CARD_%_RATE_INQUIRY
```

Bu kuyruk, CRDP. * * gibi soysal bir tanıtımla korunamaz.

IBM MQ , nesne adlarında büyük ve küçük harf karışık karakterlerin kullanılmasını sağlar. Aşağıdaki öğeleri tanımlayarak bu nesneleri koruyabilirsiniz:

1. Uygun büyük ve küçük harf karışık RACF sınıflarında karışık büyük/küçük harf profilleri ya da
2. Uygun büyük harf RACF sınıflarındaki soysal tanıtımlar.

Büyük ve küçük harf karışık RACF sınıflarını kullanmak için “z/OS kuyruk yöneticisini büyük ve küçük harf karışık güvenliğe geçirme” sayfa 264içinde açıklanan adımları izlemeniz gerekir.

Yalnızca değerler IBM MQtarafından sağlandığı için büyük harfle kalan bazı profiller ya da profillerin bölümleri vardır. Bunlar:

- Profilleri değiştir.

- Altsistem ve kuyruk paylaşım grubu tanıtıcıları da içinde olmak üzere tüm üst düzey niteleyiciler (HLQ).
- SYSTEM nesnelerine ilişkin tanıtımlar.
- Varsayılan nesnelere ilişkin tanıtımlar.
- **MQCMDS** sınıfı, bu nedenle tüm komut profilleri yalnızca büyük harfli olur.
- **MQCONN** sınıfı, bu nedenle tüm bağlantı tanıtımları yalnızca büyük harfli olur.
- **RESLEVEL** profilleri.
- Komut kaynağı profillerinde `'object'` niteliği; örneğin, `hlq.QUEUE.queuename`. Yalnızca kaynak adı büyük ve küçük harf karışık.
- Dinamik kuyruk profilleri `hlq.CSQOREXX.*`, `hlq.CSQUTIL.*`ve `CSQXCMD.*`.
- `hlq.CONTEXT.resourcename`'un `'CONTEXT'` bölümü.
- `hlq.ALTERNATE.USER.userid`'un `'ALTERNATE.USER'` bölümü.

Örneğin, aşağıdaki yollardan biriyle `PAYROLL.Dept1` kuyruk yöneticisinde `QM01` adlı bir kuyruğa erişim vermek için bir tanıtım tanımlayabilirsiniz.

- Büyük ve küçük harf karışık profiller kullanıyorsanız, aşağıdaki komutu kullanarak IBM MQ RACF sınıfında `MXQUEUE` bir profil tanımlayabilirsiniz:

```
RDEFINE MXQUEUE MQ01.PAYROLL.Dept1
```

- Büyük harfli profiller kullanıyorsanız, aşağıdaki komutu kullanarak IBM MQ RACF sınıfında `MQQUEUE` bir profil tanımlayabilirsiniz:

```
RDEFINE MQQUEUE MQ01.PAYROLL.*
```

İlk örnek, büyük ve küçük harf karışık profiller kullanarak, kaynağa erişim yetkisi verme konusunda daha ayrıntılı denetim sağlar.

z/OS Profilleri değiştir

IBM MQtarafından gerçekleştirilen güvenlik denetimini denetlemek için *anahtar profilleri*kullanılır. Anahtar profili, IBM MQiçin özel bir anlamı olan normal bir RACF profilidir. Anahtar profillerindeki erişim listesi IBM MQtarafından kullanılmaz.

IBM MQ çizelgelerde gösterilen her anahtar tipi için bir iç anahtar sağlar Altsistem düzeyinde güvenlik için anahtar profilleri, Kuyruk paylaşımı grubu ya da kuyruk yöneticisi düzeyinde güvenlik için anahtar profillerive Kaynak denetimi için anahtar profilleri. Anahtar tanıtımları, kuyruk paylaşım grubu düzeyinde ya da kuyruk yöneticisi düzeyinde ya da her ikisinin birleşiminde tutulabilir. Tek bir kuyruk paylaşım grubu güvenlik anahtarı tanıtımları kümesini kullanarak, bir kuyruk paylaşım grubu içindeki tüm kuyruk yöneticilerinin güvenliğini denetleyebilirsiniz.

Bir güvenlik anahtarı açıldığında, anahtarla ilişkili güvenlik denetimleri gerçekleştirilir. Bir güvenlik anahtarı ayarlandığında, anahtarla ilişkili güvenlik denetimleri atlanır. Varsayılan değer, tüm güvenlik anahtarlarının açık durumda olması değeridir.

z/OS *Anahtarlar ve sınıflar*

Bir kuyruk yöneticisini başlattığınızda ya da güvenliği yenilediğinizde, IBM MQ anahtarları çeşitli RACF sınıflarının durumuna göre ayarlar.

Bir kuyruk yöneticisi başlatıldığında (ya da MQADMIN ya da MXADMIN sınıfı IBM MQ REFRESH SECURITY komutu tarafından yenilendiğinde), IBM MQ önce RACF ve uygun sınıfın durumunu denetler:

- Büyük harfli tanıtımlar kullanıyorsanız MQADMIN sınıfı
- Karışık büyük/küçük harf profili kullanıyorsanız MXADMIN sınıfı.

Aşağıdaki koşullardan herhangi biri doğruysa, altsistem güvenlik anahtarını kapatır:

- RACF etkin değil ya da kurulu değil.
- MQADMIN ya da MXADMIN sınıfı tanımlı değil (bu sınıflar, sınıf tanımlayıcı tablosunda (CDT) bulunduğundan RACF için her zaman tanımlanır).
- MQADMIN ya da MXADMIN sınıfı etkinleştirilmedi.

Hem RACF hem de MQADMIN ya da MXADMIN sınıfı etkinse, IBM MQ anahtar profillerinden herhangi birinin tanımlanıp tanımlanmadığını görmek için MQADMIN ya da MXADMIN sınıfını denetler. Önce "Altsistem güvenliğini denetlemek için profiller" sayfa 186içinde açıklanan profilleri denetler. Altsistem güvenliği gerekmiyorsa, IBM MQ iç altsistem güvenlik anahtarını kapatır ve başka denetim gerçekleştirmez.

Profiller, ilgili IBM MQ anahtarının açık mı, yoksa kapalı mı olduğunu belirler.

- Anahtar kapalıysa, bu güvenlik tipi devre dışı bırakılır.
- Herhangi bir IBM MQ anahtarı açık durumdaysa, IBM MQ , IBM MQ anahtarına karşılık gelen güvenlik tipiyle ilişkili RACF sınıfının durumunu denetler. Sınıf kurulu değilse ya da etkin değilse, IBM MQ anahtarı kapalıdır. Örneğin, MQPROC ya da MXPROC sınıfı etkinleştirilmediyse, süreç güvenliği denetimleri gerçekleştirilmez. Etkin olmayan sınıf, bu RACF veritabanını kullanan her kuyruk yöneticisi ve kuyruk paylaşım grubu için NO.PROCESS.CHECKS tanıtımının tanımlanmasıyla eşdeğerdir.

z/OS *Anahtarlar nasıl çalışır*

Bir güvenlik anahtarını devre dışı bırakmak için bir NO.* tanımlayın Profili değiştirin. Bir NO.* değerini geçersiz kılabilirsiniz bir YES.* tanımlayarak kuyruk paylaşım grubu düzeyinde ayarlanan profil bir kuyruk yöneticisine ilişkin tanıtım.

Bir güvenlik anahtarını devre dışı bırakmanız için bir NO.* tanımlamanız gerekir. Profili değiştirin. Bir NO.* varlığı profil, belirli bir kuyruk yöneticisinde bir kuyruk paylaşım grubu düzeyi ayarını geçersiz kılmayı seçmediğiniz sürece, o kaynak tipi için güvenlik denetimlerinin **gerçekleştirilmediği** anlamına gelir. Bu, "Kuyruk paylaşımı grup düzeyi ayarlarının geçersiz kılınması" sayfa 186içinde açıklanmıştır.

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesi değilse, herhangi bir kuyruk paylaşım grubu düzeyi tanıtımı ya da geçersiz kılma tanıtımı tanımlamanız gerekmez. Ancak, kuyruk yöneticisi daha sonraki bir tarihte bir kuyruk paylaşım grubuna katılırsa, bu tanıtımları tanımlamayı unutmayın.

Her NO.* IBM MQ tarafından saptanan anahtar profili, bu kaynak tipine ilişkin denetimi kapatır. Kuyruk yöneticisi başlatılırken anahtar tanıtımları etkinleştirilir. Etkilenen kuyruk yöneticileri çalışırken anahtar profillerini değiştirirseniz, IBM MQ ' un IBM MQ REFRESH SECURITY komutunu vererek değişiklikleri tanımasını sağlayabilirsiniz.

Anahtar profilleri her zaman MQADMIN ya da MXADMIN sınıfında tanımlanmalıdır. Bunları GMQADMIN ya da GMXADMIN sınıfında tanımlamayın. Tablolar Altsistem düzeyi güvenlik için anahtar profilleri ve Kaynak denetimi için anahtar profilleri geçerli anahtar profillerini ve bunların denetim güvenlik tipini gösterir.

Kuyruk paylaşımı grup düzeyi ayarlarının geçersiz kılınması

Bu grubun üyesi olan belirli bir kuyruk yöneticisi için kuyruk paylaşımı grubu düzeyinde güvenlik ayarlarını geçersiz kılabilirsiniz. Gruptaki diğer kuyruk yöneticilerine uygulanmayan tek bir kuyruk yöneticisi üzerinde kuyruk yöneticisi denetimleri gerçekleştirmek istiyorsanız, (qmgr-name.YES. *) Profilleri değiştirin.

Tersine, bir kuyruk paylaşım grubu içindeki belirli bir kuyruk yöneticisi üzerinde belirli bir denetim gerçekleştirmek istemiyorsanız, bir (qmgr-name.NO. *) tanımlayın Kuyruk yöneticisindeki belirli bir kaynak tipine ilişkin tanıtım ve kuyruk paylaşım grubu için tanıtım tanımlamayın. (IBM MQ , kuyruk paylaşım grubu düzeyi tanıtımını yalnızca kuyruk yöneticisi düzeyinde bir tanıtım bulamazsa denetler.)

z/OS *Altsistem güvenliğini denetlemek için profiller*

IBM MQ , altsistem, kuyruk yöneticisi ve kuyruk paylaşım grubu için altsistem güvenlik denetimlerinin gerekli olup olmadığını denetler.

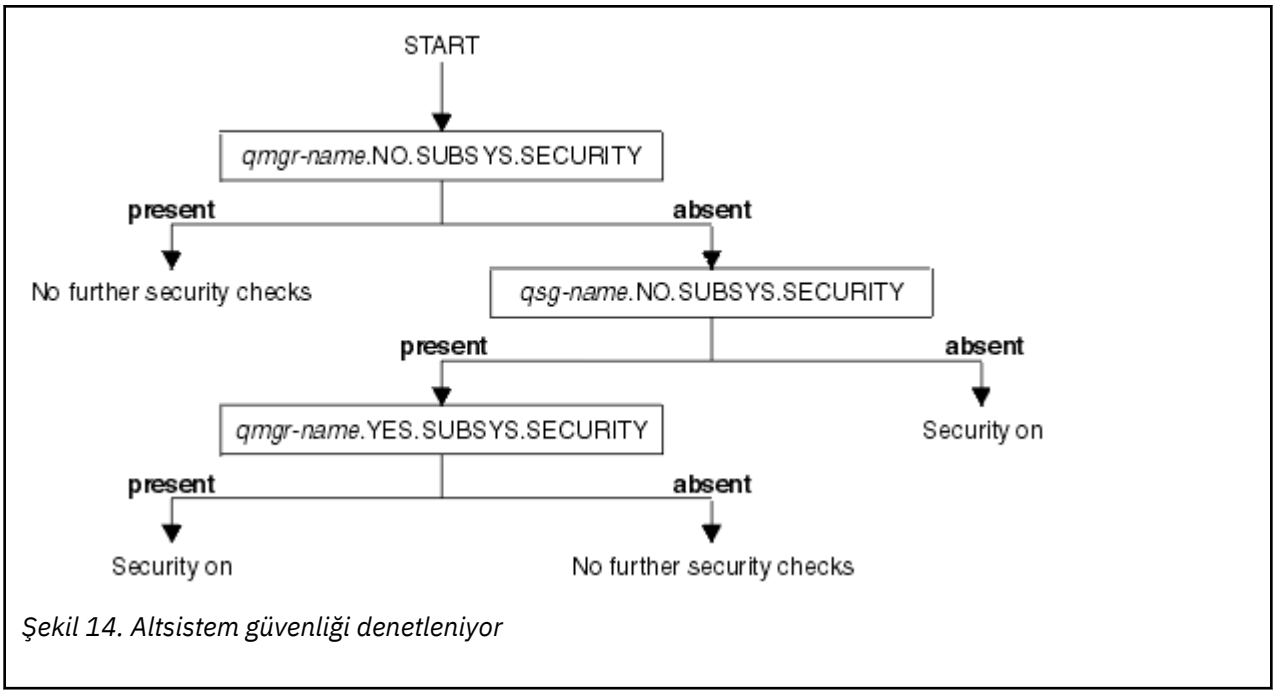
IBM MQ tarafından yapılan ilk güvenlik denetimi, tüm IBM MQ altsistemi için güvenlik denetimlerinin gerekip gerekmediğini belirlemek için kullanılır. Altsistem güvenliğini istemediğinizi belirlerseniz, başka denetim yapılmaz.

Altsistem güvenliğinin gerekli olup olmadığını belirlemek için aşağıdaki anahtar profilleri denetlenir. Şekil 14 sayfa 187 , denetlendikleri sırayı gösterir.

Çizelge 24. Altsistem düzeyi güvenlik için anahtar profilleri

Profil adını değiştir	**Denetlenen kaynağın ya da denetimin tipi**
qmgr-name.NO.SUBSYS.SECURITY	Bu kuyruk yöneticisine ilişkin altsistem güvenliği
qsg-name.NO.SUBSYS.SECURITY	Bu kuyruk paylaşım grubuna ilişkin altsistem güvenliği
qmgr-name.YES.SUBSYS.SECURITY	Bu kuyruk yöneticisi için altsistem güvenliği geçersiz kılma değeri

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesi değilse, IBM MQ yalnızca qmgr-name.NO.SUBSYS.SECURITY anahtar tanıtımını denetler.



Şekil 14. Altsistem güvenliği denetleniyor

z/OS *Kuyruk paylaşım grubunu ya da kuyruk yöneticisi düzeyinde güvenliği denetlemek için profiller*

Altsistem güvenlik denetimi gerekiyorsa, IBM MQ kuyruk paylaşım grubu ya da kuyruk yöneticisi düzeyinde güvenlik denetiminin gerekli olup olmadığını denetler.

IBM MQ , güvenlik denetiminin gerekli olduğunu belirlediğinde, kuyruk paylaşım grubunda mı, kuyruk yöneticisi düzeyinde mi, yoksa her ikisinde mi denetiminin gerekli olduğunu belirler. Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesi değilse, bu denetimler gerçekleştirilmez.

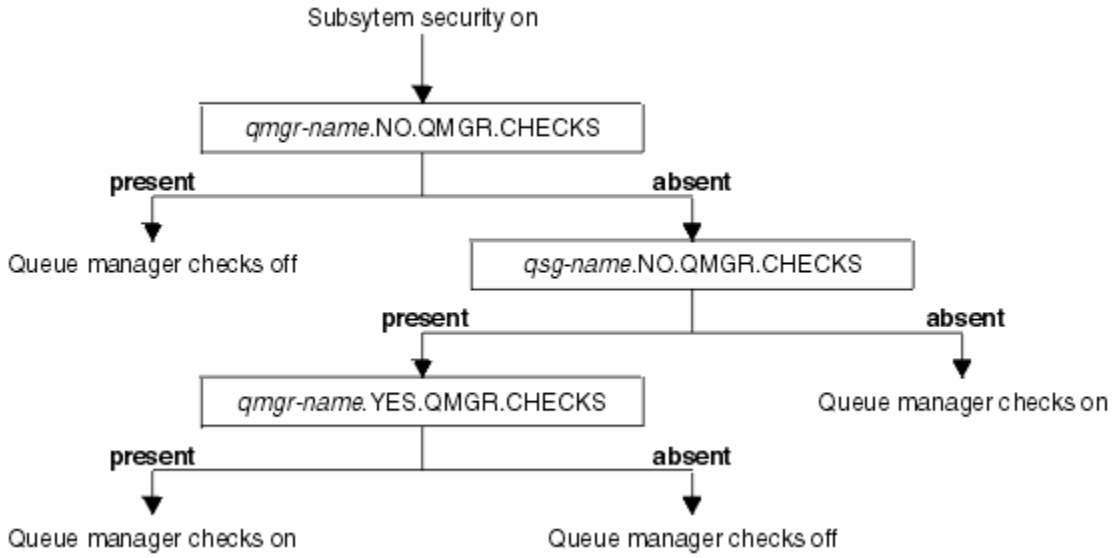
Gerekli düzeyi belirlemek için aşağıdaki anahtar profilleri denetlenir. Şekil 15 sayfa 188 ve Şekil 16 sayfa 189 , denetlendikleri sırayı gösterir.

Çizelge 25. Kuyruk paylaşım grubu ya da kuyruk yöneticisi düzeyi güvenlik için profilleri değiştir

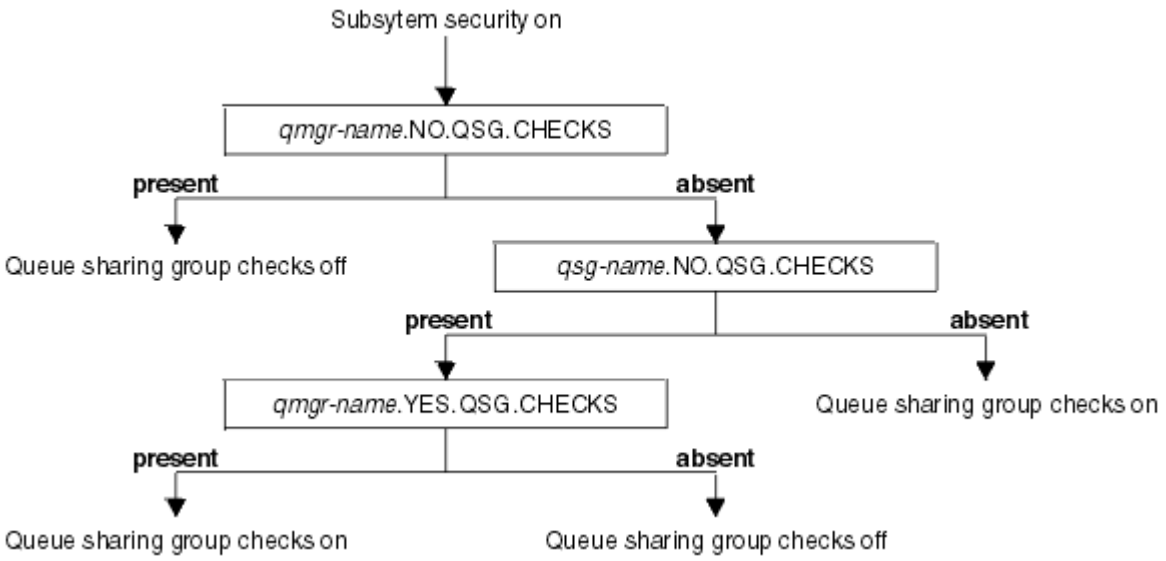
Profil adını değiştir	**Denetlenen kaynağın ya da denetimin tipi**
qmgr-name.NO.QMGR.CHECKS	Bu kuyruk yöneticisi için kuyruk yöneticisi düzeyi denetimi yok

Çizelge 25. Kuyruk paylaşım grubu ya da kuyruk yöneticisi düzeyi güvenlik için profilleri değiştir (devamı var)	
Profil adını değiştir	**Denetlenen kaynağın ya da denetimin tipi**
qsg-name.NO.QMGR.CHECKS	Bu kuyruk paylaşım grubu için kuyruk yöneticisi düzeyi denetimi yok
qmgr-name.YES.QMGR.CHECKS	Kuyruk yöneticisi düzeyi denetimleri bu kuyruk yöneticisi için geçersiz kılma değeri
qmgr-name.NO.QSG.CHECKS	Bu kuyruk yöneticisi için kuyruk paylaşımı grup düzeyi denetimi yok
qsg-name.NO.QSG.CHECKS	Bu kuyruk paylaşım grubu için kuyruk paylaşım grubu düzeyi denetimi yok
qmgr-name.YES.QSG.CHECKS	Kuyruk paylaşım grubu düzeyi denetimleri bu kuyruk yöneticisi için geçersiz kılma değeri

Altsistem güvenliği etkinse, hem kuyruk paylaşım grubunu hem de kuyruk yöneticisi düzeyinde güvenliği kapamazsınız. Bunu yapmaya çalışırsanız, IBM MQ her iki düzeyde de güvenlik denetimini ayarlar.



Şekil 15. Kuyruk yöneticisi düzeyinde güvenlik denetleniyor



Şekil 16. Kuyruk paylaşımı grup düzeyi güvenliği denetleniyor

z/OS *Geçerli güvenlik anahtarı birleşimleri*

Yalnızca belirli anahtar birleşimleri geçerlidir. Geçerli olmayan bir anahtar ayarları birleşimi kullanırsanız, CSQH026I iletisi yayınlanır ve güvenlik denetimi hem kuyruk paylaşım grubu hem de kuyruk yöneticisi düzeyinde ayarlanır.

Çizelge 26 sayfa 189, Çizelge 27 sayfa 189, Çizelge 28 sayfa 190ve Çizelge 29 sayfa 190 , her bir güvenlik düzeyi tipi için geçerli olan anahtar ayarları birleşimlerini gösterir.

Çizelge 26. Kuyruk yöneticisi düzeyinde güvenlik için geçerli güvenlik anahtarı birleşimleri

Birleşikler
qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS
qsg-name.NO.QSG.CHECKS qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS

Çizelge 27. Kuyruk paylaşım grubu düzeyinde güvenlik için geçerli güvenlik anahtarı birleşimleri

Birleşikler
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

Çizelge 27. Kuyruk paylaşım grubu düzeyinde güvenlik için geçerli güvenlik anahtarı birleşimleri (devamı var)
Birleşikler
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS qmgr-name.YES.QSG.CHECKS

Çizelge 28. Kuyruk yöneticisi ve kuyruk paylaşımı grup düzeyi güvenliği için geçerli güvenlik anahtarı birleşimleri
Birleşikler
qsg-name.NO.QMGR.CHECKS qmgr-name.YES.QMGR.CHECKS QSG.* yok tanımlı profiller
QMGR.* yok tanımlı profiller 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
Her iki anahtar için de profil tanımlanmadı

*Çizelge 29. Her iki denetim düzeyi **on**arasında geçiş yapan diğer geçerli güvenlik anahtarı birleşimleri.*
Birleşikler
qmgr-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qmgr-name.NO.QMGR.CHECKS qsg-name.NO.QSG.CHECKS
qsg-name.NO.QMGR.CHECKS qmgr-name.NO.QSG.CHECKS

z/OS *Kaynak düzeyi denetimleri*

Kaynaklara erişimi denetlemek için bir dizi anahtar profili kullanılır. Bazıları, bir kuyruk yöneticisinde ya da bir kuyruk paylaşım grubunda gerçekleştirilen denetlemeyi durdurur. Bunlar, belirli kuyruk yöneticilerinin denetlenmesini sağlayan profiller tarafından geçersiz kılınabilir.

Çizelge 30 sayfa 191 , IBM MQ kaynaklarına erişimi denetlemek için kullanılan anahtar profillerini gösterir.

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun parçasıysa ve hem kuyruk yöneticisi hem de kuyruk paylaşım grubu güvenliğiniz etkinse, bir YES.* kullanabilirsiniz. Kuyruk paylaşım grubu düzeyi tanıtımlarını geçersiz kılmak ve belirli bir kuyruk yöneticisine ilişkin güvenliği özellikle açmak için tanıtımı değiştirin.

Bazı tanıtımlar hem kuyruk yöneticileri hem de kuyruk paylaşım grupları için geçerlidir. Bunlara *hlq* dizgisi eklenir ve kuyruk paylaşım grubunuz ya da kuyruk yöneticinizin adını (hangisi geçerliyse) değiştirmeniz gerekir. *qmgr-name* öneki ile gösterilen tanıtım adları kuyruk yöneticisi geçersiz kılma profilleridir; kuyruk yöneticinizin adını değiştirmeniz gerekir.

Çizelge 30. Kaynak denetimi için profilleri değiştir

Denetlenen kaynak denetiminin tipi	**Profil adını değiştir**	**Belirli bir kuyruk yöneticisine ilişkin tanıtımı geçersiz kıl**
Bağlantı güvenliği	hlq.NO.CONNECT.CHECKS	qmgr-name.YES.CONNECT.CHECKS
Kuyruk güvenliği	hlq.NO.QUEUE.CHECKS	qmgr-name.YES.QUEUE.CHECKS
Süreç güvenliği	hlq.NO.PROCESS.CHECKS	qmgr-name.YES.PROCESS.CHECKS
Namelist güvenliği	hlq.NO.NLIST.CHECKS	qmgr-name.YES.NLIST.CHECKS
Bağlam güvenliği	hlq.NO.CONTEXT.CHECKS	qmgr-name.YES.CONTEXT.CHECKS
Diğer kullanıcı güvenliği	hlq.NO.ALTERNATE.USER.CHECKS	qmgr-name.YES.ALTERNATE.USER.CHECKS
Komut güvenliği	hlq.NO.CMD.CHECKS	qmgr-name.YES.CMD.CHECKS
Komut kaynağı güvenliği	hlq.NO.CMD.RESC.CHECKS	qmgr-name.YES.CMD.RESC.CHECKS
Konu güvenliği	hlq.NO.TOPIC.CHECKS	qmgr-name.YES.TOPIC.CHECKS
Not: hlq.NO. * * gibi soysal anahtar tanıtımları IBM MQ tarafından yoksayılır		

Örneğin, QSG3 kuyruk paylaşım grubunun bir üyesi olan QM01kuyruk yöneticisinde işlem güvenliği denetimleri gerçekleştirmek istiyorsanız, ancak gruptaki diğer kuyruk yöneticilerinden herhangi birinde işlem güvenliği denetimleri gerçekleştirmek istemiyorsanız, aşağıdaki anahtar profillerini tanımlayın:

```
QSG3.NO.PROCESS.CHECKS
QM01.YES.PROCESS.CHECKS
```

Kuyruk paylaşım grubundaki tüm kuyruk yöneticilerine QM02dışında kuyruk güvenliği denetimlerinin gerçekleştirilmesini istiyorsanız, aşağıdaki anahtar tanıtımını tanımlayın:

```
QM02.NO.QUEUE.CHECKS
```

(Tanımlı tanıtım yoksa, denetimler otomatik olarak etkinleştirildiğinden, kuyruk paylaşım grubu için tanıtım tanımlanması gerekmez.)

z/OS *Anahtarların tanımlanmasına bir örnek*

Farklı IBM MQ altsistemlerinin farklı anahtar profilleri kullanılarak uygulanabilen farklı güvenlik gereksinimleri vardır.

Dört IBM MQ altsistemi tanımlandı:

- MQP1 (bir üretim sistemi)
- MQP2 (üretim sistemi)
- MQD1 (geliştirme sistemi)
- MQT1 (sınama sistemi)

Dört kuyruk yöneticisinin tümü QS01kuyruk paylaşım grubunun üyeleridir. Tüm IBM MQ RACF sınıfları tanımlandı ve etkinleştirildi.

Bu altsistemlerin farklı güvenlik gereksinimleri vardır:

- Üretim sistemlerinin her iki sistemde de kuyruk paylaşım grubu düzeyinde etkin olması için tam IBM MQ güvenlik denetimi gerekir.

  Bu, aşağıdaki profil belirtilerek yapılır:

  ```
  RDEFINE MQADMIN QS01.NO.QMGR.CHECKS
  ```

  Bu, kuyruk paylaşım grubundaki tüm kuyruk yöneticileri için kuyruk paylaşım grubu düzeyi denetimini ayarlar. Bu sistemlere ilişkin her şeyi denetlemek istediğiniz için, üretim kuyruğu yöneticileri için başka bir anahtar profili tanımlamanıza gerek yoktur.
- MQT1 test kuyruğu yöneticisi de tam güvenlik denetimi gerektirir. Ancak, bunu daha sonra değiştirmek isteyebileceğiniz için, kuyruk yöneticisi düzeyinde güvenlik tanımlanabilir; böylece, kuyruk paylaşım grubunun diğer üyelerini etkilemeden bu kuyruk yöneticisine ilişkin güvenlik ayarlarını değiştirebilirsiniz.

  Bu, MQT1 için NO.QSG.CHECKS profili aşağıdaki şekilde tanımlanarak yapılır:

  ```
  RDEFINE MQADMIN MQT1.NO.QSG.CHECKS
  ```
- Geliştirme kuyruğu yöneticisi MQD1 , kuyruk paylaşım grubunun geri kalanından farklı güvenlik gereksinimlerine sahip. Yalnızca bağlantı ve kuyruk güvenliğinin etkin olmasını gerektirir.

  Bu, bu kuyruk yöneticisi için bir `MQD1.YES.QMGR.CHECKS` tanıtımı tanımlanarak ve daha sonra, denetlenmesi gerekmeyen kaynaklar için güvenlik denetimini kapatabilmek üzere aşağıdaki tanıtımlar tanımlanarak gerçekleştirilir:

  ```
  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
  ```

Kuyruk yöneticisi etkinken, DISPLAY SECURITY MQSC komutunu vererek yürürlükteki güvenlik ayarlarını görüntüleyebilirsiniz.

Kuyruk yöneticisi çalışırken, MQADMIN sınıfında uygun anahtar tanıtımını tanımlayarak ya da silerek de anahtar ayarlarını değiştirebilirsiniz. Anahtar ayarlarında değişiklik yapmak için, MQADMIN sınıfına ilişkin REFRESH SECURITY komutunu vermeniz gerekir.

DISPLAY SECURITY ve REFRESH SECURITY komutlarının kullanılmasıyla ilgili daha fazla ayrıntı için bkz. “z/OS üzerinde kuyruk yöneticisi güvenliği yenileniyor” sayfa 246 .

z/OS IBM MQ kaynaklarına erişimi denetlemek için kullanılan profiller

Tanımlanmış olabilecek anahtar profillerine ek olarak IBM MQ kaynaklarına erişimi denetlemek için RACF profillerini tanımlamanız gerekir. Bu konu derlemi, IBM MQ kaynağının farklı tiplerine ilişkin RACF tanıtımları hakkında bilgi içerir.

Belirli bir güvenlik denetimi için tanımlanmış bir kaynak profiliniz yoksa ve bir kullanıcı bu denetimi yapmayı içerecek bir istek gönderirse, IBM MQ erişimi reddeder. Devre dışı bırakılan güvenlik anahtarlarıyla ilgili güvenlik tipleri için profiller tanımlamanıza gerek yoktur.

z/OS *Bağlantı güvenliğine ilişkin tanıtımlar*

Bağlantı güvenliği etkinse, MQCONN sınıfında tanıtımlar tanımlamanız ve bu tanıtımlara gereken grupların ya da kullanıcı kimliklerinin IBM MQ' e bağlanabilmeleri için bu tanıtımlara erişmelerine izin vermeniz gerekir.

Bir bağlantının kurulabilmesi için, kullanıcılara uygun tanıtım için RACF OKUMA erişimi vermeniz gerekir. (Kuyruk yöneticisi düzeyinde bir tanıtım yoksa ve kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesiyse, güvenlik bunu yapacak şekilde ayarlandıysa, kuyruk paylaşım grubu düzeyi tanıtımlarıyla ilgili denetimler yapılabilir.)

Kuyruk yöneticisi adıyla nitelenmiş bir bağlantı tanıtımı, belirli bir kuyruk yöneticisine erişimi denetler ve bu tanıtıma erişim verilen kullanıcılar o kuyruk yöneticisine bağlanabilir. Kuyruk paylaşım grubu adıyla nitelenmiş bir bağlantı tanıtımı, o bağlantı tipine ilişkin kuyruk paylaşım grubu içindeki tüm kuyruk yöneticilerine erişimi denetler. Örneğin, `QS01.BATCH` erişimi olan bir kullanıcı, kuyruk yöneticisi düzeyi tanıtımı tanımlanmamış QS01 kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisine toplu iş bağlantısı kullanabilir.

Not:

1. Farklı güvenlik istekleri için denetlenen kullanıcı kimlikleri hakkında bilgi için bkz. "z/OS üzerinde güvenlik denetimi için kullanıcı kimlikleri" sayfa 234.
2. Bağlantı sırasında kaynak düzeyi güvenlik (RESLEVEL) denetimleri de yapılır. Ayrıntılar için bkz. "RESLEVEL güvenlik profili" sayfa 229.

IBM MQ güvenliği, aşağıdaki farklı bağlantı tiplerini tanır:

- Toplu iş (ve toplu iş tipi) bağlantıları şunlardır:
  - z/OS toplu işler
  - TSO uygulamaları
  - z/OS UNIX System Services oturum açma
  - Db2 saklı yordamlar
- CICS bağlantılar
- Denetim ve uygulama işleme bölgelerinden IMS bağlantıları
- IBM MQ kanal başlatıcısı

z/OS *Toplu bağlantılar için bağlantı güvenliği tanıtımları*

Toplu iş tipi bağlantılarının denetlenmesine ilişkin tanıtımlar, kuyruk yöneticisi ya da kuyruk paylaşım grubu adından ve ardından *BATCH*sözcüğünden oluşur. Bağlantı tanıtımı için, bağlanan adres alanıyla ilişkili kullanıcı kimliğine OKUMA erişimi verin.

Toplu iş ve toplu iş tipi bağlantıları denetlemeye ilişkin profiller şu formu alır:

```
hlq.BATCH
```

Burada `hlq` , `qmgr-name` (kuyruk yöneticisi adı) ya da `qsg-name` (kuyruk paylaşım grubu adı) olabilir. Hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ , kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar. Tanıtımlardan birini bulamazsa, bağlantı isteği başarısız olur.

Toplu iş ya da toplu iş tipi bağlantı istekleri için, bağlanan adres alanıyla ilişkili kullanıcı kimliğinin bağlantı tanıtımına erişmesine izin vermeniz gerekir. Örneğin, aşağıdaki RACF komutu CONNTQM1 grubundaki kullanıcıların TQM1; bu kullanıcı kimliklerinin herhangi bir toplu iş ya da toplu iş tipi bağlantıyı kullanmalarına izin verilir.

```
RDEFINE MQCONN TQM1.BATCH UACC(NONE)
PERMIT TQM1.BATCH CLASS(MQCONN) ID(CONNTQM1) ACCESS(READ)
```

z/OS ***CHCKLOCL*** *' un yerel olarak bağlı uygulamalarda kullanılması*

CHCKLOCL yalnızca BATCH bağlantıları aracılığıyla yapılan bağlantılar için geçerlidir ve CICS ya da IMStarafından yapılan bağlantılar için geçerli değildir. Kanal başlatıcı aracılığıyla yapılan bağlantılar **CHCKCLNT**tarafından denetlenir.

Genel Bakış

z/OS kuyruk yöneticinizi, yerel olarak bağlı uygulamalarınızın bazıları için, ancak tümü için değil, kullanıcı kimliği ve parola denetimini zorunlu yapacak şekilde yapılandırmak istiyorsanız, bazı ek yapılandırma gerçekleştirmeniz gerekir.

Bunun nedeni, **CHCKLOCL** (*REQUIRED*) yapılandırıldıktan sonra, MQCONN API çağrısını kullanan kalıt toplu iş uygulamalarının artık kuyruk yöneticisine bağlanamamasıdır.

Yalnızca z/OS için, bir adres alanının bağlantı güvenliğine dayalı daha ayrıntılı bir düzenek, genel CHCKLOCL (REQUIRED) konfigürasyonunu özel olarak tanımlanmış kullanıcı kimlikleri için CHCKLOCL (İSTEğE BAğLı) konfigürasyonuna indirgemek üzere kullanılabilir. Kullanılan mekanizma, bir örnekle birlikte aşağıdaki metinde açıklanmıştır.

CHCKLOCL (*REQUIRED*) üzerinde yalnızca HERKES değerinden daha fazla ayrıntı düzeyine izin vermek için, **CHCKLOCL** öğesini MQCONN sınıfındaki `hlq.batch` bağlantı tanıtımlarına bağlanan adres alanıyla ilişkili kullanıcı kimliğinin erişim düzeyini değiştirdiğiniz gibi değiştirmeniz gerekir.

Adres alanı kullanıcı kimliğinin yalnızca READ erişimi varsa (bu, bağlanabilmeniz için gereken en alt sınırdır), **CHCKLOCL** yapılandırması yazıldığı gibi uygulanır.

Adres alanı kullanıcı kimliğinin UPDATE erişimi (ya da üstü) varsa, **CHCKLOCL** yapılandırması *İSTEğE BAğLı* kipinde çalışır. Yani, bir kullanıcı kimliği ve parola girmeniz gerekmez; girdiyseniz, kullanıcı kimliği ve parola geçerli bir çift olmalıdır.

z/OS kuyruk yöneticiniz için bağlantı güvenliği önceden yapılandırıldı

z/OS kuyruk yöneticiniz için bağlantı güvenliği yapılandırıldıysa ve **CHCKLOCL** (*GEREKLİ*) olanağının yerel olarak bağlı WAS uygulamalarına uygulanmasını istiyorsanız, başka uygulama yoksa, aşağıdaki adımları izleyin:

1. Yapılandırmanız olarak **CHCKLOCL** (*İSTEğE BAğLı*) ile başlayın. Bu, sağlanan kullanıcı kimliği ve parolaların geçerliliği denetlendiği, ancak zorunlu olmadığı anlamına gelir.
2. Komutu vererek bağlantı güvenliği tanıtımlarına erişimi olan tüm kullanıcıları listeleyin:

   ```
   RLIST MQCONN MQ23.BATCH AUTHUSER
   ```

   Bu komut görüntülenir; örneğin:

   ```
   CLASS    NAME
   -----    ----
   MQCONN   MQ23.BATCH

   USER     ACCESS  ACCESS COUNT
   ----     ------  ------ -----
   JOHNDOE  READ    000009
   JDOE1    READ    000003
   WASUSER  READ    000000
   ```

3. Okuma erişimine sahip olarak listelenen her bir kullanıcı kimliği için, erişimi şu şekilde değiştirin:

   ```
   UPDATE:- PERMIT MQ23.BATCH CLASS(MQCONN) ID(JOHNDOE) ACCESS(UPDATE)
   ```

4. IBM MQ yapılandırmasını **CHCKLOCL** (*REQUIRED*) olarak güncelleyin.

   `MQ23.`BATCH için UPDATE erişimi ve geçerli ayar birleşimi, **CHCKLOCL** (*İSTEğE BAğLı*) kullanmakta olduğunuz anlamına gelir.
5. Şimdi, **CHCKLOCL** (*REQUIRED*) davranışını belirli bir kullanıcı kimliğine (örneğin, WASUSER) uygulayın; böylece o bölgeden gelen tüm bağlantılar bir kullanıcı kimliği ve parola sağlamalıdır.

   Daha önce yaptığınız değişikliği tersine çevirmek için şu komutu verin:

```
PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
```

z/OS kuyruk yöneticiniz için bağlantı güvenliği yapılandırılmadı

Bu durumda şunları yapmanız gerekir:

1. Şu komutu vererek MQCONN sınıfında `hlq.`BATCH için bağlantı tanıtımları yaratın:

   ```
   RDEFINE MQCONN MQ23.BATCH UACC(NONE)
   ```

2. Kuyruk yöneticisiyle toplu iş bağlantıları yaratan tüm kullanıcı kimliklerini yetkilendirerek, bu tanıtıma UPDATE (güncelleme) erişimine sahip olmanızı sağlar. Bu işlem, bağlantı sırasında kullanıcı kimliği ve parola için **CHCKLOCL** (*REQUIRED*) gereksinmesini atlar.

   Bunu yapmak için şu komutu verin:

   ```
   PERMIT MQ23.BATCH CLASS(MQCONN)ID(JOHNDOE) ACCESS(UPDATE)
   ```

   Bunlar kullanıcı kimliklerini içerir:

   a. CSQUTIL, ISPF panoları ve diğer yerel olarak bağlı araçlar için kullanılır.

   b. Kuyruk yöneticisine yönelik toplu iş benzeri bağlantılarla ilişkilendirilir. Örneğin, Advanced Message Security, IBM Integration Bus, Db2 saklanmış yordamlar, z/OS UNIX System Services ve TSO kullanıcıları ve Java uygulamaları

3. Şu komutu girerek kuyruk yöneticisine ilişkin anahtar profilini silin:

   ```
   hlq.NO.CONNECT.CHECKS
   ```

4. Şimdi, **CHCKLOCL** (*REQUIRED*) davranışını belirli bir kullanıcı kimliğine (örneğin, WASUSER) uygulayın; böylece o bölgeden gelen tüm bağlantılar bir kullanıcı kimliği ve parola sağlamalıdır.

   Daha önce yaptığınız değişikliği tersine çevirmek için şu komutu verin:

   ```
   PERMIT MQ23.BATCH CLASS(MQCONN) ID(WASUSER) ACCESS(READ)
   ```

z/OS *CICS bağlantıları için bağlantı güvenliği tanıtımları*

CICS bağlantılarını denetleyen profiller, kuyruk yöneticisi ya da kuyruk paylaşım grubu adından ve ardından *CICS* sözcüğünden oluşur. CICS adres alanıyla ilişkili kullanıcı kimliğine bağlantı tanıtımına OKUMA erişimi verin.

CICS ' den gelen bağlantıları denetlemeye ilişkin profiller şu formu alır:

```
hlq.CICS
```

Burada `hlq` , `qmgr-name` (kuyruk yöneticisi adı) ya da `qsg-name` (kuyruk paylaşım grubu adı) olabilir. Hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ , kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar. Herhangi bir tanıtımı bulamazsa, bağlantı isteği başarısız olur

CICSile yapılan bağlantı istekleri için, bağlantı tanıtımına yalnızca CICS adres alanı kullanıcı kimliği erişimine izin vermeniz gerekir.

Örneğin, aşağıdaki RACF komutları CICS adres alanı kullanıcı kimliği KCBCICS ' in TQM1: kuyruk yöneticisine bağlanmasını sağlar:

```
RDEFINE MQCONN TQM1.CICS UACC(NONE)
PERMIT TQM1.CICS CLASS(MQCONN) ID(KCBCICS) ACCESS(READ)
```

z/OS *IMS bağlantıları için bağlantı güvenliği tanıtımları*

IMS bağlantılarını denetleyen profiller, kuyruk yöneticisi ya da kuyruk paylaşım grubu adından ve ardından *IMS* sözcüğünden oluşur. IMS denetim ve bağımlı bölge kullanıcı kimliklerine bağlantı tanıtımı için OKUMA erişimi verin.

IMS ' den gelen bağlantıları denetlemeye ilişkin profiller şu formu alır:

```
hlq.IMS
```

Burada `hlq` , `qmgr-name` (kuyruk yöneticisi adı) ya da `qsg-name` (kuyruk paylaşım grubu adı) olabilir. Hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ , kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar. Herhangi bir tanıtımı bulamazsa, bağlantı isteği başarısız olur

IMStarafından yapılan bağlantı istekleri için, IMS denetim ve bağımlı bölge kullanıcı kimliklerine ilişkin bağlantı tanıtımına erişime izin verin.

Örneğin, aşağıdaki RACF komutları aşağıdakileri sağlar:

- Kuyruk yöneticisine TQM1bağlanmak için kullanılan IMS bölgesi kullanıcı kimliği (IMSREG).
- BMP işlerini göndermek için BMPGRP grubundaki kullanıcılar.

```
RDEFINE MQCONN TQM1.IMS UACC(NONE)
PERMIT TQM1.IMS CLASS(MQCONN) ID(IMSREG,BMPGRP) ACCESS(READ)
```

z/OS *Kanal başlatıcısına ilişkin bağlantı güvenliği tanıtımları*

Kanal başlatıcısından gelen bağlantıları denetlemeye ilişkin profiller, kuyruk yöneticisi ya da kuyruk paylaşım grubu adından ve ardından *CHIN*sözcüğünden oluşur. Kanal başlatıcı tarafından başlatılan görev adresi alanı tarafından kullanılan kullanıcı kimliğine bağlantı tanıtımı için OKUMA erişimi verin.

Kanal başlatıcısından gelen bağlantıları denetlemeye ilişkin profiller şu formu alır:

```
hlq.CHIN
```

Burada `hlq` , `qmgr-name` (kuyruk yöneticisi adı) ya da `qsg-name` (kuyruk paylaşım grubu adı) olabilir. Hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ , kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar. Herhangi bir tanıtımı bulamazsa, bağlantı isteği başarısız olur

Kanal başlatıcısı tarafından yapılan bağlantı istekleri için, kanal başlatıcısı tarafından başlatılan görev adresi alanı tarafından kullanılan kullanıcı kimliğine ilişkin bağlantı tanıtımına erişim tanımlayın.

Örneğin, aşağıdaki RACF komutları, DQCTRL kullanıcı kimliğiyle çalışan kanal başlatıcı adres alanının TQM1: kuyruk yöneticisine bağlanmasına izin verir:

```
RDEFINE MQCONN TQM1.CHIN UACC(NONE)
PERMIT TQM1.CHIN CLASS(MQCONN) ID(DQCTRL) ACCESS(READ)
```

z/OS *Kuyruk güvenliğine ilişkin profiller*

Kuyruk güvenliği etkinse, uygun sınıflarda tanıtımlar tanımlamanız ve bu tanıtımlara gereken grupların ya da kullanıcı kimliklerinin erişmesine izin vermeniz gerekir. Kuyruk güvenliği tanıtımları, kuyruk yöneticisi ya da kuyruk paylaşım grubunun ve açılacak kuyruğun adını taşır.

Kuyruk güvenliği etkinse, aşağıdakileri yapmanız gerekir:

- Büyük harfli tanıtımlar kullanılıyorsa, tanıtımları **MQQUEUE** ya da **GMQQUEUE** sınıflarında tanımlayın.
- Büyük ve küçük harf karışık tanıtımlar kullanılıyorsa, **MXQUEUE** ya da **GMXQUEUE** sınıflarındaki tanıtımları tanımlayın.
- Gerekli grupların ya da kullanıcı kimliklerinin, kuyrukları kullanan IBM MQ API isteklerini yayınlayabilmesi için bu profillere erişmesine izin verin.

Kuyruk güvenliğine ilişkin profiller şu formu alır:

```
hlq.queuename
```

Burada `hlq` , `qmgr-name` (kuyruk yöneticisi adı) ya da `qsg-name` (kuyruk paylaşım grubu adı) olabilir ve `queuename` , `MQOPEN` ya da `MQPUT1` çağrısındaki nesne tanımlayıcısında belirtildiği şekilde, açılmakta olan kuyruğun adıdır.

Kuyruk yöneticisi adı öneki eklenen bir tanıtım, o kuyruk yöneticisindeki tek bir kuyruğa erişimi denetler. Kuyruk paylaşım grubu adı öneki eklenen bir tanıtım, kuyruk paylaşım grubu içindeki tüm kuyruk yöneticilerine bir ya da daha fazla kuyruk adı içeren bir ya da daha fazla kuyruğa erişimi ya da grup içindeki herhangi bir kuyruk yöneticisi tarafından paylaşılan bir kuyruğa erişimi denetler. Bu erişim, söz konusu kuyruk yöneticisinde o kuyruk için bir kuyruk yöneticisi düzeyi tanıtımı tanımlanarak, tek bir kuyruk yöneticisinde geçersiz kılınabilir.

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesiyse ve hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ önce kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar.

Paylaşılan kuyruklar kullanıyorsanız, kuyruk paylaşım grubu düzeyinde güvenlik kullanmanız önerilir.

Kuyruk adı bir diğer adla ya da model kuyruğuyla aynı olduğunda kuyruk güvenliğinin nasıl çalıştığına ilişkin ayrıntılar için bkz. “Diğer ad kuyruklarına ilişkin dikkat edilecek noktalar” sayfa 199 ve “Model kuyruklarıyla ilgili dikkat edilecek noktalar” sayfa 200 .

Bir kuyruğu açmak için gereken RACF erişimi, belirtilen `MQOPEN` ya da `MQPUT1` seçeneklerine bağlıdır. Birden çok `MQOO_ *` ve `MQPMO_ *` seçeneği kodlanmışsa, kuyruk güvenliği denetimi gereken en yüksek RACF yetkisi için gerçekleştirilir.

Çizelge 31. MQOPEN ya da MQPUT1 çağrılarını kullanarak kuyruk güvenliği için erişim düzeyleri

MQOPEN ya da MQPUT1 seçeneği	**RACF hlq.queuename için gereken erişim düzeyi**
MQOO_GÖZ AT	READ
MQOO_INQUIRE	READ
MQOO_BIND_ *	GÜNCELLE
MQOO_INPUT_ *	GÜNCELLE
MQOO_OUTPUT ya da MQPUT1	GÜNCELLE

Çizelge 31. MQOPEN ya da MQPUT1 çağrılarını kullanarak kuyruk güvenliği için erişim düzeyleri (devamı var)

MQOPEN ya da MQPUT1 seçeneği	**RACF hlq.queuename için gereken erişim düzeyi**
MQOO_PASS_ALL_CONTEXT MQPMO_PASS_ALL_CONTEXT	GÜNCELLE
MQOO_PASS_IDENTITY_CONTEXT MQPMO_PASS_IDENTITY_CONTEXT	GÜNCELLE
MQOO_SAVE_ALL_CONTEXT	GÜNCELLE
MQOO_SET_IDENTITY_CONTEXT MQPMO_SET_IDENTITY_CONTEXT	GÜNCELLE
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	GÜNCELLE
MQOO_SET	Çeviri

Örneğin, IBM MQ kuyruk yöneticisi QM77'de, RACF PAYGRP grubundaki tüm kullanıcı kimliklerine, adları' PAY ' ile başlayan tüm kuyruklardan ileti alma ya da tüm kuyruklara ileti gönderme erişimi verilir. Bunu şu RACF komutlarını kullanarak yapabilirsiniz:

```
RDEFINE MQQUEUE QM77.PAY.** UACC(NONE)
PERMIT QM77.PAY.** CLASS(MQQUEUE) ID(PAYGRP) ACCESS(UPDATE)
```

Ayrıca, PAYGRP grubundaki tüm kullanıcı kimliklerinin, PAY adlandırma kuralına uymayan iletileri kuyruklara koymak için erişimi olmalıdır. Örneğin:

```
REQUEST_QUEUE_FOR_PAYROLL
SALARY.INCREASE.SERVER
REPLIES.FROM.SALARY.MODEL
```

GMQQUEUE sınıfında bu kuyruklar için profiller tanımlayarak ve bu sınıfa aşağıdaki gibi erişim vererek bunu yapabilirsiniz:

```
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)
```

Not:

1. Bir uygulamanın bir kuyruk güvenliği tanıtımına ilişkin RACF erişim düzeyi değiştirilirse, değişiklikler yalnızca o kuyruk için elde edilen yeni nesne tanıtıcıları (yeni MQOPEN ' ler) için geçerli olur. Değişiklik sırasında var olan bu tanıtıcıları, kuyruğa var olan erişimlerini korur. Bir uygulamanın, varolan erişim düzeyi yerine, kuyrukta değiştirilen erişim düzeyini kullanması gerekiyorsa, değişikliği gerektiren her nesne tanıtıcısı için kuyruğu kapatıp yeniden açması gerekir.
2. Örnekte, QM77 kuyruk yöneticisi adı bir kuyruk paylaşım grubunun adı da olabilir.

Belirlenen açma seçeneklerine ve etkin güvenlik tiplerine bağlı olarak, kuyruk açıldığında diğer güvenlik denetimi tipleri de oluşabilir. Ayrıca bkz. “Bağlam güvenliğine ilişkin profiller” sayfa 213 ve “Diğer kullanıcı güvenliği için profiller” sayfa 211. Kuyruk, bağlam ve diğer kullanıcı güvenliği etkin olduğunda

gereken açma seçeneklerini ve güvenlik yetkilendirmesini gösteren bir özet tablo için bkz. Çizelge 36 sayfa 204.

Yayınlama/abone olma özelliğini kullanıyorsanız, aşağıdakileri göz önünde bulundurmanız gerekir. Bir MQSUB isteği işlendiğinde, isteği yapan kullanıcı kimliğinin iletileri hedef IBM MQ kuyruğuna koymak için gerekli erişime ve IBM MQ konusuna abone olmak için gerekli erişime sahip olduğundan emin olmak için bir güvenlik denetimi gerçekleştirilir.

Çizelge 32. MQSUB çağrısı kullanılarak kuyruk güvenliği için erişim düzeyleri

MQSUB seçeneği	**RACF hlq.queuename için gereken erişim düzeyi**
MQSO_ALTER, MQSO_CREATE ve MQSO_RESUME	GÜNCELLE

Not:

1. `hlq.queuename` , yayınların hedef kuyruğudur. Bu bir yönetilen kuyruk olduğunda, yönetilen kuyruk ve yaratılan dinamik kuyruk için kullanılacak uygun model kuyruğuna erişmeniz gerekir.
2. Abonelikleri yapan kullanıcılar ile hedef kuyruktan yayınları alan kullanıcılar arasında ayrım yapmak istiyorsanız, MQSUB API çağrısında sağladığınız hedef kuyruk için bu tür bir teknik kullanabilirsiniz.

z/OS *Diğer ad kuyruklarına ilişkin dikkat edilecek noktalar*

Bir diğer ad kuyruğu için MQOPEN ya da MQPUT1 çağrısı yayınladığınızda, IBM MQ çağrıda nesne tanımlayıcısında (MQOD) belirtilen kuyruk adıyla ilgili bir kaynak denetimi yapar. Kullanıcının hedef kuyruk adına erişmesine izin verilip verilmediğini denetlemez.

Örneğin, PAYROLL.REQUEST , PAY.REQUEST. Kuyruk güvenliği etkinse, yalnızca PAYROLL.REQUEST. PAY.REQUEST.

z/OS *MQGET ve MQPUT isteklerini ayırt etmek için diğer ad kuyruklarının kullanılması*

Bir erişim düzeyinde kullanılabilir MQI çağrıları aralığı, bir kuyruğa erişimi yalnızca **MQPUT** çağrısına ya da yalnızca **MQGET** çağrısına izin verecek şekilde kısıtlamak istiyorsanız soruna neden olabilir. Bir kuyruk, bu kuyruğa çözülecek iki diğer ad tanımlanarak korunabilir: biri, uygulamaların kuyruktan ileti almasını, diğeri de uygulamaların kuyruğa ileti koymasını sağlar.

Aşağıdaki metin, kuyruklarınızı IBM MQiçin nasıl tanımlayabileceğinize ilişkin bir örnek sağlar:

```
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)
```

Aşağıdaki RACF tanımlamalarını da yapmalısınız:

```
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)
```

Daha sonra, hiçbir kullanıcının hlq.MUST_USE_ALIAS_TO_ACCESSkuyruğuna erişimi olmadığından emin olun ve diğer ada uygun kullanıcılara ya da gruplara erişim verin. Bunu aşağıdaki RACF komutlarını kullanarak yapabilirsiniz:

```
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)
```

Bu, GETGRP grubundaki kullanıcı kimliği GETUSER ve kullanıcı kimliklerinin yalnızca USE_THIS_ONE_FOR_GETS; diğer ad kuyruğu aracılığıyla MUST_USE_ALIAS_TO_ACCESS ile ilgili iletileri almasına ve PUTGRP grubundaki kullanıcı kimlikleri ve PUTUSER kullanıcı kimliklerinin iletileri yalnızca USE_THIS_ONE_FOR_PUTS kuyruğuna yerleştirmesine izin verildiği anlamına gelir.

Not:

1. Böyle bir teknik kullanmak istiyorsanız, uygulama geliştiricilerinizi bilgilendirmelisiniz, böylece programlarını uygun şekilde tasarlayabilirler.
2. Abonelikleri yapan kullanıcılar ile yayınları hedef kuyruktan alan kullanıcılar arasında ayrım yapmak istiyorsanız, MQSUB API isteğinde sağladığınız hedef kuyruk için bu tür bir teknik kullanabilirsiniz.

z/OS *Model kuyruklarıyla ilgili dikkat edilecek noktalar*

Bir model kuyruğunu açmak için hem model kuyruğunun kendisini hem de çözüldüğü dinamik kuyruğu açabilmeniz gerekir. IBM MQ yardımcı programları tarafından kullanılan dinamik kuyruklar da içinde olmak üzere, dinamik kuyruklar için soysal RACF tanıtımları tanımlayın.

Bir model kuyruğu açtığınızda, IBM MQ güvenliği iki kuyruk güvenliği denetimi yapar:

1. Model kuyruğuna erişme yetkiniz var mı?
2. Model kuyruğunun çözüldüğü dinamik kuyruğa erişme yetkiniz var mı?

Dinamik kuyruk adı bir sondaki yıldız işareti (*) içeriyorsa, bu *, benzersiz adla dinamik bir kuyruk yaratmak için IBM MQtarafından oluşturulan bir karakter dizgisiyle değiştirilir. Ancak, bu oluşturulan dizgi de içinde olmak üzere tüm ad yetki denetimi için kullanıldığından, bu kuyruklar için soysal tanıtımlar tanımlamanız gerekir.

Örneğin, bir MQOPEN çağrısı CREDIT.CHECK.REPLY.MODEL ve dinamik kuyruk adı CREDIT.REPLY.* kuyruk yöneticisinde (ya da kuyruk paylaşım grubunda) MQSP. (MQSP)

Bunu yapmak için, gerekli kuyruk tanıtımlarını tanımlamak üzere aşağıdaki RACF komutlarını vermeniz gerekir:

```
RDEFINE MQQUEUE MQSP.CREDIT.CHECK.REPLY.MODEL
RDEFINE MQQUEUE MQSP.CREDIT.REPLY.**
```

Kullanıcının bu tanıtımlara erişmesine izin vermek için ilgili RACF PERMIT komutlarını da vermeniz gerekir.

MQOPEN tarafından yaratılan tipik dinamik kuyruk adı CREDIT.REPLY.A346EF00367849A0. Son niteleyicinin kesin değeri öngörülemez; bu nedenle, bu tür kuyruk adları için soysal tanıtımlar kullanmalısınız.

Bazı IBM MQ yardımcı programları iletileri dinamik kuyruklara koyar. Aşağıdaki dinamik kuyruk adları için profiller tanımlamalı ve ilgili kullanıcı kimliklerine RACF UPDATE erişimi sağlamalısınız (doğru kullanıcı kimlikleri için bkz. “z/OS üzerinde güvenlik denetimi için kullanıcı kimlikleri” sayfa 234):

```
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)
```

Uygulama programlama kopyası üyelerinde varsayılan olarak kullanılan dinamik kuyruk adının kullanımını denetlemek için bir tanıtım tanımlamayı da düşünebilirsiniz. IBM MQtarafından sağlanan copybooks, CSQ. * olan varsayılan bir *DynamicQName*içeriyor. Bu, uygun bir RACF profili oluşturulmasını sağlar.

Not: Uygulama programcılarının dinamik kuyruk adı için tek bir * belirlemesine izin vermeyin. Bunu yaparsanız, bir hlq. * * tanımlamanız gerekir MQQUEUE sınıfındaki profili ve buna geniş kapsamlı erişim vermeniz gerekir. Bu, bu tanıtımın daha belirli bir RACF tanıtımına sahip olmayan dinamik olmayan diğer

kuyruklar için de kullanılabileceği anlamına gelir. Bu nedenle, kullanıcılarınız erişmelerini istemediğiniz kuyruklara erişim elde edebilir.

z/OS *Kalıcı dinamik kuyruklardaki seçenekleri kapat*

Bir uygulama başka bir uygulama tarafından yaratılan kalıcı bir dinamik kuyruğu açarsa ve daha sonra bu kuyruğu MQCLOSE seçeneğiyle silmeyi denerse, deneme yapıldığında bazı ek güvenlik denetimleri uygulanır.

Çizelge 33. Kalıcı dinamik kuyruklardaki kapatma seçenekleri için erişim düzeyleri

MQCLOSE seçeneği	**RACF hlq.queuename için gereken erişim düzeyi**
MQCO_DELETE	Çeviri
MQCO_DELETE_PURGE	Çeviri

z/OS *Güvenlik ve uzak kuyruklar*

Uzak kuyruğa bir ileti konduğunda, yerel kuyruk yöneticisi tarafından uygulanan kuyruk güvenliği, uzak kuyruğun açıldığında nasıl belirlendiğine bağlıdır.

Aşağıdaki kurallar uygulanır:

1. Uzak kuyruk IBM MQ DEFINE QREMOTE komutuyla yerel kuyruk yöneticisinde tanımlandıysa, denetlenen kuyruk uzak kuyruğun adıdır. Örneğin, MQS1 kuyruk yöneticisinde aşağıdaki gibi bir uzak kuyruk tanımlanmışsa:

   ```
   DEFINE QREMOTE(BANK7.CREDIT.REFERENCE)
          RNAME(CREDIT.SCORING.REQUEST)
          RQMNAME(BNK7)
          XMITQ(BANK1.TO.BANK7)
   ```

   Bu durumda, BANK7.CREDIT.REFERENCE , MQQUEUE sınıfında tanımlanmalıdır.
2. İsteğe ilişkin *ObjectQMgrAdı* yerel kuyruk yöneticisine çözülmezse, küme kuyruğu adına ilişkin denetimin yapıldığı bir küme kuyruğu dışında, çözülen (uzak) kuyruk yöneticisi adına ilişkin bir güvenlik denetimi gerçekleştirilir.

   Örneğin, iletim kuyruğu BANK1.TO.BANK7 , MQS1kuyruk yöneticisinde tanımlanır. Daha sonra, MQS1 ' ta BANK1.INTERBANK.TRANSFERS olarak *ObjectName* ve BANK1.TO.BANK7 *ObjectQMgrAd* olarak belirtilmesi için bir MQPUT1 isteği yayınlanır. Bu durumda, isteği gerçekleştiren kullanıcının BANK1.TO.BANK7.
3. Bir kuyruğa MQPUT isteğinde bulunursanız ve yerel kuyruk yöneticisinin diğer adı olarak *ObjectQMgrName* belirtirseniz, kuyruk yöneticisinin adı değil, yalnızca kuyruk adı güvenlik için denetlenir.

İleti uzak kuyruk yöneticisine ulaştığında, ek güvenlik işlemlerine tabi olabilir. Daha fazla bilgi için bkz “Uzak ileti sistemi güvenliği” sayfa 99.

z/OS *Gönderilmeyen ileti kuyruğu güvenliği*

Birçok kullanıcının ileti alabilmesi gerektiği, ancak iletileri almak için erişimin sıkı bir şekilde kısıtlanması gerektiği için, bu kuyrukta dikkate alınması gereken özel noktalar vardır. Bunu, ileti kuyruğuna ve diğer ad kuyruğuna farklı RACF yetkileri uygulayarak yapabilirsiniz.

Teslim edilmeyen iletiler, teslim edilmeyen ileti kuyruğu adı verilen özel bir kuyruğa konabilir. Bu kuyrukta sona erecek hassas verileriniz varsa, yetkisiz kullanıcıların bu verileri almasını istemediğiniz için bunun güvenlik etkilerini göz önünde bulundurmanız gerekir.

Aşağıdaki iletilerin her birinin, ileti gönderilmeyen iletiler kuyruğuna yerleştirilmesine izin verilmelidir:

- Uygulama programları.

- Kanal başlatıcı adres alanı ve MCA kullanıcı kimlikleri. (RESLEVEL tanıtımı yoksa ya da kanal kullanıcı kimliklerinin denetlenmesi için tanımlandıysa, kanal kullanıcı kimliğinin iletileri gitmeyen iletiler kuyruğuna koyma yetkisi de olması gerekir.)
- CKTI, CICStarafından sağlanan CICS görev başlatıcısı.
- CSQQTRMN, IBM MQtarafından sağlanan IMS tetikleyici izleyicisi.

Gitmeyen iletiler kuyruğundan ileti alabilen tek uygulama, bu iletileri işleyen bir 'özel' uygulama olmalıdır. Ancak, MQGET çağrılarını kullanarak kuyruktan iletileri otomatik olarak alabilecekleri için, uygulamalara MQPUT ' ler için teslim etmeyen iletiler kuyruğu için RACF UPDATE yetkisi verirseniz bir sorun ortaya çıkar. Alma işlemleri için gitmeyen iletiler kuyruğunu geçersiz kılamazsınız; bunu yaparsanız, 'özel' uygulamalar bile iletileri alamaz.

Bu sorunun çözümlerinden biri, gönderilmeyen iletiler kuyruğuna iki düzeyli erişim kurmaktır. CKTI, ileti kanalı aracısı işlemleri ya da kanal başlatıcı adres alanı ve 'özel' uygulamaların doğrudan erişimi vardır; diğer uygulamalar yalnızca bir diğer ad kuyruğu aracılığıyla kullanılmaz mektup kuyruğuna erişebilir. Bu diğer ad, uygulamaların iletileri gitmeyen iletiler kuyruğuna koymalarına izin verecek, ancak ondan ileti almalarına izin vermeyecek şekilde tanımlanır.

Bu şekilde çalışabilir:

1. Örnek thlqual.SCSQPROC(CSQ4INYG) içinde gösterildiği gibi, PUT (ENABLED) ve GET (ENABLED) öznitelikleriyle gerçek teslim mektubu kuyruğunu tanımlayın.
2. Şu kullanıcı kimliklerine, gönderilmeyen iletiler kuyruğu için RACF UPDATE yetkisi verin:
   - CKTI ve MCA ' ların ya da kanal başlatıcısı adres alanının altında çalıştığı kullanıcı kimlikleri.
   - 'Özel' gizli ileti kuyruğu işleme uygulamasıyla ilişkili kullanıcı kimlikleri.
3. Gerçek ileti kuyruğuna çözülen bir diğer ad kuyruğu tanımlayın, ancak diğer ad kuyruğuna şu öznitelikleri verin: PUT (ENABLED) ve GET (DISABLED). Diğer ad kuyruğuna, sapı, gönderilmeyen ileti kuyruğu adıyla aynı olan bir ad verin, ancak bu gövdeye ". PUT" karakterlerini ekleyin. Örneğin, gönderilmeyen ileti kuyruğu adı hlq.DEAD.QUEUE, diğer ad kuyruğu adı hlq.DEAD.QUEUE.PUT(KOYUN).
4. Bir iletiyi göndermeyen iletiler kuyruğuna koymak için uygulama diğer ad kuyruğunu kullanır. Uygulamanızın yapması gereken:
   - Gerçek posta kuyruğunun adını alın. Bunu yapmak için, MQOPEN kullanarak kuyruk yöneticisi nesnesini açar ve daha sonra, gitmeyen ileti kuyruğu adını almak için bir MQINQ verir.
   - Bu ada '.PUT' karakterlerini ekleyerek diğer ad kuyruğunun adını oluşturun; bu durumda, hlq.DEAD.QUEUE.PUT(KOYUN).
   - hlq.DEAD.QUEUE.PUT(KOYUN).
   - Diğer ad kuyruğuna karşı bir MQPUT komutu vererek iletiyi gerçek teslim edilmeyen ileti kuyruğuna koyun.
5. Uygulamayla ilişkili kullanıcı kimliğine diğer ad için RACF UPDATE yetkisi verin, ancak gerçek ileti kuyruğuna erişim yok (yetki YOK). Bu şu anlama gelir:
   - Uygulama, diğer ad kuyruğunu kullanarak iletileri gönderilemeyen iletiler kuyruğuna yerleştirebilir.
   - Diğer ad kuyruğu alma işlemleri için devre dışı bırakıldığından uygulama, diğer ad kuyruğunu kullanarak gitmeyen iletiler kuyruğundan ileti alamıyor.

   Uygulama, doğru RACF yetkisine sahip olduğu için gerçek gitmeyen iletiler kuyruğundan ileti alamıyor.

Çizelge 34 sayfa 202 , bu çözümdeki çeşitli katılımcılar için gerekli RACF yetkisini özetler.

Çizelge 34. Gönderilmeyen ileti kuyruğu ve diğer adı için RACF yetkisi

İlişkili kullanıcı kimlikleri	**Gerçek ileti kuyruğu (hlq.DEAD.QUEUE)**	**Diğer ad gönderilmeyen ileti kuyruğu (hlq.DEAD.QUEUE.PUT)**
MCA ya da kanal başlatıcı adres alanı ve CKTI	GÜNCELLE	YOK

Çizelge 34. Gönderilmeyen ileti kuyruğu ve diğer adı için RACF yetkisi (devamı var)		
İlişkili kullanıcı kimlikleri	**Gerçek ileti kuyruğu (hlq.DEAD.QUEUE)**	**Diğer ad gönderilmeyen ileti kuyruğu (hlq.DEAD.QUEUE.PUT)**
'Özel' uygulama (gönderilmeyen iletiler için kuyruk işleme)	GÜNCELLE	YOK
Kullanıcı tarafından yazılan uygulama kullanıcı kimlikleri	YOK	GÜNCELLE

Bu yöntemi kullanırsanız, uygulama, ileti kuyruğunun ileti uzunluğu üst sınırını (MAXMSGL) belirleyemez. Bunun nedeni, MAXMSGL özniteliğinin bir diğer ad kuyruğundan alınamaması olabilir. Bu nedenle, uygulamanız ileti uzunluğu üst sınırının 100 MB olduğunu, IBM MQ for z/OS ' in desteklediği büyüklük üst sınırının bu olduğunu varsaymalıdır. Gerçek gitmeyen iletiler kuyruğu da 100 MB ' lik bir MAXMSGL özniteliğiyle tanımlanmalıdır.

Not: Kullanıcı tarafından yazılan uygulama programları normalde iletileri gitmeyen iletiler kuyruğuna koymak için diğer kullanıcı yetkisini kullanmaz. Bu, gönderilmeyen iletiler kuyruğuna erişimi olan kullanıcı kimliklerinin sayısını azaltır.

z/OS *Sistem kuyruğu güvenliği*

Belirli kullanıcı kimliklerinin belirli sistem kuyruklarına erişmesine izin vermek için RACF erişimini ayarlamanız gerekir.

Sistem kuyruklarının çoğuna IBM MQ' un yardımcı bölümlerinden erişilir:

- CSQUTIL yardımcı programı
- İleti güvenlik ilkesi yardımcı programı (CSQ0UTIL)
- İşlemler ve denetim panoları
- Kanal başlatıcı adres alanı (Kuyruğa yollanmış Pub/Sub Daemon dahil)
- IBM MQ Console ve REST APItarafından kullanılan mqweb sunucusu.

Bu çalıştırmanın altında çalıştırıldığı kullanıcı kimliklerine bu kuyruklar için RACF erişimi verilmelidir (bkz. Çizelge 35 sayfa 203).

Çizelge 35. IBM MQ tarafından SYSTEM kuyruklarına erişim gerekli					
SYSTEM kuyruğu	**CSQUTIL**	**CSQ0UTIL**	**mqweb sunucusu**	**Operasyonlar ve denetim panoları**	**Dağıtılmış kuyruğa alma için kanal başlatıcı**
SYSTEM.ADMIN.CHANNEL.EVENT	-	-	-	-	GÜNCELLE
SYSTEM.ADMIN.COMMAND.QUEUE	-	-	GÜNCELLE	-	-
SYSTEM.BROKER.ADMIN.STREAM	-	-	-	-	Çeviri
SYSTEM.BROKER.CONTROL.QUEUE	-	-	-	-	Çeviri
SYSTEM.BROKER.DEFAULT.STREAM	-	-	-	-	Çeviri
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	-	-	-	-	GÜNCELLE
SYSTEM.CHANNEL.INITQ	-	-	-	-	GÜNCELLE
SYSTEM.CHANNEL.SYNCQ	-	-	-	-	GÜNCELLE
SYSTEM.CLUSTER.COMMAND.QUEUE	-	-	-	-	Çeviri
SYSTEM.CLUSTER.REPOSITORY.QUEUE	-	-	-	-	GÜNCELLE

Çizelge 35. IBM MQ tarafından SYSTEM kuyruklarına erişim gerekli (devamı var)					
SYSTEM kuyruğu	**CSQUTIL**	**CSQ0UTIL**	**mqweb sunucusu**	**Operasyonlar ve denetim panoları**	**Dağıtılmış kuyruğa alma için kanal başlatıcı**
SYSTEM.CLUSTER.TRANSMIT.QUEUE	-	-	-	-	Çeviri
SYSTEM.COMMAND.INPUT	GÜNCELLE	-	-	GÜNCELLE	GÜNCELLE
SYSTEM.COMMAND.REPLY.*	-	-	-	-	GÜNCELLE
SYSTEM.COMMAND.REPLY.MODEL	GÜNCELLE	-	-	GÜNCELLE	GÜNCELLE
SYSTEM.CSQOREXX.*	-	-	-	GÜNCELLE	-
SYSTEM.CSQUTIL.*	GÜNCELLE	-	-	-	-
SYSTEM.CSQXCMD.*	-	-	-	-	GÜNCELLE
SYSTEM.HIERARCHY.STATE	-	-	-	-	GÜNCELLE
SYSTEM.INTER.QMGR.CONTROL	-	-	-	-	GÜNCELLE
SYSTEM.INTER.QMGR.PUBS	-	-	-	-	GÜNCELLE
SYSTEM.INTER.QMGR.FANREQ	-	-	-	-	GÜNCELLE
SYSTEM.PROTECTION.ERROR.QUEUE	-	-	-	-	GÜNCELLE
SYSTEM.PROTECTION.POLICY.QUEUE	-	Güncelle“ 1” sayfa 204	-	-	READ
SYSTEM.QSG.CHANNEL.SYNCQ	-	-	-	-	GÜNCELLE
SYSTEM.QSG.TRANSMIT.QUEUE	-	-	-	-	GÜNCELLE
SYSTEM.REST.REPLY.QUEUE	-	-	GÜNCELLE	-	-
SYSTEM.BLUEMIX.REGISTRATION.QUEUE	-	-	-	-	GÜNCELLE

Notlar:

1. Advanced Message Security adres alanı kullanıcısı da bu kuyruğa OKUMA erişimi gerektirir.

z/OS *API-kaynak güvenliği erişimi hızlı başvurusu*

MQOPEN, **MQPUT1**, **MQSUB**ve **MQCLOSE** seçeneklerinin özeti ve farklı kaynak güvenliği tiplerinin gerektirdiği erişim.

Çizelge 36. MQOPEN, MQPUT1, MQSUB ve MQCLOSE seçenekleri ve gereken güvenlik yetkisi. **(1)** gibi gösterilen açıklama kutuları, bu çizelgeyi izleyen notlara bakın.				
	Gerekli RACF erişim düzeyi alt sınırı			
RACF Sınıf:	**MXTOPIC**	**MQQUEUE ya da MXQUEUE (1)**	**MQADMIN ya da MXADMIN**	**MQADMIN ya da MXADMIN**
RACF tanıtımı:	**(15 ya da 16)**	**(2)**	**(3)**	**(4)**
MQOPEN seçeneği				
MQOO_INQUIRE		OKUMA **(5)**	Çek yok	Çek yok

Çizelge 36. MQOPEN, MQPUT1, MQSUB ve MQCLOSE seçenekleri ve gereken güvenlik yetkisi. **(1)** gibi gösterilen açıklama kutuları, bu çizelgeyi izleyen notlara bakın. *(devamı var)*

	Gerekli RACF erişim düzeyi alt sınırı			
RACF Sınıf:	**MXTOPIC**	**MQQUEUE ya da MXQUEUE (1)**	**MQADMIN ya da MXADMIN**	**MQADMIN ya da MXADMIN**
RACF tanıtımı:	**(15 ya da 16)**	**(2)**	**(3)**	**(4)**
MQOO_GÖZ AT		READ	Çek yok	Çek yok
MQOO_INPUT_ *		GÜNCELLE	Çek yok	Çek yok
MQOO_SAVE_ALL_CONTEXT **(6)**		GÜNCELLE	Çek yok	Çek yok
MQOO_OUTPUT (KULLANıM = OLAğAN) **(7)**		GÜNCELLE	Çek yok	Çek yok
MQOO_PASS_IDENTITY_CONTEXT **(8)**		GÜNCELLE	READ	Çek yok
MQOO_PASS_ALL_CONTEXT **(8) (9)**		GÜNCELLE	READ	Çek yok
MQOO_SET_IDENTITY_CONTEXT **(8) (9)**		GÜNCELLE	GÜNCELLE	Çek yok
MQOO_SET_ALL_CONTEXT **(8) (10)**		GÜNCELLE	CONTROL	Çek yok
MQOO_OUTPUT (KULLANıM (Xmitq) **(11)**		GÜNCELLE	CONTROL	Çek yok
MQOO_OUTPUT (konu nesnesi)	GüNCELLEME **(16)**			
MQOO_OUTPUT (konu nesnesine diğer ad kuyruğu)	GüNCELLEME **(16)**	GÜNCELLE		
MQOO_SET		Çeviri	Çek yok	Çek yok
MQOO_ALTERNATE_USER_AUTHORITY		**(12)**	**(12)**	GÜNCELLE
MQPUT1 seçeneği				
Normal bir kuyruğa koyma **(7)**		GÜNCELLE	Çek yok	Çek yok
MQPMO_PASS_IDENTITY_CONTEXT		GÜNCELLE	READ	Çek yok
MQPMO_PASS_ALL_CONTEXT		GÜNCELLE	READ	Çek yok
MQPMO_SET_IDENTITY_CONTEXT		GÜNCELLE	GÜNCELLE	Çek yok
MQPMO_SET_ALL_CONTEXT		GÜNCELLE	CONTROL	Çek yok
MQOO_ÇıKışı İletim kuyruğuna koyma **(11)**		GÜNCELLE	CONTROL	Çek yok
MQOO_OUTPUT (konu nesnesi)	GüNCELLEME **(16)**			
MQOO_OUTPUT (konu nesnesine diğer ad kuyruğu)	GüNCELLEME **(16)**	GÜNCELLE		
MQPMO_ALTERNATE_USER_AUTHORITY		**(13)**	**(13)**	GÜNCELLE
MQCLOSE seçeneği				
MQCO_DELETE **(14)**		Çeviri	Çek yok	Çek yok
MQCO_DELETE_PURGE **(14)**		Çeviri	Çek yok	Çek yok

Çizelge 36. MQOPEN, MQPUT1, MQSUB ve MQCLOSE seçenekleri ve gereken güvenlik yetkisi. **(1)** gibi gösterilen açıklama kutuları, bu çizelgeyi izleyen notlara bakın. *(devamı var)*

	Gerekli RACF erişim düzeyi alt sınırı			
RACF Sınıf:	**MXTOPIC**	**MQQUEUE ya da MXQUEUE (1)**	**MQADMIN ya da MXADMIN**	**MQADMIN ya da MXADMIN**
RACF tanıtımı:	**(15 ya da 16)**	**(2)**	**(3)**	**(4)**
MQCO_REMOVE_SUB	**(15)**			
MQSUB seçeneği				
MQSO_OLUşTURMA	**(15)**	**(17)**	**(18)**	
MQSO_ALTER	**(15)**	**(17)**	**(18)**	
MQSO_RESUME	OKUMA **(15)**	**(17)**	Çek yok	
MQSO_ALTERNATE_USER_AUTHORITY				GÜNCELLE
MQSO_SET_IDENTITY_CONTEXT			**(18)**	

Not:

1. Bu seçenek kuyruklarla sınırlı değildir. Ad listesi için MQNLIST ya da MXNLIST sınıfını ve işlemler için MQPROC ya da MXPROC sınıfını kullanın.
2. RACF tanıtımı kullan: hlq.resourcename
3. RACF tanıtımı kullan: hlq.CONTEXT.queuename
4. RACF profile: hlq.ALTERNATE.USER. `alternateuserid`

   `alternateuserid` , nesne tanımlayıcısının *`AlternateUserId`* alanında belirtilen kullanıcı kimliğidir. Bu denetim için, kullanıcı tanıtıcısının yalnızca ilk 8 karakterinin kullanıldığı diğer denetimlerden farklı olarak, *`AlternateUserId`* alanının en çok 12 karakterinin kullanıldığını unutmayın.
5. Sorgular için kuyruk yöneticisi açılırken denetim yapılmadı.
6. MQOO_INPUT_ * da belirtilmelidir. Bu, yerel, model ya da diğer ad kuyruğu için geçerlidir.
7. Bu denetim, **Usage** kuyruk özniteliği MQUS_NORMAL olan bir yerel kuyruk ya da model kuyruğu için ve bir diğer ad ya da uzak kuyruk (bağlı kuyruk yöneticisine tanımlı) için de yapılır. Kuyruk, açık bir şekilde *`ObjectQMgrName`* (bağlı kuyruk yöneticisinin adı değil) belirtilerek açılan bir uzak kuyruksa, *`ObjectQMgrName`* ile aynı ada sahip (MQUS_ILETIM **Usage** kuyruk özniteliğine sahip yerel bir kuyruk olması gerekir) kuyruk için denetim gerçekleştirilir.
8. MQOO_OUTPUT da belirtilmelidir.
9. MQOO_PASS_IDENTITY_CONTEXT bu seçenek tarafından da örtük olarak belirtilmiştir.
10. MQOO_PASS_IDENTITY_CONTEXT, MQOO_PASS_ALL_CONTEXT ve MQOO_SET_IDENTITY_CONTEXT bu seçenek tarafından da örtük olarak belirtilmiştir.
11. Bu denetim, **Usage** kuyruk özniteliği MQUS_ILETIM olan yerel ya da model kuyruğu için yapılır ve doğrudan çıkış için açılır. Bir uzak kuyruk açıldığında bu geçerli olmaz.
12. En az bir MQOO_INQUIRE, MQOO_BROWSE, MQOO_INPUT_ *, MQOO_OUTPUT ya da MQOO_SET de belirtilmelidir. Gerçekleştirilen denetim, belirtilen diğer seçenekler için yapılan denetimle aynı.
13. Gerçekleştirilen denetim, belirtilen diğer seçenekler için yapılan denetimle aynı.
14. Bu, yalnızca doğrudan açılan, bir model kuyruğuyla açılmayan kalıcı dinamik kuyruklar için geçerlidir. Geçici bir dinamik kuyruğu silmek için güvenlik gerekmez.
15. RACF profile hlq.SUBSCRIBE.topicnamekomutunu kullanın.
16. RACF profile hlq.PUBLISH.topicnamekomutunu kullanın.

17. MQSUB isteğinde, gönderilecek yayınlar için bir hedef kuyruk belirttiyseniz, o kuyruğa koyma yetkiniz olduğundan emin olmak için o kuyruk için bir güvenlik denetimi gerçekleştirilir.
18. MQSUB isteğinde MQSO_CREATE ya da MQSO_ALTER seçenekleri belirtilirse, MQSD yapısındaki kimlik bağlamı alanlarından herhangi birini ayarlamak istiyorsanız, MQSO_SET_IDENTITY_CONTEXT seçeneğini belirtmeniz ve hedef kuyruğa ilişkin bağlam profili için uygun yetkiye sahip olmanız da gerekir.

z/OS *Konu güvenliğine ilişkin profiller*

Konu güvenliği etkinse, uygun sınıflarda tanıtımlar tanımlamanız ve bu tanıtımlara gereken grupların ya da kullanıcı kimliklerinin erişmesine izin vermeniz gerekir.

Bir konu ağacındaki konu güvenliği kavramı Yayınla/abone ol güvenliğikonusunda açıklanmıştır.

Konu güvenliği etkinse, aşağıdaki işlemleri gerçekleştirmeniz gerekir:

- **MXTOPIC** ya da **GMXTOPIC** sınıflarındaki tanıtımları tanımlayın.
- Konuları kullanan IBM MQ API isteklerini yayınlayabilmeleri için gerekli grupların ya da kullanıcı kimliklerinin bu profillere erişmesine izin verin.

Konu güvenliğine ilişkin profiller şu formu alır:

```
hlq.SUBSCRIBE.topicname
hlq.PUBLISH.topicname
```

burada:

- `hlq` , `qmgr-name` (kuyruk yöneticisi adı) ya da `qsg-name` (kuyruk paylaşım grubu adı) olabilir.
- `topicname` , konu ağacındaki konu denetim düğümünün adıdır; bu ad, MQSUB çağrısı yoluyla abone olunan ya da MQOPEN çağrısıyla yayınlanmakta olan konuyla ilişkilidir.

Kuyruk yöneticisi adı öneki eklenmiş bir tanıtım, o kuyruk yöneticisindeki tek bir konuya erişimi denetler. Kuyruk paylaşım grubu adı öneki eklenmiş bir tanıtım, kuyruk paylaşım grubu içindeki tüm kuyruk yöneticilerine o konu adını taşıyan bir ya da daha çok konuya erişimi denetler. Bu erişim, söz konusu kuyruk yöneticisinde ilgili konu için kuyruk yöneticisi düzeyinde bir tanıtım tanımlanarak, tek bir kuyruk yöneticisinde geçersiz kılınabilir.

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesiyse ve hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ önce kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar.

Abone olun

Bir konuya abone olmak için, abone olmaya çalıştığınız konuya ve yayınlara ilişkin hedef kuyruğa erişmeniz gerekir.

Bir MQSUB isteği yayınladığınızda aşağıdaki güvenlik denetimleri gerçekleşmiş:

- Bu konuya abone olmak için uygun erişim düzeyine sahip olup olmadığınızı ve çıkış için hedef kuyruğun (belirtildiyse) açılıp açılmadığını belirler
- Hedef kuyruğa uygun erişim düzeyine sahip olup olmadığınızı belirler.

Çizelge 37. Konu güvenliğinin abone olması için gereken erişim düzeyi

MQSUB seçeneği	**MXTOPIC sınıfındaki `hlq.SUBSCRIBE.topicname` profili için RACF erişimi gerekli**
MQSO_CREATE ve MQSO_ALTER	Çeviri
MQSO_RESUME	READ

Çizelge 38. Yönetilmeyen bir hedef kuyruğu kullanarak abone olmak için ek yetki gerekli	
MQSUB seçeneği	**MQADMIN ya da MXADMIN sınıfındaki `hlq.CONTEXT.queuename` profili için RACF erişimi gerekli**
MQSO_CREATE, MQSO_ALTER ve MQSO_RESUME	GÜNCELLE
	MQQUEUE ya da MXQUEUE sınıfındaki `hlq.queuename` profili için RACF erişimi gerekli
MQSO_CREATE ve MQSO_ALTER	GÜNCELLE
	MQADMIN ya da MXADMIN sınıfındaki `hlq.ALTERNATE.USER.alternateuserid` profili için RACF erişimi gerekli
MQSO_ALTERNATE_USER_AUTHORITY	GÜNCELLE

Aboneliklere ilişkin yönetilen kuyruklara ilişkin dikkat edilecek noktalar

Konuya abone olmanıza izin verilip verilmediğini görmek için bir güvenlik denetimi gerçekleştirilir. Ancak, yönetilen kuyruk yaratıldığında ya da iletileri bu hedef kuyruğa koyma erişiminiz olup olmadığını belirlemek için güvenlik denetimi gerçekleştirilmez.

Yönetilen bir kuyruğu silmeyi kapatamazsınız.

Kullanılan model kuyrukları şunlardır: `SYSTEM.DURABLE.MODEL.QUEUE` ve `SYSTEM.NDURABLE.MODEL.QUEUE`.

Bu model kuyruklarından oluşturulan yönetilen kuyruklar, son niteleyicinin öngörülemediği `SYSTEM.MANAGED.DURABLE.A346EF00367849A0` ve `SYSTEM.MANAGED.NDURABLE.A346EF0036785EA0` biçimindedir.

Bu kuyruklara herhangi bir kullanıcı erişimi vermeyin. Kuyruklar, yetki verilmediği için `SYSTEM.MANAGED.DURABLE.*` ve `SYSTEM.MANAGED.NDURABLE.*` biçiminde soysal tanıtımlar kullanılarak korunabilir.

İletiler, MQSUB isteğinde döndürülen tanıtıcı kullanılarak bu kuyruklardan alınabilir.

Belirtilen `MQCO_REMOVE_SUB` seçeneğiyle bir abonelik için belirtik olarak MQCLOSE çağrısı yaparsanız ve bu tanıtıcı altında kapatmakta olduğunuz aboneliği yaratmadıysanız, işlemi gerçekleştirmek için doğru yetkiye sahip olduğunuzdan emin olmak üzere, kapanış sırasında bir güvenlik denetimi gerçekleştirilir.

Çizelge 39. Bir abone olma işleminin kapatılması için konu güvenliğine ilişkin profiller için gereken erişim düzeyi	
MQCLOSE seçeneği	**MXTOPIC sınıfındaki `hlq.SUBSCRIBE.topicname` profili için RACF erişimi gerekli**
MQCO_REMOVE_SUB	Çeviri

Yayınla

Bir konuda yayınlama yapmak için konuya ve diğer ad kuyruklarını kullanıyorsanız, diğer ad kuyruğuna da erişmeniz gerekir.

Çizelge 40. Konu güvenliğinin yayınlanması için gereken erişim düzeyi	
MQOPEN ya da MQPUT1 seçeneği	**MXTOPIC sınıfındaki `hlq.PUBLISH.topicname` profili için RACF erişimi gerekli**
MQOO_OUTPUT ya da MQPUT1	GÜNCELLE

Çizelge 41. Bir konuya çözülen bir diğer ad kuyruğunu açmak için gereken erişim düzeyi	
MQOPEN ya da MQPUT1 seçeneği	**Diğer ad kuyruğu için MQQUEUE ya da MXQUEUE sınıfındaki `hlq.queuename` profili için RACF erişimi gerekli**
MQOO_OUTPUT ya da MQPUT1	GÜNCELLE

Bir konu adına çözülen bir diğer ad kuyruğu yayınlanmak üzere açıldığında konu güvenliğinin nasıl çalıştığına ilişkin ayrıntılar için bkz. "Yayınlama işlemine ilişkin konulara çözümleyen diğer ad kuyruklarına ilişkin dikkat edilecek noktalar" sayfa 209.

PUT ya da GET kısıtlamaları için hedef kuyruklar için kullanılan diğer ad kuyruklarını göz önünde bulundurduğunuzda, bkz. "Diğer ad kuyruklarına ilişkin dikkat edilecek noktalar" sayfa 199.

Bir uygulamanın bir konu güvenliği profili için sahip olduğu RACF erişim düzeyi değiştirilirse, değişiklikler yalnızca o konu için elde edilen yeni nesne tanıtıcıları (yani, yeni bir MQSUB ya da MQOPEN) için geçerli olur. Değişiklik sırasında var olan bu tanıtıcıları, konuya var olan erişimlerini korur. Ayrıca, var olan aboneler önceden yaptıkları aboneliklere erişimlerini korur.

Yayınlama işlemine ilişkin konulara çözümleyen diğer ad kuyruklarına ilişkin dikkat edilecek noktalar

Bir konuya çözülen diğer ad kuyruğu için bir MQOPEN ya da MQPUT1 çağrısı yayınladığınızda, IBM MQ iki kaynak denetimi yapar:

- MQOPEN ya da MQPUT1 çağrısındaki nesne tanımlayıcısında (MQOD) belirtilen diğer ad kuyruğu adına ilişkin ilk ad.
- Diğer ad kuyruğunun çözüldüğü konuya ilişkin ikinci

Bu davranışın, diğer ad kuyrukları diğer kuyruklara çözüldüğünde aldığınız davranıştan farklı olduğunu unutmayın. Yayınlama işleminin devam etmesi için her iki tanıtıma da doğru erişmeniz gerekir.

Sistem konusu güvenliği

Kanal başlatıcı adres alanından aşağıdaki sistem konularına erişilir.

Bu çalıştırmanın altında çalıştırıldığı kullanıcı kimliklerine, Çizelge 42 sayfa 209içinde gösterildiği gibi bu kuyruklar için RACF erişimi verilmelidir.

Çizelge 42. SYSTEM konularına erişim gerekli		
SYSTEM konusu	**Profil**	**Dağıtılmış kuyruğa alma için kanal başlatıcı**
SYSTEM.BROKER.ADMIN.STREAM	hlq.PUBLISH.topicname	GÜNCELLE
SYSTEM.BROKER.ADMIN.STREAM	hlq.SUBSCRIBE.topicname	Çeviri

z/OS *Süreçlere ilişkin profiller*

Süreç güvenliği etkinse, uygun sınıflarda tanıtımlar tanımlamanız ve bu tanıtımlara gerekli grupların ya da kullanıcı kimliklerinin erişmesine izin vermeniz gerekir.

Süreç güvenliği etkinse şunları yapmanız gerekir:

- Büyük harfli tanıtımlar kullanılıyorsa, tanıtımları **MQPROC** ya da **GMQPROC** sınıflarında tanımlayın.
- Büyük ve küçük harf karışık tanıtımlar kullanılıyorsa, **MXPROC** ya da **GMXPROC** sınıflarındaki tanıtımları tanımlayın.

- Gerekli grupların ya da kullanıcı kimliklerinin, süreçleri kullanan IBM MQ API isteklerini yayınlayabilmesi için bu profillere erişmesine izin verin.

Süreçlere ilişkin profiller şu formu alır:

```
hlq.processname
```

Burada `hlq` , `qmgr-name` (kuyruk yöneticisi adı) ya da `qsg-name` (kuyruk paylaşım grubu adı) olabilir ve `processname` , açılmakta olan işlemin adıdır.

Kuyruk yöneticisi adı öneki eklenmiş bir tanıtım, o kuyruk yöneticisindeki tek bir süreç tanımlamasına erişimi denetler. Kuyruk paylaşım grubu adı öneki eklenen bir tanıtım, kuyruk paylaşım grubu içindeki tüm kuyruk yöneticilerine bu adı taşıyan bir ya da daha çok süreç tanımlamasına erişimi denetler. Bu erişim, o kuyruk yöneticisinde o süreç tanımlaması için kuyruk yöneticisi düzeyinde bir tanıtım tanımlanarak, tek bir kuyruk yöneticisinde geçersiz kılınabilir.

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesiyse ve hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ önce kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar.

Aşağıdaki tablo, bir süreci açmak için gereken erişimi göstermektedir.

Çizelge 43. Süreç güvenliği için erişim düzeyleri

MQOPEN seçeneği	**RACF hlq.processname için gereken erişim düzeyi**
MQOO_INQUIRE	READ

Örneğin, MQS9kuyruk yöneticisinde, RACF grubu INQVPRC sorgulayabilmelidir (`MQINQ`) V harfi ile başlayan tüm işlemlerde. Bunun için RACF tanımlamaları:

```
RDEFINE MQPROC MQS9.V* UACC(NONE)
PERMIT MQS9.V* CLASS(MQPROC) ID(INQVPRC) ACCESS(READ)
```

Bir süreç tanımlaması nesnesi açıldığında belirtilen açma seçeneklerine bağlı olarak, diğer kullanıcı güvenliği de etkin olabilir.

z/OS *Ad listesi profilleri*

Ad listesi güvenliği etkinse, tanıtımları uygun sınıflarda tanımlayın ve gereken gruplara ya da kullanıcı kimliklerine bu tanıtımlar için erişim verin.

Ad listesi güvenliği etkinse şunları yapmanız gerekir:

- Büyük harfli tanıtımlar kullanılıyorsa, tanıtımları **MQNLIST** ya da **GMQNLIST** sınıflarında tanımlayın.
- Büyük ve küçük harf karışık tanıtımlar kullanılıyorsa, **MXNLIST** ya da **GMXNLIST** sınıflarındaki tanıtımları tanımlayın.
- Gerekli grupların ya da kullanıcı kimliklerinin bu profillere erişmesine izin verin.

Ad listesi profilleri şu formu alır:

```
hlq.namelistname
```

Burada `hlq` , `qmgr-name` (kuyruk yöneticisi adı) ya da `qsg-name` (kuyruk paylaşım grubu adı) olabilir ve `namelistname` , açılmakta olan ad listesi adıdır.

Kuyruk yöneticisi adı öneki eklenen bir tanıtım, o kuyruk yöneticisindeki tek bir ad melist 'e erişimi denetler. Kuyruk paylaşım grubu adı öneki eklenen bir tanıtım, kuyruk paylaşım grubu içindeki tüm kuyruk yöneticilerine bu adı taşıyan bir ya da daha çok ad grubuna erişimi denetler. Bu erişim, söz konusu

kuyruk yöneticisinde ilgili ad listesi için bir kuyruk yöneticisi düzeyi tanıtımı tanımlanarak, tek bir kuyruk yöneticisinde geçersiz kılınabilir.

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesiyse ve hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ önce kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar.

Aşağıdaki tablo, bir ad listesi açmak için gereken erişimi göstermektedir.

Çizelge 44. Ad listesi güvenliği için erişim düzeyleri

MQOPEN seçeneği	**RACF hlq.namelistname için gereken erişim düzeyi**
MQOO_INQUIRE	READ

Örneğin, kuyruk yöneticisinde (ya da kuyruk paylaşım grubunda) PQM3, RACF grubu DEPT571 sorgulayabilmelidir (`MQINQ`) Bu namelistlerde:

- "DEPT571"ile başlayan tüm namelistler.
- PRINTER/DESTINATIONS/DEPT571
- GÜNDEM/ISTEK/KUYRUKLAR
- WAREHOUSE.BROADCAST

Bunu yapmak için RACF tanımlamaları şunlardır:

```
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)
```

Bir ad listesi nesnesi açıldığında belirlenen seçeneklere bağlı olarak, diğer kullanıcı güvenliği etkin olabilir.

Sistem ad listesi güvenliği

Sistem ad melislerinin çoğuna IBM MQ' un yan kısımları tarafından erişilir:

- CSQUTIL yardımcı programı
- İşlemler ve denetim panoları
- Kanal başlatıcı adres alanı (Kuyruğa yollanmış yayınlama/abone olma cini de içinde olmak üzere)

Bu çalıştırmanın altında çalıştırıldığı kullanıcı kimliklerine, Çizelge 45 sayfa 211içinde gösterildiği gibi bu ad listesi için RACF erişimi verilmelidir.

Çizelge 45. IBM MQ tarafından SYSTEM ad listesi için gereken erişim

SYSTEM ad listesi	**CSQUTIL**	**Operasyonlar ve denetim panoları**	**Dağıtılmış kuyruğa alma için kanal başlatıcı**
SYSTEM.QPUBSUB.QUEUE.NAMELIST	-	-	READ
SYSTEM.QPUBSUB.SUBPOINT.NAMELIST	-	-	READ

z/OS *Diğer kullanıcı güvenliği için profiller*

Diğer kullanıcı güvenliği etkinse, uygun sınıflarda tanıtımlar tanımlamanız ve bu tanıtımlara gerekli grupların ya da kullanıcı kimliklerinin erişmesine izin vermeniz gerekir.

*AlternateUserId*ile ilgili daha fazla bilgi için bkz. AlternateUserID (MQCHAR12).

Diğer kullanıcı güvenliği etkinse, aşağıdakileri yapmanız gerekir:

- Büyük harfli tanıtımlar kullanıyorsanız, MQADMIN ya da GMQADMIN sınıflarında tanıtımları tanımlayın.
- Karışık büyük ve küçük harf profilleri kullanıyorsanız MXADMIN ya da GMXADMIN sınıflarındaki profilleri tanımlayın.

Nesne açıldığında ALTERNATE_USER_AUTHORITY seçeneklerini kullanabilmeleri için, gerekli grupların ya da kullanıcı kimliklerinin bu tanıtımlara erişmesine izin verin.

Diğer kullanıcı güvenliğine ilişkin tanıtımlar altsistem düzeyinde ya da kuyruk paylaşım grubu düzeyinde belirtilebilir ve aşağıdaki formu alır:

```
hlq.ALTERNATE.USER.alternateuserid
```

Burada `hlq` , `qmgr-name` (kuyruk yöneticisi adı) ya da `qsg-name` (kuyruk paylaşım grubu adı) olabilir ve `alternateuserid` , nesne tanımlayıcıdaki *AlternateUserId* alanının değeridir.

Kuyruk yöneticisi adı öneki eklenmiş bir tanıtım, o kuyruk yöneticisinde diğer bir kullanıcı kimliğinin kullanılmasını denetler. Kuyruk paylaşım grubu adı öneki eklenmiş bir tanıtım, kuyruk paylaşım grubu içindeki tüm kuyruk yöneticilerindeki diğer bir kullanıcı kimliğinin kullanılmasını denetler. Bu alternatif kullanıcı kimliği, doğru erişime sahip bir kullanıcı tarafından kuyruk paylaşım grubu içindeki herhangi bir kuyruk yöneticisinde kullanılabilir. Bu erişim, söz konusu kuyruk yöneticisinde o diğer kullanıcı kimliği için bir kuyruk yöneticisi düzeyi tanıtımı tanımlanarak, tek bir kuyruk yöneticisinde geçersiz kılınabilir.

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesiyse ve hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ önce kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar.

Aşağıdaki çizelge, alternatif bir kullanıcı seçeneği belirtilirken erişimi gösterir.

Çizelge 46. Alternatif kullanıcı güvenliği için erişim düzeyleri

MQOPEN, MQSUB ya da MQPUT1 seçeneği	**RACF erişim düzeyi gerekli**
MQOO_ALTERNATE_USER_AUTHORITY MQSO_ALTERNATE_USER_AUTHORITY MQPMO_ALTERNATE_USER_AUTHORITY	GÜNCELLE

Alternatif kullanıcı güvenlik denetimlerine ek olarak, kuyruk, süreç, ad listesi ve bağlam güvenliği için diğer güvenlik denetimleri de yapılabilir. Sağlandıysa, diğer kullanıcı kimliği yalnızca kuyrukta, süreç tanımlamasında ya da ad listesi kaynaklarında güvenlik denetimleri için kullanılır. Diğer kullanıcı ve bağlam güvenliği denetimleri için, denetiin kullanılmasını isteyen kullanıcı kimliği. Kullanıcı kimliklerinin nasıl işlendiğine ilişkin ayrıntılar için bkz. “z/OS üzerinde güvenlik denetimi için kullanıcı kimlikleri” sayfa 234. Kuyruk, bağlam ve diğer kullanıcı güvenliğinin tümü etkin olduğunda gerekli olan açma seçeneklerini ve güvenlik denetimlerini gösteren bir özet tablo için bkz. Çizelge 36 sayfa 204.

Diğer bir kullanıcı tanıtımı, istekte bulunan kullanıcı kimliğine, diğer kullanıcı kimliğiyle belirtilen kullanıcı kimliğiyle ilişkili kaynaklara erişim yetkisi verir. Örneğin, QMPY kuyruk yöneticisinde PAYSERV kullanıcı kimliği altında çalışan bordro sunucusu, tümü PS ile başlayan personel kullanıcı kimliklerinden gelen istekleri işler. Bordro sunucusu tarafından gerçekleştirilen işin, istekte bulunan kullanıcının kullanıcı kimliği altında gerçekleştirilmesini sağlamak için diğer kullanıcı yetkisi kullanılır. İstekte bulunan programlar MQPMO_DEFAULT_CONTEXT koyma iletisi seçeneğini kullanarak ileti oluşturduğundan, bordro sunucusu alternatif kullanıcı kimliği olarak hangi kullanıcı kimliğinin belirtileceğini bilir. Alternatif kullanıcı kimliklerinin nereden alınacağı hakkında daha fazla ayrıntı için bkz. “z/OS üzerinde güvenlik denetimi için kullanıcı kimlikleri” sayfa 234 .

Aşağıdaki örnek RACF tanımlamaları, sunucu programının PS karakterleriyle başlayan diğer kullanıcı kimliklerini belirtmesini sağlar:

```
RDEFINE MQADMIN QMPY.ALTERNATE.USER.PS* UACC(NONE)
PERMIT QMPY.ALTERNATE.USER.PS* CLASS(MQADMIN) ID(PAYSERV) ACCESS(UPDATE)
```

Not:

1. Nesne tanımlayıcı ve abonelik tanımlayıcısındaki *AlternateUserId* alanları 12 bayt uzunluğundur. Tanıtım denetimlerinde 12 baytın tümü kullanılır, ancak IBM MQtarafından kullanıcı kimliği olarak yalnızca ilk 8 bayt kullanılır. Bu kullanıcı kimliği kesilmesi istenmezse, isteği yapan uygulama programları, 8 byte üzerindeki diğer kullanıcı kimliklerini daha uygun bir yere çevirmelidir.
2. MQOO_ALTERNATE_USER_AUTHORITY, MQSO_ALTERNATE_USER_AUTHORITY ya da MQPMO_ALTERNATE_USER_AUTHORITY belirtirseniz ve nesne tanımlayıcısında bir *AlternateUserId* alanı belirtmezseniz, boş bir kullanıcı kimliği kullanılır. Diğer kullanıcı güvenliği amacıyla, *AlternateUserId* niteleyicisi için kullanılan kullanıcı kimliği -BLANK-. Örneğin RDEF MQADMIN hlq.ALTERNATE.USER.-BLANK-.

   Kullanıcının bu tanıtıma erişmesine izin verilirse, diğer tüm denetimler kullanıcı kimliği boşlukla yapılır. Boş kullanıcı kimliklerine ilişkin ayrıntılar için bkz. “Boş kullanıcı kimlikleri ve UACC düzeyleri” sayfa 243.

Kullanıcı kimlikleri için soysal alternatif kullanıcı tanıtımlarını kullanmanızı sağlayan bir adlandırma kuralınız varsa, diğer kullanıcı kimliklerinin yönetimi daha kolay olur. Yoksa, RACF RACVAR özelliğini kullanabilirsiniz. RACVAR ' ın kullanılmasıyla ilgili ayrıntılar için z/OS Security Server RACF belgelerine bakın.

Bir ileti, diğer kullanıcı yetkisiyle açılmış bir kuyruğa konduğunda ve iletinin bağlamı kuyruk yöneticisi tarafından oluşturulduğunda, MQMD_USER_IDENTIFIER alanı diğer kullanıcı kimliğine ayarlanır.

z/OS *Bağlam güvenliğine ilişkin profiller*

Bağlam güvenliği etkinse, ileti bağlamı bilgilerine erişimi denetlemek için uygun sınıflarda tanıtımlar tanımlamanız ve bu tanıtımlara gereken grupların ya da kullanıcı kimliklerinin erişmesine izin vermeniz gerekir. İleti bağlamı, ileti tanımlayıcısı (MQMD) içinde bulunuyor.

Bağlam güvenliği için profilleri kullanma

Bağlam güvenliği etkinse, kullanıcıların belirli bir kuyruktaki iletilere ilişkin bağlam bilgilerine erişmelerine izin vermek için ya da belirli bir konuyu yayınlarken, aşağıdaki sınıflardan birinde bir tanıtım tanımlamanız gerekir:

- Büyük harf profilleri kullanılıyorsa MQADMIN sınıfı.
- Büyük ve küçük harf karışık profiller kullanılıyorsa MXADMIN sınıfı.

Bağlam güvenliğine ilişkin profiller, altsistem düzeyinde ya da kuyruk paylaşım grubu düzeyinde belirtilebilir ve aşağıdaki formu alır:

```
hlq.CONTEXT.queuename
hlq.CONTEXT.topicname
```

Burada *hlq* kuyruk yöneticisi adı ya da kuyruk paylaşım grubu adı olabilir ve *queuename* ve *topicname* , bağlam profilini tanımlamak istediğiniz kuyruğun ya da konunun tam ya da soysal adı olabilir.

Kuyruk yöneticisi adı öneki ve kuyruk ya da konu adı olarak ** belirtilmiş bir tanıtım, o kuyruk yöneticisine ilişkin tüm kuyruklarda ve konularda bağlam güvenliğinin denetlenmesine olanak sağlar. Bu, söz konusu kuyruktaki ya da konudaki bağlam için belirli bir profil tanımlanarak, tek bir kuyrukta ya da konuda geçersiz kılınabilir.

Kuyruk paylaşım grubu adı öneki ve kuyruk ya da konu adı olarak ** belirtilmiş bir tanıtım, kuyruk paylaşım grubu içindeki kuyruk yöneticilerine ait tüm kuyruklar ve konular için bağlam denetimi sağlar. Kuyruk yöneticisi adı öneki eklenmiş bir tanıtım belirtilerek, o kuyruk yöneticisine ilişkin bağlam için kuyruk yöneticisi düzeyinde bir tanıtım tanımlanarak, bu tanıtım tek bir kuyruk yöneticisinde geçersiz

kılınabilir. Kuyruk ya da konu adıyla bir tanıtım soneki belirlenerek, tek bir kuyrukta ya da konuda geçersiz kılınabilir.

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesiyse ve hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ önce kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar.

Bu profil için gerekli gruplara ya da kullanıcı kimliklerine erişim izni vermeniz gerekir. Aşağıdaki çizelge, kuyruk açıldığında bağlam seçeneklerinin belirtimine bağlı olarak, gereken erişim düzeyini gösterir.

Çizelge 47. Bağlam güvenliği için erişim düzeyleri

MQOPEN ya da MQPUT1 seçeneği	**RACF hlq.CONTEXT.queuename ya da hlq.CONTEXT.topicname için gereken erişim düzeyi**
MQPMO_NO_CONTEXT	Bağlam güvenliği denetimi yok
MQPMO_DEFAULT_CONTEXT	Bağlam güvenliği denetimi yok
MQOO_SAVE_ALL_CONTEXT	Bağlam güvenliği denetimi yok
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	GÜNCELLE
MQOO_SET_ALL_CONTEXT MQPMO_SET_ALL_CONTEXT	CONTROL
MQOO_OUTPUT ya da MQPUT1(KULLANıM (XMITQ))	CONTROL
MQSUB seçeneği	
MQSO_SET_IDENTITY_CONTEXT **(Not 2)**	GÜNCELLE

Not:

1. Dağıtımlı kuyruğa alma için kullanılan kullanıcı kimlikleri, iletileri hedef kuyruğa koymak için `hlq.CONTEXT.queuename` ' e `CONTROL` erişimi gerektirir. Kullanılan kullanıcı kimliklerine ilişkin bilgi için bkz. “Kanal başlatıcısı tarafından kullanılan kullanıcı kimlikleri” sayfa 238 .
2. MQSUB isteğinde `MQSO_CREATE` ya da `MQSO_ALTER` seçenekleri belirtilirse, MQSD yapısında kimlik bağlamı alanlarından herhangi birini ayarlamak istiyorsanız, `MQSO_SET_IDENTITY_CONTEXT` seçeneğini belirtmeniz gerekir. Ayrıca, hedef kuyruğa ilişkin bağlam tanıtımı için gereken yetkiyi de vermeniz gerekir.

Komutları sistem komutu giriş kuyruğuna koyarsanız, doğru kullanıcı kimliğini komutla ilişkilendirmek için varsayılan bağlam koyma iletisi seçeneğini kullanın.

Örneğin, IBM MQtarafından sağlanan yardımcı program CSQUTIL, kuyruklardaki iletileri boşaltmak ve yeniden yüklemek için kullanılabilir. Boşaltılan iletiler bir kuyruğa geri yüklendiğinde, CSQUTIL yardımcı programı iletileri özgün durumlarına döndürmek için `MQOO_SET_ALL_CONTEXT` seçeneğini kullanır. Bu açık seçeneğin gerektirdiği kuyruk güvenliğine ek olarak, bağlam yetkisi de gereklidir. Örneğin, MQS1kuyruk yöneticisindeki BACKGRP grubu için bu yetki gerekliyse, bu yetki aşağıdaki şekilde tanımlanır:

```
RDEFINE MQADMIN MQS1.CONTEXT.** UACC(NONE)
PERMIT MQS1.CONTEXT.** CLASS(MQADMIN) ID(BACKGRP) ACCESS(CONTROL)
```

Belirlenen seçeneklere ve gerçekleştirilen güvenlik tiplerine bağlı olarak, kuyruk açıldığında diğer güvenlik denetimi tipleri de oluşabilir. Bunlar, kuyruk güvenliğini (bkz. “Kuyruk güvenliğine ilişkin profiller” sayfa 197) ve diğer kullanıcı güvenliğini içerir (bkz. “Diğer kullanıcı güvenliği için profiller” sayfa 211). Kuyruk, bağlam ve diğer kullanıcı güvenliğinin tümü etkin olduğunda gerekli olan açma seçeneklerini ve güvenlik denetimlerini gösteren bir özet tablo için bkz. Çizelge 36 sayfa 204.

Sistem kuyruğu bağlam güvenliği

Sistem kuyruklarının çoğuna IBM MQ' un yardımcı kısımları (örneğin, kanal başlatıcı adres alanı) ve IBM MQ Console ve REST APItarafından kullanılan mqweb sunucusu) erişilir.

Bunların altında çalıştırıldığı kullanıcı kimliklerine, Çizelge 48 sayfa 215içinde gösterildiği gibi, bu kuyruklar için RACF erişimi verilmelidir.

Çizelge 48. Bağlam işlemleri için SYSTEM kuyruklarına erişim gerekli

SYSTEM kuyruğu	**Dağıtılmış kuyruğa alma için kanal başlatıcı**	**mqweb sunucusu**
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 *Komut güvenliği için profiller*

Komutlara ilişkin güvenlik denetimini etkinleştirmek için MQCMDS sınıfına profiller ekleyin. Tanıtım adları MQSC komutlarını temel alır, ancak hem MQSC, hem de PCF komutlarını denetler. Tanıtımlar bir kuyruk yöneticisine ya da kuyruk paylaşım grubuna uygulanabilir.

Komutlar için güvenlik denetimi yapmak istiyorsanız (bu nedenle, hlq.NO.CMD.CHECKS) MQCMDS sınıfına profil eklemeniz gerekir.

Aynı güvenlik tanıtımları hem MQSC, hem de PCF komutlarını denetler. Komut güvenliği denetimine ilişkin RACF tanıtımlarının adları, MQSC komutunun adlarını temel alır. Bu profiller şu formu alır:

```
hlq.verb.pkw
```

Burada `hlq` , `qmgr-name` (kuyruk yöneticisi adı) ya da `qsg-name` (kuyruk paylaşım grubu adı) olabilir, `verb` komut adının yüklem bölümüdür; örneğin, ALTER ve `pkw` nesne tipidir; örneğin, yerel bir kuyruk için QLOCAL.

Bu nedenle, CSQ1 altsisteminde ALTER QLOCAL komutunun tanıtım adı şöyledir:

```
CSQ1.ALTER.QLOCAL
```

Koruyacak daha az tanıtımınız ve bu nedenle daha az erişim listesi olacak şekilde komut kümelerini korumak için genel tanıtımları kullanabilirsiniz. Daha belirli bir tanıtımla korunmayan tüm komutlara uygulanan soysal bir tanıtım yaratmayı düşünün. Bu tanıtımı UACC (NONE) ile tanımlayın ve ALTER (ALTER) erişimini yalnızca denetimcileri içeren RACF gruplarına verin. Daha sonra, tüm DISPLAY komutları için geçerli soysal bir tanıtım yaratabilir ve bu tanıtıma geniş bir erişim verebilirsiniz. Bu uç noktalar arasında, belirli komut kümelerine erişmesi gereken kullanıcı gruplarını belirleyebilir, bu durumda bu kümeler için profiller oluşturabilir ve bu kullanıcı sınıflarını temsil eden RACF gruplarına erişim

verebilirsiniz. Kullanıcılara gerek duymadıkları komutlara erişim izni vermekten kaçının: Kullanıcıların yalnızca işleri için gerekli olan komutlara erişmeleri için en az ayrıcalık ilkesini uygulayın.

Kuyruk yöneticisi adının öneki olan bir tanıtım, o kuyruk yöneticisinde komutun kullanımını denetler. Kuyruk paylaşım grubu adı öneki eklenen bir tanıtım, komutun kuyruk paylaşım grubu içindeki tüm kuyruk yöneticilerindeki kullanımını denetler. Bu erişim, o kuyruk yöneticisinde o komut için kuyruk yöneticisi düzeyinde bir tanıtım tanımlanarak, tek bir kuyruk yöneticisinde geçersiz kılınabilir.

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesiyse ve hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ , kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar.

Kuyruk yöneticisi düzeyinde komut tanıtımları ayarlanarak, kullanıcının belirli bir kuyruk yöneticisinde komut yayınlaması kısıtlanabilir. Diğer bir seçenek olarak, her komut komutu için bir kuyruk paylaşım grubu için tek bir tanıtım tanımlayabilirsiniz ve tüm güvenlik denetimleri tek tek kuyruk yöneticileri yerine o tanıtımla ilgili olur.

Hem altsistem güvenliği hem de kuyruk paylaşım grubu güvenliği etkinse ve bir yerel tanıtım bulunamazsa, kullanıcının bir kuyruk paylaşım grubu tanıtımına erişimi olup olmadığını görmek için bir komut güvenliği denetimi gerçekleştirilir.

Bir komutu bir kuyruk paylaşım grubundaki diğer kuyruk yöneticilerine yönlendirmek için CMDSCOPE özniteliğini kullanırsanız, komutun çalıştırıldığı her kuyruk yöneticisinde güvenlik denetlenir, ancak komutun girildiği kuyruk yöneticisinde güvenliğin olması gerekmez.

Çizelge 49 sayfa 216 , her IBM MQ MQSC komutu için, komut güvenliği denetiminin gerçekleştirilmesi için gereken tanıtımları ve MQCMDS sınıfındaki her tanıtım için ilgili erişim düzeyini gösterir.

Çizelge 50 sayfa 222 , her IBM MQ PCF komutu için, komut güvenliği denetiminin gerçekleştirilmesi için gereken tanıtımları ve MQCMDS sınıfındaki her tanıtım için ilgili erişim düzeyini gösterir.

Çizelge 49. MQSC komutları, tanıtımları ve erişim düzeyleri

Komut	**MQCMDS için komut profili**	**MQCMDS için erişim düzeyi**	**MQADMIN ya da MXADMIN için komut kaynağı profili**	**MQADMIN ya da MXADMIN için erişim düzeyi**
AUTHINFO DEğIşTIR	hlq.ALTER.AUTHINFO	Çeviri	hlq.AUTHINFO.resourcename	Çeviri
BUFFPOOL DEğIşTIR	hlq.ALTER.BUFFPOOL	Çeviri	Çek yok	-
ALTER CFSTRUCT (CFSTRUCT)	hlq.ALTER.CFSTRUCT	Çeviri	Çek yok	-
KANAL DEğIşTIR	hlq.ALTER.CHANNEL	Çeviri	hlq.CHANNEL.channel	Çeviri
ADı LISTESI	hlq.ALTER.NAMELIST	Çeviri	hlq.NAMELIST.namelist	Çeviri
İşLEMI DEğI	hlq.ALTER.PROCESS	Çeviri	hlq.PROCESS.process	Çeviri
ALTER PSID (PSID)	hlq.ALTER.PSID	Çeviri	Çek yok	-
QALIAS DEğIşTIR	hlq.ALTER.QALIAS	Çeviri	hlq.QUEUE.queue	Çeviri
ALTER QLOCAL“5” sayfa 222	hlq.ALTER.QLOCAL	Çeviri	hlq.QUEUE.queue	Çeviri
ALTER QMGR	hlq.ALTER.QMGR	Çeviri	Çek yok	-
QMODELE DEğIşTIR“5” sayfa 222	hlq.ALTER.QMODEL	Çeviri	hlq.QUEUE.queue	Çeviri

Çizelge 49. MQSC komutları, tanıtımları ve erişim düzeyleri (devamı var)

Komut	MQCMDS için komut profili	MQCMDS için erişim düzeyi	MQADMIN ya da MXADMIN için komut kaynağı profili	MQADMIN ya da MXADMIN için erişim düzeyi
QUZAK DEğIşTIR	hlq.ALTER.QREMOTE	Çeviri	hlq.QUEUE.queue	Çeviri
GüVENLIK DEğ	hlq.ALTER.SECURITY	Çeviri	Çek yok	-
ALTER SMDS (SMDS)	hlq.ALTER.SMDS	Çeviri	Çek yok	-
STGClass DEğIşTIR	hlq.ALTER.STGCLASS	Çeviri	Çek yok	-
ALTER SUB	hlq.ALTER.SUB	Çeviri	Çek yok	-
KONUYU DEğIşTIR	hlq.ALTER.TOPIC	Çeviri	hlq.TOPIC.topic	Çeviri
İZLEME	hlq.ALTER.TRACE	Çeviri	Çek yok	-
GüNLüğü ARşIV	hlq.ARCHIVE.LOG	CONTROL	Çek yok	-
YEDEK CFSTRUCT	hlq.BACKUP.CFSTRUCT	CONTROL	Çek yok	-
QLOCAL TEMIZLE	hlq.CLEAR.QLOCAL	Çeviri	hlq.QUEUE.queue	Çeviri
“3” sayfa 222 TOPICSTR ' I TEMIZLE	hlq.CLEAR.TOPICSTR	Çeviri	hlq.TOPIC.topic	Çeviri
AUTHINFO TANıMLAYıN	hlq.DEFINE.AUTHINFO	Çeviri	hlq.AUTHINFO.resourcename	Çeviri
ARABELLEK HAVUZU TANıMLAYıN	hlq.DEFINE.BUFFPOOL	Çeviri	Çek yok	-
CFSTRUCT 'U TANıMLAYıN	hlq.DEFINE.CFSTRUCT	Çeviri	Çek yok	-
KANAL TANıMLAYıN	hlq.DEFINE.CHANNEL	Çeviri	hlq.CHANNEL.channel	Çeviri
GüNLüğü TANıM	hlq.DEFINE.LOG	Çeviri	Çek yok	-
MAXSMSGS TANıMLAYıN	hlq.DEFINE.MAXSMSGS	Çeviri	Çek yok	-
AD LISTESI TANıMLAYıN	hlq.DEFINE.NAMELIST	Çeviri	hlq.NAMELIST.namelist	Çeviri
SüREç TANıMLE	hlq.DEFINE.PROCESS	Çeviri	hlq.PROCESS.process	Çeviri
PSID TANıMLAYıN	hlq.DEFINE.PSID	Çeviri	Çek yok	-
QALIAS TANıMLAYıN	hlq.DEFINE.QALIAS	Çeviri	hlq.QUEUE.queue	Çeviri
QLOCAL TANıMLAYıN“5” sayfa 222	hlq.DEFINE.QLOCAL	Çeviri	hlq.QUEUE.queue	Çeviri
QMODELE TANıMLAYıN“5” sayfa 222	hlq.DEFINE.QMODEL	Çeviri	hlq.QUEUE.queue	Çeviri
QUZAK TANıMLAYıN	hlq.DEFINE.QREMOTE	Çeviri	hlq.QUEUE.queue	Çeviri
STGClass TANıMLAYıN	hlq.DEFINE.STGCLASS	Çeviri	Çek yok	-

Çizelge 49. MQSC komutları, tanıtımları ve erişim düzeyleri (devamı var)

Komut	MQCMDS için komut profili	MQCMDS için erişim düzeyi	MQADMIN ya da MXADMIN için komut kaynağı profili	MQADMIN ya da MXADMIN için erişim düzeyi
ALT öğEYI TAN	hlq.DEFINE.SUB	Çeviri	Çek yok	-
KONUYU TANıMLAYıN	hlq.DEFINE.TOPIC	Çeviri	hlq.TOPIC.topic	Çeviri
AUTHINFO ÖĞESINI SIL	hlq.DELETE.AUTHINFO	Çeviri	hlq.AUTHINFO.resourcename	Çeviri
ARABELLEK HAVUZUNU SIL	hlq.DELETE.BUFFPOOL	Çeviri	Çek yok	-
CFSTRUCT 'U SIL	hlq.DELETE.CFSTRUCT	Çeviri	Çek yok	-
BAğLANTı SIL	hlq.DELETE.CHANNEL	Çeviri	hlq.CHANNEL.channel	Çeviri
ADı SIL	hlq.DELETE.NAMELIST	Çeviri	hlq.NAMELIST.namelist	Çeviri
Süreci Sil	hlq.DELETE.PROCESS	Çeviri	hlq.PROCESS.process	Çeviri
PSID ' YI SIL	hlq.DELETE.PSID	Çeviri	Çek yok	-
QDIğER ADı SIL	hlq.DELETE.QALIAS	Çeviri	hlq.QUEUE.queue	Çeviri
QLOCAL ÖĞESINI SIL	hlq.DELETE.QLOCAL	Çeviri	hlq.QUEUE.queue	Çeviri
QMODEL SIL	hlq.DELETE.QMODEL	Çeviri	hlq.QUEUE.queue	Çeviri
QREMOTE ÖĞESINI SIL	hlq.DELETE.QREMOTE	Çeviri	hlq.QUEUE.queue	Çeviri
STGCLASS DEğERINI SIL	hlq.DELETE.STGCLASS	Çeviri	Çek yok	-
ALT ÖĞEYI SIL	hlq.DELETE.SUB	Çeviri	Çek yok	-
KONUYU SIL	hlq.DELETE.TOPIC	Çeviri	hlq.TOPIC.topic	Çeviri
ARşIV GöRüNTüLE “1” sayfa 221	hlq.DISPLAY.ARCHIVE	READ	Çek yok	-
AUTHINFO BILGILERINI GöRüNTüLE	hlq.DISPLAY.AUTHINFO	READ	Çek yok	-
CFSTATUS DURUMUNU GöRüNTüLE	hlq.DISPLAY.CFSTATUS	READ	Çek yok	-
CFSTRUCT 'U GöRüNTüLE	hlq.DISPLAY.CFSTRUCT	READ	Çek yok	-
KANAL GöRüNTüLE	hlq.DISPLAY.CHANNEL	READ	Çek yok	-
CHINIT GöRüNTüLE	hlq.DISPLAY.CHINIT	READ	Çek yok	-
CHLAUTH 'U GöRüNTüLE	hlq.DISPLAY.CHLAUTH	READ	Çek yok	-

Çizelge 49. MQSC komutları, tanıtımları ve erişim düzeyleri (devamı var)

Komut	MQCMDS için komut profili	MQCMDS için erişim düzeyi	MQADMIN ya da MXADMIN için komut kaynağı profili	MQADMIN ya da MXADMIN için erişim düzeyi
CHSTATUS DURUMUNU GöRüNTüLE	hlq.DISPLAY.CHSTATUS	READ	Çek yok	-
CLUSQMGR ' YI GöRüNTüLE	hlq.DISPLAY.CLUSQMGR	READ	Çek yok	-
CMDSERV 'I GöRüNTüLE	hlq.DISPLAY.CMDSERV	READ	Çek yok	-
“1” sayfa 221 GöRüNTüLEYIN	hlq.DISPLAY.CONN	READ	Çek yok	-
GRUBU GöRüNTüLE	hlq.DISPLAY.GROUP	READ	Çek yok	-
DISPLAY LOG “1” sayfa 221	hlq.DISPLAY.LOG	READ	Çek yok	-
MAXSMSGS DEğERINI GöRüNTüLE	hlq.DISPLAY.MAXSMSGS	READ	Çek yok	-
ADı LISTESI GöRüNTüLE	hlq.DISPLAY.NAMELIST	READ	Çek yok	-
SüRECI GöRüNTü	hlq.DISPLAY.PROCESS	READ	Çek yok	-
PUBSUB 'I GöRüNTüLE	hlq.DISPLAY.PUBSUB	READ	Çek yok	-
QDIğER ADıNı Gö	hlq.DISPLAY.QALIAS	READ	Çek yok	-
QKüMESINI GöRüNTüLE	hlq.DISPLAY.QCLUSTER	READ	Çek yok	-
QLOCAL GöRüNTüLE	hlq.DISPLAY.QLOCAL	READ	Çek yok	-
QMGR ' YI GöRüNTüLE	hlq.DISPLAY.QMGR	READ	Çek yok	-
QMODEL MODELINI GöRüNTüLE	hlq.DISPLAY.QMODEL	READ	Çek yok	-
QUZAK GöRüNTüLE	hlq.DISPLAY.QREMOTE	READ	Çek yok	-
QSTATUS DURUMUNU GöRüNTüLE	hlq.DISPLAY.QSTATUS	READ	Çek yok	-
KUYRUK GöRüNTüLE	hlq.DISPLAY.QUEUE	READ	Çek yok	-
SBSTATUS DURUMUNU GöRüNTüLE	hlq.DISPLAY.SBSTATUS	READ	Çek yok	-
SMDS GöRüNTüLE	hlq.DISPLAY.SMDS	READ	Çek yok	-
SMDSCONN 'YI GöRüNTüLE	hlq.DISPLAY.SMDSCONN	READ	Çek yok	-
ALT öğEYI Gö	hlq.DISPLAY.SUB	READ	Çek yok	-

Çizelge 49. MQSC komutları, tanıtımları ve erişim düzeyleri (devamı var)

Komut	MQCMDS için komut profili	MQCMDS için erişim düzeyi	MQADMIN ya da MXADMIN için komut kaynağı profili	MQADMIN ya da MXADMIN için erişim düzeyi
GüVENLIK GÖRÜNTÜLE	hlq.DISPLAY.SECURITY	READ	Çek yok	-
STGClass GöRüNTüLE	hlq.DISPLAY.STGCLASS	READ	Çek yok	-
SISTEMI GöRüNTüLE “1” sayfa 221	hlq.DISPLAY.SYSTEM	READ	Çek yok	-
İş PARçACığıN	hlq.DISPLAY.THREAD	READ	Çek yok	-
TPSTATüYü GöRüNTüLE	hlq.DISPLAY.TPSTATUS	READ	Çek yok	-
KONU GöRüNTüLE	hlq.DISPLAY.TOPIC	READ	Çek yok	-
TPSTATüYü GöRüNTüLE	hlq.DISPLAY.TPSTATUS	READ	Çek yok	-
IZLEMEYI GöRüNTüLE	hlq.DISPLAY.TRACE	READ	Çek yok	-
KULLANıM GöRüNTüLE “1” sayfa 221	hlq.DISPLAY.USAGE	READ	Çek yok	-
QLOCAL DEğERINI TAşıR	hlq.MOVE.QLOCAL	Çeviri	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	Çeviri
PING KANALı	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
BDS LERI KURTAR	hlq.RECOVER.BSDS	CONTROL	Çek yok	-
CFSTRUCT 'U KURTARıN	hlq.RECOVER.CFSTRUCT	CONTROL	Çek yok	-
KüMEYI YENILE	hlq.REFRESH.CLUSTER	Çeviri	Çek yok	-
QMGR ' YI YENILE	hlq.REFRESH.QMGR	Çeviri	Çek yok	-
Güvenliği yenileme	hlq.REFRESH.SECURITY	Çeviri	Çek yok	-
CFSTRUCT 'U İLK DURUMUNA GETIR	hlq.RESET.CFSTRUCT	CONTROL	Çek yok	-
KANALI ILK DURUMUNA GETIR	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
KüMEYI SıFı	hlq.RESET.CLUSTER	CONTROL	Çek yok	-
QMGR ' YI SıFıR	hlq.RESET.QMGR	CONTROL	Çek yok	-
QSTATS DEğIşTIRMESI	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL
SMDS ' I SıFıRLA	hlq.RESET.SMDS	CONTROL	Çek yok	-
TPUANı SıFıR	hlq.RESET.TPIPE	CONTROL	Çek yok	-
KANAL ÇöZüMLE	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
ŞüPHELI çöZüMLE	hlq.RESOLVE.INDOUBT	CONTROL	Çek yok	-

Çizelge 49. MQSC komutları, tanıtımları ve erişim düzeyleri (devamı var)

Komut	MQCMDS için komut profili	MQCMDS için erişim düzeyi	MQADMIN ya da MXADMIN için komut kaynağı profili	MQADMIN ya da MXADMIN için erişim düzeyi
QMGR ' YI SüRDüR	hlq.RESUME.QMGR	CONTROL	Çek yok	-
RVERIFY GüVENLIğI	hlq.RVERIFY.SECURITY	Çeviri	Çek yok	-
ARşIVI AYARLA	hlq.SET.ARCHIVE	CONTROL	Çek yok	-
CHLAUTH AYARLA	hlq.SET.CHLAUTH	CONTROL	Çek yok	-
GÜNLÜĞÜ AYARLA	hlq.SET.LOG	CONTROL	Çek yok	-
SISTEMI AYARLA	hlq.SET.SYSTEM	CONTROL	Çek yok	-
BAşLANGıç KANALI	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
CHINIT “4” sayfa 222 ' I BAşLATıN	hlq.START.CHINIT	CONTROL	Çek yok	-
CMDSERV ' YI BAşLAT	hlq.START.CMDSERV	CONTROL	Çek yok	-
DINLEYICIYI BAşLAYıN	hlq.START.LISTENER	CONTROL	Çek yok	-
QMGR ' YI BAşLAT	Yok“2” sayfa 221	-	-	-
SMDSCONN ' U BAşLAT	hlq.START.SMDSCONN	CONTROL	Çek yok	-
IZLEMEYI BAŞLAT	hlq.START.TRACE	CONTROL	Çek yok	-
BAğLANTı DURDUR	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
CHINIT 'I BıRAKMAK	hlq.STOP.CHINIT	CONTROL	Çek yok	-
DURDUR CMDSERV	hlq.STOP.CMDSERV	CONTROL	Çek yok	-
DINLEYICIYI DURDUR	hlq.STOP.LISTENER	CONTROL	Çek yok	-
QMGR ' YI DURDUR	hlq.STOP.QMGR	CONTROL	Çek yok	-
SMDSCONN ' U DURDUR	hlq.STOP.SMDSCONN	CONTROL	Çek yok	-
İZLEMEYI DURDUR	hlq.STOP.TRACE	CONTROL	Çek yok	-
QMGR ' YI ASKıYA AL	hlq.SUSPEND.QMGR	CONTROL	Çek yok	-

Notlar:

1. Bu komutlar, kuyruk yöneticisi tarafından dahili olarak verilebilir; bu durumlarda yetki denetlenmez.
2. IBM MQ , START QMGR komutunu veren kullanıcının yetkisini denetlemez. Ancak, START QMGR komutunun sonucu olarak verilen START xxxxMSTR komutuna erişimi denetlemek için RACFkomutunu ya da diğer güvenlik olanaklarınızı kullanabilirsiniz.

   Bu, RACF işletmen komutları (OPERCMDS) sınıfındaki MVS.START.STC.xxxxMSTR tanıtımına erişimi denetleyerek yapılır. Bu yordamın ayrıntıları için *z/OS MVS Planning: Operations*belgesinde User access to RACF OPERCMDS class başlıklı konuya bakın. Bu tekniği kullanırsanız ve yetkisiz bir kullanıcı kuyruk yöneticisini başlatmayı denerse, bu yöntem 00F30216neden koduyla sona erer.

3. **hlq.TOPIC.topic** kaynağı, TOPICSTR ' den türetilen Konu nesnesine gönderme yapar. Daha fazla ayrıntı için bkz. “Yayınlama/abone olma güvenliği” sayfa 496
4. IBM MQ for z/OSiçinde, kaynak adı MVS.START.STC.CSQ1CHIN ek bir JOBNAME niteleyicisi eklendi. Bu, kanal başlatıcısı başlatılırken sorunlara neden olabilir.

   Sorunu çözmek için MVS.START.STCyerine bakın. *ssid* CHIN, MVS.START.STCadlı bir kaynağa ilişkin tanıtımla birlikte. *ssid* CHIN **.*** ya da MVS.START.STC. *SıNıRLı* CHIN. *ssid* CHIN; burada *ssid* , kuyruk yöneticisinin altsistem tanıtıcısıdır. Bu, RACF UPDATE yetkisi gerektirir. Daha fazla ayrıntı için *z/OS MVS Planning: Operations*belgesinde MVS Commands, RACF Access Authorities, and Resource Names başlıklı konuya bakın.

   ssid MSTR için START, JOBNAME= parametresini içermez. Tutarlılık için MVS.START.STC.ssidMSTR tanıtımını MVS.START.STC.ssidMSTRolarak güncellemek isteyebilirsiniz. *.
5. V 9.3.0 STREAMQ kuyruk özniteliğinin boş olmayan bir değere ayarlanması, `hlq.ALTER.streamQ`için MQADMIN ya da MXADMIN için ALTER erişim düzeyini de gerektirir.

Çizelge 50. PCF komutları, tanıtımları ve erişim düzeyleri

Komut	**MQCMDS için komut profili**	**MQCMDS için erişim düzeyi**	**MQADMIN ya da MXADMIN için komut kaynağı profili**	**MQADMIN ya da MXADMIN için erişim düzeyi**
CF Yapısını Yedekle	hlq.BACKUP.CFSTRUCT	CONTROL	Çek yok	-
Kimlik Doğrulama Bilgileri Nesnesini Değiştir	hlq.ALTER.AUTHINFO	Çeviri	hlq.AUTHINFO.resourcename	Çeviri
CF Yapısını Değiştir	hlq.ALTER.CFSTRUCT	Çeviri	Çek yok	-
Kanalı Değiştir	hlq.ALTER.CHANNEL	Çeviri	hlq.CHANNEL.channel	Çeviri
Ad melist değerini değiştir	hlq.ALTER.NAMELIST	Çeviri	hlq.NAMELIST.namelist	Çeviri
Süreci Değiştir	hlq.ALTER.PROCESS	Çeviri	hlq.PROCESS.process	Çeviri
Kuyruğu Değiştir“2” sayfa 226	hlq.ALTER.QUEUE	Çeviri	hlq.QUEUE.queue	Çeviri
Kuyruk Yöneticisini Değiştir	hlq.ALTER.QMGR	Çeviri	Çek yok	-
Güvenliği Değiştir	hlq.ALTER.SECURITY	Çeviri	Çek yok	-
SMDS ' yi Değiştir	hlq.ALTER.SMDS	Çeviri	Çek yok	-
Depolama Sınıfını Değiştir	hlq.ALTER.STGCLASS	Çeviri	Çek yok	-
Aboneliği Değiştir	hlq.ALTER.SUB	Çeviri	Çek yok	-
Konuyu Değiştir	hlq.ALTER.TOPIC	Çeviri	hlq.TOPIC.topic	Çeviri
Kuyruğu Temizle	hlq.CLEAR.QLOCAL	Çeviri	hlq.QUEUE.queue	Çeviri
“1” sayfa 226 Konu Dizgisini Temizle	hlq.CLEAR.TOPICSTR	Çeviri	hlq.TOPIC.topic	Çeviri
Kimlik Doğrulama Bilgileri Nesnesini Kopyala	hlq.DEFINE.AUTHINFO	Çeviri	hlq.AUTHINFO.resourcename	Çeviri
CF Yapısını Kopyala	hlq.DEFINE.CFSTRUCT	Çeviri	Çek yok	-
Kanalı Kopyala	hlq.DEFINE.CHANNEL	Çeviri	hlq.CHANNEL.channel	Çeviri

Çizelge 50. PCF komutları, tanıtımları ve erişim düzeyleri (devamı var)

Komut	MQCMDS için komut profili	MQCMDS için erişim düzeyi	MQADMIN ya da MXADMIN için komut kaynağı profili	MQADMIN ya da MXADMIN için erişim düzeyi
Namelist 'i Kopyala	hlq.DEFINE.NAMELIST	Çeviri	hlq.NAMELIST.namelist	Çeviri
Süreci Kopyala	hlq.DEFINE.PROCESS	Çeviri	hlq.PROCESS.process	Çeviri
Kuyruğu Kopyala	hlq.DEFINE.QUEUE	Çeviri	hlq.QUEUE.queue	Çeviri
Aboneliği Kopyala	hlq.DEFINE.SUB	Çeviri	Çek yok	-
Depolama Sınıfını Kopyala	hlq.DEFINE.STGCLASS	Çeviri	Çek yok	-
Konuyu Kopyala	hlq.DEFINE.TOPIC	Çeviri	hlq.TOPIC.topic	Çeviri
Kimlik Doğrulama Bilgileri Nesnesi Oluştur	hlq.DEFINE.AUTHINFO	Çeviri	hlq.AUTHINFO.resourcename	Çeviri
CF Yapısı Oluştur	hlq.DEFINE.CFSTRUCT	Çeviri	Çek yok	-
Kanal Oluştur	hlq.DEFINE.CHANNEL	Çeviri	hlq.CHANNEL.channel	Çeviri
Ad listesi oluştur	hlq.DEFINE.NAMELIST	Çeviri	hlq.NAMELIST.namelist	Çeviri
Süreç Yarat	hlq.DEFINE.PROCESS	Çeviri	hlq.PROCESS.process	Çeviri
Kuyruk Yarat“2” sayfa 226	hlq.DEFINE.QUEUE	Çeviri	hlq.QUEUE.queue	Çeviri
Depolama Sınıfı Oluştur	hlq.DEFINE.STGCLASS	Çeviri	Çek yok	-
Abonelik Yarat	hlq.DEFINE.SUB	Çeviri	Çek yok	-
Konu Oluştur	hlq.DEFINE.TOPIC	Çeviri	hlq.TOPIC.topic	Çeviri
Kimlik Doğrulama Bilgileri Nesnesini Sil	hlq.DELETE.AUTHINFO	Çeviri	hlq.AUTHINFO.resourcename	Çeviri
CF Yapısını Sil	hlq.DELETE.CFSTRUCT	Çeviri	Çek yok	-
Kanalı Sil	hlq.DELETE.CHANNEL	Çeviri	hlq.CHANNEL.channel	Çeviri
Ad alanını sil	hlq.DELETE.NAMELIST	Çeviri	hlq.NAMELIST.namelist	Çeviri
Süreci Sil	hlq.DELETE.PROCESS	Çeviri	hlq.PROCESS.process	Çeviri
Kuyruğu sil	hlq.DELETE.QUEUE	Çeviri	hlq.QUEUE.queue	Çeviri
Depolama Sınıfını Sil	hlq.DELETE.STGCLASS	Çeviri	Çek yok	-
Aboneliği Sil	hlq.DELETE.SUB	Çeviri	Çek yok	-
Konuyu Sil	hlq.DELETE.TOPIC	Çeviri	hlq.TOPIC.topic	Çeviri
Sorgulama Arşivi	hlq.DISPLAY.ARCHIVE	READ	Çek yok	-
Kimlik Doğrulama Bilgileri Nesnesini Sor	hlq.DISPLAY.AUTHINFO	READ	Çek yok	-
Kimlik Doğrulama Bilgileri Nesne Adlarını Sor	hlq.DISPLAY.AUTHINFO	READ	Çek yok	-
CF Yapısını Sorgele	hlq.DISPLAY.CFSTRUCT	READ	Çek yok	-
CF Yapısı Adlarını Sor	hlq.DISPLAY.CFSTRUCT	READ	Çek yok	-
CF Yapısı Durumunu Sor	hlq.DISPLAY.CFSTATUS	READ	Çek yok	-

Çizelge 50. PCF komutları, tanıtımları ve erişim düzeyleri (devamı var)

Komut	MQCMDS için komut profili	MQCMDS için erişim düzeyi	MQADMIN ya da MXADMIN için komut kaynağı profili	MQADMIN ya da MXADMIN için erişim düzeyi
Sorma Kanalı	hlq.DISPLAY.CHANNEL	READ	Çek yok	-
Sorma Kanalı Kimlik Doğrulama Kayıtları	hlq.DISPLAY.CHLAUTH	READ	Çek yok	-
Sorma Kanalı Başlatıcısı	hlq.DISPLAY.CHINIT	READ	Çek yok	-
Kanal Adlarını Sor	hlq.DISPLAY.CHANNEL	READ	Çek yok	-
Kanal Durumunu Sor	hlq.DISPLAY.CHSTATUS	READ	Çek yok	-
Küme Kuyruğu Yöneticisini Sor	hlq.DISPLAY.CLUSQMGR	READ	Çek yok	-
Bağlantıyı sorgularken	hlq.DISPLAY.CONNPCF	READ	Çek yok	-
Sorma Grubu	hlq.DISPLAY.GROUP	READ	Çek yok	-
Sorma Günlüğü	hlq.DISPLAY.LOG	READ	Çek yok	-
Namelist 'i sor	hlq.DISPLAY.NAMELIST	READ	Çek yok	-
Ad Listesi Adlarını Sor	hlq.DISPLAY.NAMELIST	READ	Çek yok	-
Sorma Süreci	hlq.DISPLAY.PROCESS	READ	Çek yok	-
Süreç Adlarını Sor	hlq.DISPLAY.PROCESS	READ	Çek yok	-
Sorma Pub/Alt Durumu	hlq.DISPLAY.PUBSUB	READ	Çek yok	-
Sorma Kuyruğu	hlq.DISPLAY.QUEUE	READ	Çek yok	-
Sorma Kuyruğu Yöneticisi	hlq.DISPLAY.QMGR	READ	Çek yok	-
Kuyruk Adlarını Sor	hlq.DISPLAY.QUEUE	READ	Çek yok	-
Sorma Kuyruğu Durumu	hlq.DISPLAY.QSTATUS	READ	Çek yok	-
Güvenliği sorgularken	hlq.DISPLAY.SECURITY	READ	Çek yok	-
SMDS ' yi sorgulamak	hlq.DISPLAY.SMDS	READ	Çek yok	-
SMDSCONN ' ı sor	hlq.DISPLAY.SMDSCONN	READ	Çek yok	-
Sorma Depolama Sınıfı	hlq.DISPLAY.STGCLASS	READ	Çek yok	-
Depolama Sınıfı Adlarını Sor	hlq.DISPLAY.STGCLASS	READ	Çek yok	-
Aboneliği Sorgulamak	hlq.INQUIRE.SUB	READ	Çek yok	-
Abonelik Durumunu Sor	hlq.INQUIRE.SBSTATUS	READ	Çek yok	-
Sorma Sistemi	hlq.DISPLAY.SYSTEM	READ	Çek yok	-
Konuyu Sor	hlq.DISPLAY.TOPIC	READ	Çek yok	-
Konu Adlarını Sor	hlq.DISPLAY.TOPIC	READ	Çek yok	-
Konu Durumu Sorgusu	hlq.DISPLAY.TPSTATUS	READ	Çek yok	-
Sorma Kullanımı	hlq.DISPLAY.USAGE	READ	Çek yok	-
Kuyruğu Taşı	hlq.MOVE.QLOCAL	Çeviri	hlq.QUEUE.from-queue hlq.QUEUE.to-queue	Çeviri

Çizelge 50. PCF komutları, tanıtımları ve erişim düzeyleri (devamı var)

Komut	MQCMDS için komut profili	MQCMDS için erişim düzeyi	MQADMIN ya da MXADMIN için komut kaynağı profili	MQADMIN ya da MXADMIN için erişim düzeyi
Ping Kanalı	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
CF Yapısını Kurtar	hlq.RECOVER.CFSTRUCT	CONTROL	Çek yok	-
Kümeyi Yenile	hlq.REFRESH.CLUSTER	Çeviri	Çek yok	-
Kuyruk Yöneticisini Yenile	hlq.REFRESH.QMGR	Çeviri	Çek yok	-
Güvenliği Yenile	hlq.REFRESH.SECURITY	Çeviri	Çek yok	-
CF Yapısını Sıfırla	hlq.RESET.CFSTRUCT	CONTROL	Çek yok	-
Kanalı Sıfırla	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Kümeyi Sıfırla	hlq.RESET.CLUSTER	CONTROL	Çek yok	-
Kuyruk Yöneticisini İlk Durumuna Getir	hlq.RESET.QMGR	CONTROL	Çek yok	-
Kuyruk İstatistiklerini Sıfırla	hlq.RESET.QSTATS	CONTROL	hlq.QUEUE.queue	CONTROL
SMDS ' yi Sıfırla	hlq.RESET.SMDS	CONTROL	Çek yok	-
Kanalı Çözümle	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Kuyruk Yöneticisini Sürdür	hlq.RESUME.QMGR	CONTROL	Çek yok	-
Kuyruk Yöneticisi Kümesini Sürdür	hlq.RESUME.QMGR	CONTROL	Çek yok	-
Güvenliği Yeniden Doğrula	hlq.RVERIFY.SECURITY	Çeviri	Çek yok	-
Arşivi Ayarla	hlq.SET.ARCHIVE	CONTROL	Çek yok	-
Kanal Kimlik Doğrulama Kaydını Ayarla	hlq.SET.CHLAUTH	CONTROL	Çek yok	-
Günlüğü Ayarla	hlq.SET.LOG	CONTROL	Çek yok	-
Sistemi Ayarla	hlq.SET.SYSTEM	CONTROL	Çek yok	-
Kanalı Başlat	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Kanal Başlatıcıyı Başlat	hlq.START.CHINIT	CONTROL	Çek yok	-
Kanal Dinleyicisini Başlat	hlq.START.LISTENER	CONTROL	Çek yok	-
SMDS Bağlantısını Başlat	hlq.START.SMDSCONN	CONTROL	Çek yok	-
Kanalı Durdur	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Kanal Başlatıcıyı Durdur	hlq.STOP.CHINIT	CONTROL	Çek yok	-
Kanal Dinleyicisini Durdur	hlq.STOP.LISTENER	CONTROL	Çek yok	-
SMDS Bağlantısını Durdur	hlq.STOP.SMDSCONN	CONTROL	Çek yok	-
Kuyruk Yöneticisini Askıya Al	hlq.SUSPEND.QMGR	CONTROL	Çek yok	-

Çizelge 50. PCF komutları, tanıtımları ve erişim düzeyleri (devamı var)				
Komut	**MQCMDS için komut profili**	**MQCMDS için erişim düzeyi**	**MQADMIN ya da MXADMIN için komut kaynağı profili**	**MQADMIN ya da MXADMIN için erişim düzeyi**
Kuyruk Yöneticisi Kümesini Askıya Al	hlq.SUSPEND.QMGR	CONTROL	Çek yok	-

Notlar:

1. **`hlq.TOPIC.topic`** kaynağı, TOPICSTR ' den türetilen Konu nesnesine gönderme yapar. Daha fazla ayrıntı için bkz. “Yayınlama/abone olma güvenliği” sayfa 496
2. V 9.3.0 STREAMQ kuyruk özniteliğinin boş olmayan bir değere ayarlanması, `hlq.ALTER.streamQ`için MQADMIN ya da MXADMIN için ALTER erişim düzeyini de gerektirir.

IBM MQ Consolekullanırken, gereken IBM MQ PCF tanıtımlarına ilişkin ayrıntılar için bkz. “IBM MQ Console -gerekli komut güvenliği tanıtımları” sayfa 226 .

z/OS *IBM MQ Console -gerekli komut güvenliği tanıtımları*

IBM MQ Console içinde `MQWebAdmin`ya da `MQWebAdminRO`içindeki bir kullanıcı tarafından gerçekleştirilen işlemler, görevi başlatan mqweb sunucusunun güvenlik bağlamı altında gerçekleşir. IBM MQ Consolekullanmak istiyorsanız, mqweb sunucusu tarafından başlatılan görev kullanıcı kimliğinin bazı PCF komutlarını yayınlamak için yetkilendirilmesi gerekir.

Çizelge 51 sayfa 226 içinde, her IBM MQ PCF komutu için gereken komut güvenliği tanıtımları ve IBM MQ Consoleiçin gereken MQCMDS sınıfındaki her tanıtım için ilgili erişim düzeyi gösterilir.

Çizelge 51. IBM MQ Console PCF komutları, tanıtımları ve erişim düzeyleri				
Komut	**MQCMDS için komut profili**	**MQCMDS için erişim düzeyi**	**MQADMIN ya da MXADMIN için komut kaynağı profili**	**MQADMIN ya da MXADMIN için erişim düzeyi**
Kimlik Doğrulama Bilgileri Nesnesini Değiştir	hlq.ALTER.AUTHINFO	Çeviri	hlq.AUTHINFO.resourcename	Çeviri
Kanalı Değiştir	hlq.ALTER.CHANNEL	Çeviri	hlq.CHANNEL.channel	Çeviri
Kuyruğu Değiştir	hlq.ALTER.QUEUE	Çeviri	hlq.QUEUE.queue	Çeviri
Kuyruk Yöneticisini Değiştir	hlq.ALTER.QMGR	Çeviri	Çek yok	-
Konuyu Değiştir	hlq.ALTER.TOPIC	Çeviri	hlq.TOPIC.topic	Çeviri
Kuyruğu Temizle	hlq.CLEAR.QLOCAL	Çeviri	hlq.QUEUE.queue	Çeviri
Kimlik Doğrulama Bilgileri Nesnesi Oluştur	hlq.DEFINE.AUTHINFO	Çeviri	hlq.AUTHINFO.resourcename	Çeviri
Kanal Oluştur	hlq.DEFINE.CHANNEL	Çeviri	hlq.CHANNEL.channel	Çeviri
Kuyruk Yarat	hlq.DEFINE.QUEUE	Çeviri	hlq.QUEUE.queue	Çeviri
Abonelik Yarat	hlq.DEFINE.SUB	Çeviri	Çek yok	-
Konu Oluştur	hlq.DEFINE.TOPIC	Çeviri	hlq.TOPIC.topic	Çeviri
Kimlik Doğrulama Bilgileri Nesnesini Sil	hlq.DELETE.AUTHINFO	Çeviri	hlq.AUTHINFO.resourcename	Çeviri

Çizelge 51. IBM MQ Console PCF komutları, tanıtımları ve erişim düzeyleri (devamı var)

Komut	MQCMDS için komut profili	MQCMDS için erişim düzeyi	MQADMIN ya da MXADMIN için komut kaynağı profili	MQADMIN ya da MXADMIN için erişim düzeyi
Kanalı Sil	hlq.DELETE.CHANNEL	Çeviri	hlq.CHANNEL.channel	Çeviri
Kuyruğu sil	hlq.DELETE.QUEUE	Çeviri	hlq.QUEUE.queue	Çeviri
Aboneliği Sil	hlq.DELETE.SUB	Çeviri	Çek yok	-
Konuyu Sil	hlq.DELETE.TOPIC	Çeviri	hlq.TOPIC.topic	Çeviri
Kimlik Doğrulama Bilgileri Nesnesini Sor	hlq.DISPLAY.AUTHINFO	READ	Çek yok	-
Kimlik Doğrulama Bilgileri Nesne Adlarını Sor	hlq.DISPLAY.AUTHINFO	READ	Çek yok	-
Sorma Kanalı	hlq.DISPLAY.CHANNEL	READ	Çek yok	-
Sorma Kanalı Kimlik Doğrulama Kayıtları	hlq.DISPLAY.CHLAUTH	READ	Çek yok	-
Sorma Kanalı Başlatıcısı	hlq.DISPLAY.CHINIT	READ	Çek yok	-
Kanal Adlarını Sor	hlq.DISPLAY.CHANNEL	READ	Çek yok	-
Kanal Durumunu Sor	hlq.DISPLAY.CHSTATUS	READ	Çek yok	-
Sorma Kuyruğu	hlq.DISPLAY.QUEUE	READ	Çek yok	-
Sorma Kuyruğu Yöneticisi	hlq.DISPLAY.QMGR	READ	Çek yok	-
Kuyruk Adlarını Sor	hlq.DISPLAY.QUEUE	READ	Çek yok	-
Sorma Kuyruğu Durumu	hlq.DISPLAY.QSTATUS	READ	Çek yok	-
Aboneliği Sorgulamak	hlq.INQUIRE.SUB	READ	Çek yok	-
Abonelik Durumunu Sor	hlq.INQUIRE.SBSTATUS	READ	Çek yok	-
Konuyu Sor	hlq.DISPLAY.TOPIC	READ	Çek yok	-
Konu Adlarını Sor	hlq.DISPLAY.TOPIC	READ	Çek yok	-
Konu Durumu Sorgusu	hlq.DISPLAY.TPSTATUS	READ	Çek yok	-
Ping Kanalı	hlq.PING.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Kümeyi Yenile	hlq.REFRESH.CLUSTER	Çeviri	Çek yok	-
Güvenliği Yenile	hlq.REFRESH.SECURITY	Çeviri	Çek yok	-
Kanalı Sıfırla	hlq.RESET.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Kanalı Çözümle	hlq.RESOLVE.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Kanal Kimlik Doğrulama Kaydını Ayarla	hlq.SET.CHLAUTH	CONTROL	Çek yok	-
Kanalı Başlat	hlq.START.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL
Kanalı Durdur	hlq.STOP.CHANNEL	CONTROL	hlq.CHANNEL.channel	CONTROL

z/OS *Komut kaynak güvenliği için profiller*

Komutlarla ilişkilendirilmiş kaynaklar için güvenlik denetimi yapmak istediğiniz için komut kaynağı güvenlik anahtarı profilini tanımlamadıysanız, her kaynağa ilişkin kaynak tanıtımlarını uygun sınıfa eklemeniz gerekir. Aynı güvenlik tanıtımları hem MQSC, hem de PCF komutlarını denetler.

Komutlarla ilişkilendirilmiş kaynaklar için güvenlik denetimi yapmak istediğiniz için `hlq.NO.CMD.RESC.CHECKS`komut kaynağı güvenlik anahtarı profilini tanımlamadıysanız, aşağıdakileri yapmanız gerekir:

- Her kaynak için büyük harfli profiller kullanılıyorsa, **MQADMIN** sınıfına bir kaynak profili ekleyin.
- Her kaynak için karışık büyük/küçük harf profilleri kullanılıyorsa, **MXADMIN** sınıfına bir kaynak profili ekleyin.

Aynı güvenlik tanıtımları hem MQSC, hem de PCF komutlarını denetler.

Komut kaynak güvenliği denetimine ilişkin profiller şu formu alır:

```
hlq.type.resourcename
```

Burada `hlq` , `qmgr-name` (kuyruk yöneticisi adı) ya da `qsg-name` (kuyruk paylaşım grubu adı) olabilir.

Kuyruk yöneticisi adı öneki eklenen bir tanıtım, o kuyruk yöneticisindeki komutlarla ilişkilendirilmiş kaynaklara erişimi denetler. Kuyruk paylaşım grubu adı öneki eklenen bir tanıtım, kuyruk paylaşım grubu içindeki tüm kuyruk yöneticilerindeki komutlarla ilişkili kaynaklara erişimi denetler. Bu erişim, söz konusu kuyruk yöneticisinde o komut kaynağı için bir kuyruk yöneticisi düzeyi tanıtımı tanımlanarak, tek bir kuyruk yöneticisinde geçersiz kılınabilir.

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesiyse ve hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ önce kuyruk yöneticisi adının öneki olan bir tanıtımı denetler. Bir tanıtım bulamazsa, kuyruk paylaşım grubu adının öneki olan bir tanıtımı arar.

Örneğin, CSQ1 altsisteminde CREDIT.WORTHY model kuyruğuna karşı RACF komut kaynağı güvenliği denetimi için profil adı:

```
CSQ1.QUEUE.CREDIT.WORTHY
```

Tüm komut kaynağı tiplerine ilişkin tanıtımlar MQADMIN sınıfında tutulduğundan, tanıtım adının "type" kısmı, aynı adı taşıyan farklı tiplerdeki kaynakları ayırt etmek için tanıtımda gereklidir. Tanıtım adının "tip" kısmı CHANNEL, QUEUE, TOPIC, PROCESS ya da NAMELIST olabilir. Örneğin, bir kullanıcının hlq.QUEUE.PAYROLL.ONE, ancak hlq.PROCESS.PAYROLL.ONE

Kaynak tipi bir kuyruksa ve tanıtım bir kuyruk paylaşım grubu düzeyi tanıtımıysa, kuyruk paylaşım grubu içindeki bir ya da daha çok yerel kuyruğa erişimi ya da kuyruk paylaşım grubundaki herhangi bir kuyruk yöneticisinden tek bir paylaşılan kuyruğa erişimi denetler.

MQSC komutları, tanıtımları ve erişim düzeyleri , her IBM MQ MQSC komutu için, komut güvenliği denetiminin gerçekleştirilmesi için gereken tanıtımları ve MQCMDS sınıfındaki her tanıtım için ilgili erişim düzeyini gösterir.

PCF komutları, tanıtımları ve erişim düzeyleri , her IBM MQ PCF komutu için, komut güvenliği denetiminin gerçekleştirilmesi için gereken tanıtımları ve MQCMDS sınıfındaki her tanıtıma ilişkin erişim düzeyini gösterir.

z/OS *Diğer ad kuyrukları ve uzak kuyruklar için komut kaynağı güvenliği denetimi*

Diğer ad kuyruğu ve uzak kuyrukların her ikisi de başka bir kuyruğa yönlendirme sağlar. Bu kuyruklara ilişkin güvenlik denetimini dikkate aldığınızda ek noktalar uygulanır.

Diğer ad kuyrukları

Bir diğer ad kuyruğu tanımladığınızda, komut kaynağı güvenlik denetimleri yalnızca diğer ad kuyruğunun adına göre gerçekleştirilir, diğer adın çözüldüğü hedef kuyruğun adına göre gerçekleştirilmez.

Diğer ad kuyrukları hem yerel, hem de uzak kuyruklara çözülebilir. Kullanıcıların belirli yerel ya da uzak kuyruklara erişmelerine izin vermek istemiyorsanız, aşağıdakilerden ikisini de yapmalısınız:

1. Kullanıcıların bu yerel ve uzak kuyruklara erişmelerine izin vermeyin.
2. Kullanıcıların bu kuyruklar için diğer adlar tanımlayabilmelerini kısıtlayın. Yani, bunların DEFINE QALIAS ve ALTER QALIAS komutlarını vermelerini engelleyin.

Uzak kuyruklar

Bir uzak kuyruk tanımladığınızda, komut kaynağı güvenliği denetimleri yalnızca uzak kuyruğun adına göre gerçekleştirilir. Uzak kuyruk nesnesi tanımındaki RNAME ya da XMITQ özniteliklerinde belirlenen kuyruk adlarıyla ilgili denetim gerçekleştirilmez.

z/OS RESLEVEL güvenlik profili

API kaynak güvenliği için denetlenen kullanıcı kimliklerinin sayısını denetlemek için MQADMIN ya da MXADMIN sınıfında özel bir profil tanımlayabilirsiniz. Bu profile RESLEVEL tanıtımı adı verilir. Bu profilin API-kaynak güvenliğini nasıl etkilediği IBM MQ' e nasıl eriştiğinize bağlıdır.

Bir uygulama IBM MQ' a bağlanmaya çalıştığında, IBM MQ bağlantıyla ilişkili kullanıcı kimliğinin MQADMIN ya da MXADMIN sınıfındaki bir profile erişimi denetler:

```
hlq.RESLEVEL
```

Burada `hlq` , `ssid` (altsistem tanıtıcısı) ya da `qsg` (kuyruk paylaşım grubu tanıtıcısı) olabilir.

Her bağlantı tipiyle ilişkili kullanıcı kimlikleri şunlardır:

- Toplu bağlantılar için bağlanan görevin kullanıcı kimliği
- CICS bağlantıları için CICS adres alanı kullanıcı kimliği
- IMS bağlantıları için IMS bölge adresi alanı kullanıcı kimliği
- Kanal başlatıcı bağlantıları için kanal başlatıcı adres alanı kullanıcı kimliği

Uyarı: RESLEVEL çok güçlü bir seçenektir; belirli bir bağlantıya ilişkin tüm kaynak güvenliği denetimlerinin atlanmasına neden olabilir.

Tanımlı bir RESLEVEL tanıtımınız yoksa, MQADMIN sınıfındaki başka bir tanıtımın hlq.RESLEVELile eşleşmediğine dikkat etmelisiniz. Örneğin, MQADMIN içinde hlq. * * adlı bir tanıtımınız varsa ve hlq.RESLEVEL tanıtımı yok, hlq. * * sonuçlarından sakının Profil, RESLEVEL denetimi için kullanıldığından.

Bir hlq.RESLEVEL tanıtımı tanımlayın ve UACC ' yi RESLEVEL tanıtımı olmaması yerine NONE olarak ayarlayın. Erişim listesinde mümkün olduğunca az sayıda kullanıcı ya da grup bulunmalıdır. RESLEVEL erişiminin denetlenmesine ilişkin ayrıntılar için bkz. “z/OS üzerinde denetime ilişkin önemli noktalar” sayfa 254.

Yalnızca kuyruk yöneticisi düzeyinde güvenlik kullanıyorsanız, IBM MQ `qmgr-name` . RESLEVEL profili için RESLEVEL denetimleri gerçekleştirir. Yalnızca kuyruk paylaşım grubu düzeyinde güvenlik kullanıyorsanız, IBM MQ `qsg-name` . RESLEVEL profili için RESLEVEL denetimleri gerçekleştirir. Hem kuyruk yöneticisi hem de kuyruk paylaşım grubu düzeyi güvenliğinin bir birleşimini kullanıyorsanız, IBM MQ önce kuyruk yöneticisi düzeyinde bir RESLEVEL tanıtımı olup olmadığını denetler. Bir profil bulamazsa, kuyruk paylaşım grubu düzeyinde bir RESLEVEL profili olup olmadığını denetler.

Bir RESLEVEL tanıtımı bulamazsa, IBM MQ bir CICS ya da IMS bağlantısı için hem iş hem de görev (ya da diğer kullanıcı) kimliğinin denetlenmesini etkinleştirir. Toplu iş bağlantısı için IBM MQ , işin (ya da diğer) kullanıcı kimliğinin denetlenmesini etkinleştirir. Kanal başlatıcısı için IBM MQ , kanal kullanıcı kimliğinin ve MCA (ya da diğer) kullanıcı kimliğinin denetlenmesini etkinleştirir.

Bir RESLEVEL profili varsa, denetim düzeyi, profile ilişkin ortama ve erişim düzeyine bağlıdır.

Kuyruk yöneticiniz bir kuyruk paylaşım grubunun üyesiyse ve bu tanıtımı kuyruk yöneticisi düzeyinde tanımlamadıysanız, kuyruk paylaşım grubu düzeyinde tanımlanan bir tanesinin denetim düzeyini etkileyebileceğini unutmayın. İki kullanıcı kimliği denetimini etkinleştirmek için, bir UACC (NONE) ile bir RESLEVEL tanıtımı (kuyruk paylaşım grubu adının kuyruk yöneticisi adı öneki) tanımlayın ve ilgili kullanıcıların bu tanıtım için erişim izni olmadığından emin olun.

Kanal başlatıcısının kullanıcı kimliğinin RESLEVEL değerini göz önünde bulundurduğunuzda, kanal başlatıcısı tarafından kurulan bağlantının kanallar tarafından da kullanılan bağlantı olduğunu unutmayın. Kanal başlatıcısının kullanıcı kimliği için tüm kaynak güvenliği denetimlerinin atlanmasına neden olan bir ayar, tüm kanallar için güvenlik denetimlerini etkili bir şekilde atlar. Kanal başlatıcısının RESLEVEL için kullanıcı kimliği erişimi NONE dışında bir değerse, erişim için yalnızca bir kullanıcı kimliği (READ ya da UPDATE erişim düzeyi için) ya da kullanıcı kimliği (CONTROL ya da ALTER erişim düzeyi için) denetlenmez. Kanal başlatıcısının kullanıcı kimliğine NONE-RESLEVEL dışında bir erişim düzeyi belirlerseniz, kanallar için yapılan güvenlik denetimlerinde bu ayarın etkisini anladığınızdan emin olun.

RESLEVEL profilinin kullanılması, olağan güvenlik denetimi kayıtlarının alınmadığı anlamına gelir. Örneğin, bir kullanıcıya UAUDIT değerini koyarsanız, MQADMIN içindeki hlq.RESLEVEL tanıtımına erişim denetlenmez.

hlq.RESLEVEL tanıtımında RACF UYARI seçeneğini kullanırsanız, RESLEVEL sınıfındaki tanıtımlar için RACF uyarı iletisi üretilmez.

COD gibi rapor iletileri için güvenlik denetimi, kaynak uygulamayla ilişkili RESLEVEL profili tarafından denetlenir. Örneğin, bir toplu işin kullanıcı kimliğinin bir RESLEVEL tanıtımı için CONTROL ya da ALTER yetkisi varsa, rapor iletilerinin güvenlik denetimi de içinde olmak üzere, toplu iş tarafından gerçekleştirilen tüm kaynak denetimi atlanır.

RESLEVEL tanıtımını değiştirirseniz, değişiklik gerçekleşmeden önce kullanıcıların bağlantıyı kesmeleri ve yeniden bağlanmaları gerekir. (Dağıtılmış kuyruğa alma adres alanı kullanıcı kimliğinin RESLEVEL tanıtımına erişimi değiştirildiyse, kanal başlatıcının durdurulması ve yeniden başlatılması da buna dahildir.)

RESLEVEL denetimini devre dışı bırakmak için RESAUDIT sistem parametresini kullanın.

z/OS *RESLEVEL ve toplu iş bağlantıları*

Varsayılan olarak, toplu iş ve toplu iş tipi bağlantılarıyla bir IBM MQ kaynağına erişilirken, kullanıcının belirli bir işlem için o kaynağa erişme yetkisi olmalıdır. Uygun bir RESLEVEL tanımlaması ayarlayarak güvenlik denetimini atlayabilirsiniz.

Kullanıcının bağlantı sırasında kullanılan kullanıcı kimliğine dayalı olup olmadığı, bağlantı denetimi için kullanılan kullanıcı kimliğiyle aynıdır.

Örneğin, RESLEVEL ayarını, güvendiğiniz bir kullanıcı belirli kaynaklara toplu bağlantıyla eriştiğinde, API kaynak güvenliği denetimi yapılmayacak şekilde ayarlayabilirsiniz; ancak güvenmediğiniz bir kullanıcı aynı kaynaklara erişmeye çalıştığında, güvenlik denetimleri normal olarak gerçekleştirilir. RESLEVEL denetimini, yalnızca o kullanıcı tarafından çalıştırılan kullanıcıya ve programlara yeterince güvendiğinizde API kaynak güvenliği denetimlerini atlamak için ayarlamanız gerekir.

Aşağıdaki tabloda toplu bağlantılar için yapılan denetimler gösterilmektedir.

Çizelge 52. Toplu bağlantılar için farklı RACF erişim düzeylerinde yapılan denetimler

RACF erişim düzeyi	**Denetim düzeyi**
YOK	Gerçekleştirilen kaynak denetimleri
READ	Gerçekleştirilen kaynak denetimleri
GÜNCELLE	Gerçekleştirilen kaynak denetimleri
CONTROL	Çek yok.
Çeviri	Çek yok.

z/OS *RESLEVEL ve sistem işlevleri*

RESLEVEL 'in işlem ve denetim panolarına ve CSQUTIL' e uygulanması.

İşlem ve denetim panoları ve CSQUTIL yardımcı programı, kuyruk yöneticisinin komut sunucusuna istekte bulunan toplu iş tipi uygulamalardır; bu nedenle bunlar “RESLEVEL ve toplu iş bağlantıları” sayfa 230içinde açıklanan noktalara tabidir. Kullandıkları SYSTEM.COMMAND.INPUT ve SYSTEM.COMMAND.REPLY.MODEL kuyruklarına ilişkin güvenlik denetimini atlamak için RESLEVEL komutunu kullanabilirsiniz, ancak dinamik kuyruklar için kullanamazsınız SYSTEM.CSQXCMD. *, SYSTEM.CSQOREXX.*, ve SYSTEM.CSQUTIL. *.

Komut sunucusu kuyruk yöneticisinin ayrılmaz bir parçasıdır ve ilişkili bağlantı ya da RESLEVEL denetimi yoktur. Bu nedenle, güvenliği korumak için komut sunucusu, istekte bulunan uygulamanın kullanıcı kimliğinin, yanıtlar için kullanılmakta olan kuyruğu açma yetkisine sahip olduğunu doğrulamalıdır. İşlemler ve denetim panoları için bu SYSTEM.CSQOREXX. *. CSQUTIL için SYSTEM.CSQUTIL. *. Kullanıcıların, kendilerine verilen RESLEVEL yetkisinin yanı sıra, “Sistem kuyruğu güvenliği” sayfa 203başlıklı konuda açıklandığı şekilde bu kuyrukları kullanma yetkisi de olmalıdır.

Komut sunucusunu kullanan diğer uygulamalar için bu, yanıt kuyruğu olarak adlandırdıkları kuyruktur. Bu tür diğer uygulamalar, komut sunucusunu, komut sunucusuna kendinden daha güvenilir bir kullanıcı kimliği geçirerek (ileti bağlamında) yetkisiz kuyruklara ileti yerleştirmesi için yanıltabilir. Bunu önlemek için, SYSTEM.COMMAND.INPUTGIRIşI.

z/OS *RESLEVEL ve CICS bağlantıları*

Varsayılan olarak, bir CICS bağlantısında API kaynağı güvenlik denetimi yapıldığında, iki kullanıcı kimliği denetlenir. Bir RESLEVEL tanıtımı ayarlayarak, denetlenen kullanıcı kimliklerini değiştirebilirsiniz.

İşaretlenen ilk kullanıcı kimliği, CICS adres alanının kullanıcı kimliğidir. Bu, CICS işinin iş kartındaki kullanıcı kimliği ya da z/OS BAŞLATILAN sınıfı ya da başlatılan yordamlar çizelgesi tarafından CICS tarafından başlatılan göreve atanan kullanıcı kimliğidir. (Bu, CICS DFLTUSER değil.)

Denetlenen ikinci kullanıcı kimliği, CICS işlemiyle ilişkili kullanıcı kimliğidir.

Bu kullanıcı kimliklerinden birinin kaynağa erişimi yoksa, istek MQRC_NOT_AUTHORIZED tamamlanma koduyla başarısız olur. Hem CICS adres alanı kullanıcı kimliği hem de CICS işlemini çalıştıran kişinin kullanıcı kimliği, kaynağa doğru düzeyde erişime sahip olmalıdır.

RESLEVEL, yapılan denetimleri nasıl etkileyebilir?

RESLEVEL tanıtımınızı nasıl ayarladığınıza bağlı olarak, bir kaynağa erişim istendiğinde hangi kullanıcı kimliklerinin denetleneceğini değiştirebilirsiniz. Ek bilgi için bkz. Çizelge 53 sayfa 231 .

İşaretlenen kullanıcı kimlikleri, bağlantı sırasında kullanılan kullanıcı kimliğine, yani CICS adres alanı kullanıcı kimliğine bağlıdır. Bu denetim, bir sistemden gelen IBM MQ istekleri (örneğin, bir test sistemi, TESTCICS) için API kaynak güvenliği denetimini atlamanızı, ancak bunları başka bir sistem (örneğin, bir üretim sistemi, PRODCICS) için uygulamanızı sağlar.

Not: CICS adres alanı kullanıcı kimliğinizi BAŞLATILAN sınıftaki "güvenilen" özniteliğiyle ya da RACF başlatılan yordamlar çizelgesi ICHRIN03ile ayarlarsanız, bu, kuyruk yöneticiniz için RESLEVEL profili tarafından oluşturulan CICS adres alanını geçersiz kılar (başka bir deyişle, kuyruk yöneticisi CICS adres alanı için güvenlik denetimlerini gerçekleştirmez). Daha fazla bilgi için bkz. Securing CICS.

Aşağıdaki tabloda CICS bağlantıları için yapılan denetimler gösterilmektedir.

Çizelge 53. CICS bağlantıları için farklı RACF erişim düzeylerinde yapılan denetimler

RACF erişim düzeyi	**Denetim düzeyi**
YOK	IBM MQ , CICS adres alanı kullanıcı kimliğini ve işlem kullanıcı kimliğini denetler.
READ	IBM MQ yalnızca CICS adres alanı kullanıcı kimliğini denetler.

Çizelge 53. CICS bağlantıları için farklı RACF erişim düzeylerinde yapılan denetimler (devamı var)	
RACF erişim düzeyi	**Denetim düzeyi**
GÜNCELLE	İşlem RESSEC (YES) ile CICS olarak tanımlanırsa, IBM MQ CICS adres alanı kullanıcı kimliğini ve hareket kullanıcı kimliğini denetler.
GÜNCELLE	İşlem RESSEC (NO) ile CICS olarak tanımlanmışsa, IBM MQ yalnızca CICS adres alanı kullanıcı kimliğini denetler.
CONTROL ya da ALTER	IBM MQ , herhangi bir kullanıcı kimliğini denetlemez.

z/OS *RESLEVEL ve IMS bağlantıları*

Varsayılan olarak, bir IMS bağlantısı için API kaynağı güvenlik denetimi yapıldığında, iki kullanıcı kimliği denetlenir. Bir RESLEVEL tanıtımı ayarlayarak, denetlenen kullanıcı kimliklerini değiştirebilirsiniz.

Varsayılan olarak, bir IMS bağlantısı için API kaynağı güvenlik denetimi yapıldığında, kaynağa erişime izin verilip verilmediğini görmek için iki kullanıcı kimliği denetlenir.

Denetlenen ilk kullanıcı kimliği, IMS bölgesinin adres alanıdır. Bu, z/OS BAŞLATAN sınıfından ya da başlatılan yordamlar çizelgesinden (SPT) iş kartındaki USER alanından ya da bölgeye atanan kullanıcı kimliğinden alınır.

Denetlenen ikinci kullanıcı kimliği, bağımlı bölgede yapılmakta olan işle ilişkilendirilir. Bu, IMS(tm) bağlantısı için ikinci kullanıcı kimliğinin nasıl belirlendiğikonusunda gösterildiği gibi, bağımlı bölgenin tipine göre belirlenir.

Birinci ya da ikinci IMS kullanıcı kimliğinin kaynağa erişimi yoksa, istek MQRC_NOT_AUTHORIZED tamamlanma koduyla başarısız olur.

IBM MQ RESLEVEL tanıtımları ayarı, IMS hareketlerinin zamanlandığı kullanıcı kimliğini IBMtarafından sağlanan MQ-IMS tetikleyici izleme programı CSQQTRMN ' den değiştiremez. Bu kullanıcı kimliği, varsayılan olarak CSQQTRMN olan tetikleyici izleme programının PSBNAME değeridir.

RESLEVEL, yapılan denetimleri nasıl etkileyebilir?

RESLEVEL tanıtımınızı nasıl ayarladığınıza bağlı olarak, bir kaynağa erişim istendiğinde hangi kullanıcı kimliklerinin denetleneceğini değiştirebilirsiniz. Olası denetimler şunlardır:

- IMS bölge adresi alanı kullanıcı kimliğini ve ikinci kullanıcı kimliğini ya da diğer kullanıcı kimliğini denetleyin.
- Yalnızca IMS bölge adresi alanı kullanıcı kimliğini denetleyin.
- Herhangi bir kullanıcı kimliğini işaretmeyin.

Aşağıdaki tabloda IMS bağlantıları için yapılan denetimler gösterilmektedir.

Çizelge 54. IMS bağlantıları için farklı RACF erişim düzeylerinde yapılan denetimler	
RACF erişim düzeyi	**Denetim düzeyi**
YOK	IMS adres alanı kullanıcı kimliğini ve IMS ikinci kullanıcı kimliğini ya da diğer kullanıcı kimliğini denetleyin.
READ	IMS adres alanı kullanıcı kimliğini denetleyin.
GÜNCELLE	IMS adres alanı kullanıcı kimliğini denetleyin.
CONTROL	Çek yok.
Çeviri	Çek yok.

z/OS *RESLEVEL ve kanal başlatıcı bağlantısı*

Varsayılan olarak, kanal başlatıcı tarafından bir API kaynağı güvenlik denetimi yapıldığında, iki kullanıcı kimliği denetlenir. Bir RESLEVEL tanıtımı ayarlayarak, denetlenen kullanıcı kimliklerini değiştirebilirsiniz.

Varsayılan olarak, kanal başlatıcısı bir API kaynağı güvenlik denetimi yaptığında, kaynağa erişime izin verilip verilmediğini görmek için iki kullanıcı kimliği denetlenir.

Denetlenen kullanıcı kimlikleri, ağdan alınan MCAUSER kanal özniteliği, kanal başlatıcı adres alanı ya da ileti tanımlayıcısına ilişkin diğer kullanıcı kimliği tarafından belirlenebilir. Hangi kullanıcı kimliklerinin denetleneceği, kullandığınız iletişim protokolüne ve PUTAUT kanal özniteliğinin ayarına bağlıdır. Ek bilgi için bkz. “Kanal başlatıcısı tarafından kullanılan kullanıcı kimlikleri” sayfa 238 .

Bu kullanıcı kimliklerinden birinin kaynağa erişimi yoksa, istek MQRC_NOT_AUTHORIZED tamamlanma koduyla başarısız olur.

RESLEVEL, yapılan denetimleri nasıl etkileyebilir?

RESLEVEL tanıtımınızı nasıl ayarladığınıza bağlı olarak, bir kaynağa erişim istendiğinde hangi kullanıcı kimliklerinin denetlendiğini ve kaç tanesinin denetlendiğini değiştirebilirsiniz.

Aşağıdaki tabloda, kanal başlatıcısının bağlantısı için yapılan denetimler ve bu bağlantıyı kullandıklarından bu yana tüm kanallar için yapılan denetimler gösterilmektedir.

Çizelge 55. Kanal başlatıcı bağlantıları için farklı RACF erişim düzeylerinde yapılan denetimler

RACF erişim düzeyi	**Denetim düzeyi**
YOK	İki kullanıcı kimliğini denetleyin.
READ	Bir kullanıcı kimliğini denetleyin.
GÜNCELLE	Bir kullanıcı kimliğini denetleyin.
CONTROL	Çek yok.
Çeviri	Çek yok.
Not: Denetlenen kullanıcı kimliklerinin tanımlaması için bkz. “Kanal başlatıcısı tarafından kullanılan kullanıcı kimlikleri” sayfa 238	

z/OS *RESLEVEL ve grup içi kuyruğa alma*

Varsayılan olarak, grup içi kuyruğa alma aracısı tarafından bir API kaynağı güvenlik denetimi yapıldığında, kaynağa erişime izin verilip verilmediğini görmek için iki kullanıcı kimliği denetlenir. Bir RESLEVEL tanıtımı ayarlayarak, denetlenen kullanıcı kimliklerini değiştirebilirsiniz.

İşaretlenen kullanıcı kimlikleri, iletiyi alan kuyruk yöneticisinin IGQUSER özniteliği, iletiyi SYSTEM.QSG.TRANSMIT.QUEUEya da iletinin ileti tanımlayıcısının *UserIdentifier* alanında belirtilen diğer kullanıcı kimliği. Ek bilgi için bkz. “Grup içi kuyruğa alma aracısı tarafından kullanılan kullanıcı kimlikleri” sayfa 242 .

Grup içi kuyruğa alma aracısı bir iç kuyruk yöneticisi görevi olduğundan, belirtik bir bağlanma isteği yayınlamaz ve kuyruk yöneticisinin kullanıcı kimliği altında çalışır. Grup içi kuyruğa alma aracısı, kuyruk yöneticisi kullanıma hazırlanırken başlar. Grup içi kuyruğa alma aracısının kullanıma hazırlanması sırasında IBM MQ , kuyruk yöneticisiyle ilişkili kullanıcı kimliğinin MQADMIN sınıfındaki şu adı verilen bir tanıtıma erişimini denetler:

```
hlq.RESLEVEL
```

Bu denetim, hlq.NO.SUBSYS.SECURITY anahtarı ayarlanmadıkça her zaman gerçekleştirilir.

RESLEVEL tanıtımı yoksa, IBM MQ iki kullanıcı kimliğinin denetlenmesini etkinleştirir. RESLEVEL tanıtımı varsa, denetim düzeyi, tanıtıma ilişkin kuyruk yöneticisinin kullanıcı kimliğine verilen erişim düzeyine

bağlıdır. Grup içi kuyruğa alma aracısı için farklı RACF(r) erişim düzeylerinde yapılan denetimler , grup içi kuyruğa alma aracısı için yapılan denetimleri gösterir.

Çizelge 56. Grup içi kuyruğa alma aracısı için farklı RACF erişim düzeylerinde yapılan denetimler

RACF erişim düzeyi	**Denetim düzeyi**
YOK	İki kullanıcı kimliğini denetleyin.
READ	Bir kullanıcı kimliğini denetleyin.
GÜNCELLE	Bir kullanıcı kimliğini denetleyin.
CONTROL	Çek yok.
Çeviri	Çek yok.
Not: Denetlenen kullanıcı kimliklerinin tanımlaması için bkz. “Grup içi kuyruğa alma aracısı tarafından kullanılan kullanıcı kimlikleri” sayfa 242	

Kuyruk yöneticisinin kullanıcı kimliği için RESLEVEL tanıtımına verilen izinler değiştirilirse, yeni izinleri almak için grup içi kuyruğa alma aracısı durdurulmalı ve yeniden başlatılmalıdır. Grup içi kuyruğa alma aracısını bağımsız olarak durdurup yeniden başlatmanın bir yolu olmadığından, kuyruk yöneticisinin durdurulması ve yeniden başlatılması gerekir.

z/OS *RESLEVEL ve kullanıcı kimlikleri denetlendi*

Bir RESLEVEL profili ayarlama ve bu tanıtıma erişim verme örneği.

Toplu bağlantılara ilişkin tanıtım adına göre kullanıcı kimliği denetimi - LU 6.2 ve TCP/IP sunucu bağlantısı kanallarına ilişkin tanıtım adıyla karşılaştırılarak denetlenen kullanıcı kimlikleri , RESLEVEL ' in farklı MQI istekleri için denetlenen kullanıcı kimliklerini nasıl etkilediğini gösterir.

Örneğin, aşağıdaki gereksinimleri olan QM66 adlı bir kuyruk yöneticiniz vardır:

- WS21B kullanıcısı kaynak güvenliğinden muaf tutulacaktır.
- CICS başlatılan WXNCICS, CICSWXN adres alanı kullanıcı kimliği altında çalışıyor; yalnızca RESSEC (YES) ile tanımlanan hareketler için tam kaynak denetimi gerçekleştirir.

Uygun RESLEVEL profilini tanımlamak için aşağıdaki RACF komutunu verin:

```
RDEFINE MQADMIN QM66.RESLEVEL UACC(NONE)
```

Daha sonra aşağıdaki komutları kullanarak kullanıcılara bu tanıtım için erişim verin:

```
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(WS21B) ACCESS(CONTROL)
PERMIT QM66.RESLEVEL CLASS(MQADMIN) ID(CICSWXN) ACCESS(UPDATE)
```

Kullanıcı kimlikleri QM66kuyruk yöneticisine bağlıyken bu değişiklikleri yaparsanız, değişiklik yapılmadan önce kullanıcıların bağlantıyı kesmeleri ve yeniden bağlanmaları gerekir.

Bir kullanıcı bağlandığında altsistem güvenliği etkin değilse, ancak bu kullanıcı bağlıyken altsistem güvenliği etkin duruma gelirse, kullanıcıya tam kaynak güvenliği denetimi uygulanır. Doğru RESLEVEL işlemini almak için kullanıcının yeniden bağlanması gerekir.

z/OS z/OS üzerinde güvenlik denetimi için kullanıcı kimlikleri

IBM MQ , kullanıcılarla, uçbirimlerle, uygulamalarla ve diğer kaynaklarla ilişkili kullanıcı kimliklerine dayalı olarak güvenlik denetimlerini başlatır. Bu konu derlemi, her güvenlik denetimi tipi için kullanılacak kullanıcı kimliklerini listeler.

z/OS *Bağlantı güvenliği için kullanıcı kimlikleri*

Bağlantı güvenliği için kullanılan kullanıcı kimliği, bağlantı tipine bağlıdır.

Bağlantı tipi	**Kullanıcı Kimliği içeriği**
Toplu iş bağlantısı	Bağlanan görevin kullanıcı kimliği. Örneğin: • TSO kullanıcı kimliği • USER JCL parametresi tarafından bir toplu işe atanan kullanıcı kimliği • BAŞLATILAN sınıf ya da başlatılan yordamlar çizelgesi tarafından başlatılan bir göreve atanan kullanıcı kimliği
CICS bağlantı	CICS adres alanı kullanıcı kimliği.
IMS bağlantı	IMS bölgesi adres alanı kullanıcı kimliği.
Kanal başlatıcı bağlantısı	Kanal başlatıcı adres alanı kullanıcı kimliği.

z/OS *Komut ve komut kaynağı güvenliği için kullanıcı kimlikleri*

Komut güvenliği ya da komut kaynağı güvenliği için kullanılan kullanıcı kimliği, komutun verildiği yere bağlıdır.

Verildiği kaynak ...	**Kullanıcı Kimliği içeriği**
CSQINP1, CSQINP2ya da CSQINPT	Çek yapılmadı.
Sistem komutu giriş kuyruğu	Komutu içeren iletinin ileti tanımlayıcısının *UserIdentifier* içinde bulunan kullanıcı kimliği. İleti bir *UserIdentifier*içermiyorsa, güvenlik yöneticisine bir kullanıcı kimliği boşluklar geçirilir.
Konsol	Konsolda oturum açan kullanıcı kimliği. Konsol oturum açmamışsa, CSQ6SYSP' de CMDUSER sistem parametresi tarafından ayarlanan varsayılan kullanıcı kimliği. Bir konsoldan komut yayınlamak için konsolun z/OS SYS AUTHORITY özniteliği olmalıdır.
SDSF/TSO konsolu	TSO ya da iş kullanıcısı kimliği.
Operasyonlar ve denetim panoları	TSO kullanıcı kimliği. İşlemleri ve denetim panolarını kullanacaksanız, seçtiğiniz işlemlerle ilgili komutları yayınlamak için gereken yetkiye sahip olmanız gerekir. Ayrıca, tüm hlq.DISPLAYürününe okuma erişiminiz olmalıdır. Paneller, sundukları bilgileri toplamak için çeşitli DISPLAY komutlarını kullandığından, MQCMDS sınıfındaki *nesne* tanıtımları.
MGCRE	MGCRE, UTOKEN ile kullanılıyorsa, UTOKEN içindeki kullanıcı kimliği. MGCRE, UTOKEN olmadan verilirse, TSO ya da iş kullanıcı kimliği kullanılır.
CSQ0UTIL	İş kullanıcı kimliği.
CSQUTIL	İş kullanıcı kimliği.
CSQINPX	Kanal başlatıcı adres alanının kullanıcı kimliği.

z/OS *Kaynak güvenliği için kullanıcı kimlikleri (MQOPEN, MQSUB ve MQPUT1)*

Bu bilgiler, her bağlantı tipi için olağan ve diğer kullanıcı kimliklerine ilişkin kullanıcı kimliklerinin içeriğini gösterir. Denetim sayısı, RESLEVEL tanıtımı tarafından tanımlanır. Denetlenen kullanıcı kimliği, **MQOPEN**, **MQSUB**ya da **MQPUT1** çağrıları için kullanılır.

Not: Tüm kullanıcı kimliği alanları tam olarak alındıkları şekilde işaretlenir. Dönüştürme gerçekleşmez ve örneğin, "Bob", "BOB"ve "bob" içeren üç kullanıcı kimliği alanı eşdeğer değildir.

z/OS *Toplu bağlantılar için denetlenen kullanıcı kimlikleri*

Toplu iş bağlantısı için denetlenen kullanıcı kimliği, görevin nasıl çalıştırılacağına ve diğer bir kullanıcı kimliğinin belirtilip belirtilmediğine bağlıdır.

Çizelge 57. Toplu bağlantılar için profil adına göre kullanıcı kimliği denetimi

Açılışta diğer kullanıcı kimliği belirtildi mi?	hlq.ALTERNATE.USER.userid tanıtımı	hlq.CONTEXT.queuename tanıtımı	hlq.resourcename tanıtımı
Hayır	-	İŞ	İŞ
Evet	İŞ	İŞ	Alt

Anahtar:

Alt
Diğer kullanıcı kimliği.

İŞ

- TSO ya da z/OS UNIX System Services oturum açma kullanıcı kimliği.
- Bir toplu işe atanan kullanıcı kimliği.
- BAŞLATMA sınıfı ya da başlatılan yordamlar çizelgesi tarafından başlatılan bir göreve atanan kullanıcı kimliği.
- Yürütülen Db2 saklanmış yordamıyla ilişkili kullanıcı kimliği

Toplu iş, RESLEVEL değeri READ olarak ayarlanmış ve diğer kullanıcı kimliği denetimi kapalı olan Q1 adlı bir kuyruğa MQPUT1 gerçekleştiriyor.

Toplu iş bağlantıları için farklı RACF(r) erişim düzeylerinde yapılan denetimler ve Toplu iş bağlantıları için tanıtım adına göre kullanıcı kimliği denetimi , iş kullanıcı kimliğinin hlq.Q1tanıtımıyla karşılaştırılarak denetlendiğini gösterir.

z/OS *CICS bağlantıları için denetlenen kullanıcı kimlikleri*

CICS bağlantıları için denetlenen kullanıcı kimlikleri, bir ya da iki denetime ve diğer bir kullanıcı kimliğinin belirtilip belirtilmediğine bağlıdır.

Çizelge 58. CICS-type kullanıcı kimlikleri için profil adına göre kullanıcı kimliği denetimi

Açılışta diğer kullanıcı kimliği belirtildi mi?	hlq.ALTERNATE.USER.userid tanıtımı	hlq.CONTEXT.queuename tanıtımı	hlq.resourcename tanıtımı
Hayır, 1 denetim	-	ADS	ADS
Hayır, 2 denetim	-	ADS + TXN	ADS + TXN
Evet, 1 denetim	ADS	ADS	ADS
Evet, 2 denetim	ADS + TXN	ADS + TXN	ADS + ALT

Anahtar:

Alt
Diğer kullanıcı kimliği

ADS
CICS toplu işiyle ilişkili kullanıcı kimliği ya da CICS başlatılan bir görev olarak çalışıyorsa, BAŞLATILAN sınıf ya da başlatılan yordamlar çizelgesi aracılığıyla.

TXN
CICS işlemiyle ilişkili kullanıcı kimliği. Bu, normalde hareketi başlatan uçbirim kullanıcısının kullanıcı kimliğidir. CICS DFLTUSER, bir PRESET güvenlik uçbirimi ya da el ile oturum açmış bir kullanıcı olabilir.

Aşağıdaki koşullar için denetlenen kullanıcı kimliklerini belirleyin:

- CICS adres alanı kullanıcı kimliği için RESLEVEL tanıtımına ilişkin RACF erişim düzeyi NONE olarak ayarlanır.
- MQOO_OUTPUT ve MQOO_PASS_IDENTITY_CONTEXT içeren bir kuyruğa yönelik MQOPEN çağrısı yapıldı.

Önce, RESLEVEL tanıtımına CICS adres alanı kullanıcı kimliği erişimine dayalı olarak kaç CICS kullanıcı kimliğinin denetlendiğini görün. “RESLEVEL ve CICS bağlantıları” sayfa 231konusunda Çizelge 53 sayfa 231 ' den RESLEVEL profili NONE olarak ayarlanırsa iki kullanıcı kimliği denetlenir. Daha sonra, Çizelge 58 sayfa 236 ' den bu denetimler gerçekleştirilir:

- hlq.ALTERNATE.USER.userid tanıtımı denetlenmedi.
- hlq.CONTEXT.queuename tanıtımı, hem CICS adres alanı kullanıcı kimliği hem de CICS hareket kullanıcı kimliği ile denetlenir.
- hlq.resourcename tanıtımı, hem CICS adres alanı kullanıcı kimliği, hem de CICS hareket kullanıcı kimliği ile denetlenir.

Bu, bu MQOPEN çağrısı için dört güvenlik denetimi yapıldığı anlamına gelir.

z/OS *IMS bağlantıları için denetlenen kullanıcı kimlikleri*

IMS bağlantıları için denetlenen kullanıcı kimlikleri, bir ya da iki denetime ve diğer bir kullanıcı kimliğinin belirtilip belirtilmediğine bağlıdır. İkinci bir kullanıcı kimliği imlenirse, bu, bağımlı bölgenin tipine ve kullanılabilir kullanıcı kimliklerine bağlıdır.

Çizelge 59. IMS-type kullanıcı kimlikleri için profil adına göre kullanıcı kimliği denetimi

Açılışta diğer kullanıcı kimliği belirtildi mi?	hlq.ALTERNATE.USER.userid tanıtımı	hlq.CONTEXT.queuename tanıtımı	hlq.resourcename tanıtımı
Hayır, 1 denetim	-	REG.	REG.
Hayır, 2 denetim	-	REG + SEC	REG + SEC
Evet, 1 denetim	REG.	REG.	REG.
Evet, 2 denetim	REG + SEC	REG + SEC	REG + ALT

Anahtar:

Alt
Diğer kullanıcı kimliği.

REG.
Kullanıcı kimliği olağan durumda, BAŞLATILAN sınıf ya da başlatılan yordamlar çizelgesiyle ya da IMS çalışıyorsa, sunulan bir işten USER JCL değiştirgesiyle belirlenir.

sn
İkinci kullanıcı kimliği, bağımlı bir bölgede yapılmakta olan işle ilişkilendirilir. Çizelge 60 sayfa 238' e göre belirlenir.

Çizelge 60. IMS bağlantısı için ikinci kullanıcı kimliğinin nasıl belirlendiği	
Bağımlı bölge tipleri	**İkinci kullanıcı kimliğini belirlemek için sıradüzen**
• BMP iletisiyle yönlendirilen ve başarılı GET UNIQUE yayınlandı. • IFP ve GET UNIQUE yayınlandı. • -MPP.	Kullanıcı oturum açmışsa, IMS işlemiyle ilişkili kullanıcı kimliği. Varsa, LTERM adı. PSBNAME.
• BMP iletisiyle yönlendirilen ve başarılı GET UNIQUE komutu verilmedi. • BMP ileti yönlendirilmedi. • IFP ve GET UNIQUE yayınlanmadı.	Tüm boşluklar ya da tüm sıfırlar değilse, IMS bağımlı bölge adres alanıyla ilişkili kullanıcı kimliği. PSBNAME.

z/OS *Kanal başlatıcısı tarafından kullanılan kullanıcı kimlikleri*

Bu konu derlemi, sunucu bağlantısı kanalları üzerinden yayınlanan istemci MQI istekleri ve alma kanalları için kullanılan ve denetlenen kullanıcı kimliklerini açıklar. TCP/IP ve LU6.2 için bilgi sağlanır.

Kullanılan güvenlik denetiminin tipini saptamak için, alıcı kanal tanımlamasının PUTAUT değiştirgesini kullanabilirsiniz. IBM MQ ağınızda tutarlı güvenlik denetimi elde etmek için ONLYMCA ve ALTMCA seçeneklerini kullanabilirsiniz.

MCA tarafından kullanılan kullanıcı kimliğini belirlemek için DISPLAY CHSTATUS komutunu kullanabilirsiniz.

z/OS *TCP/IP kullanarak kanal alınması*

Denetlenen kullanıcı kimlikleri, kanalın PUTAUT seçeneğine ve bir ya da iki denetimin gerçekleştirilip gerçekleştirilmeyeceğine bağlıdır.

Çizelge 61. TCP/IP kanallarına ilişkin tanıtım adıyla karşılaştırılarak denetlenen kullanıcı kimlikleri			
Alıcı ya da istekte bulunan kanalda belirtilen PUTAUT seçeneği	**hlq.ALTERNATE.USER.userid tanıtımı**	**hlq.CONTEXT.queuename tanıtımı**	**hlq.resourcename tanıtımı**
DEF, 1 denetim	-	CHL	CHL
DEF, 2 denetim	-	CHL + MCA	CHL + MCA
CTX, 1 denetim	CHL	CHL	CHL
CTX, 2 denetim	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 denetim	-	MCA	MCA
ONLYMCA, 2 denetim	-	MCA	MCA
ALTMCA, 1 denetim	MCA	MCA	MCA
ALTMCA, 2 denetim	MCA	MCA	MCA + ALT

Anahtar:

MCA (MCA kullanıcı kimliği)
Alıcıdaki MCAUSER kanal özniteliği için belirlenen kullanıcı kimliği; boş bırakılırsa, alıcı ya da istekte bulunan tarafın kanal başlatıcı adres alanı kullanıcı kimliği kullanılır.

CHL (Kanal kullanıcı kimliği)
TCP/IP ' de, kanala ilişkin iletişim sistemi güvenliği desteklemez. TLS (Transport Layer Security; İletim Katmanı Güvenliği) kullanılıyorsa ve iş ortağından bir sayısal sertifika akmışsa, bu sertifikayla ilişkili kullanıcı kimliği (kuruluysa) ya da RACF Certificate Name Filtering (CNF) kullanılarak bulunan eşleşen bir süzgeçle ilişkilendirilmiş kullanıcı kimliği kullanılır. İlişkili bir kullanıcı kimliği bulunamazsa ya da TLS kullanılmıyorsa, alıcı ya da istekte bulunan ucunun kanal başlatıcı adres alanının kullanıcı kimliği, PUTAUT parametresi DEF ya da CTX olarak ayarlanmış kanallarda kanal kullanıcı kimliği olarak kullanılır.

Not: RACF Sertifika Adı Süzgeci (CNF) kullanımı, aynı RACF kullanıcı kimliğini birden çok uzak kullanıcıya (örneğin, aynı kuruluş birimindeki, doğal olarak tümü aynı güvenlik yetkisine sahip olan tüm kullanıcılara) atamanızı sağlar. Bu, sunucunun dünyadaki her olası uzak kullanıcının sertifikasının bir kopyasına sahip olması gerekmediği ve sertifika yönetimini ve dağıtımını büyük ölçüde basitleştirdiği anlamına gelir.

Kanal için PUTAUT parametresi ONLYMCA ya da ALTMCA olarak ayarlanırsa, kanal kullanıcı kimliği yoksayılır ve alıcı ya da istekçinin MCA kullanıcı kimliği kullanılır. Bu, TLS kullanan TCP/IP kanalları için de geçerlidir.

ALT (Diğer kullanıcı kimliği)
İletinin ileti tanımlayıcısı içindeki bağlam bilgilerindeki (*UserIdentifier* alanı) kullanıcı kimliği. Bu kullanıcı kimliği, hedef kuyruk için bir **MQOPEN** ya da **MQPUT1** çağrısı yayınlanmadan önce nesne tanımlayıcısındaki *AlternateUserID* alanına taşınır.

z/OS *LU 6.2 kullanılarak kanal alınması*
Denetlenen kullanıcı kimlikleri, kanalın PUTAUT seçeneğine ve bir ya da iki denetimin gerçekleştirilip gerçekleştirilmeyeceğine bağlıdır.

Çizelge 62. LU 6.2 kanallarına ilişkin tanıtım adına göre denetlenen kullanıcı kimlikleri

Alıcı ya da istekte bulunan kanalda belirtilen PUTAUT seçeneği	hlq.ALTERNATE.USER.userid tanıtımı	hlq.CONTEXT.queuename tanıtımı	hlq.resourcename tanıtımı
DEF, 1 denetim	-	CHL	CHL
DEF, 2 denetim	-	CHL + MCA	CHL + MCA
CTX, 1 denetim	CHL	CHL	CHL
CTX, 2 denetim	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 denetim	-	MCA	MCA
ONLYMCA, 2 denetim	-	MCA	MCA
ALTMCA, 1 denetim	MCA	MCA	MCA
ALTMCA, 2 denetim	MCA	MCA	MCA + ALT

Anahtar:

MCA (MCA kullanıcı kimliği)
Alıcıdaki MCAUSER kanal özniteliği için belirlenen kullanıcı kimliği; boş bırakılırsa, alıcı ya da istekte bulunan tarafın kanal başlatıcı adres alanı kullanıcı kimliği kullanılır.

CHL (Kanal kullanıcı kimliği)

İstekte bulunan sunucu kanalları
Kanal istekte bulunandan başlatılırsa, bir ağ kullanıcı kimliği (kanal kullanıcı kimliği) alma fırsatı yoktur.

PUTAUT parametresi, istekte bulunan kanalda DEF ya da CTX olarak ayarlanırsa, kanal kullanıcı kimliği, ağdan kullanıcı kimliği alınmadığı için istekte bulunanın kanal başlatıcısı adres alanıdır.

PUTAUT parametresi ONLYMCA ya da ALTMCA olarak ayarlanırsa, kanal kullanıcı kimliği yoksayılır ve istekte bulunanın MCA kullanıcı kimliği kullanılır.

Diğer kanal tipleri
PUTAUT parametresi, alıcı ya da istekte bulunan kanalda DEF ya da CTX olarak ayarlanırsa, kanal kullanıcı kimliği, kanal başlatıldığında iletişim sisteminden alınan kullanıcı kimliğidir.

- Gönderen kanal z/OSüzerindeyse, alınan kanal kullanıcı kimliği, gönderenin kanal başlatıcısı adres alanı kullanıcı kimliğidir.
- Gönderen kanal farklı bir platformdaysa (örneğin, AIX), alınan kanal kullanıcı kimliği genellikle kanal tanımının USERID parametresi tarafından sağlanır.

Alınan kullanıcı kimliği boşsa ya da kullanıcı kimliği alınmazsa, kanal kullanıcı kimliği olarak boşluk kullanılır.

ALT (Diğer kullanıcı kimliği)
İletinin ileti tanımlayıcısı içindeki bağlam bilgilerindeki (*UserIdentifier* alanı) kullanıcı kimliği. Bu kullanıcı kimliği, hedef kuyruk için bir MQOPEN ya da MQPUT1 çağrısı yayınlanmadan önce nesne tanımlayıcısındaki *AlternateUserID* alanına taşınır.

z/OS *İstemci MQI istekleri*

Hangi kullanıcı kimliklerinin ve ortam değişkenlerinin ayarlandığına bağlı olarak çeşitli kullanıcı kimlikleri kullanılabilir. Bu kullanıcı kimlikleri, kullanılan PUTAUT seçeneğine ve diğer bir kullanıcı kimliğinin belirtilip belirtilmediğine bağlı olarak çeşitli tanıtımlarla karşılaştırılarak denetlenir.

Bu kısımda, TCP/IP ve LU 6.2için sunucu bağlantısı kanalları üzerinden yayınlanan istemci MQI istekleri için denetlenen kullanıcı kimlikleri açıklanmaktadır. MCA kullanıcı kimliği ve kanal kullanıcı kimliği, önceki kısımlarda açıklanan TCP/IP ve LU 6.2 kanalları içindir.

Sunucu bağlantısı kanalları için, MCAUSER özniteliği boşsa istemciden alınan kullanıcı kimliği kullanılır.

Ek bilgi için bkz. “İstemciler için erişim denetimi” sayfa 101 .

İstemci **MQOPEN**, **MQSUB**ve **MQPUT1** istekleri için, denetlenen profili belirlemek üzere aşağıdaki kuralları kullanın:

- İstek diğer kullanıcı yetkisini belirtiyorsa, *hlq*.ALTERNATE.USER. *kullanıcı kimliği* profili.
- İstek bağlam yetkisini belirtiyorsa, *hlq*için bir denetim yapılır.BAğLAM. *queuename* profili.
- Tüm **MQOPEN**, **MQSUB**ve **MQPUT1** istekleri için *hlq.resourcename* profili için bir denetim yapılır.

Hangi tanıtımların denetlendiğini saptadığınızda, bu tanıtımlara göre hangi kullanıcı kimliklerinin denetlendiğini saptamak için aşağıdaki çizelgeyi kullanın.

Çizelge 63. LU 6.2 ve TCP/IP sunucu bağlantısı kanallarına ilişkin tanıtım adına göre denetlenen kullanıcı kimlikleri

Sunucu bağlantısı kanalında belirtilen PUTAUT seçeneği	Açılışta diğer kullanıcı kimliği belirtildi mi?	hlq.ALTERNATE.USER.userid tanıtımı	hlq.CONTEXT.queuename tanıtımı	hlq.resourcename tanıtımı
DEF, 1 denetim	Hayır	-	CHL	CHL
DEF, 1 denetim	Evet	CHL	CHL	CHL

Çizelge 63. LU 6.2 ve TCP/IP sunucu bağlantısı kanallarına ilişkin tanıtım adına göre denetlenen kullanıcı kimlikleri (devamı var)

Sunucu bağlantısı kanalında belirtilen PUTAUT seçeneği	Açılışta diğer kullanıcı kimliği belirtildi mi?	hlq.ALTERNATE.USER.userid tanıtımı	hlq.CONTEXT.queuename tanıtımı	hlq.resourcename tanıtımı
DEF, 2 denetim	Hayır	-	CHL + MCA	CHL + MCA
DEF, 2 denetim	Evet	CHL + MCA	CHL + MCA	CHL + ALT
ONLYMCA, 1 denetim	Hayır	-	MCA	MCA
ONLYMCA, 1 denetim	Evet	MCA	MCA	MCA
ONLYMCA, 2 denetim	Hayır	-	MCA	MCA
ONLYMCA, 2 denetim	Evet	MCA	MCA	MCA + ALT

Anahtar:

MCA (MCA kullanıcı kimliği)
Sunucu bağlantısında MCAUSER kanal özniteliği için belirtilen kullanıcı kimliği; boşsa, kanal başlatıcı adresi alanı kullanıcı kimliği kullanılır.

CHL (Kanal kullanıcı kimliği)
TCP/IP ' de, kanala ilişkin iletişim sistemi güvenliği desteklemez. TLS (Transport Layer Security; İletim Katmanı Güvenliği) kullanılıyorsa ve iş ortağından bir sayısal sertifika akmışsa, bu sertifikayla ilişkili kullanıcı kimliği (kuruluysa) ya da RACF Certificate Name Filtering (CNF) kullanılarak bulunan eşleşen bir süzgeçle ilişkilendirilmiş kullanıcı kimliği kullanılır. İlişkili bir kullanıcı kimliği bulunamazsa ya da TLS kullanılmıyorsa, kanal başlatıcı adresi alanının kullanıcı kimliği, PUTAUT parametresi DEF ya da CTX olarak ayarlanmış olarak tanımlanan kanallarda kanal kullanıcı kimliği olarak kullanılır.

Not: RACF Sertifika Adı Süzgeci (CNF) kullanımı, aynı RACF kullanıcı kimliğini birden çok uzak kullanıcıya (örneğin, aynı kuruluş birimindeki, doğal olarak tümü aynı güvenlik yetkisine sahip olan tüm kullanıcılara) atamanızı sağlar. Bu, sunucunun dünyadaki her olası uzak kullanıcının sertifikasının bir kopyasına sahip olması gerekmediği ve sertifika yönetimini ve dağıtımını büyük ölçüde basitleştirdiği anlamına gelir.

Kanal için PUTAUT parametresi ONLYMCA ya da ALTMCA olarak ayarlanırsa, kanal kullanıcı kimliği yoksayılır ve sunucu bağlantısı kanalının MCA kullanıcı kimliği kullanılır. Bu, TLS kullanan TCP/IP kanalları için de geçerlidir.

ALT (Diğer kullanıcı kimliği)
İletinin ileti tanımlayıcısı içindeki bağlam bilgilerindeki (*UserIdentifier* alanı) kullanıcı kimliği. Bu kullanıcı kimliği, istemci uygulaması adına bir **MQOPEN**, **MQSUB** ya da **MQPUT1** çağrısı yayınlanmadan önce nesne ya da abonelik tanımlayıcısındaki *AlternateUserID* alanına taşınır.

z/OS *Kanal başlatıcı örneği*
Kullanıcı kimliklerinin RACF tanıtımlarına karşı nasıl denetlendiğine ilişkin bir örnek.

Bir kullanıcı, kuyruk yöneticisinde (QM01) bulunan ve QM02kuyruk yöneticisinde QB adlı bir kuyruğa çözülen bir **MQPUT1** işlemi gerçekleştirir. İleti, QM01.TO.QM02. RESLEVEL, NONE olarak ayarlanır ve açma, diğer kullanıcı kimliği ve bağlam denetimiyle gerçekleştirilir. Alıcı kanal tanımında PUTAUT (CTX) var ve

MCA kullanıcı kimliği ayarlandı. İletiyi QB kuyruğuna koymak için alıcı kanalda hangi kullanıcı kimlikleri kullanılır?

Yanıt: Çizelge 55 sayfa 233 , RESLEVEL NONE olarak ayarlandığı için iki kullanıcı kimliğinin denetlendiğini gösterir.

Çizelge 61 sayfa 238 , PUTAUT CTX olarak ayarlandığında ve 2 denetim yapıldığında, aşağıdaki kullanıcı kimliklerinin denetlendiğini gösterir:

- Kanal başlatıcı kullanıcı kimliği ve MCAUSER kullanıcı kimliği, hlq.ALTERNATE.USER.userid profili.
- Kanal başlatıcı kullanıcı kimliği ve MCAUSER kullanıcı kimliği, hlq.CONTEXT.queuename tanıtımıyla karşılaştırılarak denetlenir.
- Kanal başlatıcı kullanıcı kimliği ve ileti tanımlayıcısında (MQMD) belirtilen diğer kullanıcı kimliği, hlq.Q2 tanıtımıyla karşılaştırılarak denetlenir.

z/OS *Grup içi kuyruğa alma aracısı tarafından kullanılan kullanıcı kimlikleri*

Grup içi kuyruğa alma aracısı hedef kuyrukları açtığında denetlenen kullanıcı kimlikleri, **IGQAUT** ve **IGQUSER** kuyruk yöneticisi özniteliklerinin değerlerine göre belirlenir.

Olası kullanıcı kimlikleri şunlardır:

Grup içi kuyruğa alma kullanıcı kimliği (IGQ)
Alıcı kuyruk yöneticisinin **IGQUSER** özniteliği tarafından belirlenen kullanıcı kimliği. Bu de§er boŸluk olarak belirlenirse, alan kuyruk yöneticisinin kullanc kimli§i kullanlr. Ancak, alan kuyruk yöneticisinin kendisine tanımlanan tüm kuyruklara erişim yetkisi olduğu için, alan kuyruk yöneticisinin kullanıcı kimliği için güvenlik denetimleri gerçekleştirilmez. Bu durumda:

- Yalnızca bir kullanıcı kimliği imlenirse ve kullanıcı kimliği alan kuyruk yöneticisininse, güvenlik denetimi yapılmaz. Bu, **IGQAUT** ONLYIGQ ya da ALTIGQolarak ayarlandığında ortaya çıkabilir.
- İki kullanıcı kimliği denetleniyorsa ve kullanıcı kimliklerinden biri, alan kuyruk yöneticisiyse, güvenlik denetimleri yalnızca diğer kullanıcı kimliği için geçerli olur. Bu, **IGQAUT** DEF, CTXya da ALTIGQolarak ayarlandığında ortaya çıkabilir.
- İki kullanıcı kimliği denetlenecek ve her iki kullanıcı kimliği de alıcı kuyruk yöneticisininse, güvenlik denetimi gerçekleştirilmez. Bu, **IGQAUT** ONLYIGQolarak ayarlandığında ortaya çıkabilir.

Gönderen kuyruk yöneticisi kullanıcı kimliği (SND)
İletiyi SYSTEM.QSG.TRANSMIT.QUEUE.

Diğer kullanıcı kimliği (ALT)
İletinin ileti tanımlayıcısındaki *UserIdentifier* alanında belirtilen kullanıcı kimliği.

Çizelge 64. Grup içi kuyruğa alma için profil adına göre denetlenen kullanıcı kimlikleri

Alma kuyruğu yöneticisinde IGQAUT seçeneği belirtildi	hlq.ALTERNATE.USER.userid tanıtımı	hlq.CONTEXT.queuename tanıtımı	hlq.resourcename tanıtımı
DEF, 1 denetim	-	SND	SND
DEF, 2 denetim	-	SND + IGQ	SND + IGQ
CTX, 1 denetim	SND	SND	SND
CTX, 2 denetim	SND + IGQ	SND + IGQ	SND + ALT
ONLYIGQ, 1 denetim	-	IGQ	IGQ
ONLYIGQ, 2 denetim	-	IGQ	IGQ
ALTIGQ, 1 denetim	-	IGQ	IGQ
ALTIGQ, 2 denetim	IGQ	IGQ	IGQ + ALT

Anahtar:

Alt
Diğer kullanıcı kimliği.

IGQ
IGQ kullanıcı kimliği.

SND
Kuyruk yöneticisi kullanıcı kimliği gönderiliyor.

z/OS *Boş kullanıcı kimlikleri ve UACC düzeyleri*

Boş bir kullanıcı kimliği ortaya çıkarsa, RACF tanımsız bir kullanıcı oturum açmış olur. Tanımsız kullanıcıya geniş aralıklı erişim vermeyin.

Bir kullanıcı bağlam ya da diğer kullanıcı güvenliği kullanarak iletileri işlediğinde ya da IBM MQ ' e boş bir kullanıcı kimliği geçirildiğinde boş kullanıcı kimlikleri olabilir. Örneğin, sistem komutu giriş kuyruğuna bağlam olmadan bir ileti yazıldığında boş bir kullanıcı kimliği kullanılır.

Not: Kullanıcı kimliği: " * " (yani, bir yıldız işareti ve ardından yedi boşluk) tanımsız bir kullanıcı kimliği olarak işlenir.

IBM MQ , boş kullanıcı kimliğini RACF ' e iletir ve RACF tanımsız bir kullanıcı oturum açmış olur. Tüm güvenlik denetimleri daha sonra ilgili profil için evrensel erişimi (UACC) kullanır. Erişim düzeylerinizi nasıl ayarladığınıza bağlı olarak UACC, tanımsız kullanıcıya geniş bir erişim verebilir.

Örneğin, TSO ' dan bu RACF komutunu yayınlarsanız:

```
RDEFINE MQQUEUE Q.AVAILABLE.TO.EVERYONE UACC(UPDATE)
```

Hem z/OStanımlı kullanıcı kimliklerini (erişim listesine konmamış), hem de RACF tanımsız kullanıcı kimliğini bu kuyruğa ileti koymak ve kuyruktan ileti almak için geçerli kılan bir tanıtım tanımlayın.

Boş kullanıcı kimliklerine karşı koruma sağlamak için erişim düzeylerinizi dikkatli bir şekilde planlamalı ve bağlam ve alternatif kullanıcı güvenliğini kullanabilecek kişi sayısını sınırlandırmalısınız. RACF tanımsız kullanıcı kimliğini kullanan kişilerin erişmemesi gereken kaynaklara erişmelerini önlemelisiniz. Ancak, aynı zamanda, tanımlı kullanıcı kimliklerine sahip kişilere erişime izin vermeniz gerekir. Bunu yapmak için, PERMIT RACF komutunda yıldız imi (*) kullanıcı kimliğini belirterek, tanımlı tüm kullanıcı kimlikleri için kaynaklara erişim verebilirsiniz. Bu nedenle, tüm tanımsız kullanıcı kimlikleri (örneğin, " * ") erişim verilmedi. Örneğin, bu RACF komutları, RACF tanımsız kullanıcı kimliğinin ileti koymak ya da almak için kuyruğa erişmesini önler:

```
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 kullanıcı kimlikleri ve çok faktörlü kimlik doğrulaması (MFA)

IBM Multi-Factor Authentication for z/OS , z/OS güvenlik denetimcilerinin, tanımlanan kullanıcıların z/OS sisteminde oturum açmak için birden çok kimlik doğrulama etmenini (örneğin, bir parola ve bir şifreleme simgesi) kullanmalarını gerektirerek SAF kimlik doğrulamasını geliştirmelerini sağlar. IBM MFA, RSA SecureIdgibi zamana dayalı tek seferlik parola oluşturma teknolojileri için de destek sağlar.

Çoğu zaman IBM MQ , kullanıcıların CICS ya da IBM MQ işini kullanan toplu iş sistemlerinde nasıl "oturum açtığından" habersiz olur; oturum açan kullanıcı kimliği kimlik bilgileri z/OS görev ya da adres alanıyla ilişkilendirilir ve IBM MQ kaynak yetkilendirmesini denetlemek için bunu kullanır. MFA için etkinleştirilen kullanıcı kimlikleri, CICS ve IMS köprüleriyle kullanılan geçiş kartları aracılığıyla IBM MQ kaynaklarına ve kimlik doğrulamasına yetki vermek için kullanılabilir.

Önemli: Ancak, *MQCSP_AUTH_USER_ID_AND_PWD* seçeneğiyle bir MQCONNX API çağrısında bir kullanıcı kimliği ve parola kimlik bilgilerini ileten IBM MQ Explorergibi uygulamalar kullanılırken dikkat edilmesi gereken özel noktalar vardır. IBM MQ ' in bu API isteğinde ek kimlik bilgisi iletmek için bir olanağı yok.

Sınırlamalar ve olası geçici çözümler aşağıdaki metinde açıklanmıştır.

IBM MQ Explorer

IBM MQ Explorer , MFA 'nın etkinleştirildiği bir kullanıcı kimliğiyle z/OS sisteminde oturum açmak için kullanılamaz; IBM MQ Explorer ' den z/OS' e ikinci bir kimlik doğrulama katsayısı geçirme olanağı yoktur.

Buna ek olarak, IBM MQ Explorer tarafından bir kullanıcı kimliği ve parola kimlik bilgilerini yeniden kullanmak için kullanılan ve bir kerelik kullanım parolaları etkin olduğunda özel dikkat edilmesi gereken iki farklı mekanizma vardır:

1. IBM MQ Explorer , parolaları daha sonra oturum açmak üzere yerel makinede gizlenmiş biçimde saklama yeteneğine sahiptir. Bu yetenek, z/OS kuyruk yöneticisiyle her bağlantı kurulduğunda Explorer (Gezgin) tarafından parola istenerek devre dışı bırakılmalıdır.

   Bunu yapmak için aşağıdaki yordamı kullanın:

   a. **Kuyruk Yöneticileri**seçeneğini belirleyin.
   b. Görüntülenen listeden, istediğiniz kuyruk yöneticisini seçin ve o kuyruk yöneticisini sağ tıklatın.
   c. Görüntülenen menü listesinden **Bağlantı Ayrıntıları** seçeneğini belirleyin.
   d. Sonraki menü listesinden **Özellikler** seçeneğini belirleyin ve **Kullanıcı kimliği** sekmesini seçin.

      Parola istemi radyo düğmesini seçtiğinizden emin olun.
2. Kuyruklardaki iletilere göz atma, abonelikleri sınama gibi IBM MQ Explorericindeki çeşitli işlemler, oturum açmada ilk kullanılan kimlik bilgilerini kullanarak IBM MQ kimlik doğrulamasını başlatan yeni bir iş parçacığı başlatır. Parola kimlik bilgileri yeniden kullanılamadığı için bu işlemleri kullanamazsınız.

Bu sorunlar için MFA yapılandırma düzeyinde iki olası geçici çözüm vardır:

- IBM MQ görevlerini MFA işlemesinden tamamen dışlamak için MFA ' nın uygulama tanıtıcısı hariç tutmasını kullanın.

  Bunu yapmak için aşağıdaki komutları verin:

  1. ```
     RDEFINE MFADEF MFABYPASS.USERID.chinuser
     ```

     Burada *chinuser* , kanal başlatıcı adres alanı düzeyi kullanıcı kimliğidir (STC sınıfı aracılığıyla kanal başlatıcısıyla ilişkilendirilir).
  2. ```
     PERMIT MFABYPASS.USERID.chinuser CLASS MFADEF ACCESS(READ) ID(explorer user)
     ```

  Bu yaklaşımla ilgili daha fazla bilgi için bkz. Uygulamalar için IBM MFA.
- IBM MFA 1.2ile tanıtılan MFA ' da bant dışı desteği kullanın. Bu yaklaşımla, IBM MFA web sunucusunda önceden kimlik doğrulamasını gerçekleştirdiniz ve kullanıcı kimliğiniz ve parolanızın yanı sıra, ilke aracılığıyla belirlendiği şekilde ek kimlik doğrulamasını da belirlersiniz. IBM MFA sunucusu, IBM MQ Explorer kimlik doğrulama iletişim penceresinde belirttiğiniz bir önbellek simgesi kimlik bilgisi oluşturur. Güvenlik yöneticisi, bu kimlik bilgilerinin makul bir süre boyunca yeniden yürütülmesine izin verebilir, bu nedenle normal IBM MQ Explorer kullanımı etkinleştirilir.

Bu yaklaşımla ilgili daha fazla bilgi için bkz. Introduction to IBM MFA.

z/OS IBM MQ for z/OS güvenlik yönetimi

IBM MQ , her bir kullanıcıyla ilgili bilgileri ve her bir kullanıcı tarafından yapılan erişim isteklerini tutmak için bir depolama alanı içinde çizelge kullanır. Bu çizelgeyi verimli bir şekilde yönetmek ve IBM MQ ' den dış güvenlik yöneticisine (ESM) yapılan isteklerin sayısını azaltmak için bir dizi denetim kullanılabilir.

Bu denetimler, hem işlemler hem de denetim panoları ve IBM MQ komutları aracılığıyla kullanılabilir.

z/OS Kullanıcı kimliğini yeniden doğrulama

IBM MQ kaynaklarını kullanan bir kullanıcının RACF tanımlaması değiştirildiyse (örneğin, kullanıcıyı yeni bir gruba bağlayarak), kuyruk yöneticisine bir IBM MQ kaynağına yeniden erişmeye çalıştığında bu kullanıcıyı yeniden oturum açmasını söyleyebilirsiniz. Bunu IBM MQ RVERIFY SECURITY komutunu kullanarak yapabilirsiniz.

- HX0804 kullanıcısı, PRD1kuyruk yöneticisindeki BORDRO kuyruklarını alıyor ve bu kuyruklara ileti yerleştiriyor. Ancak HX0804 artık aynı kuyruk yöneticisindeki (PRD1) PENSION kuyruklarının bazılarına erişim gerektirir.
- Veri güvenliği yöneticisi, HX0804 adlı kullanıcıyı PENSION kuyruklarına erişime izin veren RACF grubuna bağlar.
- HX0804 'ün PENSION kuyruklarına hemen erişebilmesi için (yani, kuyruk yöneticisi PRD1 ' i kapatmadan ya da HX0804 ' ü zamanaşımından beklemeden) IBM MQ komutunu kullanmanız gerekir:

```
RVERIFY SECURITY(HX0804)
```

Not: Kuyruk yöneticisi çalışırken kullanıcı kimliği zamanaşımını uzun süre (gün ya da hafta) kapatırsanız, o süre içinde iptal edilen ya da silinen kullanıcılar için RVERIFY SECURITY komutunu çalıştırmayı unutmayın.

z/OS Kullanıcı kimliği zamanaşımları

Belirli bir süre boşta kaldıktan sonra IBM MQ ' in bir kullanıcıyı kuyruk yöneticisinden çıkarmasını sağlayabilirsiniz.

Bir kullanıcı bir IBM MQ kaynağına eriştiğinde, kuyruk yöneticisi bu kullanıcıyı kuyruk yöneticisinde oturum açma girişiminde bulunur (altsistem güvenliği etkinse). Bu, kullanıcının ESM ' de kimliğinin doğrulandığı anlamına gelir. Bu kullanıcı, kuyruk yöneticisi kapatılıncaya ya da kullanıcı kimliği *zamanaşımına uğratılıncaya* (kimlik doğrulama süresi doluncaya) ya da tersine çevrilinceye (yeniden doğrulanıncaya) kadar IBM MQ ' da oturum açmış olarak kalır.

Bir kullanıcı zamanaşımına uğradığında, kuyruk yöneticisinde kullanıcı kimliği *oturum kapatılır* ve bu kullanıcı için korunan güvenlikle ilgili bilgiler atılır. Kuyruk yöneticisi içindeki kullanıcının oturum açma ve kapatma işlemi, uygulama programı ya da kullanıcı tarafından anlaşılmıyor.

Kullanıcılar, önceden belirlenmiş bir süre boyunca herhangi bir IBM MQ kaynağı kullanmadıklarında zamanaşımına uğramaya hak kazanırlar. Bu zaman dönemi MQSC ALTER SECURITY komutuyla ayarlanır.

ALTER SECURITY komutunda iki değer belirtilebilir:

TIMEOUT
Kullanılmayan bir kullanıcı kimliğinin ve ilişkili kaynaklarının IBM MQ kuyruk yöneticisi içinde kalabileceği süre (dakika).

Aralık
ZAMANAşıMı süresinin dolup dolmadığını belirlemek için kullanıcı kimlikleri ve ilişkili kaynakları arasındaki dakika cinsinden süre.

Örneğin, *TIMEOUT* değeri 30 ise ve *INTERVAL* değeri 10 ise, her 10 dakikada bir IBM MQ , 30 dakika boyunca kullanılmamış olup olmadığını belirlemek için kullanıcı kimliklerini ve ilişkili kaynaklarını denetler. Zamanaşımına uğrayan bir kullanıcı kimliği bulunursa, bu kullanıcı kimliği kuyruk yöneticisi içinde kapatılır. Zamanaşımına uğramayan kullanıcı kimlikleriyle ilişkili zamanaşımına uğrayan kaynak bilgileri bulunursa, bu kaynak bilgileri atılır. Kullanıcı kimliklerini zamanaşımı yapmak istemiyorsanız, *INTERVAL* değerini sıfır olarak ayarlayın. Ancak, *INTERVAL* değeri sıfırsa, siz bir **REFRESH SECURITY** ya da **RVERIFY SECURITY** komutu verinceye kadar, kullanıcı kimliklerinin kapladığı saklama alanı ve ilişkili kaynakları serbest bırakılmaz.

Birden çok tek seferlik kullanıcınız varsa, bu değerin ayarlanması önemli olabilir. Küçük aralık ve zamanaşımı değerleri ayarlarsanız, artık gerekli olmayan kaynaklar serbest bırakılır.

Not: Varsayılan değerler dışında *INTERVAL* ya da *TIMEOUT* değerlerini kullanırsanız, her kuyruk yöneticisi başlangıcında komutu yeniden girmeniz gerekir. **ALTER SECURITY** komutunu ilgili kuyruk yöneticisine ilişkin CSQINP1 veri kümesine koyarak otomatik olarak bunu yapabilirsiniz.

z/OS *z/OS üzerinde kuyruk yöneticisi güvenliği yenileniyor*

Performansı artırmak için IBM MQ for z/OS verileri önbelleğe alır RACF . Belirli güvenlik sınıflarını değiştirdiğinizde, önbelleğe alınan bu bilgileri yenilemeniz gerekir. Performans nedenleriyle güvenliği sık sık yenileyin. Yalnızca TLS güvenlik bilgilerini yenilemeyi de seçebilirsiniz.

Bir kuyruk ilk kez açıldığında (ya da bir güvenlik yenilemesinden bu yana ilk kez) IBM MQ , kullanıcının erişim haklarını almak için bir RACF denetimi gerçekleştirir ve bu bilgileri önbelleğe yerleştirir. Önbelleğe alınan veriler, güvenlik denetiminin gerçekleştirildiği kullanıcı kimliklerini ve kaynakları içerir. Kuyruk aynı kullanıcı tarafından yeniden açılırsa, önbelleğe alınan verilerin varlığı, IBM MQ ' in RACF denetimlerinin yapılmasına gerek olmadığı anlamına gelir ve bu da performansı artırır. Bir güvenlik yenilemesinin işlemi, önbelleğe alınan güvenlik bilgilerini atmak ve IBM MQ 'i RACF' e karşı yeni bir denetim yapmaya zorlamaktır. MQADMIN, MXADMIN, MQPROC, MXPROC, MQQUEUE, MXQUEUE, MQNLIST, MXNLIST ya da MXTOPIC sınıfında tutulan bir RACF kaynak profilini eklediğinizde, değiştirdiğinizde ya da sildiğinizde, tuttukları güvenlik bilgilerini yenilemek için bu sınıfı kullanan kuyruk yöneticilerine söylemeniz gerekir. Bunu yapmak için aşağıdaki komutları verin:

- RACF düzeyinde yenilemek için RACF SETROPTS RACLIST (classname) REFRESH komutu.
- Kuyruk yöneticisi tarafından tutulan güvenlik bilgilerini yenilemek için IBM MQ REFRESH SECURITY komutu. Bu komutun, değişen profillere erişen her kuyruk yöneticisi tarafından verilmesi gerekir. Bir kuyruk paylaşım grubunuz varsa, komutu gruptaki tüm kuyruk yöneticilerine yönlendirmek için komut kapsamı özniteliğini kullanabilirsiniz.

  Not: Var olan bir gruba yeni bir kullanıcı bağladıysanız, IBM MQ RVERIFY SECURITY(userid) komutunu çalıştırmanız gerekir. REFRESH SECURITY (*) komutu, bir IBM MQ kaynağına yeniden erişmeye çalıştığında kuyruk yöneticisinin bu kullanıcıyı yeniden oturum açmasına izin vermez.

IBM MQ sınıflarının herhangi birinde soysal tanıtımlar kullanıyorsanız, soysal tanıtımları değiştirirseniz, eklerseniz ya da silerseniz, olağan RACF yenileme komutlarını da vermeniz gerekir. Örneğin, SETROPTS GENERIC (sınıf adı) YENILE.

Ancak, bir RACF kaynak profili eklenirse, değiştirilirse ya da silinirse ve geçerli olduğu kaynağa henüz erişilmediyse (bu nedenle, hiçbir bilgi önbelleğe alınmaz), IBM MQ , REFRESH SECURITY komutu verilmeden yeni RACF bilgilerini kullanır.

RACF denetimi açıksa (örneğin, RACF RALTER AUDIT (access-girişim (audit_access_level)) komutunu kullanarak), önbelleğe alma işlemi gerçekleşmez ve bu nedenle IBM MQ her denetime ilişkin RACF veri alanına doğrudan başvuruda bulunur. Bu nedenle değişiklikler hemen çekildi ve değişikliklere erişmek için REFRESH SECURITY gerekli değil. RACF RLIST komutunu kullanarak RACF denetiminin açık olup olmadığını onaylayabilirsiniz. Örneğin, komutu

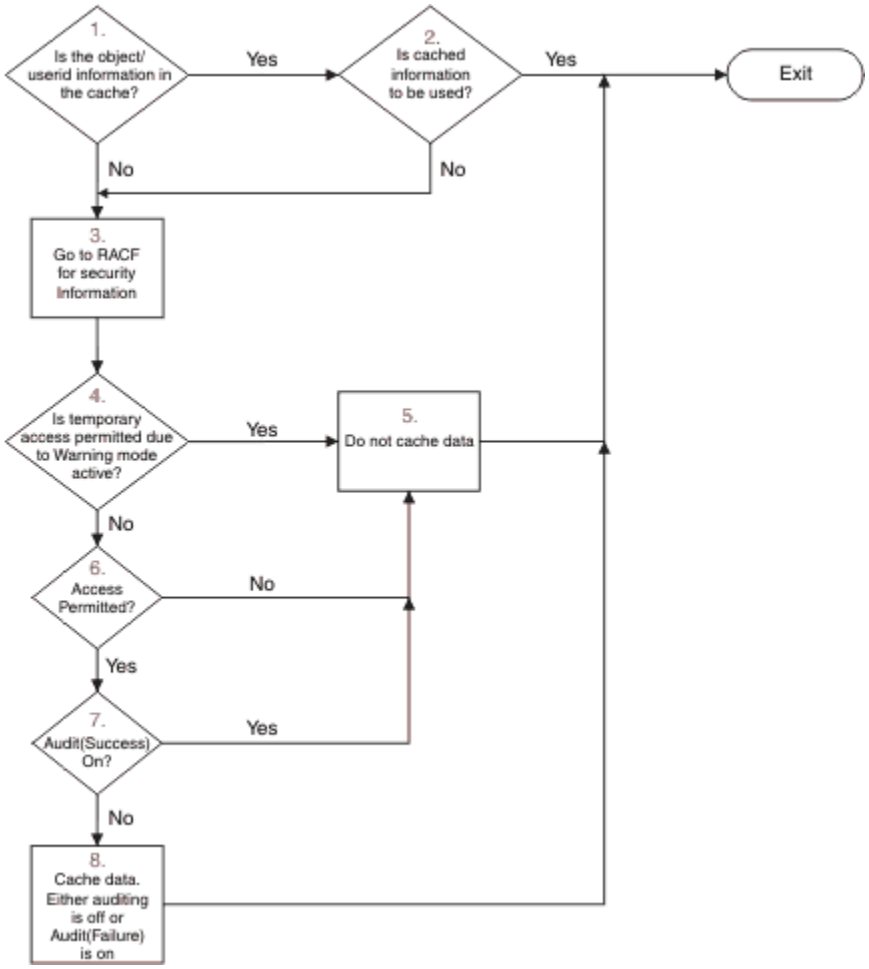
```
RLIST MQQUEUE (qmgr.SYSTEM.COMMAND.INPUT) GEN
```

ve sonuçları alın

```
CLASS      NAME
-----      ----
MQQUEUE    QP*.SYSTEM.COMMAND.*.** (G)
    AUDITING
    --------
    FAILURES(READ)
```

Bu, denetimin açık olduğunu gösterir. Daha fazla bilgi için bkz. *z/OS Security Server RACF Auditor's Guide* ve *z/OS Security Server RACF Command Language Reference*.

Şekil 17 sayfa 247 , güvenlik bilgilerinin önbelleğe alındığı ve önbelleğe alınan bilgilerin kullanıldığı durumları özetler.



Şekil 17. IBM MQ güvenliği önbelleğe alma için mantık akışı

MQADMIN ya da MXADMIN sınıflarına anahtar profilleri ekleyerek ya da bunları silerek güvenlik ayarlarınızı değiştirirseniz, bu değişiklikleri dinamik olarak almak için aşağıdaki komutlardan birini kullanın:

GüVENLIğI YENILE (*)
GüVENLIğI YENILE (MQAdmin)
GüVENLIğI YENILE (MXADMIN)

Bu, yeni güvenlik tiplerini etkinleştirebileceğiniz ya da kuyruk yöneticisini yeniden başlatmak zorunda kalmadan devre dışı bırakabileceğiniz anlamına gelir.

Başarım nedenleriyle, bunlar REFRESH SECURITY komutundan etkilenen tek sınıflardır. MQCONN ya da MQCMDS sınıflarındaki bir tanıtımı değiştirirseniz, REFRESH SECURITY kullanmanız gerekmez.

Not: RESLEVEL güvenlik profilini değiştirirseniz MQADMIN ya da MXADMIN sınıfının yenilenmesi gerekmez.

Performans nedenleriyle, REFRESH SECURITY ' yi ideal olarak yoğun olmayan zamanlarda, mümkün olduğunca seyrek olarak kullanın. Kullanıcıları, erişim listelerine tek tek kullanıcıları yerleştirmek yerine, IBM MQ profillerine ilişkin erişim listesinde bulunan RACF gruplarına bağlayarak güvenlik yenilemelerinin sayısını en aza indirebilirsiniz. Bu şekilde, kaynak profili yerine kullanıcıyı değiştirmeniz gerekir. Güvenliği yenilemek yerine, uygun kullanıcı için RVERIFY SECURITY de yapabilirsiniz.

REFRESH SECURITY örneği olarak, INSURANCE.LIFE . Aşağıdaki RACF komutlarını kullanırsınız:

```
RDEFINE MQQUEUE PRMQ.INSURANCE.LIFE.** UACC(NONE)
PERMIT PRMQ.INSURANCE.LIFE.** ID(LIFEGRP) ACCESS(UPDATE)
```

RACF ' e, tuttuğu güvenlik bilgilerini yenilemesini söylemek için aşağıdaki komutu vermeniz gerekir; örneğin:

```
SETROPTS RACLIST(MQQUEUE) REFRESH
```

Bu tanıtımlar soysal olduğu için, RACF ' e MQQUEUE için soysal tanıtımları yenilemesini söylemelisiniz. Örneğin:

```
SETROPTS GENERIC(MQQUEUE) REFRESH
```

Daha sonra kuyruk yöneticisine kuyruk profillerinin değiştiğini söylemek için bu komutu kullanmanız gerekir:

```
REFRESH SECURITY(MQQUEUE)
```

SSL/TLS güvenliği yenileniyor

TLS Anahtar Havuzunun önbelleğe alınmış görünümünü yenilemek için, REFRESH SECURITY komutunu TYPE (SSL) seçeneğiyle çalıştırın. Bu, kanal başlatıcınızı yeniden başlatmak zorunda kalmadan TLS ayarlarınızdan bazılarını güncellemenizi sağlar.

z/OS *Güvenlik durumunun görüntülenmesi*

Güvenlik anahtarlarının ve diğer güvenlik denetimlerinin durumunu görüntülemek için MQSC DISPLAY SECURITY komutunu verin.

Aşağıdaki şekil, DISPLAY SECURITY ALL komutunun tipik çıkışını göstermektedir.

```
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
```

Şekil 18. DISPLAY SECURITY komutunun tipik çıkışı

Örnekte, komutu yanıtlayan kuyruk yöneticisinin kuyruk yöneticisi düzeyinde etkin, ancak kuyruk paylaşım grubu düzeyinde etkin olmayan altsistem, komut, diğer kullanıcı, işlem, ad ve kuyruk güvenliği vardır. Bağlantı, komut kaynağı ve bağlam güvenliği etkin değil. Ayrıca, kullanıcı kimliği zamanaşımlarının etkin olduğunu ve kuyruk yöneticisinin bu kuyruk yöneticisinde 54 dakikadır kullanılmayan kullanıcı kimliklerini her 12 dakikada bir denetleyip kaldırdığını da gösterir.

Not: Bu komut geçerli güvenlik durumunu gösterir. RACFiçin tanımlanan anahtar profillerinin geçerli durumunu ya da RACF sınıflarının durumunu yansıtması gerekmez. Örneğin, bu kuyruk yöneticisinin ya da REFRESH SECURITY komutunun son yeniden başlatılmasından bu yana anahtar tanıtımları değiştirilmiş olabilir.

z/OS z/OS için güvenlik kuruluşu görevleri

IBM MQürününü kurduktan ve uyarladıktan sonra, başlatılan görev yordamlarını RACFiçin yetkilendirin, çeşitli kaynaklara erişim yetkisi verin ve RACF tanımlamalarını ayarlayın. İsteğe bağlı olarak, sisteminizi TLS için yapılandırın.

IBM MQ ilk kurulduğunda ve özelleştirildiğinde, güvenlikle ilgili şu görevleri gerçekleştirmeniz gerekir:

1. IBM MQ veri kümesini ve sistem güvenliğini aşağıdaki şekilde ayarlayın:
   - Kuyruk yöneticisi başlatıldı-görev yordamı xxxxMSTR ve dağıtılmış kuyruğa alma başlatıldı-görev yordamı xxxxCHIN RACFaltında çalıştırılacak.
   - Kuyruk yöneticisi veri kümelerine erişim yetkisi veriliyor.
   - Kuyruk yöneticisi ve yardımcı programları kullanacak kullanıcı kimlikleri için kaynaklara erişim yetkisi verilmesi.
   - Bağlaşım olanağı listesi yapılarını kullanacak kuyruk yöneticileri için erişim yetkisi verilmesi.
   - Db2' u kullanacak kuyruk yöneticilerine erişim yetkisi verilmesi.
2. IBM MQ güvenliği için RACF tanımlamalarını ayarlayın.
3. TLS (Transport Layer Security; İletim Katmanı Güvenliği) kullanmak istiyorsanız, sisteminizi sertifikaları ve anahtarları kullanacak şekilde hazırlayın.

z/OS *IBM MQ for z/OS veri kümesi güvenliğini ayarlama*

Birçok IBM MQ kullanıcı tipi vardır. Sistem veri kümelerine erişimlerini denetlemek için RACF kullanın.

IBM MQ veri kümelerinin olası kullanıcıları aşağıdaki varlıkları içerir:

- Kuyruk yöneticisinin kendisi.
- Kanal başlatıcı
- IBM MQ veri kümeleri oluşturması gereken IBM MQ yöneticileri, yardımcı programları ve benzer görevleri çalıştırır.
- IBM MQtarafından sağlanan copybook 'ları kullanması gereken uygulama programcıları arasında veri kümeleri, makrolar ve benzeri kaynaklar yer alır.
- Aşağıdakilerden birini ya da birkaçını içeren uygulamalar:
  - Toplu işler
  - TSO kullanıcıları
  - CICS bölge
  - IMS bölge
- CSQOUTX ve CSQSNAP veri kümeleri
- Dinamik kuyruklar SYSTEM.CSQXCMD.*

Tüm bu olası kullanıcılar için, IBM MQ veri kümelerini RACFile koruyun.

Tüm 'CSQINP' veri kümelerinize erişimi de denetlemeniz gerekir.

z/OS *Başlatılan görev yordamlarına ilişkin RACF yetkisi*

Bazı IBM MQ veri kümeleri, kuyruk yöneticisinin özel kullanımı içindir. IBM MQ veri kümelerinizi RACFkullanarak koruyorsanız, kuyruk yöneticisi tarafından başlatılan xxxxMSTRgörev yordamını ve dağıtılmış kuyruğa alma işlemi xxxxCHINyordamını RACFkullanarak yetkilendirmeniz gerekir. Bunu yapmak için BAŞLATAN sınıfını kullanın. Diğer bir seçenek olarak, başlatılan yordamlar çizelgesini (ICHRIN03) kullanabilirsiniz, ancak değişikliklerin yürürlüğe girmesinden önce z/OS sisteminize ilişkin bir IPL gerçekleştirmeniz gerekir.

Daha fazla bilgi için bkz. *z/OS Security Server RACF System Programmer's Guide*.

Tanımlanan RACF kullanıcı kimliği, başlatılan görev yordamındaki veri kümelerine gerekli erişime sahip olmalıdır. Örneğin, CSQ1MSTR adlı bir kuyruk yöneticisi başlatılan görev yordamını RACF kullanıcı kimliği QMGRCSQ1ile ilişkilendirirseniz, QMGRCSQ1 kullanıcı kimliğinin CSQ1 kuyruk yöneticisi tarafından erişilen z/OS kaynaklarına erişimi olmalıdır.

Ayrıca, kuyruk yöneticisinin kullanıcı kimliğine ilişkin GROUP alanının içeriği, o kuyruk yöneticisine ilişkin BAŞLATILAN tanıtımdaki GROUP alanının içeriğiyle aynı olmalıdır. Her bir GROUP alanındaki içerik eşleşmezse, ilgili kullanıcı kimliğinin sisteme girmesi engellenir. Bu durum, IBM MQ ' in tanımsız bir kullanıcı kimliğiyle çalışmasına ve dolayısıyla bir güvenlik ihlali nedeniyle kapanmasına neden olur.

Kuyruk yöneticisi ve kanal başlatıcısı tarafından başlatılan görev yordamlarıyla ilişkili RACF kullanıcı kimliklerinin TRUSTED özniteliği ayarlanmamalıdır.

z/OS *Veri kümelerine erişimin yetkilendirilmesi*

IBM MQ veri kümeleri korunmalıdır; böylece, yetkisiz bir kullanıcı bir kuyruk yöneticisi yönetim ortamını çalıştıramaz ya da herhangi bir kuyruk yöneticisi verilerine erişim kazanamaz. Bunu yapmak için normal z/OS RACF veri kümesi korumasını kullanın.

Çizelge 65 sayfa 250 , kuyruk yöneticisi tarafından başlatılan görev yordamının farklı veri kümeleri için sahip olması gereken RACF erişimini özetler.

Çizelge 65. Bir kuyruk yöneticisiyle ilişkili veri kümelerine RACF erişimi

RACF erişim	**Veri kümeleri**
READ	• `thlqual.SCSQAUTH` ve `thlqual.SCSQANLx` (burada x, ulusal dilinizin dil harfidir). • Kuyruk yöneticisinin başlatılan görev yordamındaki CSQINP1, CSQINP2 ve CSQXLIB tarafından gönderme yapılan veri kümeleri. • Gruptaki diğer kuyruk yöneticilerinin sahip olduğu SMDS veri kümeleri. • Gruptaki diğer kuyruk yöneticileri için günlük, BSDS ve arşiv günlük veri kümeleri.
GÜNCELLE	• Tüm sayfa kümeleri ve günlük ve BSDS veri kümeleri. • Bir kuyruk yöneticisine ait SMDS veri kümeleri • Kuyruk yöneticisinin RECOVER CFSTRUCT komutunu gerçekleştirdiği yapılar için, gruptaki diğer kuyruk yöneticilerinin sahip olduğu SMDS veri kümeleri.
Çeviri	• Tüm arşiv günlüğü veri kümeleri.

Çizelge 66 sayfa 250 , dağıtılmış kuyruğa alma için başlatılan görev yordamının farklı veri kümeleri için sahip olması gereken RACF erişimini özetler.

Çizelge 66. Dağıtılmış kuyruğa alma ile ilişkili veri kümelerine RACF erişimi

RACF erişim	**Veri kümeleri**
READ	• thlqual.SCSQAUTH, thlqual.SCSQANLx (burada x, ulusal dilinizin dil harfidir) ve thlqual.SCSQMVR1. • LE kitaplık veri kümeleri. • Kanal başlatıcısında CSQXLIB ve CSQINPX tarafından başvurulan veri kümeleri görev yordamını başlattı.
GÜNCELLE	• CSQOUTX ve CSQSNAP veri kümeleri

Daha fazla bilgi için bkz. *z/OS Security Server RACF Security Administrator's Guide*.

z/OS *Veri kümelerini şifreleme*

IBM MQ veri kümeleri z/OS veri kümesi şifrelemesiyle şifrelenebilir, böylece veriler korunur ya da yasal nedenlerle.

z/OS veri kümesi şifrelemesiyle tüm sayfa kümelerini, etkin günlüğü, arşiv günlüğünü ve önyükleme (BSDS) veri kümelerini koruyabilirsiniz.

Uyarı: Paylaşılan ileti veri kümelerini (SMDS), IBM MQ for z/OS 9.1.4 ya da daha önceki bir sürümle z/OS veri kümesi şifrelemesiyle koruyamazsınız.

veri kümesi şifrelemesiyle IBM MQ for z/OS üzerinde atıl durumdaki veriler için gizlilik. başlıklı bölüme bakın. ek bilgi için.

z/OS *IBM MQ for z/OS kaynak güvenliğini ayarlama*

Birçok IBM MQ kullanıcı tipi vardır. IBM MQ kaynaklarına erişimlerini denetlemek için RACF komutunu kullanın.

Kuyruklar ve kanallar gibi IBM MQ kaynaklarının olası kullanıcıları aşağıdaki varlıkları içerir:

- Kuyruk yöneticisinin kendisi.
- Kanal başlatıcı
- IBM MQ veri kümeleri yaratması, yardımcı programları çalıştırması ve benzeri görevleri gerçekleştirmesi gereken IBM MQ yöneticileri
- IBM MQtarafından sağlanan copybook 'ları kullanması gereken uygulama programcıları arasında veri kümeleri, makrolar ve benzeri kaynaklar yer alır.
- Aşağıdakilerden birini ya da birkaçını içeren uygulamalar:
  - Toplu işler
  - TSO kullanıcıları
  - CICS bölge
  - IMS bölge
- CSQOUTX ve CSQSNAP veri kümeleri
- Dinamik kuyruklar SYSTEM.CSQXCMD.*

Tüm bu olası kullanıcılar için, IBM MQ kaynaklarını RACFile koruyun. Özellikle, kanal başlatıcının “z/OS üzerinde kanal başlatıcısına ilişkin güvenlikle ilgili önemli noktalar” sayfa 257içinde açıklandığı gibi çeşitli kaynaklara erişmesi gerektiğini ve bu nedenle, kanal başlatıcının çalıştırıldığı kullanıcı kimliğinin bu kaynaklara erişim yetkisi olması gerektiğini unutmayın.

Bir kuyruk paylaşım grubu kullanıyorsanız, kuyruk yöneticisi dahili olarak çeşitli komutlar sunabilir; bu nedenle, kullandığı kullanıcı kimliğinin bu komutları verme yetkisi olmalıdır. Komutlar şunlardır:

- QSGDISP (GROUP) nesnesine sahip her nesne için DEFINE, ALTER ve DELETE
- CHLDISP (SHARED) ile kullanılan her kanal için START ve STOP CHANNEL

z/OS *z/OS sisteminizi TLS kullanacak şekilde yapılandırma*

Bu konuyu, RACF komutlarını kullanarak IBM MQ for z/OS ' un TLS (Transport Layer Security; İletim Katmanı Güvenliği) ile nasıl yapılandırılacağına ilişkin örnek olarak kullanın.

Kanal güvenliği için TLS kullanmak istiyorsanız, sisteminizde gerçekleştirmeniz gereken bazı görevler vardır. (Sertifikalar ve anahtar havuzları (anahtar halkaları) için RACF komutlarının kullanılmasıyla ilgili ayrıntılar için z/OS üzerinde TLS ile çalışma başlıklı konuya bakın.)

1. RACF RACDCERT komutunu kullanarak sisteminize ilişkin tüm anahtarları ve sertifikaları tutmak için RACF içinde bir anahtarlık oluşturun. Örneğin:

```
RACDCERT ID(CHINUSER) ADDRING(QM1RING)
```

Kimlik, kanal başlatıcı adres alanı kullanıcı kimliği ya da anahtarlık paylaşılan bir anahtarlık olacaksa, anahtarlık sahibi olmasını istediğiniz kullanıcı kimliği olmalıdır.

2. RACF RACDCERT komutunu kullanarak, her kuyruk yöneticisi için bir sayısal sertifika yaratın.

   Sertifikanın etiketi, ayarlandıysa IBM MQ **CERTLABL** özniteliğinin değeri ya da kuyruk yöneticisinin ya da kuyruk paylaşım grubunun adının eklendiği varsayılan `ibmWebSphereMQ` değeri olmalıdır. Ayrıntılar için bkz. Dijital sertifika etiketleri . Bu örnekte `ibmWebSphereMQQM1`.

   Örneğin:

```
RACDCERT ID(USERID) GENCERT
SUBJECTSDN(CN('username') O('IBM') OU('departmentname') C('England'))
WITHLABEL('ibmWebSphereMQQM1')
```

3. RACF RACDCERT komutunu kullanarak RACF içindeki sertifikayı anahtar halkasına bağlayın. Örneğin:

```
RACDCERT CONNECT(ID(USERID) LABEL('ibmWebSphereMQQM1') RING(QM1RING))
CONNECT ID(CHINUSER)
```

   Ayrıca ilgili imzalayıcı sertifikalarını (bir sertifika yetkilisinden) anahtar halkasına bağlamanız da gerekir. Yani, bu kuyruk yöneticisinin TLS sertifikasına ilişkin tüm sertifika yetkilileri ve bu kuyruk yöneticisinin iletişim kurduğu tüm TLS sertifikalarına ilişkin tüm sertifika yetkilileri. Örneğin:

```
RACDCERT ID(CHINUSER)
CONNECT(CERTAUTH LABEL('My CA') RING(QM1RING) USAGE(CERTAUTH))
```

4. Kuyruk yöneticilerinizin her birinde, kuyruk yöneticisinin işaret etmesi gereken anahtar havuzunu belirtmek için IBM MQ ALTER QMGR komutunu kullanın. Örneğin, anahtarlık kanal başlatıcı adres alanına aitse:

```
ALTER QMGR SSLKEYR(QM1RING)
```

   ya da paylaşılan bir anahtarlık kullanıyorsanız:

```
ALTER QMGR SSLKEYR(userid/QM1RING)
```

   Burada *userid* , paylaşılan anahtar halkasına sahip olan kullanıcı kimliğidir.

5. Sertifika İptal Listeleri (CRL), sertifika yetkililerinin artık güvenilemeyen sertifikaları iptal etmelerini sağlar. CRL ' ler LDAP sunucularında depolanır. LDAP sunucusundaki bu listeye erişmek için öncelikle IBM MQ DEFINE AUTHINFO komutunu kullanarak AUTHTYPE CRLLDAP için bir AUTHINFO nesnesi oluşturmanız gerekir. Örneğin:

```
DEFINE AUTHINFO(LDAP1)
AUTHTYPE(CRLLDAP)
CONNAME(ldap.server(389))
LDAPUSER('')
LDAPPWD('')
```

Bu örnekte, sertifika iptal listesi LDAP sunucusunun genel bir alanında saklandığı için LDAPUSER ve LDAPPWD alanlarına gerek yoktur.

Daha sonra, IBM MQ DEFINE NAMELIST komutunu kullanarak AUTHINFO nesnenizi bir ad listesi içine koyun. Örneğin:

```
DEFINE NAMELIST(LDAPNL) NAMES(LDAP1)
```

Son olarak, IBM MQ ALTER QMGR komutunu kullanarak namelist değerini her kuyruk yöneticisiyle ilişkilendirin. Örneğin:

```
ALTER QMGR SSLCRLNL(LDAPNL)
```

6. IBM MQ ALTER QMGR komutunu kullanarak, kuyruk yöneticinizi TLS çağrılarını çalıştıracak şekilde ayarlayın. Bu, yalnızca SSL çağrılarını işleyen sunucu alt görevlerini tanımlar ve normal dağıtıcıların herhangi bir SSL çağrılarından etkilenmeden işlemeye normal olarak devam etmesini sağlar. Bu alt görevlerden en az iki tanesinin olması gerekir. Örneğin:

```
ALTER QMGR SSLTASKS(8)
```

   Bu değişiklik yalnızca kanal başlatıcı yeniden başlatıldığında geçerli olur.

7. IBM MQ DEFINE CHANNEL ya da ALTER CHANNEL komutunu kullanarak her kanal için kullanılacak şifre belirtimini belirtin. Örneğin:

```
ALTER CHANNEL(LDAPCHL)
CHLTYPE(SDR)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

   Kanalın her iki ucu da aynı şifre belirtimini belirtmelidir.

z/OS QSG ' de kanal kimlik doğrulama kayıtlarının yönetilmesi

Kanal kimlik doğrulama kayıtları, yaratıldıkları kuyruk yöneticisi için geçerlidir, kuyruk paylaşım grubu (QSG) boyunca paylaşılmaz. Bu nedenle, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinin aynı kurallara sahip olması gerekiyorsa, tüm kuralları tutarlı tutmak için bazı yönetim işlemleri gerçekleştirilmelidir.

1. `CMDSCOPE(*)` seçeneğini her zaman tüm `SET CHLAUTH` komutlarına ekleyin. Bu işlem, komutu kuyruk paylaşım grubundaki tüm çalışan kuyruk yöneticilerine gönderir
2. `CMDSCOPE(*)` seçeneğiyle `DISPLAY CHLAUTH` komutunu kullanın ve daha sonra, kayıtların tüm kuyruk yöneticilerinden aynı olup olmadığını görmek için yanıtları analiz edin. Bir tutarsızlık bulunduğunda, `CMDSCOPE(*)` ya da `CMDSCOPE(`*qmgr-name*`)` ile aynı kuralı içeren bir `SET CHLAUTH` komutu verilebilir.

3. Kuyruk yöneticisinin tam kural kümesine sahip CSQINP2 birleştirmesine bir üye ekleyin (ayrıntılar için Başlatma komutları konusuna bakın). Bunlar, kuyruk yöneticisinin kullanıma hazırlama işleminin bir parçası olarak okunur. `SET CHLAUTH` komutu `ACTION(ADD)` kullanıyorsa, kural ancak var değilse eklenir. `ACTION(REPLACE)` ' ın kullanılması, önceden varsa var olan bir kuralı değiştirir ya da yoksa, bu kuralı ekler. Aynı üye, kuyruk paylaşım grubundaki tüm kuyruk yöneticilerinin CSQINP2 birleştirmesine yerleştirilebilir.
4. `MAKEDEF` ya da `MAKEREP` seçeneğini kullanarak kuralları bir kuyruk yöneticisinden çıkarmak için CSQUTIL yardımcı programını kullanın (ayrıntılar için Komut verme IBM MQ (COMMAND) konusuna bakın). Daha sonra, çıkışı CSQUTIL kullanarak hedef kuyruk yöneticisine yeniden yürüt.

İlgili kavramlar

Kanal kimlik doğrulama kayıtları

Kanal düzeyinde bağlanan sistemlere verilen erişim üzerinde daha kesin denetim sağlamak için kanal kimlik doğrulama kayıtlarını kullanabilirsiniz.

z/OS z/OS üzerinde denetime ilişkin önemli noktalar

Olağan RACF denetim denetimleri, bir kuyruk yöneticisinin güvenlik denetimini gerçekleştirmek için kullanılabilir. IBM MQ , kendi güvenlik istatistiklerini toplamaz. Tek istatistik, denetim yoluyla yaratılabilenlerdir.

RACF denetimi aşağıdakilere dayalı olabilir:

- Kullanıcı Kimlikleri
- Kaynak sınıfları
- Profiller

Daha fazla ayrıntı için bkz. *z/OS Security Server RACF Auditor's Guide*.

Not: Denetim, performansı düşürüyor; ne kadar çok denetim uygularsanız, performans o kadar düşer. Bu, RACF UYARI seçeneğinin kullanımı için de dikkate alınır.

z/OS *RESLEVEL denetimi*

RESLEVEL denetim kayıtlarının üretimini denetlemek için RESAUDIT sistem parametresini kullanın. RACF GENEL denetim kayıtları üretilir.

RESAUDIT sistem parametresini YES değerine ayarlayarak RESLEVEL denetim kayıtları üretin. RESAUDIT parametresi NO olarak ayarlanırsa, denetleme kayıtları üretilmez. Bu parametreyi ayarlamaya ilişkin ek bilgi için CSQ6SYSPbaşlıklı konuya bakın.

RESAUDIT YES olarak ayarlanırsa, hlq.RESLEVEL tanıtımına ilişkin adres alanı kullanıcı kimliğine hangi erişimin olduğunu görmek için RESLEVEL denetimi yapıldığında olağan RACF denetim kaydı alınmaz. Bunun yerine IBM MQ , RACF ' in bir GENERAL denetim kaydı (olay numarası 27) oluşturması için istekte bulunur. Bu denetimler yalnızca bağlantı sırasında gerçekleştirilir, bu nedenle performans maliyeti en düşük düzeyde olur.

Uyarı: RACFRW, RACF denetim kayıtlarını işlemek için artık önerilen yardımcı program değildir. Bu tercih edilen raporlama yöntemi olduğundan RACF SMF veri boşa alma yardımcı programını kullanmanız gerekir.

IBM MQ genel denetim kayıtlarını RACF rapor yazarını (RACFRW) kullanarak bildirebilirsiniz. RESLEVEL erişimini bildirmek için aşağıdaki RACFRW komutlarını kullanabilirsiniz:

```
RACFRW
SELECT PROCESS
EVENT GENERAL
LIST
END
```

Date, *Time*ve *SYSID* alanları hariç olmak üzere RACFRW ' dan bir örnek rapor Şekil 19 sayfa 255içinde gösterilmektedir.

```
       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
```

Şekil 19. RESLEVEL genel denetim kayıtlarını gösteren RACFRW örnek çıktısı

Bu örnek çıkıştaki LOGSTR verilerini denetleyerek, TSO kullanıcısı WS21B ' nin QM66.RESLEVEL. Başka bir deyişle, WS21B kullanıcısı QM66 kaynaklarına eriştiğinde tüm kaynak güvenliği denetimleri atlanır.

RACFRW kullanımı hakkında daha fazla bilgi için *z/OS Security Server RACF Auditor's Guide*içindeki RACF rapor yazıcısı başlıklı konuya bakın.

z/OS Güvenliğin özelleştirilmesi

IBM MQ güvenliğinin çalışma şeklini değiştirmek istiyorsanız, bunu SAF çıkışı (ICHRFR00) aracılığıyla yapmanız ya da dış güvenlik yöneticinizden çıkmanız gerekir.

RACF çıkışlarıyla ilgili daha fazla bilgi için *z/OS Security Server RACROUTE Macro Reference* belgesine bakın.

Not: IBM MQ , ESM ' ye yapılan çağrıları eniyilediği için, örneğin belirli bir kullanıcı tarafından belirli bir kuyruk için her açılışta RACROUTE istekleri yapılmayabilir.

z/OS z/OS üzerinde güvenlik ihlali iletileri

Güvenlik ihlali, bir uygulama programında MQRC_NOT_AUTHORIZED dönüş koduyla ya da iş günlüğündeki bir iletiyle gösterilir.

Aşağıdaki nedenlerden ötürü, bir uygulama programına MQRC_NOT_AUTHORIZED dönüş kodu döndürülebilir:

- Bir kullanıcının kuyruk yöneticisine bağlanmasına izin verilmiyor. Bu durumda, Toplu İş/TSO, CICSya da IMS iş günlüğünde bir ICH408I iletisi alırsınız.
- Örneğin, iş kullanıcı kimliği geçerli ya da uygun olmadığından ya da görev kullanıcı kimliği ya da diğer kullanıcı kimliği geçerli olmadığından, kuyruk yöneticisinde oturum açma işlemi başarısız oldu. Bu kullanıcı kimliklerinden biri ya da daha fazlası iptal edildikleri ya da silindikleri için geçerli olmayabilir. Bu durumda, kuyruk yöneticisi iş günlüğünde oturum açma hatasının nedenini belirten bir ICHxxxx iletisi ve IRRxxxx iletisi alırsınız. Örneğin:

  ```
  ICH408I USER(NOTDFND ) GROUP(        ) NAME(???                 )
    LOGON/JOB INITIATION - USER AT TERMINAL          NOT RACF-DEFINED
  IRR012I  VERIFICATION FAILED. USER PROFILE NOT FOUND
  ```

- Başka bir kullanıcı istendi, ancak iş ya da görev kullanıcı kimliğinin diğer kullanıcı kimliğine erişimi yok. Bu hata için, ilgili kuyruk yöneticisinin iş günlüğünde bir ihlal iletisi alırsınız.
- Bir bağlam seçeneği, çıkış için bir iletim kuyruğu açılarak kullanıldı ya da örtük olarak ima edildi, ancak iş kullanıcı kimliği ya da varsa, görevin ya da diğer kullanıcı kimliğinin bağlam seçeneğine erişimi yok. Bu durumda, ilgili kuyruk yöneticisinin iş günlüğüne bir ihlal iletisi yerleştirilir.

- Yetkisiz bir kullanıcı, güvenli bir kuyruk yöneticisi nesnesine (örneğin, bir kuyruğa) erişme girişiminde bulundu. Bu durumda, ilgili kuyruk yöneticisinin iş günlüğüne ihlale ilişkin bir ICH408I iletisi yerleştirilir. Bu aykırılık, iş ya da uygunsa, görev ya da diğer kullanıcı kimliğinden kaynaklanıyor olabilir.

Kuyruk yöneticisinin iş günlüğünde, komut güvenliği ve komut kaynağı güvenliğine ilişkin ihlal iletileri de bulunabilir.

ICH408I kural dışı durum iletisi, kullanıcı kimliği yerine kuyruk yöneticisi iş adını gösteriyorsa, bu normalde boş bir diğer kullanıcı kimliğinin belirtilmesinin sonucudur. Örneğin:

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          INSUFFICIENT ACCESS AUTHORITY
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

hlq.ALTERNATE.USER.-BLANK-.

Aşağıdakiler tarafından bir ICH408I ihlal iletisi de oluşturulabilir:

- Bağlam olmadan sistem komutu giriş kuyruğuna gönderilen bir komut. Sistem komutu giriş kuyruğuna yazan kullanıcı tarafından yazılan programlar her zaman bir bağlam seçeneği kullanmalıdır. Daha fazla bilgi için bkz "Bağlam güvenliğine ilişkin profiller" sayfa 213.
- IBM MQ kaynağına erişen işle ilişkilendirilmiş bir kullanıcı kimliği olmadığında ya da bir IBM MQ bağdaştırıcısı kullanıcı kimliğini bağdaştırıcı ortamından çıkaramadığında.

Hem kuyruk paylaşım grubu hem de kuyruk yöneticisi düzeyinde güvenlik kullanıyorsanız, kural dışı iletiler de yayınlanabilir. Kuyruk yöneticisi düzeyinde bir tanıtım bulunamadığını, ancak kuyruk paylaşım grubu düzeyi tanıtımı nedeniyle erişim izni verildiğini bildiren iletiler alabilir.

```
ICH408I JOB(MQS1MSTR) STEP(MQS1MSTR)
          MQS1.PAYROLL.REQUEST CL(MQQUEUE)
          PROFILE NOT FOUND - REQUIRED FOR AUTHORITY CHECKING
          ACCESS INTENT(UPDATE )  ACCESS ALLOWED(NONE   )
```

ICH408I iletilerine ilişkin ek bilgi için z/OS for Security Server RACF Messages and Codes (Güvenlik Sunucusu İçin İletiler ve Kodlar) belgelerine bakın.

z/OS Erişime izin verilirse ya da yanlış bir şekilde izin verilmezse ne yapmanız gerekir?

z/OS belgelerinde ayrıntılı olarak açıklanan bilgilere ek olarak, bir kaynağa erişim yanlış denetleniyorsa bu denetim listesini kullanın.

Erişime izin verilirse ya da izin verilmezse, ayrıntılı adımlar için *z/OS Security Server RACF Security Administrator's Guide* adlı yayına bakın.

- Anahtar profilleri doğru ayarlandı mı?
  - RACF etkin mi?
  - IBM MQ RACF sınıfları kurulu ve etkin mi?

    Bunu denetlemek için RACF komutunu (SETROPTS LIST) kullanın.
  - Kuyruk yöneticisinden geçerli anahtar durumunu görüntülemek için IBM MQ DISPLAY SECURITY komutunu kullanın.
  - MQADMIN sınıfındaki anahtar profillerini denetleyin.

    Bunun için RACF komutlarını (SEARCH ve RLIST) kullanın.

  - IBM MQ REFRESH SECURITY (MQADMIN) komutunu vererek RACF anahtar profillerini yeniden denetleyin.
- RACF kaynak profili değişti mi? Örneğin, tanıtıma ilişkin genel erişim değiştirildi mi ya da tanıtıma ilişkin erişim listesi değiştirildi mi?
  - Profil genel mi?

    Varsa, RACF komutunu verin, SETROPTS GENERIC (sınıf adı) YENILE.
  - Bu kuyruk yöneticisindeki güvenliği yeniledin mi?

    Gerekirse, RACF komutunu SETROPTS RACLIST (sınıf adı) verin YENILE.

    Gerekirse, IBM MQ REFRESH SECURITY (*) komutunu verin.
- Kullanıcının RACF tanımı değişti mi? Örneğin, kullanıcı yeni bir gruba bağlandı mı ya da kullanıcı erişimi yetkisi iptal edildi mi?
  - IBM MQ RVERIFY SECURITY (userid) komutunu çalıştırarak kullanıcıyı tersine çevirdiniz mi?
- RESLEVEL nedeniyle güvenlik denetimleri atlanıyor mu?
  - RESLEVEL tanıtımına bağlanan kullanıcı kimliğinin erişimini denetleyin. RESLEVEL ' in hangi değere ayarlandığını belirlemek için RACF denetim kayıtlarını kullanın.
  - Kanallar için, kanal başlatıcısının kullanıcı kimliğinin RESLEVEL için sahip olduğu erişim düzeyinin tüm kanallar tarafından devralındığını unutmayın; bu nedenle, ALTER gibi, tüm denetimlerin atlanmasına neden olan bir erişim düzeyi, tüm kanallar için güvenlik denetimlerinin atlanmasına neden olur.
  - CICS' den çalışıyorsanız, hareketin RESSEC ayarını denetleyin.
  - Bir kullanıcı bağlıyken RESLEVEL değiştirildiyse, yeni RESLEVEL ayarı yürürlüğe girmeden önce bağlantının kesilmesi ve yeniden bağlanması gerekir.
- Kuyruk paylaşım gruplarını mı kullanıyorsunuz?
  - Hem kuyruk paylaşım grubunu hem de kuyruk yöneticisi düzeyinde güvenlik kullanıyorsanız, doğru tüm tanıtımları tanımladığınızı doğrulayın. Kuyruk yöneticisi tanıtımı tanımlı değilse, günlüğe tanıtımın bulunamadığını bildiren bir ileti gönderilir.
  - Tam güvenlik denetiminin açık olması için geçerli olmayan bir anahtar ayarları birleşimi kullandınız mı?
  - Kuyruk yöneticinizin bazı kuyruk paylaşım grubu ayarlarını geçersiz kılmak için güvenlik anahtarları tanımlamanız gerekiyor mu?
  - Kuyruk yöneticisi düzeyinde bir profil, kuyruk paylaşım grubu düzeyi profilinden öncelikli mi?

z/OS z/OS üzerinde kanal başlatıcısına ilişkin güvenlikle ilgili önemli noktalar

Dağıtılmış bir kuyruğa alma ortamında kaynak güvenliğini kullanıyorsanız, Kanal başlatıcı adres alanının çeşitli IBM MQ kaynaklarına uygun erişimi olması gerekir. Parola koruma algoritmasını tohumlamak için ICSF ' yi (Tümleşik Cryptographic Support Olanağı) kullanabilirsiniz.

ICSF ' ye ilişkin ek bilgi için z/OS Cryptographic Services belgelerine bakın.

Kaynak güvenliğinin kullanılması

Kaynak güvenliği kullanıyorsanız, dağıtımlı kuyruğa alma kullanıyorsanız aşağıdaki noktaları göz önünde bulundurun:

Sistem kuyrukları
Kanal başlatıcı adres alanı, “Sistem kuyruğu güvenliği” sayfa 203adresinde listelenen sistem kuyruklarına ve tüm kullanıcı hedef kuyruklarına ve gönderilmeyen iletiler kuyruğuna RACF UPDATE erişimi gerektirir (ancak bkz. “Gönderilmeyen ileti kuyruğu güvenliği” sayfa 201).

İletim kuyrukları
Kanal başlatıcı adres alanı, tüm kullanıcı iletim kuyruklarına ALTER erişimi gerektirir.

Bağlam güvenliği

Kanal kullanıcı kimliği (ve belirtildiyse MCA kullanıcı kimliği) için MQADMIN sınıfındaki hlq.CONTEXT.queuename tanıtımlarına RACF CONTROL erişimi gerekir. RESLEVEL tanıtımına bağlı olarak, kanal kullanıcı kimliğinin bu tanıtımlara da CONTROL erişimi olması gerekebilir.

Tüm kanallar için MQADMIN hlq.CONTEXTiçin CONTROL erişimi gerekir. Ölü mektup kuyruğu profili. Tüm kanallar (başlatma ya da yanıt verme) raporlar oluşturabilir ve sonuç olarak hlq.CONTEXT.reply-q profiline CONTROL erişimine gereksinim duyarlar.

SENDER, CLUSSDR ve SERVER kanalları için hlq.CONTEXT.xmit-queue-name tanıtımlarına CONTROL erişimi gerekir; çünkü iletiler iletim kuyruğuna konarak kanalı zarif bir şekilde sona erdirebilir.

Not: Kanal kullanıcı kimliği ya da kanal kullanıcı kimliğinin bağlı olduğu bir RACF grubu hlq.RESLEVELiçin CONTROL ya da ALTER erişimine sahipse, kanal başlatıcısı ya da kanallarından herhangi biri için kaynak denetimi yoktur.

Daha fazla bilgi için bkz. “Bağlam güvenliğine ilişkin profiller” sayfa 213 “RESLEVEL ve kanal başlatıcı bağlantısı” sayfa 233 ve “z/OS üzerinde güvenlik denetimi için kullanıcı kimlikleri” sayfa 234 .

CSQINPX

CSQINPX giriş veri kümesini kullanıyorsanız, kanal başlatıcının CSQINPX 'e okuma erişimi ve CSQOUTX veri kümesine ve SYSTEM.CSQXCMD. *.

Bağlantı güvenliği

Kanal başlatıcı adres alanı bağlantı istekleri, uygun erişim güvenliğinin ayarlanması gereken bir CHIN bağlantı tipini kullanıyor, bkz. “Kanal başlatıcısına ilişkin bağlantı güvenliği tanıtımları” sayfa 196.

Veri kümeleri

Kanal başlatıcı adres alanının kuyruk yöneticisi veri kümelerine uygun erişimi olması gerekir, bkz. “Veri kümelerine erişimin yetkilendirilmesi” sayfa 250.

Komutlar

Dağıtılmış kuyruğa alma komutları (örneğin, DEFINE CHANNEL, START CHINIT, START LISTENER ve diğer kanal komutları) uygun komut güvenliği kümesine sahip olmalıdır, bkz. Çizelge 49 sayfa 216.

Bir kuyruk paylaşım grubu kullanıyorsanız, kanal başlatıcısı dahili olarak çeşitli komutlar sunabilir; bu nedenle, kullandığı kullanıcı kimliğinin bu tür komutları verme yetkisi olmalıdır. Bu komutlar, CHLDISP (SHARED) ile kullanılan her kanal için START ve STOP CHANNEL komutlarıdır.

Kuyruk yöneticisinin PSMODE 'si DISABLED değilse, kanal başlatıcının DISPLAY PUBSUB komutuna okuma erişimi olmalıdır.

Kanal güvenliği

Kanallar, özellikle günlük nesneleri ve sunucu bağlantıları için uygun güvenliğin ayarlanması gerekir; ek bilgi için bkz. “z/OS üzerinde güvenlik denetimi için kullanıcı kimlikleri” sayfa 234 .

Kanallarda güvenlik sağlamak için TLS (Transport Layer Security; İletim Katmanı Güvenliği) protokolünü de kullanabilirsiniz. TLS ' yi IBM MQile kullanma hakkında daha fazla bilgi için bkz. “IBM MQ içinde TLS güvenlik iletişim kuralları” sayfa 24 .

Sunucu bağlantısı güvenliğiyle ilgili bilgi için bkz. “İstemciler için erişim denetimi” sayfa 101 .

Kullanıcı Kimlikleri

“Kanal başlatıcısı tarafından kullanılan kullanıcı kimlikleri” sayfa 238 ve “Grup içi kuyruğa alma aracısı tarafından kullanılan kullanıcı kimlikleri” sayfa 242 içinde açıklanan kullanıcı kimlikleri için aşağıdaki erişim gerekir:

- RACF Uygun hedef kuyruklara ve gönderilmeyen ileti kuyruğuna güncelleme erişimi
- Alıcıda bağlam denetimi gerçekleştirilirse, RACF `hlq.CONTEXT.queuename` tanıtımına CONTROL erişimi
- hlq.ALTERNATE.USER.userid profillerini kullanmaları gerekebilir.
- İstemciler için, kullanılacak kaynaklara uygun RACF erişimi.

APPC güvenliği

LU 6.2 iletim protokolünü kullanıyorsanız, uygun APPC güvenliğini ayarlayın. (Örneğin, APPCLU RACF sınıfını kullanın.) APPC için güvenlik ayarlarına ilişkin bilgi için aşağıdaki belgelere bakın:

- *z/OS MVS Planlama: APPC Yönetimi*
- *z/OS MVS Programlama: Writing Servers for APPC/MVS*

Giden iletimlerde "SECURITY (SAME)" APPC seçeneği kullanılır. Sonuç olarak, kanal başlatıcı adres alanının ve varsayılan tanıtımının (RACF GROUP) kullanıcı kimliği, kullanıcı kimliğinin önceden doğrulandığına (ALREADYV) ilişkin bir göstergeyle ağ üzerinden alıcıya taşınır.

Alıcı taraf da z/OSise, kullanıcı kimliği ve tanıtım APPC tarafından doğrulanır ve kullanıcı kimliği alıcı kanala sunulur ve kanal kullanıcı kimliği olarak kullanılır.

Kuyruk yöneticisinin aynı ya da başka bir z/OS sisteminde başka bir kuyruk yöneticisiyle iletişim kurmak için APPC kullandığı bir ortamda aşağıdakilerden birini doğrulamanız gerekir:

- İletişim LU için VTAM tanımı SETACPT (ALREADYV) belirtiyor
- CONVSEC (ALREADYV) belirleyen LU ' lar arasındaki bağlantıya ilişkin bir RACF APPCLU tanıtımı vardır.

Güvenlik ayarlarının değiştirilmesi

Kanal kullanıcı kimliğinin ya da MCA kullanıcı kimliğinin bir hedef kuyruğa sahip olduğu RACF erişim düzeyi değiştirilirse, bu değişiklik yalnızca hedef kuyruğa ilişkin yeni nesne tanıtıcıları (yeni `MQOPEN` ' ler) için geçerli olur. MCA açık ve kapalı kuyruklarının değişken olduğu zamanlar; böyle bir erişim değişikliği yapıldığında bir kanal zaten çalışıyorsa, MCA, güncellenen güvenlik erişimi yerine kullanıcı kimliklerinin var olan güvenlik erişimini kullanarak iletileri hedef kuyruğa koymaya devam edebilir. Güncellenen erişim düzeyini zorlamak için kanalların durdurulması ve yeniden başlatılması bu senaryoyu önler.

Otomatik yeniden başlatma

Kanal başlatıcısını yeniden başlatmak için z/OS Automatic Restart Manager (ARM) olanağını kullanıyorsanız, XCFAS adres alanıyla ilişkili kullanıcı kimliğinin IBM MQ START CHINIT komutunu verme yetkisi olmalıdır.

Integrated Cryptographic Service Facility (ICSF) olanağının kullanılması

Kanal başlatıcısı, TLS kullanılmıyorsa istemci kanalları üzerinden akan parolaları karartmak için parola koruma algoritmasını dağıtırken rasgele bir sayı oluşturmak için ICSF ' yi kullanabilir. Rasgele bir sayı oluşturma işlemine *entropi*denir.

z/OS özelliği kuruluysa, ancak ICSF ' yi başlatmadıysanız, CSQX213E iletisini görürsünüz ve kanal başlatıcı entropi için STCK kullanır.

CSQX213E iletisi, parola koruma algoritmasının olabileceği kadar güvenli olmadığı konusunda sizi uyarır. Ancak, sürecinize devam edebilirsiniz; çalıştırma zamanı üzerinde başka bir etki yoktur.

z/OS özelliği kurulu değilse, kanal başlatıcısı otomatik olarak STCK kullanır.

Notlar:

1. ICSF 'nin entropi için kullanılması, STCK' yi kullanmaktan daha fazla rastgele dizi üretir.
2. ICSF ' yi başlatacaksanız, kanal başlatıcıyı yeniden başlatmanız gerekir.
3. Belirli CipherSpecsiçin ICSF gereklidir. Bu CipherSpecs ' lerden birini kullanma girişiminde bulunursanız ve sisteminizde ICSF kurulu değilse, CSQX629Eiletisini alırsınız.

z/OS z/OS üzerindeki kuyruk yöneticisi kümelerinde güvenlik

Kümeler için güvenlikle ilgili önemli noktalar, kümelenmemiş kuyruk yöneticileri ve kanallar için aynıdır. Kanal başlatıcının bazı ek sistem kuyruklarına erişmesi gerekir ve bazı ek komutlar için uygun güvenlik kümesi gerekir.

Küme kanallarını (geleneksel kanallarda olduğu gibi) doğrulamak için MCA kullanıcı kimliğini, kanal kimlik doğrulama kayıtlarını, TLS ve güvenlik çıkışlarını kullanabilirsiniz. Küme-alıcı kanalına ilişkin kanal kimlik doğrulama kayıtları ya da güvenlik çıkışı, uzak kuyruk yöneticisinin sunucu kuyruk yöneticisinin küme kuyruklarına erişmesine izin verildiğini doğrulamalıdır. Var olan kuyruk erişim güvenliğinizi değiştirmeden IBM MQ küme desteğini kullanmaya başlayabilirsiniz. Ancak, kümedeki diğer kuyruk yöneticilerinin SYSTEM.CLUSTER.COMMAND.QUEUE .

IBM MQ küme desteği, bir kümenin üyesini yalnızca istemci rolüyle sınırlamak için bir mekanizma sağlamaz. Sonuç olarak, kümede bulunmasına izin verdiğiniz kuyruk yöneticilerine güvendiğinizden emin olmanız gerekir. Kümedeki herhangi bir kuyruk yöneticisi belirli bir ada sahip bir kuyruk oluştururursa, bu kuyruk için, uygulamanın iletileri bu kuyruğa koyup koymadığına bakılmaksızın, bu kuyruk için ileti alabilir.

Bir kümenin üyeliğini sınırlamak için, kuyruk yöneticilerinin alıcı kanallara bağlanmasını önlemek için yapmanız gerekenle aynı işlemi gerçekleştirin. Kanal kimlik doğrulama kayıtlarını kullanarak ya da alıcı kanala bir güvenlik çıkış programı yazarak bir kümenin üyeliğini sınırlayabilirsiniz. Yetkisiz kuyruk yöneticilerinin SYSTEM.CLUSTER.COMMAND.QUEUE.

Not: Uygulamaların SYSTEM.CLUSTER.TRANSMIT.QUEUE . Ayrıca, bir uygulamanın başka bir iletim kuyruğunu doğrudan açmasına izin verilmemesi de önerilir.

Kaynak güvenliği kullanıyorsanız, “z/OS üzerinde kanal başlatıcısına ilişkin güvenlikle ilgili önemli noktalar” sayfa 257içinde yer alan noktalara ek olarak aşağıdaki noktaları da göz önünde bulundurun:

Sistem kuyrukları
Kanal başlatıcı için aşağıdaki sistem kuyruklarına RACF ALTER erişimi gerekir:

- SYSTEM.CLUSTER.COMMAND KUYRUğU
- SYSTEM.CLUSTER.TRANSMIT.QUEUE.

ve SYSTEM.CLUSTER.REPOSITORY.QUEUE

Ayrıca, kümeleme için kullanılan tüm ad bilgilerine de OKUMA erişimi gerekir.

Komutlar
Uygun komut güvenliğini ayarlama (Çizelge 49 sayfa 216 içinde açıklandığı gibi) Küme desteği komutları için (REFRESH ve RESET CLUSTER, SUSPEND ve RESUME QMGR).

z/OS CICS ile IBM MQ kullanımı için güvenlikle ilgili önemli noktalar

IBM MQ 9.0.0ve sonraki tarafından desteklenen tüm CICS sürümleri, bağdaştırıcının ve köprünün CICS tarafından sağlanan sürümünü kullanır.

Güvenlikle ilgili dikkat edilmesi gereken noktalar için aşağıdaki konulara bakın:

- CICS-MQ bağdaştırıcısıiçin güvenlik.
- CICS-MQ köprüsüiçin güvenlik.

z/OS IMS ile IBM MQ kullanımı için güvenlikle ilgili önemli noktalar

IMSile IBM MQ ' yi kullanırken güvenlik gereksinimlerinizi planlamak için bu konuyu kullanın.

OPERCMDS sınıfının kullanılması

OPERCMDS sınıfındaki kaynakları korumak için RACF kullanıyorsanız, IBM MQ kuyruk yöneticisi adres alanınızla ilişkili kullanıcı kimliğinin bağlanabileceği herhangi bir IMS sisteminde MODIFY komutunu verme yetkisi olduğundan emin olun.

IMS köprüsü için güvenlikle ilgili önemli noktalar

IMS köprüsüne ilişkin güvenlik gereksinimlerinizi belirlerken göz önünde bulundurmanız gereken dört şey vardır:

- IBM MQ 'in IMS ' e bağlanması için gereken güvenlik yetkisi

- IMS ' e erişmek için köprüyü kullanan uygulamalarda ne kadar güvenlik denetimi gerçekleştirilir?
- Bu uygulamaların kullanmasına izin verilen IMS kaynakları
- Köprü tarafından konan ve elde edilen mesajlar için hangi yetkinin kullanılması gerekir?

IMS köprüsü için güvenlik gereksinimlerinizi tanımlarken aşağıdakileri göz önünde bulundurmanız gerekir:

- Köprüden geçen iletiler, güçlü güvenlik özellikleri sunmayan platformlardaki uygulamalardan kaynaklanmış olabilir.
- Köprüden geçen iletiler, aynı kuruluş ya da kuruluş tarafından denetlenmeyen uygulamalardan kaynaklanmış olabilir.

z/OS *IMS ile bağlantı kurulurken güvenlikle ilgili dikkat edilmesi gereken noktalar*

IBM MQ kuyruk yöneticisi adres alanı kullanıcı kimliğine OTMA grubu için erişim verin.

IMS köprüsü bir OTMA istemcisidir. IMS bağlantısı, IBM MQ kuyruk yöneticisi adres alanının kullanıcı kimliği altında çalışır. Olağan durumda bu, başlatılan görev grubunun bir üyesi olarak tanımlanır. Bu kullanıcı kimliğine OTMA grubu için erişim verilmelidir (/SECURE OTMA ayarı NONE değilse).

Bunu yapmak için, FACILITY sınıfında aşağıdaki tanıtımı tanımlayın:

```
IMSXCF.xcfgname.mqxcfmname
```

Burada `xcfgname` XCF grubu adı, `mqxcfmname` ise IBM MQXCF üyesi adıdır.

IBM MQ kuyruk yöneticisi kullanıcı kimliğinize bu profil için okuma erişimi vermelisiniz.

Not:

1. FACILITY sınıfındaki yetkileri değiştirirseniz, değişiklikleri etkinleştirmek için RACF komutunu SETROPTS RACLIST (FACILITY) REFRESH komutunu vermeniz gerekir.
2. MQADMIN sınıfında hlq.NO.SUBSYS.SECURITY tanıtımı varsa, IMS ' e hiçbir kullanıcı kimliği geçirilmez ve /SECURE OTMA ayarı NONE olmadıkça bağlantı başarısız olur.

z/OS *IMS köprüsü için uygulama erişim denetimi*

Her IMS sistemi için FACILITY sınıfında bir RACF profili tanımlayın. IBM MQ kuyruk yöneticisi kullanıcı kimliğine uygun bir erişim düzeyi verin.

IMS köprüsünün bağlandığı her bir IMS sistemi için, IMS sistemine geçirilen her bir ileti için ne kadar güvenlik denetiminin gerçekleştirildiğini belirlemek üzere FACILITY sınıfında aşağıdaki RACF profilini tanımlayabilirsiniz.

```
IMSXCF.xcfgname.imsxcfmname
```

Burada `xcfgname` , XCF grubu adı ve `imsxcfmname` , IMSiçin XCF üyesi adıdır. (Her IMS sistemi için ayrı bir profil tanımlamanız gerekir.)

Bu profilde IBM MQ kuyruk yöneticisi kullanıcı kimliği için izin verdiğiniz erişim düzeyi, IMS köprüsü IMS'e bağlandığında IBM MQ ' e döndürülür ve sonraki işlemlerde gerekli olan güvenlik düzeyini gösterir. Sonraki işlemler için IBM MQ , RACF 'den uygun hizmetleri ister ve kullanıcı kimliğinin yetkili olduğu yerde, iletiyi IMS' e iletir.

OTMA, IMS /SIGN komutunu desteklemez; ancak IBM MQ , gerekli denetim düzeyinin uygulanmasını etkinleştirmek için her bir iletiye ilişkin erişim denetimini ayarlamanızı sağlar.

Aşağıdaki erişim düzeyi bilgileri döndürülebilir:

YOK ya da TANıTıM BULUNAMADı
Bu değerler, maksimum güvenliğin gerekli olduğunu, yani her işlem için kimlik doğrulamasının gerekli olduğunu gösterir. MQIIH yapısının *Authenticator* alanındaki MQMD yapısının *UserIdentifier* alanında belirtilen kullanıcı kimliğinin ve parolanın ya da PassTicket ' ın RACFtarafından bilindiğini ve geçerli bir birleşim olduğunu doğrulamak için bir denetim yapılır. Parola ya da PassTicketile bir UTOKEN oluşturulur ve IMS ' e iletilir; UTOKEN önbelleğe alınmadı.

Not: hlq.NO.SUBSYS.SECURITY tanıtımı MQADMIN sınıfında varsa, bu güvenlik düzeyi tanıtımda tanımlı olan değerleri geçersiz kılar.

READ
Bu değer, aşağıdaki koşullarda NONE ile aynı kimlik doğrulamasının gerçekleştirileceğini gösterir:

- Belirli bir kullanıcı kimliğiyle ilk kez karşılaşıldığında
- Kullanıcı kimliği daha önce saptandığında, ancak önbelleğe alınan UTOKEN bir parolayla ya da PassTicket ile oluşturulmadığında

IBM MQ gerekirse bir UTOKEN ister ve IMS' e iletir.

Not: Güvenliği yeniden doğrulama isteği üzerinde işlem yapıldıysa, önbelleğe alınan tüm bilgiler kaybolur ve daha sonra her kullanıcı kimliğiyle ilk kez karşılaşıldığında bir UTOKEN istenir.

GÜNCELLE
MQMD yapısının *UserIdentifier* alanındaki kullanıcı kimliğinin RACFtarafından bilinmesi denetlenir.

Bir UTOKEN oluşturulur ve IMS ; UTOKEN önbelleğe alınır.

DENET/DEĞIŞTIR
Bu değerler, bu IMS sistemine ilişkin kullanıcı kimlikleri için güvenlik UTOKEN ' lerinin sağlanmasına gerek olmadığını gösterir. (Büyük olasılıkla bu seçeneği yalnızca geliştirme ve test sistemleri için kullanırsınız.)

Uyarı: MQMD yapısının *UserIdentifier* alanında bulunan kullanıcı kimliğinin **CONTROL/ ALTER**için hala iletildiğini unutmayın.

Not:

1. Bu erişim, IBM MQ IMSile bağlantı kurduğunda tanımlanır ve bağlantı süresi boyunca sürer. Güvenlik düzeyini değiştirmek için, güvenlik profiline erişim değiştirilmeli ve köprü durdurulup yeniden başlatılmalıdır (örneğin, OTMA durdurulup yeniden başlatılarak).
2. FACILITY sınıfındaki yetkileri değiştirirseniz, değişiklikleri etkinleştirmek için RACF komutunu SETROPTS RACLIST (FACILITY) REFRESH komutunu vermeniz gerekir.
3. Bir parola ya da PassTicketkullanabilirsiniz, ancak IMS köprüsünün verileri şifrelemediğini unutmamanız gerekir. PassTicketskullanma hakkında bilgi için bkz. “IMS üstbilgisinde RACF PassTickets ' nı kullanma” sayfa 263.
4. Bu sonuçlardan bazıları, /SECURE OTMA komutu kullanılarak IMSiçindeki güvenlik ayarlarından etkilenebilir.
5. Önbelleğe alınan UTOKEN bilgileri, IBM MQ ALTER SECURITY komutunun INTERVAL ve TIMEOUT parametreleriyle tanımlanan süre boyunca tutulur.
6. RACF UYARI seçeneğinin IMSXCF.xcfgname.imsxcfmname profili üzerinde bir etkisi yoktur. Kullanımı, verilen erişim düzeyini etkilemez ve RACF UYARI iletisi üretilmez.

z/OS *IMS üzerinde güvenlik denetimi*

Köprüden geçen iletiler güvenlik bilgilerini içerir. Yapılan güvenlik denetimleri, IMS /SECURE OTMA komutunun ayarına bağlıdır.

Köprüden geçen her IBM MQ iletisi aşağıdaki güvenlik bilgilerini içerir:

- MQMD yapısının *UserIdentifier* alanında bulunan bir kullanıcı kimliği
- MQIIH yapısının *SecurityScope* alanında bulunan güvenlik kapsamı (MQIIH yapısı varsa)

- Bir UTOKEN (IBM MQ alt sisteminin ilgili IMSXCF.`xcfgname`.`imsxcfmname` tanıtımına CONTROL ya da ALTER erişimi yoksa)

Yapılan güvenlik denetimleri IMS /SECURE OTMA komutunun ayarına bağlıdır:

/SECURE OTMA YOK
Hareket için güvenlik denetimi yapılmadı.

/GüVENLI Otma DENETIMI
MQMD yapısının *UserIdentifier* alanı, hareket ya da komut yetkisi denetimi için IMS ' e aktarılır.

IMS denetim bölgesinde bir ACEE (Erişimci Ortam Öğesi) oluşturulur.

/GüVENLI Otma TAMAMı
MQMD yapısının *UserIdentifier* alanı, hareket ya da komut yetkisi denetimi için IMS ' e aktarılır.

IMS bağımlı bölgesinde ve IMS denetim bölgesinde bir ACEE oluşturulmuştur.

/SECURE OTMA PROFILI
MQMD yapısının *UserIdentifier* alanı, hareket ya da komut yetkisi denetimi için IMS ' e geçirilir.

MQIIH yapısındaki *SecurityScope* alanı, denetim bölgesinin yanı sıra IMS bağımlı bölgesinde de ACEE oluşturulup oluşturulmayacağını belirlemek için kullanılır.

Not:

1. TIMS ya da CIMS sınıfındaki ya da GIMS ya da DIMSilişkili grup sınıflarındaki yetkileri değiştirirseniz, değişiklikleri etkinleştirmek için aşağıdaki IMS komutlarını vermeniz gerekir:
   - /MODIFY HAZıRLAMASı RACF
   - /MODIFY KESINLEşTIRME
2. /SECURE OTMA PROFILE kullanmıyorsanız, MQIIH yapısının **SecurityScope** alanında belirtilen herhangi bir değer yoksayılır.

z/OS *IMS köprüsü tarafından güvenlik denetimi yapıldı*

Gerçekleştirilmekte olan işleme bağlı olarak farklı yetkiler kullanılır.

Köprü bir ileti gönderdiğinde ya da aldığında, aşağıdaki yetkiler kullanılır:

Köprü kuyruğundan ileti alınıyor
Güvenlik denetimi gerçekleştirilmez.

Kural dışı durum koyma ya da COA rapor iletisi
MQMD yapısının *UserIdentifier* alanında kullanıcı kimliğinin yetkisini kullanır.

Yanıt iletisi koyma
Özgün iletinin MQMD yapısının *UserIdentifier* alanında kullanıcı kimliğinin yetkisini kullanır.

İleti gönderilmeyen iletiler kuyruğuna yerleştirilmesi
Güvenlik denetimi gerçekleştirilmez.

Not:

1. IBM MQ sınıf profillerini değiştirirseniz, değişiklikleri etkinleştirmek için IBM MQ REFRESH SECURITY (*) komutunu vermeniz gerekir.
2. Bir kullanıcının yetkisini değiştirirseniz, değişikliği etkinleştirmek için MQSC RVERIFY SECURITY komutunu vermeniz gerekir.

z/OS *IMS üstbilgisinde RACF PassTickets ' nı kullanma*

IMS üstbilgisinde bir parola yerine PassTicket kullanabilirsiniz.

IMS üstbilgisinde (MQIIH) bir parola yerine PassTicket kullanmak istiyorsanız, iletinin yöneltileceği IMS köprü kuyruğunun STGCLASS tanımının PASSTKTA özniteliğinde PassTicket ' ın doğrulandığı uygulama adını belirtin.

PASSTKTA değeri boş bırakılırsa, bir PassTicket oluşturulmasını ayarlamanız gerekir. Bu durumda uygulama adı MVSxxxx biçiminde olmalıdır; burada xxxx, hedef kuyruk yöneticisinin çalıştığı z/OS sisteminin SMFID 'si olur.

PassTicket , bir kullanıcı kimliği, hedef uygulama adı ve bir gizli anahtardan oluşturulur. Bu, büyük harf alfabetik ve sayısal karakterler içeren 8 baytlık bir değerdir. Yalnızca bir kez kullanılabilir ve 20 dakikalık bir süre için geçerlidir. Yerel bir RACF sistemi tarafından bir PassTicket oluşturulursa, RACF yalnızca profilin var olup olmadığını ve kullanıcının profil üzerinde yetkisi olup olmadığını denetler. PassTicket uzak bir sistemde oluşturulduysa, RACF kullanıcı kimliğinin tanıtıma erişimini doğrular. PassTicketshakkında tam bilgi için bkz. *z/OS Security Server RACF Security Administrator's Guide*.

IMS üstbilgilerindeki PassTickets , IMSdeğil, IBM MQtarafından RACF ' e verilir.

z/OS z/OS kuyruk yöneticisini büyük ve küçük harf karışık güvenliğe geçirme

Bir kuyruk yöneticisini büyük ve küçük harf karışık güvenliğe geçirmek için aşağıdaki adımları izleyin. Kullanmakta olduğunuz güvenlik ürünü düzeyini gözden geçirin ve yeni IBM MQ dış güvenlik yöneticisi sınıflarını etkinleştirin. Büyük ve küçük harf karışık profilleri etkinleştirmek için **REFRESH SECURITY** komutunu çalıştırın.

Başlamadan önce

1. Tüm IBM MQ dış güvenlik yöneticisi sınıflarının etkinleştirildiğinden emin olun.
2. Kuyruk yöneticinizin başlatıldığından emin olun.

Bu görev hakkında

Bir kuyruk yöneticisini büyük ve küçük harf karışık güvenliğe dönüştürmek için aşağıdaki adımları izleyin.

Yordam

1. Büyük harfli sınıflardan var olan tüm tanıtımlarınızı ve erişim düzeylerinizi eşdeğer büyük harfli dış güvenlik yöneticisi sınıfına kopyalayın.
   a) `MQADMIN` - `MXADMIN`.
   b) `MQPROC` - `MXPROC`.
   c) `MQNLIST` - `MXNLIST`.
   d) `MQQUEUE` - `MXQUEUE`.
2. Aşağıdaki komutu vererek `SCYCASE` kuyruk yöneticisi özniteliğinin değerini `MIXED` olarak değiştirin.

```
ALTER QMGR SCYCASE(MIXED)
```

3. Aşağıdaki komutu vererek güvenlik profillerini etkinleştirin.

```
REFRESH SECURITY(*) TYPE(CLASSES)
```

4. Güvenlik profillerinizin doğru çalışıp çalışmadığını test edin.

Sonraki adım

Nesne tanımlamalarınızı gözden geçirin ve profilleri etkinleştirmek için gereken şekilde **REFRESH SECURITY** komutunu kullanarak yeni büyük ve küçük harf karışık profiller oluşturun.

IBM MQ MQI client güvenliğinin ayarlanması

İstemci uygulamalarının sunucudaki kaynaklara sınırsız erişimi olmaması için IBM MQ MQI client güvenliğini göz önünde bulundurmanız gerekir.

Bir istemci uygulamasını çalıştırırken, uygulamayı gerekenden fazla erişim haklarına sahip bir kullanıcı kimliği kullanarak çalıştırmayın; örneğin, `mqm` grubundaki bir kullanıcı ya da `mqm` kullanıcısının kendisi.

Bir uygulamayı çok fazla erişim hakkı olan bir kullanıcı olarak çalıştırarak, uygulamanın kuyruk yöneticisinin kısımlarına erişmesi ve bu kısımları değiştirmesi riskini yanlışlıkla ya da kötü amaçlı olarak çalıştırabilirsiniz.

Bir istemci uygulaması ile kuyruk yöneticisi sunucusu arasında güvenliğin iki yönü vardır: kimlik doğrulaması ve erişim denetimi.

- Kimlik doğrulaması, belirli bir kullanıcı olarak çalışan istemci uygulamasının, olduğunu söyledikleri kişi olduğundan emin olmak için kullanılabilir. Kimlik doğrulamasını kullanarak, bir saldırganın uygulamalarınızdan birinin kimliğine bürünerek kuyruk yöneticinize erişmesini engelleyebilirsiniz.

  IBM MQ 8.0' den kimlik doğrulaması iki seçenekten biriyle sağlanır:

  – Bağlantı kimlik doğrulama özelliği.

    Bağlantı kimlik doğrulamasıyla ilgili daha fazla bilgi için bkz. “Bağlantı kimlik doğrulaması” sayfa 69.

  – TLS içinde karşılıklı kimlik doğrulaması kullanılıyor.

    TLS hakkında daha fazla bilgi için bkz. “SSL/TLS ile çalışma” sayfa 271.

- Erişim denetimi, belirli bir kullanıcıya ya da kullanıcı grubuna ilişkin erişim haklarını vermek ya da kaldırmak için kullanılabilir. Bir istemci uygulamasını özel olarak yaratılmış bir kullanıcıyla (ya da belirli bir gruptaki bir kullanıcıyla) çalıştırarak, uygulamanın kuyruk yöneticinizin, uygulamanın erişmemesi gereken kısımlarına erişememesini sağlamak için erişim denetimlerini kullanabilirsiniz.

  Erişim denetimini ayarlarken, kanal kimlik doğrulama kurallarını ve bir kanaldaki MCAUSER alanını göz önünde bulundurmanız gerekir. Bu özelliklerin her ikisi de, erişim denetimi haklarını doğrulamak için hangi kullanıcı kimliğinin kullanıldığını değiştirme yeteneğine sahiptir.

  Erişim denetimine ilişkin daha fazla bilgi için bkz. “Nesnelere erişim yetkisi verilmesi” sayfa 366.

Sınırlı kimlikli belirli bir kanala bağlanmak için bir istemci uygulaması ayarladıysanız, ancak kanal MCAUSER alanında bir denetimci kimliği ayarlıysa, istemci uygulamasının başarıyla bağlanması koşuluyla, erişim denetimi denetimleri için denetimci kimliği kullanılır. Bu nedenle, istemci uygulaması kuyruk yöneticinize tam erişim haklarına sahip olacaktır.

MCAUSER özniteliğine ilişkin daha fazla bilgi için bkz. “Bir istemci kullanıcı kimliğinin MCAUSER kullanıcı kimliğiyle eşlenmesi” sayfa 401.

Kanal kimlik doğrulama kuralları, kabul edilecek bir bağlantıya ilişkin belirli kurallar ve ölçütler ayarlanarak bir kuyruk yöneticisine erişimi denetlemek için bir yöntem olarak da kullanılabilir.

Kanal kimlik doğrulama kurallarıyla ilgili daha fazla bilgi için bkz. “Kanal kimlik doğrulama kayıtları” sayfa 50.

MQI istemcisinde çalıştırma zamanında yalnızca FIPS onaylı CipherSpecs kullanılmasının belirtilmesi

FIPS uyumlu yazılımı kullanarak anahtar havuzlarınızı oluşturun ve daha sonra, kanalın FIPS onaylı CipherSpecsözelliğini kullanması gerektiğini belirtin.

Not: AIX, Linux, and Windowsişletim sistemlerinde IBM MQ , IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, IBM Crypto for C (ICC) sertifikasını görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu İşlem listesindeki NIST CMVP modüllerindearanarak görüntülenebilir.

Çalıştırma zamanında FIPS uyumlu olmak için, anahtar havuzlarının yalnızca `-fips` seçeneğiyle **`runmqakm`** gibi FIPS uyumlu yazılımlar kullanılarak oluşturulmuş ve yönetilmiş olması gerekir.

TLS kanalının yalnızca FIPS onaylı CipherSpecs özelliğini öncelik sırasına göre listelenmiş üç şekilde kullanması gerektiğini belirtebilirsiniz:

1. MQSCO yapısındaki FipsRequired alanını MQSSL_FIPS_YESolarak ayarlayın.
2. **MQSSLFIPS** ortam değişkenini YESdeğerine ayarlayın.
3. İstemci yapılandırma dosyasının SSL kısmına ilişkin **SSLFipsRequired** özniteliğini YESdeğerine ayarlayın.

Varsayılan olarak, FIPS onaylı CipherSpecs gerekmez.

Bu değerler, **ALTER QMGR SSLFIPS** ' daki eşdeğer parametre değerleriyle aynı anlamlara sahiptir (bkz. **ALTER QMGR** (kuyruk yöneticisi ayarlarının değiştirilmesi)). İstemci işleminin etkin TLS bağlantısı yoksa ve SSL MQCONNX 'te geçerli bir FipsRequired değeri belirtildiyse, bu işlemle ilişkili sonraki tüm TLS bağlantıları yalnızca bu değerle ilişkilendirilmiş CipherSpecs kullanmalıdır. Bu, bu ve diğer tüm TLS bağlantıları duruncaya kadar geçerlidir; bu aşamada sonraki bir MQCONNX, FipsRequirediçin yeni bir değer sağlayabilir.

Şifreleme donanımı varsa, IBM MQ tarafından kullanılan şifreleme modülleri donanım ürünü tarafından sağlanan modüller olacak şekilde yapılandırılabilir ve bunlar belirli bir düzeyde FIPS onaylı olabilir. Yapılandırılabilir modüller ve bunların FIPS onaylı olup olmamaları, kullanılmakta olan donanım ürününe bağlıdır.

Olanaklıysa, yalnızca FIPS CipherSpecs yapılandırılırsa, MQI istemcisi MQRC_SSL_INITIALIZATION_ERRORile FIPS olmayan CipherSpec belirten bağlantıları reddeder. IBM MQ , bu tür bağlantıların tümünü reddetmeyi garanti etmez ve IBM MQ yapılandırmanızın FIPS uyumlu olup olmadığını belirlemek sizin sorumluluğunuzdadır.

İlgili kavramlar

“AIX, Linux, and Windows için Federal Bilgi İşleme Standartları (FIPS)” sayfa 34
AIX, Linux, and Windows sistemlerinde bir SSL/TLS kanalında şifreleme gerektiğinde, IBM MQ IBM Crypto for C (ICC)adlı bir şifreleme paketi kullanır. AIX, Linux, and Windows platformlarında ICC yazılımı, ABD Ulusal Standartlar ve Teknoloji Enstitüsü 'nün Federal Bilgi İşleme Standartları (FIPS) Şifreleme Modülü Doğrulama Programı 'nı 140-2 düzeyinde geçti.

AIX TLS istemci uygulamalarını AIX üzerinde birden çok GSKit 8.0 kuruluşu ile çalıştırma

AIX üzerinde TLS istemci uygulamaları, birden çok IBM Global Security Kit (GSKit) sürüm 8.0 kuruluşu olan AIX sistemlerinde çalışırken MQRC_CHANNEL_CONFIG_ERROR ve hata AMQ6175 ile karşılaşabilir.

Birden çok GSKit 8.0 kuruluşu olan bir AIX sisteminde istemci uygulamalarını çalıştırırken, TLS kullanılırken istemci bağlantısı çağrıları MQRC_CHANNEL_CONFIG_ERROR ' i döndürebilir. Arızalı istemci uygulamasına ilişkin /var/mqm/errors günlük kaydı hatası AMQ6175 ve AMQ9220 , örneğin:

```
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 ----------------------------------------------------------
```

Bu hatanın yaygın bir nedeni, LIBPATH ya da LD_LIBRARY_PATH ortam değişkeninin ayarının IBM MQ istemcisinin iki farklı GSKit 8.0 kuruluşundan karışık kitaplık kümesini yüklemesine neden olması olabilir. Db2 ortamında bir IBM MQ istemci uygulamasının yürütülmesi bu hataya neden olabilir.

Bu hatayı önlemek için, kitaplık yolunun önüne IBM MQ kitaplık dizinlerini ekleyin. Böylece, IBM MQ kitaplıkları öncelikli olur. Bu, **-k** parametresiyle **setmqenv** komutu kullanılarak gerçekleştirilebilir; örneğin:

```
. /usr/mqm/bin/setmqenv -s -k
```

setmqenv komutunun kullanımıyla ilgili daha fazla bilgi için bkz. setmqenv (set IBM MQ environment)

TLS kanallarını MQSC ile yapılandırma

TLS kanallarını yapılandırmak için **runmqsc** ve ALTER CHANNEL komutlarını kullanın. İsteğe bağlı olarak, bir kanalı yalnızca belirtilen değerlerle eşleşen sahibin ayırt edici adında öznitelikleri olan sertifikaları kabul edecek şekilde yapılandırabilirsiniz. Ayrıca, isteğe bağlı olarak, başlatma tarafı kendi kişisel sertifikasını göndermezse kuyruk yöneticisinin bağlantıyı reddetmesi için bir kuyruk yöneticisi kanalı da yapılandırabilirsiniz.

Bu görev hakkında

IBM MQ Exploreriçinde kanalları yapılandırmak için bkz. IBM MQ Explorerile TLS kanallarını yapılandırma.

Kanalları **runmqsc**kullanarak yapılandırmak için aşağıdaki adımları tamamlayın.

Yordam

1. Hedef kuyruk yöneticisine bağlanan **runmqsc** komutunu çağırın.
2. TLS için etkinleştirmek istediğiniz kanalı tanımlayın.
   Kanal adını ve kanal tipini not edin.
3. Bir IBM MQ kanalının çeşitli özelliklerini değiştirmek için ALTER CHANNEL komutunu kullanın.
   Komuta ek olarak kanal adını ve kanal tipini de girmeniz gerekir. Örneğin, MQ.TEST şu komutu çalıştırın:

   ```
   ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR)
   ```

   TLS ile ilgili, IBM MQ kanal tanımlarında ayarlayabileceğiniz çeşitli kanal öznitelikleri vardır.

Sonraki adım

İleti güvenliğinin ayarlanması

TLS etkin ileti sistemi, ileti güvenliğini sağlamak amacıyla iki yöntem sunar:

- Şifreleme, iletiye müdahale edilirse iletinin okunamaz olmasını sağlar.
- Hash işlevleri, ileti değiştirildiğinde bunun saptanmasını sağlar.

Bu yöntemlerin birleşimine şifre belirtimi ya da CipherSpecdenir. Aynı CipherSpec bir kanalın her iki ucu için de ayarlanmalıdır, aksi takdirde TLS etkin ileti sistemi başarısız olur. Daha fazla bilgi için bkz “güvenlikIBM MQ” sayfa 7.

Bir IBM MQ kanalını etkinleştirme TLS ' yi değiştirmek için SSLCIPH özniteliğinde bir değer belirtin. Bu öznitelik, “CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432listesindeki kuyruk yöneticisinin kuyruk altyapısı için geçerli bir CipherSpec değerine ayarlanmalıdır.

Bir IBM MQ kanalını TLS 'yi devre dışı bırakacak şekilde değiştirmek için SSLCIPH' yi boş bir değere ayarlayın. Örneğin:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCIPH(ANY_TLS12_OR_HIGHER)
```

Not: Büyük ve küçük harf karakterinin korunduğundan emin olmak için kanal adını tek tırnak içine almanız gerekir. Tek tırnak işareti olmadan IBM MQ , dizgiyi büyük harf olacak şekilde dönüştürür.

Sahibin adına göre sertifikaları süzme

Sertifikalar, sertifikanın sahibinin ayırt edici adını içerir. İsteğe bağlı olarak kanalı, yalnızca belirli değerlerle eşleşen sahibin ayırt edici adında öznitelikleri olan sertifikaları kabul edecek şekilde yapılandırabilirsiniz.

IBM MQ ' in süzgeçten geçirebileceği öznitelik adları aşağıdaki tabloda listelenir:

Öznitelik adları	**Anlamı**
SERI NUMARASı	Sertifika seri numarası
POSTA	E-posta adresi
Deprecated P	E-posta adresi (MAIL tercihine göre kullanımdan kaldırıldı)
UID ya da USERID	Kullanıcı kimliği
CN	Ortak Ad
T	Başlık
Kuruluş Birimi	Kuruluş Birimi adı
DAĞITIM MERKEZİ	Etki alanı bileşeni
O	Kuruluş adı
Sokak	Açık/İlk adres satırı
L	İlçe adı
ST (ya da SP ya da S)	Eyalet ya da İl adı
PC	Posta kodu/posta kodu
C	Ülke
UNSTRUCTUREDNAME	Anasistem adı
YAPıLMAMış ELBISE	IP adresi
DNQ	Ayırt edici ad niteleyicisi

Herhangi bir sayıda karakter yerine öznitelik değerinin başında ya da sonunda genel arama karakterini (*) kullanabilirsiniz. Örneğin, GBiçinde `Smith` çalışma `IBM` ile biten bir ada sahip bir kişinin yalnızca sertifikalarını kabul etmek için şunu yazın:

```
CN=*Smith, O=IBM, C=GB
```

Örneğin:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLPEER('CN=*Smith, O=IBM, C=GB')
```

Not: Karakter büyük ve küçük harf durumunu korumak için SSLPEER dizgisini tek tırnak içine almanız gerekir. Tek tırnak işareti olmadan IBM MQ , dizgiyi büyük harf olacak şekilde dönüştürür.

Kuyruk yöneticisiyle bağlantı başlatan tarafların kimliklerinin doğrulanması

Başka bir taraf bir kuyruk yöneticisine TLS etkin bir bağlantı başlattığında, kuyruk yöneticisinin kimlik kanıtı olarak kişisel sertifikasını başlatan tarafa göndermesi gerekir. İsteğe bağlı olarak kuyruk yöneticisi kanalını, başlatan taraf kendi kişisel sertifikasını göndermezse kuyruk yöneticisinin bağlantıyı reddetmesi için de yapılandırabilirsiniz.

Bunu yapmak için SSLCAUTH özniteliğini ayarlayın. Bu öznitelik bir Boole özniteliğidir ve OPTIONAL ya da REQUIRED değerlerini içerebilir:

- İSTEğE BAğLı İSTEğE BAğLı bir istemci sağlandıysa, ancak bir istemcinin göndermesini gerektirmiyorsa, istemcinin sertifikasını doğrular. Bir istemci, geçerli olmayan bir sertifika gönderirse reddedilir.
- REQUIRED, geçerli bir TLS sertifikası sağlamayan bağlanan istemcileri reddeder

Örneğin:

```
ALTER CHANNEL('MQ.TEST') CHLTYPE(SDR) SSLCAUTH(REQUIRED)
```

IBM i IBM i üzerinde SSL ya da TLS için iletişim ayarlanması

SSL ya da TLS şifreleme güvenliği iletişim kurallarını kullanan güvenli iletişim, iletişim kanallarının ayarlanmasını ve kimlik doğrulaması için kullanacağınız dijital sertifikaların yönetilmesini içerir.

SSL ya da TLS kuruluşunuzu ayarlamak için kanallarınızı SSL ya da TLS kullanacak şekilde tanımlamanız gerekir. Ayrıca dijital sertifikalarınızı oluşturmalı ve yönetmelisiniz. Bazı işletim sistemlerinde, testleri kendinden onaylı sertifikalarla gerçekleştirebilirsiniz. Ancak IBM iüzerinde, yerel bir CA tarafından imzalanmış kişisel sertifikaları kullanmanız gerekir.

Sertifika oluşturma ve yönetme hakkında tam bilgi için bkz. "IBM i üzerinde SSL/TLS ile çalışma" sayfa 271.

Bu konu derlemi, SSL ya da TLS iletişimi kurulmasına ilişkin bazı görevleri tanıtır ve bu görevlerin tamamlanmasına ilişkin adım adım yol gösterir.

SSL ya da TLS istemci kimlik doğrulamasını da sınamak isteyebilirsiniz; bunlar SSL ve TLS iletişim kurallarının isteğe bağlı kısımlarıdır. SSL ya da TLS anlaşması sırasında, SSL ya da TLS istemcisi her zaman sunucudan bir dijital sertifika alır ve doğrular. IBM MQ uygulamasıyla, SSL ya da TLS sunucusu her zaman istemciden bir sertifika ister.

IBM iüzerinde, SSL ya da TLS istemcisi yalnızca doğru IBM MQ biçiminde etiketlenmiş bir sertifika gönderir:

- Bir kuyruk yöneticisi için, `ibmwebspheremq` ve ardından kuyruk yöneticinizin adı küçük harfe çevrildi. Örneğin, `QM1`, `ibmwebspheremqqm1`için.
- Bir IBM MQ C Client for IBM iiçin, `ibmwebspheremq` ve ardından oturum açma kullanıcı kimliğiniz küçük harfe çevrildi; örneğin, `ibmwebspheremqmyuserid`.

IBM MQ , diğer ürünlere ilişkin sertifikalarla karışıklığı önlemek için bir etikette `ibmwebspheremq` önekini kullanır. Tüm sertifika etiketini küçük harfle belirttiğinizden emin olun.

SSL ya da TLS sunucusu, gönderildiyse, istemci sertifikasını her zaman doğrular. SSL ya da TLS istemcisi bir sertifika göndermezse, SSL ya da TLS sunucusu olarak hareket eden kanalın sonu, SSLCAUTH parametresi REQUIRED ya da bir SSLPEER parametre değeri ayarlanmış olarak tanımlanırsa kimlik doğrulaması başarısız olur. Daha fazla bilgi için SSL ya da TLS kullanarak iki kuyruk yöneticisinin bağlanmasıbaşlıklı konuya bakın.

ALW AIX, Linux, and Windows üzerinde SSL ya da TLS için iletişim ayarlanması

SSL ya da TLS şifreleme güvenliği iletişim kurallarını kullanan güvenli iletişim, iletişim kanallarının ayarlanmasını ve kimlik doğrulaması için kullanacağınız dijital sertifikaların yönetilmesini içerir.

SSL ya da TLS kuruluşunuzu ayarlamak için kanallarınızı SSL ya da TLS kullanacak şekilde tanımlamanız gerekir. Ayrıca dijital sertifikalarınızı oluşturmalı ve yönetmelisiniz. AIX, Linux, and Windows sistemlerinde, testleri kendinden onaylı sertifikalarla gerçekleştirebilirsiniz.

Uyarı: TLS etkin kanalları kullanarak birleştirmek istediğiniz kuyruk yöneticilerindeki Elliptic Curve imzalı sertifikalarla RSA imzalı sertifikaların bir karışımını kullanmak mümkün değildir.

TLS etkinleştirilmiş kanalları kullanan kuyruk yöneticilerinin tümü RSA imzalı sertifikaları kullanmalı ya da hepsinin her ikisinin bir karışımı değil, EC imzalı sertifikaları kullanmalıdır.

Ek bilgi için bkz. “IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu” sayfa 46 .

Kendinden imzalı sertifikalar geri alınamaz; bu, bir saldırganın özel bir anahtar açığa çıktıktan sonra kimlik sahteciliği yapmasına izin verebilir. Sertifika yetkilisi, ihlal edilen bir sertifikayı iptal edebilir ve bu da sertifikanın daha fazla kullanılmasını önler. Bu nedenle CA imzalı sertifikaların üretim ortamında kullanılması daha güvenlidir, ancak kendinden imzalı sertifikalar test sistemi için daha uygundur.

Sertifika oluşturma ve yönetme hakkında tam bilgi için bkz. “AIX, Linux, and Windows üzerinde SSL/TLS ile çalışma” sayfa 289.

Bu konu grubu, SSL iletişimini ayarlamaya ilişkin bazı görevleri tanıtır ve bu görevlerin tamamlanmasına ilişkin adım adım yol gösterir.

İletişim kurallarının isteğe bağlı bir parçası olan SSL ya da TLS istemci kimlik doğrulamasını da test etmek isteyebilirsiniz. SSL ya da TLS anlaşması sırasında, SSL ya da TLS istemcisi her zaman sunucudan bir dijital sertifika alır ve doğrular. IBM MQ uygulamasıyla, SSL ya da TLS sunucusu her zaman istemciden bir sertifika ister.

AIX, Linux, and Windowsüzerinde, SSL ya da TLS istemcisi yalnızca doğru IBM MQ biçiminde etiketli bir sertifika varsa bir sertifika gönderir:

- Bir kuyruk yöneticisinde, `ibmwebspheremq` biçiminin ardından kuyruk yöneticinizin adı küçük harfe çevrilir. Örneğin, `QM1`için, `ibmwebspheremqqm1`
- Bir IBM MQ istemcisi için `ibmwebspheremq` ve ardından oturum açma kullanıcı kimliğiniz küçük harfe çevrildi; örneğin, `ibmwebspheremqmyuserid`.

IBM MQ , diğer ürünlere ilişkin sertifikalarla karışıklığı önlemek için bir etikette `ibmwebspheremq` önekini kullanır. Tüm sertifika etiketini küçük harfle belirttiğinizden emin olun.

SSL ya da TLS sunucusu, gönderildiyse, istemci sertifikasını her zaman doğrular. İstemci bir sertifika göndermezse, kimlik doğrulaması yalnızca SSL ya da TLS sunucusu olarak hareket eden kanalın sonu, SSLCAUTH parametresi REQUIRED olarak ya da bir SSLPEER parametre değeri olarak ayarlandığında başarısız olur. Daha fazla bilgi için SSL ya da TLS kullanarak iki kuyruk yöneticisinin bağlanmasıbaşlıklı konuya bakın.

z/OS z/OS üzerinde SSL ya da TLS için iletişim ayarlanması

SSL ya da TLS şifreleme güvenliği iletişim kurallarını kullanan güvenli iletişim, iletişim kanallarının ayarlanmasını ve kimlik doğrulaması için kullanacağınız dijital sertifikaların yönetilmesini içerir.

SSL ya da TLS kuruluşunuzu ayarlamak için kanallarınızı SSL ya da TLS kullanacak şekilde tanımlamanız gerekir. Ayrıca dijital sertifikalarınızı oluşturmalı ve yönetmelisiniz. z/OS üzerinde, sınamaları kendinden onaylı sertifikalarla ya da bir yerel sertifika yetkilisi (CA) tarafından imzalanmış kişisel sertifikalarla gerçekleştirebilirsiniz.

Kendinden imzalı sertifikalar geri alınamaz; bu, bir saldırganın özel bir anahtar açığa çıktıktan sonra kimlik sahteciliği yapmasına izin verebilir. Sertifika yetkilisi, ihlal edilen bir sertifikayı iptal edebilir ve

bu da sertifikanın daha fazla kullanılmasını önler. Bu nedenle CA imzalı sertifikaların üretim ortamında kullanılması daha güvenlidir, ancak kendinden imzalı sertifikalar test sistemi için daha uygundur.

Sertifika oluşturma ve yönetme hakkında tam bilgi için bkz. “z/OS üzerinde SSL/TLS ile çalışma” sayfa 327.

Ek bilgi için ALTER QMGR komutunun CERTLABL ve CERTQSGL değiştirgelerine ve DEFINE CHANNEL komutunun CERLABL değiştirgesine bakın.

Öncelik sırası:

- Kanal CERTLABL parametresi
- Kanal paylaşılıyorsa QMGR CERTQSGL parametresi.

  Gönderen kanal için, iletim kuyruğunun (XMITQ) paylaşıldığı anlamına gelir. Bir alıcı kanal için, kanal paylaşılan dinleyici aracılığıyla başlatıldı, yani INDISP (GROUP) içeren dinleyici.
- QMGR CERTLABL
- Varsayılan `ibmWebSphereMQ` etiketi ve ardından paylaşılan kanallar için kuyruk paylaşım grubunun adı ya da kuyruk yöneticisinin adı gelir.

Bu konu derlemi, SSL ya da TLS iletişimi kurulmasına ilişkin bazı görevleri tanıtır ve bu görevlerin tamamlanmasına ilişkin adım adım yol gösterir.

İletişim kurallarının isteğe bağlı bir parçası olan SSL ya da TLS istemci kimlik doğrulamasını da test etmek isteyebilirsiniz. SSL ya da TLS anlaşması sırasında, SSL ya da TLS istemcisi her zaman sunucudan bir dijital sertifika alır ve doğrular. IBM MQ uygulamasıyla, SSL ya da TLS sunucusu her zaman istemciden bir sertifika ister.

Kanal paylaşılıyorsa, kanal önce kuyruk paylaşım grubu için bir sertifika bulmaya çalışır. Bir kuyruk paylaşım grubuna ilişkin bir sertifika bulamazsa, kuyruk yöneticisine ilişkin bir sertifika bulmaya çalışır.

z/OSsistemlerinde IBM MQ , diğer ürünlere ilişkin sertifikalarla karışıklığı önlemek için bir etikette `ibmWebSphereMQ` önekini kullanır.

SSL ya da TLS sunucusu, gönderildiyse, istemci sertifikasını her zaman doğrular. SSL ya da TLS istemcisi bir sertifika göndermezse, SSL ya da TLS sunucusu olarak hareket eden kanalın sonu, SSLCAUTH parametresi REQUIRED ya da bir SSLPEER parametre değeri ayarlanmış olarak tanımlanırsa kimlik doğrulaması başarısız olur. Daha fazla bilgi için SSL ya da TLS kullanarak iki kuyruk yöneticisinin bağlanmasıbaşlıklı konuya bakın.

SSL/TLS ile çalışma

Bu konularda, TLS ' nin IBM MQile kullanılmasına ilişkin tek görevlerin gerçekleştirilmesine ilişkin yönergeler yer alır.

Bunların çoğu, aşağıdaki bölümlerde açıklanan üst düzey görevlerde adım olarak kullanılır:

- “Kullanıcıların tanımlanması ve kimliğinin doğrulanmesi” sayfa 339
- “Nesnelere erişim yetkisi verilmesi” sayfa 366
- “İletilerin gizliliği” sayfa 432
- “İletilerin veri bütünlüğü” sayfa 489
- “Kümelerin güvenliğini sağlama” sayfa 490

IBM i IBM i üzerinde SSL/TLS ile çalışma

Bu konu derlemi, IBM MQ for IBM iiçinde TLS (Transport Layer Security; İletim Katmanı Güvenliği) ile çalışan tek tek görevlere ilişkin yönergeler sağlar.

IBM i için TLS desteği, işletim sisteminin ayrılmaz bir parçasıdır. IBM i üzerinde donanım ve yazılım gereksinimleriiçinde listelenen önkoşulları kurduğunuzdan emin olun.

IBM iüzerinde, Digital Certificate Manager (DCM) aracıyla anahtarları ve sayısal sertifikaları yönetebilirsiniz.

DCM ' ye Erişilmesi
DCM arabirimine erişmek için bu yönergeleri izleyin.

Bu görev hakkında
Çerçeveleri destekleyen bir web tarayıcısında aşağıdaki adımları gerçekleştirin.

Yordam
1. `http://`*`machine`*`.domain:2001` ya da `https://`*`machine`*`.domain:2010`adresine gidin; burada *makine* bilgisayarınızın adıdır.
2. İstendiğinde geçerli bir kullanıcı tanıtımı ve parola yazın.

   Yeni sertifika depoları yaratmanızı sağlamak için kullanıcı tanıtımınızda *ALLOBJ ve *SECADM özel yetkilerinin bulunduğundan emin olun. Özel yetkilere sahip değilseniz, yalnızca kişisel sertifikalarınızı yönetebilir ya da yetkili olduğunuz nesnelere ilişkin nesne imzalarını görüntüleyebilirsiniz. Bir nesne imzalama uygulamasını kullanma yetkiniz varsa, DCM ' den nesneleri de imzalayabilirsiniz.
3. Internet Yapılandırmaları sayfasında **Digital Certificate Manager**(Dijital Sertifika Yöneticisi) seçeneğini tıklatın.

   Digital Certificate Manager (Sertifika Yöneticisi) sayfası görüntülenir.

IBM i üzerinde bir kuyruk yöneticisine sertifika atanması
Bir kuyruk yöneticisine sertifika atamak için DCM ' yi kullanın.

Bir kuyruk yöneticisine sertifika atamak için geleneksel IBM i dijital sertifika yönetimini kullanın. Bu, bir kuyruk yöneticisinin sistem sertifika deposunu kullandığını ve kuyruk yöneticisinin Digital Certificate Managerile uygulama olarak kullanılmak üzere kaydedildiğini belirtebileceğiniz anlamına gelir. Bunu yapmak için, kuyruk yöneticisi **SSLKEYR** özniteliğinin değerini *SYSTEMolarak değiştirin.

SSLKEYR parametresi *SYSTEMolarak değiştirildiğinde IBM MQ , kuyruk yöneticisini sunucu uygulaması olarak QIBM_WEBSPHERE_MQ_QMGRNAME benzersiz uygulama etiketiyle ve Qmgrname (WMQ) açıklamasına sahip bir etiketle kaydeder. *SYSTEM sertifika deposunu kullanıyorsanız, kanal **CERTLABL** özniteliklerinin kullanılmadığını unutmayın. Kuyruk yöneticisi daha sonra Digital Certificate Manager' da bir sunucu uygulaması olarak görünür ve bu uygulamaya sistem deposundaki herhangi bir sunucu ya da istemci sertifikası atayabilirsiniz.

Kuyruk yöneticisi bir uygulama olarak kayıtlı olduğundan, DCM ' nin gelişmiş özellikleri (CA güven listelerinin tanımlanması gibi) gerçekleştirilebilir.

SSLKEYR parametresi *SYSTEMdışında bir değere değiştirilirse, IBM MQ , kuyruk yöneticisini Digital Certificate Managerile bir uygulama olarak kaydettirir. Bir kuyruk yöneticisi silinirse, DCM ' den de kaydı silinir. Yeterli *SECADM yetkisine sahip bir kullanıcı, DCM ' ye el ile uygulama ekleyebilir ya da kaldırabilir.

IBM i üzerinde bir anahtar havuzu ayarlama
Bağlantının her iki ucunda da bir anahtar havuzu ayarlanmalıdır. Varsayılan sertifika depoları kullanılabilir ya da kendi sertifika depolarınızı yaratabilirsiniz.

TLS bağlantısı, bağlantının her sonunda bir *anahtar havuzu* gerektirir. Her kuyruk yöneticisinin ve IBM MQ MQI client ' in bir anahtar havuzuna erişimi olmalıdır. Anahtar havuzuna bir dosya adı ve parola (*SYSTEM seçeneğini kullanmayan) kullanarak erişmek istiyorsanız, QMQM kullanıcı tanıtımının aşağıdaki yetkilere sahip olduğundan emin olun:

- Anahtar havuzunu içeren dizin için yürütme yetkisi
- Anahtar havuzunu içeren dosya için okuma yetkisi

Ek bilgi için bkz. “SSL/TLS anahtar havuzu” sayfa 25 . *SYSTEM sertifika deposunu kullanıyorsanız, kanal **CERTLABL** özniteliklerinin kullanılmadığını unutmayın.

IBM iişletim sisteminde dijital sertifikalar, DCM ile yönetilen bir sertifika deposunda saklanır. Bu dijital sertifikaların, bir sertifikayı bir kuyruk yöneticisi ya da IBM MQ MQI clientile ilişkilendirebilen etiketleri vardır. TLS, kimlik doğrulama amacıyla sertifikaları kullanır.

Etiket, **CERTLABL** özniteliğinin değeri (ayarlanmışsa) ya da kuyruk yöneticisinin adı ya da sonuna IBM MQ MQI client kullanıcı oturum açma kimliği eklenmiş olarak varsayılan `ibmwebspheremq` değeridir (tümü küçük harfli olarak). Ayrıntılar için bkz. Dijital sertifika etiketleri .

Kuyruk yöneticisi ya da IBM MQ MQI client sertifika deposu adı bir yol ve kök addan oluşuyor. Varsayılan yol `/QIBM/UserData/ICSS/Cert/Server/` ve varsayılan kök adı `Default`' dır. IBM isistemlerinde varsayılan sertifika deposu `/QIBM/UserData/ICSS/Cert/Server/Default.kdb`, `*SYSTEM`olarak da bilinir. İsteğe bağlı olarak, kendi yol ve kök adınızı tanımlayabilirsiniz.

Kendi yolunuzu ya da dosya adınızı tanımlarsanız, dosyaya erişimi sıkı bir şekilde denetlemek için dosyanın izinlerini ayarlayın.

“IBM i üzerinde bir kuyruk yöneticisi için anahtar havuzu konumunu değiştirme” sayfa 275 , sertifika deposu adının belirtilmesine ilişkin bilgi verir. Sertifika deposunu yaratmadan önce ya da yarattıktan sonra sertifika deposu adını belirtebilirsiniz.

Not: DCM ile gerçekleştirebileceğiniz işlemler, kullanıcı tanıtımınızın yetkisiyle sınırlı olabilir. Örneğin, bir CA sertifikası yaratmak için *ALLOBJ ve *SECADM yetkilerinin olması gerekir.

V 9.3.0 IBM i V 9.3.0 *IBM i üzerinde anahtar havuzu parolalarını şifreleme*

Bazı IBM MQ bileşenlerinin sayısal sertifikaları ya da simetrik anahtarları içeren bir anahtar havuzuna erişmesi gerekir. Bir anahtar havuzu, hassas bilgiler içerdiği için bir parolayla korunmuştur. Anahtar havuzu parolası, anahtar havuzuna erişildiğinde IBM MQ ' un okuyabileceği bir konumda saklanmalıdır. Anahtar havuzuna yetkisiz erişim olasılığını azaltmak için parola da şifrelenmelidir.

Aşağıdaki IBM MQ bileşenleri ve özellikleri, anahtar havuzu parolalarını saklamak için iki farklı yöntemi destekler:

- Kuyruk yöneticisi TLS anahtar havuzu.
- TLS kullanan IBM MQ MQI clients .

Bu bileşenler tarafından kullanılan anahtar havuzu parolaları, IBM MQ parola koruma sistemi kullanılarak korunur. Parola sağlama ve şifreleme mekanizması bileşene bağlı olarak değişiklik gösterir:

Kuyruk yöneticisi TLS anahtar havuzu

Parola, **SSLKEYRPWD** kuyruk yöneticisi özniteliği CHGMQM (İleti Kuyruğu Yöneticisini Değiştir) komutu kullanılarak ayarlandığında şifrelenir.

Parola, AES-128 algoritmasıyla şifrelenir. Bu algoritmanın ayrıntıları kamuya açıktır ve güvenli olarak kabul edilir.

Parola, anahtar havuzuna erişebilecek diğer yazılımlar tarafından anlaşılmayan özel bir biçimde bir parola saklama dosyasında saklanır.

Bir IBM MQ bileşeni tarafından şifrelenen bir parola farklı bir IBM MQ bileşeni tarafından kullanılamaz.

Anahtar havuzu parolası şifrelendiğinde benzersiz bir şifreleme anahtarı sağlanabilir. Benzersiz bir şifreleme anahtarı, şifreleme anahtarına erişimi olmayan herkesin parolanın şifresini çözmesini önler. Bu anahtarı, şifrelenecek bir parola sağlamadan önce ayarlanması gereken **INITKEY** kuyruk yöneticisi özniteliği aracılığıyla sağlarsanız.

IBM MQ parola koruma sistemi hakkında daha fazla bilgi için bkz. “IBM MQ bileşeni yapılandırma dosyalarındaki parolaları koruma” sayfa 575.

TLS kullanan IBM MQ MQI clients

“IBM MQ SSL Client Utility (amqrsslc) for IBM i” sayfa 286 , anahtar havuzu parolasını bir parola saklama dosyasında saklayabilir. Ayrıca bkz. IBM iüzerinde MQSC komutlarını kullanarak yönetme.

Parola, AES-128 algoritmasıyla şifrelenir. Bu algoritmanın ayrıntıları kamuya açıktır ve güvenli olarak kabul edilir.

Parola, anahtar havuzuna erişebilecek diğer yazılımlar tarafından anlaşılmayan özel bir biçimde bir parola saklama dosyasında saklanır.

Anahtar havuzu parolası şifrelendiğinde benzersiz bir şifreleme anahtarı sağlanabilir. Benzersiz bir şifreleme anahtarı, şifreleme anahtarına erişimi olmayan herkesin parolanın şifresini çözmesini önler. Bu anahtarı **-sf** parametresiyle sağlayın.

Şifrelenmiş parola, anahtar havuzu dosyasıyla aynı dizindeki bir parola saklama dosyasında saklanır.

IBM MQ MQI clients , diğer mekanizmalar aracılığıyla sağlanan parolaları da destekler. Bkz. “IBM i üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması” sayfa 277.

Anahtar havuzu parolasını şifrelemeyi seçtiğiniz yöntemden bağımsız olarak, saklanan parolaları şifreleme sınırlamalarını bildiğinizden emin olun. Bkz. “Parola şifreleme yoluyla koruma sınırları” sayfa 583.

İlgili kavramlar

“IBM i üzerinde bir kuyruk yöneticisi için anahtar havuzu parolası sağlanması” sayfa 276
Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla korunmuştur. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

“IBM i üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması” sayfa 277
Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla korunmuştur. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

“IBM i üzerinde SSL/TLS ile çalışma” sayfa 271
Bu konu derlemi, IBM MQ for IBM iiçinde TLS (Transport Layer Security; İletim Katmanı Güvenliği) ile çalışan tek tek görevlere ilişkin yönergeler sağlar.

IBM i üzerinde sertifika deposu oluşturma
Varsayılan sertifika deposunu kullanmak istemiyorsanız, kendi sertifika deposunu yaratmak için bu yordamı izleyin.

Bu görev hakkında

IBM i varsayılan sertifika deposunu kullanmak istemiyorsanız yeni bir sertifika deposu oluşturun.

IBM i sistem sertifika deposunun kullanılacağını belirtmek için, kuyruk yöneticisinin SSLKEYR özniteliğinin değerini *SYSTEM olarak değiştirin. Bu değer, kuyruk yöneticisinin sistem sertifika deposunu kullandığını ve kuyruk yöneticisinin Digital Certificate Manager (DCM) ile uygulama olarak kullanılmak üzere kaydedildiğini gösterir.

Yordam

1. “DCM ' ye Erişilmesi” sayfa 272 içinde açıklandığı gibi DCM arabirimine erişin
2. Gezinme panosunda **Create New Certificate Store**(Yeni Sertifika Deposu Oluştur) seçeneğini tıklatın.
   Görev çerçevesinde Yeni Sertifika Deposu Yarat sayfası görüntülenir.
3. Görev çerçevesinde **Other System Certificate Store** (Diğer Sistem Sertifika Deposu) seçeneğini belirleyin ve **Continue**(Devam) düğmesini tıklatın.
   Yeni Sertifika Deposunda Sertifika Yarat sayfası görev çerçevesinde görüntülenir.
4. **Hayır-Sertifika deposunda sertifika yaratma** seçeneğini belirleyin ve **Devam**düğmesini tıklatın.
   Görev çerçevesinde Sertifika Deposu Adı ve Parola sayfası görüntülenir.
5. **Sertifika deposu yolu ve dosya adı** alanında bir IFS yolu ve dosya adı yazın; örneğin, `/QIBM/UserData/mqm/qmgrs/qm1/key.kdb`
6. **Parola** alanına bir parola yazın ve **Parolayı Onayla** alanına parolayı yeniden yazın. **Continue**(Devam) seçeneğini tıklatın.
   Havuz anahtarını sakladığınızda gerekli olduğu için parolayı not edin (büyük ve küçük harfe duyarlıdır).
7. DCM ' den çıkmak için tarayıcı pencerenizi kapatın.

Sonraki adım

Sertifika deposunu DCM kullanarak yarattığınızda, parolayı “IBM i sistemlerinde sertifika deposu parolasının saklanması” sayfa 275 içinde açıklandığı gibi sakladığınızdan emin olun.

İlgili görevler

“IBM i üzerinde bir sertifikayı anahtar havuzuna aktarma” sayfa 284
Bir sertifikayı içe aktarmak için bu yordamı izleyin.

IBM i sistemlerinde sertifika deposu parolasının saklanması

Sertifika deposu parolasını CL komutlarını kullanarak saklayın.

Aşağıdaki yönergeler, IBM i ' da bir kuyruk yöneticisi için sertifika deposu parolasının depolanması için geçerlidir. Diğer bir seçenek olarak, IBM MQ MQI clientiçin *SYSTEM sertifika deposunu kullanmıyorsanız (yani, MQSSLKEYR ortamı *SYSTEM dışında bir değere ayarlıysa), “IBM MQ SSL Client Utility (amqrsslc) for IBM i” sayfa 286' in “Sertifika deposu parolasını sakla” sayfa 287 kısmında açıklanan yordamı izleyin.

*SYSTEM sertifika deposunun kullanılacağını belirlediyseniz (kuyruk yöneticisinin SSLKEYR özniteliğinin değerini *SYSTEM olarak değiştirerek), bu adımları izlememelisiniz.

Sertifika deposunu DCM kullanarak yarattığınızda, parolayı saklamak için aşağıdaki komutları kullanın:

```
STRMQM MQMNAME('queue_manager_name')
CHGMQM MQMNAME('queue_manager_name') SSLKEYRPWD('password')
```

Parola, büyük ve küçük harfe duyarlıdır. Tam olarak “IBM i üzerinde sertifika deposu oluşturma” sayfa 274' un 6. adımında girdiğiniz gibi tek tırnak işareti içine alınmalıdır.

Not: Varsayılan sistem sertifika deposunu kullanmıyorsanız ve parolayı saklamıyorsanız, TLS kanallarını başlatma girişimleri sertifika deposuna erişmek için gereken parolayı alamadıkları için başarısız olur.

Parola koruması

V 9.3.0

Bir anahtar havuzu parolası belirtildiğinde, IBM MQ parolayı IBM MQ Parola Koruma sistemini kullanarak şifreler. Parolayı şifrelemek için bir ilk anahtar kullanılır; bu, kuyruk yöneticisine sağlanmazsa, bunun yerine varsayılan bir anahtar kullanılır.

Anahtar havuzu parolasını sağlamadan önce, kuyruk yöneticisi için benzersiz bir başlangıç anahtarı ayarlamanız gerekir. Bunu, **ALTER QMGR** MQSC komutunun **INITKEY** özniteliğini kullanarak yapabilirsiniz:

```
ALTER QMGR INITKEY('value')
```

IBM i üzerinde bir kuyruk yöneticisine ilişkin anahtar havuzunun bulunması

Kuyruk yöneticinizin sertifika deposunun yerini almak için bu yordamı kullanın.

Yordam

1. Aşağıdaki komutu kullanarak kuyruk yöneticinizin özniteliklerini görüntüleyin:

   ```
   DSPMQM MQMNAME('queue manager name')
   ```

2. Sertifika deposunun yolu ve kök adı için komut çıkışını inceleyin.

   Örneğin: `/QIBM/UserData/ICSS/Cert/Server/Default`; burada `/QIBM/UserData/ICSS/Cert/Server` yol, `Default` ise kök addır.

IBM i üzerinde bir kuyruk yöneticisi için anahtar havuzu konumunu değiştirme

CHGMQM ya da ALTER QMGR kullanarak kuyruk yöneticinizin sertifika deposunun konumunu değiştirin.

Yordam

Kuyruk yöneticinizin anahtar havuzu özniteliğini ayarlamak için CHGMQM komutunu ya da ALTER QMGR MQSC komutunu kullanın.

a) CHGMQM kullanılıyor: `CHGMQM MQMNAME('qm1') SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

b) ALTER QMGR kullanılarak: `ALTER QMGR SSLKEYR('/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb')`

Her iki durumda da, sertifika deposu tam olarak nitelenmiş dosya adına sahiptir: `/QIBM/UserData/ICSS/Cert/Server/MyKey.kdb`

Sonraki adım

Bir kuyruk yöneticisinin sertifika deposunun konumunu değiştirdiğinizde, sertifikalar eski konumdan aktarılmaz. Sertifika deposunu yarattığınızda önceden kurulan CA sertifikaları yetersizse, yeni sertifika deposunu "IBM i üzerinde bir sertifikayı anahtar havuzuna aktarma" sayfa 284içinde açıklandığı gibi sertifikalarla doldurmanız gerekir. "IBM i sistemlerinde sertifika deposu parolasının saklanması" sayfa 275içinde açıklandığı gibi yeni konuma ilişkin parolayı da saklamanız gerekir.

V 9.3.0 IBM i V 9.3.0 *IBM i üzerinde bir kuyruk yöneticisi için anahtar havuzu parolası sağlanması*

Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla korunmuştur. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

IBM MQ , bir kuyruk yöneticisine anahtar havuzu parolası sağlamak için bir mekanizma sağlar:

- **CHGMQM** komutundaki **SSLKEYRPWD** parametresi

Anahtar havuzu parolası, IBM MQ parola koruma sistemi kullanılarak şifrelenir. Anahtar havuzu parolasını koruma yöntemleri hakkında daha fazla bilgi için bkz. "IBM i üzerinde anahtar havuzu parolalarını şifreleme" sayfa 273.

Ayrıca bkz. IBM iüzerinde MQSC komutlarını kullanarak yönetme.

SSLKEYRPWD özniteliği

Bir anahtar havuzu parolasını doğrudan kuyruk yöneticisine sağlamak için, *queue_manager* yerine kuyruk yöneticisi adınızı ve *password* değerini anahtar havuzu parolanızla değiştirerek aşağıdaki **CHGMQM** komutunu çalıştırın.

```
CHGMQM MQMNAME('queue_manager') SSLKEYRPWD('password')
```

Uyarı: Kuyruk yöneticisi adını ve parolasını tek tırnak işareti içine aldığınızdan emin olun, tersi durumda IBM MQ karakterleri büyük harfe dönüştürür.

Bu yöntem kullanılarak bir anahtar havuzu parolası belirtildiğinde, parola saklanmadan önce IBM MQ parola koruma sistemi kullanılarak şifrelenir.

Parolayı şifrelemek için ilk anahtar olarak bilinen bir şifreleme anahtarı kullanılır. Kuyruk yöneticisini, parolayı güvenli bir şekilde korumak için benzersiz bir başlangıç anahtarı kullanacak şekilde ayarlayın. Başlangıç anahtarı belirtmezseniz, varsayılan anahtar kullanılır.

Anahtar havuzu parolasını ayarlamadan önce kuyruk yöneticisinin benzersiz bir başlangıç anahtarıyla yapılandırıldığından emin olun. **ALTER QMGR** komutundaki **INITKEY** özniteliğini kullanarak ilk anahtarı değiştirebilirsiniz. Örneğin:

```
ALTER QMGR INITKEY('mykey')
```

Uyarı: Anahtar havuzu parolasını ayarladıktan sonra ilk anahtarı değiştirirseniz, anahtar havuzu parolası yeni başlangıç anahtarıyla şifrelenmez. İlk anahtarı değiştirirseniz, anahtar havuzu

parolasını da sıfırlamanız gerekir. Tersi durumda, IBM MQ anahtar havuzu parolasının şifresini çözemez ve bu nedenle anahtar havuzuna erişemez.

SSLKEYRPWD özniteliğiyle ilgili daha fazla bilgi için bkz. **CHGMQM** komutundaki **SSLKEYRPWD** parametresi.

İlgili kavramlar

"IBM i üzerinde anahtar havuzu parolalarını şifreleme" sayfa 273
Bazı IBM MQ bileşenlerinin sayısal sertifikaları ya da simetrik anahtarları içeren bir anahtar havuzuna erişmesi gerekir. Bir anahtar havuzu, hassas bilgiler içerdiği için bir parolayla korunmuştur. Anahtar havuzu parolası, anahtar havuzuna erişildiğinde IBM MQ ' un okuyabileceği bir konumda saklanmalıdır. Anahtar havuzuna yetkisiz erişim olasılığını azaltmak için parola da şifrelenmelidir.

"IBM i üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması" sayfa 277
Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla korunmuştur. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

V 9.3.0 IBM i V 9.3.0 *IBM i üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması*

Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla korunmuştur. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

IBM MQ , anahtar havuzu parolasını bir IBM MQ MQI client' e sağlamak için dört mekanizma sağlar:

- "MQSCO ' nun KeyRepoPassword alanları " sayfa 277
- "MQKEYRPWD ortam değişkeni" sayfa 278
- "İstemci yapılandırma dosyasının SSLKeyRepositoryPassword özniteliği" sayfa 278
- "Anahtar havuzu saklama dosyası" sayfa 278

Bir anahtar havuzu parola saklama dosyası kullanmıyorsanız, anahtar havuzu parolasını düz metin dizgisi olarak ya da IBM MQ parola koruma sistemi kullanılarak şifrelenen bir dizgi olarak sağlayabilirsiniz. Anahtar havuzu parolasını koruma yöntemleri hakkında daha fazla bilgi için bkz. "IBM i üzerinde anahtar havuzu parolalarını şifreleme" sayfa 273.

MQSCO ' nun KeyRepoPassword alanları

MQSCO yapısını kullanarak bir anahtar havuzu parolası sağlamak için aşağıdaki üç değişken dizgi alanının bir birleşimini kullanmanız gerekir:

KeyRepoPasswordLength
Parolanın uzunluğu.

KeyRepoPasswordPtr
Parolayı içeren bellekteki konuma ilişkin gösterge.

KeyRepoPasswordOffset
Bellekteki parolanın yeri; MQSCO yapısının başlangıcından itibaren byte sayısı olarak gösterilir.

Not: KeyRepoPasswordPtr ya da **KeyRepoPasswordOffset**ürünlerinden yalnızca birini sağlayabilirsiniz.

Örneğin:

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

Uyarı: Bu yöntemi kullanarak parolayı sağlarsanız, parolayı IBM MQ client uygulamasına sağlanmadan önce şifreleyin. Daha fazla bilgi için bkz "Anahtar havuzu parolasını şifreleme" sayfa 278.

MQCSO yapısıyla ilgili daha fazla bilgi için bkz. MQSCO-SSL/TLS yapılandırma seçenekleri.

MQKEYRPWD ortam değişkeni

MQSCO yapısı kullanılarak istemciye anahtar havuzu parolası sağlanmazsa, *MQKEYRPWD* ortam değişkenini kullanarak anahtar havuzu parolasını belirtebilirsiniz. Örneğin:

```
export MQKEYRPWD=passw0rd
```

veya

```
set MQKEYRPWD=passw0rd
```

Burada *passw0rd* , parolanızdır.

Uyarı: Parolayı bu yöntemi kullanarak girdiyseniz, ortam değişkeninin değerini ayarlamadan önce parolayı şifreleyin. Daha fazla bilgi için bkz "Anahtar havuzu parolasını şifreleme" sayfa 278.

İstemci yapılandırma dosyasının SSLKeyRepositoryPassword özniteliği

İstemciye diğer yöntemlerden birini kullanarak bir anahtar havuzu parolası sağlanmazsa, istemci yapılandırma dosyasının **SSL** kısmına ilişkin **SSLKeyRepositoryPassword** özniteliğini kullanarak anahtar havuzu parolasını belirtebilirsiniz. Örneğin:

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

Uyarı: Parolayı bu yöntemi kullanarak girdiyseniz, **SSLKeyRepositoryPassword** özniteliğinin değerini ayarlamadan önce parolayı şifreleyin. Daha fazla bilgi için bkz "Anahtar havuzu parolasını şifreleme" sayfa 278.

İstemci yapılandırma dosyasının SSL kısmı hakkında daha fazla bilgi için bkz. İstemci yapılandırma dosyasının SSL kısmı.

Anahtar havuzu saklama dosyası

Anahtar havuzu parolası istemciye diğer yöntemlerden biri kullanılarak sağlanmazsa, IBM MQ anahtar havuzuyla aynı dizinde bir parola saklama dosyası olduğunu varsayar. Saklama dosyası, anahtar havuzuyla aynı kök ada sahip, ancak `.sth` uzantısına sahip.

amqrsslc komut satırı aracı kullanılarak bir anahtar havuzu saklama dosyası yaratılır. Saklama dosyasını yaratmak için aşağıdaki komutu çalıştırın:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

Bu komut, şifrelenecek parolayı girmenizi ister. Parola, **-sf** parametresi kullanılarak sağlanmamışsa, IBM MQ parola koruma sistemi tarafından varsayılan bir şifreleme anahtarıyla şifrelenir.

Daha fazla bilgi için bkz. "IBM MQ SSL Client Utility (amqrsslc) for IBM i" sayfa 286 ve "Anahtar havuzu parolasını şifreleme" sayfa 278.

Anahtar havuzu parolasını şifreleme

Anahtar havuzu parolasını parola saklama dosyası dışında bir yöntem kullanarak sağlıyorsanız, IBM MQ parola koruma sistemini kullanarak parolayı şifreleyin. Parolayı şifrelemek için **runmqicred** komutunu çalıştırın. İstendiğinde anahtar havuzu parolasını girin. Komut şifrelenmiş parolayı çıkarır. Şifrelenmiş parola, açıklanan yöntemlerden herhangi biri kullanılarak düz metin parolası yerine IBM MQ MQI client ' e sağlanabilir.

Parolayı şifrelemek için ilk anahtar olarak bilinen bir şifreleme anahtarı kullanılır. Parolayı şifrelediğinizde, parolayı güvenli bir şekilde korumak için benzersiz bir başlangıç anahtarı kullanın. Kendi ilk anahtarınızı sağlamak için **runmqicred** komutuna ilişkin **-sf** parametresini kullanın. Başlangıç anahtarı belirtmezseniz, varsayılan anahtar kullanılır.

Daha fazla bilgi için bkz. runmqicred (IBM MQ istemci parolalarını koruma).

Anahtar havuzu parolası şifrelendiğinde kendi ilk anahtarınızı sağlarsanız ve IBM MQ MQI clientiçin şifrelenmiş parolayı sağlarsanız, IBM MQ MQI clientiçin aynı ilk anahtarı sağladığınızdan da emin olmanız gerekir. IBM MQ MQI clientiçin ilk anahtarın nasıl sağlanacağıyla ilgili daha fazla bilgi için bkz. "IBM i üzerinde bir IBM MQ MQI client için başlangıç anahtarı sağlanması" sayfa 279.

İlgili kavramlar

"IBM i üzerinde anahtar havuzu parolalarını şifreleme" sayfa 273
Bazı IBM MQ bileşenlerinin sayısal sertifikaları ya da simetrik anahtarları içeren bir anahtar havuzuna erişmesi gerekir. Bir anahtar havuzu, hassas bilgiler içerdiği için bir parolayla korunmuştur. Anahtar havuzu parolası, anahtar havuzuna erişildiğinde IBM MQ ' un okuyabileceği bir konumda saklanmalıdır. Anahtar havuzuna yetkisiz erişim olasılığını azaltmak için parola da şifrelenmelidir.

"IBM i üzerinde bir kuyruk yöneticisi için anahtar havuzu parolası sağlanması" sayfa 276
Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla korunmuştur. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

V 9.3.0 IBM i V 9.3.0 *IBM i üzerinde bir IBM MQ MQI client için başlangıç anahtarı sağlanması*

IBM MQ Parola Koruma Sistemi kullanılarak şifrelenmiş bir IBM MQ MQI client ' e değişkenler sağlarsanız, değeri şifrelemek için kullanılan ilgili başlangıç anahtarını sağlamanız gerekebilir.

Değeri şifrelerken bir başlangıç anahtarı belirtmediyseniz, IBM MQ clientiçin herhangi bir başlangıç anahtarı değeri sağlamanız gerekmez. Ancak, benzersiz bir başlangıç anahtarı kullandıysanız, IBM MQ client için ilk anahtarı aşağıdaki yöntemleri kullanarak sağlayabilirsiniz:

- "MQCSP yapısını kullanarak ilk anahtarın sağlanmasını" sayfa 279
- "MQS_MQI_KEYFILE ortam değişkenini kullanarak ilk anahtar belirtiliyordu" sayfa 279
- "İstemci yapılandırma dosyasını kullanarak ilk anahtarın sağlanmasına" sayfa 280

MQCSP yapısını kullanarak ilk anahtarın sağlanmasını

MQCSP yapısını kullanarak ilk anahtarı sağlamak için aşağıdaki üç değişken dizgi alanının bir birleşimini kullanmanız gerekir:

InitialKeyLength
İlk anahtarın uzunluğu

InitialKeyPtr
İlk anahtarı içeren bellekteki konuma ilişkin gösterge

InitialKeyOffset
MQCSP yapısının başlangıcından itibaren byte sayısı olarak gösterilen, bellekteki ilk anahtarın yeri.

Not: **InitialKeyPtr** ya da **InitialKeyOffset**ürünlerinden yalnızca birini sağlayabilirsiniz.

Örneğin:

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

MQS_MQI_KEYFILE ortam değişkenini kullanarak ilk anahtar belirtiliyordu

MQCSP yapısı kullanılarak istemciye ilk anahtar sağlanmazsa, IBM MQ *MQS_MQI_KEYFILE* ortam değişkenini denetler. Bu ortam değişkenini, kullanmak istediğiniz ilk anahtardan oluşan tek satırlık metni içeren bir dosyanın konumuna ayarlamanız gerekir.

Örneğin, kök dizinde `mykey.key` adlı bir dosya varsa ve ilk anahtarı içeriyorsa, ortam değişkenini aşağıdaki gibi ayarlamanız gerekir:

```
export MQS_MQI_KEYFILE=/mykey.key
```

veya

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

İstemci yapılandırma dosyasını kullanarak ilk anahtarın sağlanmasına

İstemciye önceki bir düzenek kullanılarak bir başlangıç anahtarı sağlanmazsa, IBM MQ `mqclient.ini` dosyasının Güvenlik kısmına ilişkin **`MQIInitialKeyFile`** özniteliğini denetler. Bu özniteliği, kullanmak istediğiniz ilk anahtardan oluşan tek bir metin satırı içeren bir dosyanın konumuna ayarlamalısınız.

Örneğin, kök dizinde `mykey.key` adlı bir dosya varsa ve ilk anahtarı içeriyorsa, istemci yapılandırma dosyası aşağıdakileri içermelidir:

```
Security:
    MQIInitialKeyFile=/mykey.key
```

İlgili kavramlar

“IBM i üzerinde anahtar havuzu parolalarını şifreleme” sayfa 273
Bazı IBM MQ bileşenlerinin sayısal sertifikaları ya da simetrik anahtarları içeren bir anahtar havuzuna erişmesi gerekir. Bir anahtar havuzu, hassas bilgiler içerdiği için bir parolayla korunmuştur. Anahtar havuzu parolası, anahtar havuzuna erişildiğinde IBM MQ ' un okuyabileceği bir konumda saklanmalıdır. Anahtar havuzuna yetkisiz erişim olasılığını azaltmak için parola da şifrelenmelidir.

“IBM i üzerinde SSL/TLS ile çalışma” sayfa 271
Bu konu derlemi, IBM MQ for IBM iiçinde TLS (Transport Layer Security; İletim Katmanı Güvenliği) ile çalışan tek tek görevlere ilişkin yönergeler sağlar.

IBM i üzerinde test için sertifika yetkilisi ve sertifika yaratılması

Sertifika isteklerini imzalamak ve CA sertifikasını yaratmak ve kurmak üzere yerel bir CA sertifikası yaratmak için bu yordamı kullanın.

Başlamadan önce

Bu konudaki yönergeler, bir yerel sertifika yetkilisinin (CA) var olmadığını varsayar. Yerel bir CA varsa, bkz. “IBM i üzerinde sunucu sertifikası isteme” sayfa 281.

Bu görev hakkında

TLS ' yi kurduğunuzda sağlanan CA sertifikaları, düzenleyen CA tarafından imzalanır. IBM iişletim sisteminde, sisteminizdeki TLS iletişimini test etmek için sunucu sertifikalarını imzalayabilen bir yerel sertifika yetkilisi oluşturabilirsiniz. Yerel bir CA sertifikası yaratmak için Web tarayıcısında aşağıdaki adımları izleyin:

Yordam

1. “DCM ' ye Erişilmesi” sayfa 272içinde açıklandığı gibi DCM arabirimine erişin.
2. Gezinme panosunda **Create a Certificate Authority**(Sertifika Yetkilisi Oluştur) seçeneğini tıklatın. Görev çerçevesinde Sertifika Yetkilisi Yarat sayfası görüntülenir.
3. **Sertifika deposu parolası** alanına bir parola yazın ve **Parolayı onayla** alanına parolayı yeniden yazın.
4. **Sertifika Yetkilisi (CA) adı** alanında bir ad yazın; örneğin, `TLS Test Certificate Authority`.
5. **Ortak Ad** ve **Kuruluş** alanlarına uygun değerleri yazın ve bir ülke seçin. Geri kalan isteğe bağlı alanlar için, gerek duyduğunuz değerleri yazın.
6. **Geçerlilik süresi** alanında, yerel sertifika kuruluşu (CA) için bir geçerlilik süresi yazın.

Varsayılan değer 1095 gündür.

7. **Continue**(Devam) seçeneğini tıklatın.

   CA yaratılır ve DCM, yerel sertifika kuruluşunuz için bir sertifika deposu ve sertifika kuruluşu (CA) sertifikası yaratır.
8. **Install certificate**(Sertifikayı kur) seçeneğini tıklatın.

   Karşıdan yükleme yöneticisi iletişim kutusu görüntülenir.
9. CA sertifikasını saklamak istediğiniz geçici dosyanın tam yol adını yazın ve **Sakla**düğmesini tıklatın.
10. Karşıdan yükleme tamamlandığında **Aç**' ı tıklatın.

    Sertifika penceresi görüntülenir.
11. **Install certificate**(Sertifikayı kur) seçeneğini tıklatın.

    Sertifika İçe Aktarma sihirbazı görüntülenir.
12. **İleri**'yi tıklatın.
13. **Sertifika deposunu sertifika tipine dayalı olarak otomatik olarak seç** seçeneğini belirleyin ve **İleri**düğmesini tıklatın.
14. **Bitir**'i tıklatın.

    Bir doğrulama penceresi görüntülenir.
15. **Tamam**'ı tıklatın.
16. Sertifika penceresinde **Tamam**düğmesini tıklatın.
17. **Continue**(Devam) seçeneğini tıklatın.

    Görev çerçevesinde Sertifika Yetkilisi İlkesi sayfası görüntülenir.
18. **Kullanıcı sertifikalarının oluşturulmasına izin ver** alanında **Evet**seçeneğini belirleyin.
19. **Geçerlilik süresi** alanında, yerel sertifika yetkiliniz tarafından verilen sertifikaların geçerlilik süresini yazın.

    Varsayılan değer 365 gündür.
20. **Continue**(Devam) seçeneğini tıklatın.

    Yeni Sertifika Deposunda Sertifika Yarat sayfası görev çerçevesinde görüntülenir.
21. Uygulamaların hiçbirinin seçilmediğini denetleyin.
22. Yerel CA ' nın kurulumunu tamamlamak için **Continue** (Devam) seçeneğini tıklatın.

Sonraki adım

Var olan bir sertifikayı yenilemeniz gerekirse, IBM i belgelerinde Var olan bir sertifikanın yenilenmesi başlıklı konuya bakın.

IBM i üzerinde sunucu sertifikası isteme

Dijital sertifikalar, bir genel anahtarın belirli bir varlığa ait olduğunu onaylayan, kimliğe bürünmeye karşı koruma sunar. Sayısal Certificate Manager (DCM) kullanılarak bir sertifika yetkilisinden yeni bir sunucu sertifikası istenebilir.

Bu görev hakkında

Bir Web tarayıcısında aşağıdaki adımları izleyin:

Yordam

1. “DCM ' ye Erişilmesi” sayfa 272içinde açıklandığı gibi DCM arabirimine erişin.
2. Gezinme panosunda **Sertifika Deposu Seç**seçeneğini tıklatın.

   Görev çerçevesinde Sertifika Deposu Seç sayfası görüntülenir.
3. Kullanmak istediğiniz sertifika deposunu seçin ve **Devam**düğmesini tıklatın.
4. İsteğe bağlı: 3. adımda ***SYSTEM** değerini seçtiyseniz, sistem deposu parolasını girin ve **Continue**(Devam) düğmesini tıklatın.

5. İsteğe bağlı: 3. adımda **Diğer Sistem Sertifika Deposu** öğesini seçtiyseniz, **Sertifika deposu yolu ve dosya adı** alanında, sertifika deponizi yaratırken ayarladığınız IFS yolunu ve dosya adını yazın. **Sertifika Deposu Parolası** alanında bir parola da yazın. Daha sonra **Continue** (Devam) seçeneğini tıklatın.
6. Gezinme panosunda **Create Certificate**(Sertifika Oluştur) seçeneğini tıklatın.
7. Görev çerçevesinde, **Sunucu ya da istemci sertifikası** radyo düğmesini seçin ve **Devam**düğmesini tıklatın.

   Görev çerçevesinde Sertifika Yetkilisi Seç (CA) sayfası görüntülenir.
8. İş istasyonunuzda yerel bir CA varsa, sertifikayı imzalamak için yerel CA 'yı ya da ticari bir CA' yı seçin. İstediğiniz sertifika kuruluşu (CA) için radyo düğmesini seçin ve **Continue**(Devam) düğmesini tıklatın.

   Görev çerçevesinde Sertifika Yarat sayfası görüntülenir.
9. İsteğe bağlı: Bir kuyruk yöneticisi için, **Sertifika etiketi** alanına sertifika etiketini girin.

   Etiket, ayarlanmışsa **CERTLABL** özniteliğinin değeri ya da sonuna kuyruk yöneticisinin adı eklenmiş olan varsayılan `ibmwebspheremq` değeridir (tümü küçük harfli olarak). Ayrıntılar için bkz. Dijital sertifika etiketleri .

   Örneğin, QM1kuyruk yöneticisi için varsayılan değeri kullanmak üzere `ibmwebspheremqqm1` yazın.
10. İsteğe bağlı: Bir IBM MQ MQI clientiçin, **Sertifika etiketi** alanına `ibmwebspheremq` yazın ve ardından oturum açma kullanıcı kimliğiniz küçük harfe katlandı.
    Örneğin, şunları yazın `ibmwebspheremqmyuserID`
11. **Ortak Ad** ve **Kuruluş** alanlarına uygun değerleri yazın ve bir ülke seçin. Geri kalan isteğe bağlı alanlar için, gerek duyduğunuz değerleri yazın.

Sonuçlar

Sertifikanızı imzalamak için ticari bir sertifika kuruluşu seçtiyseniz, DCM PEM (Gizlilik-Gelişmiş Posta) biçiminde bir sertifika isteği oluşturur. İsteği seçtiğiniz sertifika kuruluşuna iletiin.

Sertifikanızı imzalamak için yerel CA ' yı seçtiyseniz, DCM sertifikanın sertifika deposunda oluşturulduğunu ve kullanılabileceğini bildirir.

IBM i üzerinde IBM Key Manager için sunucu sertifikası isteme

Yerel sertifika kuruluşunuz (CA) tarafından imzalanmış bir sertifika yaratmak ya da IBM Key Management (iKeyman) yardımcı programına içe aktarmak üzere ticari bir CA tarafından imzalanmış bir sunucu sertifikası için başvurmak üzere bu yordamı izleyin.

Bu görev hakkında

Digital Certificate Manager (DCM), birden çok altyapıda IBM MQ için sertifika yöneticisi olarak görev yaptığında bir kullanıcı sertifikası kullanılmalıdır. Diğer platformlara dağıtılan kişisel sertifikalar ve iKeyman yardımcı programına içe aktarmak için, bir Web tarayıcısında aşağıdaki adımları gerçekleştirin:

Yordam

1. “DCM ' ye Erişilmesi” sayfa 272içinde açıklandığı gibi DCM arabirimine erişin.
2. **Gezinme** bölmesinde **Sertifika Oluştur**' u tıklatın.

   Görev çerçevesinde **Sertifika Yarat** sayfası görüntülenir.
3. **Create Certificate** (Sertifika Oluştur) panosunda **User certificate** (Kullanıcı sertifikası) radyo düğmesini seçin ve **Continue**(Devam) düğmesini tıklatın.

   Kullanıcı Sertifikası Yarat sayfası görüntülenir.
4. **Create User Certificate** (Kullanıcı Sertifikası Oluştur) panosunda, **Kuruluş adı**, **İl** ya da **bölge**, **Ülke** ya da **bölge**için Sertifika Bilgileri altındaki zorunlu alanları doldurun. İsteğe bağlı olarak, değerleri **Kuruluş birimi** ve **İlçe** ya da **İlçe** alanlarına koyun. **Continue**(Devam) seçeneğini tıklatın.

   Ortak ad , otomatik olarak iSeries sisteminde oturum açtığınız kullanıcı kimliğine ayarlanır.
5. Sonraki **Create User Certificate** (Kullanıcı Sertifikası Oluştur) panosunda **Install certificate** (Sertifikayı kur) seçeneğini tıklatın ve **Continue**(Devam) seçeneğini tıklatın.

Şunu belirten bir ileti görüntülenir: `Kişisel sertifikanız kuruldu. Bu sertifikanın yedek kopyasını saklamanız gerekir.`

6. **Tamam**'ı tıklatın.
7. DCM ' ye erişmek için kullandığınız İnternet tarayıcısına bağlı olarak aşağıdaki adımları gerçekleştirin:
   a) Microsoft Edge için: **Araçlar > Internet Seçenekleri > İçerik sekmesi > Sertifikalar düğmesi > Kişisel sekmesi >**seçeneklerini belirleyin. Sertifikayı seçin ve **Dışa Aktar**' ı tıklatın.
   b) Mozilla Firefox için: **Araçlar > Seçenekler > Avantajlar > Şifreleme sekmesi > Sertifikaları Görüntüle düğmesi > Sertifikalarınız sekmesi >**seçeneklerini belirleyin. Sertifikayı seçin ve **Yedekle**düğmesini tıklatın. Yolu ve dosya adını seçin ve **Tamam**' ı tıklatın.
8. Dışa aktarılan sertifikayı, ikili biçimde FTP kullanarak uzak sisteme aktarın.
9. 7. adımdaki dışa aktarılan sertifikayı anahtar veritabanındaki iKeyman yardımcı programına ekleyin.
   a) Sertifika Microsoft Edge kullanılarak kaydedildiyse, Microsoft `.pfx` dosyasından içe aktarma dosyasında açıklanan yönergeleri kullanın.
   b) Sertifika Mozilla Firefoxkullanılarak kaydedildiyse, Kişisel sertifikanın anahtar havuzuna aktarılmasıbaşlıklı konuda açıklanan yönergeleri kullanın.

   İçe aktarma sırasında, kişisel sertifikanın ve imzalayıcı sertifikasının etiket adının IBM MQ ' in beklediği gibi değiştirildiğinden emin olun. Etiket, ayarlanmışsa IBM MQ **CERTLABL** özniteliğinin değeri ya da sonuna kuyruk yöneticisinin adı eklenmiş olarak varsayılan `ibmwebspheremq` değeri olmalıdır. Ayrıntılar için bkz. Dijital sertifika etiketleri .

IBM i üzerinde bir anahtar havuzuna sunucu sertifikaları eklenmesi

Anahtar havuzuna istenen bir sertifika eklemek için bu yordamı izleyin.

Bu görev hakkında

CA size yeni bir sunucu sertifikası gönderdikten sonra, bu sertifikayı isteği oluşturacağınız sertifika deposuna eklersiniz. Sertifika yetkilisi sertifikayı e-posta iletisinin bir parçası olarak gönderirse, sertifikayı ayrı bir dosyaya kopyalayın.

Not:

- Sunucu sertifikası yerel sertifika kuruluşunuz tarafından imzalandıysa, bu yordamı gerçekleştirmeniz gerekmez.
- PKCS #12 biçiminde bir sunucu sertifikasını DCM ' ye aktarmadan önce, ilgili CA sertifikasını içe aktarmanız gerekir.

Kuyruk yöneticisi sertifika deposuna bir sunucu sertifikası almak için aşağıdaki yordamı kullanın:

Yordam

1. “DCM ' ye Erişilmesi” sayfa 272içinde açıklandığı gibi DCM arabirimine erişin.
2. Gezinme panosunda **Sertifikaları Yönet** görev kategorisinde **Sertifikayı İçe Aktar**' ı tıklatın.

   Görev çerçevesinde Sertifikayı İçe Aktar sayfası görüntülenir.
3. Sertifika tipinize ilişkin radyo düğmesini seçin ve **Continue**(Devam) düğmesini tıklatın.

   Görev çerçevesinde Import Server ya da Client Certificate ya da Import Certificate Authority (CA) Certificate sayfası görüntülenir.
4. **Dosyayı İçe Aktar** alanında, içe aktarmak istediğiniz sertifikanın dosya adını yazın ve **Devam**düğmesini tıklatın.

   DCM, dosyanın biçimini otomatik olarak belirler.
5. Sertifika bir **Sunucu ya da istemci** sertifikaysa, görev çerçevesinde parolayı yazın ve **Devam**düğmesini tıklatın.

   DCM, sertifikanın içe aktarıldığını bildirir.

IBM i üzerindeki bir anahtar havuzundan bir sertifikanın dışa aktarılması

Bir sertifikanın dışa aktarılması hem genel hem de özel anahtarı dışa aktarır. Özel bir anahtarın geçirilmesi güvenliğinizi tamamen tehlikeye atacağı için, bu işlem son derece dikkatli yapılmalıdır.

Başlamadan önce

Bir kullanıcının sertifikasını başka bir kullanıcıyla paylaştığınızda, ortak anahtarları değiştirmiş olur. Bu işlem **Görev 5 'te açıklanmıştır. Sertifikaları Paylaşma** , “AIX and Linux üzerinde AMS için Hızlı Başlangıç Kılavuzu” sayfa 623' in Sertifikaları Paylaşma bölümünde bulunur. Bir sertifikayı burada açıklandığı gibi dışa aktardığınızda, hem genel hem de özel anahtarı dışa aktarır. Özel bir anahtarın geçirilmesi güvenliğinizi tamamen tehlikeye atacağı için, bu işlem son derece dikkatli yapılmalıdır.

Bu görev hakkında

Sertifikayı dışa aktarmak istediğiniz bilgisayarda aşağıdaki adımları gerçekleştirin:

Yordam

1. “DCM ' ye Erişilmesi” sayfa 272içinde açıklandığı gibi DCM arabirimine erişin.
2. Gezinme panosunda **Sertifika Deposu Seç**seçeneğini tıklatın.

   Görev çerçevesinde Sertifika Deposu Seç sayfası görüntülenir.
3. Kullanmak istediğiniz sertifika deposunu seçin ve **Devam**düğmesini tıklatın.
4. İsteğe bağlı: 3. adımda ***SYSTEM** değerini seçtiyseniz, sistem deposu parolasını girin ve **Continue**(Devam) düğmesini tıklatın.
5. İsteğe bağlı: 3. adımda **Diğer Sistem Sertifika Deposu** öğesini seçtiyseniz, **Sertifika deposu yolu ve dosya adı** alanında, sertifika deposunu yaratırken ayarladığınız IFS yolunu ve dosya adını yazın ve **Sertifika Deposu Parolası** alanına bir parola yazın. Daha sonra **Continue** (Devam) seçeneğini tıklatın.
6. Gezinme panosunda **Sertifikaları Yönet** görev kategorisinde **Sertifikayı Dışa Aktar**' ı tıklatın.

   Görev çerçevesinde bir Sertifikayı Dışa Aktar sayfası görüntülenir.
7. Sertifika tipinize ilişkin radyo düğmesini seçin ve **Continue**(Devam) düğmesini tıklatın.

   Görev çerçevesinde, Dışa Aktarma Sunucusu ya da İstemci Sertifikası sayfası ya da Sertifika Yetkilisini Dışa Aktar (CA) Sertifikası sayfası görüntülenir.
8. Dışa aktarmak istediğiniz sertifikayı seçin.
9. Sertifikayı bir dosyaya mı, yoksa doğrudan başka bir sertifika deposuna mı aktarmak istediğinizi belirtmek için radyo düğmesini seçin.
10. Bir sunucu ya da istemci sertifikasını bir dosyaya aktarmayı seçtiyseniz, aşağıdaki bilgileri belirtin:
    - Dışa aktarılan sertifikayı saklamak istediğiniz yerin yolu ve dosya adı.
    - Kişisel sertifika için, dışa aktarılan sertifikayı ve hedef yayını şifrelemek için kullanılan parola. CA sertifikaları için parolayı belirtmenize gerek yoktur.
11. Bir sertifikayı doğrudan başka bir sertifika deposuna aktarmayı seçtiyseniz, hedef sertifika deposunu ve parolasını belirtin.
12. **Continue**(Devam) seçeneğini tıklatın.

IBM i üzerinde bir sertifikayı anahtar havuzuna aktarma

Bir sertifikayı içe aktarmak için bu yordamı izleyin.

Başlamadan önce

PKCS #12 biçiminde bir kişisel sertifikayı DCM ' ye aktarmadan önce, ilgili CA sertifikasını içe aktarmanız gerekir.

Bu görev hakkında

Sertifikayı içe aktarmak istediğiniz makinede bu adımları gerçekleştirin.

Yordam

1. “DCM ' ye Erişilmesi” sayfa 272içinde açıklandığı gibi DCM arabirimine erişin.
2. Gezinme panosunda **Sertifika Deposu Seç**seçeneğini tıklatın.
   Görev çerçevesinde Sertifika Deposu Seç sayfası görüntülenir.
3. Kullanmak istediğiniz sertifika deposunu seçin ve **Devam**düğmesini tıklatın.
4. İsteğe bağlı: 3. adımda ***SYSTEM** değerini seçtiyseniz, sistem deposu parolasını girin ve **Continue**(Devam) düğmesini tıklatın.
5. İsteğe bağlı: 3. adımda **Diğer Sistem Sertifika Deposu** öğesini seçtiyseniz, **Sertifika deposu yolu ve dosya adı** alanında, sertifika deposunu yaratırken ayarladığınız IFS yolunu ve dosya adını yazın ve **Sertifika Deposu Parolası** alanına bir parola yazın. Daha sonra **Continue** (Devam) seçeneğini tıklatın.
6. Gezinme panosunda **Sertifikaları Yönet** görev kategorisinde **Sertifikayı İçe Aktar**' ı tıklatın.
   Görev çerçevesinde Sertifikayı İçe Aktar sayfası görüntülenir.
7. Sertifika tipinize ilişkin radyo düğmesini seçin ve **Continue**(Devam) düğmesini tıklatın.
   Görev çerçevesinde Import Server ya da Client Certificate sayfası ya da Import Certificate Authority (CA) Certificate sayfası görüntülenir.
8. **Dosyayı İçe Aktar** alanında, içe aktarmak istediğiniz sertifikanın dosya adını yazın ve **Devam**düğmesini tıklatın.
   DCM, dosyanın biçimini otomatik olarak belirler.
9. Sertifika bir **Sunucu ya da istemci** sertifikaysa, görev çerçevesinde parolayı yazın ve **Devam**düğmesini tıklatın. DCM, sertifikanın içe aktarıldığını bildirir.

IBM i içindeki sertifikaları kaldırma

Kişisel sertifikaları kaldırmak için bu yordamı kullanın.

Yordam

1. “DCM ' ye Erişilmesi” sayfa 272içinde açıklandığı gibi DCM arabirimine erişin.
2. Gezinme panosunda **Sertifika Deposu Seç**seçeneğini tıklatın.
   Görev çerçevesinde Sertifika Deposu Seç sayfası görüntülenir.
3. **Diğer Sistem Sertifikası Deposu** onay kutusunu seçin ve **Devam**düğmesini tıklatın.
   Sertifika Deposu ve Parola sayfası görüntülenir.
4. **Sertifika deposu yolu ve dosya adı** alanında, sertifika deposunu yaratırken ayarladığınız IFS yolunu ve dosya adını yazın.
5. **Sertifika Deposu Parolası** alanında bir parola yazın. **Continue**(Devam) seçeneğini tıklatın.
   Yürürlükteki Sertifika Deposu sayfası görev çerçevesinde görüntülenir.
6. Gezinme panosunda **Sertifikaları Yönet** görev kategorisinde **Sertifikayı Sil**düğmesini tıklatın.
   Görev çerçevesinde Sertifikayı Silmeyi Onayla sayfası görüntülenir.
7. Silmek istediğiniz sertifikayı seçin. **Sil** düğmesini tıklatın.
8. Sertifikayı silmek istediğinizi onaylamak için **Evet** düğmesini tıklatın. Tersi durumda, **Hayır**' ı tıklatın.
   DCM, sertifikayı silip silmediğini size bildirir.

*IBM i üzerinde tek yönlü kimlik doğrulaması için *SYSTEM sertifika deposunun kullanılması*

Tek yönlü kimlik doğrulamasını ayarlamak için bu yönergeleri izleyin.

Başlamadan önce

- Bir kuyruk yöneticisi, kanallar ve iletim kuyrukları yaratır.
- Sunucu kuyruk yöneticisinde bir sunucu ya da istemci sertifikası yaratın.
- CA sertifikasını istemci kuyruk yöneticisine aktarın ve anahtar havuzuna aktarın.

- Sunucuda ve istemci kuyruk yöneticilerinde bir dinleyici başlatın.

Bu görev hakkında

TLS sunucusu olarak IBM i çalıştıran bir bilgisayarı kullanarak tek yönlü kimlik doğrulamasını kullanmak için SSL Anahtar Havuzu (SSLKEYR) parametresini *SYSTEM olarak ayarlayın. Bu ayar, IBM MQ kuyruk yöneticisini uygulama olarak kaydeder. Daha sonra tek yönlü kimlik doğrulamasını etkinleştirmek için kuyruk yöneticisine bir sertifika atayabilirsiniz.

Anahtar havuzunda istemci kuyruk yöneticisi için kukla sertifika yaratarak tek yönlü kimlik doğrulaması gerçekleştirmek için özel anahtar depolarını da kullanabilirsiniz.

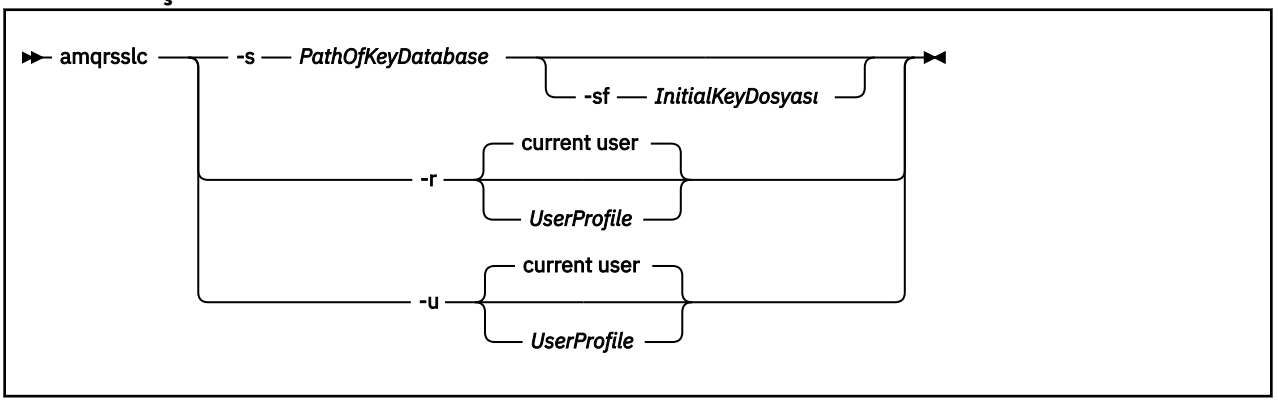
Yordam

1. Sunucu ve istemci kuyruk yöneticisinde aşağıdaki adımları izleyin:
   a) `CHGMQM MQMNAME(SSL) SSLKEYR(*SYSTEM)` komutunu vererek kuyruk yöneticisini değiştirerek SSLKEYR parametresini ayarlayın.
   b) `CHGMQM MQMNAME(SSL) SSLKEYRPWD('xxxxxxx')` komutunu çalıştırarak varsayılan anahtar havuzu için parolayı saklayın.
      Parola tek tırnak işareti içinde olmalıdır.
   c) Kanalları, SSLCIPHER parametresinde doğru CipherSpec değerini belirtecek şekilde değiştirin.
   d) `RFRMQMAUT QMNAME(QMGRNAME) TYPE(*SSL)` komutunu vererek TLS güvenliğini yenileyin.
2. Sertifikayı, DCM kullanarak sunucu kuyruk yöneticisine aşağıdaki gibi atayın:
   a) “DCM ' ye Erişilmesi” sayfa 272içinde açıklandığı gibi DCM arabirimine erişin.
   b) Gezinme panosunda **Sertifika Deposu Seç**seçeneğini tıklatın.
      Görev çerçevesinde Sertifika Deposu Seç sayfası görüntülenir.
   c) *SYSTEM sertifika deposunu seçin ve **Continue**(Devam) düğmesini tıklatın.
   d) Sol panoda **Uygulamaları Yönet**seçeneğini genişletin.
   e) Kuyruk yöneticisinin bir uygulama olarak kaydedildiğini denetlemek için **Uygulamayı Görüntüle** tanımlamasını seçin.
      `SSL (WMQ)` çizelgede listelenir.
   f) **Sertifika Atamasını Güncelle**seçeneğini belirleyin.
   g) **Sunucu** seçeneğini belirleyin ve **Devam**düğmesini tıklatın.
   h) QMGRNAME (WMQ) öğesini seçin ve **Sertifika atamasını güncelle**öğesini tıklatın.
   i) Sertifikayı seçin ve **Yeni Sertifika Ata**düğmesini tıklatın. Sertifikanın uygulamaya atandığını belirten bir pencere açılır.

IBM MQ SSL Client Utility (amqrsslc) for IBM i

IBM i için IBM MQ SSL İstemcisi yardımcı programı (amqrsslc), istemci kullanıcı tanıtımını kaydetmek ya da kaydını silmek ya da sertifika deposu parolasını saklamak için IBM MQ MQI client on IBM i sistemleri tarafından kullanılır. Yardımcı program yalnızca *ALLOBJ özel yetkisi olan bir kullanıcı ya da Digital Certificate Manager (DCM) içinde uygulama kayıtları yaratma ya da silme seçenekleri bulunan bir QMQMADM üyesi tarafından çalıştırılabilir.

Sözdizimi şeması



İstemci kullanıcı profilini kaydet

IBM MQ MQI client , *SYSTEM sertifika deposunu kullanıyorsa, istemci kullanıcı tanıtımını (oturum açma kullanıcısı) uygulama olarak kullanmak üzere Digital Certificate Manager (DCM)ile kaydettirmeniz gerekir.

İstemci kullanıcı tanıtımını kaydetmek istiyorsanız, **amqrsslc** programını `-r` seçeneğiyle *UserProfile*ile çalıştırın. **amqrsslc** çağrılırken kullanılan kullanıcı tanıtımının *USE yetkisi olmalıdır. *UserProfile* 'ın `-r` seçeneğiyle birlikte sağlanması, *UserProfile* ' ı sunucu uygulaması olarak QIBM_WEBSPHERE_MQ_*UserProfile* etiketine ve *UserProfile* (WMQ) açıklamasına sahip bir etikete kaydeder. Daha sonra bu sunucu uygulaması DCM ' de görüntülenir ve bu uygulamaya sistem deposundaki herhangi bir sunucu ya da istemci sertifikası atayabilirsiniz.

Not: Bir kullanıcı profili `-r` seçeneğiyle belirtilmezse, **amqrsslc** aracını çalıştıran kullanıcının kullanıcı profili kaydedilir.

Aşağıdaki kod, bir kullanıcı profilini kaydetmek için **amqrsslc** kodunu kullanır. İlk örnekte, belirtilen kullanıcı profili kaydedilir; ikinci örnekte oturum açan kullanıcının profili olur:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-r' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-r')
```

İstemci kullanıcı profilinin kaydını kaldır

İstemci tanıtımının kaydını kaldırmak için, **amqrsslc** programını `-u` seçeneğiyle *UserProfile*ile çalıştırın. **amqrsslc** çağrılırken kullanılan kullanıcı tanıtımının *USE yetkisi olmalıdır. *UserProfile* `-u` seçeneğiyle birlikte sağlanması, DCM ' den QIBM_WEBSPHERE_MQ_*UserProfile* etiketli *UserProfile* kaydını kaldırır.

Not: `-u` seçeneğiyle bir kullanıcı profili belirtilmezse, **amqrsslc** aracını çalıştıran kullanıcının kullanıcı profilinin kaydı kaldırılır.

Aşağıdaki kod, bir kullanıcı profilinin kaydını kaldırmak için **amqrsslc** kodunu kullanır. İlk örnekte, belirtilen kullanıcı profilinin kaydı kaldırılır; saniye içinde oturum açan kullanıcının profili olur:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-u' UserProfile)
CALL PGM(QMQM/AMQRSSLC) PARM('-u')
```

Sertifika deposu parolasını sakla

IBM MQ MQI client , *SYSTEM sertifika deposunu kullanmıyorsa ve başka bir sertifika deposu kullanıyorsa (MQSSLKEYR, *SYSTEM dışında bir değere ayarlandıysa), anahtar veritabanının parolasının , istemci uygulaması tarafındançalıştırıldığında belirtilmesi gerekmemesi için parola kütüğüne saklanabilir.

Anahtar veritabanının parolasını saklamak için `-s` seçeneğini kullanın. V 9.3.0 Anahtar veritabanının tam yolunu ve adını belirtin. Dosya uzantısı sağlanmazsa, dosya uzantısının `.kdb`olduğu varsayılır.

Aşağıdaki kodda, sertifika deposunun tam olarak nitelenmiş dosya adı şöyledir: `/Path/Of/KeyDatabase/MyKey.kdb`:

```
CALL PGM(QMQM/AMQRSSLC) PARM('-s' '/Path/Of/KeyDatabase/MyKey')
```

Bu kodun çalıştırılması, bu anahtar veritabanının parolasıyla ilgili bir istekle sonuçlanır. Bu parola, `.sth` uzantılı anahtar veritabanıyla aynı adı taşıyan bir dosyada saklanır.

V 9.3.0 Ayrıca, parolayı şifrelemek için kullanılan ilk anahtar da belirtilebilir. İlk anahtar bir dosyada tek bir metin satırı olarak saklanmalıdır ve daha sonra, bu dosyanın konumu programa **-sf** işareti aracılığıyla sağlanır. İlk anahtar dosyası sağlanmazsa, parolayı şifrelemek için varsayılan anahtar kullanılır.

Saklama kütüğü anahtar veritabanıyla aynı yolda saklanır. Kod örneği, `/Path/Of/KeyDatabase/MyKey.sth`adlı bir parola saklama dosyası oluşturur.

QMQM kullanıcı sahibi ve QMQMADM bu dosyanın grup sahibi. QMQM ve QMQMADM ' nin okuma, yazma izni var ve diğer tanıtımların yalnızca okuma izni var.

Sertifikalarda ya da sertifika deposunda yapılan değişiklikler IBM i üzerinde yürürlüğe girdiğinde

Bir sertifika deposundaki sertifikaları ya da sertifika deposunun konumunu değiştirdiğinizde, değişiklikler kanalın tipine ve kanalın çalışma şekline bağlı olarak yürürlüğe girer.

Sertifika deposundaki ve anahtar havuzu özniteliğindeki sertifikalarda yapılan değişiklikler aşağıdaki durumlarda etkili olur:

- Yeni bir giden tek kanal işlemi ilk olarak bir TLS kanalı çalıştırdığında.
- Yeni bir gelen TCP/IP tek kanal işlemi ilk olarak TLS kanalı başlatma isteği aldığında.
- IBM MQ TLS ortamını yenilemek için MQSC komutu REFRESH SECURITY TYPE (SSL) verildiğinde.
- İstemci uygulaması işlemleri için, süreçteki son TLS bağlantısı kapatıldığında. Sonraki TLS bağlantısı, sertifika değişikliklerini alır.
- Bir süreç havuzlama işleminin (amqrmppa) iş parçacıkları olarak çalışan kanallar için, süreç havuzlama işlemi başlatıldığında ya da yeniden başlatıldığında ve önce bir TLS kanalı çalıştırıldığında. Süreç havuzlama işlemi zaten bir TLS kanalı çalıştırdıysa ve değişikliğin hemen yürürlüğe girmesini istiyorsanız, MQSC komutunu REFRESH SECURITY TYPE (SSL) çalıştırın.
- Kanal başlatıcısının iş parçacıkları olarak çalışan kanallar için, kanal başlatıcı başlatıldığında ya da yeniden başlatıldığında ve ilk olarak bir TLS kanalı çalıştığında. Kanal başlatıcı işlemi zaten bir TLS kanalı çalıştırdıysa ve değişikliğin hemen yürürlüğe girmesini istiyorsanız, MQSC komutunu REFRESH SECURITY TYPE (SSL) çalıştırın.
- Bir TCP/IP dinleyicisinin iş parçacığı olarak çalışan kanallar için, dinleyici başlatıldığında ya da yeniden başlatıldığında ve ilk olarak TLS kanalı başlatma isteği aldığında. Dinleyici zaten bir TLS kanalı çalıştırdıysa ve değişikliğin hemen yürürlüğe girmesini istiyorsanız, REFRESH SECURITY TYPE (SSL) MQSC komutunu çalıştırın.

IBM i üzerinde şifreleme donanımının yapılandırılması

IBM i üzerinde Cryptographic Coprocessor olanağını yapılandırmak için bu yordamı kullanın.

Başlamadan önce

Yardımcı işlemci donanımının konfigürasyonunu tanımlamanızı sağlamak için kullanıcı tanıtımınızda *ALLOBJ ve *SECADM özel yetkilerinin bulunduğundan emin olun.

Yordam

1. `http://`*`machine`*`.domain:2001` ya da `https://`*`machine`*`.domain:2010`adresine gidin; burada *makine* bilgisayarınızın adıdır.

   Kullanıcı adı ve parola isteyen bir iletişim kutusu görüntülenir.

2. Geçerli bir IBM i kullanıcı tanıtımı ve parolası yazın.
3. Daha fazla bilgi için Şifreleme bölümüne gidin ve uygun bağlantıları izleyin.

Sonraki adım

4767 Cryptographic Coprocessorürününü yapılandırma hakkında daha ayrıntılı bilgi için bkz. 4767 Cryptographic Coprocessor.

ALW AIX, Linux, and Windows üzerinde SSL/TLS ile çalışma

AIX, Linux, and Windows sistemlerinde, TLS (Transport Layer Security; İletim Katmanı Güvenliği) desteği IBM MQile kurulur.

Sertifika doğrulama ilkeleriyle ilgili daha ayrıntılı bilgi için Sertifika doğrulama ve güven ilkesi tasarımıbaşlıklı konuya bakın.

ALW *Dijital sertifikaları yönetmek için runmqckm, runmqakm ve strmqikm kullanılıyor*

AIX, Linux, and Windows sistemlerinde, anahtarları ve dijital sertifikaları **strmqikm** (iKeyman) ile yönetin GUI ya da **runmqckm** (iKeycmd) ya da **runmqakm** (GSKCapiCmd) komutunu kullanarak komut satırından.

Not: V 9.3.4 Deprecated CMS IBM MQ Java uygulamaları için anahtar deposu desteği, AMQP ve MQTT IBM MQ 9.3.4' den kullanımdan kaldırılmıştır. IBM MQ Java uygulamaları, AMQP ve MQTT ile CMS anahtar deposu kullanıyorsanız, IBM MQ 9.3.0içinde yayınlanan PKCS#12 anahtar havuzu desteğine geçmeniz gerekir.

runmqckm, **strmqikm**, **mqiptKeycmd** ve **mqiptKeyman** araçları da kullanımdan kaldırılmıştır. IBM MQ ve JRE ' deki **keytool** komutu için sağlanan **runmqakm** komutu alternatif olarak kullanılabilir.

Uyarı: runmqckm ve **strmqikm** komutlarının her ikisi de IBM MQ Java Runtime Environment 'a (JRE) dayanır. IBM MQ 9.1' den JRE kurulu değilse, AMQ9183iletisini alırsınız.

- Linux AIX **AIX and Linux** sistemleri için:
  - iKeyman GUI 'sini başlatmak için **strmqikm** (iKeyman) komutunu kullanın.
  - Görevleri komut satırı arabirimiyle gerçekleştirmek için **runmqckm** komutunu kullanın.
  - Runmqakm komut satırı arabirimiyle görevleri gerçekleştirmek için **runmqakm** (GSKCapiCmd) komutunu kullanın. **runmqakm** komut sözdizimi, **runmqckm**sözdizimiyle aynıdır.

    TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqckm** ya da **strmqikm** komutları yerine **runmqakm** komutunu kullanın.

  runmqckm ve **runmqakm** komutlarına ilişkin komut satırı arabirimlerinin tam açıklaması için Anahtarların ve sertifikaların yönetilmesi başlıklı konuya bakın.

  PKCS #11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve iKeyman ' ın 64 bitlik programlar olduğunu unutmayın. PKCS #11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS #11 kitaplığınızın kurulu olması gerekir. Windows ve Linux x86 32 bit platformları, bu platformlarda iKeyman ve **runmqckm** programları 32 bit olduğu için tek istisnadır.

  Daha fazla bilgi için bkz. IBM Global Security Kit (GSKit): PKCS#11 ve IBM MQ JRE adresleme kipi .

  iKeyman GUI 'sini başlatmak için **strmqikm** komutunu çalıştırmadan önce, X Pencere Sistemi 'ni çalıştırabilen bir makine üzerinde çalıştığınızdan ve aşağıdakileri yaptığınızdan emin olun:

  - DISPLAY ortam değişkenini ayarlayın, örneğin:

```
export DISPLAY=mypc:0
```

- PATH ortam değişkeninizin **/usr/bin** ve **/bin**içerdiğini doğrulayın. Bu, **runmqckm** ve **runmqakm** komutları için de gereklidir. Örneğin:

```
export PATH=$PATH:/usr/bin:/bin
```

- Windows **Windows** sistemleri için:
  - iKeyman GUI 'sini başlatmak için **strmqikm** komutunu kullanın.
  - Görevleri komut satırı arabirimiyle gerçekleştirmek için **runmqckm** komutunu kullanın.

    TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqckm** ya da **strmqikm** komutları yerine **runmqakm** komutunu kullanın.
  - **runmqakm -keydb** komutunu *stashpw* ya da *stash* seçeneğiyle kullanın.

    runmqakm -keydb komutunu bu şekilde kullanırken, örneğin:

```
runmqakm -keydb -create -db key.kdb -pw secretpwd -stash
```

    sonuçtaki `.sth` dosyasında `mqm` grubu için okuma izni etkinleştirilmemiş.

    Dosyayı yalnızca yaratan okuyabilir. **runmqakm** komutunu kullanarak bir saklama dosyası oluşturduktan sonra, dosya izinlerini denetleyin ve kuyruk yöneticisini çalıştıran hizmet hesabına ya da yerel `mqm`gibi bir gruba izin verin.

ALW AIX, Linux, and Windows sistemlerinde TLS izlemesini istemek için bkz. strmqtrc.

İlgili başvurular

“AIX, Linux, and Windows üzerinde runmqckm ve runmqakm komutları” sayfa 553
Bu bölümde, komutun nesnesine göre **runmqckm** ve **runmqakm** komutları açıklanmaktadır.

ALW *AIX, Linux, and Windows üzerinde bir anahtar havuzu ayarlama*

strmqikm (iKeyman) komutunu kullanarak bir anahtar havuzu ayarlayabilirsiniz. GUI ya da **runmqckm** (iKeycmd) ya da **runmqakm** (GSKCapiCmd) komutlarını kullanarak komut satırından.

Başlamadan önce

Anahtar havuzu, hassas bilgiler içerdiği için bir parolayla korunmuştur. Anahtar deposunu yaratmadan önce, anahtar havuzu parolasını güvenli bir şekilde saklamak için IBM MQ tarafından sağlanan seçenekleri gözden geçirin. Daha fazla bilgi için bkz “AIX, Linux, and Windows üzerinde anahtar havuzu parolalarını şifreleme” sayfa 293.

Not: V 9.3.4 Deprecated CMS IBM MQ Java uygulamaları için anahtar deposu desteği, AMQP ve MQTT IBM MQ 9.3.4' den kullanımdan kaldırılmıştır. IBM MQ Java uygulamaları, AMQP ve MQTT ile CMS anahtar deposu kullanıyorsanız, IBM MQ 9.3.0içinde yayınlanan PKCS#12 anahtar havuzu desteğine geçmeniz gerekir.

runmqckm, **strmqikm**, **mqiptKeycmd** ve **mqiptKeyman** araçları da kullanımdan kaldırılmıştır. IBM MQ ve JRE ' deki **keytool** komutu için sağlanan **runmqakm** komutu alternatif olarak kullanılabilir.

Bu görev hakkında

TLS bağlantısı, bağlantının her sonunda bir *anahtar havuzu* gerektirir. Her IBM MQ kuyruk yöneticisinin ve IBM MQ MQI client ' in bir anahtar havuzuna erişimi olmalıdır. Daha fazla bilgi için bkz “SSL/TLS anahtar havuzu” sayfa 25.

AIX, Linux, and Windows sistemlerinde sayısal sertifikalar, **strmqikm** kullanıcı arabirimi kullanılarak ya da **runmqckm** ya da **runmqakm** komutları kullanılarak yönetilen bir anahtar veritabanı dosyasında saklanır. Bu dijital sertifikaların etiketleri var. Belirli bir etiket, kişisel sertifikayı bir kuyruk yöneticisiyle ya da IBM MQ MQI clientile ilişkilendirir. TLS, kimlik doğrulama amacıyla bu sertifikayı kullanır. AIX, Linux, and Windows sistemlerinde IBM MQ , ayarlandıysa **CERTLABL** özniteliğinin değerini ya da kuyruk yöneticisinin adı ya da

sonuna IBM MQ MQI client kullanıcı oturum açma kimliği eklenmiş olarak varsayılan `ibmwebspheremq` değerini kullanır. Ayrıntılar için bkz. Dijital sertifika etiketleri .

Anahtar veritabanı dosyası adı bir yol ve kök addan oluşuyor:

- AIX and Linux sistemlerinde, bir kuyruk yöneticisinin varsayılan yolu (kuyruk yöneticisini yarattığınızda ayarlanır) şudur: `/var/mqm/qmgrs/`*`queue_manager_name`*`/ssl`.

  Windows sistemlerinde varsayılan yol *`MQ_INSTALLATION_PATH`*`\Qmgrs\`*`queue_manager_name`*`\ssl`dizinidir; burada *`MQ_INSTALLATION_PATH`* , IBM MQ ' in kurulu olduğu dizindir. Örneğin, `C:\Program Files\IBM\MQ\Qmgrs\QM1\ssl`.

  V 9.3.0 Varsayılan dosya adı: `key.kdb`. İsteğe bağlı olarak, kendi yol ve dosya adınızı kullanabilirsiniz.

  Kendi yolunuzu ya da dosya adınızı seçerseniz, dosyaya erişimi sıkı bir şekilde denetlemek için dosyanın izinlerini ayarlayın.

- V 9.3.0 Bir IBM MQ istemcisi için varsayılan yol ya da dosya adı yoktur. Bu dosyaya erişimi sıkı bir şekilde denetleyin.

Dosya düzeyi kilitlerini desteklemeyen bir dosya sisteminde anahtar havuzları yaratmayın; örneğin, Linux sistemlerinde NFS sürüm 2.

Anahtar veritabanı dosyası adının denetlenmesine ve belirtilmesine ilişkin bilgi için bkz. “AIX, Linux, and Windows üzerinde bir kuyruk yöneticisi için anahtar havuzu konumunu değiştirme” sayfa 297 . Anahtar veri tabanı dosyası yaratılmadan önce ya da yaratıldıktan sonra anahtar veri tabanı dosyası adını belirtebilirsiniz.

strmqikm ya da **runmqckm** komutlarını çalıştırdığınız kullanıcı kimliğinin, anahtar veritabanı dosyasının yaratıldığı ya da güncellendiği dizin için yazma izni olmalıdır. Varsayılan `ssl` dizinini kullanan bir kuyruk yöneticisi için, **strmqikm** ya da **runmqckm** komutunu çalıştırdığınız kullanıcı kimliği mqm grubunun bir üyesi olmalıdır. Bir IBM MQ MQI clientiçin, **strmqikm** ya da **runmqckm** istemcinin çalıştığından farklı bir kullanıcı kimliğinden çalıştırılırsa, IBM MQ MQI client ' ın yürütme sırasında anahtar veritabanı dosyasına erişmesini sağlamak için dosya izinlerini değiştirmeniz gerekir. Daha fazla bilgi için bkz. “Windows üzerinde anahtar veritabanı dosyalarınıza erişilmesi ve bunların güvenliğinin sağlanması” sayfa 294 ya da “AIX and Linux sistemlerinde anahtar veritabanı dosyalarınıza erişilmesi ve bunların güvenliğinin sağlanması” sayfa 295.

strmqikm ya da **runmqckm** for IBM Global Security Kit (GSKit) version 7.0' da, yeni anahtar veritabanlarına otomatik olarak bir dizi önceden tanımlanmış sertifika yetkilisi (CA) sertifikası yerleştirilir. Anahtar veritabanı dosyanıza yalnızca istediğiniz CA sertifikalarını eklediğiniz için, **strmqikm** ya da GSKit 8.0için **runmqckm** anahtar veritabanlarına otomatik olarak veri yerleştirilmez.

Not: GSKit 8.0 davranışındaki bu değişiklik CA sertifikalarının artık havuza otomatik olarak eklenmesine neden olmadığından, tercih ettiğiniz CA sertifikalarını el ile eklemeniz gerekir. Bu davranış değişikliği, kullanılan CA sertifikaları üzerinde daha ayrıntılı denetim sağlar. Bkz. “GSKit 8.0 ile AIX, Linux, and Windows üzerinde boş bir anahtar havuzuna varsayılan CA sertifikalarının eklenmesi” sayfa 295.

Anahtar veritabanını komut satırını kullanarak ya da **strmqikm** (iKeyman) kullanıcı arabirimini kullanarak yaratırsınız.

Not: TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın. **strmqikm** kullanıcı arabirimi, FIPS uyumlu bir seçenek sağlamaz.

Yordam

Komut satırını kullanarak bir anahtar veritabanı yaratın.

1. Aşağıdaki komutlardan birini çalıştırın:

- **runmqckm**komutunu kullanarak:

  V 9.3.0 V 9.3.0
  ```
  runmqckm -keydb -create -db filename -pw password -type cms | p12 -stash
  ```

- **runmqakm**komutunu kullanarak:

  V 9.3.0 V 9.3.0
  ```
  runmqakm -keydb -create -db filename -pw password -type cms | p12
  -stash -fips -strong
  ```

Burada:

-db *kütükadı*
CMS anahtar veritabanınınolmalıdır.

-pw *parola*
CMS V 9.3.0 V 9.3.0 ya da PKCS#12 anahtar veritabanına ilişkin parolayı belirtir.

V 9.3.0 V 9.3.0 **-type *cms* | *p12***
Veri tabanının tipini belirtir. (IBM MQiçin cms ya da pkcs12olmalıdır).

-stash
V 9.3.0 V 9.3.0 İsteğe bağlı. Anahtar veritabanı parolasını bir dosyaya kaydeder. Anahtar veritabanı parolasını bir parola saklama dosyasında saklamak için bu seçeneği belirleyin. Parolayı IBM MQ parola koruma sistemini kullanarak şifrelediyseniz, parolayı bir parola saklama dosyasında saklamanız gerekmez.

-fips
Komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.

-Güçlü
Girilen parolanın, parola güvenlik düzeyi için minimum gereksinimleri karşıladığını denetler. Bir parola için minimum gereksinimler şunlardır:

- Parola en az 14 karakter uzunluğunda olmalıdır.
- Parola en az bir küçük harf, bir büyük harf ve bir rakam ya da özel karakter içermelidir. Özel karakterler arasında yıldız işareti (*), dolar işareti ($), sayı işareti (#) ve yüzde işareti (%) yer alır. Boşluk özel karakter olarak sınıflandırılır.
- Her karakter bir parolada en çok üç kez geçebilir.
- Parolada en çok iki ardışık karakter aynı olabilir.
- Tüm karakterler standart ASCII yazdırılabilir karakter takımında, 0x20 - 0x7Earalığındadır.

Alternatif olarak, **strmqikm** (iKeyman) kullanıcı arabirimini kullanarak bir anahtar veritabanı oluşturun.

2. AIX and Linux sistemlerinde kök kullanıcı olarak oturum açın. Windows sistemlerinde Administrator (Yönetici) ya da MQM grubunun bir üyesi olarak oturum açın.
3. **strmqikm** komutunu çalıştırarak kullanıcı arabirimini başlatın.
4. **Anahtar Veritabanı Dosyası** menüsünden **Yeni**' yi tıklatın.
   Yeni penceresi açılır.
5. **Anahtar veritabanı tipi** ' ni tıklatın ve **CMS** (Sertifika Yönetimi Sistemi) V 9.3.0 V 9.3.0 ya da **PKCS#12**seçeneğini belirleyin.
6. **Dosya Adı** alanında bir dosya adı yazın.
   Bu alan zaten key . kdb V 9.3.0 V 9.3.0 ya da key . p12metnini içerir. Kök adınız keyise, bu alanı değiştirmeden bırakın. Farklı bir kök adı belirttiyseniz, key yerine kök adınızı koyun. değiştirmemelisiniz.
7. **Konum** alanına yolu yazın.

Örneğin:

- Bir kuyruk yöneticisi için: `/var/mqm/qmgrs/QM1/ssl` (AIX and Linux sistemlerinde) ya da `C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl` (Windows sistemlerinde).

  Yol, kuyruk yöneticisinin **SSLKeyRepository** özniteliğinin değeriyle eşleşmelidir.
- Bir IBM MQ istemcisi için: `/var/mqm/ssl` (AIX and Linux sistemlerinde) ya da `C:\mqm\ssl` (Windows sistemlerinde).

8. **Tamam**'ı tıklatın.

   Parola İstemi penceresi görüntülenir.
9. **Parola** alanında bir parola yazın ve **Parolayı Onayla** alanına parolayı yeniden yazın.
10. V 9.3.0

    İsteğe bağlı: Anahtar veritabanı parolasını bir dosyaya kaydetmek için **Parolayı bir dosyaya kaydet** onay kutusunu işaretleyin.

    Anahtar veritabanı parolasını bir parola saklama dosyasında saklamak için bu seçeneği belirleyin. Parolayı IBM MQ parola koruma sistemini kullanarak şifrelediyseniz, parolayı bir parola saklama dosyasında saklamanız gerekmez.
11. **Tamam**'ı tıklatın.

    Kişisel Sertifikalar penceresi görüntülenir.
12. Erişim izinlerini "Windows üzerinde anahtar veritabanı dosyalarınıza erişilmesi ve bunların güvenliğinin sağlanması" sayfa 294 ya da "AIX and Linux sistemlerinde anahtar veritabanı dosyalarınıza erişilmesi ve bunların güvenliğinin sağlanması" sayfa 295içinde açıklandığı gibi ayarlayın.
13. V 9.3.0

    Parola saklama dosyası kullanmıyorsanız, "AIX, Linux, and Windows üzerinde bir kuyruk yöneticisi için anahtar havuzu parolası sağlanması" sayfa 297 ya da "AIX, Linux, and Windows üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması" sayfa 299içindeki yönergeleri izleyerek anahtar deposu parolasını kuyruk yöneticisine ya da istemci uygulamasına sağlayın.

V 9.3.0 V 9.3.0 ALW *AIX, Linux, and Windows üzerinde anahtar havuzu parolalarını şifreleme*

Bazı IBM MQ bileşenlerinin sayısal sertifikaları ya da simetrik anahtarları içeren bir anahtar havuzuna erişmesi gerekir. Bir anahtar havuzu, hassas bilgiler içerdiği için bir parolayla korunmuştur. Anahtar havuzu parolası, anahtar havuzuna erişildiğinde IBM MQ ' un okuyabileceği bir konumda saklanmalıdır. Anahtar havuzuna yetkisiz erişim olasılığını azaltmak için parola da şifrelenmelidir.

Aşağıdaki IBM MQ bileşenleri ve özellikleri, anahtar havuzu parolalarını saklamak için iki farklı yöntemi destekler:

- Kuyruk yöneticisi TLS anahtar havuzu.
- TLS kullanan IBM MQ MQI clients .
- V 9.3.2 `qm.ini` dosyasının **NativeHALocalInstance** kısmına ilişkin Native HA yapılandırması.
- V 9.3.4 `qm.ini` dosyasının **AuthToken** kısmına ilişkin simge kimlik doğrulama yapılandırması.

Bu bileşenler tarafından kullanılan anahtar havuzu parolaları, aşağıdaki yöntemlerden biri kullanılarak şifrelenebilir ve saklanabilir:

IBM MQ parola koruma sistemi.

Her IBM MQ bileşeni, anahtar havuzu parolasını şifrelemek için bir komut sağlar. Komut çıkışlarının bir dosyada saklandığı şifrelenmiş komut.

Kuyruk yöneticisi TLS anahtar havuzu için parola, **SSLKEYRPWD** kuyruk yöneticisi özniteliği ayarlandığında şifrelenir.

Parola, AES-128 algoritmasıyla şifrelenir. Bu algoritmanın ayrıntıları kamuya açıktır ve güvenli olarak kabul edilir.

Parola, anahtar havuzuna erişebilecek diğer yazılımlar tarafından anlaşılmayan özel bir biçimde saklanır.

Bir IBM MQ bileşeni tarafından şifrelenen bir parola farklı bir IBM MQ bileşeni tarafından kullanılamaz.

Anahtar havuzu parolası şifrelendiğinde benzersiz bir şifreleme anahtarı sağlanabilir. Benzersiz bir şifreleme anahtarı, şifreleme anahtarına erişimi olmayan herkesin parolanın şifresini çözmesini önler.

Anahtar havuzundaki sertifikaları yönetmek için düz metin anahtarı havuzu parolası gerekir. Anahtar havuzu parolasını IBM MQ parola koruma sistemini kullanarak şifrelemenin yanı sıra, anahtar havuzu parolasını da bu amaçla erişilebilecek güvenli bir yerde saklamanız gerekir.

IBM MQ parola koruma sistemi hakkında daha fazla bilgi için bkz. “IBM MQ bileşeni yapılandırma dosyalarındaki parolaları koruma” sayfa 575.

Anahtar havuzu saklama dosyası.

runmqakm ve **runmqckm** komutları, anahtar havuzu parolasını bir parola saklama dosyasında saklayabilir.

Parola, IBM MQ şifreleme sağlayıcısına IBM Global Security Kit (GSKit)özgü özel bir yöntemle şifrelenir.

Benzersiz bir şifreleme anahtarı sağlanamıyor.

Şifrelenmiş parola, anahtar havuzu dosyasıyla aynı dizindeki bir parola saklama dosyasında saklanır.

Hem anahtar havuzuna hem de saklama dosyasına okuma erişimi olan herkes, anahtar havuzunun içeriğine erişebilir ve bunları yönetebilir.

Anahtar havuzu parolasını şifrelemeyi seçtiğiniz yöntemden bağımsız olarak, saklanan parolaları şifreleme sınırlamalarını bildiğinizden emin olun. Daha fazla bilgi için bkz “Parola şifreleme yoluyla koruma sınırları” sayfa 583.

İlgili kavramlar

“AIX, Linux, and Windows üzerinde bir kuyruk yöneticisi için anahtar havuzu parolası sağlanması” sayfa 297
Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla güvenli hale getirilmiştir. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

“AIX, Linux, and Windows üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması” sayfa 299
Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla güvenli hale getirilmiştir. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

“AIX, Linux, and Windows üzerinde SSL/TLS ile çalışma” sayfa 289
AIX, Linux, and Windows sistemlerinde, TLS (Transport Layer Security; İletim Katmanı Güvenliği) desteği IBM MQile kurulur.

Windows *Windows üzerinde anahtar veritabanı dosyalarınıza erişilmesi ve bunların güvenliğinin sağlanması*
Anahtar veritabanı dosyaları uygun erişim izinlerine sahip olmayabilir. Bu dosyalara uygun erişimi ayarlamanız gerekir.

Sınırlı bir kullanıcı kümesine yetki vermek için, V 9.3.0 V 9.3.0 *key*.p12, *key*.kdb, *key*.sth, *key*.crlve *key*.rdbdosyalarına erişim denetimini ayarlayın; burada *anahtar* , anahtar veritabanınızın kök adıdır.

V 9.3.0 V 9.3.0 .p12 ya da .kdbdışında farklı bir anahtar havuzu uzantısı kullandıysanız, bu dosyanın izinlerinin de ayarlandığından emin olmanız gerekir.

Aşağıdaki gibi erişim vermeyi düşünün:

tam yetki
BUILTIN\Administrators, NT AUTHORITY\SYSTEM ve veri tabanı dosyalarını yaratan kullanıcı.

okuma yetkisi
Bir kuyruk yöneticisi için yalnızca yerel mqm grubu. Bu, MCA ' nın mqm grubundaki bir kullanıcı kimliği altında çalıştığını varsayar.

Bir istemci için, istemci işleminin altında çalıştığı kullanıcı kimliği.

Linux AIX *AIX and Linux sistemlerinde anahtar veritabanı dosyalarınıza erişilmesi ve bunların güvenliğinin sağlanması*
Anahtar veritabanı dosyaları uygun erişim izinlerine sahip olmayabilir. Bu dosyalara uygun erişimi ayarlamanız gerekir.

Bir kuyruk yöneticisi için, kuyruk yöneticisi ve kanal işlemlerinin gerektiğinde bunları okuyabilmesi, ancak diğer kullanıcıların bunları okuyabilmesi ya da değiştirebilmesi için anahtar veritabanı dosyalarına ilişkin izinleri ayarlayın. Olağan durumda, mqm kullanıcısının okuma izinlerine ihtiyacı vardır. Anahtar veritabanı dosyasını mqm kullanıcısı olarak oturum açarak yarattıysanız, izinler büyük olasılıkla yeterlidir; mqm kullanıcısı değilseniz, ancak mqm grubundaki başka bir kullanıcı varsa, mqm grubundaki diğer kullanıcılara okuma izinleri vermeniz gerekir.

Bir istemci için de benzer şekilde, istemci uygulaması işlemlerinin gerektiğinde okuyabilmesi, ancak diğer kullanıcıların bunları okuyabilmesi ya da değiştirebilmesi için anahtar veritabanı dosyalarına ilişkin izinleri ayarlayın. Olağan durumda, istemci işleminin çalıştırıldığı kullanıcının okuma izinlerine gereksinimi vardır. Anahtar veritabanı dosyasını o kullanıcı olarak oturum açarak yarattıysanız, izinler büyük olasılıkla yeterlidir; istemci işlemi kullanıcısı değilseniz, ancak o gruptaki başka bir kullanıcı değilseniz, büyük olasılıkla gruptaki diğer kullanıcılara okuma izinleri vermeniz gerekir.

V 9.3.0 V 9.3.0 *key*.p12, *key*.kdb, *key*.sth, *key*.crlve *key*.rdbdosyalarına ilişkin izinleri ayarlayın; burada *anahtar* , anahtar veritabanınızın kök adıdır, dosya sahibi için okuma ve yazma ve mqm ya da istemci kullanıcı grubu için okuma (-rw-r -----).

V 9.3.0 V 9.3.0 .p12 ya da .kdbdışında farklı bir anahtar havuzu uzantısı kullandıysanız, bu dosyanın izinlerinin de ayarlandığından emin olmanız gerekir.

ALW *GSKit 8.0 ile AIX, Linux, and Windows üzerinde boş bir anahtar havuzuna varsayılan CA sertifikalarının eklenmesi*
IBM Global Security Kit (GSKit) sürüm 8.0olan boş bir anahtar havuzuna varsayılan CA sertifikalarından birini ya da daha fazlasını eklemek için bu yordamı izleyin.

GSKit 7.0içinde, yeni bir anahtar havuzu oluşturma davranışı, yaygın olarak kullanılan Sertifika Yetkilileri için varsayılan CA sertifikaları kümesini otomatik olarak eklemekti. GSKit 8.0için bu davranış, CA sertifikalarının artık havuza otomatik olarak eklenmeyecek şekilde değişmiştir. Kullanıcının artık anahtar havuzuna CA sertifikalarını el ile eklemesi gerekir.

Kullanılan `strmqikm`

CA sertifikasını eklemek istediğiniz makinede aşağıdaki adımları gerçekleştirin:

1. **`strmqikm`** komutunu kullanarak (AIX, Linux, and Windowsüzerinde) GUI ' yi başlatın.
2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın. Aç penceresi açılır.
3. **Anahtar veritabanı tipi** seçeneğini tıklatın ve **CMS** (Sertifika Yönetimi Sistemi)V 9.3.0 V 9.3.0 ya da PKCS#12seçeneğini belirleyin.
4. Anahtar veri tabanı dosyalarını içeren dizine gitmek için **Göz At** düğmesini tıklatın.
5. Sertifikayı eklemek istediğiniz anahtar veritabanı dosyasını seçin; örneğin, key.kdb.
6. **Aç**' ı tıklatın. Password Prompt (Parola İstemi) penceresi açılır.
7. Anahtar veritabanını yaratırken ayarladığınız parolayı yazın ve **Tamam**düğmesini tıklatın. Anahtar veritabanı dosyanızın adı **Dosya Adı** alanında görüntülenir.

8. **Anahtar veritabanı içeriği** alanında **İmzalayan Sertifikaları**seçeneğini belirleyin.
9. **Doldur**düğmesini tıklatın. Add CA's Certificate (CA ' nın Sertifikasını Ekle) penceresi açılır.
10. Havuza eklenebilecek CA sertifikaları sıradüzenli bir ağaç yapısında görüntülenir. Geçerli CA sertifikalarının tam listesini görüntülemek için CA sertifikalarına güvenmek istediğiniz kuruluşun en üst düzey girişini seçin.
11. Güvenmek istediğiniz CA sertifikalarını listeden seçin ve **Tamam**' ı tıklatın. Sertifikalar anahtar havuzuna eklenir.

Komut satırının kullanılması

Listelemek için aşağıdaki komutları kullanın ve **runmqckm**komutunu kullanarak CA sertifikaları ekleyin:

- Varsayılan CA sertifikalarını, bunları yayınlayan kuruluşlarla birlikte listelemek için aşağıdaki komutu verin:

```
runmqckm -cert -listsigners
```

- *label* (etiket) alanında belirtilen kuruluşa ilişkin tüm CA sertifikalarını eklemek için aşağıdaki komutu verin:

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

Burada:

-db *filename*	anahtar veritabanının tam olarak nitelenmiş yol adıdır.
-pw *password*	anahtar veritabanının parolasıdır.
-label *label*	sertifikaya iliştirilen etikettir.

Not: Anahtar havuzuna bir CA sertifikası eklenmesi, IBM MQ ' in bu CA sertifikası tarafından imzalanan tüm kişisel sertifikalara güvenmesiyle sonuçlanır. Güvenmek istediğiniz Sertifika Yetkililerini dikkatle göz önünde bulundurun ve yalnızca istemcilerinizi ve yöneticilerinizi doğrulamak için gereken CA sertifikaları kümesini ekleyin. Bu, güvenlik ilkeniz için kesin bir gereklilik değilse, varsayılan CA sertifikalarının tam kümesini eklemeniz önerilmez.

ALW *AIX, Linux, and Windows üzerinde bir kuyruk yöneticisine ilişkin anahtar havuzunun bulunması*

Kuyruk yöneticinizin anahtar veritabanı dosyasının yerini almak için bu yordamı kullanın.

Yordam

1. Aşağıdaki MQSC komutlarından birini kullanarak kuyruk yöneticinizin özniteliklerini görüntüleyin:

```
DISPLAY QMGR ALL
DISPLAY QMGR SSLKEYR
```

   Kuyruk yöneticinizin özniteliklerini IBM MQ Explorer ya da PCF komutlarını kullanarak da görüntüleyebilirsiniz.
2. Anahtar veritabanı dosyasının yolu ve kök adı için komut çıkışını inceleyin.
   Örneğin,
   a. AIX and Linux: `/var/mqm/qmgrs/QM1/ssl/key`üzerinde, burada `/var/mqm/qmgrs/QM1/ssl` yol, `key` ise kök addır
   b. Windows: *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl\key`üzerinde, burada *MQ_INSTALLATION_PATH*`\qmgrs\QM1\ssl` yol, `key` ise kök addır. *MQ_INSTALLATION_PATH* , IBM MQ ' in kurulu olduğu üst düzey dizini gösterir.

Not: V 9.3.0 V 9.3.0 IBM MQ 9.3.0 ' den SSLKEYR alanı, hem tam dosya adını (uzantı da içinde olmak üzere) hem de bir kök adını (uzantı olmadan) destekler. Bir kök adı ayarlanırsa, IBM MQ otomatik olarak eklenir `.kdb` ve bu anahtar havuzunu kullanır.

ALW *AIX, Linux, and Windows üzerinde bir kuyruk yöneticisi için anahtar havuzu konumunu değiştirme*

Kuyruk yöneticinizin anahtar veritabanı dosyasının yerini, ALTER QMGR MQSC komutu da içinde olmak üzere çeşitli yollarla değiştirebilirsiniz.

Kuyruk yöneticinizin anahtar veritabanı dosyasının yerini, kuyruk yöneticinizin anahtar havuzu özniteliğini ayarlamak için ALTER QMGR MQSC komutunu kullanarak değiştirebilirsiniz. Örneğin, AIX and Linuxüzerinde:

V 9.3.0 V 9.3.0
```
ALTER QMGR SSLKEYR('/var/mqm/qmgrs/QM1/ssl/MyKey.kdb')
```

Windows'ta:

V 9.3.0 V 9.3.0
```
ALTER QMGR SSLKEYR('C:\Program Files\IBM\MQ\Qmgrs\QM1\ssl\Mykey.kdb')
```

Anahtar veritabanı dosyası tam olarak nitelenmiş dosya adına sahip: `C:\Program Files\IBM\MQ\Qmgrs\QM1\ssl\Mykey.kdb`

V 9.3.0 V 9.3.0

Uyarı: Windows ve Linuxüzerinde TLS AMQP kanalları kullanılıyorsa, anahtar havuzu dosyasının soneki aşağıdakilerden biri olmalıdır:

- `.kdb`, CMS anahtar havuzu için
- Bir PKCS #12 anahtar havuzu için `.p12` ya da `.pkcs12`.

Kuyruk yöneticinizin özniteliklerini IBM MQ Explorer ya da PCF komutlarını kullanarak da değiştirebilirsiniz.

Bir kuyruk yöneticisinin anahtar veritabanı dosyasının yerini değiştirdiğinizde, sertifikalar eski yerden aktarılmaz. Şu anda erişmekte olduğunuz anahtar veritabanı dosyası yeni bir anahtar veritabanı dosyasıdır, bu dosyayı “AIX, Linux, and Windows üzerindeki bir anahtar havuzuna kişisel sertifika aktarılması” sayfa 316içinde açıklandığı şekilde, gereksinim duyduğunuz CA ve kişisel sertifikalarla doldurmanız gerekir.

V 9.3.0 V 9.3.0 *AIX, Linux, and Windows üzerinde bir kuyruk yöneticisi için anahtar havuzu parolası sağlanması*

Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla güvenli hale getirilmiştir. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

IBM MQ , bir kuyruk yöneticisine anahtar havuzu parolası sağlamak için iki mekanizma sağlar:

- “KEYRPWD özniteliği” sayfa 298
- “Anahtar havuzu saklama dosyası” sayfa 298

Bir anahtar havuzu parola saklama dosyası kullanmazsanız, anahtar havuzu parolası IBM MQ parola koruma sistemi kullanılarak şifrelenir. Anahtar havuzu parolasını koruma yöntemleri hakkında daha fazla bilgi için bkz. “AIX, Linux, and Windows üzerinde anahtar havuzu parolalarını şifreleme” sayfa 293.

KEYRPWD özniteliği

Bir anahtar havuzu parolasını doğrudan kuyruk yöneticisine sağlamak için, *password* yerine anahtar havuzu parolanızı koyarak aşağıdaki MQSC komutunu çalıştırın:

```
ALTER QMGR KEYRPWD('password')
```

Uyarı: Parolayı tek tırnak işareti içine aldığınızdan emin olun, tersi durumda IBM MQ karakterleri büyük harfe dönüştürür.

Bu yöntem kullanılarak bir anahtar havuzu parolası belirtildiğinde, parola saklanmadan önce IBM MQ parola koruma sistemi kullanılarak şifrelenir.

Parolayı şifrelemek için ilk anahtar olarak bilinen bir şifreleme anahtarı kullanılır. Kuyruk yöneticisini, parolayı güvenli bir şekilde korumak için benzersiz bir başlangıç anahtarı kullanacak şekilde ayarlayın. Başlangıç anahtarı belirtmezseniz, varsayılan anahtar kullanılır.

Anahtar havuzu parolasını ayarlamadan önce kuyruk yöneticisinin benzersiz bir başlangıç anahtarıyla yapılandırıldığından emin olun. **ALTER QMGR** komutundaki **INITKEY** özniteliğini kullanarak ilk anahtarı değiştirebilirsiniz. Örneğin:

```
ALTER QMGR INITKEY('mykey')
```

Uyarı: Anahtar havuzu parolasını ayarladıktan sonra ilk anahtarın değiştirilmesi, anahtar havuzu parolasının yeni başlangıç anahtarıyla şifrelenmesine neden olmaz. Anahtar havuzu parolasını sıfırlamadan ilk anahtarın değiştirilmesi, IBM MQ ' in anahtar havuzu parolasının şifresini çözememesine ve bu nedenle anahtar havuzuna erişememesine neden olur.

KEYRPWD özniteliği hakkında daha fazla bilgi için bkz. KEYRPWD.

Anahtar havuzu saklama dosyası

Kuyruk yöneticisine **KEYRPWD** özniteliği kullanılarak bir anahtar havuzu parolası sağlanmazsa, IBM MQ anahtar havuzuyla aynı dizinde bir parola saklama dosyası olduğunu varsayar. Saklama dosyası, anahtar havuzuyla aynı kök ada sahip, ancak `.sth` uzantısına sahip.

Anahtar havuzu saklama dosyası, anahtar havuzuyla aynı anda ya da daha sonra ayrı bir **runmqakm** komutu olarak yaratılır.

Uyarı: Şifreleme dosyasının biçimi IBM MQ şifreleme sağlayıcısına IBM Global Security Kit (GSKit)özgüdür ve farklı bir şifreleme sağlayıcısı kullanan platformlarda kullanılamaz.

Anahtar havuzu yaratıldığında bir saklama dosyası yaratmak için **-stash** değiştirgesini belirtin. Örneğin:

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

Burada *passw0rd* , anahtar havuzu parolasıdır.

Daha sonra bir saklama dosyası yaratmak için aşağıdaki komutu çalıştırın:

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

Burada *passw0rd* , anahtar havuzu parolasıdır.

İlgili kavramlar

“AIX, Linux, and Windows üzerinde anahtar havuzu parolalarını şifreleme” sayfa 293
Bazı IBM MQ bileşenlerinin sayısal sertifikaları ya da simetrik anahtarları içeren bir anahtar havuzuna erişmesi gerekir. Bir anahtar havuzu, hassas bilgiler içerdiği için bir parolayla korunmuştur. Anahtar havuzu parolası, anahtar havuzuna erişildiğinde IBM MQ ' un okuyabileceği bir konumda saklanmalıdır. Anahtar havuzuna yetkisiz erişim olasılığını azaltmak için parola da şifrelenmelidir.

“AIX, Linux, and Windows üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması” sayfa 299

Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla güvenli hale getirilmiştir. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

ALW AIX, Linux, and Windows üzerinde IBM MQ MQI client için anahtar havuzunun bulunması

Anahtar havuzunun yeri MQSSLKEYR değişkeni tarafından verilir ya da MQCONNX çağrısında belirtilir.

IBM MQ MQI clientile ilgili anahtar veritabanı dosyasının yerini bulmak için MQSSLKEYR ortam değişkenini inceleyin. Örneğin:

```
echo $MQSSLKEYR
```

Anahtar veritabanı dosyası adı"AIX, Linux, and Windows üzerinde IBM MQ MQI client için anahtar havuzu konumunu belirtme" sayfa 299içinde açıklandığı gibi bir MQCONNX çağrısında da ayarlanabileceğinden uygulamanızı da denetleyin. MQCONNX çağrısında ayarlanan değer, MQSSLKEYR değerini geçersiz kılar.

ALW AIX, Linux, and Windows üzerinde IBM MQ MQI client için anahtar havuzu konumunu belirtme

Bir IBM MQ MQI clientiçin varsayılan anahtar havuzu yoktur. Konumunu iki şekilde de belirtebilirsiniz. Anahtar veritabanı dosyasına, diğer sistemlere yetkisiz kopyalamayı önlemek için yalnızca amaçlanan kullanıcılar ya da yöneticiler tarafından erişilebildiğinden emin olun.

IBM MQ MQI client için anahtar veritabanı dosyasının yerini iki şekilde belirtebilirsiniz:

- MQSSLKEYR ortam değişkeni ayarlanıyor. Örneğin, AIX and Linuxüzerinde:

  V 9.3.0 V 9.3.0
  ```
  export MQSSLKEYR=/var/mqm/ssl/key.kdb
  ```

  Windows'ta:

  V 9.3.0 V 9.3.0
  ```
  set MQSSLKEYR=C:\Program Files\IBM\MQ\ssl\key.kdb
  ```

- Bir uygulama MQCONNX çağrısı yaptığında MQSCO yapısının *KeyRepository* alanında anahtar veritabanı dosyasının yol ve kök adını sağlar. MQCONNX içinde MQSCO yapısının kullanılmasıyla ilgili ek bilgi için MQSCO için genel bakışbaşlıklı konuya bakın.

V 9.3.0 V 9.3.0 AIX, Linux, and Windows üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması

Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla güvenli hale getirilmiştir. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

IBM MQ , anahtar havuzu parolasını bir IBM MQ MQI client' e sağlamak için dört mekanizma sağlar:

- "MQSCO ' nun KeyRepoPassword alanları " sayfa 300
- "MQKEYRPWD ortam değişkeni" sayfa 300
- "İstemci yapılandırma dosyasının SSLKeyRepositoryPassword özniteliği" sayfa 300
- "Anahtar havuzu saklama dosyası" sayfa 301

Bir anahtar havuzu parola saklama dosyası kullanmıyorsanız, anahtar havuzu parolasını düz metin dizgisi olarak ya da IBM MQ parola koruma sistemi kullanılarak şifrelenen bir dizgi olarak sağlayabilirsiniz. Anahtar havuzu parolasını koruma yöntemleri hakkında daha fazla bilgi için bkz. "AIX, Linux, and Windows üzerinde anahtar havuzu parolalarını şifreleme" sayfa 293.

MQSCO ' nun KeyRepoPassword alanları

MQSCO yapısını kullanarak bir anahtar havuzu parolası sağlamak için aşağıdaki üç değişken dizgi alanının bir birleşimini kullanmanız gerekir:

KeyRepoPasswordLength
Parolanın uzunluğu.

KeyRepoPasswordPtr
Parolayı içeren bellekteki konuma ilişkin gösterge.

KeyRepoPasswordOffset
Bellekteki parolanın yeri; MQSCO yapısının başlangıcından itibaren byte sayısı olarak gösterilir.

Not: KeyRepoPasswordPtr ya da **KeyRepoPasswordOffset**ürünlerinden yalnızca birini sağlayabilirsiniz.

Örneğin:

```
char * pwd = "passw0rd";
MQSCO  SslConnOptions = {MQSCO_DEFAULT};

SslConnOptions.KeyRepoPasswordPtr = pwd;
SslConnOptions.KeyRepoPasswordLength = (MQLONG)strlen(SslConnOptions.KeyRepoPasswordPtr);
SslConnOptions.Version = MQSCO_VERSION_6;
```

Uyarı: Bu yöntemi kullanarak parolayı sağlarsanız, parolayı IBM MQ client uygulamasına sağlanmadan önce şifreleyin. Daha fazla bilgi için bkz “Anahtar havuzu parolasını şifreleme” sayfa 301.

MQCSO yapısıyla ilgili daha fazla bilgi için bkz. MQSCO-SSL/TLS yapılandırma seçenekleri.

MQKEYRPWD ortam değişkeni

MQSCO yapısı kullanılarak istemciye anahtar havuzu parolası sağlanmazsa, *MQKEYRPWD* ortam değişkenini kullanarak anahtar havuzu parolasını belirtebilirsiniz. Örneğin:

```
export MQKEYRPWD=passw0rd
```

veya

```
set MQKEYRPWD=passw0rd
```

Burada `passw0rd` , parolanızdır.

Uyarı: Parolayı bu yöntemi kullanarak girdiyseniz, ortam değişkeninin değerini ayarlamadan önce parolayı şifreleyin. Daha fazla bilgi için bkz “Anahtar havuzu parolasını şifreleme” sayfa 301.

İstemci yapılandırma dosyasının SSLKeyRepositoryPassword özniteliği

İstemciye diğer yöntemlerden birini kullanarak bir anahtar havuzu parolası sağlanmazsa, istemci yapılandırma dosyasının **SSL** kısmına ilişkin **SSLKeyRepositoryPassword** özniteliğini kullanarak anahtar havuzu parolasını belirtebilirsiniz. Örneğin:

```
SSL:
    SSLKeyRepositoryPassword=passw0rd
```

Uyarı: Parolayı bu yöntemi kullanarak girdiyseniz, **SSLKeyRepositoryPassword** özniteliğinin değerini ayarlamadan önce parolayı şifreleyin. Daha fazla bilgi için bkz “Anahtar havuzu parolasını şifreleme” sayfa 301.

İstemci yapılandırma dosyasının SSL kısmı hakkında daha fazla bilgi için bkz. İstemci yapılandırma dosyasının SSL kısmı.

Anahtar havuzu saklama dosyası

Anahtar havuzu parolası diğer yöntemlerden biri kullanılarak istemciye sağlanmazsa, IBM MQ anahtar havuzuyla aynı dizinde bir parola saklama dosyası olduğunu varsayar. Saklama dosyası, anahtar havuzuyla aynı kök ada sahip, ancak `.sth` uzantısına sahip.

Anahtar havuzu saklama dosyası, anahtar havuzuyla aynı anda ya da daha sonra ayrı bir **runmqakm** komutu kullanılarak yaratılır.

Uyarı: Şifreleme dosyasının biçimi IBM MQ şifreleme sağlayıcısına IBM Global Security Kit (GSKit)özgüdür ve farklı bir şifreleme sağlayıcısı kullanan platformlarda kullanılamaz.

Anahtar havuzu yaratıldığında bir saklama dosyası yaratmak için **-stash** değiştirgesini belirtin. Örneğin:

```
runmqakm -keydb -create -db key.kdb -pw passw0rd -stash
```

Burada *passw0rd* , anahtar havuzu parolasıdır.

Daha sonra bir saklama dosyası yaratmak için aşağıdaki komutu çalıştırın:

```
runmqakm -keydb -stashpw -db key.kdb -pw passw0rd
```

Burada *passw0rd* , anahtar havuzu parolasıdır.

Anahtar havuzu parolasını şifreleme

Anahtar havuzu parolasını parola saklama dosyası dışında bir yöntem kullanarak sağlıyorsanız, IBM MQ parola koruma sistemini kullanarak parolayı şifreleyin. Parolayı şifrelemek için **runmqicred** komutunu çalıştırın. İstendiğinde anahtar havuzu parolasını girin. Komut şifrelenmiş parolayı çıkarır. Şifrelenmiş parola, açıklanan yöntemlerden herhangi biri kullanılarak düz metin parolası yerine IBM MQ MQI client ' e sağlanabilir.

Parolayı şifrelemek için ilk anahtar olarak bilinen bir şifreleme anahtarı kullanılır. Parolayı şifrelediğinizde, parolayı güvenli bir şekilde korumak için benzersiz bir başlangıç anahtarı kullanın. Kendi ilk anahtarınızı sağlamak için **runmqicred** komutuna ilişkin **-sf** parametresini kullanın. Başlangıç anahtarı belirtmezseniz, varsayılan anahtar kullanılır.

Daha fazla bilgi için bkz. runmqicred (IBM MQ istemci parolalarını koruma).

Anahtar havuzu parolası şifrelendiğinde kendi ilk anahtarınızı sağlarsanız ve IBM MQ MQI clientiçin şifrelenmiş parolayı sağlarsanız, IBM MQ MQI clientiçin aynı ilk anahtarı sağladığınızdan da emin olmanız gerekir. IBM MQ MQI clientiçin ilk anahtarın nasıl sağlanacağıyla ilgili daha fazla bilgi için bkz. “AIX, Linux, and Windows üzerinde bir IBM MQ MQI client için başlangıç anahtarı sağlanması” sayfa 301.

İlgili kavramlar

“AIX, Linux, and Windows üzerinde anahtar havuzu parolalarını şifreleme” sayfa 293
Bazı IBM MQ bileşenlerinin sayısal sertifikaları ya da simetrik anahtarları içeren bir anahtar havuzuna erişmesi gerekir. Bir anahtar havuzu, hassas bilgiler içerdiği için bir parolayla korunmuştur. Anahtar havuzu parolası, anahtar havuzuna erişildiğinde IBM MQ ' un okuyabileceği bir konumda saklanmalıdır. Anahtar havuzuna yetkisiz erişim olasılığını azaltmak için parola da şifrelenmelidir.

“AIX, Linux, and Windows üzerinde bir kuyruk yöneticisi için anahtar havuzu parolası sağlanması” sayfa 297
Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla güvenli hale getirilmiştir. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

AIX, Linux, and Windows üzerinde bir IBM MQ MQI client için başlangıç anahtarı sağlanması

IBM MQ Parola Koruma Sistemi kullanılarak şifrelenmiş bir IBM MQ MQI client ' e değişkenler sağlarsanız, değeri şifrelemek için kullanılan ilgili başlangıç anahtarını sağlamanız gerekebilir.

Değeri şifrelerken bir başlangıç anahtarı belirtmediyseniz, IBM MQ clientiçin herhangi bir başlangıç anahtarı değeri sağlamanız gerekmez. Ancak, benzersiz bir başlangıç anahtarı kullandıysanız, IBM MQ client için ilk anahtarı aşağıdaki yöntemleri kullanarak sağlayabilirsiniz:

- "MQCSP yapısını kullanarak ilk anahtarın sağlanmasını" sayfa 302
- "MQS_MQI_KEYFILE ortam değişkenini kullanarak ilk anahtar belirtiliyordu" sayfa 302
- "İstemci yapılandırma dosyasını kullanarak ilk anahtarın sağlanmasına" sayfa 302

MQCSP yapısını kullanarak ilk anahtarın sağlanmasını

MQCSP yapısını kullanarak ilk anahtarı sağlamak için aşağıdaki üç değişken dizgi alanının bir birleşimini kullanmanız gerekir:

InitialKeyLength
İlk anahtarın uzunluğu

InitialKeyPtr
İlk anahtarı içeren bellekteki konuma ilişkin gösterge

InitialKeyOffset
MQCSP yapısının başlangıcından itibaren byte sayısı olarak gösterilen, bellekteki ilk anahtarın yeri.

Not: InitialKeyPtr ya da **InitialKeyOffset**ürünlerinden yalnızca birini sağlayabilirsiniz.

Örneğin:

```
char * initialKey = "myInitialKey";
MQCSP  cspOptions = {MQCSP_DEFAULT};


cspOptions.InitialKeyPtr = initialKey;
cspOptions.InitialKeyLength = (MQLONG)strlen(cspOptions.InitialKeyPtr);
cspOptions.Version = MQCSP_VERSION_2;
```

MQS_MQI_KEYFILE ortam değişkenini kullanarak ilk anahtar belirtiliyordu

MQCSP yapısı kullanılarak istemciye ilk anahtar sağlanmazsa, IBM MQ *MQS_MQI_KEYFILE* ortam değişkenini denetler. Bu ortam değişkenini, kullanmak istediğiniz ilk anahtardan oluşan tek satırlık metni içeren bir dosyanın konumuna ayarlamanız gerekir.

Örneğin, kök dizinde `mykey.key` adlı bir dosya varsa ve ilk anahtarı içeriyorsa, ortam değişkenini aşağıdaki gibi ayarlamanız gerekir:

```
export MQS_MQI_KEYFILE=/mykey.key
```

veya

```
set MQS_MQI_KEYFILE=C:\mykey.key
```

İstemci yapılandırma dosyasını kullanarak ilk anahtarın sağlanmasına

İstemciye önceki bir düzenek kullanılarak bir başlangıç anahtarı sağlanmazsa, IBM MQ `mqclient.ini` dosyasının Güvenlik kısmına ilişkin **MQIInitialKeyFile** özniteliğini denetler. Bu özniteliği, kullanmak istediğiniz ilk anahtardan oluşan tek bir metin satırı içeren bir dosyanın konumuna ayarlamanız gerekir.

Örneğin, kök dizinde `mykey.key` adlı bir dosya varsa ve ilk anahtarı içeriyorsa, istemci yapılandırma dosyası aşağıdakileri içermelidir:

```
Security:
    MQIInitialKeyFile=/mykey.key
```

İlgili kavramlar

“AIX, Linux, and Windows üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması” sayfa 299

Anahtar havuzu hassas bilgiler içerdiğinden, bir parolayla güvenli hale getirilmiştir. TLS işlemlerini gerçekleştirmek üzere anahtar havuzu içeriğine erişebilmek için IBM MQ anahtar havuzu parolasını almalıdır.

“SSL/TLS ile çalışma” sayfa 271

Bu konularda, TLS ' nin IBM MQile kullanılmasına ilişkin tek görevlerin gerçekleştirilmesine ilişkin yönergeler yer alır.

ALW *Sertifikalarda ya da sertifika deposunda yapılan değişiklikler AIX, Linux, and Windows üzerinde yürürlüğe girdiğinde*

Bir sertifika deposundaki sertifikaları ya da sertifika deposunun konumunu değiştirdiğinizde, değişiklikler kanalın tipine ve kanalın çalışma şekline bağlı olarak yürürlüğe girer.

Anahtar veritabanı dosyasındaki sertifikalarda ve anahtar havuzu özniteliğinde yapılan değişiklikler aşağıdaki durumlarda etkili olur:

- Yeni bir giden tek kanal işlemi ilk olarak bir TLS kanalı çalıştırdığında.
- Yeni bir gelen TCP/IP tek kanal işlemi ilk olarak TLS kanalı başlatma isteği aldığında.
- TLS ortamını yenilemek için MQSC komutu REFRESH SECURITY TYPE (SSL) verildiğinde.
- İstemci uygulaması işlemleri için, süreçteki son TLS bağlantısı kapatıldığında. Sonraki TLS bağlantısı, sertifika değişikliklerini alacak.
- Bir süreç havuzlama işleminin (amqrmppa) iş parçacıkları olarak çalışan kanallar için, süreç havuzlama işlemi başlatıldığında ya da yeniden başlatıldığında ve önce bir TLS kanalı çalıştırıldığında. Süreç havuzlama işlemi zaten bir TLS kanalı çalıştırdıysa ve değişikliğin hemen yürürlüğe girmesini istiyorsanız, MQSC komutunu REFRESH SECURITY TYPE (SSL) çalıştırın.
- Kanal başlatıcısının iş parçacıkları olarak çalışan kanallar için, kanal başlatıcı başlatıldığında ya da yeniden başlatıldığında ve ilk olarak bir TLS kanalı çalıştığında. Kanal başlatıcı işlemi zaten bir TLS kanalı çalıştırdıysa ve değişikliğin hemen yürürlüğe girmesini istiyorsanız, MQSC komutunu REFRESH SECURITY TYPE (SSL) çalıştırın.
- Bir TCP/IP dinleyicisinin iş parçacıkları olarak çalışan kanallar için, dinleyici başlatıldığında ya da yeniden başlatıldığında ve ilk olarak TLS kanalı başlatma isteği aldığında. Dinleyici zaten bir TLS kanalı çalıştırdıysa ve değişikliğin hemen yürürlüğe girmesini istiyorsanız, MQSC komutunu REFRESH SECURITY TYPE (SSL) çalıştırın.

IBM MQ TLS ortamını IBM MQ Explorer ya da PCF komutlarını kullanarak da yenileyebilirsiniz.

Önemli: . Anahtar deposu yapılanış dosyasında ve/ya da AMS MCA kesicisi (ve olağan istemcide AMS) tarafından kullanılan anahtar deposunda yapılan değişiklikler bir kuyruk yöneticisi ya da uygulama yeniden başlatmada toplanır.

ALW *AIX, Linux, and Windows üzerinde kendinden onaylı kişisel sertifika oluşturma*

strmqikm (iKeyman) öğesini kullanarak kendinden onaylı bir sertifika yaratabilirsiniz. GUI ya da **runmqckm** (iKeycmd) ya da **runmqakm** (GSKCapiCmd) komutunu kullanarak komut satırından.

Not: IBM MQ , SHA-3 ya da SHA-5 algoritmalarını desteklemez. Her iki algoritma da SHA-2 ailesinin üyesi olduğundan, SHA384WithRSA ve SHA512WithRSA dijital imza algoritması adlarını kullanabilirsiniz.

Deprecated Sayısal imza algoritması adları SHA3WithRSA ve SHA5WithRSA , sırasıyla SHA384WithRSA ve SHA512WithRSA kısaltması oldukları için kullanımdan kaldırılmıştır.

Kendinden onaylı sertifikaları neden kullanmak isteyebileceğinize ilişkin ek bilgi için İki kuyruk yöneticisinin karşılıklı kimlik doğrulaması için kendinden onaylı sertifikaları kullanmabaşlıklı konuya bakın.

Tüm dijital sertifikalar CipherSpecsile birlikte kullanılamaz. Kullanmanız gereken CipherSpecs ile uyumlu bir sertifika yarattığınızdan emin olun. IBM MQ , üç farklı CipherSpectipini destekler. Ayrıntılar için “IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu” sayfa 46 başlıklı konudaki “Eliptik Eğri ve RSA CipherSpecs Birlikte Çalışabilirlik” sayfa 47 konusuna bakın.

Tip 1 CipherSpecs ' i (ECDHE_ECDSA_ile başlayan adları olanlar) kullanmak için sertifikayı yaratmak üzere **runmqakm** komutunu kullanmanız ve bir Eliptik Eğri ECDSA imza algoritması parametresi belirtmeniz gerekir; örneğin, **-sig_alg** EC_ecdsa_with_SHA384.

-sig_alg hash algoritması ile kullanılabilen seçeneklerin bir listesi için bkz. “AIX, Linux, and Windows üzerinde runmqckm ve runmqakm seçenekleri” sayfa 565 .

Aşağıdaki işlemleri kullanıyorsanız:

- GUI, bkz. “strmqikm kullanıcı arabiriminin kullanılması” sayfa 304
- Komut satırı, bkz. “Komut satırının kullanılması” sayfa 305

ALW ***strmqikm*** *kullanıcı arabiriminin kullanılması*

strmqikm (iKeyman) kullanarak kişisel sertifika yaratabilirsiniz. Grafik kullanıcı arabirimi.

Bu görev hakkında

strmqikm , FIPS uyumlu bir seçenek sağlamaz. TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın.

Yordam

Grafik kullanıcı arabirimini kullanarak kuyruk yöneticiniz ya da IBM MQ MQI client için kişisel sertifika yaratmak üzere aşağıdaki adımları tamamlayın:

1. **strmqikm** komutunu kullanarak GUI ' yi başlatın.
2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın.

   Open (Aç) penceresi görüntülenir.
3. **Anahtar veritabanı tipi** seçeneğini tıklatın ve **CMS** (Sertifika Yönetimi Sistemi) seçeneğini belirleyin.
4. Anahtar veri tabanı dosyalarını içeren dizine gitmek için **Göz At** düğmesini tıklatın.
5. İsteği oluşturmak istediğiniz anahtar veritabanı dosyasını seçin; örneğin, key . kdb.
6. **Tamam**'ı tıklatın.

   Parola Bilgi İstemi penceresi açılır.
7. Anahtar veritabanını yaratırken ayarladığınız parolayı yazın ve **Tamam**düğmesini tıklatın.

   Anahtar veritabanı dosyanızın adı **Dosya Adı** alanında gösterilir.
8. **Create** (Oluştur) menüsünden **New Self-Signed Certificate**(Kendinden Onaylı Yeni Sertifika) seçeneğini tıklatın. Yeni Kendinden Onaylı Sertifika Yarat penceresi görüntülenir.
9. **Anahtar Etiketi** alanına sertifika etiketini girin.

   Etiket, **CERTLABL** özniteliğinin değeri (ayarlandıysa) ya da kuyruk yöneticisinin adı ya da IBM MQ MQI client oturum açma kullanıcı kimliği eklenmiş olarak varsayılan ibmwebspheremq değeridir (tümü küçük harfli olarak). Ayrıntılar için bkz. Dijital sertifika etiketleri .
10. **Ayırt edici ad** alanında ya da **Konu diğer adı** alanlarından herhangi biri için bir değer yazın ya da seçin.
11. Kalan alanlar için varsayılan değerleri kabul edin ya da yeni değerler yazın ya da seçin.

    Ayırt Edici Adlar hakkında daha fazla bilgi için bkz. “Ayırt Edici Adlar” sayfa 14.
12. **Tamam**'ı tıklatın.

    Kişisel Sertifikalar listesi, oluşturduğunuz kendinden onaylı kişisel sertifikanın etiketini gösterir.

ALW *Komut satırının kullanılması*

runmqckm (iKeycmd) ya da **runmqakm** (GSKCapiCmd) komutlarını kullanarak komut satırından kişisel sertifika yaratabilirsiniz. SSL ya da TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın.

Yordam

runmqckm ya da **runmqakm** (GSKCapiCmd) komutunu kullanarak kendinden imzalı bir kişisel sertifika oluşturun.

- **runmqckm**komutunu kullanarak:

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

  -dn *distinguished_name*yerine -san_dnsname *DNS_names*, -san_emailaddr *email_addresses*ya da -san_ipaddr *IP_addresses*kullanabilirsiniz.

- **runmqakm**komutunu kullanarak:

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

Burada:

-db *kütükadı*
CMS anahtar veritabanının tam olarak nitelenmiş dosya adını belirtir.

-pw *parola*
CMS anahtar veri tabanına ilişkin parolayı belirtir.

-label *etiket*
Sertifikaya eklenen anahtar etiketini belirtir. Etiket, ayarlanmışsa, **CERTLABL** özniteliğinin değeri ya da kuyruk yöneticisinin adı ya da sonuna IBM MQ MQI client oturum açma kullanıcı kimliği eklenmiş olarak varsayılan ibmwebspheremq değeridir. Ayrıntılar için bkz. "Dijital sertifika etiketleri, gereksinimlerin anlaşılması" sayfa 26.

-dn *ayırt edici ad_adı*
Çift tırnak içine alınmış X.500 ayırt edici adını belirtir. En az bir öznitelik gereklidir. Birden çok kuruluş birimi ve DC özniteliği sağlayabilirsiniz.

Not: runmqckm ve **runmqakm** araçları, posta kodu özniteliğini PColarak değil, POSTALCODEolarak ifade eder. Posta koduyla sertifika istemek için her zaman bu sertifika yönetimi komutlarını kullandığınızda **-dn** parametresinde POSTALCODE değerini belirleyin.

-size *anahtar_büyüklüğü*
Anahtar boyutunu belirtir. **runmqckm**kullanıyorsanız, değer 512 ya da 1024olabilir. **runmqakm**kullanıyorsanız, değer 512, 1024ya da 2048olabilir.

x509version *sürüm*
Yaratılacak X.509 sertifikasının sürümü. Değer 1, 2 ya da 3 olabilir. Varsayılan 3'tür.

-file *kütükadı*
Sertifika isteğine ilişkin dosya adını belirtir.

-süre bitimi *gün*
Sertifikanın son kullanma tarihi (gün). Varsayılan değer, bir sertifika için 365 gündür.

-fips
Komutun FIPS kipinde çalıştırılacağını belirtir. Yalnızca FIPS IBM Crypto for C (ICC) bileşeni kullanılır ve bu bileşen FIPS kipinde başarıyla başlatılmalıdır. FIPS kipindeyken, ICC bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.

-sig_alg
runmqckmiçin, girdinin anahtar çiftinin oluşturulması için kullanılan asimetrik imza algoritmasını belirtir. Değer MD2_WITH_RSA, MD2WithRSA, MD5_WITH_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, SHA2/ECDSA, SHA224WithECDSA, SHA256_WITH_RSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithECDSA, SHA3/ECDSA, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, SHA3WithECDSA, SHA5/ECDSA, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHA5WithECDSA, SHA_WITH_DSA, SHA_WITH_RSA, SHAWithDSA, SHAWithRSA. Varsayılan değer SHA1WithRSAdeğeridir.

-sig_alg
runmqakmiçin, bir sertifika isteği oluşturulurken kullanılan hash algoritmasını belirtir. Bu hash algoritması, yeni yaratılan sertifika isteğiyle ilişkili imzayı yaratmak için kullanılır. Değer 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_SHA384ya da EC_ecdsa_with_SHA512. Varsayılan değer SHA1WithRSAdeğeridir.

-san_dnsname *DNS_names*
Yaratılmakta olan girdiye ilişkin DNS adlarının virgülle ayrılmış ya da boşlukla ayrılmış bir listesini belirtir.

-san_emailaddr *eposta_adresleri*
Yaratılmakta olan girdiye ilişkin e-posta adreslerinin virgülle ayrılmış ya da boşlukla ayrılmış bir listesini belirtir.

-san_ipaddr *IP_addresses* (IP_adresleri)
Yaratılmakta olan girdiye ilişkin IP adreslerinin virgülle ayrılmış ya da boşlukla ayrılmış bir listesini belirtir.

ALW *AIX, Linux, and Windows üzerinde kişisel sertifika isteme*

strmqikm (iKeyman) komutunu kullanarak kişisel sertifika isteyebilirsiniz. GUI ya da **runmqckm** (iKeycmd) ya da **runmqakm** (GSKCapiCmd) komutlarını kullanarak komut satırından. SSL ya da TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın.

Bu görev hakkında

strmqikm GUI 'sini kullanarak ya da komut satırından aşağıdaki noktalara bağlı olarak kişisel sertifika isteyebilirsiniz:

- IBM MQ , SHA-3 ya da SHA-5 algoritmalarını desteklemez. Her iki algoritma da SHA-2 ailesinin üyesi olduğundan, SHA384WithRSA ve SHA512WithRSA dijital imza algoritması adlarını kullanabilirsiniz.
- Deprecated Sayısal imza algoritması adları SHA3WithRSA ve SHA5WithRSA , sırasıyla SHA384WithRSA ve SHA512WithRSA kısaltması oldukları için kullanımdan kaldırılmıştır.
- Tüm dijital sertifikalar CipherSpecsile birlikte kullanılamaz. Kullanmanız gereken CipherSpecs ile uyumlu bir sertifika isteğinde bulunduğunuzdan emin olun. IBM MQ , üç farklı CipherSpectipini destekler. Ayrıntılar için “IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu” sayfa 46 başlıklı konudaki “Eliptik Eğri ve RSA CipherSpecs Birlikte Çalışabilirlik” sayfa 47 konusuna bakın.
- Tip 1 CipherSpecs ' i (adları ECDHE_ECDSA_ile başlayan) kullanmak için sertifikayı istemek için **runmqakm** komutunu kullanmanız ve bir Eliptik Eğri ECDSA imza algoritması parametresi belirtmeniz gerekir; örneğin, **-sig_alg** EC_ecdsa_with_SHA384.

  -sig_alg hash algoritması ile kullanılabilen seçeneklerin bir listesi için bkz. “AIX, Linux, and Windows üzerinde runmqckm ve runmqakm seçenekleri” sayfa 565 .
- Yalnızca **runmqakm** komutu FIPS uyumlu bir seçenek sağlar.
- Şifreleme donanımı kullanıyorsanız, bkz. “PKCS #11 donanımınız için kişisel sertifika isteme” sayfa 325.

Aşağıdaki işlemleri kullanıyorsanız:

- GUI, bkz. “strmqikm kullanıcı arabiriminin kullanılması” sayfa 307
- Komut satırı, bkz. “Komut satırının kullanılması” sayfa 307

ALW ***strmqikm*** *kullanıcı arabiriminin kullanılması*

strmqikm (iKeyman) komutunu kullanarak kişisel sertifika isteyebilirsiniz. Grafik kullanıcı arabirimi. SSL ya da TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın.

Bu görev hakkında

strmqikm , FIPS uyumlu bir seçenek sağlamaz. TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın.

Yordam

iKeyman kullanıcı arabirimini kullanarak kişisel sertifika için başvuru yapmak üzere aşağıdaki adımları tamamlayın:

1. **strmqikm** komutunu kullanarak kullanıcı arabirimini başlatın.
2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın.

   Aç penceresi açılır.
3. **Anahtar veritabanı tipi** seçeneğini tıklatın ve **CMS** (Sertifika Yönetimi Sistemi) seçeneğini belirleyin.
4. Anahtar veri tabanı dosyalarını içeren dizine gitmek için **Göz At** düğmesini tıklatın.
5. İsteği oluşturmak istediğiniz anahtar veritabanı dosyasını seçin; örneğin, `key.kdb`.
6. **Aç**'ı tıklatın.

   Parola Bilgi İstemi penceresi açılır.
7. Anahtar veritabanını yaratırken ayarladığınız parolayı yazın ve **Tamam**düğmesini tıklatın.

   Anahtar veritabanı dosyanızın adı **Dosya Adı** alanında gösterilir.
8. **Oluştur** menüsünden **Yeni Sertifika İsteği**seçeneğini tıklatın. **Yeni Anahtar ve Sertifika İsteği Oluştur** penceresi açılır.
9. **Anahtar Etiketi** alanına sertifika etiketini girin.

   Etiket, **CERTLABL** özniteliğinin değeri (ayarlandıysa) ya da kuyruk yöneticisinin adı ya da IBM MQ MQI client oturum açma kullanıcı kimliği eklenmiş olarak varsayılan `ibmwebspheremq` değeridir (tümü küçük harfli olarak). Ayrıntılar için bkz. Dijital sertifika etiketleri .
10. **Ayırt edici ad** alanında ya da **Konu diğer adı** alanlarından herhangi biri için bir değer yazın ya da seçin. Kalan alanlar için varsayılan değerleri kabul edin ya da yeni değerler yazın ya da seçin.

    Ayırt Edici Adlar hakkında daha fazla bilgi için bkz. “Ayırt Edici Adlar” sayfa 14.
11. **Sertifika isteğinin saklanacağı dosyanın adını girin** alanına varsayılan değeri kabul edin `certreq.arm`ya da tam yol ile yeni bir değer yazın.
12. **Tamam**'ı tıklatın.

    Bir doğrulama penceresi görüntülenir.
13. **Tamam**'ı tıklatın.

    Kişisel Sertifika İstekleri listesi, yarattığınız yeni kişisel sertifika isteğinin etiketini gösterir. Sertifika isteği, “11” sayfa 307. adımda seçtiğiniz dosyada saklanır.
14. Dosyayı bir sertifika yetkilisine (CA) göndererek ya da dosyayı CA web sitesindeki istek formuna kopyalayarak yeni kişisel sertifikayı isteyin.

ALW *Komut satırının kullanılması*

runmqckm (iKeycmd) ya da **runmqakm** (GSKCapiCmd) komutlarını kullanarak komut satırından kişisel sertifika isteyebilirsiniz. SSL ya da TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın.

Yordam

runmqckm ya da **runmqakm** (GSKCapiCmd) komutunu kullanarak kişisel sertifika isteyin.

- **runmqckm**komutunu kullanarak:

```
runmqckm -certreq -create -db filename -pw
password -label label
         -dn distinguished_name -size key_size
        -file filename -sig_alg algorithm
```

  -dn *distinguished_name*yerine -san_dsname *DNS_names*, -san_emailaddr *email_addresses*ya da -san_ipaddr *IP_addresses*kullanabilirsiniz.

- **runmqakm**komutunu kullanarak:

```
runmqakm -certreq -create -db filename -pw
password -label label
         -dn distinguished_name -size key_size
         -file filename -fips -sig_alg algorithm
```

Burada:

-db *kütükadı*
CMS anahtar veritabanının tam olarak nitelenmiş dosya adını belirtir.

-pw *parola*
CMS anahtar veri tabanına ilişkin parolayı belirtir.

-label *etiket*
Sertifikaya eklenen anahtar etiketini belirtir. Etiket, ayarlanmışsa, **CERTLABL** özniteliğinin değeri ya da kuyruk yöneticisinin adı ya da sonuna IBM MQ MQI client oturum açma kullanıcı kimliği eklenmiş olarak varsayılan ibmwebspheremq değeridir. Ayrıntılar için bkz. “Dijital sertifika etiketleri, gereksinimlerin anlaşılması” sayfa 26.

-dn *ayırt edici ad_adı*
Çift tırnak içine alınmış X.500 ayırt edici adını belirtir. En az bir öznitelik gereklidir. Birden çok kuruluş birimi ve DC özniteliği sağlayabilirsiniz.

Not: runmqckm ve **runmqakm** araçları, posta kodu özniteliğini PColarak değil, POSTALCODEolarak ifade eder. Posta koduyla sertifika istemek için her zaman bu sertifika yönetimi komutlarını kullandığınızda **-dn** parametresinde POSTALCODE değerini belirleyin.

-size *anahtar_büyüklüğü*
Anahtar boyutunu belirtir. **runmqckm**kullanıyorsanız, değer 512 ya da 1024olabilir. **runmqakm**kullanıyorsanız, değer 512, 1024ya da 2048olabilir.

-file *kütükadı*
Sertifika isteğine ilişkin dosya adını belirtir.

-fips
Komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.

-sig_alg
runmqckmiçin, girdinin anahtar çiftinin oluşturulması için kullanılan asimetrik imza algoritmasını belirtir. Değer MD2_WITH_RSA, MD2WithRSA, MD5_WITH_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, SHA2/ECDSA, SHA224WithECDSA, SHA256_WITH_RSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithECDSA, SHA3/ECDSA, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, SHA3WithECDSA, SHA5/ECDSA, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHA5WithECDSA, SHA_WITH_DSA, SHA_WITH_RSA, SHAWithDSA, SHAWithRSA. Varsayılan değer SHA1WithRSAdeğeridir.

-sig_alg
runmqakmiçin, bir sertifika isteği oluşturulurken kullanılan hash algoritmasını belirtir. Bu hash algoritması, yeni yaratılan sertifika isteğiyle ilişkili imzayı yaratmak için kullanılır. Değer 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_SHA384ya da EC_ecdsa_with_SHA512. Varsayılan değer SHA1WithRSAdeğeridir.

-san_dnsname *DNS_names*
Yaratılmakta olan girdiye ilişkin DNS adlarının virgülle ayrılmış ya da boşlukla ayrılmış bir listesini belirtir.

-san_emailaddr *eposta_adresleri*
Yaratılmakta olan girdiye ilişkin e-posta adreslerinin virgülle ayrılmış ya da boşlukla ayrılmış bir listesini belirtir.

-san_ipaddr *IP_addresses* (IP_adresleri)
Yaratılmakta olan girdiye ilişkin IP adreslerinin virgülle ayrılmış ya da boşlukla ayrılmış bir listesini belirtir.

Sonraki adım

Sertifika yetkilisine bir sertifika isteği gönderin. Daha fazla bilgi için bkz. “AIX, Linux, and Windows üzerindeki bir anahtar havuzuna kişisel sertifikaları alma” sayfa 310 .

ALW *AIX, Linux, and Windows üzerinde var olan bir kişisel sertifikayı yenileme*

strmqikm (iKeyman) kullanarak kişisel sertifikayı yenileyebilirsiniz. GUI ya da **runmqckm** (iKeycmd) ya da **runmqakm** (GSKCapiCmd) komutlarını kullanarak komut satırından.

Bu görev hakkında

Kişisel sertifikalarınız için daha büyük anahtar boyutları kullanma gereksiniminiz varsa, var olan bir sertifikayı yenileyemezsiniz. Gerek duyduğunuz anahtar boyutlarını kullanan yeni bir sertifika isteği oluşturmak için “AIX, Linux, and Windows üzerinde kişisel sertifika isteme” sayfa 306 içinde açıklanan adımları izleyerek var olan anahtarınızı değiştirmeniz gerekir.

Kişisel sertifikanın süre bitim tarihi vardır, bu tarihten sonra sertifika kullanılamaz. Bu görev, süresi dolmadan önce var olan bir kişisel sertifikanın nasıl yenileneceğini açıklar.

strmqikm *kullanıcı arabiriminin kullanılması*

Bu görev hakkında

strmqikm , FIPS uyumlu bir seçenek sağlamaz. TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın.

Yordam

strmqikm kullanıcı arabirimini kullanarak kişisel sertifika için başvuru yapmak üzere aşağıdaki adımları tamamlayın:

1. AIX, Linux, and Windowsüzerinde **strmqikm** komutunu kullanarak kullanıcı arabirimini başlatın.
2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın.
   Aç penceresi açılır.
3. **Anahtar veritabanı tipi** seçeneğini tıklatın ve **CMS** (Sertifika Yönetimi Sistemi) seçeneğini belirleyin.
4. Anahtar veri tabanı dosyalarını içeren dizine gitmek için **Göz At** düğmesini tıklatın.

5. İsteği oluşturmak istediğiniz anahtar veritabanı dosyasını seçin; örneğin, `key.kdb`.
6. **Aç**'ı tıklatın.
   Parola Bilgi İstemi penceresi açılır.
7. Anahtar veritabanını yaratırken ayarladığınız parolayı yazın ve **Tamam**düğmesini tıklatın.
   Anahtar veritabanı dosyanızın adı **Dosya Adı** alanında gösterilir.
8. Açılan seçim menüsünden **Kişisel Sertifikalar** seçeneğini belirleyin ve yenilemek istediğiniz sertifikayı listeden seçin.
9. **İsteği Yeniden Oluştur ...** düğmesini tıklatın. düğmesini tıklatın.
   Dosya adını ve dosya konumu bilgilerini girebileceğiniz bir pencere açılır.
10. **Dosya adı** alanında, varsayılan `certreq.arm`değerini kabul edin ya da tam dosya yolu da içinde olmak üzere yeni bir değer yazın.
11. **Tamam**'ı tıklatın. Sertifika isteği, “9” sayfa 310. adımda seçtiğiniz dosyada saklanır.
12. Dosyayı bir sertifika yetkilisine (CA) göndererek ya da dosyayı CA web sitesindeki istek formuna kopyalayarak yeni kişisel sertifikayı isteyin.

Komut satırının kullanılması

Yordam

runmqckm ya da **runmqakm** komutunu kullanarak kişisel sertifika istemek için aşağıdaki komutları kullanın:

- **runmqckm**komutunu kullanarak:

```
runmqckm -certreq -recreate -db filename -pw
password -label label
-target filename
```

- Runmqakm kullanılıyor:

```
runmqakm -certreq -recreate -db filename -pw
password -label label
-target filename
```

Burada:

-db *kütükadı*
CMS anahtar veritabanının tam olarak nitelenmiş dosya adını belirtir.

-pw *parola*
CMS anahtar veri tabanına ilişkin parolayı belirtir.

-target *kütükadı*
Sertifika isteğine ilişkin dosya adını belirtir.

Not: Eski sertifika bilgileri önbellekte olduğundan, REFRESH SECURITY TYPE (SSL) komutunu çalıştırmanız gerekir.

Sonraki adım

Sertifika yetkilisinden imzalı kişisel sertifikayı aldıktan sonra, “AIX, Linux, and Windows üzerindeki bir anahtar havuzuna kişisel sertifikaları alma” sayfa 310içinde açıklanan adımları kullanarak sertifikayı anahtar veritabanınıza ekleyebilirsiniz.

ALW *AIX, Linux, and Windows üzerindeki bir anahtar havuzuna kişisel sertifikaları alma*

Anahtar veritabanı dosyasına kişisel sertifika almak için bu yordamı kullanın. Anahtar havuzu, sertifika isteğini yarattığınız havuzla aynı olmalıdır.

CA size yeni bir kişisel sertifika gönderdikten sonra, bu sertifikayı yeni sertifika isteğini oluşturacağınız anahtar veritabanı dosyasına eklersiniz. Sertifika yetkilisi sertifikayı e-posta iletisinin bir parçası olarak gönderirse, sertifikayı ayrı bir dosyaya kopyalayın.

Kullanılan `strmqikm`

TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın. **strmqikm** , FIPS uyumlu bir seçenek sağlamaz.

İçe aktarılacak sertifika dosyasının yürürlükteki kullanıcı için yazma izni olduğunu doğrulayın ve anahtar veritabanı dosyasına kişisel sertifika almak için bir kuyruk yöneticisi ya da IBM MQ MQI client için aşağıdaki yordamı kullanın:

1. **strmqikm** komutunu kullanarak GUI ' yi başlatın.
2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın. Aç penceresi açılır.
3. **Anahtar veritabanı tipi** seçeneğini tıklatın ve **CMS** (Sertifika Yönetimi Sistemi) seçeneğini belirleyin.
4. Anahtar veri tabanı dosyalarını içeren dizine gitmek için **Göz At** düğmesini tıklatın.
5. Sertifikayı eklemek istediğiniz anahtar veritabanı dosyasını seçin; örneğin, `key.kdb`.
6. **Aç**'ı ve ardından **Tamam**' ı tıklatın. Password Prompt (Parola İstemi) penceresi açılır.
7. Anahtar veritabanını yaratırken ayarladığınız parolayı yazın ve **Tamam**düğmesini tıklatın. Anahtar veritabanı dosyanızın adı **Dosya Adı** alanında görüntülenir. **Kişisel Sertifikalar** görünümünü seçin.
8. **Al**düğmesini tıklatın. Dosyadan Sertifika Al penceresi açılır.
9. Yeni kişisel sertifika için sertifika dosyası adını ve yerini yazın ya da adı ve yeri seçmek için **Göz At** düğmesini tıklatın.
10. **Tamam** düğmesini tıklatın. Anahtar veritabanınızda zaten kişisel bir sertifikanız varsa, eklemekte olduğunuz anahtarı veritabanına varsayılan anahtar olarak ayarlamak isteyip istemediğinizi soran bir pencere açılır.
11. **Evet** ya da **Hayır**' ı tıklatın. Enter a Label (Etiket Girin) penceresi açılır.
12. **Tamam** düğmesini tıklatın. **Kişisel Sertifikalar** alanı, eklediğiniz yeni kişisel sertifikanın etiketini gösterir.

Komut satırının kullanılması

Bir anahtar veritabanı dosyasına kişisel sertifika eklemek için aşağıdaki komutlardan birini kullanın:

- **runmqckm**komutunu kullanarak:

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

- **runmqakm**komutunu kullanarak:

```
runmqakm -cert -receive -file filename -db filename -pw password -fips
```

Burada:

-file *kütükadı*
Kişisel sertifikanın tam olarak nitelenmiş dosya adını belirtir.

-db *kütükadı*
CMS anahtar veritabanının tam olarak nitelenmiş dosya adını belirtir.

-pw *parola*
CMS anahtar veri tabanına ilişkin parolayı belirtir.

-format *ascii*
Sertifikanın biçimini belirtir. Değer, Base64-encoded ASCII için `ascii` ya da İkili DER verileri için `binary` olabilir. Varsayılan değer `ascii`' dir.

-fips
Komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.

Şifreleme donanımı kullanıyorsanız, bkz. "PKCS #11 donanımınıza kişisel sertifika alınması" sayfa 326.

ALW *AIX, Linux, and Windows üzerindeki bir anahtar havuzundan CA sertifikasının çekilmesi*

Bir CA sertifikasını çıkarmak için bu yordamı izleyin.

Kullanılan strmqikm

TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın. **strmqikm** (iKeyman), FIPS uyumlu bir seçenek sağlamaz.

Sertifika yetkilisi sertifikasını çıkarmak istediğiniz makinede aşağıdaki adımları gerçekleştirin:

1. **strmqikm** komutunu kullanarak GUI ' yi başlatın.
2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın. Aç penceresi açılır.
3. **Anahtar veritabanı tipi** seçeneğini tıklatın ve **CMS** (Sertifika Yönetimi Sistemi) seçeneğini belirleyin.
4. Anahtar veri tabanı dosyalarını içeren dizine gitmek için **Göz At** düğmesini tıklatın.
5. İçinden çıkarmak istediğiniz anahtar veritabanı dosyasını seçin; örneğin, key.kdb.
6. **Aç**' ı tıklatın. Password Prompt (Parola İstemi) penceresi açılır.
7. Anahtar veritabanını yaratırken ayarladığınız parolayı yazın ve **Tamam**düğmesini tıklatın. Anahtar veritabanı dosyanızın adı **Dosya Adı** alanında görüntülenir.
8. **Anahtar veritabanı içeriği** alanında **İmzalayan Sertifikaları** seçeneğini belirleyin ve ayıklamak istediğiniz sertifikayı seçin.
9. **Al**' ı tıklatın. Bir Sertifikayı Dosyaya Çıkar penceresi açılır.
10. Sertifikanın **Veri tipi** ' ni seçin; örneğin, .arm uzantılı bir dosya için **Base64-encoded ASCII verileri** .
11. Sertifikayı saklamak istediğiniz sertifika dosyası adını ve yerini yazın ya da adı ve yeri seçmek için **Göz At** düğmesini tıklatın.
12. **Tamam** düğmesini tıklatın. Sertifika belirttiğiniz dosyaya yazılır.

Komut satırının kullanılması

runmqckm komutunu ya da **runmqakm** komutunu kullanarak bir CA sertifikasını çıkarmak için aşağıdaki komutları kullanın:

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

veya

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

Burada:

-db *filename*	CMS anahtar veritabanının tam olarak nitelenmiş yol adıdır.
-pw *password*	CMS anahtar veritabanının parolasıdır.
-label *label*	sertifikaya iliştirilen etikettir.
-target *filename*	hedef dosyanın adıdır.

`-format ascii`	sertifikanın biçimidir. Değer, Base64-encoded ASCII için `ascii` ya da İkili DER verileri için `binary` olabilir. Varsayılan değer `ascii`' dir.
`-fips`	komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.

ALW *AIX, Linux, and Windows üzerindeki bir anahtar havuzundan kendinden onaylı sertifikanın genel kısmının çıkarılması*

Kendinden onaylı bir sertifikanın genel kısmını çıkarmak için bu yordamı izleyin.

Kullanılan strmqikm

TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın. **strmqikm** (iKeyman), FIPS uyumlu bir seçenek sağlamaz.

Kendinden onaylı bir sertifikanın genel kısmını çıkarmak istediğiniz makinede aşağıdaki adımları gerçekleştirin:

1. **strmqikm** komutunu kullanarak GUI ' yi başlatın.
2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın. Aç penceresi açılır.
3. **Anahtar veritabanı tipi** seçeneğini tıklatın ve **CMS** (Sertifika Yönetimi Sistemi) seçeneğini belirleyin.
4. Anahtar veri tabanı dosyalarını içeren dizine gitmek için **Göz At** düğmesini tıklatın.
5. Sertifikayı almak istediğiniz anahtar veritabanı dosyasını seçin; örneğin, `key.kdb`.
6. **Tamam** düğmesini tıklatın. Password Prompt (Parola İstemi) penceresi açılır.
7. Anahtar veritabanını yaratırken ayarladığınız parolayı yazın ve **Tamam**düğmesini tıklatın. Anahtar veritabanı dosyanızın adı **Dosya Adı** alanında görüntülenir.
8. **Anahtar veritabanı içeriği** alanında **Kişisel Sertifikalar** seçeneğini belirleyin ve sertifikayı seçin.
9. **Sertifikayı çek**' i tıklatın. Bir Sertifikayı Dosyaya Çıkar penceresi açılır.
10. Sertifikanın **Veri tipi** ' ni seçin; örneğin, `.arm` uzantılı bir dosya için **Base64-encoded ASCII verileri** .
11. Sertifikayı saklamak istediğiniz sertifika dosyası adını ve yerini yazın ya da adı ve yeri seçmek için **Göz At** düğmesini tıklatın.
12. **Tamam** düğmesini tıklatın. Sertifika belirttiğiniz dosyaya yazılır. Bir sertifikayı çıkardığınızda (dışa aktarmak yerine), sertifikanın yalnızca genel kısmı içerilir; bu nedenle parola gerekmez.

Komut satırının kullanılması

runmqckm ya da **runmqakm**kullanarak kendinden onaylı bir sertifikanın genel kısmını çıkarmak için aşağıdaki komutları kullanın:

- runmqckm kullanılıyor:

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

- Runmqakm kullanılıyor:

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

Burada:

`-db filename`	CMS anahtar veritabanının tam olarak nitelenmiş yol adıdır.
`-pw password`	CMS anahtar veritabanının parolasıdır.

`-label` *label*	sertifikaya iliştirilen etikettir.
`-target` *filename*	hedef dosyanın adıdır.
`-format` *ascii*	sertifikanın biçimidir. Değer, Base64-encoded ASCII için `ascii` ya da İkili DER verileri için `binary` olabilir. Varsayılan değer `ascii`' dir.
`-fips`	komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.

ALW *AIX, Linux, and Windows üzerindeki bir anahtar havuzuna CA sertifikası ya da kendinden onaylı bir sertifikanın genel kısmını ekleme*

Anahtar havuzuna kendinden onaylı bir sertifika kuruluşu (CA) sertifikası ya da sertifikanın genel kısmını eklemek için bu yordamı izleyin.

Eklemek istediğiniz sertifika bir sertifika zincirinde yer aldıysa, zincirin üstündeki tüm sertifikaları da eklemeniz gerekir. Sertifikaları kökten başlayarak tam olarak azalan düzende, ardından zincirde hemen altındaki sertifika kuruluşu (CA) sertifikasını eklemeniz gerekir.

Aşağıdaki yönergeler bir CA sertifikasını işaret ediyorlarsa, bunlar kendinden onaylı bir sertifikanın genel bölümü için de geçerlidir.

Not: IBM Global Security Kit (GSKit) , diğer kodlama tiplerine sahip sertifikaları desteklemediğinden, sertifikanın ASCII (UTF-8) ya da ikili (DER) kodlamasında olduğundan emin olmanız gerekir.

Kullanılan `strmqikm`

TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın. **strmqikm** , FIPS uyumlu bir seçenek sağlamaz.

CA sertifikasını eklemek istediğiniz makinede aşağıdaki adımları gerçekleştirin:

1. **strmqikm** komutunu kullanarak GUI ' yi başlatın.
2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın. Aç penceresi açılır.
3. **Anahtar veritabanı tipi** seçeneğini tıklatın ve **CMS** (Sertifika Yönetimi Sistemi) seçeneğini belirleyin.
4. Anahtar veri tabanı dosyalarını içeren dizine gitmek için **Göz At** düğmesini tıklatın.
5. Sertifikayı eklemek istediğiniz anahtar veritabanı dosyasını seçin; örneğin, `key.kdb`.
6. **Tamam** düğmesini tıklatın. Password Prompt (Parola İstemi) penceresi açılır.
7. Anahtar veritabanını yaratırken ayarladığınız parolayı yazın ve **Tamam**düğmesini tıklatın. Anahtar veritabanı dosyanızın adı **Dosya Adı** alanında görüntülenir.
8. **Anahtar veritabanı içeriği** alanında **İmzalayan Sertifikaları**seçeneğini belirleyin.
9. **Ekle**düğmesini tıklatın. Dosyadan CA Sertifikası Ekle penceresi açılır.
10. Sertifikanın saklandığı sertifika dosyası adını ve yerini yazın ya da adı ve yeri seçmek için **Göz At** düğmesini tıklatın.
11. **Tamam** düğmesini tıklatın. Enter a Label (Etiket Girin) penceresi açılır.
12. Etiket Girin penceresinde sertifikanın adını yazın.
13. **Tamam** düğmesini tıklatın. Sertifika anahtar veritabanına eklenir.

Komut satırının kullanılması

Bir anahtar veritabanına CA sertifikası eklemek için aşağıdaki komutlardan birini kullanın:

- **runmqckm**komutunu kullanarak:

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

- **runmqakm**komutunu kullanarak:

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

Burada:

-db *kütükadı*
CMS anahtar veritabanının tam olarak nitelenmiş dosya adını belirtir.

-pw *parola*
CMS anahtar veri tabanına ilişkin parolayı belirtir.

-label *etiket*
Sertifikaya eklenen etiketi belirtir.

-file *kütükadı*
Sertifikayı içeren dosyanın adını belirtir.

-format *ascii*
Sertifikanın biçimini belirtir. Değer, Base64-encoded ASCII için `ascii` ya da İkili DER verileri için `binary` olabilir. Varsayılan değer `ascii`' dir.

-fips
Komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.

ALW *AIX, Linux, and Windows üzerindeki bir anahtar havuzundan kişisel sertifikayı dışa aktarma*

Kişisel sertifikayı dışa aktarmak için bu yordamı izleyin.

Kullanılan `strmqikm`

TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın. **strmqikm** (iKeyman), FIPS uyumlu bir seçenek sağlamaz.

Kişisel sertifikayı dışa aktarmak istediğiniz makinede aşağıdaki adımları gerçekleştirin:

1. **strmqikm** komutunu kullanarak GUI ' yi başlatın.
2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın. Aç penceresi açılır.
3. **Anahtar veritabanı tipi** seçeneğini tıklatın ve **CMS** (Sertifika Yönetimi Sistemi) seçeneğini belirleyin.
4. Anahtar veri tabanı dosyalarını içeren dizine gitmek için **Göz At** düğmesini tıklatın.
5. Sertifikayı dışa aktarmak istediğiniz anahtar veritabanı dosyasını seçin; örneğin, `key.kdb`.
6. **Aç**' ı tıklatın. Password Prompt (Parola İstemi) penceresi açılır.
7. Anahtar veritabanını yaratırken ayarladığınız parolayı yazın ve **Tamam**düğmesini tıklatın. Anahtar veritabanı dosyanızın adı **Dosya Adı** alanında görüntülenir.
8. **Anahtar veritabanı içeriği** alanında **Kişisel Sertifikalar** seçeneğini belirleyin ve dışa aktarmak istediğiniz sertifikayı seçin.
9. **Dışa/İçe Aktar**' ı tıklatın. Anahtarı Dışa Aktar/İçe Aktar penceresi açılır.
10. **Anahtarı Dışa Aktar**seçeneğini belirleyin.
11. Dışa aktarmak istediğiniz sertifikanın **anahtar dosyası tipini** seçin; örneğin, **PKCS12**.
12. Sertifikayı dışa aktarmak istediğiniz dosya adını ve konumu yazın ya da adı ve yeri seçmek için **Göz At** düğmesini tıklatın.

13. **Tamam** düğmesini tıklatın. Password Prompt (Parola İstemi) penceresi açılır. Bir sertifikayı dışa aktardığınızda (ayıklamak yerine), sertifikanın hem genel hem de özel bölümlerinin içerildiğini unutmayın. Bu nedenle, dışa aktarılan dosya bir parolayla korunur. Bir sertifikayı çıkardığınızda, sertifikanın yalnızca genel kısmı eklenir, bu nedenle parola gerekmez.
14. **Parola** alanına bir parola yazın ve **Parolayı Onayla** alanına parolayı yeniden yazın.
15. **Tamam** düğmesini tıklatın. Sertifika belirttiğiniz dosyaya aktarılır.

Komut satırının kullanılması

runmqckm komutunu ya da **runmqakm** komutunu kullanarak kişisel sertifikayı dışa aktarın:

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

veya

```
runmqakm -cert -export -db filename -pw password -label label -type cms
        -target filename -target_pw password -target_type pkcs12
        -encryption strong | weak -fips
```

Burada:

-db *filename*	CMS anahtar veritabanının tam olarak nitelenmiş yol adıdır.
-encryption	sertifika dışa aktarma komutunda kullanılan şifreleme gücüdür. Değer güçlü ya da zayıfolabilir. Varsayılan değer strong(güçlü) değeridir.
-fips	komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.
-pw *password*	CMS anahtar veritabanının parolasıdır.
-label *label*	sertifikaya iliştirilen etikettir.
-type *cms*	Veritabanının tipi.
-target *filename*	hedef dosyanın tam olarak nitelenmiş yol adıdır.
-target_pw *password*	sertifikayı şifrelemek için kullanılan paroladır.
-target_type *pkcs12*	sertifikanın tipidir.

ALW *AIX, Linux, and Windows üzerindeki bir anahtar havuzuna kişisel sertifika aktarılması*

Kişisel bir sertifikayı içe aktarmak için bu yordamı izleyin

PKCS #12 biçimindeki bir kişisel sertifikayı anahtar veritabanı dosyasına aktarmadan önce, anahtar veritabanı dosyasına sertifika veren CA sertifikalarının tam geçerli zincirini eklemelisiniz (bkz. “AIX, Linux, and Windows üzerindeki bir anahtar havuzuna CA sertifikası ya da kendinden onaylı bir sertifikanın genel kısmını ekleme” sayfa 314).

PKCS #12 dosyaları geçici olarak kabul edilmeli ve kullanıldıktan sonra silinmelidir.

Kullanılan strmqikm

TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın. **strmqikm** , FIPS uyumlu bir seçenek sağlamaz.

Kişisel sertifikayı almak istediğiniz makinede aşağıdaki adımları gerçekleştirin:

1. **strmqikm** komutunu kullanarak GUI ' yi başlatın.

2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın. Open (Aç) penceresi görüntülenir.
3. **Anahtar veritabanı tipi** seçeneğini tıklatın ve **CMS** (Sertifika Yönetimi Sistemi) seçeneğini belirleyin.
4. Anahtar veri tabanı dosyalarını içeren dizine gitmek için **Göz At** düğmesini tıklatın.
5. Sertifikayı eklemek istediğiniz anahtar veritabanı dosyasını seçin; örneğin, `key.kdb`.
6. **Aç**'ı tıklatın. Password Prompt (Parola İstemi) penceresi görüntülenir.
7. Anahtar veritabanını yaratırken ayarladığınız parolayı yazın ve **Tamam**düğmesini tıklatın. Anahtar veritabanı dosyanızın adı **Dosya Adı** alanında görüntülenir.
8. **Anahtar veritabanı içeriği** alanında **Kişisel Sertifikalar**seçeneğini belirleyin.
9. Kişisel Sertifikalar görünümünde sertifikalar varsa, aşağıdaki adımları izleyin:
   a. **Dışa/İçe Aktar**' ı tıklatın. Anahtarı Dışa/İçe Aktar penceresi görüntülenir.
   b. **Import Key**(Anahtarı İçe Aktar) seçeneğini belirleyin.
10. Kişisel Sertifikalar görünümünde sertifika yoksa **Al**' ı tıklatın.
11. İçe aktarmak istediğiniz sertifikanın **anahtar dosyası tipini** seçin; örneğin, PKCS12.
12. Sertifikanın saklandığı sertifika dosyası adını ve yerini yazın ya da adı ve yeri seçmek için **Göz At** düğmesini tıklatın.
13. **Tamam**'ı tıklatın. Password Prompt (Parola İstemi) penceresi görüntülenir.
14. **Parola** alanında, sertifika dışa aktarıldığında kullanılan parolayı yazın.
15. **Tamam**'ı tıklatın. Etiketleri Değiştir penceresi görüntülenir. Örneğin, hedef anahtar veritabanında aynı etikete sahip bir sertifika varsa, içe aktarılmakta olan sertifikaların etiketlerini değiştirebilirsiniz. Sertifika etiketlerinin değiştirilmesi, sertifika zinciri doğrulaması üzerinde bir etki yaratmaz. Sertifikayı belirli bir kuyruk yöneticisiyle ya da IBM MQ MQI clientile ilişkilendirmek için IBM MQ , ayarlanmışsa **CERTLABL** özniteliğinin değerini ya da kuyruk yöneticisinin adı ya da IBM MQ MQI client kullanıcı oturum açma kimliği eklenmiş olarak varsayılan `ibmwebspheremq` değerini kullanır. Ayrıntılar için bkz. Dijital sertifika etiketleri .
16. Bir etiketi değiştirmek için, **Değiştirilecek bir etiket seçin** listesinden gerekli etiketi seçin. Etiket, **Yeni bir etiket girin** giriş alanına kopyalanır. Etiket metnini yeni etiketle değiştirin ve **Uygula**' yı tıklatın.
17. **Yeni bir etiket girin** giriş alanındaki metin, **Değiştirilecek bir etiket seç** alanına geri kopyalanır ve özgün olarak seçilen etiketi değiştirir ve böylece ilgili sertifikayı yeniden ilişkilendirir.
18. Değiştirilmesi gereken tüm etiketleri değiştirdiğinizde **Tamam**düğmesini tıklatın. Etiketleri Değiştir penceresi kapanır ve özgün IBM Anahtar Yönetimi penceresi **Kişisel Sertifikalar** ve **İmzalayan Sertifikaları** alanları doğru etiketli sertifikalarla güncellenerek yeniden görüntülenir.
19. Sertifika hedef anahtar veritabanına aktarılır.

Komut satırının kullanılması

runmqckmkomutunu kullanarak kişisel bir sertifikayı içe aktarmak için aşağıdaki komutu kullanın:

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

runmqakmkomutunu kullanarak kişisel bir sertifikayı içe aktarmak için aşağıdaki komutu kullanın:

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

Burada:

`-file` *filename*	PKCS #12 sertifikasını içeren dosyanın tam olarak nitelenmiş dosya adıdır.
`-pw` *password*	PKCS #12 sertifikasının parolasıdır.

`-type` *pkcs12*	dosyanın tipidir.
`-target` *filename*	Hedef CMS anahtar veritabanının adıdır.
`-target_pw` *password*	CMS anahtar veritabanının parolasıdır.
`-target_type` *cms*	`-target` tarafından belirtilen veritabanının tipi
`-label` *label*	kaynak anahtar veritabanından içe aktarılacak sertifikanın etiketidir.
`-new_label` *label*	Sertifikanın hedef veritabanında atanacağı etikettir. `-new_label` seçeneğini atlarsanız, varsayılan değer `-label` seçeneğiyle aynı seçeneği kullanmaktır.
`-fips`	komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.

runmqckm , sertifika etiketlerini doğrudan değiştirmek için bir komut sağlamaz. Bir sertifika etiketini değiştirmek için aşağıdaki adımları kullanın:

1. **-cert -export** komutunu kullanarak sertifikayı bir PKCS #12 dosyasına aktarın. `-label` seçeneği için var olan sertifika etiketini belirtin.
2. **-cert -delete** komutunu kullanarak sertifikanın var olan kopyasını özgün anahtar veritabanından kaldırın.
3. **-cert -import** komutunu kullanarak sertifikayı PKCS #12 dosyasından içe aktarın. `-label` seçeneği için eski etiketi ve `-new_label` seçeneği için gerekli yeni etiketi belirtin. Sertifika, gerekli etiketle anahtar veritabanına geri aktarılacak.

ALW *Microsoft.pfx dosyasından kişisel sertifikanın içe aktarılması*

AIX, Linux, and Windowsüzerindeki bir Microsoft.pfx dosyasından içe aktarmak için bu yordamı izleyin.

Bir .pfx dosyası, aynı anahtarla ilgili iki sertifika içerebilir. Biri kişisel ya da site sertifikasıdır (hem genel hem de özel anahtar içerir). Diğeri bir CA (imzalayıcı) sertifikasıdır (yalnızca ortak anahtar içerir). Bu sertifikalar aynı CMS anahtar veritabanı dosyasında birlikte var olamayacağı için bunlardan yalnızca biri içe aktarılabilir. Ayrıca, "kullanımı kolay ad" ya da etiket yalnızca imzalayıcı sertifikasına eklenir.

Kişisel sertifika, sistem tarafından oluşturulan Benzersiz Kullanıcı Tanıtıcısı (UUID) ile tanımlanır. Bu bölümde, daha önce CA (imzalayıcı) sertifikasına atanan kullanımı kolay adla etiketlenirken bir pfx dosyasından kişisel sertifikanın içe aktarılması gösterilir. Veren CA (imzalayıcı) sertifikaları hedef anahtar veritabanına önceden eklenmelidir. PKCS#12 dosyalarının geçici olarak kabul edilmesi ve kullanıldıktan sonra silinmesi gerektiğini unutmayın.

Bir kişisel sertifikayı kaynak pfx anahtar veritabanından içe aktarmak için aşağıdaki adımları izleyin:

1. **strmqikm** komutunu kullanarak GUI ' yi başlatın. IBM Key Management (Anahtar Yönetimi) penceresi görüntülenir.
2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın. Aç penceresi görüntülenir.
3. **PKCS12**anahtar veritabanı tipini seçin.
4. **Bu adımı gerçekleştirmeden önce pfx veritabanının bir yedeğini almanız önerilir.** İçe aktarmak istediğiniz pfx anahtar veritabanını seçin. **Aç**' ı tıklatın. Parola İstemi penceresi görüntülenir.
5. Anahtar veritabanı parolasını girin ve **Tamam**düğmesini tıklatın. IBM Key Management (Anahtar Yönetimi) penceresi görüntülenir. Başlık çubuğunda, seçilen pfx anahtar veritabanı dosyasının adı gösterilir; bu, dosyanın açık ve hazır olduğunu gösterir.
6. Listeden **İmzalayıcı Sertifikaları** ' nı seçin. Gerekli sertifikanın "kullanımı kolay adı", İmzalayıcı Sertifikaları panosunda bir etiket olarak görüntülenir.
7. Etiket girişini seçin ve imzalayıcı sertifikasını kaldırmak için **Sil** düğmesini tıklatın. Confirm (Onayla) penceresi görüntülenir.

8. **Evet**' i tıklatın. Seçilen etiket artık İmzalayıcı Sertifikaları panosunda görüntülenmez.
9. Tüm imzalayıcı sertifikaları için adım 6, 7 ve 8 'i yineleyin.
10. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın. Aç penceresi görüntülenir.
11. pfx dosyasının içe aktarıldığı hedef anahtar CMS veritabanını seçin. **Aç**' ı tıklatın. Parola İstemi penceresi görüntülenir.
12. Anahtar veritabanı parolasını girin ve **Tamam**düğmesini tıklatın. IBM Key Management (Anahtar Yönetimi) penceresi görüntülenir. Başlık çubuğu, dosyanın açık ve hazır olduğunu gösteren seçili anahtar veritabanı dosyasının adını gösterir.
13. Listeden **Kişisel Sertifikalar** ' ı seçin.
14. Kişisel Sertifikalar görünümünde sertifikalar varsa, aşağıdaki adımları izleyin:
    a. **Anahtarı dışa/içe aktar**düğmesini tıklatın. Anahtarı Dışa/İçe Aktar penceresi görüntülenir.
    b. Select Action Type (İşlem Tipi Seç) içinden **Import** (İçe Aktar) seçeneğini belirleyin.
15. Kişisel Sertifikalar görünümünde sertifika yoksa **Al**' ı tıklatın.
16. PKCS12 dosyasını seçin.
17. Adım 4 'te kullanılan pfx dosyasının adını girin. **Tamam** düğmesini tıklatın. Parola İstemi penceresi görüntülenir.
18. İmzalayıcı sertifikasını sildiğinizde belirttiğiniz parolayı belirtin. **Tamam** düğmesini tıklatın.
19. Etiketleri Değiştir penceresi görüntülenir (içe aktarma için yalnızca tek bir sertifika olması gerektiği için). Sertifikanın etiketi, `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`biçiminde olan bir UUID olmalıdır.
20. Etiketi değiştirmek için, **Değiştirilecek bir etiket seçin:** panosundan UUID ' yi seçin. Etiket, **Yeni bir etiket girin:** alanına eşlenir. Etiket metnini Adım 7 'de silinen kullanımı kolay adla değiştirin ve **Uygula**' yı tıklatın. Kolay ad, ayarlanmışsa IBM MQ **CERTLABL** özniteliğinin değeri ya da kuyruk yöneticisinin adı ya da sonuna IBM MQ MQI client kullanıcı oturum açma kimliği eklenmiş olarak varsayılan `ibmwebspheremq` değeri olmalıdır. Ayrıntılar için bkz. Dijital sertifika etiketleri .
21. **Tamam** düğmesini tıklatın. Etiketleri Değiştir penceresi kaldırılır ve özgün IBM Anahtar Yönetimi penceresi, Kişisel Sertifikalar ve İmzalayıcı Sertifikaları panolarının doğru etiketli kişisel sertifikayla güncellenmesiyle yeniden görüntülenir.
22. pfx kişisel sertifikası şimdi (hedef) veritabanına aktarılır.

runmqckm ya da **runmqakm**kullanılarak bir sertifika etiketi değiştirilemez.

Komut satırının kullanılması

runmqckmkomutunu kullanarak kişisel bir sertifikayı içe aktarmak için aşağıdaki komutu kullanın:

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

runmqakmkomutunu kullanarak kişisel bir sertifikayı içe aktarmak için aşağıdaki komutu kullanın:

```
runmqakm -cert -import -file filename -pw password -type pkcs12 -target filename
-target_pw password -target_type cms -label label -fips -pfx
```

Burada:

`-file filename`	PKCS #12 sertifikasını içeren dosyanın tam olarak nitelenmiş dosya adıdır.
`-pw password`	PKCS #12 sertifikasının parolasıdır.
`-type pkcs12`	dosyanın tipidir.
`-target filename`	Hedef CMS anahtar veritabanının adıdır.

`-target_pw` *password*	CMS anahtar veritabanının parolasıdır.
`-target_type` *cms*	`-target` tarafından belirtilen veritabanının tipi
`-label` *label*	kaynak anahtar veritabanından içe aktarılacak sertifikanın etiketidir.
`-new_label` *label*	Sertifikanın hedef veritabanında atanacağı etikettir. `-new_label` seçeneğini atlarsanız, varsayılan değer `-label` seçeneğiyle aynı seçeneği kullanmaktır.
`-fips`	komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.
`-pfx`	PFX dosya biçimini gösterir.

runmqckm , sertifika etiketlerini doğrudan değiştirmek için bir komut sağlamaz. Bir sertifika etiketini değiştirmek için aşağıdaki adımları kullanın:

1. **-cert -export** komutunu kullanarak sertifikayı bir PKCS #12 dosyasına aktarın. `-label` seçeneği için var olan sertifika etiketini belirtin.
2. **-cert -delete** komutunu kullanarak sertifikanın var olan kopyasını özgün anahtar veritabanından kaldırın.
3. **-cert -import** komutunu kullanarak sertifikayı PKCS #12 dosyasından içe aktarın. `-label` seçeneği için eski etiketi ve `-new_label` seçeneği için gerekli yeni etiketi belirtin. Sertifika, gerekli etiketle anahtar veritabanına geri aktarılacak.

ALW *Kişisel sertifikayı PKCS #7 dosyasından içe aktarma*

strmqikm (iKeyman) ve **runmqckm** (iKeycmd) araçları PKCS #7 ' yi (`.p7b`) desteklemez dosyalar. AIX, Linux, and Windowsüzerindeki bir PKCS #7 dosyasından sertifikaları içe aktarmak için **runmqakm** aracını kullanın.

Bir PKCS #7 dosyasından CA sertifikası eklemek için aşağıdaki komutu kullanın:

```
runmqakm -cert -add -db filename -pw password -type cms -file filename
-label label
```

`-db` *filename*	CMS anahtar veritabanının tam olarak nitelenmiş dosya adıdır.
`-pw` *password*	anahtar veritabanının parolasıdır.
`-type` *cms*	anahtar veritabanının tipidir.
`-file` *filename*	PKCS #7 dosyasının adıdır.
`-label` *label*	Sertifikanın hedef veritabanında atandığı etikettir. İlk sertifika verilen etiketi alır. Varsa, diğer tüm sertifikalar konu adlarıyla etiketlenir.

Bir PKCS #7 dosyasından kişisel sertifikayı içe aktarmak için aşağıdaki komutu kullanın:

```
runmqakm -cert -import -db filename -pw password -type pkcs7 -target filename
-target_pw password -target_type cms -label label -new_label label
```

`-db` *filename*	PKCS #7 sertifikasını içeren dosyanın tam olarak nitelenmiş dosya adıdır.
`-pw` *password*	PKCS #7 sertifikasının parolasıdır.
`-type` *pkcs7*	dosyanın tipidir.
`-target` *filename*	hedef anahtar veritabanının adıdır.
`-target_pw` *password*	hedef anahtar veritabanının parolasıdır.
`-target_type` *cms*	`-target` tarafından belirtilen veritabanının tipi

`-label label`	içe aktarılacak sertifikanın etiketidir.
`-new_label label`	Sertifikanın hedef veritabanında atanacağı etikettir. `-new_label` seçeneğini atlarsanız, varsayılan değer `-label` seçeneğiyle aynı seçeneği kullanmaktır.

ALW *AIX, Linux, and Windows üzerindeki bir anahtar havuzundan sertifika silinmesi*

Kişisel ya da CA sertifikalarını kaldırmak için bu yordamı kullanın.

Kullanılan `strmqikm`

TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqakm** komutunu kullanın. **strmqikm** (iKeyman), FIPS uyumlu bir seçenek sağlamaz.

1. **strmqikm** komutunu kullanarak GUI ' yi başlatın.
2. **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın. Aç penceresi açılır.
3. **Anahtar veritabanı tipi** seçeneğini tıklatın ve **CMS** (Sertifika Yönetimi Sistemi) seçeneğini belirleyin.
4. Anahtar veri tabanı dosyalarını içeren dizine gitmek için **Göz At** düğmesini tıklatın.
5. Sertifikayı silmek istediğiniz anahtar veritabanı dosyasını seçin; örneğin, `key.kdb`.
6. **Aç**' ı tıklatın. Password Prompt (Parola İstemi) penceresi açılır.
7. Anahtar veritabanını yaratırken ayarladığınız parolayı yazın ve **Tamam**düğmesini tıklatın. Anahtar veritabanı dosyanızın adı **Dosya Adı** alanında görüntülenir.
8. Açılan listeden **Kişisel Sertifikalar** ya da **İmzalayıcı Sertifikaları** öğesini seçin.
9. Silmek istediğiniz sertifikayı seçin.
10. Sertifikanın bir kopyası yoksa ve sertifikayı kaydetmek istiyorsanız, **Dışa/İçe Aktar** ' ı tıklatın ve dışa aktarın (bkz. “AIX, Linux, and Windows üzerindeki bir anahtar havuzundan kişisel sertifikayı dışa aktarma” sayfa 315).
11. Sertifika seçildiğinde **Sil**düğmesini tıklatın. Confirm (Onayla) penceresi açılır.
12. **Evet**' i tıklatın. **Kişisel Sertifikalar** alanı artık sildiğiniz sertifikanın etiketini göstermez.

Komut satırının kullanılması

runmqckm komutunu ya da **runmqakm** komutunu kullanarak bir sertifikayı silmek için aşağıdaki komutları kullanın:

runmqckm kullanılıyor:

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

Runmqakm kullanılıyor:

```
runmqakm -cert -delete -db filename -pw password -label label -fips
```

Burada:

`-db filename`	CMS anahtar veritabanının tam olarak nitelenmiş dosya adıdır.
`-pw password`	CMS anahtar veritabanının parolasıdır.
`-label label`	kişisel sertifikaya iliştirilen etikettir.

`-fips`	komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.

ALW AIX, Linux, and Windows üzerinde anahtar havuzu koruması için güçlü parolalar oluşturma

runmqakm (GSKCapiCmd) komutunu kullanarak anahtar havuzu koruması için güçlü parolalar oluşturabilirsiniz.

Güçlü bir parola oluşturmak için aşağıdaki parametrelerle **runmqakm** komutunu kullanabilirsiniz:

```
runmqakm -random -create -length 14 -strong -fips
```

Sonraki sertifika yönetimi komutlarının **-pw** değiştirgesinde oluşturulan parolayı kullanırken, parolanın etrafına her zaman çift tırnak işareti koyun. AIX and Linux sistemlerinde, parola dizgisinde görüntüleniyorsa, aşağıdaki karakterlerden kaçmak için ters eğik çizgi karakteri de kullanmanız gerekir:

```
!  \  "  '
```

Parolayı **runmqckm**, **runmqakm** ya da **strmqikm** GUI 'sinden gelen bir bilgi istemine yanıt olarak girerken, parolanın tırnak içine alınması ya da paroladan çıkış yapılması gerekmez. İşletim sistemi kabuğu bu durumlarda veri girişini etkilemediği için bu gerekli değildir.

ALW AIX, Linux, and Windows üzerinde şifreleme donanımı için yapılandırma

Bir kuyruk yöneticisi ya da istemci için şifreleme donanımını çeşitli şekillerde yapılandırabilirsiniz.

Aşağıdaki yöntemlerden birini kullanarak AIX, Linux, and Windows üzerinde bir kuyruk yöneticisi için şifreleme donanımını yapılandırabilirsiniz:

- **ALTER QMGR** MQSC komutunu SSLCRYP parametresiyle birlikte kullanın; açıklamalar için bkz. ALTER QMGR.
- AIX, Linux, and Windows sisteminizde şifreleme donanımını yapılandırmak için IBM MQ Explorer komutunu kullanın. Ek bilgi için çevrimiçi yardıma bakın.

Aşağıdaki yöntemlerden birini kullanarak AIX, Linux, and Windows üzerinde bir IBM MQ istemcisi için şifreleme donanımı yapılandırabilirsiniz:

- **MQSSLCRYP** ortam değişkenini ayarlayın. **MQSSLCRYP** için izin verilen değerler, ALTER QMGRiçinde açıklandığı gibi **SSLCRYP** parametresiyle aynıdır. Bu ortam değişkenini ayarlamak için aşağıdaki komutlardan birini kullanın:
  - Linux AIX AIX and Linux sistemlerinde:

    ```
    export MQSSLCRYP=string
    ```

  - Windows Windows sistemlerinde:

    ```
    SET MQSSLCRYP=string
    ```

  Burada *string* , sistemde bulunan şifreleme donanımını yapılandırmak için kullanılacak parametre dizgisini temsil eder.

  SSLCRYP değiştirgesinin GSK_PKCS11 sürümünü kullanıyorsanız, PKCS #11 simge etiketi, donanımınızı yapılandırdığınız etiketle eşleşmelidir.
- IBM MQ client yapılandırma dosyasının SSL kısmına ilişkin **SSLCryptoHardware** özniteliğini ayarlayın. İzin verilen değerler, **ALTER QMGR**içinde açıklandığı gibi **SSLCRYP** parametresiyle aynıdır.

SSLCRYP değiştirgesinin GSK_PKCS11 sürümünü kullanıyorsanız, PKCS #11 simge etiketi, donanımınızı yapılandırdığınız etiketle eşleşmelidir.

- Bir MQCONNX çağrısında SSL yapılandırma seçenekleri yapısının (MQSCO) **CryptoHardware** alanını ayarlayın. Daha fazla bilgi için bakınız: Overview for MQSCO.

Uyarı: V 9.3.0 **MQSSLCRYP** ortam değişkeni ya da **SSLCryptoHardware** özniteliği aracılığıyla şifreleme donanımı için yapılandırma sağladığınızda, depolamadan önce parolayı korumanız gerekir. Daha fazla bilgi için bkz "Şifreleme donanımını kullanan IBM MQ clients" sayfa 580.

Bu yöntemlerden herhangi birini kullanarak PKCS #11 arabirimini kullanan şifreleme donanımını yapılandırdıysanız, kişisel sertifikayı, yapılandırdığınız şifreleme simgesi için anahtar veritabanı dosyasında kanallarınızda kullanmak üzere saklamanız gerekir. Bu, "PKCS #11 donanımında sertifikaların yönetilmesi" sayfa 323içinde açıklanmıştır.

ALW *PKCS #11 donanımında sertifikaların yönetilmesi*

PKCS #11 arabirimini destekleyen şifreleme donanımında sayısal sertifikaları yönetebilirsiniz.

Bu görev hakkında

Sertifika yetkilisi (CA) sertifikalarını depolamak istemeseniz de, IBM MQ ortamını hazırlamak için bir anahtar veritabanı oluşturmanız gerekir, ancak tüm sertifikalarınızı şifreleme donanımınızda saklar. Kuyruk yöneticisinin SSLKEYR alanında ya da istemci uygulamasının MQSSLKEYR ortam değişkeninde başvurması için bir anahtar veritabanı gereklidir. Bir sertifika isteği yaratıyorsanız, bu anahtar veritabanı da gereklidir.

Anahtar veritabanını komut satırını kullanarak ya da **strmqikm** (iKeyman) kullanıcı arabirimini kullanarak yaratırsınız.

Yordam

Komut satırını kullanarak bir anahtar veritabanı yaratın.

1. Aşağıdaki komutlardan birini çalıştırın:
   - **runmqckm**komutunu kullanarak:

     V 9.3.0 V 9.3.0
     ```
     runmqckm -keydb -create -db filename -pw password -type type -stash
     ```
   - **runmqakm**komutunu kullanarak:

     V 9.3.0 V 9.3.0
     ```
     runmqakm -keydb -create -db filename -pw password -type type
     ```

   Burada:

   -db *kütükadı*
   CMS anahtar veritabanınınolmalıdır.

   -pw *parola*
   CMS V 9.3.0 V 9.3.0 ya da PKCS#12 anahtar veritabanına ilişkin parolayı belirtir.

   V 9.3.0 V 9.3.0 **-type *tip***
   Veri tabanının tipini belirtir. (IBM MQiçin cms ya da pkcs12olmalıdır).

   -stash
   V 9.3.0 V 9.3.0 İsteğe bağlı. Anahtar veritabanı parolasını bir dosyaya kaydeder.

Alternatif olarak, **strmqikm** (iKeyman) kullanıcı arabirimini kullanarak bir anahtar veritabanı oluşturun.

2. AIX and Linux sistemlerinde kök kullanıcı olarak oturum açın. Windows sistemlerinde Administrator (Yönetici) ya da MQM grubunun bir üyesi olarak oturum açın.

3. Java güvenlik özellikleri dosyasını (`java.security`) açın.

   - AIX and Linux sistemlerinde Java güvenlik özellikleri dosyası, IBM MQ kuruluş dizininin `java/jre64/jre/lib/security` altdizininde bulunur.
   - Windows sistemlerinde Java güvenlik özellikleri dosyası, IBM MQ kuruluş dizininin `java\jre\lib\security` altdizininde bulunur.

   Dosyada önceden yoksa, IBMPKCS11Impl güvenlik sağlayıcısını ekleyin. Örneğin, aşağıdaki satırı ekleyerek:

   ```
   security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
   ```

4. **strmqikm** komutunu çalıştırarak kullanıcı arabirimini başlatın.
5. **Anahtar Veritabanı Dosyası** > **Aç**düğmesini tıklatın.
6. **Anahtar veritabanı tipi** ' ni tıklatın ve **PKCS11Direct**seçeneğini belirleyin.
7. **Dosya Adı** alanında, şifreleme donanımınızı yönetmeye ilişkin modülün adını yazın; örneğin, `PKCS11_API.so`.

   PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.
8. **Konum** alanına yolu girin.

   - AIX and Linux sistemlerinde bu `/usr/lib/pksc11`olabilir.
   - Windows sistemlerinde kitaplık adını yazın. `cryptoki`, örneğin.
9. **Tamam**'ı tıklatın.

   Open Cryptographic Token (Şifreleme Simgesini Aç) penceresi görüntülenir.
10. Sertifikaları saklamak için kullanmak istediğiniz şifreleme aygıtı simgesi etiketini seçin.
11. **Cryptographic Token Password** (Şifreleme Aygıtı Parolası) alanında, şifreleme donanımını yapılandırırken ayarladığınız parolayı yazın.
12. Şifreleme donanımınız, kişisel sertifika almak ya da almak için gereken imzalayıcı sertifikalarını tutma kapasitesine sahipse, her iki ikincil anahtar veritabanı onay kutusunu temizleyin ve “17” sayfa 325. adımdan devam edin.

    İmzalayıcı sertifikalarını tutmak için ikincil bir CMS V 9.3.0 V 9.3.0 ya da PKCS#12 anahtar veritabanı gerekiyorsa, **Var olan ikincil anahtar veritabanı dosyasını aç** ya da **Yeni ikincil anahtar veritabanı dosyası yarat**seçeneğini belirleyin.
13. **Dosya Adı** alanında bir dosya adı yazın.

    Bu alan zaten `key.kdb`metnini içeriyor. Kök adınız `key`ise, bu alanı değiştirmeden bırakın. Farklı bir kök adı belirttiyseniz, `key` yerine kök adınızı koyun.
14. **Konum** alanına yolu yazın. Örneğin:

    - Bir kuyruk yöneticisi için: `/var/mqm/qmgrs/QM1/ssl`
    - IBM MQ MQI clientiçin: `/var/mqm/ssl`
15. **Tamam**'ı tıklatın.

    Parola İstemi penceresi görüntülenir.
16. Bir parola girin.

    “12” sayfa 324. adımda **Var olan ikincil anahtar veritabanı dosyasını aç** seçeneğini belirlediyseniz, **Parola** alanına bir parola yazın.

    “12” sayfa 324. adımda **Yeni ikincil anahtar veritabanı dosyası yarat** seçeneğini belirlediyseniz, aşağıdaki alt adımları tamamlayın:

    a) **Parola** alanında bir parola yazın ve **Parolayı Onayla** alanına parolayı yeniden yazın.

b) V 9.3.0 V 9.3.0

Parolayı bir dosyaya saklamak istiyorsanız, **Parolayı bir dosyaya kaydedin**seçeneğini belirleyin. Parolayı saklamazsanız, anahtar veritabanı parolasını KEYRPWD özniteliğini kullanarak kuyruk yöneticisine ya da "AIX, Linux, and Windows üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması" sayfa 299içinde açıklanan yöntemlerden birini kullanarak IBM MQ MQI client ' e sağlamanız gerekir.

c) V 9.3.0 V 9.3.0

Tamam'ı tıklatın.

Parolayı bir dosyaya saklamayı seçerseniz, parolanın `key.sth` dosyasında (farklı bir kök ad belirtmediyseniz) olduğunu doğrulayan bir pencere açılır.

17. **Tamam**'ı tıklatın.

    Anahtar veritabanı içerik çerçevesi görüntülenir.

ALW *PKCS #11 donanımınız için kişisel sertifika isteme*

Şifreleme donanımınız için kişisel sertifika istemek üzere bir kuyruk yöneticisi ya da IBM MQ MQI client için bu yordamı kullanın.

Bu görev hakkında

Bu görev, kişisel sertifika istemek için **`strmqikm`** kullanıcı arabirimini nasıl kullanacağınızı açıklar. Komut satırı arabirimini kullanıyorsanız, bkz. "Komut satırının kullanılması" sayfa 307.

Not: IBM MQ , SHA-3 ya da SHA-5 algoritmalarını desteklemez. Her iki algoritma da SHA-2 ailesinin üyesi olduğundan, SHA384WithRSA ve SHA512WithRSA dijital imza algoritması adlarını kullanabilirsiniz.

Deprecated Sayısal imza algoritması adları SHA3WithRSA ve SHA5WithRSA , sırasıyla SHA384WithRSA ve SHA512WithRSA kısaltması oldukları için kullanımdan kaldırılmıştır.

Yordam

`strmqikm` (iKeyman) kullanıcı arabiriminden kişisel sertifika istemek için aşağıdaki adımları tamamlayın:

1. Şifreleme donanımınızla çalışmak için aşağıdaki adımları tamamlayın. Bkz. "PKCS #11 donanımında sertifikaların yönetilmesi" sayfa 323.
2. **Oluştur** menüsünden **Yeni Sertifika İsteği**seçeneğini tıklatın.

   Create New Key and Certificate Request (Yeni Anahtar ve Sertifika İsteği Oluştur) penceresi açılır.
3. **Anahtar Etiketi** alanına sertifika etiketini girin.

   Etiket, **CERTLABL** özniteliğinin değeri (ayarlandıysa) ya da kuyruk yöneticisinin adı ya da IBM MQ MQI client oturum açma kullanıcı kimliği eklenmiş olarak varsayılan `ibmwebspheremq` değeridir (tümü küçük harfli olarak). Ayrıntılar için bkz. Dijital sertifika etiketleri .
4. Gerek duyduğunuz **Anahtar Boyutu** ve **İmza Algoritması** ' nı seçin.
5. **Ortak Ad** ve **Kuruluş**için değerleri girin ve bir **Ülke**seçin. Geri kalan isteğe bağlı alanlar için varsayılan değerleri kabul edin ya da yeni değerler yazın ya da seçin.

   Kuruluş Birimi alanında yalnızca bir ad sağlayabileceğinizi unutmayın. Bu alanlarla ilgili daha fazla bilgi için bkz. "Ayırt Edici Adlar" sayfa 14.
6. **Sertifika isteğinin saklanacağı dosyanın adını girin** alanına varsayılan değeri kabul edin `certreq.arm`ya da tam yol ile yeni bir değer yazın.
7. **Tamam**'ı tıklatın.

   Bir doğrulama penceresi açılır.
8. **Tamam**'ı tıklatın.

   Kişisel Sertifika İstekleri listesi, yarattığınız yeni kişisel sertifika isteğinin etiketini gösterir. Sertifika isteği, "6" sayfa 325. adımda seçtiğiniz dosyada saklanır.

9. Dosyayı bir sertifika yetkilisine (CA) göndererek ya da dosyayı CA web sitesindeki istek formuna kopyalayarak yeni kişisel sertifikayı isteyin.

ALW *PKCS #11 donanımınıza kişisel sertifika alınması*

Şifreleme donanımınıza kişisel sertifika almak için bir kuyruk yöneticisi ya da IBM MQ MQI client için bu yordamı kullanın.

Başlamadan önce

Kişisel sertifikayı imzalayan CA sertifikasını ekleyin. Bunu şifreleme donanımına ya da ikincil CMS anahtar veritabanına ekleyin. İmzalı sertifikayı şifreleme donanımına almadan önce bunu yapın. Bir anahtar halkasına CA sertifikası eklemek için "AIX, Linux, and Windows üzerindeki bir anahtar havuzuna CA sertifikası ya da kendinden onaylı bir sertifikanın genel kısmını ekleme" sayfa 314başlıklı konudaki yordamı izleyin.

Yordam

- **strmqikm** (iKeyman) kullanıcı arabirimini kullanarak kişisel sertifika almak için aşağıdaki adımları tamamlayın:
  a) Şifreleme donanımınızla çalışmak için aşağıdaki adımları tamamlayın. Bkz. "PKCS #11 donanımında sertifikaların yönetilmesi" sayfa 323.
  b) **Al**düğmesini tıklatın. Dosyadan Sertifika Al penceresi açılır.
  c) Yeni kişisel sertifika için sertifika dosyası adını ve yerini yazın ya da adı ve yeri seçmek için **Göz At** düğmesini tıklatın.
  d) **Tamam**'ı tıklatın. Anahtar veritabanınızda zaten kişisel bir sertifikanız varsa, eklemekte olduğunuz anahtarı veritabanına varsayılan anahtar olarak ayarlamak isteyip istemediğinizi soran bir pencere açılır.
  e) **Evet** ya da **Hayır**' ı tıklatın. Enter a Label (Etiket Girin) penceresi açılır.
  f) **Tamam**'ı tıklatın. **Kişisel Sertifikalar** listesi, eklediğiniz yeni kişisel sertifikanın etiketini gösterir. Bu etiket, belirttiğiniz etiketin önüne şifreleme simgesi etiketi eklenerek oluşturulur.
- **runmqakm** (GSKCapiCmd) komutunu kullanarak kişisel sertifika almak için aşağıdaki adımları tamamlayın:
  a) Ortamınız için yapılandırılmış bir komut penceresi açın.
  b) **runmqakm** (GSKCapiCmd) komutunu kullanarak kişisel sertifikayı alın:

```
runmqakm -cert -receive -file filename -crypto module_name
         -tokenlabel hardware_token -pw hardware_password
         -format cert_format -fips
         -secondaryDB filename -secondaryDBpw password
```

  Burada:

  -file *kütükadı*
  Kişisel sertifikayı içeren dosyanın tam olarak nitelenmiş dosya adını belirtir.

  -crypto *modülü_adı*
  Şifreleme donanımıyla birlikte sağlanan PKCS #11 kitaplığının tam olarak nitelenmiş adını belirler.

  -tokenlabel *donanım_simgesi*
  PKCS #11 şifreleme aygıtı simgesi etiketini belirtir.

  -pw *donanım_parolası*
  Şifreleme donanımına erişmek için kullanılacak parolayı belirler.

  -format *cert_format*
  Sertifikanın biçimini belirtir. Değer, ikili DER verileri için Base64-encoded ASCII ya da `ikili` için `ascii` olabilir. Varsayılan değer ASCII 'dir.

-fips
Komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.

-secondaryDB *kütükadı*
CMS anahtar veritabanının tam olarak nitelenmiş dosya adını belirtir.

-secondaryDBpw *parola*
CMS anahtar veri tabanına ilişkin parolayı belirtir.

MQ Appliance IBM MQ Appliance üzerinde SSL/TLS ile çalışma

IBM MQ Appliance , TLS (Transport Layer Security; İletim Katmanı Güvenliği) desteğine sahiptir.

IBM MQ Appliance , sertifikaları yönetmek için ayrı komutlara sahiptir. Sertifika yönetimi hakkında ayrıntılı bilgi için IBM MQ Appliance belgelerine bakın, TLS sertifika yönetimi

z/OS z/OS üzerinde SSL/TLS ile çalışma

Bu bilgiler, z/OSüzerinde Transport Layer Security (TLS) olanağını nasıl kuracağınızı ve bu güvenlikle nasıl çalışacağınızı açıklar.

Her konu, RACFkomutunu kullanarak her görevin gerçekleştirilmesine ilişkin örnekleri içerir. Diğer dış güvenlik yöneticilerini kullanarak benzer görevleri gerçekleştirebilirsiniz.

z/OSüzerinde, her kuyruk yöneticisinin TLS çağrılarını işlemek için kullandığı sunucu alt görevlerinin sayısını da ayarlamanız gerekir (bkz. “z/OS üzerinde SSLTASKS parametresinin ayarlanması” sayfa 328).

z/OS TLS desteği, işletim sisteminin ayrılmaz bir parçasıdır ve *Sistem SSL*olarak bilinir. Sistem SSL 'si, z/OSCryptographic Services Base öğesinin bir parçasıdır. Şifreleme Hizmetleri Temel üyeleri *pdsname*içinde kurulur. SIEALNKE bölümlenmiş veri kümesi (PDS). Sistem SSL 'yi kurarken, gereksinim duyduğunuz CipherSpecs ' i sağlamak için uygun seçenekleri belirlediğinizden emin olun.

Kendinden onaylı bir sertifikayı yenilemeniz gerekirse, daha fazla bilgi için RACF' de kendinden onaylı sertifikayı yenilemeye ilişkin adımlar başlıklı konuya bakın.

z/OS *z/OS üzerinde TLS için ek kullanıcı kimliği gereksinimleri*

Bu bilgiler, z/OSüzerinde TLS ' yi ayarlamak ve TLS ile çalışmak için kullanıcı kimliğinizin gereksinim duyduğu ek gereksinimleri açıklar.

Sisteminizde uygun tüm High Impact ya da Pervasive (HIPER) güncellemelerinin bulunduğundan emin olun.

Anahtar havuzu CHINIT kullanıcı kimliğine aitse, bu kullanıcı kimliğinin IRR.DIGTCERT.LISTRING tanıtımı, tersi durumda erişimi güncelleyin ve IRR.DIGTCERT.LIST tanıtımı. PERMIT komutunu ACCESS (UPDATE) ya da ACCESS (READ) ile uygun şekilde kullanarak erişim verin

Aşağıdaki önkoşulları ayarladığınızdan emin olun:

- *ssidCHIN* kullanıcı kimliği RACFiçinde doğru şekilde tanımlanır ve *ssidCHIN* kullanıcı kimliğinin aşağıdaki tanıtımlara uygun erişimi vardır:
  - IRR.DIGTCERT.LIST
  - IRR.DIGTCERT.LISTRING

  Bu değişkenler RACF FACILITY Sınıfında tanımlanır.
- *ssidCHIN* kullanıcı kimliği, anahtarlık sahibinin kimliğidir.
- Kuyruk yöneticisinin kişisel sertifikası (RACDCERT komutuyla yaratıldıysa), *ssidCHIN* kullanıcı kimliğiyle de aynı olan bir sertifika tipi kullanıcı kimliğiyle yaratılır.
- Anahtarlık üzerinde yaptığınız değişiklikleri almak için kanal başlatıcı geri dönüştürüldü ya da **REFRESH SECURITY TYPE(SSL)** komutu verildi.

- IBM MQ Kanal Başlatıcı yordamı, *pdsname*.SIEALNKE sistem SSL çalıştırma zamanı kitaplığına bağlantı listesi, LPA ya da STEPLIB DD deyimiyle erişebilir. Bu kitaplık APF-yetkili olmalıdır.
- Kanal başlatıcısının çalıştığı yetki altındaki kullanıcı kimliği, z/OS UNIX System Services Planning belgesinde açıklandığı gibi z/OS UNIX System Services (z/OS UNIX) kullanacak şekilde yapılandırılır.

  Kanal başlatıcısının, guest/default UID ve OMVS kesimini kullanarak z/OS UNIX ' i çağırmasını istemeyen kullanıcılar, kanal başlatıcısı özel izinler gerektirmediği ve bir ayrıcalıklı kullanıcı olarak UNIX içinde çalışmadığı için, yalnızca varsayılan kesime dayalı olarak yeni bir OMVS kesimini modellemeleri gerekir.

  Bazı örnek komutlar için bkz. “Kanal başlatıcıya z/OS üzerinde doğru erişim haklarını verme” sayfa 329 .

z/OS *z/OS üzerinde SSLTASKS parametresinin ayarlanması*

TLS çağrılarının işlenmesine ilişkin sunucu alt görevlerinin sayısını ayarlamak için ALTER QMGR komutunu kullanın.

TLS kanallarını kullanmak için, ALTER QMGR komutunu kullanarak SSLTASKS parametresini ayarlayarak en az iki sunucu alt görevi olduğundan emin olun. Örneğin:

```
ALTER QMGR SSLTASKS(5)
```

Depolama ayırmasıyla ilgili sorunları önlemek için, Sertifika İptal Listesi (CRL) denetimi olmayan bir ortamda SSLTASKS özniteliğini sekizden büyük bir değere ayarlamayın.

CRL denetimi kullanılırsa, bir SSLTASK, söz konusu denetim süresince ilgili kanal tarafından tutulur. Her SSLTASK bir z/OS görev denetimi bloğu olduğundan, ilgili LDAP sunucusuyla iletişim kurulurken bu önemli bir süre için olabilir.

SSLTASKS özniteliğinin değerini değiştirirseniz kanal başlatıcıyı yeniden başlatmanız gerekir.

z/OS *z/OS üzerinde bir anahtar havuzu ayarlama*

Bağlantının her iki ucunda bir anahtar havuzu ayarlayın. Her bir anahtar havuzunu kuyruk yöneticisiyle ilişkilendirir.

TLS bağlantısı, bağlantının her sonunda bir *anahtar havuzu* gerektirir. Her kuyruk yöneticisinin bir anahtar havuzuna erişimi olmalıdır. Bir anahtar havuzunu bir kuyruk yöneticisiyle ilişkilendirmek için ALTER QMGR komutundaki SSLKEYR parametresini kullanın. Ek bilgi için bkz. “SSL/TLS anahtar havuzu” sayfa 25 .

z/OSişletim sisteminde dijital sertifikalar, Dış Güvenlik Yöneticiniz (ESM) tarafından yönetilen bir *anahtar halkasında* saklanır. Bu sayısal sertifikaların, sertifikayı bir kuyruk yöneticisiyle ilişkilendirebilen etiketleri vardır. TLS, kimlik doğrulama amacıyla bu sertifikaları kullanır. Aşağıdaki tüm örnekler RACF komutlarını kullanır. Diğer ESM programları için eşdeğer komutlar vardır.

z/OSsistemlerinde IBM MQ , ayarlanmışsa **CERTLABL** özniteliğinin değerini ya da sonuna kuyruk yöneticisinin adı eklenmiş olarak varsayılan `ibmWebSphereMQ` değerini kullanır. Ayrıntılar için bkz. Dijital sertifika etiketleri .

Bir kuyruk yöneticisine ilişkin anahtar havuzu adı, RACF veritabanınızdaki bir anahtarlık adıdır. Anahtarlık adını, anahtarlık oluşturulmadan önce ya da oluşturulduktan sonra belirleyebilirsiniz.

Bir kuyruk yöneticisine ilişkin yeni bir anahtarlık yaratmak için aşağıdaki yordamı kullanın:

1. RACDCERT komutunu vermek için gereken yetkiye sahip olduğunuzdan emin olun (daha fazla ayrıntı için RACDCERT komutunun kullanımını denetleme başlıklı konuya bakın).
2. Şu komutu verin:

   ```
   RACDCERT ID( userid1 ) ADDRING( ring-name )
   ```

   Burada:

   - *userid1* , kanal başlatıcı adres alanının ya da anahtarlık sahibi olacak kullanıcı kimliğinin (anahtarlık paylaşılıyorsa) kullanıcı kimliğidir.

- *halka-adı* , anahtarlık halkasına vermek istediğiniz addır. Bu adın uzunluğu en çok 237 karakter olabilir. Bu ad, büyük ve küçük harfe duyarlıdır. Sorunları önlemek için büyük harfli *halka-adı* belirtin.

z/OS *CA sertifikalarının z/OS üzerindeki bir kuyruk yöneticisinin kullanımına açılması*

Anahtarlık anahtarlığı oluşturduktan sonra, ilgili CA sertifikalarını bu anahtara bağlayın.

Bir veri kümesinde CA sertifikanız varsa, önce aşağıdaki komutu kullanarak sertifikayı RACF veritabanına eklemeniz gerekir:

```
RACDCERT ID( userid1 ) ADD( input-data-set-name ) WITHLABEL( 'My CA' )
```

Daha sonra My CA için bir CA sertifikasını anahtarlık halkanıza bağlamak üzere aşağıdaki komutu kullanın:

```
RACDCERT ID(userid1)
CONNECT(CERTAUTH LABEL('My CA') RING(ring-name) USAGE(CERTAUTH))
```

Burada *userid1* , kanal başlatıcı kullanıcı kimliği ya da paylaşılan anahtarlık sahibi olabilir.

CA sertifikaları hakkında daha fazla bilgi için bkz. “dijital sertifikalar” sayfa 13.

z/OS *z/OS üzerinde bir kuyruk yöneticisine ilişkin anahtar havuzunun bulunması*

Kuyruk yöneticinizin anahtar halkasının yerini almak için bu yordamı kullanın.

1. Aşağıdaki MQSC komutlarından birini kullanarak kuyruk yöneticinizin özniteliklerini görüntüleyin:

   ```
   DISPLAY QMGR ALL
   DISPLAY QMGR SSLKEYR
   ```

2. Anahtarlık konumu için komut çıkışını inceleyin.

z/OS *z/OS üzerinde bir kuyruk yöneticisi için anahtar havuzu konumunun belirtilmesi*

Kuyruk yöneticinizin anahtar halkasının yerini belirtmek için, kuyruk yöneticinizin anahtar havuzu özniteliğini ayarlamak üzere ALTER QMGR MQSC komutunu kullanın.

Örneğin:

```
ALTER QMGR SSLKEYR(CSQ1RING)
```

Anahtarlık, kanal başlatıcısı adres alanına aitse ya da:

```
ALTER QMGR SSLKEYR(userid1/CSQ1RING)
```

Bu bir paylaşılan anahtarlık ise, burada *userid1* , anahtarlık sahibi olan kullanıcı kimliğidir.

z/OS *Kanal başlatıcıya z/OS üzerinde doğru erişim haklarını verme*

Kanal başlatıcının (CHINIT) anahtar havuzuna ve belirli güvenlik profillerine erişmesi gerekir.

Anahtar havuzunu okumak için CHINIT erişimi verilmesi

Anahtar havuzu CHINIT kullanıcı kimliğine aitse, bu kullanıcı kimliğinin IRR.DIGTCERT.LISTRING tanıtımı, tersi durumda erişimi güncelleyin ve IRR.DIGTCERT.LIST tanıtımı. PERMIT komutunu ACCESS (UPDATE) ya da ACCESS (READ) ile uygun şekilde kullanarak erişim verin:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID( userid ) ACCESS(UPDATE)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID( userid ) ACCESS(READ)
```

Burada *kullanıcı kimliği* , kanal başlatıcısı adres alanının kullanıcı kimliğidir.

Uygun CSF* tanıtımlarına CHINIT okuma erişimi verilmesi

Kullanılacak Integrated Cryptographic Service Facility (ICSF) aracılığıyla sağlanan donanım desteği için, CHINIT kullanıcı kimliğinizin aşağıdaki komutu kullanarak CSFSERV sınıfındaki uygun CSF* tanıtımlarına okuma erişimine sahip olduğundan emin olun:

```
PERMIT csf-resource CLASS(CSFSERV) ID( userid ) ACCESS(READ)
```

Burada *csf-resource* , CSF* tanıtımının adı ve *userid* kanal başlatıcısı adres alanının kullanıcı kimliğidir.

Aşağıdaki CSF* tanıtımlarının her biri için bu komutu yineleyin:

- CSFDSG
- CSFDSV
- CSFPKD
- CSFPKE
- CSFPKI

CHINIT kullanıcı kimliğinizin diğer CSF* tanıtımlarına okuma erişimi de gerekebilir. Örneğin, ECDHE_RSA_AES_256_GCM_SHA384 şifreleme belirtimini kullanıyorsanız, CHINIT kullanıcı kimliğinizin aşağıdaki CSF* tanıtımlarına da okuma erişimi gerekir:

- CSF1DVK
- CSF1GAV
- CSF1GKP
- CSF1SKE
- CSF1TRC
- CSF1TRD

Daha fazla bilgi için bkz. RACF CSFSERV kaynak gereksinimleri.

Sertifika anahtarlarınız ICSF 'de saklanıyorsa ve kuruluşunuz ICSF' de saklanan anahtarlar üzerinde erişim denetimi oluşturduysa, aşağıdaki komutu kullanarak CHINIT kullanıcı kimliğinizin CSFKEYS sınıfındaki tanıtıma okuma erişimine sahip olduğundan emin olun:

```
PERMIT IRR.DIGTCERT. userid.* CLASS(CSFKEYS) ID( userid ) ACCESS(READ)
```

Burada *kullanıcı kimliği* , kanal başlatıcısı adres alanının kullanıcı kimliğidir.

Integrated Cryptographic Service Facility (ICSF) olanağının kullanılması

Kanal başlatıcısı, TLS kullanılmıyorsa istemci kanalları üzerinden akan parolaları karartmak için parola koruma algoritmasını dağıtırken rasgele bir sayı oluşturmak için ICSF ' yi kullanabilir.

Daha fazla bilgi için bkz. “Integrated Cryptographic Service Facility (ICSF) olanağının kullanılması” sayfa 259

z/OS *Sertifikalarda ya da anahtar havuzunda yapılan değişiklikler z/OS üzerinde yürürlüğe girdiğinde*

Kanal başlatıcı başlatıldığında ya da havuz yenilendiğinde değişiklikler etkili olur.

Özellikle, anahtar halkasındaki sertifikalarda ve anahtar havuzu özniteliğinde yapılan değişiklikler aşağıdaki durumlardan birinde etkili olur:

- Kanal başlatıcı başlatıldığında ya da yeniden başlatıldığında.
- Anahtar havuzunun içeriğini yenilemek için REFRESH SECURITY TYPE (SSL) komutu verildiğinde.

z/OS z/OS üzerinde kendinden onaylı kişisel sertifika oluşturma

Kendinden onaylı bir kişisel sertifika yaratmak için bu yordamı kullanın.

1. Aşağıdaki komutu kullanarak bir sertifika ve genel ve özel anahtar çifti oluşturun:

```
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. Aşağıdaki komutu kullanarak sertifikayı anahtarlık halkanıza bağlayın:

```
RACDCERT ID(userid1)
CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
```

Burada:

- *userid1* , kanal başlatıcı adres alanının ya da paylaşılan anahtarlık sahibinin kullanıcı kimliğidir.
- *userid2* , sertifikayla ilişkili kullanıcı kimliğidir ve kanal başlatıcı adres alanının kullanıcı kimliği olmalıdır.

  userid1 ve *userid2* aynı kimlik olabilir.
- *halka-adı* , “z/OS üzerinde bir anahtar havuzu ayarlama” sayfa 328içinde anahtarlık verdiğiniz addır.
- *etiket-adı* , ayarlandıysa IBM MQ **CERTLABL** özniteliğinin değeri ya da sonuna kuyruk yöneticisinin adı eklenmiş olarak varsayılan `ibmWebSphereMQ` değeri olmalıdır. Ayrıntılar için bkz. Dijital sertifika etiketleri .

z/OS z/OS üzerinde kişisel sertifika isteme

RACFkomutunu kullanarak kişisel sertifika için başvurun.

Kişisel sertifika için başvurmak üzere RACF ' yi aşağıdaki gibi kullanın:

1. “z/OS üzerinde kendinden onaylı kişisel sertifika oluşturma” sayfa 331içinde olduğu gibi kendinden onaylı bir kişisel sertifika oluşturun. Bu sertifika, isteği Ayırt Edici Ada ilişkin öznitelik değerleriyle birlikte sağlar.
2. Aşağıdaki komutu kullanarak bir veri kümesine yazılan PKCS No. 10 Base64-encoded sertifika isteği yaratın:

   ```
   RACDCERT ID(userid2) GENREQ(LABEL(' label_name ')) DSN(' output_data_set_name ')
   ```

   burada:

   - *userid2* , sertifikayla ilişkili kullanıcı kimliğidir ve kanal başlatıcı adres alanının kullanıcı kimliği olmalıdır
   - *label_name* , kendinden onaylı sertifika oluşturulurken kullanılan etikettir

   Ayrıntılar için bkz. “Dijital sertifika etiketleri, gereksinimlerin anlaşılması” sayfa 26.
3. Yeni bir kişisel sertifika istemek için veri kümesini bir Sertifika Yetkilisi 'ne (CA) gönderin.
4. İmzalı sertifika Sertifika Yetkilisi tarafından size geri gönderildiğinde, sertifikayı, “z/OS üzerinde bir anahtar havuzuna kişisel sertifikalar ekleme” sayfa 332başlıklı konuda açıklandığı gibi özgün etiketi kullanarak RACF veritabanına geri ekleyin.

z/OS RACF imzalı kişisel sertifika oluşturma

RACF , bir sertifika yetkilisi olarak işlev görür ve kendi CA sertifikasını verir.

Bu bölümde, RACFtarafından verilen bir CA sertifikasını belirtmek için *imzalayıcı sertifikası* terimi kullanılır.

Aşağıdaki yordamı gerçekleştirmeden önce imzalayıcı sertifikasına ilişkin özel anahtarın RACF veritabanında olması gerekir:

1. RACF veritabanınızda bulunan imzalayıcı sertifikasını kullanarak RACFtarafından imzalanmış bir kişisel sertifika oluşturmak için aşağıdaki komutu kullanın:

```
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. Aşağıdaki komutu kullanarak sertifikayı anahtarlık halkanıza bağlayın:

```
RACDCERT ID(userid1)
CONNECT(ID(userid2) LABEL('label-name') RING(ring-name) USAGE(PERSONAL))
```

Burada:

- *userid1* , kanal başlatıcı adres alanının ya da paylaşılan anahtarlık sahibinin kullanıcı kimliğidir.
- *userid2* , sertifikayla ilişkili kullanıcı kimliğidir ve kanal başlatıcı adres alanının kullanıcı kimliği olmalıdır.

  userid1 ve *userid2* aynı kimlik olabilir.
- *halka-adı* , “z/OS üzerinde bir anahtar havuzu ayarlama” sayfa 328içinde anahtarlık verdiğiniz addır.
- *etiket-adı* , ayarlandıysa IBM MQ **CERTLABL** özniteliğinin değeri ya da sonuna kuyruk yöneticisi ya da kuyruk paylaşım grubu eklenmiş varsayılan `ibmWebSphereMQ` değeri olmalıdır. Ayrıntılar için bkz. Dijital sertifika etiketleri .
- *imzalayıcı-etiket* , kendi imzalayıcı sertifikanız etiketidir.

z/OS z/OS üzerinde bir anahtar havuzuna kişisel sertifikalar ekleme

Bir anahtar halkasına kişisel sertifika eklemek ya da almak için bu yordamı kullanın.

Sertifika yetkilisi size yeni bir kişisel sertifika gönderdikten sonra, aşağıdaki yordamı kullanarak bunu anahtar halkasına ekleyin:

1. Aşağıdaki komutu kullanarak sertifikayı RACF veritabanına ekleyin:

```
RACDCERT ID( userid2 ) ADD( input-data-set-name ) WITHLABEL(' label-name ')
```

2. Aşağıdaki komutu kullanarak sertifikayı anahtarlık halkanıza bağlayın:

```
RACDCERT ID( userid1 )
CONNECT(ID( userid2 ) LABEL(' label-name ') RING( ring-name ) USAGE(PERSONAL))
```

Burada:

- *userid1* , kanal başlatıcı adres alanının ya da paylaşılan anahtarlık sahibinin kullanıcı kimliğidir.
- *userid2* , sertifikayla ilişkili kullanıcı kimliğidir ve kanal başlatıcı adres alanının kullanıcı kimliği olmalıdır.
- *halka-adı* , “z/OS üzerinde bir anahtar havuzu ayarlama” sayfa 328içinde anahtarlık verdiğiniz addır.
- *giriş-veri-kümesi-adı* , CA imzalı sertifikayı içeren veri kümesinin adıdır. Veri kümesi kataloğa alınmalı ve bir PDS ya da PDS üyesi olmamalıdır. RACDCERT tarafından beklenen kayıt biçimi (RECFM) VB 'dir. RACDCERT, veri kümesini dinamik olarak ayırır ve açar ve sertifikayı ikili veri olarak okur.

- *etiket-adı* , özgün isteği yarattığınızda kullanılan etiket adıdır. Ayarlıysa, IBM MQ **CERTLABL** özniteliğinin değeri ya da kuyruk yöneticisinin ya da kuyruk paylaşım grubunun adının eklendiği varsayılan `ibmWebSphereMQ` değeri olmalıdır. Ayrıntılar için bkz. Dijital sertifika etiketleri .

z/OS z/OS üzerindeki bir anahtar havuzundan kişisel sertifikayı dışa aktarma

RACDCERT komutunu kullanarak sertifikayı dışa aktarın.

Sertifikayı dışa aktarmak istediğiniz sistemde aşağıdaki komutu kullanın:

```
RACDCERT ID(userid2) EXPORT(LABEL('label-name'))
DSN(output-data-set-name) FORMAT(CERTB64)
```

Burada:

- *userid2* , sertifikanın anahtar halkasına eklendiği kullanıcı kimliğidir.
- *etiket-adı* , çıkarmak istediğiniz sertifikanın etiketidir.
- *çıkış-veri-kümesi-adı* , sertifikanın yerleştirildiği veri kümeidir.
- CERTB64 , Base64 biçiminde olan DER kodlu X.509 sertifikasıdır. Alternatif bir biçim seçebilirsiniz, örneğin:

  CERTDER
  DER kodlu X.509 sertifika ikili biçimde

  PKCS12B64
  Base64 biçiminde PKCS #12 sertifikası

  PKCS12DER
  İkili biçimde PKCS #12 sertifikası

z/OS z/OS üzerindeki bir anahtar havuzundan kişisel sertifikanın silinmesi

RACDCERT komutunu kullanarak kişisel bir sertifikayı silin.

Kişisel bir sertifikayı silmeden önce, bir kopyasını kaydetmek isteyebilirsiniz. Kişisel sertifikanızı silmeden önce bir veri kümesine kopyalamak için “z/OS üzerindeki bir anahtar havuzundan kişisel sertifikayı dışa aktarma” sayfa 333başlıklı konudaki yordamı izleyin. Kişisel sertifikanızı silmek için aşağıdaki komutu kullanın:

```
RACDCERT ID( userid2 ) DELETE(LABEL(' label-name '))
```

Burada:

- *userid2* , sertifikanın anahtar halkasına eklendiği kullanıcı kimliğidir.
- *etiket-adı* , silmek istediğiniz sertifikanın adıdır.

z/OS z/OS üzerinde bir anahtar havuzunda kişisel sertifikanın yeniden adlandırılması

RACDCERT komutunu kullanarak bir sertifikayı yeniden adlandırın.

Belirli bir etikete sahip bir sertifikanın bulunmasını istemiyorsanız, aşağıdaki komutu kullanarak sertifikayı geçici olarak yeniden adlandırabilirsiniz:

```
RACDCERT ID( userid2 ) LABEL(' label-name ') NEWLABEL(' new-label-name ')
```

Burada:

- *userid2* , sertifikanın anahtar halkasına eklendiği kullanıcı kimliğidir.
- *etiket-adı* , yeniden adlandırmak istediğiniz sertifikanın adıdır.
- *yeni-etiket-adı* , sertifikanın yeni adıdır.

Bu, TLS istemci kimlik doğrulaması test edilirken yararlı olabilir.

z/OS *z/OS üzerinde bir dijital sertifikayla kullanıcı kimliğini ilişkilendirme*

IBM MQ , kanal kullanıcı kimliği olarak RACF sertifikasıyla ilişkilendirilmiş bir kullanıcı kimliğini kullanabilir. Bir kullanıcı kimliğini o kullanıcı kimliği altına kurarak ya da bir Sertifika Adı Süzgeci kullanarak bir sertifikayla ilişkilendirin.

Bu konuda açıklanan yöntem, bir kullanıcı kimliğini kanal kimlik doğrulama kayıtlarını kullanan bir sayısal sertifikayla ilişkilendirmek için platformdan bağımsız yönteme bir alternatiftir. Kanal kimlik doğrulama kayıtlarıyla ilgili daha fazla bilgi için bkz. “Kanal kimlik doğrulama kayıtları” sayfa 50.

TLS kanalının bir ucundaki bir varlık uzak bağlantıdan bir sertifika aldığında, varlık RACF ' e bu sertifikayla ilişkilendirilmiş bir kullanıcı kimliği olup olmadığını sorar. Varlık, kanal kullanıcı kimliği olarak bu kullanıcı kimliğini kullanır. Sertifikayla ilişkilendirilmiş bir kullanıcı kimliği yoksa, varlık kanal başlatıcısının çalıştığı kullanıcı kimliğini kullanır.

Bir kullanıcı kimliğini aşağıdaki yollardan biriyle bir sertifikayla ilişkilendirin:

- Bu sertifikayı, “z/OS üzerinde bir anahtar havuzuna kişisel sertifikalar ekleme” sayfa 332içinde açıklandığı şekilde, ilişkilendirmek istediğiniz kullanıcı kimliği altındaki RACF veritabanına kurun.
- Sertifika verenin ya da sertifikayı verenin ayırt edici adını kullanıcı kimliğiyle eşlemek için bir Sertifika Adı Süzgeci (CNF) kullanın (açıklamalar için bkz. “z/OS üzerinde bir sertifika adı süzgeci ayarlama” sayfa 334).

z/OS *z/OS üzerinde bir sertifika adı süzgeci ayarlama*

Ayırt edici adı bir kullanıcı kimliğiyle eşleyen bir sertifika adı süzgeci (CNF) tanımlamak için RACDCERT komutunu kullanın.

Bir CNF oluşturmak için aşağıdaki adımları gerçekleştirin.

1. Aşağıdaki komutu kullanarak CNF işlevlerini etkinleştirin. Bunu yapmak için, DIGTNMAP sınıfında güncelleme yetkinizin olması gerekir.

   ```
   SETROPTS CLASSACT(DIGTNMAP) RACLIST(DIGTNMAP)
   ```

2. CNF ' yi tanımlayın. Örneğin:

   ```
   RACDCERT ID(USER1) MAP WITHLABEL('filter1') TRUST
   SDNFILTER('O=IBM.C=UK') IDNFILTER('O=ExampleCA.L=Internet')
   ```

   Burada USER1 , aşağıdaki durumlarda kullanılacak kullanıcı kimliğidir:

   - Öznenin ayırt edici adı, bir IBM Kuruluşuna ve bir UKülkesine sahiptir.
   - Sertifika verenin DN 'si ExampleCA Kuruluşuna ve Internet' un İlçe 'sine sahiptir.

3. CNF eşlemelerini yenile:

   ```
   SETROPTS RACLIST(DIGTNMAP) REFRESH
   ```

Not:

1. Gerçek sertifika RACF veritabanında saklandıysa, kurulu olduğu kullanıcı kimliği, herhangi bir CNF ile ilişkili kullanıcı kimliği için değil, kullanılır. Sertifika RACF veritabanında saklanmazsa, en belirli eşleşen CNF ile ilişkili kullanıcı kimliği kullanılır. İlgili DN 'nin eşleşmeleri, sertifika veren DN' nin eşleşmesinden daha spesifik olarak kabul edilir.
2. CNF eşlemeleri yenileninceye kadar CNF ' lerde yapılan değişiklikler uygulanmaz.
3. DN, CNF 'deki DN süzgeciyle ancak DN süzgeci, DN' nin *en az anlamlı kısmı* ile aynıysa eşleşir. DN 'nin en az önemli kısmı, genellikle DN' nin en sağında listelenen, ancak sertifikanın başında görünen özniteliklerden oluşur.

Örneğin, SDNFILTER `'O=IBM.C=UK'` değerini göz önünde bulundurun. `'CN=QM1.O=IBM.C=UK'` özne DN 'si bu süzgeçle eşleşiyor, ancak `'CN=QM1.O=IBM.L=Hursley.C=UK'` özne DN 'si bu süzgeçle eşleşmiyor.

Bazı sertifikaların en az önemli kısmı, DN süzgeciyle eşleşmeyen alanlar içerebilir. DEFINE CHANNEL komutundaki SSLPEER örüntüsünde bir DN örüntüsü belirterek bu sertifikaları dışlayabilirsiniz.

4. En özel eşleşen CNF, RACF için NOTRUST olarak tanımlanırsa varlık, kanal başlatıcısının altında çalıştığı kullanıcı kimliğini kullanır.
5. RACF , ayırıcı olarak `'.'` karakterini kullanır. IBM MQ , virgül ya da noktalı virgül kullanır.

Varlığın kanal kullanıcı kimliğini hiçbir zaman varsayılan değere (kanal başlatıcısının çalıştığı kullanıcı kimliği) ayarlamadığından emin olmak için CNF ' leri tanımlayabilirsiniz. Varlıkla ilişkili anahtarlık halkasındaki her CA sertifikası için, o CA sertifikasının konu DN 'si ile tam olarak eşleşen bir IDNFILTER ile bir CNF tanımlayın. Bu, varlığın kullanabileceği tüm sertifikaların bu CNF ' lerden en az biriyle eşleşmesini sağlar. Bunun nedeni, tüm bu sertifikaların varlıkla ilişkili anahtar halkasına bağlanması ya da bir sertifikanın varlıkla ilişkili anahtar halkasına bağlı olduğu bir CA tarafından verilmesi gerekolmasıdır.

CNF ' leri işlemek için kullandığınız komutlarla ilgili daha fazla bilgi için *z/OS Security Server RACF Security Administrator's Guide* adlı yayına bakın.

z/OS *z/OS üzerinde QMA ' da bir gönderen kanalı ve iletim kuyruğunun tanımlanması*

Gerekli nesneleri ayarlamak için **DEFINE CHANNEL** ve **DEFINE QLOCAL** komutlarını kullanın.

Yordam

QMA ' da aşağıdaki örnek gibi komutlar yayınlayın:

```
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)
```

Sonuçlar

Bir gönderici kanalı, TO.QMBve QMB iletim kuyruğu oluşturulur.

z/OS *z/OS üzerinde QMB ' de bir alıcı kanalı tanımlama*

Gerekli nesneyi ayarlamak için **DEFINE CHANNEL** komutunu kullanın.

Yordam

QMB ' de aşağıdaki örnek gibi bir komut verin:

```
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')
```

Sonuçlar

Bir alıcı kanal, TO.QMBoluşturuldu.

z/OS *z/OS üzerinde QMA ' da gönderen kanalının başlatılması*

Gerekiyorsa, bir dinleyici programı başlatın ve güvenliği yenileyin. Daha sonra, **START CHANNEL** komutunu kullanarak kanalı başlatın.

Yordam

1. İsteğe bağlı: Henüz yapmadıysanız, QMB ' de bir dinleyici programı başlatın.

Dinleyici programı gelen ağ isteklerini dinler ve gerektiğinde alıcı kanalı başlatır. Bir dinleyicinin nasıl başlatılacağına ilişkin bilgi için Kanal dinleyicisinin başlatılmasıbaşlıklı konuya bakın.

2. İsteğe bağlı: Daha önce SSL/TLS kanalları çalıştırıldıysa, `REFRESH SECURITY TYPE(SSL)` komutunu verin.

   Bu, anahtar havuzunda yapılan tüm değişikliklerin kullanılabilir olmasını sağlar.
3. `START CHANNEL(TO.QMB)` komutunu kullanarak QMA ' da kanalı başlatın.

Sonuçlar

Gönderen kanal başlatıldı.

z/OS *z/OS üzerinde kendinden onaylı sertifikaların değiştirilmesi*

Önceden çıkardığınız sertifikaları değiştirin. FTP kullanıyorsanız, doğru biçimi kullanın.

Yordam

QM1 sertifikasının CA bölümünü QM2 sistemine ve tersi durumda FTP ile aktarın.

Sertifikaları FTP kullanarak aktardıysanız, doğru biçimde aktarmanız gerekir.

Aşağıdaki sertifika tiplerini *binary* biçiminde aktarın:

- DER kodlu ikili X.509
- PKCS #7 (CA sertifikaları)
- PKCS #12 (kişisel sertifikalar)

Aşağıdaki sertifika tiplerini ASCII biçiminde aktarın:

- PEM (gizlilik-gelişmiş posta)
- Base64 kodlanmış X.509

z/OS *z/OS üzerinde QM1 üzerinde bir gönderen kanalı ve iletim kuyruğunun tanımlanması*

Gerekli nesneleri ayarlamak için **`DEFINE CHANNEL`** ve **`DEFINE QLOCAL`** komutlarını kullanın.

Yordam

QM1' de, aşağıdaki örnek gibi komutlar verin:

```
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)
```

Kanalın her bir ucundaki CipherSpecs aynı olmalıdır.

Kanalınızın TLS kullanmasını istiyorsanız yalnızca SSLCIPH parametresi zorunludur. SSLCIPH parametresi için izin verilen değerlerle ilgili bilgi için bkz. “IBM MQ içinde CipherSpecs (Şifre Belirtimleri) ve CipherSuites (CipherSuites)” sayfa 41 .

Sonuçlar

Bir gönderen kanal, QM1.TO.QM2ve bir iletim kuyruğu (QM2) oluşturulur.

z/OS *z/OS üzerinde QM2 üzerinde alıcı kanalı tanımlanması*

Gerekli nesneyi ayarlamak için **`DEFINE CHANNEL`** komutunu kullanın.

Yordam

QM2' de, aşağıdaki örnek gibi bir komut verin:

```
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')
```

Kanal, “z/OS üzerinde QM1 üzerinde bir gönderen kanalı ve iletim kuyruğunun tanımlanması” sayfa 336içinde tanımladığınız gönderen kanalla aynı ada sahip olmalı ve aynı CipherSpecseçeneğini kullanmalıdır.

z/OS *z/OS üzerinde QM1 üzerinde gönderen kanalının başlatılması*

Gerekiyorsa, bir dinleyici programı başlatın ve güvenliği yenileyin. Daha sonra, **START CHANNEL** komutunu kullanarak kanalı başlatın.

Yordam

1. İsteğe bağlı: Henüz başlatmadıysanız, QM2üzerinde bir dinleyici programı başlatın.

   Dinleyici programı gelen ağ isteklerini dinler ve gerektiğinde alıcı kanalı başlatır. Bir dinleyicinin nasıl başlatılacağına ilişkin bilgi için Kanal dinleyicisinin başlatılması başlıklı konuya bakın.
2. İsteğe bağlı: Daha önce SSL/TLS kanalları çalıştırıldıysa, REFRESH SECURITY TYPE (SSL) komutunu verin.

   Bu, anahtar havuzunda yapılan tüm değişikliklerin kullanılabilir olmasını sağlar.
3. QM1' de `START CHANNEL(QM1.TO.QM2)` komutunu kullanarak kanalı başlatın.

Sonuçlar

Gönderen kanal başlatıldı.

z/OS *z/OS üzerinde SSL ya da TLS ortamı yenileniyor*

REFRESH SECURITY komutunu kullanarak kuyruk yöneticisi QMA ' daki TLS ortamını yenileyin.

Yordam

QMA ' da aşağıdaki komutu girin:

```
REFRESH SECURITY TYPE(SSL)
```

Bu, anahtar havuzunda yapılan tüm değişikliklerin kullanılabilir olmasını sağlar.

z/OS *z/OS üzerinde bir alıcı kanalında anonim bağlantılara izin verme*

SSL ya da TLS istemci kimlik doğrulamasını isteğe bağlı yapmak için **ALTER CHANNEL** komutunu kullanın.

Yordam

QMB ' de şu komutu girin:

```
ALTER CHANNEL(TO.QMB) CHLTYPE(RCVR) SSLCAUTH(OPTIONAL)
```

z/OS *z/OS üzerinde QM1 üzerinde gönderen kanalının başlatılması*

Gerekiyorsa, kanal başlatıcıyı başlatın, bir dinleyici programı başlatın ve güvenliği yenileyin. Daha sonra, **START CHANNEL** komutunu kullanarak kanalı başlatın.

Yordam

1. İsteğe bağlı: Henüz başlatmadıysanız, kanal başlatıcıyı başlatın.
2. İsteğe bağlı: Henüz başlatmadıysanız, QM2üzerinde bir dinleyici programı başlatın.

   Dinleyici programı gelen ağ isteklerini dinler ve gerektiğinde alıcı kanalı başlatır. Bir dinleyicinin nasıl başlatılacağına ilişkin bilgi için Kanal dinleyicisinin başlatılması başlıklı konuya bakın.
3. İsteğe bağlı: Kanal başlatıcısı zaten çalışıyorsa ya da SSL/TLS kanalları önceden çalıştırıldıysa, REFRESH SECURITY TYPE (SSL) komutunu verin.

   Bu, anahtar havuzunda yapılan tüm değişikliklerin kullanılabilir olmasını sağlar.
4. QM1' de `START CHANNEL(QM1.TO.QM2)` komutunu kullanarak kanalı başlatın.

Sonuçlar

Gönderen kanal başlatıldı.

z/OS *z/OS üzerinde QMA ' da gönderen kanalının başlatılması*

Gerekiyorsa, kanal başlatıcıyı başlatın, bir dinleyici programı başlatın ve güvenliği yenileyin. Daha sonra, **START CHANNEL** komutunu kullanarak kanalı başlatın.

Yordam

1. İsteğe bağlı: Henüz başlatmadıysanız, kanal başlatıcıyı başlatın.
2. İsteğe bağlı: Henüz yapmadıysanız, QMB ' de bir dinleyici programı başlatın.

   Dinleyici programı gelen ağ isteklerini dinler ve gerektiğinde alıcı kanalı başlatır. Bir dinleyicinin nasıl başlatılacağına ilişkin bilgi için Kanal dinleyicisinin başlatılmasıbaşlıklı konuya bakın.
3. İsteğe bağlı: Kanal başlatıcı zaten çalışıyorsa ya da SSL/TLS kanalları önceden çalıştırıldıysa, REFRESH `SECURITY TYPE(SSL)` komutunu verin.

   Bu, anahtar havuzunda yapılan tüm değişikliklerin kullanılabilir olmasını sağlar.
4. `START CHANNEL(TO.QMB)` komutunu kullanarak QMA ' da kanalı başlatın.

Sonuçlar

Gönderen kanal başlatıldı.

z/OS *z/OS üzerinde eliptik eğri anahtarı uzunluğunun değiştirilmesi*

GSK_CLIENT_ECURVE_LIST ortam değişkenini değiştirme, istemci tarafından belirtilen eliptik eğriler ya da desteklenen grupların listesini, kullanım tercihi sırasına göre bir ya da daha fazla 4 karakterlik değerden oluşan bir dizgi olarak ayarlama.

Önemli: TLS 1.0, TLS 1.1 ve/veya TLS 1.2 anlaşmalı bağlantıları kullanırken belirli eliptik eğrileri işletim sistemi tarafından yürürlüğe girmesine izin vermek için z/OS APAR OA61783 içindeki düzeltmeyi uygulamanız gerekir.

Bu TLS ortam değişkenini, CEEOPTS DD deyimini kullanarak kanal başlatıcı başlatma JCL ' de ayarlayabilirsiniz:

```
CEEOPTS DD DSN=<dataset-name>,DISP=SHR
```

Yukarıda başvurulan veri kümesinde kullanmak istediğiniz listeyi belirtin; örneğin:

```
ENVAR("GSK_CLIENT_ECURVE_LIST=002300240025")
```

Önemli: Bu CEEOPTS deyimini akış içindeki verilerle kullanmayın; bu, ortam değişkeninin o deyimi kullanan tüm TLS görevleri için ayarlanmasını önler.

Birden büyük bir SSLTASKS değeri kullanırken bunun çalışmasına izin vermek için sıralı bir veri kümesine ya da bölümlenmiş veri kümesi üyesine başvurduğunuzdan emin olun.

GSK_SERVER_ALLOWED_KEX_ECURVES olan GSK_CLIENT_ECURVE_LIST ' in sunucu analog eşdeğerini de kullanabilirsiniz. Ek bilgi için Anahtar değiştokuşu eliptik eğrileri sınırlama başlıklı konuya bakın.

Ayrıca, geçerli 4 karakterlik eliptik eğri ve desteklenen grup belirtimlerinin bir listesi için Şifre takımı tanımlamaları içindeki Tablo 5 'e bakın.

Varsayılan belirtim `00210023002400250019`' dir. TLS V1.3 etkinleştirildiyse, `0029 (x25519)` varsayılan listenin sonuna eklenir.

Kullanıcıların tanımlanması ve kimliğinin doğrulanmesi

X.509 sertifikalarını, MQCSP yapısını ya da çeşitli kullanıcı çıkış programı tiplerini kullanarak kullanıcıları tanımlayabilir ve kimliklerini doğrulayabilirsiniz.

X.509 sertifikalarının kullanılması

SET CHLAUTH komutu ve **SSLPEER** parametresiyle X.509 sertifikalarını kullanarak kullanıcıları tanımlayabilir ve kimliklerini doğrulayabilirsiniz. **SSLPEER** parametresi, kanalın diğer ucundaki eş kuyruk yöneticisinden ya da istemciden alınan sertifikanın Konu Ayırt Edici Adıyla karşılaştırmak için kullanılacak süzgeci belirtir.

SET CHLAUTH komutunu ve **SSLPEER** parametresini kullanma hakkında daha fazla bilgi için bkz. SET CHLAUTH.

Sayısal sertifikalar Sertifika Yetkilileri tarafından iptal edilebilir. Platforma bağlı olarak, OCSP ya da LDAP sunucularındaki CRL ' leri kullanarak sertifikaların iptal durumunu denetleyebilirsiniz. Daha fazla bilgi için bkz “İptal edilen sertifikalarla çalışma” sayfa 355.

MQCSP yapısının kullanılması

MQCONNX çağrısında MQCSP bağlantı güvenliği değiştirgeleri yapısı belirtildi. Bu yapı, uygulama tarafından sağlanan kimlik bilgilerini içerebilir. Uygulama, MQCSP yapısında bir kullanıcı kimliği ve parola sağlayabilir. IBM MQ 9.3.4olanağından, uygulamalar bir kimlik doğrulama simgesi de sağlayabilir. Gerekiyorsa, MQCSP bir güvenlik çıkışında değiştirilebilir.

Uyarı: MQCSP yapısındaki kimlik bilgileri bazen ağ üzerinden düz metin olarak gönderilir. İstemci uygulaması kimlik bilgilerinin korunduğundan emin olmak için bkz. “MQCSP parola koruması” sayfa 31.

Daha fazla bilgi için bkz. “MQCSP yapısını kullanarak kullanıcıların belirlenmesi ve doğrulanmaları” sayfa 341 ve “Kimlik doğrulama belirteçleriyle çalışma” sayfa 344.

Linux AIX AIX ve Linuxsistemlerinde, MQCSP yapısında belirtilen kullanıcı kimliği ve parola, işletim sistemi ya da PAM (Pluggable Authentication Method; Takılabilir Kimlik Doğrulama Yöntemi) kullanılarak doğrulanabilir. PAM, hizmetlerden ayrıntıları gizleyen kullanıcı kimlik doğrulaması için genel bir mekanizma sağlar. Daha fazla bilgi için bkz “Takılabilir Kimlik Doğrulama Yönteminin (PAM) Kullanılması” sayfa 366.

Çıkışlarda kimlik belirleme ve kimlik doğrulama uygulanıyor

Birkaç tip kullanıcı çıkış programını kullanarak kullanıcıları tanımlayabilir ve kimliklerini doğrulayabilirsiniz. Daha fazla bilgi için bkz. “Güvenlik çıkışlarında kimlik belirleme ve kimlik doğrulama uygulanması” sayfa 342, “İleti çıkışlarında kimlik eşleme” sayfa 343ve “API çıkışında ve API geçişi çıkışında kimlik eşlemesi” sayfa 343.

Ayrıcalıklı kullanıcılar

Ayrıcalıklı kullanıcı, IBM MQiçin tam yönetici yetkilerine sahip kullanıcıdır.

Aşağıdaki tabloda listelenen kullanıcılara ek olarak, kuyruk yöneticisinin bütünlüğünü ve güvenliğini sağlamak için erişim verilirken ek özen gösterilmesi gereken belirli nesneler ve yetkiler vardır. Aşağıdaki yetkilerden herhangi biri verilirken ek inceleme uygulanmalıdır:

- SYSTEM nesneleri için herhangi bir yetki
- Nesneleri yaratmak, değiştirmek ve silmek için denetim yetkileri.

  z/OS z/OSüzerinde bu yetki, DEFINE, ALTER ve DELETE komutlarını yayınlamak için komut güvenliği ve komut kaynağı güvenliği yetkisidir.

  Multi Diğer tüm platformlarda bu yetkiler, `+crt`, `+chg` ve `+dlt`gibi yönetim yetkileridir.
- Kuyrukları temizlemek için denetim yetkisi.

  z/OS z/OSüzerinde bu yetki, CLEAR komutlarını yayınlamak için komut güvenliği ve komut kaynak güvenliği yetkisidir.

  Multi Diğer tüm platformlarda bu yetki `+clr`' dir.
- Kanalları durdurmak, iletileri geri almak ya da kesinleştirmek için yönetim yetkileri.

  z/OS z/OSüzerinde bu yetki, RESET CHANNEL, START CHANNEL ve STOP CHANNEL gibi komutları yayınlamak için komut güvenliği ve komut kaynağı güvenliği yetkisidir.

  Multi Diğer tüm platformlarda bu yetkiler `+ctrl` ve `+ctrlx`' dir.
- Uygulamaların yetkilendirme denetimleri için ayrıcalıkları yükseltmesine olanak sağlayan alternatif kullanıcı MQI yetkilendirmesi.

  z/OS z/OSüzerinde bu yetki, diğer kullanıcı güvenlik tanıtımlarına verilen herhangi bir yetkilidir.

  Multi Diğer tüm platformlarda bu yetki `+altusr`' dir.
- Uygulamaların iletilerin güvenlik bağlamını değiştirmesini sağlayan bağlam yetkileri.

  z/OS z/OSüzerinde bu yetki, bağlam güvenliği tanıtımlarına verilen herhangi bir yetkilidir.

  Multi Diğer tüm platformlarda bu yetkiler `+setall` ve `+setid`' dir.

Genel bir birincil kullanıcı olarak, ileti sistemi uygulamalarına yalnızca gereken kuyruklar ya da konular için temel MQI yetkileri verilmelidir. Ayrıcalıklı olmayan bir MCAUSER altında çalışan MCA kanalları ve ölü mektup kuyruğu işleyicileri gibi diğer bazı özel uygulama türleri, normalde uygulamalara doğru çalışması için verilmemiş ek yetkiler gerektirebilir.

Çizelge 67. Platforma göre ayrıcalıklı kullanıcılar

Hizmet olarak sunulan	**Ayrıcalıklı kullanıcılar**
Windows sistemleri	• SYSTEM • mqm grubunun üyeleri • Denetimciler (Administrators) grubunun üyeleri
AIX and Linux sistemleri	• mqm grubunun üyeleri
IBM i IBM i IBM i sistemleri	• Profiller qmqm ve qmqmadm • qmqmadm grubunun tüm üyeleri • *ALLOBJ ayarıyla tanımlanan herhangi bir kullanıcı

Çizelge 67. Platforma göre ayrıcalıklı kullanıcılar (devamı var)	
Hizmet olarak sunulan	**Ayrıcalıklı kullanıcılar**
z/OS	Kanal başlatıcısı, kuyruk yöneticisi ve gelişmiş ileti güvenliği adres alanlarının altında çalıştığı kullanıcı kimliği. Bu kullanıcı kimlikleri, IBM MQiçin otomatik olarak tam yönetici yetkisine sahip değildir, ancak genellikle bu kullanıcı kimliklerine verilen erişim düzeyi nedeniyle ayrıcalıklı olarak kabul edilir.

MQCSP yapısını kullanarak kullanıcıların belirlenmesi ve doğrulanmaları

MQCONNX çağrısında MQCSP bağlantı güvenliği değiştirgeleri yapısını belirtebilirsiniz. MQCSP yapısı, kimlik doğrulaması için kullanılan kimlik bilgilerini denetlemek için ileti kuyruğu arabirimini (MQI) kullanan uygulamalar için birincil yoldur.

MQCSP yapısı, yetkilendirme hizmetinin kullanıcıyı tanımlamak ve doğrulamak için kullanabileceği kimlik bilgileriiçerir.

MQCSP yapısı, uygulama belirtik olarak MQCSP yapısını sağlamasa da, istemci ya da sunucu tarafı güvenlik çıkışları tarafından değiştirilebilir. Belirtik olarak MQCSP yapısı sağlamayan bir uygulama örneği, IBM MQ classes for JMSkullanan bir uygulamadır. MQCSP yapısına bir kullanıcı kimliği ve parola ekleyen istemci tarafı güvenlik çıkışı örneği için bkz. “Kullanıcı kimliği ve parola eklemek için istemci tarafı güvenlik çıkışı (mqccred)” sayfa 79.

V 9.3.4 MQCSP yapısı bir kullanıcı kimliği ve parola ya da kimlik doğrulama simgesi içeriyor. MQCSP yapısında sağlanan kimlik bilgileri için aşağıdaki kısıtlamalar geçerlidir:

- Bir uygulama ya da çıkış, bir kullanıcı kimliği ve parola ya da bir kimlik doğrulama simgesi sağlamalıdır, ancak her ikisini birden sağlamamalıdır.
- IBM MQ' e erişmek için yalnızca belirli biçimleri ve gereksinimleri karşılayan kimlik doğrulama belirteçleri kullanılabilir. IBM MQiçinde kimlik doğrulama belirteçlerine ilişkin gereksinimler hakkında daha fazla bilgi için bkz. “Kimlik doğrulama belirteçlerine ilişkin gereksinimler” sayfa 346.
- Kimlik doğrulama simgesindeki kimlik uygulama bağlamı olarak benimsenecek olursa, belirteç uygun bir kullanıcı talebi sağlamalı ve talep değeri geçerli bir IBM MQ kullanıcı kimliği olmalıdır. Örneğin, kullanıcı adı uzunluk üst sınırı ve özel karakter kısıtlamalarına uygun olmalıdır. Bir kullanıcı kimliğini benimseme hakkında daha fazla bilgi için bkz. “MQCSP ve ADOPTCTX ayarları arasındaki ilişki” sayfa 341.

MQCSP yapısıyla ilgili daha fazla bilgi için bakınız: MQCSP-Security parameters.

Uyarı: Bir istemci uygulamasına ilişkin MQCSP yapısındaki kimlik bilgileri bazen ağ üzerinden düz metin olarak gönderilir. İstemci uygulaması kimlik bilgilerinin korunduğundan emin olmak için bkz. “MQCSP parola koruması” sayfa 31.

MQCSP ve ADOPTCTX ayarları arasındaki ilişki

IBM MQ , bağlantı kimlik doğrulama özelliği etkinleştirildiyse, MQCSP yapısında geçirilen kimlik bilgilerini her zaman doğrular. Kimlik bilgileri başarıyla doğrulandıktan sonra IBM MQ , bağlı uygulama tarafından gerçekleştirilen işlemlerle ilgili sonraki yetkilendirme denetimleri için kullanıcı kimliğini kabul edebilir. Kuyruk yöneticisinin **CONNAUTH** özniteliği tarafından başvurulan kimlik doğrulama bilgileri (AUTHINFO) nesnesi **ADOPTCTX(YES)** ile tanımlandıysa, MQCSP kimlik bilgilerindeki kullanıcı kimliği benimsenir.

IBM MQ yetki denetimlerinde kullanabileceği kullanıcı kimliklerinin uzunluğuna ilişkin bir sınırlama vardır. Bu sınırlar hakkında daha fazla bilgi için bkz. “Kullanıcı Kimlikleri” sayfa 87. MQCSP yapısında geçirilen bir kullanıcı kimliği benimsendiğinde, IBM MQ diğer yapılandırma seçeneklerine bağlı olarak farklı davranır:

- LDAP bağlantısı kimlik doğrulamasını kullanırken IBM MQ , kullanıcının LDAP kaydının kısa kullanıcı adı özniteliğindeki kullanıcı kimliğini benimser. Kısa kullanıcı adı özniteliği, AUTHINFO nesnesinin **SHORTUSR** özniteliği kullanılarak ayarlanır.

Örneğin, **SHORTUSR** `'CN'` olarak ayarlanırsa ve LDAP kaydı kullanıcıyı `'CN=Test,SN=MQ,O=IBM,C=UK'` olarak listelerse, `Test` kullanıcı kimliği kullanılır.

- İşletim sistemi bağlantısı kimlik doğrulaması ya da PAM kimlik doğrulaması kullanılırken, ADOPTCTX YES ise, MQCSP yapısına geçirilen kullanıcı kimliği, bağlantı bağlamı olarak benimsendiğinde 12 karakterlik IBM MQ kullanıcı kimliği sınırını karşılamak için kesilir.

  ChlAuthEarlyAdopt etkinleştirilirse, kullanıcı kimlik bilgileri doğrulandıktan sonra kesme gerçekleşir.

  ChlAuthEarlyAdopt etkinleştirilmezse, kısaltma benimsemeden önce gerçekleşir. Windows' da, kullanıcı `user@domain`biçiminde sağlanırsa, bu, kullanıcı 12 karakterden kısa olduğunda geçerli olmayan bir etki alanı belirtimine neden olabilir anlamına gelir.

  Örneğin, bir kullanıcı `` `ibmmq@windowsdomain` `` MQCSP aracılığıyla sağlanırsa, bu kullanıcı bu senaryoda `` `ibmmq@window` `` olarak kesilir. Bu, aşağıdaki hatayla sonuçlanır:

  ```
  AMQ8074W: 'SID' varlığı 'ibmmq@window' ile eşleşmediğinden yetkilendirme başarısız oldu
  ```

  Bu temelde, Windows formdaki etki alanı kullanıcı kimliği `user@domain`gibi 12 karakterden uzun bir kullanıcı kimliğini MQCSP aracılığıyla geçirirseniz, bu hatayı önlemek için qm.ini dosyasındaki **ChlAuthEarlyAdopt**=*Y* değerini yapılandırmanız gerekir.

  Alternatif olarak, CONNAUTH AUTHINFO yapılandırmasında ADOPTCTX (NO) kullanın ve kanala ilişkin kullanıcı kimliğini ayarlamak için CHLAUTH USERMAP kuralı, güvenlik çıkışı ya da kanal nesnesi MCAUSER ayarı gibi alternatif bir yaklaşım kullanın.

Güvenlik çıkışlarında kimlik belirleme ve kimlik doğrulama uygulanması

Tek yönlü ya da karşılıklı kimlik doğrulaması gerçekleştirmek için bir güvenlik çıkışı kullanabilirsiniz.

Bir güvenlik çıkışının birincil amacı, bir kanalın her ucunda MCA ' nın ortağının kimliğini doğrulamasını sağlamaktır. Bir ileti kanalının her bir ucunda ve bir MQI kanalının sunucu ucunda, MCA genellikle bağlı olduğu kuyruk yöneticisi adına hareket eder. Bir MQI kanalının istemci ucunda, bir MCA genellikle IBM MQ MQI client uygulamasının kullanıcısı adına hareket eder. Bu durumda, karşılıklı kimlik doğrulaması iki kuyruk yöneticisi arasında ya da bir kuyruk yöneticisi ile bir IBM MQ MQI client uygulamasının kullanıcısı arasında gerçekleşir.

Sağlanan güvenlik çıkışı (SSPI kanal çıkışı), Kerberosgibi güvenilir bir kimlik doğrulama sunucusu tarafından oluşturulan ve daha sonra denetlenen kimlik doğrulama belirteçlerinin değiş tokuş edilmesiyle karşılıklı kimlik doğrulamasının nasıl gerçekleştirileceğini gösterir. Daha fazla ayrıntı için bkz. “Windows üzerinde SSPI kanal çıkış programı” sayfa 152.

Karşılıklı kimlik doğrulama, Public Key Infrastructure (PKI) teknolojisi kullanılarak da uygulanabilir. Her güvenlik çıkışı bazı rasgele veriler oluşturur, bu verileri temsil ettiği kuyruk yöneticisinin ya da kullanıcının özel anahtarını kullanarak imzalarlar ve imzalanmış verileri bir güvenlik iletisinde iş ortağına gönderir. Ortak güvenlik çıkışı, kuyruk yöneticisinin ya da kullanıcının genel anahtarını kullanarak dijital imzayı denetleyerek kimlik doğrulamasını gerçekleştirir. Dijital imzaları değiştirmeden önce, birden fazla algoritma kullanılabiliyorsa, güvenlik çıkışlarının bir ileti özeti oluşturmak için algoritmayı kabul etmesi gerekebilir.

Bir güvenlik çıkışı imzalı verileri iş ortağına gönderdiğinde, temsil ettiği kuyruk yöneticisini ya da kullanıcıyı tanımlamak için de bazı yollar göndermesi gerekir. Bu bir Ayırt Edici Ad ya da dijital sertifika olabilir. Bir sayısal sertifika gönderilirse, ortak güvenlik çıkışı, sertifika zincirinden kök sertifika kuruluşu (CA) sertifikasına kadar çalışarak sertifikayı doğrulayabilir. Bu, dijital imzayı denetlemek için kullanılan açık anahtarın sahipliğinin güvencesini sağlar.

İş ortağı güvenlik çıkışı, yalnızca sertifika zincirinde kalan sertifikaları içeren bir anahtar havuzuna erişimi varsa dijital sertifikayı doğrulayabilir. Kuyruk yöneticisi ya da kullanıcı için bir sayısal sertifika gönderilmezse, ortak güvenlik çıkışının erişimi olan anahtar havuzunda bir sertifika bulunmalıdır. İş ortağı güvenlik çıkışı, imzalayanın genel anahtarını bulamadığı sürece dijital imzayı denetleyemez.

Taşıma Katmanı Güvenliği (TLS), sadece açıklananlar gibi PKI tekniklerini kullanır. Güvenli Yuva Katmanı 'nın kimlik doğrulamasını nasıl gerçekleştirdiğine ilişkin ek bilgi için bkz. “Aktarım Katmanı Güvenliği (TLS) kavramları” sayfa 18.

Güvenilir bir kimlik doğrulama sunucusu ya da PKI desteği yoksa, başka teknikler de kullanılabilir. Güvenlik çıkışlarında uygulanabilen yaygın bir teknik, bir simetrik anahtar algoritması kullanır.

Güvenlik çıkışlarından biri, A çıkışı, rasgele bir numara oluşturur ve bunu bir güvenlik iletisiyle ortak güvenlik çıkışına gönderir, B çıkışı. Çıkış B, yalnızca iki güvenlik çıkışında bilinen bir anahtarın kopyasını kullanarak numarayı şifreler. B çıkışı, B çıkışının oluşturduğu ikinci bir rasgele numarayla A ' dan çıkmak için şifrelenmiş numarayı gönderir. Çıkış A, ilk rasgele sayının doğru şekilde şifrelendiğini doğrular, anahtarın kopyasını kullanarak ikinci rasgele sayıyı şifreler ve şifrelenmiş numarayı bir güvenlik iletisinde B ' den çıkmak için gönderir. B çıkışından sonra, ikinci rasgele sayının doğru şekilde şifrelendiğini doğrular. Bu değişim sırasında, iki güvenlik çıkışından biri diğerinin gerçekliğinden memnun değilse, MCA ' ya kanalı kapatması için talimat verebilir.

Bu tekniğin bir yararı, değiş tokuş sırasında iletişim bağlantısı üzerinden hiçbir anahtar ya da parola gönderilmez. Dezavantajı, paylaşılan anahtarın güvenli bir şekilde nasıl dağıtılacağına ilişkin soruna çözüm sağlamamaktır. Bu soruna ilişkin bir çözüm “Kullanıcı çıkış programlarında gizlilik uygulanması” sayfa 480içinde açıklanmıştır. Benzer bir teknik, bir oturum oluşturmak üzere bağlandıklarında iki LU 'nun karşılıklı kimlik doğrulaması için SNA' da kullanılır. Teknik “Oturum düzeyi kimlik doğrulaması” sayfa 120içinde açıklanmıştır.

Karşılıklı kimlik doğrulama için önceki tüm teknikler tek yönlü kimlik doğrulaması sağlayacak şekilde uyarlanabilir.

İleti çıkışlarında kimlik eşleme

Bir kullanıcı kimliğini doğrulamak üzere bilgileri işlemek için ileti çıkışlarını kullanabilirsiniz, ancak uygulama düzeyinde kimlik doğrulamasını uygulamak daha iyi olabilir.

Bir uygulama bir iletiyi kuyruğa koyduğunda, ileti tanımlayıcısındaki *UserIdentifier* alanı, uygulamayla ilişkilendirilmiş bir kullanıcı kimliği içerir. Ancak, kullanıcı kimliğini doğrulamak için kullanılabilecek veri yok. Bu veriler, bir kanalın gönderme sonundaki bir ileti çıkışıyla eklenebilir ve kanalın alma ucundaki bir ileti çıkışı tarafından denetlenir. Kimlik doğrulayan veriler şifrelenmiş bir parola ya da dijital imza olabilir.

Bu hizmet, uygulama düzeyinde uygulanırsa daha etkili olabilir. Temel gereksinim, iletiyi alan kullanıcının iletiyi gönderen uygulamanın kullanıcısını tanımlayabilmesi ve kimliğini doğrulayabilmesi içindir. Bu nedenle, bu hizmeti uygulama düzeyinde uygulamayı düşünmek doğaldır. Daha fazla bilgi için “API çıkışında ve API geçişi çıkışında kimlik eşlemesi” sayfa 343 başlıklı konuya bakın.

API çıkışında ve API geçişi çıkışında kimlik eşlemesi

İleti alan bir uygulama, iletiyi gönderen uygulamanın kullanıcısını tanımlayıp kimliğini doğrulayabilmelidir. Bu hizmet genellikle uygulama düzeyinde en iyi şekilde uygulanır. API çıkışları hizmeti çeşitli şekillerde uygulayabilir.

Tek bir ileti düzeyinde, kimlik belirleme ve kimlik doğrulama, iki kullanıcıyı, iletiyi gönderen ve alıcıyı içeren bir hizmettir. Temel gereksinim, iletiyi alan kullanıcının iletiyi gönderen uygulamanın kullanıcısını tanımlayabilmesi ve kimliğini doğrulayabilmesi içindir. Bu gereksinimin iki yönlü değil, tek yönlü kimlik doğrulaması için olduğunu unutmayın.

Nasıl uygulandığına bağlı olarak, kullanıcıların ve uygulamalarının hizmetle arabirim kurması ya da hatta etkileşim kurması gerekebilir. Ayrıca, hizmetin ne zaman ve nasıl kullanılacağı, kullanıcıların ve uygulamalarının nerede bulunduğuna ve uygulamaların niteliğine bağlı olabilir. Bu nedenle, hizmeti bağlantı düzeyinde değil, uygulama düzeyinde uygulamayı düşünmek doğaldır.

Bu hizmeti bağlantı düzeyinde uygulamayı düşünüyorsanız, aşağıdakiler gibi sorunları çözmeniz gerekebilir:

- Bir ileti kanalında, hizmeti yalnızca bu iletiyi gerektiren iletilere nasıl uygulayabilirsiniz?
- Bu bir gereksinimse, kullanıcıların ve uygulamalarının hizmetle arabirim ya da etkileşim kurmasını nasıl sağlayabilirsiniz?
- Bir iletinin hedefe giderken birden çok ileti kanalı üzerinden gönderildiği çok sekmeli bir durumda, hizmetin bileşenlerini nerede çağırıyorsunuz?

Aşağıda, tanımlama ve kimlik doğrulama hizmetinin uygulama düzeyinde nasıl uygulanabileceğine ilişkin bazı örnekler verilmiştir. *API çıkışı* terimi, bir API çıkışı ya da API geçişi çıkışı anlamına gelir.

- Bir uygulama kuyruğa bir ileti yerleştirdiğinde, API çıkışı Kerberosgibi güvenilir bir kimlik doğrulama sunucusundan bir kimlik doğrulama belirteci alabilir. API çıkışı, iletideki uygulama verilerine bu belirteci ekleyebilir. İleti alan uygulama tarafından alındığında, ikinci bir API çıkışı kimlik doğrulama sunucusundan simgeyi denetleyerek gönderenin kimliğini doğrulamasını isteyebilir.
- Bir uygulama bir iletiyi kuyruğa koyduğunda, bir API çıkışı iletideki uygulama verilerine aşağıdaki öğeleri ekler:
  - Gönderenin dijital sertifikası
  - Gönderenin dijital imzası

  Bir ileti özeti oluşturmak için farklı algoritmalar kullanılabiliyorsa, API çıkışı kullandığı algoritmanın adını içerebilir.

  İleti alan uygulama tarafından alındığında, ikinci bir API çıkışı aşağıdaki denetimleri gerçekleştirebilir:
  - API çıkışı, sertifika zincirinden kök sertifika kuruluşu (CA) sertifikasına kadar çalışarak sayısal sertifikayı doğrulayabilir. Bunu yapmak için API çıkışının, sertifika zincirinde kalan sertifikaları içeren bir anahtar havuzuna erişimi olmalıdır. Bu denetim, Ayırt Edici Ad ile tanımlanan gönderenin, sertifikada bulunan genel anahtarın gerçek sahibi olduğuna dair güvence sağlar.
  - API çıkışı, sertifikada bulunan genel anahtarı kullanarak dijital imzayı denetleyebilir. Bu denetim, gönderenin kimliğini doğrular.

  Gönderenin ayırt edici adı, tüm dijital sertifika yerine gönderilebilir. Bu durumda, ikinci API çıkışının gönderenin genel anahtarını bulabilmesi için anahtar havuzunun gönderenin sertifikasını içermesi gerekir. Başka bir olasılık da sertifika zincirindeki tüm sertifikaları göndermektir.
- Bir uygulama bir iletiyi kuyruğa koyduğunda, ileti tanımlayıcısındaki *UserIdentifier* alanı, uygulamayla ilişkilendirilmiş bir kullanıcı kimliği içerir. Kullanıcı kimliği, göndereni tanımlamak için kullanılabilir. Kimlik doğrulamayı etkinleştirmek için, bir API çıkışı iletideki uygulama verilerine şifrelenmiş parola gibi bazı verileri ekleyebilirsiniz. İleti alan uygulama tarafından alındığında, ikinci bir API çıkışı, iletiyle birlikte gönderilen verileri kullanarak kullanıcı kimliğini doğrulayabilir.

  Bu teknik, denetimli ve güvenilir bir ortamdan kaynaklanan iletiler için ve güvenilir bir kimlik doğrulama sunucusu ya da PKI desteğinin kullanılamadığı durumlarda yeterli olarak kabul edilebilir.

Linux AIX V 9.3.4 Kimlik doğrulama belirteçleriyle çalışma

IBM MQ 9.3.4 istemci uygulamalarından, AIX ya da Linuxüzerinde çalışan bir kuyruk yöneticisiyle kimlik doğrulaması için belirteçler sağlayabilir. Simgedeki kullanıcı kimliği, IBM MQ kaynaklarına erişim yetkisi için de kullanılabilir.

JWT ' ler (JSON Web Simgeleri), taleplere dayalı bir kimlik modeli benimser. Kimlik ve erişim denetimi, talepler ve belirteç veren kişiler fikirlerine özdeştir.

- Talep, bir kullanıcı hakkında bilgi içeren ve kullanıcının kim olduğunu belirleyen bir ad değeri çiftidir, ne yapabileceğini değil.
- Belirteç veren, yalnızca kullanıcının kimliğine dayalı olarak bir kullanıcı için belirteç yayınlayan güvenilir bir üçüncü taraf ya da sunucudur. Belirteç yayıncısı, kullanıcının yapabilecekleriyle ilgilenmiyor.

Belirteç, talepleri içeren ve internet üzerinden taraflar arasında kolayca aktarılabilen basit bir yapıdır. Kimlik doğrulama için belirteçlerin kullanılması, merkezi kimlik yönetimi avantajına sahiptir. Uygulamalarınızın her bir hizmete ayrı olarak kaydolmadan birçok hizmette kimlik doğrulaması yapabilmesi için tek bir güvenilir belirteç yayıncısı kullanabilirsiniz. Kimlik bilgileri her hizmete değil, yalnızca güvenilir sertifika verene gönderildiğinden belirteçler daha fazla güvenlik sağlar.

JWT, önerilen İnternet standardı RFC7519aracılığıyla tanımlanır.

Belirteçler IBM MQ ile nasıl çalışır?

IBM MQ ile kullanılan belirteçler, IBM MQ tarafından desteklenen bir algoritmayla imzalanmış olan geçerli JWT ' ler olmalıdır. JWT, JSON Web İmzası (JWS) standardına göre imzalanmalıdır. JSON Web Encryption (JWE) ve JSON Web Key (JWK) JOSE teknolojilerini kullanan belirteçler IBM MQile birlikte kullanılamaz. Daha fazla bilgi için, bkz. “Kimlik doğrulama belirteçlerine ilişkin gereksinimler” sayfa 346.

Kimlik doğrulama belirtecini sağlayan uygulama, IBM MQ clients' i destekleyen herhangi bir platformda çalışabilir. Uygulamanın C 

ya da IBM MQ 9.3.5, Java, içinde yazılması ve istemci bağ tanımlarını kullanarak kuyruk yöneticisine bağlanması gerekir. Ancak, kuyruk yöneticisinin AIX ya da Linuxüzerinde çalışması gerekir. Kuyruk yöneticisi, kimlik doğrulama belirteçlerini kabul edecek şekilde yapılandırılmalıdır. Bir anahtar havuzu, simgeyi imzalamak için hangi algoritmanın kullanıldığına bağlı olarak, güvenilir belirteç verenin genel anahtar sertifikasını ya da simetrik anahtarını içermelidir.

Belirteç veren, yetkilendirilen güvenlik erişimine sahip güvenilir taraftır; bu, uygulama kullanıcısının kimliğini doğrular. Kuyruk yöneticisi, bir kimlik doğrulama simgesinin geçerli olup olmadığını ve kimliği doğrulanmış kullanıcının IBM MQ nesnelerine erişme yetkisinin olup olmadığını denetler. Kuyruk yöneticisi, bir simgeyle ilk kez bağlanmadan önce kullanıcıları bilebilir, ancak bilmesine gerek yoktur. IBM MQ yöneticisi, kuyruk yöneticisine bağlanan uygulamalar için kimlik doğrulama ve yetkilendirme ayarlamalı ve belirteçlerin ne içermesi gerektiğine ilişkin gereksinimleri ayarlamalıdır.

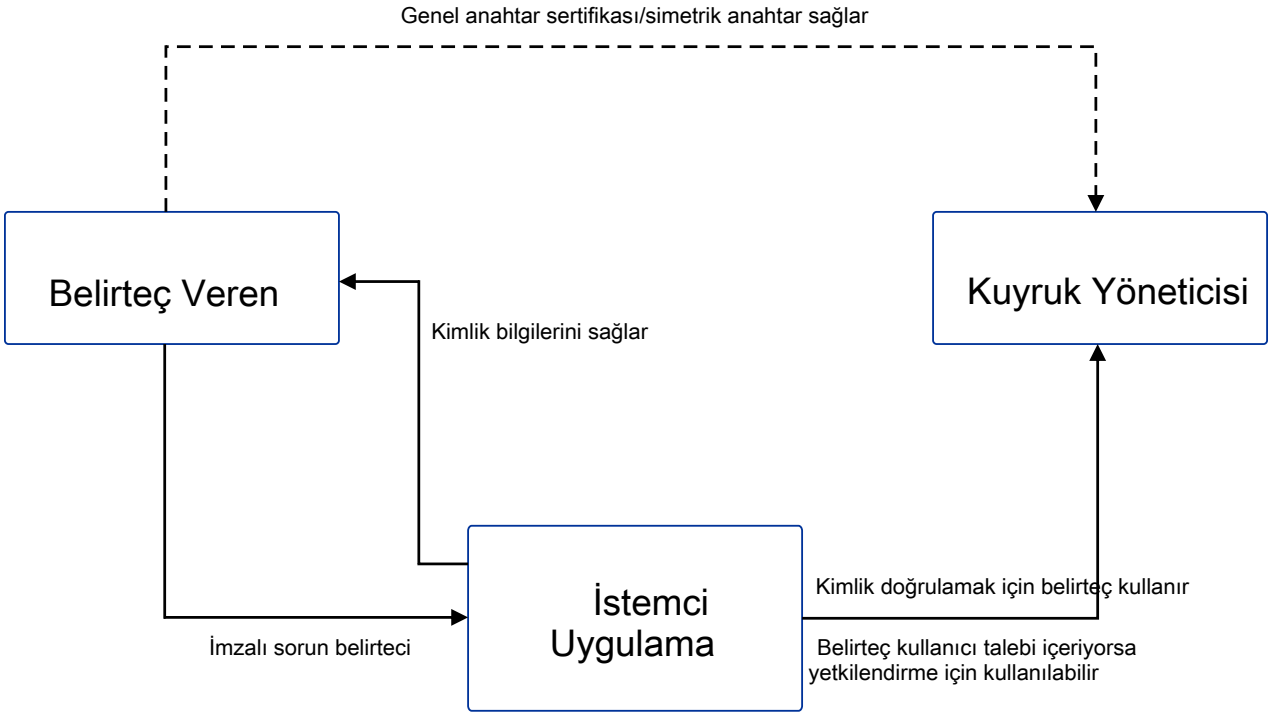
İstemci uygulaması, IBM MQ' e bağlandığında kimlik doğrulaması için kullandığı bir belirteci yayıncıdan dinamik olarak isteyebilir. Daha sonra uygulama, simgeyi bağlandığında kuyruk yöneticisine geçirmek için 

ya da seçilen API 'nin eşdeğeri olan IBM MQ 9.3.5' den MQCSP yapısını kullanır.

Uygulama, bir kimlik doğrulama belirteci istemek ve bağlandığında kuyruk sorumlusuna belirteç sunmak üzere değiştirilemezse, MQCSP yapısında bir belirteç sağlamak için alternatif olarak bir güvenlik çıkışı kullanılabilir.

Belirteç, kimlik doğrulama belirteçlerine ilişkin gereksinimleri karşılıyorsa ve belirteç imzası geçerliyse, bağlantı kurulur. Kuyruk yöneticisi, isteğe bağlı kullanıcı talebi simgede bulunuyorsa, IBM MQ kaynaklarına erişmek için yetki denetimi yapmak üzere simgede bulunan kullanıcı kimliğini de kullanabilir. Kullanıcı talebi, kuyruk yöneticisinin yetki denetimi için benimsediği kullanıcı kimliğini içeren belirteç içindeki taleptir. Kullanıcı talebinin bu adı, `qm.ini` dosyasının **`AuthToken`** kısmına ilişkin **`UserClaim`** özniteliğiyle belirtilir.

Daha fazla bilgi için bkz. “Uygulamada kimlik doğrulama belirteçlerini kullanma” sayfa 352 ve MQCSP-Security parameters.



Çizge, IBM MQile belirteç kullanımı için beklenen akışın temel bir örneğini gösterir. Beklenen yaşam çevrimi aşağıdaki gibidir:

- Belirteç, güvenilir veren tarafından bir uygulamaya verilir. Daha fazla bilgi için Kimlik doğrulama belirteçlerine ilişkin gereksinimlerbaşlıklı konuya bakın.
- Uygulama, bağlanırken simgeyi kuyruk yöneticisine iletir. Daha fazla bilgi için Uygulamada kimlik doğrulama belirteçlerini kullanmabaşlıklı konuya bakın.
- Kuyruk yöneticisi, anahtar havuzundaki güvenilir sertifika veren genel anahtarına ya da simetrik anahtara karşı belirteç imzasını doğrular. Kuyruk yöneticisini ayarlamak için Kimlik doğrulama belirteçlerini kabul edecek bir kuyruk yöneticisi yapılandırılmasıkonusundaki adımları izleyin.
- Kimlik doğrulama belirteci geçerli bir kullanıcı talebi içeriyorsa, belirteçteki kullanıcı, IBM MQ kaynaklarına erişmek için yetkilendirme denetimleri için benimsenebilir. Daha fazla bilgi için Yetkilendirme için kullanıcıları adoptingbaşlıklı konuya bakın.
- IBM MQ yöneticisi, güvenilir belirteç veren sertifikalarını yönetir. Sertifikanın süresi dolduğunda, belirteç veren kişiden yeni bir sertifika alınmalı ve anahtar havuzuna eklenmelidir.
- Kuyruk yöneticinizi yapılandırdıysanız ve uygulama bağlanıyorsa, ancak simgeyle ilgili sorunlarla karşılaşıyorsa, bkz. Kimlik doğrulama simgesi sorunlarının giderilmesi ve Simge kimlik doğrulama hata kodları.

IBM MQ , JWT ve JWS standartlarına uygun belirteçler sağlayan herhangi bir belirteçle çalışır.

Belirteç kullanmıyorsanız, ancak belirteç sunucusunun ayakta durmasında nelerin rol aldığınızı anlamak istiyorsanız, ücretsiz ve açık kaynak Keycloak projesiiçin Başlangıç Kılavuzu ' na bakın.

İlgili başvurular

`qm.ini` dosyasının AuthToken kısmı

Linux AIX V 9.3.4 Kimlik doğrulama belirteçlerine ilişkin gereksinimler

IBM MQile kullanılan kimlik doğrulama belirteçleri için doğrulama gereksinimleri, yapısı ve algoritmaları.

Gereksinimler

IBM MQ ile kullanılan kimlik doğrulama belirteçlerinin aşağıdaki gereksinimleri karşılaması gerekir.

- Simge uzunluğu 8192 karakter üst sınırını aşmamalıdır. Daha fazla bilgi için bakınız: TokenLength (MQLONG) for MQCSP.
- Belirteç yapısı ve kodlaması, RFC7519içindeki JSON Web Token (JWT) belirtimi ve RFC7515içindeki JSON Web Signature (JWS) belirtimi tarafından tanımlandığı şekilde geçerlidir.
- Çizelge 68 sayfa 348 içinde belirtilen gerekli simge üstbilgisi parametreleri var ve parametrelerin değerleri geçerli.
- Çizelge 69 sayfa 348 içinde belirtilen gerekli bilgi yükü talepleri var ve taleplerin değerleri geçerli.
- Simge, Çizelge 70 sayfa 348 içinde IBM MQ tarafından desteklenen bir algoritmayla imzalanmıştır.
- Süre bitimi (**exp**) talebinin değeri, geçerli zamandan sonra.
- Daha önce (**nbf**) talebi yoksa, değer geçerli zamandan önce olur.
- Bir kullanıcı talebi varsa, değerin “Kimlik doğrulama belirteçlerindeki kullanıcı kimlikleri” sayfa 349için gereksinimleri karşılaması gerekir.

Simge yapısı

IBM MQ , RFC7519 standardına uygun JWT ' leri kabul eder. JWT imzalanmalı ve RFC7515içinde tanımlanan JWS standardına göre kodlanmalıdır.

IBM MQ , JWS güvenli simgesinin aşağıdaki üç bileşeni içermesini bekler:

JOSE üstbilgisi

Simge tipini ve içeriğini korumak için kullanılan şifreleme algoritmalarını tanımlayan parametreleri içeren bir JSON nesnesi.

Aşağıdaki üstbilgi örneği, kodlanmış nesnenin bir JWT olduğunu ve üstbilginin ve bilgi yükünün HMAC SHA-256 algoritması kullanılarak güvenli kılındığını bildirir.

```
{
 "typ":"JWT",
 "alg":"HS256"
}
```

JWS bilgi yükü

JWT standardında belirtilen talepleri içeren bir JSON nesnesi. JSON nesnesinin her bir üyesi bir taleptir. Talepler, belirteç veren kişinin kimliğini ya da taşıyıcının kullanıcı kimliğini belirtebilir.

```
{
 "exp": 1685529153,
 "nbf": 1685528150,
 "AppUser": "MyUserName"
 }
```

JWS imzası

Belirtecin güvenilir bir sertifika veren tarafından verildiğini doğrulamak için kullanılır.

Bu bileşenler, JWS güvenli simgesinde nokta ('.') ile ayrılmış base64url-encoded dizeleri olarak gösterilir.

JWS standardına uyan bir kimlik doğrulama simgesi, simgenin gerçekliğinin doğrulanmasına izin vermek için imzalanır, ancak şifrelenmez. Bu nedenle, simgeye erişimi olan herkes tarafından okunabilir ve yeniden kullanılabilir. Kimlik doğrulamanın ağ üzerinden gönderildiğinde (örneğin, TLS kullanılarak) şifreleme kullanılarak korunduğundan emin olmak için kuyruk yöneticisiyle bağlantıyı yapılandırın. Bir uygulama tarafından sağlanan kimlik bilgilerini korumaya ilişkin seçenekler hakkında daha fazla bilgi için bkz. MQCSP parola koruması.

IBM MQ , üstbilgide ve kimlik doğrulama belirteçlerinin bilgi yükünde aşağıdaki parametreleri ve talepleri destekler. Bir simgedeki ek parametreler ya da talepler yoksayılır. Bir simge aynı ada sahip birden çok parametre ya da talep içeriyorsa, yinelenen ada sahip son parametre ya da talep kullanılır.

Çizelge 68. Belirteç üstbilgisi parametre açıklamaları				
Belirteç kısmı	**Parametre adı**	**Veri tipi**	**Zorunlu**	**Açıklama**
Üstbilgi	**typ**	Dizgi	Evet	Simge tipi. Bu değiştirgenin değeri "JWT" olmalıdır.
	alg	Dizgi	Evet	Üstbilgiyi ve bilgi yükünü korumak için kullanılan algoritma. Bu parametrenin değeri, Çizelge 70 sayfa 348içindeki algoritmalardan biri olmalıdır.

Çizelge 69. Belirteç bilgi yükü talepleri açıklamaları				
Belirteç kısmı	**Parametre adı**	**Veri tipi**	**Zorunlu**	**Açıklama**
Bilgi Yükü	**exp**	Tamsayı	Evet	1 Ocak 1979, 00:00 Eşgüdümlü Evrensel Saat 'ten bu yana saniye sayısı olarak ifade edilen belirteç süre sonu. Belirteç şu andan sonra kabul edilmez.
	nbf	Tamsayı	Hayır	1 Ocak 1979 'dan bu yana saniye sayısı olarak ifade edilen ve simgenin kabul edilmediği Eşgüdümlü Evrensel Saat 00:00 olarak ifade edilen süre.
	Kullanıcı talebi adı, `qm.ini` dosyasındaki **AuthToken** kısmına ilişkin **UserClaim** alanında belirtilir.	Dizgi	Yalnızca belirteçteki kullanıcı talebi yetkilendirme için kullanılıyorsa gereklidir.	Yetkilendirme denetimleri için benimsenen kullanıcı kimliğini içeren talebin adı. Örneğin, simgenin kullanıcı talebi varsa `"AppUser": "MyUserName"`, `qm.ini` dosyasının **AuthToken** kısmına **UserClaim**=`AppUser` belirtmeniz gerekir.

Kodlanmış ve kodu çözülen bir simgeye ilişkin iyi bir örnek için `jwt.io` web sitesindeki hata ayıklayıcı sayfasına bakın.

Algoritmalar

IBM MQ , JWS güvenli belirteçleri için JSON Web Algoritmaları (JWA) belirtiminde yer alan algoritmaların bir alt kümesini destekler.

Çizelge 70. JWS güvenli belirteçleri için IBM MQ tarafından desteklenen JSON Web Algoritmaları (JWA)	
alg değiştirge değeri	**Dijital İmza ya da MAC Algoritması**
HS256	SHA-256 kullanan HMAC
HS384	SHA-384 kullanan HMAC
HS512	SHA-512 kullanan HMAC
RS256	RSASSA-PKCS1-v1_5 , SHA-256 kullanılarak
RS384	RSASSA-PKCS1-v1_5 SHA-384 kullanılarak
RS512	RSASSA-PKCS1-v1_5 SHA-512 kullanılarak

Asimetrik anahtar sertifikası gereksinimleri

Bir belirteç asimetrik bir anahtarla imzalandıysa, belirteç veren kuruluşun genel anahtar sertifikası, kuyruk yöneticisinin belirteç kimlik doğrulaması için kullandığı anahtar havuzunda olmalıdır. Kimlik doğrulama belirteci alındığında, sertifikanın geçerlilik süresi içinde olması gerekir. Belirteç veren kuruluşun sertifikasının iptal edilmediğinden emin olmak için herhangi bir denetim yapılmadı.

Kimlik doğrulama belirteçlerindeki kullanıcı kimlikleri

Kuyruk yöneticisi, uygulamanın bağlamı olarak bir kimlik doğrulama simgesinin kullanıcı talebinde bulunan kullanıcı kimliğini benimseyecek şekilde yapılandırıldıysa, benimsenen kullanıcı kimliği aşağıdaki gereksinimleri karşılamalıdır:

- En çok 12 karakter içerebilir.
- Aşağıdaki karakterlerden biriyle başlamalıdır:

  A-Z a-z.
- Aşağıdaki karakterlerden herhangi birini içerebilir:

  0-9 A-Z a-z +,-. : = _
- UNKNOWN ve UNKNOWN ayrılmış kullanıcı kimliklerinden biri olmamalıdır.

İlgili görevler

AuthTokens kuyruk yöneticisini kabul edecek şekilde yapılandırma

İlgili başvurular

qm.ini dosyasının AuthToken kısmı

Linux AIX V 9.3.4

Bir kuyruk yöneticisinin kimlik doğrulama belirteçlerini kabul edecek şekilde yapılandırılması

Kimlik doğrulama belirteçleriyle kullanıcıların ve uygulamaların kimliğini doğrulamak için AIX ya da Linux üzerinde çalışan IBM MQ kuyruk yöneticinizi yapılandırın.

Başlamadan önce

Simgelerin IBM MQile nasıl çalıştığına ilişkin daha fazla bilgi için Kimlik doğrulama belirteçleriyle çalışma.

Kuyruk yöneticinizi yapılandırmadan önce, kuyruk yöneticisi **CONNAUTH** özniteliğinde başvurulan AUTHINFO nesnesinin IDPWOStipinde olup olmadığını denetleyin. Simge kimlik doğrulaması yalnızca, kuyruk yöneticisi işletim sistemi kullanıcı kimliği ve parola denetimi için yapılandırıldığında kullanılabilir.

Hizmet kısmına ilişkin **SecurityPolicy** özniteliğinin Grupolarak ayarlanmadığını denetleyin. **SecurityPolicy** belirtik olarak Grupolarak ayarlanırsa belirteç kimlik doğrulaması kullanılamaz. **SecurityPolicy** Grupolarak ayarlanırsa, **SecurityPolicy** özniteliğini Hizmet dörtünden kaldırın ve kuyruk yöneticisini yeniden başlatın.

Bu görev hakkında

IBM MQ 9.3.4 uygulamalarından, belirteçler kullanarak kuyruk yöneticisiyle kimlik doğrulaması yapabilir. IBM MQ , önerilen İnternet standardını RFC7519izleyen güvenilir sertifika verenlerden JSON Web Simgelerini (*JWT*) kabul eder. Bir kimliğin kimliğini doğrulamak için belirteçleri kullanabilirsiniz; daha sonra bu, gelecekteki yetkilendirme denetimleri için benimsenebilir.

Güvenilen sertifika verenin genel anahtar sertifikasını ya da simetrik anahtarı kuyruk yöneticisinin anahtar havuzuna kaydederek, kuyruk yöneticinizi belirteçleri kabul edecek şekilde yapılandırın. AuthToken kısmı qm.ini dosyasına ekleyin ve kuyruk yöneticisinin yeni yapılandırmayı seçmesi için güvenlik yapılandırmasını yenileyin.

Yordam

1. Anahtar havuzunu yaratın.
   a) Güvenilir sertifika verenden alınan genel anahtar sertifikası ya da simetrik anahtar için bir anahtar havuzu oluşturun. Dosya uzantısı `.kdb` olan bir CMS anahtar havuzunu ya da dosya uzantısı `.p12`olan bir PKCS#12 anahtar havuzunu kullanabilirsiniz.

      CMS anahtar havuzu yaratmak için aşağıdaki komutu verin:

      ```
      runmqakm -keydb -create -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
      -type cms
      ```

      runmqakm komutu bir hata döndürürse, bkz. runmqakm hata kodları. Komut başarıyla tamamlanırsa, dizinin içeriğini listelemek için `ls` komutunu kullanın:

      ```
      ls -l /var/mqm/qmgrs/qm1/tokenissuer
      ```

      Aşağıdaki dosyalar görüntülenir:

      ```
      -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) Gerekirse, mqm grubuna okuma erişimi verilebilmesi için yarattığınız anahtar havuzu dosyalarının grup sahipliğini değiştirin. Başlangıçta, yalnızca komutu çalıştıran yönetici kullanıcının oluşturulan dosyalara erişimi vardır.

      ```
      chgrp mqm /var/mqm/qmgrs/qm1/tokenissuer/key.*
      ```

   c) Grup mqm için okuma izinleri eklemek üzere anahtar havuzu dosyalarının kipini değiştirin. Örneğin, aşağıdaki komut dosya sahibi için okuma/yazma izinleri ve grup için salt okuma izni ekler.

      ```
      chmod 640 /var/mqm/qmgrs/qm1/tokenissuer/key.*
      ```

2. Anahtar havuzu parolasını **runqmcred** komutuyla şifreleyin ve şifrelenmiş dizgiyi bir dosyaya kaydedin.
   a) Anahtar havuzu parolasını şifrelemek için kullanılan ilk anahtarı içerecek bir dosya oluşturun.

      Dosya, ilk anahtarı tek bir metin satırı olarak içermelidir. İlk anahtarın uzunluk üst sınırı 256 bayttır. **INITKEY** kuyruk yöneticisi özniteliğini kullanarak kuyruk yöneticisi için bir başlangıç anahtarı ayarladıysanız, **INITKEY** özniteliğinin değerini yeni dosyaya kopyalayın. Kuyruk yöneticisi için bir başlangıç anahtarı ayarlamadıysanız, yeni, benzersiz bir şifreleme anahtarı yaratın ve ilk anahtar dosyasına ekleyin.

      Not: Daha fazla bilgi için bkz. INITKEY. İlk anahtarı belirtmezseniz, varsayılan bir anahtar kullanılır. Kendi ilk anahtarınızı kullanmak daha güvenlidir.

      Not: Dosyanın içeriğini güvenli tutmak için ilk anahtar dosyası için gerekli minimum izinleri verin. İlk anahtar dosyası yalnızca anahtar havuzu parolasını şifrelemek için kullanılır. Bu nedenle, yalnızca parolaları şifrelemek için ilk anahtarı kullanan yöneticilerin ilk anahtar dosyasını okuma erişimine sahip olması gerekir.

   b) Kuyruk yöneticisi ilk anahtarı önceden ayarlanmadıysa, kuyruk yöneticisi **INITKEY** özniteliğinin değerini, “2.a” sayfa 350. adımda yarattığınız ilk anahtara ayarlayın. Kuyruk yöneticisi ilk anahtarını ayarlamak için **ALTER QMGR** komutunu kullanın. Örneğin:

      ```
      ALTER QMGR INITKEY('myEncrypt10nK3y')
      ```

   c) Anahtar havuzu parolasını şifrelemek için **runqmcred** komutunu verin. İlk anahtarı içeren dosyanın yolunu belirtmek için **-sf** parametresini kullanın.

      ```
      runqmcred -sf initial.key
      ```

      İstendiğinde, anahtar havuzu parolasını girin. Şifrelenmiş parola, komut tarafından çıkışa yazılır.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Enter password:
*************
<QM>!2!b5rb01sMzFzc1ClZeQMryruWFM3HSm8DKyEaZK7qzWY=!TrWdU57DCDXM0Qah99I/Lg==
```

Son satırdaki dizgiyi kopyalayın ve bir dosyaya kaydedin.

3. Belirteç düzenleyicinin genel anahtar sertifikasını ya da simetrik anahtarını anahtar havuzuna eklemek için aşağıdaki yöntemlerden birini kullanın.
   - RSA genel anahtar sertifikasını anahtar havuzuna eklemek için aşağıdaki komutu verin:

     ```
     runmqakm -cert -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
     -label keylabel
              -file keyfile
     ```

   - Anahtar havuzuna base64 kodlanmış bir simetrik anahtar eklemek için aşağıdaki komutu verin:

     ```
     runmqakm -secretkey -add -db /var/mqm/qmgrs/qm1/tokenissuer/key.kdb -pw MyKeystorePassword
     -label keylabel
              -file keyfile -format ascii
     ```

   Burada *keylabel* , sertifikaya ya da gizli anahtara eklenecek etikettir ve *keyfile* , sertifikayı ya da base64 kodlanmış gizli anahtarı içeren dosyanın adıdır.
4. **AuthToken** kısmı ve aşağıdaki öznitelikleri `qm.ini` dosyasına ekleyin:
   - **KeyStore** özniteliği kullanılarak belirtilen anahtar havuzunun yolu.
   - **KeyStorePwdFile** özniteliği kullanılarak belirtilen anahtar havuzuna ilişkin parolayı içeren dosya.
   - **CertLabel** özniteliğini kullanarak, “3” sayfa 351. adımda eklediğiniz sertifika ya da simetrik anahtarın etiketi.

   Örneğin:

   ```
   AuthToken:
      KeyStore=/var/mqm/qmgrs/qm1/tokenissuer/key.kdb
      KeyStorePwdFile=/var/mqm/qmgrs/qm1/tokenissuer/key.pw
      CertLabel=rsakey
   ```

   Burada `key.kdb` , “1.a” sayfa 350. adımda oluşturduğunuz anahtar havuzunun adı ve `key.pw` , “2.c” sayfa 350. adımda oluşturduğunuz anahtar havuzuna ilişkin şifrelenmiş parolayı içeren dosyadır.

   AuthToken kısmı hakkında daha fazla bilgi için bkz. `qm.ini` dosyasının AuthToken kısmı.
5. Kuyruk yöneticisi, sonraki yetki denetimlerinde kullanılmak üzere belirteç kullanıcı talebinde bulunan kullanıcı kimliğini benimseyecek şekilde yapılandırıldıysa, **UserClaim** özniteliğini **AuthToken** kısmına ekleyin.

   Kuyruk yöneticisinin simgede kullanıcı kimliğini benimseyecek şekilde yapılandırılıp yapılandırılmadığını saptamak için aşağıdaki MQSC komutunu verin:

   ```
   DISPLAY AUTHINFO(authinfo_name) ADOPTCTX
   ```

   Burada *yetki_bilgileri_adı* , kuyruk yöneticisi **CONNAUTH** özniteliğinin değeridir. **ADOPTCTX** özniteliğinin değeri YESise, kuyruk yöneticisi simgede kullanıcı kimliğini kabul edecek şekilde yapılandırılır ve **UserClaim** özniteliği **AuthToken** kısmı içinde belirtilmelidir.

   UserClaim özniteliğinin değerini, benimsenecek kullanıcı kimliğini içeren belirteç talebinin adına ayarlayın. Örneğin, belirteç `"AppUser": "MyUserName"`talebini içeriyorsa, **AuthToken** kısmına şu satırı ekleyin:

   ```
   UserClaim=AppUser
   ```
6. `qm.ini` dosyasından belirteç yapılandırmasını alacak şekilde kuyruk yöneticisinin güvenlik yapılandırmasını yenileyin. **runmqsc** komutunu başlatmak için aşağıdaki komutu verin:

   ```
   runmqsc qm1
   ```

Daha sonra şu MQSC komutunu verin:

```
REFRESH SECURITY TYPE(CONNAUTH)
```

Sonraki adım

Kuyruk yöneticisiyle kimlik doğrulaması yapmak için geliştiricilerinizle birlikte çalışarak uygulamalardaki belirteçleri nasıl kullanabileceklerini anlamalarına yardımcı olun.

İlgili kavramlar

Kimlik doğrulama belirteci sorunlarını giderme

İlgili görevler

Uygulamada kimlik doğrulama belirteçlerini kullanma

İlgili başvurular

`qm.ini` dosyasının AuthToken kısmı

V 9.3.4 Seçtiğiniz belirteç verenizden bir kimlik doğrulama belirtecinin alınması

Bir IBM MQ kuyruk yöneticisine bağlandığında, seçtiğiniz belirteç düzenleyiciden kimlik doğrulama simgesi almak için uygulamanızı yazın.

Başlamadan önce

"Uygulamada kimlik doğrulama belirteçlerini kullanma" sayfa 352içindeki bilgilere bakın.

Yordam

- Bir kimlik doğrulama belirtecini nasıl elde edeceğinizi ve belirtecin tam içeriğini nasıl elde edeceğinizi, farklı belirteç yayıncıları arasında farklılık gösterir.

  Uygulamanızı, kimlik doğrulama belirtecini istemek ve elde etmek için seçtiğiniz belirteç veren ile etkileşimde bulunacak şekilde yazın.

  Kimlik doğrulama belirteci, kimlik doğrulama belirteçlerine ilişkin IBM MQ gereksinimlerine uygun olmalıdır. Bu gereksinimlerle ilgili daha fazla bilgi için bkz. "Kimlik doğrulama belirteçlerine ilişkin gereksinimler" sayfa 346.

  Uygulamanın bağlamı olarak bir belirteç talebinde bulunan bir kullanıcı kimliğini benimsemek istiyorsanız, kimlik doğrulama belirtecinin aşağıdaki gereksinimleri de karşılaması gerekir:

  - Kimlik doğrulama belirteci, kuyruk yöneticisinin belirteç kimlik doğrulama yapılandırmasındaki kullanıcı talebi adıyla eşleşen bir talep içermelidir.
  - Kullanıcı talebinin değeri, kimlik doğrulama belirteçlerinde kullanıcı kimliklerine ilişkin gereksinimleri karşılamalıdır. Daha fazla bilgi için bkz "Kimlik doğrulama belirteçlerindeki kullanıcı kimlikleri" sayfa 349.

Sonuçlar

Şimdi, geçerlilik denetimi için IBM MQ ' e sunulabilecek doğru biçimlendirilmiş bir JWT edindiniz.

İlgili görevler

AuthTokens kuyruk yöneticisini kabul edecek şekilde yapılandırma

İlgili başvurular

`qm.ini` dosyasının AuthToken kısmı

MQCSP-Güvenlik değiştirgeleri

V 9.3.4 Uygulamada kimlik doğrulama belirteçlerini kullanma

Bir IBM MQ kuyruk yöneticisine bağlandığında bir kimlik doğrulama simgesi sağlamak için uygulamanızı yazın.

Başlamadan önce

IBM MQ 9.3.4' den uygulamalar, bir kuyruk yöneticisine bağlandıklarında bir kimlik doğrulama belirteci sağlayabilir.

Uygulamanın aşağıdaki gereksinimleri karşılaması gerekir:

- V 9.3.5 C ya da Java biçiminde yazılmalıdır (IBM MQ classes for JMS/ Jakarta Messagingkullanılarak)
- Kuyruk yöneticisine IBM MQ clientolarak bağlanmalıdır. Yani, uygulamanın yerel bağ tanımlarını kullanmak yerine bir ağ üzerinden kuyruk yöneticisine bağlanması gerekir.
- AIX ya da Linuxüzerinde çalışan bir kuyruk yöneticisine bağlanmalıdır.

Uygulama bu gereksinimleri karşılamıyorsa, bağlantı başarısız olur ve uygulamaya MQRC_FUNCTION_NOT_SUPPORTED (2298) neden kodu döndürülür.

Kimlik doğrulama belirtecini sağlayan uygulama, IBM MQ MQI clients' i destekleyen herhangi bir platformda çalışabilir.

Otomatik istemci yeniden bağlantısı kullanan istemciler bağlanırken kimlik doğrulama simgesi sağlayamaz. Bir uygulama bir kimlik doğrulama simgesi belirtiyorsa ve MQCNO yapısında `MQCNO_RECONNECT` ya da `MQCNO_RECONNECT_Q_MGR` seçeneğini belirtiyorsa, bağlantı başarısız olur ve uygulamaya MQRC_RECONNECT_INCOMPATIBLE (2547) neden kodu döndürülür. Otomatik istemci yeniden bağlantısıyla ilgili ek bilgi için Otomatik istemci yeniden bağlantısıbaşlıklı konuya bakın.

Bu gereksinimler nedeniyle bir kimlik doğrulama simgesi sağlamak için uygulamayı yazamıyorsanız, istemci güvenlik çıkışını kullanarak uygulamanızı kimlik doğrulama belirteçlerini kullanacak şekilde geçirebilirsiniz. MQCSP yapısında kimlik doğrulama simgesini ayarlamak için istemci güvenlik çıkışı yazılabilir. Güvenlik çıkışlarıyla ilgili ek bilgi için İstemci bağlantısında güvenlik çıkışlarıbaşlıklı konuya bakın.

V 9.3.5 IBM MQ 9.3.5' den JMS istemci uygulamaları bağlanırken doğrudan bir simge sağlayabilir (bkz. “Seçtiğiniz belirteç verenizden bir kimlik doğrulama belirtecinin alınması” sayfa 352). IBM MQ 9.3.4' de Java uygulamaları, bir çıkış programı aracılığıyla dolaylı olarak bir simge sağlayabilir. Daha fazla bilgi için bkz. Java sınıfı MQCSP.

Bu görev hakkında

Not: JSON Web Signature (JWS) standardına uyan bir kimlik doğrulama simgesi, simgenin gerçekliğinin doğrulanmasına izin vermek için imzalandı, ancak şifrelenmedi. Bu nedenle, simgeye erişimi olan herkes tarafından okunabilir ve yeniden kullanılabilir. Kimlik doğrulama simgesinin ağ üzerinden gönderildiğinde (örneğin, TLS kullanılarak) şifreleme kullanılarak korunduğundan emin olmak için kuyruk yöneticisiyle bağlantıyı yapılandırın. Bir uygulama tarafından sağlanan kimlik bilgilerini korumaya ilişkin seçenekler hakkında daha fazla bilgi için bkz. “MQCSP parola koruması” sayfa 31.

Uygulamaları simge kullanarak bağlanacak şekilde değiştirmeden önce aşağıdakilerden emin olun:

- Kuyruk yöneticisi, “Bir kuyruk yöneticisinin kimlik doğrulama belirteçlerini kabul edecek şekilde yapılandırılması” sayfa 349 içindeki adımları izleyerek kimlik doğrulama belirteçlerini kabul edecek şekilde yapılandırıldı
- Uygulamanız, kimlik doğrulama sunucunuzda gerektiği şekilde geçerli bir simge alabilir, bkz. “Seçtiğiniz belirteç verenizden bir kimlik doğrulama belirtecinin alınması” sayfa 352.

Uygulama bir IBM MQ kuyruk yöneticisine bağlandığında bir kimlik doğrulama simgesi sağlamak için aşağıdaki işlemi ekleyin.

Yordam

- Bir C (MQI) uygulamasından kimlik doğrulama simgesi sağlamak için:

  Uygulamanın MQCONNX (MQCONN yerine) kullanarak bağlanması ve bir MQCSP yapısı sağlaması gerekir:

– **AuthenticationType** alanı MQCSP_AUTH_ID_TOKEN olarak ayarlanmalıdır.
– Yapının sürümü MQCSP_VERSION_3olarak ayarlanmalıdır.
– **TokenPtr** ya da **TokenOffset** alanı, kimlik doğrulama belirtecinize başvurmalıdır.
– **TokenLength** alanı, kimlik doğrulama simgesinin uzunluğuna ayarlanmalıdır.

MQCSP Sürüm 3 ve kimlik doğrulama simgesi kullanılarak bir kuyruk yöneticisine bağlanmak için örnek C kodu:

```
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         */
```

- V 9.3.5 Java uygulamasından bir kimlik doğrulama simgesi sağlamak için:

  IBM MQ classes for JMS/Jakarta Messaging kullanan uygulamalar, kullanıcı adı ve parola alan `createContext`ya da `createConnection` yöntemlerinden herhangi biri aracılığıyla bir simge sağlayabilir.

  Bir kimlik doğrulama belirteci sağlamak için:

  – **UserID** boş değerli ya da boş bir dizgi olarak ayarlanmalıdır; yani, boşluk olmadan " "
  – Belirteç, **Password** dizgisi olarak sağlanır.

  Bu, `ConnectionFactory` arabiriminin tüm IBM MQ uygulamaları için geçerlidir.

  Belirtik değiştirge biçimleri (örneğin, `createContext`(String **userID**, String **password**) ya da örtük değiştirge sürümleri (örneğin, `createContext`()) kullanılabilir.

  İkinci durumda, boş **userID** ve Token **Password** öncelikle bağlantı üreticisinde özellikler olarak sağlanmalıdır.

  Kimlik doğrulama simgesi kullanarak bir kuyruk yöneticisine bağlanmak için örnek Java kodu:

```
// 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
```

  Bağlantı MQRC_NOT_AUTHORIZED (2035) ya da MQRC_SECURITY_ERROR (2063) neden koduyla başarısız olursa, hatanın nedeniyle ilgili ek bilgi içeren bir hata iletisi için kuyruk yöneticisi hata

günlüğüne bakın. Kimlik doğrulama belirteçleriyle ilgili sorunların tanılanmasına ilişkin ek bilgi için Kimlik doğrulama simgesi sorunlarının giderilmesibaşlıklı konuya bakın.

Sonuçlar

Uygulama artık kuyruk yöneticisine bağlı. Kimlik doğrulaması için kullanılan belirteç sona erse bile, bağlantı kesilinceye kadar bağlı kalır. Uygulama kuyruk yöneticisiyle bağlantısını keserse ve yeniden bağlanması gerekiyorsa, yeniden bağlanmadan önce daha sonraki bir süre bitimine sahip yeni bir kimlik doğrulama belirteci edinmesi gerekebilir.

İlgili görevler

AuthTokens kuyruk yöneticisini kabul edecek şekilde yapılandırma

İlgili başvurular

qm.ini dosyasının AuthToken kısmı

MQCSP-Güvenlik değiştirgeleri

İptal edilen sertifikalarla çalışma

Sayısal sertifikalar Sertifika Yetkilileri tarafından iptal edilebilir. Platforma bağlı olarak, OCSP ya da LDAP sunucularındaki CRL ' leri kullanarak sertifikaların iptal durumunu denetleyebilirsiniz.

TLS el sıkışması sırasında, iletişim ortakları dijital sertifikalarla birbirlerini doğruluyor. Kimlik doğrulaması, alınan sertifikanın hala güvenilir olup olmadığını denetleyebilir. Sertifika Yetkilileri (CA), aşağıdakiler de dahil olmak üzere çeşitli nedenlerle sertifikaları iptal etmelidir:

- Sahip farklı bir kuruluşa taşındı
- Özel anahtar artık gizli değil

Sertifika kuruluşları, bir CRL ' de (Certificate Revocation List; Sertifika İptal Listesi) iptal edilen kişisel sertifikaları yayınlar. İptal edilen CA sertifikaları, bir Yetki İptal Listesinde (ARL) yayınlanır.

ALW AIX, Linux, and Windows platformlarında IBM MQ SSL desteği, OCSP (Online Certificate Status Protocol; Çevrimiçi Sertifika Durumu İletişim Kuralı) ya da LDAP (Lightweight Directory Access Protocol; Temel Dizin Erişimi İletişim Kuralı) sunucularında CRL 'leri ve ARL' leri kullanarak iptal edilen sertifikaları denetler. OCSP, tercih edilen yöntemdir.

IBM MQ classes for Java ve IBM MQ classes for JMS , istemci kanal tanımlama çizelgesi kütüğünde OCSP bilgilerini kullanamıyor. Ancak, OCSP ' yi Çevrimiçi Sertifika İletişim Kuralını Kullanmabaşlıklı konuda açıklandığı gibi yapılandırabilirsiniz.

z/OS IBM i IBM i ve z/OS platformlarında IBM MQ SSL desteği, yalnızca LDAP sunucularında CRL 'leri ve ARL' leri kullanarak iptal edilen sertifikaları denetler.

Sertifika Yetkilileri hakkında daha fazla bilgi için bkz. “dijital sertifikalar” sayfa 13.

OCSP/CRL denetimi

Uzak gelen sertifikalar için OCSP (Online Certificate Status Protocol; Çevrimiçi Sertifika Durumu İletişim Kuralı) /CRL (Certificate Revocation List; Sertifika İptal Listesi) denetimi gerçekleştirilir. Süreç, uzak sistemin kişisel sertifikasından gelen tüm zinciri doğrudan kök sertifikasına kadar kontrol eder.

OCSP doğrulamasını doğrulamak için openSSL kullanılması

Kuruluşunuz OCSP ' yi doğrulamak için openSSL kullanıyorsa ve IBM Global Security Kit (GSKit) TLS bağlantısını kullanmayı denerseniz, UNKNOWN durum uyarısı alırsınız.

Bunun nedeni, kökten başka zincirdeki tüm sertifikaların iptal durumu için GSKit tarafından denetlenolmasıdır. GSKit işlemi RFC 5280 'e uygundur ve bu, GSKit Güven İlkesinde açıklanmıştır. GSKit algoritması, RFC 5280 ve GSGSKitKit Trust Policy 'de açıklandığı gibi iptal bilgileri için tüm kullanılabilir kaynakları dener.

OCSP/CRL denetimi IBM MQiçinde nasıl çalışır?

IBM MQ , sertifikaları sertifika uzantısında ya da AUTHINFO nesnelerinde tanımlandığı şekilde, adlandırılmış OCSP ya da CRL uç noktalarına karşı denetlerken davranışı denetlemek için iki mekanizmayı destekler:

- qm.ini dosyasının SSL kısmıürününün **OCSPCheckExtensions**, **CDPCheckExtensions**ve **OCSPAuthentication** öznitelikleri ve
- Kuyruk yöneticisinin ve AUTHINFO OCSP ve CRLLDAP yapılandırmalarının SSLCRLNL parametresi kullanılarak. Ek bilgi için ALTER AUTHINFO ve ALTER QMGR başlıklı konuya bakın.

Uyarı:

AUTHTYPE(OCSP) içeren ALTER AUTHINFO komutu, IBM i ya da z/OS kuyruk yöneticisinde kullanım için geçerli değildir. Ancak, istemci kullanımı için istemci kanal tanımlama çizelgesine (CCDT) kopyalanacak altyapılarda belirtilebilir.

OCSPCheckExtensions ve **CDPCheckExtensions** SSL kısmı öznitelikleri, IBM MQ ' un sertifikanın AIA uzantısında ayrıntılı olarak açıklanan OCSP ya da CRL sunucusuna karşı bir sertifikayı doğrulayıp doğrulamayacağını denetler.

Etkinleştirilmezse, sertifika uzantısındaki OCSP ya da CRL sunucusuyla iletişim kurulmaz.

OCSP ya da CRL sunucuları AUTHINFO nesneleri aracılığıyla ayrıntılandırılırsa ve SSLCRLNL **QMGR** özniteliği kullanılarak başvurulursa, sertifika iptal işlemi sırasında IBM MQ bu sunucularla iletişim kurmayı dener.

Önemli: SSLCRLNL ad alanında tek bir OCSP AUTHINFO nesnesi tanımlanabilir.

Eğer:

OCSPCheckExtensions= NO ve **CDPCheckExtensions**=*NO* değerleri belirlenir ve AUTHINFO nesnelerinde tanımlı OCSP ya da CRL sunucusu yok

Sertifika iptal denetimi gerçekleştirilmez.

Bir sertifikayı iptal durumu için doğrularken IBM MQ , geçerli kılındıysa, aşağıdaki sırada belirtilen OCSP ya da CRL sunucularıyla iletişim kurar:

1. Bir **AUTHTYPE(OCSP)** nesnesinde ayrıntılı olarak açıklanan ve SSLCRLNL **QMGR** özniteliğinde başvurulan OCSP sunucusu.
2. **OCSPCheckExtensions**=*YES*ise, sertifikaların AIA uzantısında ayrıntılı olarak açıklanan OCSP sunucuları.
3. **CDPCheckExtensions** =*YES*ise, sertifikaların **CRLDistributionPoints** uzantısında ayrıntılı olarak açıklanan CRL sunucuları.
4. **AUTHINFO(CRLLDAP)** nesnelerinde ayrıntılı olarak açıklanan ve SSLCRLNL **QMGR** özniteliğinde başvurulan CRL sunucuları.

Bir sertifikayı doğrularken, bir adım OCSP sunucusu ya da CRL sunucusu tarafından sertifikaya ilişkin bir sorguya kesin REVOKED ya da VALID yanıtı döndürülmesine neden olursa, başka denetim gerçekleştirilmez ve sertifikanın durumu, sertifikaya güvenilip güvenilmeyeceğini belirlemek için kullanılır.

Bir OCSP sunucusu ya da CRL sunucusu UNKNOWNsonucunu döndürürse, bir OCSP ya da CRL sunucusu kesin sonuç döndürünceye ya da tüm seçenekler tükeninceye kadar işlem devam eder.

OCSP ve CRL sunucuları için, bir sertifikanın durumunun belirlenememesi durumunda iptal edilip edilmediğine ilişkin davranış farklıdır:

- CRL sunucuları için, CRL alınamıyorsa, sertifika NOT_REVOKED olarak değerlendirilir.
- OCSP sunucuları için, adı belirtilen bir OCSP sunucusundan iptal durumu alınamıyorsa, davranış qm.ini dosyasının SSL Stanza 'sındaki **OCSPAuthentication** özniteliğiyle denetlenir.

  Bu özniteliği, bir bağlantıyı engelleyecek, bağlantıya izin verecek ya da bir uyarı iletisiyle bağlantıya izin verecek şekilde yapılandırabilirsiniz.

OCSP denetimleri için qm.ini ve mqclient.ini dosyalarının SSL kıtasındaki **SSLHTTPProxyName**=*dizgi* özniteliğini kullanabilirsiniz. Dizgi, GSKit tarafından OCSP denetimleri için kullanılacak HTTP Yetkili sunucusunun anasistem adı ya da ağ adresidir.

IBM MQ 9.1.5 ' den, bir iptal denetimi gerçekleştirirken OCSP yanıtlayıcısının bekleneceği süreyi saniye cinsinden belirleyen `qm.ini` ya da `mqclient.ini` dosyalarının SSL kısmına **OCSPTimeout** değerini ayarlayabilirsiniz.

ALW İptal edilen sertifikalar ve OCSP

IBM MQ , hangi OCSP (Online Certificate Status Protocol; Çevrimiçi Sertifika Durumu İletişim Kuralı) yanıtlayıcısının kullanılacağını belirler ve alınan yanıtı işler. OCSP yanıtlayıcısını erişilebilir kılmak için adımlar atmanız gerekebilir.

Not: Bu bilgiler yalnızca AIX, Linux, and Windows sistemlerinde IBM MQ için geçerlidir.

OCSP kullanarak bir sayısal sertifikanın iptal durumunu denetlemek için IBM MQ , hangi OCSP yanıtlayıcısının iletişim kuracağını saptamak üzere iki yöntem kullanabilir:

- Denetlenecek sertifikada AuthorityInfoAccess (AIA) sertifika uzantısı kullanılarak.
- Bir kimlik doğrulama bilgi nesnesinde belirtilen ya da bir istemci uygulaması tarafından belirtilen bir URL kullanarak.

Bir kimlik doğrulama bilgisi nesnesinde ya da istemci uygulaması tarafından belirtilen bir URL , AIA sertifika uzantısındaki bir URL ' den önceliklidir.

OCSP yanıtlayıcısının URL bir güvenlik duvarının arkasında yer alıyorsa, OCSP yanıtlayıcısının erişilebilmesi için güvenlik duvarını yeniden yapılandırın ya da bir OCSP yetkili sunucusu ayarlayın. SSL kısmına ilişkin SSLHTTPProxyName değişkenini kullanarak yetkili sunucunun adını belirtin. İstemci sistemlerinde, MQSSLPROXY ortam değişkenini kullanarak yetkili sunucunun adını da belirtebilirsiniz. Daha fazla ayrıntı için ilgili bilgilere bakın.

TLS sertifikalarının iptal edilip edilmemesinden endişe etmiyorsanız, bir sınama ortamında çalışıyor olabilirsiniz; OCSPCheckExtensions öğesini SSL kısmına NO (HAYIR) olarak ayarlayabilirsiniz. Bu değişkeni ayarlarsanız, AIA sertifikası uzantısı yoksayılır. Bu çözüm, iptal edilen sertifikaları sunan kullanıcılardan erişime izin vermek istemediğiniz bir üretim ortamında kabul edilebilir değildir.

OCSP yanıtlayıcısına erişim çağrısı aşağıdaki üç sonuçtan biriyle sonuçlanabilir:

İyi
Sertifika geçerli.

İptal edildi
Sertifika iptal edildi.

Bilinmeyen
Bu sonuç üç nedenden biri ile ortaya çıkabilir:

- IBM MQ , OCSP yanıtlayıcıına erişemiyor.
- OCSP yanıtlayıcısı bir yanıt gönderdi, ancak IBM MQ yanıtın dijital imzasını doğrulayamıyor.
- OCSP yanıtlayıcısı, sertifika için iptal verisi olmadığını belirten bir yanıt gönderdi.

IBM MQ , Unknown(Bilinmiyor) değerinin OCSP sonucunu alırsa, bunun davranışı OCSPAuthentication özniteliğinin ayarına bağlıdır. Kuyruk yöneticileri için bu öznitelik aşağıdaki konumlardan birinde tutulur:

- Linux AIX AIX and Linuxüzerindeki `qm.ini` dosyasının SSL kısmına.
- Windows Windows kaydında.

Bu öznitelik IBM MQ Explorerkullanılarak ayarlanabilir. İstemciler için öznitelik, istemci yapılandırma dosyasının SSL kısmı içinde tutulur.

`Bilinmiyor` sonucu alınırsa ve OCSPAuthentication REQUIRED (varsayılan değer) olarak ayarlanırsa, IBM MQ bağlantıyı reddeder ve AMQ9716tipinde bir hata iletisi verir. Kuyruk

yöneticisi SSL olay iletileri etkinleştirildiyse, MQRC_CHANNEL_SSL_ERROR tipinde ve ReasonQualifier MQRQ_SSL_HANDSHAKE_ERROR değerine ayarlı bir SSL olay iletisi oluşturulur.

`Bilinmiyor` sonucu alınırsa ve OCSPAuthentication OPTIONAL olarak ayarlanırsa, IBM MQ SSL kanalının başlatılmasına izin verir ve uyarı ya da SSL olay iletileri oluşturulmaz.

`Bilinmiyor` sonucu alınırsa ve OCSPAuthentication UYARI olarak ayarlanırsa, SSL kanalı başlar, ancak IBM MQ hata günlüğüne AMQ9717 tipinde bir uyarı iletisi gönderir. Kuyruk yöneticisi SSL olay iletileri etkinleştirildiyse, ReasonQualifier değeri MQRQ_SSL_UNKNOWN_REVOCATION olarak ayarlanmış MQRC_CHANNEL_SSL_WARNING tipinde bir SSL olay iletisi üretilir.

OCSP yanıtlarının dijital imzası

Bir OCSP yanıtlayıcısı yanıtlarını üç şekilde imzalayabilir. Yanıtlayıcınız hangi yöntemin kullanıldığını size bildirecektir.

- OCSP yanıtı, denetmekte olduğunuz sertifikayı veren CA sertifikası kullanılarak dijital olarak imzalanabilir. Bu durumda, herhangi bir ek sertifika ayarlamanıza gerek yoktur; TLS bağlantısını oluşturmak için önceden uyguladığınız adımlar OCSP yanıtını doğrulamak için yeterlidir.
- OCSP yanıtı, denetmekte olduğunuz sertifikayı veren aynı sertifika yetkilisi (CA) tarafından imzalanmış başka bir sertifika kullanılarak dijital olarak imzalanabilir. İmzalama sertifikası, bu durumda OCSP yanıtıyla birlikte gönderilir. OCSP yanıtlayıcısından aktarılan sertifikanın bu amaçla güvenilebilmesi için `id-kp-OCSPSigning` olarak ayarlanmış bir Genişletilmiş Anahtar Kullanım Uzantısı olmalıdır. OCSP yanıtı, bunu imzalayan sertifikayla birlikte gönderildiğinden (ve bu sertifika, TLS bağlantısı için zaten güvenilir olan bir CA tarafından imzalandığından), ek sertifika kuruluşu gerekmez.
- OCSP yanıtı, denetmekte olduğunuz sertifikayla doğrudan ilişkili olmayan başka bir sertifika kullanılarak dijital olarak imzalanabilir. Bu durumda, OCSP yanıtı OCSP yanıtının kendisi tarafından verilen bir sertifika tarafından imzalanır. OCSP yanıtlayıcı sertifikasının bir kopyasını, OCSP denetimini gerçekleştiren istemcinin ya da kuyruk yöneticisinin anahtar veritabanına eklemeniz gerekir. Bkz. “AIX, Linux, and Windows üzerindeki bir anahtar havuzuna CA sertifikası ya da kendinden onaylı bir sertifikanın genel kısmını ekleme” sayfa 314. Bir CA sertifikası eklendiğinde, varsayılan olarak bu bağlamda gerekli olan güvenilir bir kök olarak eklenir. Bu sertifika eklenmezse, IBM MQ OCSP yanıtındaki dijital imzayı doğrulayamaz ve OCSP denetimi, OCSPAuthentication değerine bağlı olarak IBM MQ ' in kanalı kapatmasına neden olabilecek Bilinmeyen bir sonuçla sonuçlanır.

Java ve JMS istemci uygulamalarında OCSP (Online Certificate Status Protocol; Çevrimiçi Sertifika Durumu Protokolü)

Java API ' nin bir sınırlaması nedeniyle IBM MQ , TLS güvenli yuvaları için OCSP (Online Certificate Status Protocol; Çevrimiçi Sertifika Durumu İletişim Kuralı) sertifika iptal denetimini yalnızca tüm Java sanal makinesi (JVM) işlemi için OCSP etkinleştirildiğinde kullanabilir. JVM 'deki tüm güvenli yuvalar için OCSP' yi etkinleştirmenin iki yolu vardır:

- JRE java.security dosyasını, Tablo 1 'de gösterilen OCSP yapılandırma ayarlarını içerecek şekilde düzenleyin ve uygulamayı yeniden başlatın.
- java.security.Security.setProperty() API, yürürlükte olan herhangi bir Java Security Manager ilkesine tabidir.

Minimum olarak, ocsp.enable ve ocsp.responderURL değerlerinden birini belirtmeniz gerekir.

Özellik Adı	**Açıklama**
ocsp.enable	Bu özelliğin değeri `true` ya da `false`. `true`ise, sertifika iptal denetimi yapılırken OCSP denetimi etkinleştirilir; `false` ise ya da ayarlanmamışsa, OCSP denetimi devre dışı bırakılır.
ocsp.responderURL	Bu özelliğin değeri, OCSP yanıtlayıcısının konumunu tanımlayan bir URL ' dir. İşte bir örnek; `ocsp.responderURL=http://ocsp.example.net:80`. Varsayılan olarak, OCSP yanıtlayıcısının yeri, doğrulanmakta olan sertifikadan örtük olarak saptanır. Özellik,

Özellik Adı	Açıklama
	Yetki Bilgileri Erişimi uzantısı (RFC 3280 'de tanımlanan) sertifikada yoksa ya da geçersiz kılınması gerektiriyorsa kullanılır.
ocsp.responderCertSubjectName	Bu özelliğin değeri, OCSP yanıt verenin sertifikasının konu adıdır. İşte bir örnek; `ocsp.responderCertSubjectName="CN=OCSP Responder, O=XYZ Corp"`. Varsayılan olarak, OCSP yanıtlayıcısının sertifikası, doğrulanmakta olan sertifikayı veren kişinin sertifikasıdır. Bu özellik, varsayılan değer geçerli olmadığında OCSP yanıtlayıcısının sertifikasını tanımlar. Değeri, sertifika yolu geçerlilik denetimi sırasında sağlanan sertifika kümesindeki bir sertifikayı tanıtan bir dizgi ayırt edici adıdır (RFC 2253 'te tanımlanmıştır). Konu adının tek başına sertifikayı benzersiz olarak tanımlamak için yeterli olmadığı durumlarda, bunun yerine hem ocsp.responderCertIssuerName , hem de ocsp.responderCertSerialNumber özellikleri kullanılmalıdır. Bu özellik ayarlandığında, ocsp.responderCertIssuerName ve ocsp.responderCertSerialNumber özellikleri yoksayılır.
ocsp.responderCertIssuerName	Bu özelliğin değeri, OCSP yanıt verenin sertifikasının veren adıdır. İşte bir örnek; `ocsp.responderCertIssuerName="CN=Enterprise CA, O=XYZ Corp"`. Varsayılan olarak, OCSP yanıtlayıcısının sertifikası, doğrulanmakta olan sertifikayı veren kişinin sertifikasıdır. Bu özellik, varsayılan değer geçerli olmadığında OCSP yanıtlayıcısının sertifikasını tanımlar. Değeri, sertifika yolu geçerlilik denetimi sırasında sağlanan sertifika kümesindeki bir sertifikayı tanıtan bir dizgi ayırt edici adıdır (RFC 2253 'te tanımlanmıştır). Bu özellik ayarlandığında, ocsp.responderCertSerialNumber özelliği de ayarlanmalıdır. ocsp.responderCertSubjectName özelliği ayarlandığında bu özellik yoksayılır.
ocsp.responderCertSerialNumber	Bu özelliğin değeri, OCSP yanıtlayıcısının sertifikasının seri numarasıdır. İşte bir örnek; `ocsp.responderCertSerialNumber=2A:FF:00`. Varsayılan olarak, OCSP yanıtlayıcısının sertifikası, doğrulanmakta olan sertifikayı veren kişinin sertifikasıdır. Bu özellik, varsayılan değer geçerli olmadığında OCSP yanıtlayıcısının sertifikasını tanımlar. Bu değer, sertifika yolu geçerlilik denetimi sırasında sağlanan sertifika kümesindeki bir sertifikayı tanıtan onaltılı basamak dizgisidir (iki nokta ya da boşluk ayırıcıları olabilir). Bu özellik ayarlandığında, ocsp.responderCertIssuerName özelliği de ayarlanmalıdır. ocsp.responderCertSubjectName özelliği ayarlandığında bu özellik yoksayılır.

OCSP ' yi bu şekilde etkinleştirmeden önce dikkat edilmesi gereken bazı noktalar vardır:

- OCSP yapılandırmasının ayarlanması, JVM işlemindeki tüm güvenli yuvaları etkiler. Bazı durumlarda, JVM TLS güvenli yuvalarını kullanan diğer uygulama koduyla paylaşıldığında bu yapılandırmanın istenmeyen yan etkileri olabilir. Seçilen OCSP yapılanışının, aynı JVM ' de çalışan tüm uygulamalar için uygun olduğunu doğrulayın.
- JRE ' ye bakım uygulanması, java.security dosyasının üzerine yazılmasına neden olabilir. java.security dosyasının üzerine yazmaktan kaçınmak için Java ara düzeltmelerini ve ürün bakımını uygularken dikkatli olun. Bakım uyguladıktan sonra java.security değişikliklerinizi yeniden uygulamak gerekebilir. Bu nedenle, java.security.Security.setProperty() API.
- OCSP denetiminin etkinleştirilmesi, yalnızca iptal denetimi de etkinleştirildiyse etkili olur. `PKIXParameters.setRevocationEnabled()` yöntemiyle iptal denetimi etkinleştirilir.

- Yerel kesicilerin OCSP geri verilmesini etkinleştirmebaşlıklı konuda açıklanan AMS Java Interceptor kullanıyorsanız, anahtar deposu yapılanış dosyasındaki AMS OCSP yapılanışıyla çakışan bir java.security OCSP yapılanışını kullanmaktan kaçının.

Sertifika İptal Listeleri ve Yetki İptal Listeleriyle Çalışma

CRL ve ARL ' ler için IBM MQ desteği, platforma göre değişir.

Her platformda CRL ve ARL desteği aşağıdaki gibidir:

- z/OSişletim sisteminde Sistem SSL, Tivoli Public Key Infrastructure ürünü tarafından LDAP sunucularında saklanan CRL 'leri ve ARL' leri destekler.
- Diğer platformlarda, CRL ve ARL desteği PKIX X.509 V2 CRL profil önerileriyle uyumludur.

IBM MQ , önceki 12 saatte erişilen CRL 'lerin ve ARL' lerin önbelleğini sağlar.

Bir kuyruk yöneticisi ya da IBM MQ MQI client bir sertifika aldığında, sertifikanın hala geçerli olduğunu onaylamak için CRL ' yi denetler. Önbellek varsa, IBM MQ önce önbelleği geri verir. CRL önbellekte değilse, IBM MQ , LDAP CRL sunucusu konumlarını, IBM MQ kullanılabilir bir CRL buluncaya kadar *SSLCRLNL* özniteliği tarafından belirtilen kimlik doğrulama bilgileri nesnelerinin ad en başında gerçekleştikleri sırayla sorgular. Ad listesi belirtilmezse ya da boş bir değerle belirtilirse, CRL ' ler denetlenmez.

LDAP sunucularının ayarlanması

LDAP Dizin Bilgileri Ağacı yapısını, CA ' ların Ayırt Edici Adları sıradüzenini yansıtacak şekilde yapılandırın. Bunu LDAP Veri Değiştokuş Biçimi dosyalarını kullanarak yapın.

Sertifikaları ve CRL 'leri veren CA' ların Ayırt Edici Adlarına karşılık gelen sıradüzenini kullanmak için LDAP Dizin Bilgi Ağacı 'nı (DIT) yapılandırın. DIT yapısını, LDAP Veri Değiştokuş Biçimi 'ni (LDIF) kullanan bir dosyayla ayarlayabilirsiniz. Bir dizini güncellemek için LDIF dosyalarını da kullanabilirsiniz.

LDIF kütükleri, LDAP dizinindeki nesneleri tanımlamak için gereken bilgileri içeren ASCII metin kütükleridir. LDIF dosyaları, her biri bir Ayırt Edici Addan, en az bir nesne sınıfı tanımlamasından ve isteğe bağlı olarak birden çok öznitelik tanımlamasından oluşan bir ya da daha çok giriş içerir.

`certificateRevocationList;binary` özniteliği, iptal edilen kullanıcı sertifikalarının ikili biçiminde bir listesini içerir. `authorityRevocationList;binary` özniteliği, iptal edilen CA sertifikalarının ikili bir listesini içerir. IBM MQ TLS ile kullanmak için, bu özniteliklere ilişkin ikili veriler DER (Tanımlı Kodlama Kuralları) biçimine uymalıdır. LDIF dosyalarıyla ilgili ek bilgi için LDAP sunucunuzla birlikte sağlanan belgelere bakın.

Şekil 20 sayfa 360 , CA1tarafından verilen CRL 'leri ve ARL' leri yüklemek için LDAP sunucunuza giriş olarak yaratabileceğiniz, "CN=CA1, OU=Test, O=IBM, C=GB", IBMiçindeki Test kuruluşu tarafından ayarlanan hayali bir Sertifika Yetkilisi olan örnek bir LDIF dosyasını gösterir.

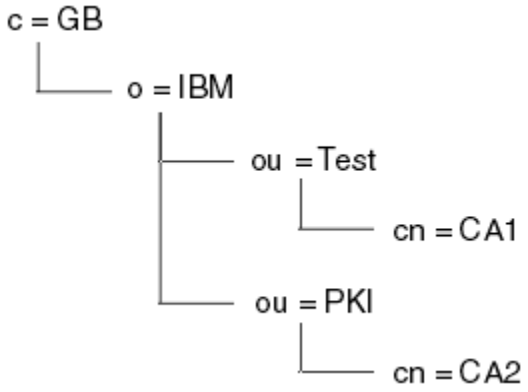
```
dn: o=IBM, c=GB
o: IBM
objectclass: top
objectclass: organization

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)
```

Şekil 20. Bir Sertifika Yetkilisi için örnek LDIF dosyası. Bu, uygulamadan uygulamaya kadar değişiklik gösterebilir.

Şekil 21 sayfa 361 , Şekil 20 sayfa 360 içinde gösterilen örnek LDIF dosyasını yüklediğinizde LDAP sunucunuzun oluşturduğu DIT yapısını ve PKI kuruluşu tarafından IBMiçinde de oluşturulan hayali bir Sertifika Yetkilisi olan CA2için benzer bir dosyayı gösterir.



Şekil 21. LDAP Dizin Bilgileri Ağacı yapısı örneği

IBM MQ , hem CRL 'leri hem de ARL' leri denetler.

Not: LDAP sunucunuza ilişkin erişim denetimi listesinin, yetkili kullanıcıların CRL 'leri ve ARL' leri tutan girişleri okumasına, aramasına ve karşılaştırmasına izin verdiğinden emin olun. IBM MQ , LDAP sunucusuna AUTHINFO nesnesinin LDAPUSER ve LDAPPWD özelliklerini kullanarak erişir.

LDAP sunucularının yapılandırılması ve güncellenmesi
LDAP sunucunuzu yapılandırmak ya da güncellemek için bu yordamı kullanın.

1. Sertifika Yetkilinizden ya da Yetkilerinizden DER biçiminde CRL 'leri ve ARL' leri edinin.
2. LDAP sunucunuzla birlikte sağlanan bir metin düzenleyicisini ya da aracı kullanarak, CA ' nın Ayırt Edici Adını ve gerekli nesne sınıfı tanımlamalarını içeren bir ya da daha çok LDIF dosyası yaratın. DER biçim verilerini LDIF dosyasına, CRL 'ler için `certificateRevocationList;binary` özniteliğinin, ARL' ler için `authorityRevocationList;binary` özniteliğinin ya da her ikisinin değerleri olarak kopyalayın.
3. LDAP sunucunuzu başlatın.
4. “2” sayfa 361. adımda oluşturduğunuz LDIF kütüğündeki ya da kütüklerdeki girişleri ekleyin.

LDAP CRL sunucunuzu yapılandırdıktan sonra, sunucunun doğru şekilde kurulup kurulmadığını denetleyin. Önce, kanalda iptal edilmeyen bir sertifika kullanmayı deneyin ve kanalın doğru başlatılıp başlatılmadığını denetleyin. Daha sonra iptal edilen bir sertifika kullanın ve kanalın başlatılıp başlatılmadığını denetleyin.

Sertifikasyon Yetkilerinden sık sık güncellenen CRL ' leri edinin. Bunu her 12 saatte bir LDAP sunucularınızda yapmayı düşünün.

CRL 'lere ve ARL' lere bir kuyruk yöneticisiyle erişilmesi

Kuyruk yöneticisi, LDAP CRL sunucusunun adresini tutan bir ya da daha çok kimlik doğrulama bilgisi nesnesiyle ilişkilendirilir. IBM i IBM MQ on IBM i , diğer platformlardan farklı davranır.

Bu bölümde, Sertifika İptal Listelerine (CRL) ilişkin bilgilerin, Yetki İptal Listeleri (ARL) için de geçerli olduğunu unutmayın.

Kuyruk yöneticisine, her biri LDAP CRL sunucusunun adresini bulunduran kimlik doğrulama bilgileri nesnelerini kuyruk yöneticisine sağlayarak CRL ' lere nasıl erişileceğini anlatırsınız. Kimlik doğrulama bilgileri nesneleri, *SSLCRLNL* kuyruk yöneticisi özniteliğinde belirtilen bir ad çubuğunda tutulur.

Aşağıdaki örnekte, değiştirgeleri belirtmek için MQSC kullanılmıştır:

1. AUTHTYPE parametresi CRLLDAP olarak ayarlanmış olarak, DEFINE AUTHINFO MQSC komutunu kullanarak kimlik doğrulama bilgileri nesnelerini tanımlayın. IBM i IBM iüzerinde, CRTMQMAUTI CL komutunu da kullanabilirsiniz.

AUTHTYPE parametresine ilişkin CRLLDAP değeri, LDAP sunucularında CRL ' lere erişildiğini gösterir. Oluşturduğunuz CRLLDAP tipindeki her kimlik doğrulama bilgileri nesnesi, bir LDAP sunucusunun adresini tutar. Birden fazla kimlik doğrulama bilgisi nesnesine sahip olduğunuzda, işaret ettikleri LDAP sunucularının aynı bilgileri içermesi gerekir. Bu, bir ya da daha fazla LDAP sunucusu başarısız olursa hizmet sürekliliği sağlar.

z/OS Ayrıca, yalnızca z/OS üzerinde, tüm LDAP sunucularına aynı kullanıcı kimliği ve parola kullanılarak erişilmelidir. Kullanılan kullanıcı kimliği ve parola, ad listesi içindeki ilk AUTHINFO nesnesinde belirtilenlerdir.

Tüm platformlarda, kullanıcı kimliği ve parola LDAP sunucusuna şifresiz olarak gönderilir.

2. DEFINE NAMELIST MQSC komutunu kullanarak, kimlik doğrulama bilgileri nesnelerinizin adları için bir ad listesi tanımlayın. z/OS z/OSüzerinde, NLTYPE ad listesi özniteliğinin AUTHINFO olarak ayarlandığından emin olun.
3. ALTER QMGR MQSC komutunu kullanarak, kuyruk yöneticisine ad belirtin. Örneğin:

```
ALTER QMGR SSLCRLNL(sslcrlnlname)
```

Burada `sslcrlnlname` , kimlik doğrulama bilgileri nesnelerinizin en önemli nesnesidir.

Bu komut, *SSLCRLNL*adlı bir kuyruk yöneticisi özniteliğini ayarlar. Kuyruk yöneticisinin bu özniteliğe ilişkin ilk değeri boş.

IBM i IBM iüzerinde kimlik doğrulama bilgisi nesnelerini belirtebilirsiniz; ancak, kuyruk yöneticisi kimlik doğrulama bilgisi nesnelerini ya da kimlik doğrulama bilgisi nesnelerini kullanmaz. Yalnızca IBM i kuyruk yöneticisi tarafından oluşturulan bir istemci bağlantı çizelgesini kullanan IBM MQ istemcileri, o IBM i kuyruk yöneticisi için belirtilen kimlik doğrulama bilgilerini kullanır. IBM i üzerindeki *SSLCRLNL* kuyruk yöneticisi özniteliği, istemcilerin hangi kimlik doğrulama bilgilerini kullandığını belirler. Bir IBM i kuyruk yöneticisine CRL ' lere nasıl erişeceğinin anlatılmasıyla ilgili bilgi için bkz. “IBM i üzerinde CRL 'lere ve ARL' lere erişme” sayfa 362 .

Bir ya da daha çok LDAP sunucusunun arızalanması durumunda hizmetin devamlılığını sağlamak için ad alanına alternatif LDAP sunucularına en fazla 10 bağlantı ekleyebilirsiniz. LDAP sunucularının aynı bilgileri içermesi gerektiğini unutmayın.

IBM i *IBM i üzerinde CRL 'lere ve ARL' lere erişme*

IBM iüzerindeki CRL 'lere ya da ARL' lere erişmek için bu yordamı kullanın.

Bu bölümde, Sertifika İptal Listelerine (CRL) ilişkin bilgilerin, Yetki İptal Listeleri (ARL) için de geçerli olduğunu unutmayın.

IBM iüzerinde belirli bir sertifika için CRL konumu ayarlamak üzere aşağıdaki adımları izleyin:

1. “DCM ' ye Erişilmesi” sayfa 272içinde açıklandığı gibi DCM arabirimine erişin.
2. Gezinme panosunda **CRL konumlarını yönet** görev kategorisinde **CRL konumu ekle**seçeneğini tıklatın. Görev çerçevesinde CRL Konumlarını Yönet sayfası görüntülenir.
3. **CRL Konumu Adı** alanına bir CRL konumu adı yazın; örneğin, `LDAP Server #1`
4. **LDAP Sunucusu** alanında LDAP sunucusu adını yazın.
5. **Güvenli Yuva Katmanı (SSL) Kullan** alanında, TLS kullanarak LDAP sunucusuna bağlanmak istiyorsanız **Evet** seçeneğini belirleyin. Tersi durumda, **Hayır**seçeneğini belirleyin.
6. **Kapı Numarası** alanında LDAP sunucusu için bir kapı numarası yazın; örneğin, 389.
7. LDAP sunucunuz anonim kullanıcıların dizini sorgulamasına izin vermiyorsa, **oturum açma ayırt edici adı** alanına sunucu için bir oturum açma ayırt edici adı yazın.
8. **Tamam** düğmesini tıklatın. DCM, CRL konumunu oluşturduğunu bildirir.
9. Gezinme panosunda **Sertifika Deposu Seç**seçeneğini tıklatın. Görev çerçevesinde Sertifika Deposu Seç sayfası görüntülenir.

10. **Diğer Sistem Sertifikası Deposu** onay kutusunu seçin ve **Devam**düğmesini tıklatın. Sertifika Deposu ve Parola sayfası görüntülenir.
11. **Sertifika deposu yolu ve dosya adı** alanında, “IBM i üzerinde sertifika deposu oluşturma” sayfa 274olduğunda ayarladığınız IFS yolunu ve dosya adını yazın.
12. **Sertifika Deposu Parolası** alanında bir parola yazın. **Continue**(Devam) seçeneğini tıklatın. Yürürlükteki Sertifika Deposu sayfası görev çerçevesinde görüntülenir.
13. Gezinme panosunda **Sertifikaları Yönet** görev kategorisinde **CRL konum atamasını güncelle**seçeneğini tıklatın. Görev çerçevesinde CRL Konum Ataması sayfası görüntülenir.
14. CRL konumunu atamak istediğiniz CA sertifikasına ilişkin radyo düğmesini seçin. **CRL Konum Atamasını Güncelle**' yi tıklatın. Görev çerçevesinde CRL Konum Atamasını Güncelle sayfası görüntülenir.
15. Sertifikaya atamak istediğiniz CRL konumuna ilişkin radyo düğmesini seçin. **Atamayı Güncelle**' yi tıklatın. DCM, atamayı güncellediğini bildirir.

DCM ' nin Sertifika Yetkilisi tarafından farklı bir LDAP sunucusu atamanıza izin verdiğini unutmayın.

IBM MQ Explorer kullanarak CRL 'lere ve ARL' lere erişme
You can use IBM MQ Explorer to tell a queue manager how to access CRLs.

Bu bölümde, Sertifika İptal Listelerine (CRL) ilişkin bilgilerin Yetki İptal Listeleri (ARL) için de geçerli olduğunu unutmayın.

Bir CRL ' ye LDAP bağlantısı kurmak için aşağıdaki yordamı kullanın:

1. Kuyruk yöneticinizi başlattığınızdan emin olun.
2. **Kimlik Doğrulama Bilgileri** klasörünü sağ tıklatın ve **Yeni-> Kimlik Doğrulama Bilgileri**seçeneğini belirleyin. Açılan özellik sayfasında:
   a. İlk sayfada **Kimlik Doğrulama Bilgileri Oluştur**, CRL (LDAP) nesnesi için bir ad girin.
   b. **Özellikleri Değiştir**seçeneğinin **Genel** sayfasında bağlantı tipini seçin. İsteğe bağlı olarak bir açıklama girebilirsiniz.
   c. **Özellikleri Değiştir**seçeneğinin **CRL (LDAP)** sayfasını seçin.
   d. Ağ adı ya da IP adresi olarak LDAP sunucusu adını girin.
   e. Sunucu oturum açma ayrıntıları gerektiriyorsa, bir kullanıcı kimliği ve gerekirse bir parola sağlayın.
   f. **Tamam**'ı tıklatın.
3. Namelists klasörünü sağ tıklatın ve **New-> Namelist**(Yeni-> Ad) seçeneklerini belirleyin. Açılan özellik sayfasında:
   a. Ad listesi için bir ad yazın.
   b. CRL (LDAP) nesnesinin adını ekleyin (adım “2.a” sayfa 363 ' den) Listeye.
   c. **Tamam**'ı tıklatın.
4. Kuyruk yöneticisini sağ tıklatın, **Özellikler**seçeneğini belirleyin ve **SSL** sayfasını seçin:
   a. **Bu kuyruk yöneticisi tarafından alınan sertifikaları Sertifikasyon İptal Listelerine göre denetle** onay kutusunu işaretleyin.
   b. Ad listesi adını yazın (adım “3.a” sayfa 363 ' den) **CRL Nadeğiştiridır alanı deki** .

CRL 'lere ve ARL' lere IBM MQ MQI client ile erişme

Bir IBM MQ MQI clienttarafından denetlenmek üzere CRL ' leri tutan LDAP sunucularını belirtmek için üç seçeneğiniz vardır.

Bu bölümde, Sertifika İptal Listelerine (CRL) ilişkin bilgilerin, Yetki İptal Listeleri (ARL) için de geçerli olduğunu unutmayın.

LDAP sunucularını belirtmenin üç yolu şunlardır:

- Kanal tanımlama çizelgesinin kullanılması

- MQCONNX çağrısında SSL yapılandırma seçenekleri yapısının (MQSCO) kullanılması
- Active Directory ' yi kullanma (Active Directory desteği olan Windows sistemlerinde)

Daha fazla ayrıntı için ilgili bilgilere bakın.

Bir ya da daha fazla LDAP sunucusu başarısız olursa hizmet sürekliliğini sağlamak için alternatif LDAP sunucularına en fazla 10 bağlantı ekleyebilirsiniz. LDAP sunucularının aynı bilgileri içermesi gerektiğini unutmayın.

LDAP CRL ' lerine Linux (zSeries platformu) üzerinde çalışan bir IBM MQ MQI client kanalından erişemezsiniz.

OCSP yanıtlayıcısının ve CRL ' leri tutan LDAP sunucularının konumu
Bir IBM MQ MQI client sisteminde, sertifika iptal listelerini (CRL) bulunduran bir OCSP yanıtlayıcısının ve LDAP sunucularının konumunu belirtebilirsiniz.

Bu konumları, öncelik derecesinin azalması için burada açıklanan üç şekilde belirtebilirsiniz.

IBM i IBM iiçin bkz. IBM iüzerinde CRL 'lere ve ARL' lere erişme.

Bir IBM MQ MQI client uygulaması bir MQCONNX çağrısı yayınladığında

MQCONNX çağrısında CRL ' leri bulunduran bir OCSP yanıtlayıcısı ya da LDAP sunucusu belirtebilirsiniz.

Bir **MQCONNX** çağrısında, bağlanma seçenekleri yapısı (MQCNO) bir SSL yapılandırma seçenekleri yapısına (MQSCO) gönderme yapabilir. MQSCO yapısı, bir ya da daha çok kimlik doğrulama bilgisi kayıt yapısına (MQAIR) gönderme yapabilir. Her MQAIR yapısı, bir IBM MQ MQI client 'in CRL' leri tutan bir OCSP yanıtlayıcısına ya da LDAP sunucusuna erişmek için gereksinim duyduğu tüm bilgileri içerir. Örneğin, MQAIR yapısındaki alanlardan biri, bir yanıt verenin iletişim kurabileceği URL ' dir. MQAIR yapısıyla ilgili ek bilgi için MQAIR-Authentication information recordbaşlıklı konuya bakın.

OCSP yanıtlayıcısına ya da LDAP sunucularına erişmek için istemci kanal tanımlama çizelgesinin (ccdt) kullanılması

Böylece bir IBM MQ MQI client , CRL ' leri bulunduran bir OCSP yanıtlayıcısına ya da LDAP sunucularına erişebilir ve istemci kanal tanımlama çizelgesindeki bir ya da daha çok kimlik doğrulama bilgisi nesnesinin özniteliklerini içerir.

Bir sunucu kuyruk yöneticisinde, bir ya da daha çok kimlik doğrulama bilgisi nesnesi tanımlayabilirsiniz. Bir kimlik doğrulama nesnesinin öznitelikleri, bir OCSP yanıtlayıcısına (OCSP 'nin desteklendiği platformlarda) ya da CRL' leri bulunduran bir LDAP sunucusuna erişmek için gereken tüm bilgileri içerir. Özniteliklerden biri OCSP yanıtlayıcısının URLadresini, diğeri ise LDAP sunucusunun çalıştığı sistemin IP adresini belirtir.

z/OS IBM i AUTHTYPE (OCSP) içeren bir kimlik doğrulama bilgileri nesnesi IBM i ya da z/OS kuyruk yöneticisinde kullanım için geçerli değildir, ancak istemci kullanımı için istemci kanal tanımlama çizelgesine (CCDT) kopyalanacak bu altyapılarda belirtilebilir.

Bir IBM MQ MQI client 'in CRL' leri bulunduran bir OCSP yanıtlayıcısına ya da LDAP sunucularına erişmesini sağlamak için, bir ya da daha çok kimlik doğrulama bilgisi nesnesinin öznitelikleri bir istemci kanal tanımlama çizelgesine eklenebilir. Bu tür öznitelikleri aşağıdaki yollardan biriyle ekleyebilirsiniz:

Multi

Sunucu platformlarında AIX, Linux, IBM ive Windows

Bir ya da daha çok kimlik doğrulama bilgisi nesnesinin adlarını içeren bir ad listesi tanımlayabilirsiniz. Daha sonra kuyruk yöneticisi özniteliğini (**SSLCRLNL**) bu ad değerinin adına ayarlayabilirsiniz.

CRL kullanıyorsanız, daha yüksek kullanılabilirlik sağlamak için birden çok LDAP sunucusu yapılandırılabilir. Amaç, her bir LDAP sunucusunun aynı CRL ' leri tutmaktır. Bir LDAP sunucusu gerektiğinde kullanılamıyorsa, IBM MQ MQI client diğerine erişmeyi deneyebilir.

Ad listesi tarafından tanımlanan kimlik doğrulama bilgileri nesnelerinin öznitelikleri toplu olarak burada *sertifika iptal konumu*olarak adlandırılır. **SSLCRLNL**kuyruk yöneticisi özniteliğini ad listesi adına ayarladığınızda, sertifika iptal yeri kuyruk yöneticisiyle ilişkili istemci kanal tanımlaması çizelgesine kopyalanır. CCDT 'ye istemci sistemden paylaşılan dosya olarak erişilebiliyorsa ya da CCDT bir istemci sistemine kopyalandıysa, bu sistemdeki IBM MQ MQI client , CRL' leri tutan bir OCSP yanıtlayıcısına ya da LDAP sunucularına erişmek için CCDT ' deki sertifika iptal konumunu kullanabilir.

Kuyruk yöneticisinin sertifika iptal konumu daha sonra değiştirilirse, değişiklik kuyruk yöneticisiyle ilişkili CCDT ' ye yansıtılır. **SSLCRLNL**kuyruk yöneticisi özniteliği boş olarak ayarlanırsa, sertifika iptal konumu CCDT ' den kaldırılır. Bu değişiklikler, bir istemci sistemindeki çizelgenin hiçbir kopyasına yansıtılmaz.

Bir MQI kanalının istemci ve sunucu uçlarındaki sertifika iptal konumunun farklı olması gerekiyorsa ve sunucu kuyruk yöneticisi sertifika iptal konumunu yaratmak için kullanılan konumsa, bunu aşağıdaki gibi yapabilirsiniz:

1. Sunucu kuyruk yöneticisinde, istemci sisteminde kullanmak üzere sertifika iptal konumunu oluşturun.
2. Sertifika iptal konumunu içeren CCDT ' yi istemci sistemine kopyalayın.
3. Sunucu kuyruk yöneticisinde, sertifika iptal konumunu MQI kanalının sunucu sonunda gerekli olan şekilde değiştirin.
4. İstemci makinesinde, **runmqsc** komutunu **-n** parametresiyle kullanabilirsiniz.

Multi

İstemci platformlarında AIX, Linux, IBM ive Windows

CCDT dosyasında **-n** parametresi ve **DEFINE AUTHINFO** nesneleri ile runmqsc komutunu kullanarak istemci makinesinde bir CCDT oluşturabilirsiniz. Nesnelerin tanımlandığı sıra, dosyada kullanıldıkları sıradır. **DEFINE AUTHINFO** nesnesinde kullanabileceğiniz herhangi bir ad dosyada tutulmaz. CCDT kütüğündeki **DISPLAY AUTHINFO** nesnelerini kullandığınızda yalnızca konumlu numaralar kullanılır.

Not: -n değiştirgesini belirtirseniz, başka bir değiştirge belirtmemelisiniz.

Windows üzerinde Active Directory ' yi kullanma

Windows

Windows sistemlerinde, geçerli CRL bilgilerini Active Directory' de yayınlamak için **setmqcrl** denetim komutunu kullanabilirsiniz.

setmqcrl komutu OCSP bilgilerini yayınlamıyor.

Bu komut ve sözdizimi hakkında bilgi için bkz. setmqcrl.

IBM MQ classes for Java ve IBM MQ classes for JMS ile CRL 'lere ve ARL' lere erişme

IBM MQ classes for Java ve IBM MQ classes for JMS , CRL ' lere diğer platformlardan farklı şekilde erişirler.

CRL 'lerle ve ARL' lerle IBM MQ classes for Javaile çalışma hakkında bilgi için bkz. Sertifika iptal listelerini kullanma

CRL 'lerle ve ARL' lerle IBM MQ classes for JMSile çalışma hakkında bilgi için bkz. SSLCERTSTORES object property

Kimlik doğrulama bilgileri nesnelerini işleme

MQSC ya da PCF komutlarını ya da IBM MQ Explorerkomutlarını kullanarak kimlik doğrulama bilgileri nesnelerini işleyebilirsiniz.

Aşağıdaki MQSC komutları, kimlik doğrulama bilgileri nesneleri üzerinde işlem yapar:

- AUTHINFO TANıMLAYıN

- AUTHINFO DEğIşTIR
- AUTHINFO ÖĞESINI SIL
- AUTHINFO BILGILERINI GöRüNTüLE

Bu komutların tam açıklaması için bkz. MQSC komutları.

Aşağıdaki Programlanır Komut Biçimi (PCF) komutları, kimlik denetimi bilgi nesneleri üzerinde işlem yapar:

- Kimlik Doğrulama Bilgileri Oluştur
- Kimlik Doğrulama Bilgilerini Kopyala
- Kimlik Doğrulama Bilgilerini Değiştir
- Kimlik Doğrulama Bilgilerini Sil
- Sorma Kimlik Doğrulama Bilgileri
- Kimlik Doğrulama Bilgileri Adlarını Sor

Bu komutların tam açıklaması için bkz. Programlanabilir Komut Biçimleri Tanımları.

Kullanılabilir olduğu platformlarda, IBM MQ Explorerolanağını da kullanabilirsiniz.

Linux AIX Takılabilir Kimlik Doğrulama Yönteminin (PAM) Kullanılması

PAM ' yi yalnızca AIX and Linux platformlarında kullanabilirsiniz. Tipik bir AIX ya da Linux sistemi, geleneksel kimlik doğrulama mekanizmasını uygulayan PAM modüllerine sahiptir; ancak daha fazlası olabilir. Parolaların doğrulanması temel görevinin yanı sıra, ek kuralları gerçekleştirmek için PAM modülleri de çağrılabilir.

Yapılandırma dosyaları, her uygulama için hangi kimlik doğrulama yönteminin kullanılacağını tanımlar. Örnek uygulamalar arasında standart uçbirim oturum açma, ftp ve Telnet yer alır.

PAM ' nin avantajı, uygulamanın kullanıcı kimliğinin nasıl doğrulandığını bilmesine ya da umursamasına gerek olmamasıdır. Uygulama PAM ' ye doğru bir kimlik doğrulama verisi sağlayabildiği sürece, arkasındaki mekanizma saydamdır.

Kimlik doğrulama verilerinin biçimi, kullanılmakta olan sisteme bağlıdır. Örneğin, IBM MQ , `MQCONNX` API çağrısında kullanılan MQCSP yapısı gibi parametreler aracılığıyla bir parola alır.

Önemli: IBM MQ 8.0.0 Fix Pack 3kuruluncaya kadar **AUTHENMD** özniteliğini ayarlayamaz ve daha sonra, gerek duyduğunuz komut düzeyini ayarlamak için *802* ' nin **-e CMDLEVEL=***düzeyini* (strmqm komutunda) kullanarak kuyruk yöneticisini yeniden başlatamazsınız.

Sisteminizin PAM kullanacak şekilde yapılandırılması

PAM ' yi çağırırken IBM MQtarafından kullanılan hizmet adı: *ibmmq*.

IBM MQ kuruluşunun, farklı işletim sistemleri için bilinen varsayılanlara dayalı olarak işletim sistemi kullanıcılarından gelen bağlantılara izin veren varsayılan bir PAM yapılandırmasını sürdürmeyi denediğini unutmayın.

Ancak, sistem yöneticiniz `/etc/pam.conf`ya da `/etc/pam.d/ibmmq`içinde tanımlanan kuralların uygun olduğunu doğrulamalıdır.

Nesnelere erişim yetkisi verilmesi

Bu bölümde, nesnelere erişimi denetlemek için nesne yetkisi yöneticisi ve kanal çıkış programlarının kullanılmasına ilişkin bilgiler yer alır.

ALW AIX, Linux, and Windows sistemlerinde. nesne yetki yöneticisini (OAM) kullanarak nesnelere erişimi denetleyebilirsiniz. Bu konu derlemi, OAM için komut arabiriminin kullanılmasına ilişkin bilgileri içerir.

Bu bölümde ayrıca, tüm platformlarda sisteminize güvenlik uygulamak için gerçekleştirilecek görevleri belirlemek üzere kullanabileceğiniz bir denetim listesi ve kullanıcılara IBM MQ nesnelerini yönetme ve IBM MQ nesneleriyle çalışma yetkisi verilmesine ilişkin dikkat edilecek noktalar da bulunur.

Sağlanan güvenlik mekanizmaları gereksinimlerinizi karşılamıyorsa, kendi kanal çıkış programlarınızı geliştirebilirsiniz.

Yetki için hangi kullanıcının kullanıldığının belirlenmesi

Kaynaklara erişim yetkisi, kullanıcının üyesi olduğu gruplara ya da belirli kiplerde, doğrudan bağlantıyla ilişkili kullanıcıya verilir. Bağlantı işlemi sırasında ve özellikle uzak (istemci) bağlantılar için, bu kimlik kuyruk yöneticisinin yapılandırması tarafından değiştirilebilir. Bu sayfa, IBM MQ ürününün farklı özelliklerini ve bu özelliklerin yapılandırma seçeneklerini listeler. Bu seçenekler, bağlantı kuran bir uygulamanın kimliğini ve bu özelliklerin yürürlüğe girdiği öncelik sırasını etkileyebilir.

Hangi kullanıcının benimsendiğini değiştirebilen özellikler

Hangi kullanıcının yetkilendirilmesi gerektiğini ayarlayabilecek farklı özellikler şunlardır:

Uygulama tarafından bildirilir kullanıcı
IBM MQtarafından bir uzak bağlantı başlatıldığında, işlemin çalıştığı işletim sistemi kullanıcısı, alan kuyruk yöneticisine gönderilir. Kullanıcıyı değiştiren başka bir yapılandırma yoksa, yetkilendirme denetimi için kullanılabilecek bir kullanıcı olduğundan emin olmak için bu kullanıcı gönderilir.

Bağlantıların herhangi bir sunucu tarafı doğrulaması olmadan kimliklerini belirtmesine izin verdiğinden, bu kullanıcının yetki temeli olarak kullanılması önerilmez. Bu, yönetimle görevli kullanıcıyı ('mqm ') da içerebilir.

Kanal MCAUSER ayarı
Ağ bağlamaları aracılığıyla bağlanan uygulamalar bunu bir IBM MQ kanal tanımı kullanılarak yapar. Kanal tanımları, bağlanan uygulamalar tarafından bildirilen kullanıcı yerine yetkilendirme için kullanılacak farklı bir kullanıcı belirtmek için kullanılabilecek **MCAUSER** özniteliğini destekler.

Bağlantı doğrulaması ADOPTCTX
Uygulamalar, kimlik doğrulama amacıyla bir kuyruk yöneticisine gönderilecek bir kullanıcı ve parola belirleyebilir. Bu kimlik bilgilerinin kimliği, Bağlantı Kimlik Doğrulaması özelliği için belirtilen yapılandırma kullanılarak doğrulanır. Bağlantı Kimlik Doğrulaması için **ADOPTCTX** seçeneği, bir kullanıcının başarıyla doğrulandıktan sonra yetkilendirme için kullanılıp kullanılmayacağını denetler. YES(EVET) olarak ayarlanırsa, kimlik doğrulaması için sağlanan kullanıcı yetkilendirme denetimleri için kullanılır.

V 9.3.4 IBM MQ 9.3.4' den kimlik doğrulaması için bir belirteç sağlanabilir; **ADOPTCTX** YESolarak ayarlanırsa, simgenin içerdiği taleplerden bir kullanıcı benimsenir.

Kanal kimlik doğrulama kaydı MCAUSER
Bağlantı işlenirken kuyruk yöneticisi, bağlantıyla eşleşen bir kanal kimlik doğrulama kaydı bulmaya çalışır. Bir kanal kimlik doğrulama kaydı eşleşirse ve **USERSRC** öznitelik değeri MAPolarak ayarlanırsa, IBM MQ yetkilendirmeler için kullanılan kullanıcıyı **MCAUSER** özniteliğinin değeriyle değiştirir.

Güvenlik çıkışları
Güvenlik çıkışları, IBM MQ güvenlik işlemesi sırasında yazılıp çağrılabilen özel işlevlerdir. İşlev çağrıldığında, yetki denetimi için kullanılacak bağlantı kullanıcısıyla ilgili birkaç alanı içeren MQCD yapısının bir kopyasıyla birlikte verilir. Güvenlik çıkışları, yetkilendirilecek kullanıcıyı değiştirmek için bu alanları değiştirebilir.

öncelik sırası

Aşağıdaki çizelgede, IBM MQ yetki vermek üzere bir kullanıcı seçtiğinde “Hangi kullanıcının benimsendiğini değiştirebilen özellikler” sayfa 367 içinde açıklanan her güvenlik özelliği için öncelik sırası

gösterilmektedir. Sıralama, en düşükten en yükseğe doğru, yani ilk satırdaki bir güvenlik özelliği ayarı, diğer satırların herhangi biri tarafından geçersiz kılınır.

Çizelge 71. Güvenlik özellikleri için öncelik sırası	
Sipariş	**Özellik**
1 (en düşük)	Uygulama Bildirildi Tanıtıcısı
2	Kanal tanımlaması **MCAUSER** özniteliği
3	**ADOPTCTX(YES)** ile bağlantı kimlik doğrulaması
4	**USERSRC(MAP)** ile kanal kimlik doğrulama kayıtları
5 (en yüksek)	Güvenlik Çıkışı

Erken evlat edinmenin etkileri

Bağlantı kimlik doğrulaması ve kanal kimlik doğrulama kayıtları, bağlantı kimlik doğrulaması kullanıcı benimsemesinin ne zaman gerçekleştirileceğini denetleyen bir yapılandırma seçeneği sağlar. Bu ayar, erken benimseme olarak adlandırılır. Erken benimseme etkinleştirilirse, kanal kimlik doğrulama kayıtları işlenmeden önce bağlantı kimlik doğrulamasını benimseme gerçekleşir (bu, kanal kimlik doğrulama kayıtlarının herhangi bir **CONNAUTH** benimsemesini geçersiz kıldığı anlamına gelir).

Devre dışı bırakılırsa, sipariş tersine çevrilir; başka bir değişle, kanal kimlik doğrulama kayıtları **CONNAUTH** benimsemeden önce işlenir. Bu durumda, bağlantı kimlik doğrulamasının benimsenmesi, kanal kimlik doğrulamasının kaydedilmesi için daha yüksek etkili bir önceliğe sahiptir.

Erken benimseme için varsayılan ayar `enabled`(etkin) değeridir.

ALW AIX, Linux, and Windows üzerinde OAM kullanarak nesnelere erişimi denetleme

Nesne yetki yöneticisi (OAM), IBM MQ nesnelerine yetki vermek ve bu nesneleri geri almak için bir komut arabirimi sağlar.

Bu komutları “AIX, Linux, and Windows üzerinde IBM MQ yönetimi yetkisi” sayfa 414içinde açıklandığı şekilde kullanmak için uygun yetkiye sahip olmanız gerekir. IBM MQ denetim yetkisi olan kullanıcı kimliklerinin kuyruk yöneticisi üzerinde *ayrıcalıklı kullanıcı* yetkisi vardır; bu, onlara MQI isteklerini ya da komutlarını yayınlamak için ek izin vermeniz gerekmediği anlamına gelir.

Linux AIX AIX and Linux üzerinde OAM kullanıcı tabanlı izinler

From IBM MQ 8.0, on UNIX and Linux systems, the object authority manager (OAM) can use user-based authorization as well as group-based authorization.

IBM MQ 8.0'den önce, UNIX and Linux üzerindeki erişim denetim listeleri (EDL' ler) yalnızca grupları temel alır. IBM MQ 8.0'den EDL' ler hem kullanıcı kimliklerini hem de grupları temel alır ve **SecurityPolicy** özniteliğini Kurulabilir hizmetlerin yapılandırılması ve AIX and Linux üzerinde yetkilendirme hizmeti stanzlarının yapılandırılmasıbaşlıklı konuda açıklandığı gibi uygun değere ayarlayarak yetki için kullanıcı tabanlı modeli ya da grup tabanlı modeli kullanabilirsiniz.

IBM MQ 8.0 ve sonrası için davranıştaki değişiklikler

IBM MQ 8.0' den kullanıcı tabanlı ilkeyle çalışırken, bazı komutlar ürünün önceki sürümlerinden farklı bilgiler döndürür:

- **dmpmqaut** ve **dmpmqcfg** komutları, PCF eşdeğer işlemlerinde olduğu gibi kullanıcı tabanlı kayıtları gösterir.
- IBM MQ Explorer için OAM eklentisi, kullanıcı tabanlı kayıtları gösterir ve kullanıcı tabanlı değişikliklere izin verir.

- OAM **Inquire** işlevi, kullanıcı yeteneğine sahip olduğunu gösteren sonuçları döndürür.

qm.ini dosyasında kullanıcı tabanlı yetkiler etkinleştirildiğinde, **setmqaut** komutunda **-p** özniteliğinin kullanılması, qm.ini dosyasının Hizmet kısmıkısmında açıklandığı gibi aynı birincil gruptaki tüm kullanıcılara erişim vermez.

Kullanıcı tabanlı yetkilendirmeyi kullanmaya başlarsanız ve çok sayıda kullanıcı kullanırsanız, büyük olasılıkla AUTH kuyruğunda grup tabanlı modelden daha fazla kayıt saklanır ve yetkilendirme işlemi, doğrulanacak daha fazla kayıt olduğundan daha önce olduğundan biraz daha uzun sürebilir. Bu artışın önemli olması beklenmiyor. Gerekirse, kullanıcı ve grup izinlerinin bir karışımını kullanabilirsiniz.

Geçişle ilgili önemli noktalar

Var olan bir kuyruk yöneticisi için modeli gruptan kullanıcıya değiştirirseniz, hemen bir etkisi olmaz. Önceden yapılan yetkilendirmeler uygulanmaya devam eder. Kuyruk yöneticisine bağlanan her kullanıcı öncekiyle aynı ayrıcalıkları alır: Kimliğin ait olduğu tüm grupların birleşimi. Kullanıcı kimlikleri için yeni **setmqaut** komutları yayınlandığında, bu komutlar hemen yürürlüğe girer.

Kullanıcı ilkesiyle yeni bir kuyruk yöneticisi yaratırsanız, bu kuyruk yöneticisi yalnızca onu yaratan kullanıcı için izinlere sahip olur (normalde, ancak her zaman değil, mqm kullanıcı kimliği). mqm grubuna otomatik olarak verilen izinler de vardır. Ancak, birincil grup olarak mqm ' e sahip değilseniz, ilk yetki kümesine mqm grubu dahil edilmez.

Bir kullanıcıdan grup ilkesine taşınırsanız, kullanıcı tabanlı yetkilendirmeler otomatik olarak silinmez. Ancak, bunlar artık izin denetimi sırasında kullanılmaz. İlkeyi geri döndürmeden önce, yürürlükteki yapılanışı saklayın, ilkeyi değiştirin, kuyruk yöneticisini yeniden başlatın ve komut dosyasını yeniden yürüt. Artık grup tabanlı bir kuyruk yöneticisi olduğu için, etki, kullanıcı kimliği kurallarının birincil gruba dayalı olarak saklanmasıdır.

İlgili kavramlar

Nesne yetkisi yöneticisi (OAM)

“AIX, Linux, and Windows üzerindeki asıl adlar ve gruplar” sayfa 419

Birincil kullanıcılar gruplara ait olabilir. Kişilere değil, gruplara kaynak erişimi vererek, gereken yönetim miktarını azaltabilirsiniz. Erişim Denetim Listeleri (EDL) hem gruplara, hem de kullanıcı kimliklerine dayalıdır.

İlgili başvurular

qm.ini dosyasının hizmet kısmı

crtmqm (kuyruk yöneticisi yarat) komutu

ALW AIX, Linux, and Windows üzerinde bir IBM MQ nesnesine erişim verilmesi

Kullanıcılara ve kullanıcı gruplarına IBM MQ nesnelerine erişim vermek için **setmqaut** denetim komutunu, **SET AUTHREC** MQSC komutunu ya da **MQCMD_SET_AUTH_REC** PCF komutunu kullanın. IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabildiğinizi unutmayın.

setmqaut denetim komutunun tam tanımlaması ve sözdizimi için bkz. setmqaut.

SET AUTHREC MQSC komutunun tam tanımlaması ve sözdizimi için bkz. SET AUTHREC.

MQCMD_SET_AUTH_REC PCF komutunun tam tanımı ve sözdizimi için Set Authority Recordbaşlıklı konuya bakın.

Bu komutu kullanmak için kuyruk yöneticisi çalışıyor olmalıdır. Bir birincil kullanıcının erişimini değiştirdiğinizde, değişiklikler OAM tarafından hemen yansıtılır.

Kullanıcılara bir nesne için erişim vermek üzere aşağıdakileri belirtmeniz gerekir:

- Çalıştığınız nesnelerin iyesi olan kuyruk yöneticisinin adı; bir kuyruk yöneticisinin adını belirtmezseniz, varsayılan kuyruk yöneticisi varsayılır.

- Nesnenin adı ve tipi (nesneyi benzersiz olarak tanımlamak için). Adı *profil* olarak belirtirsiniz; bu, nesnenin belirtik adı ya da genel arama karakterleri de içinde olmak üzere soysal bir addır. Soysal tanıtımların ayrıntılı açıklaması ve bu tanıtımlarda genel arama karakteri kullanılması için bkz. "AIX, Linux, and Windows üzerinde OAM genel profillerini kullanma" sayfa 371.
- Yetkinin geçerli olduğu bir ya da daha çok birincil kullanıcı ve grup adı.

  Bir kullanıcı kimliği boşluk içeriyorsa, bu komutu kullandığınızda bunu tırnak işareti içine alın. Windows sistemlerinde, bir kullanıcı kimliğini bir etki alanı adıyla niteleyebilirsiniz. Gerçek kullanıcı kimliği bir at işareti (@) simgesi içeriyorsa, bunun kullanıcı kimliği ile etki alanı adı arasındaki sınırlayıcı değil, kullanıcı kimliğinin bir parçası olduğunu göstermek için @ @ ile değiştirin.
- Yetki listesi. Listedeki her öğe, o nesneye verilecek (ya da bu nesneden geri alınacak) bir erişim tipini belirtir. Listedeki her yetki, başında artı işareti (+) ya da eksi işareti (-) olan bir anahtar sözcük olarak belirtilir. Belirtilen yetkiyi eklemek için artı işareti, yetkiyi kaldırmak için eksi işareti kullanın. + ya da-işareti ile anahtar sözcük arasında boşluk olmamalıdır.

  Tek bir komutta istediğiniz sayıda yetki belirtebilirsiniz. Örneğin, bir kullanıcının ya da grubun iletileri kuyruğa koymasına ve bu iletilere göz atmasına izin veren yetkiler listesi, ancak iletileri almak için erişimi iptal etmektir:

```
+browse -get +put
```

`setmqaut` komutunu kullanma örnekleri

Aşağıdaki örneklerde, bir nesneyi kullanma izni vermek ve geri almak için `setmqaut` komutunun nasıl kullanılacağı gösterilmektedir:

```
setmqaut -m saturn.queue.manager -t queue -n RED.LOCAL.QUEUE
         -g groupa +browse -get +put
```

Bu örnekte:

- `saturn.queue.manager` , kuyruk yöneticisi adıdır
- `queue` nesne tipidir
- `RED.LOCAL.QUEUE` nesne adıdır
- `groupa` , değiştirilecek yetkilere sahip grubun tanıtıcısıdır
- `+browse -get +put` , belirtilen kuyruğa ilişkin yetki listesidir
  - `+browse` , kuyruktaki iletilere göz atma yetkisi ekler (göz atma seçeneğiyle **MQGET** komutunu yayınlamak için)
  - `-get` , kuyruktan (**MQGET**) ileti alma yetkisini kaldırır
  - `+put` , (**MQPUT**) iletilerini kuyruğa koyma yetkisi ekler

Aşağıdaki komut, birincil kullanıcı fvuser ve groupa ve groupb gruplarından MyQueue kuyruğuna ilişkin koyma yetkisini iptal eder. AIX and Linux sistemlerinde bu komut, fvuser ile aynı birincil grupta yer alan tüm birincil kullanıcılar için koyma yetkisini de geri çağırır.

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

`setmqaut` komutunu farklı bir yetkilendirme hizmetiyle kullanma

OAM yerine kendi yetkilendirme hizmetinizi kullanıyorsanız, komutu bu hizmete yönlendirmek için **setmqaut** komutunda bu hizmetin adını belirtebilirsiniz. Aynı anda çalışan birden çok kurulabilir bileşeniniz varsa bu parametreyi belirtmeniz gerekir; yoksa, güncelleme yetkilendirme hizmetine ilişkin ilk kurulabilir bileşende yapılır. Varsayılan olarak bu, sağlanan OAM 'dır.

SET AUTHREC için kullanım notları

Eklenecek yetkiler listesi ve kaldırılacak yetkiler listesi çakışmamalıdır. Örneğin, görüntüleme yetkisi ekleyemez ve aynı komutla görüntüleme yetkisini kaldıramazsınız. Bu kural, yetkiler farklı seçenekler kullanılarak ifade edilse bile geçerlidir. Örneğin, DSP yetkisi ALLADM yetkisiyle çakıştığından aşağıdaki komut başarısız olur:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALLADM)
```

Bu çakışma davranışının kural dışı durumu ALL yetkisidir? Aşağıdaki komut önce TÜM yetkileri ekler, daha sonra SETID yetkisini kaldırır:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(ALL) AUTHRMV(SETID)
```

Aşağıdaki komut önce TÜM yetkileri kaldırır ve ardından DSP yetkisini ekler:

```
SET AUTHREC PROFILE(*) OBJTYPE(QUEUE) PRINCIPAL(PRINC01) AUTHADD(DSP) AUTHRMV(ALL)
```

Komutta hangi sırada sağlandıklarına bakılmaksızın, önce ALL işlenir.

ALW AIX, Linux, and Windows üzerinde OAM genel profillerini kullanma

Tek bir işlemde, bir kullanıcının birçok nesne için ayrıcalıklarını ayarlamak için OAM soysal tanıtımlarını kullanın; yaratıldığında her bir nesne için ayrı **setmqaut** komutları ya da **SET AUTHREC** komutları yayınlamak yerine. IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabildiğinizi unutmayın.

setmqaut ya da SET AUTHREC komutlarındaki soysal tanıtımları kullanarak, o tanıtıma uyan tüm nesneler için soysal bir yetki belirlemenizi sağlar.

Bu konu derlemi, soysal tanıtımların daha ayrıntılı olarak kullanılmasını açıklar.

OAM profillerinde genel arama karakterlerini kullanma

Bir tanıtımı soysal yapan şey, tanıtım adında özel karakterlerin (genel arama karakterleri) kullanılmasıdır. Örneğin, soru işareti (?) genel arama karakteri, bir addaki herhangi bir tek karakterle eşleşir. Bu nedenle, `ABC.?EF`değerini belirlerseniz, o tanıtıma ilişkin yetki `ABC.DEF`, `ABC.CEF`, `ABC.BEF`gibi adlara sahip nesneler için de geçerlidir.

Kullanılabilecek genel arama karakterleri şunlardır:

?

Herhangi bir tek karakter yerine soru işaretini (?) kullanın. Örneğin, `AB.?D` , `AB.CD`, `AB.ED`ve `AB.FD`nesneleri için geçerlidir.

*****

Yıldız işaretini (*) aşağıdaki gibi kullanın:

- Bir profil adında, nesne adındaki herhangi bir niteleyiciyle eşleşecek *niteleyici* . Niteleyici, bir nokta ile sınırlanmış bir nesne adının parçasıdır. Örneğin, `ABC.DEF.GHI`içinde niteleyiciler `ABC`, `DEF`ve `GHI`' dir.

  Örneğin, `ABC.*.JKL` , `ABC.DEF.JKL`ve `ABC.GHI.JKL`nesneleri için geçerlidir. (`ABC.JKL` için geçerli **olmadığını** unutmayın; * bu bağlamda kullanılan her zaman bir niteleyiciyi gösterir.)
- Bir profil adındaki niteleyici içindeki, bir nesne adındaki niteleyici içindeki sıfır ya da daha fazla karakterle eşleşecek karakter.

  Örneğin, `ABC.DE*.JKL` , `ABC.DE.JKL`, `ABC.DEF.JKL`ve `ABC.DEGH.JKL`nesneleri için geçerlidir.

******

Profil adında çift yıldız işaretini (**) **bir kez** kullanın:

- Tüm nesne adlarıyla eşleşecek tüm profil adı. Örneğin, süreçleri tanımlamak için `-t prcs` , tanıtım adı olarak ** kullanırsanız, tüm süreçlere ilişkin yetkileri değiştirirsiniz.
- Bir tanıtım adında, nesne adında sıfır ya da daha fazla niteleyiciyle eşleşecek başlangıç, ikinci ya da bitiş niteleyicisi olarak. Örneğin, `**.`ABC son niteleyicisi ABC olan tüm nesneleri tanıtır.

Çift yıldız işaretini ** yalnızca tam niteleyici olarak kullanabilirsiniz:

```
**.DEF
  ABC.**
  A*.**
```

ancak bu şekilde değil

```
A**
```

Aksi takdirde AMQ7226E: `Profil adı geçersiz.`

Not: AIX and Linux sistemlerinde genel arama karakterlerini kullanırken, profil adını tek tırnak içine almanız **gerekir** .

Profil öncelikleri

Soysal tanıtımlar kullanılırken anlaşılması gereken önemli bir nokta, yaratılmakta olan bir nesneye uygulanacak yetkiler belirlenirken tanıtımların verildiği önceliktir. Örneğin, şu komutları yayınladığınızı varsayalım:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

İlki, adları AB. * tanıtımıyla eşleşen birincil kullanıcı Fred için tüm kuyruklara koyma yetkisi verir; İkincisi, AB.C*.

Şimdi AB.CD. Joker karakter eşleştirmeye ilişkin kurallara göre, her iki setmqaut bu kuyruğa uygulanabilir. Yani, otoriteye sahip mi?

Yanıtı bulmak için, bir nesneye birden çok profil uygulanabildiğinde **yalnızca en özel profilin geçerli olduğu**kuralını uygulayabilirsiniz. Bu kuralı uygulama şekliniz, profil adlarını soldan sağa karşılaştırmaktır. Farklı oldukları her yerde, soysal olmayan bir karakter soysal bir karakterden daha belirlidir. Bu örnekte, kuyruk AB.CD ' nin **get** yetkisi (AB.C*, AB. * ' den daha özeldir).

Soysal karakterleri karşılaştırırken *özgüllük* sırası şöyledir:

1. ?
2. *
3. **

Profil ayarlarının dökümü yapılıyor

dmpmqaut denetim komutunun tam tanımlaması ve sözdizimi için bkz. dmpmqaut.

DISPLAY AUTHREC MQSC komutunun tam tanımlaması ve sözdizimi için DISPLAY AUTHRECbaşlıklı konuya bakın.

MQCMD_INQUIRE_AUTH_RECS PCF komutunun tam tanımı ve sözdizimi için Inquire Authority Recordsbaşlıklı konuya bakın.

Aşağıdaki örneklerde, soysal tanıtımlara ilişkin yetki kayıtlarının dökümünü almak için **dmpmqaut** denetim komutunun kullanımı gösterilmektedir:

1. Bu örnek, birincil kullanıcı user1için a.b.c kuyruğuyla eşleşen bir tanıtımla tüm yetki kayıtlarının dökümünü sağlar.

```
dmpmqaut -m qm1 -n a.b.c -t q -p user1
```

Ortaya çıkan çöplük şöyle bir şeye benziyor:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

Not: AIX and Linux üzerindeki kullanıcılar **dmpmqaut** komutu için -p seçeneğini kullanabilseler de, yetkiler tanımlarken -g groupname komutunu kullanmaları gerekir.

2. Bu örnek, a.b.ckuyruğuyla eşleşen bir tanıtımla tüm yetki kayıtlarının dökümünü sağlar.

```
dmpmqaut -m qmgr1 -n a.b.c -t q
```

Ortaya çıkan çöplük şöyle bir şeye benziyor:

```
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. Bu örnek, a.b. * profili için tüm yetki kayıtlarının dökümünü sağlar. kuyruk tipinde.

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

Ortaya çıkan çöplük şöyle bir şeye benziyor:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. Bu örnek, qmXkuyruk yöneticisine ilişkin tüm yetki kayıtlarının dökümünü oluşturur.

```
dmpmqaut -m qmX
```

Ortaya çıkan çöplük şöyle bir şeye benziyor:

```
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. Bu örnek, qmXkuyruk yöneticisi için tüm profil adlarının ve nesne tiplerinin dökümünü alır.

```
dmpmqaut -m qmX -l
```

Ortaya çıkan çöplük şöyle bir şeye benziyor:

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

Not: Yalnızca IBM MQ for Windows için, görüntülenen tüm birincil kullanıcılar etki alanı bilgilerini içerir; örneğin:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW *AIX, Linux, and Windows üzerinde OAM profillerinde genel arama karakterlerini kullanma*

Bir tanıtımın birden çok nesne için geçerli olmasını sağlamak üzere bir nesne yetki yöneticisi (OAM) tanıtımı adında genel arama karakterleri kullanın.

Bir tanıtımı soysal yapan şey, tanıtım adında özel karakterlerin (genel arama karakterleri) kullanılmasıdır. Örneğin, soru işareti (?) genel arama karakteri, bir addaki herhangi bir tek karakterle eşleşir. Bu nedenle, `ABC.?EF`değerini belirlerseniz, o tanıtıma ilişkin yetki `ABC.DEF`, `ABC.CEF`, `ABC.BEF`gibi adlara sahip nesneler için de geçerlidir.

Kullanılabilecek genel arama karakterleri şunlardır:

?

Herhangi bir tek karakter yerine soru işaretini (?) kullanın. Örneğin, `AB.?D` , `AB.CD`, `AB.ED`ve `AB.FD`nesneleri için geçerlidir.

*****

Yıldız işaretini (*) aşağıdaki gibi kullanın:

- Bir profil adında, nesne adındaki herhangi bir niteleyiciyle eşleşecek *niteleyici* . Niteleyici, bir nokta ile sınırlanmış bir nesne adının parçasıdır. Örneğin, `ABC.DEF.GHI`içinde niteleyiciler şunlardır: `ABC`, `DEF`ve `GHI`.

  Örneğin, `ABC.*.JKL` , `ABC.DEF.JKL`ve `ABC.GHI.JKL`nesneleri için geçerlidir. (`ABC.JKL` için geçerli **olmadığını** unutmayın; * bu bağlamda kullanılan her zaman bir niteleyiciyi gösterir.)
- Bir profil adındaki niteleyici içindeki, bir nesne adındaki niteleyici içindeki sıfır ya da daha fazla karakterle eşleşecek karakter.

  Örneğin, `ABC.DE*.JKL` , `ABC.DE.JKL`, `ABC.DEF.JKL`ve `ABC.DEGH.JKL`nesneleri için geçerlidir.

******

Profil adında çift yıldız işaretini (**) **bir kez** kullanın:

- Tüm nesne adlarıyla eşleşecek tüm profil adı. Örneğin, süreçleri tanımlamak için `-t prcs` , tanıtım adı olarak ** kullanırsanız, tüm süreçlere ilişkin yetkileri değiştirirsiniz.
- Bir tanıtım adında, nesne adında sıfır ya da daha fazla niteleyiciyle eşleşecek başlangıç, ikinci ya da bitiş niteleyicisi olarak. Örneğin, `**.ABC` son niteleyicisi ABC olan tüm nesneleri tanıtır.

Not: AIX and Linux sistemlerinde genel arama karakterlerini kullanırken, profil adını tek tırnak içine almanız **gerekir** .

ALW *AIX, Linux, and Windows ile ilgili profil öncelikleri*

Tek bir nesneye birden çok soysal tanıtım uygulanabilir. Bu durumda, en özel kural geçerlidir.

Soysal tanıtımlar kullanılırken anlaşılması gereken önemli bir nokta, yaratılmakta olan bir nesneye uygulanacak yetkiler belirlenirken tanıtımların verildiği önceliktir. Örneğin, şu komutları yayınladığınızı varsayalım:

```
setmqaut -n AB.* -t q +put -p fred
setmqaut -n AB.C* -t q +get -p fred
```

İlki, adları AB. * tanıtımıyla eşleşen birincil kullanıcı Fred için tüm kuyruklara koyma yetkisi verir; İkincisi, AB.C*.

Şimdi AB.CD. Joker karakter eşleştirmeye ilişkin kurallara göre, her iki setmqaut bu kuyruğa uygulanabilir. Yani, otoriteye sahip mi?

Yanıtı bulmak için, bir nesneye birden çok profil uygulanabildiğinde **yalnızca en özel profilin geçerli olduğu**kuralını uygulayabilirsiniz. Bu kuralı uygulama şekliniz, profil adlarını soldan sağa karşılaştırmaktır. Farklı oldukları her yerde, soysal olmayan bir karakter soysal bir karakterden daha belirlidir. Bu örnekte, kuyruk AB.CD ' nin **get** yetkisi (AB.C*, AB. * ' den daha özeldir).

Soysal karakterleri karşılaştırırken *özgüllük* sırası şöyledir:

1. ?
2. *
3. **

Bu MQSC komutunu kullanırken eşdeğer bilgiler için SET AUTHREC konusuna bakın.

ALW *AIX, Linux, and Windows ' da profil ayarlarının dökümü yapılıyor*

Belirtilen bir tanıtımla ilişkili yürürlükteki yetkilerin dökümünü almak için **dmpmqaut** denetim komutunu, **DISPLAY AUTHREC** MQSC komutunu ya da **MQCMD_INQUIRE_AUTH_RECS** PCF komutunu kullanın. IBM MQ Appliance üzerinde yalnızca **DISPLAY AUTHREC** komutunu kullanabildiğinizi unutmayın.

dmpmqaut denetim komutunun tam tanımlaması ve sözdizimi için bkz. dmpmqaut.

DISPLAY AUTHREC MQSC komutunun tam tanımlaması ve sözdizimi için DISPLAY AUTHRECbaşlıklı konuya bakın.

MQCMD_INQUIRE_AUTH_RECS PCF komutunun tam tanımı ve sözdizimi için Inquire Authority Recordsbaşlıklı konuya bakın.

Aşağıdaki örneklerde, soysal tanıtımlara ilişkin yetki kayıtlarının dökümünü almak için **dmpmqaut** denetim komutunun kullanımı gösterilmektedir:

1. Bu örnek, birincil kullanıcı user1için a.b.c kuyruğuyla eşleşen bir tanıtımla tüm yetki kayıtlarının dökümünü sağlar.

   ```
   dmpmqaut -m qm1 -n a.b.c -t q -p user1
   ```

   Ortaya çıkan döküm şu örneğe benzer:

   ```
   profile:     a.b.*
   object type: queue
   ```

```
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

Not: AIX and Linux kullanıcıları `-p` seçeneğini kullanamaz; bunun yerine `-g groupname` kullanmalıdır.

2. Bu örnek, a.b.ckuyruğuyla eşleşen bir tanıtımla tüm yetki kayıtlarının dökümünü sağlar.

```
dmpmqaut -m qmgr1 -n a.b.c -t q
```

Ortaya çıkan döküm şu örneğe benzer:

```
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. Bu örnek, a.b. * profili için tüm yetki kayıtlarının dökümünü sağlar. kuyruk tipinde.

```
dmpmqaut -m qmgr1 -n a.b.* -t q
```

Ortaya çıkan döküm şu örneğe benzer:

```
profile:     a.b.*
object type: queue
entity:      user1
type:        principal
authority:   get, browse, put, inq
```

4. Bu örnek, qmXkuyruk yöneticisine ilişkin tüm yetki kayıtlarının dökümünü oluşturur.

```
dmpmqaut -m qmX
```

Ortaya çıkan döküm şu örneğe benzer:

```
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. Bu örnek, qmXkuyruk yöneticisi için tüm profil adlarının ve nesne tiplerinin dökümünü alır.

```
dmpmqaut -m qmX -l
```

Ortaya çıkan döküm şu örneğe benzer:

```
profile: q1, type: queue
profile: q*, type: queue
profile: name.*, type: namelist
profile: pr1, type: process
```

Not: Yalnızca IBM MQ for Windows için, görüntülenen tüm birincil kullanıcılar etki alanı bilgilerini içerir; örneğin:

```
profile:     a.b.*
object type: queue
entity:      user1@domain1
type:        principal
authority:   get, browse, put, inq
```

ALW AIX, Linux, and Windows üzerinde erişim ayarlarının görüntülenmesi

Belirli bir birincil kullanıcının ya da grubun belirli bir nesne için sahip olduğu yetkileri görüntülemek için **dspmqaut** denetim komutunu, **DISPLAY AUTHREC** MQSC komutunu ya da **MQCMD_INQUIRE_ENTITY_AUTH** PCF komutunu kullanın. IBM MQ Appliance üzerinde yalnızca **DISPLAY AUTHREC** komutunu kullanabildiğinizi unutmayın.

Bu komutu kullanmak için kuyruk yöneticisi çalışıyor olmalıdır. Bir birincil kullanıcının erişimini değiştirdiğinizde, değişiklikler OAM tarafından hemen yansıtılır. Bir kerede yalnızca bir grup ya da birincil kullanıcı için yetki görüntülenebilir.

dmpmqaut denetim komutunun tam tanımlaması ve sözdizimi için bkz. dmpmqaut.

DISPLAY AUTHREC MQSC komutunun tam tanımlaması ve sözdizimi için DISPLAY AUTHRECbaşlıklı konuya bakın.

MQCMD_INQUIRE_AUTH_RECS PCF komutunun tam tanımı ve sözdizimi için Inquire Authority Recordsbaşlıklı konuya bakın.

Aşağıdaki örnekte, GpAdmin grubunun Annuities kuyruk yöneticisinde QueueMan1adlı bir süreç tanımlaması için sahip olduğu yetkileri görüntülemek üzere **dspmqaut** denetim komutunun kullanımı gösterilmektedir.

```
dspmqaut -m QueueMan1 -t process -n Annuities -g GpAdmin
```

ALW AIX, Linux, and Windows üzerinde bir IBM MQ nesnesine erişimin değiştirilmesi ve geri çekilmesi

Bir kullanıcının ya da grubun bir nesneye erişim düzeyini değiştirmek için **setmqaut** denetim komutunu, **DELETE AUTHREC** MQSC komutunu ya da **MQCMD_DELETE_AUTH_REC** PCF komutunu kullanın.

MQ Appliance IBM MQ Appliance üzerinde yalnızca **DELETE AUTHREC** komutunu kullanabildiğinizi unutmayın.

Kullanıcıyı gruptan kaldırma işlemi aşağıda açıklanmıştır:

- Windows “Windows üzerinde grup oluşturma ve yönetme” sayfa 145

- AIX “AIX üzerinde grup oluşturma ve yönetme” sayfa 143
- Linux “Linux üzerinde grup oluşturma ve yönetme” sayfa 144

.

Bir IBM MQ nesnesi oluşturan kullanıcı kimliğine o nesne için tam denetim yetkileri verilir. Bu kullanıcı kimliğini yerel mqm grubundan (ya da Windows sistemlerinde Denetimciler (Administrators) grubundan) kaldırırsanız, bu yetkiler iptal edilmez. **setmqaut** denetim komutunu ya da **MQCMD_DELETE_AUTH_REC** PCF komutunu kullanarak, nesneyi yaratan kullanıcı kimliği için, nesneyi mqm ya da Administrators (Yöneticiler) grubundan kaldırdıktan sonra nesneye erişimi iptal edin.

setmqaut denetim komutunun tam tanımlaması ve sözdizimi için bkz. setmqaut.

DELETE AUTHREC MQSC komutunun tam tanımlaması ve sözdizimi için bkz. DELETE AUTHREC.

MQCMD_DELETE_AUTH_REC PCF komutunun tam tanımı ve sözdizimi için Yetki Kaydının Silinmesibaşlıklı konuya bakın.

Windows Windowsüzerinde, IBM MQ 8.0içinden, **setmqaut -u** *SID* parametresini kullanarak belirli bir Windows kullanıcı hesabına karşılık gelen OAM girdilerini istediğiniz zaman silebilirsiniz.

IBM MQ 8.0' den önce, kullanıcı profilini silmeden önce belirli bir Windows kullanıcı hesabına karşılık gelen OAM girdilerini silmeniz gerekiyordu. Kullanıcı hesabını kaldırdıktan sonra OAM girdilerini kaldırmak imkansızdı.

ALW AIX, Linux, and Windows sistemlerinde güvenlik erişimi denetimlerinin önlenmesi

Not: Bu konu, etkinleştirilmesi önerilmeyen işlevleri açıklar. Güvenlik denetimini kapatmak için nesne yetki yöneticisini (OAM) devre dışı bırakabilirsiniz. Bu, bir test ortamı için uygun olabilir. Geçersiz kılındığında, kuyruk yöneticisi artık yetkilendirme ya da bağlantı kimlik doğrulaması denetimleri gerçekleştiremez. TLS, Kanal Kimlik Doğrulaması kayıtları ve güvenlik çıkışları hala kullanılabilir. OAM geçersiz kılındıktan ya da kaldırıldıktan sonra, var olan bir kuyruk yöneticisine OAM ekleyemezsiniz.

Güvenlik denetimlerini gerçekleştirmek istemediğinize karar verirseniz (örneğin, bir test ortamında), OAM ' yi iki yoldan biriyle devre dışı bırakabilirsiniz:

- Bir kuyruk yöneticisi yaratmadan önce, **MQSNOAUT**işletim sistemi ortam değişkenini ayarlayın.

  MQSNOAUT ortam değişkeninin ayarlanmasının etkileri ve AIX, Linux, and Windowsüzerinde **MQSNOAUT** ' un nasıl ayarlandığı hakkında bilgi için bkz. Ortam değişkenleri açıklamaları.
- Hizmeti kaldırmak için kuyruk yöneticisi yapılanış dosyasını düzenleyin.

Uyarı: Bir OAM kaldırıldığında, var olan bir kuyruk yöneticisine geri konamaz. Bunun nedeni, OAM ' nin nesne oluşturma sırasında yerinde olması gerekolmasıdır. IBM MQ OAM kaldırıldıktan sonra yeniden kullanmak için kuyruk yöneticisini yeniden oluşturun.

OAM devre dışı bırakıldığında **setmqaut**ya da **dspmqaut** komutunu kullanırsanız, aşağıdaki noktalara dikkat edin:

- OAM, belirtilen birincil kullanıcının ya da grubun geçerliliğini denetlemez; bu, komutun geçersiz değerleri kabul edebileceği anlamına gelir.
- OAM, güvenlik denetimleri gerçekleştirmez ve tüm birincil kullanıcı ve grupların tüm geçerli nesne işlemlerini gerçekleştirme yetkisine sahip olduğunu gösterir.
- Kimlik doğrulama denetimleri için OAM ' ye iletilen kimlik bilgileri doğrulanmaz.

İlgili kavramlar

AIX, Linux, and Windows için kurulabilir hizmetler ve bileşenler

İlgili görevler

Kurulabilir hizmetlerin yapılandırılması

İlgili başvurular

Kurulabilir hizmetler için başvuru bilgileri

Kaynaklara gerekli erişim verilmesi

IBM MQ sisteminize güvenlik uygulamak için gerçekleştirilecek görevleri belirlemek üzere bu konuyu kullanın.

Bu görev hakkında

Bu görev sırasında, IBM MQ kuruluşunuzun öğelerine uygun güvenlik düzeyini uygulamak için hangi işlemlerin gerekli olduğuna karar verirsiniz. Başvurulan her bir görev, tüm platformlar için adım adım yönergeler sağlar.

Yordam

1. Kuyruk yöneticinize erişimi belirli kullanıcılarla sınırlandırmanız gerekiyor mu?
   a) Hayır: Daha fazla işlem yapma.
   b) Evet, bir sonraki soruya geçin.
2. Bu kullanıcıların bir kuyruk yöneticisi kaynakları alt kümesinde kısmi denetimci erişimine gereksinimi var mı?
   a) Hayır: Bir sonraki soruya git.
   b) Evet: Bkz. “Kuyruk yöneticisi kaynaklarının bir alt kümesine kısmi denetim erişimi verilmesi” sayfa 379.
3. Bu kullanıcıların bir kuyruk yöneticisi kaynakları alt kümesinde tam denetimci erişimine gereksinimi var mı?
   a) Hayır: Bir sonraki soruya git.
   b) Evet: Bkz. “Kuyruk yöneticisi kaynaklarının bir alt kümesinde tam denetim erişimi verilmesi” sayfa 388.
4. Bu kullanıcıların tüm kuyruk yöneticisi kaynaklarına salt okunur erişmeleri gerekiyor mu?
   a) Hayır: Bir sonraki soruya git.
   b) Evet: Bkz. “Kuyruk yöneticisindeki tüm kaynaklara salt okunur erişim verilmesi” sayfa 394.
5. Bu kullanıcıların tüm kuyruk yöneticisi kaynaklarında tam denetimci erişimine gereksinimi var mı?
   a) Hayır: Bir sonraki soruya git.
   b) Evet: Bkz. “Bir kuyruk yöneticisindeki tüm kaynaklara tam denetim erişimi verilmesi” sayfa 395.
6. Kuyruk yöneticinize bağlanmak için kullanıcı uygulamalarına gerek var mı?
   a) Hayır: “Kuyruk yöneticisine bağlanırlığın kaldırılması” sayfa 396 içinde açıklandığı gibi bağlanırlığı devre dışı bırakın
   b) Evet: Bkz. “Kullanıcı uygulamalarının kuyruk yöneticinizle bağlantı kurmasına izin verme” sayfa 397.

z/OS Multi Kuyruk yöneticisi kaynaklarının bir alt kümesine kısmi denetim erişimi verilmesi

Belirli kullanıcılara kuyruk yöneticisi kaynaklarının tümü değil, bazıları için kısmi denetim erişimi vermeniz gerekir. Gerçekleştirmeniz gereken işlemleri saptamak için bu çizelgeyi kullanın.

Çizelge 72. Kuyruk yöneticisi kaynaklarının bir altkümesine kısmi denetim erişimi verilmesi	
Kullanıcıların bu tipteki nesneleri denetmeleri gerekir	**Bu işlemi gerçekleştir**
Kuyruklar	"Bazı kuyruklara sınırlı denetim erişimi verilmesi" sayfa 380 başlıklı konuda açıklandığı gibi, gerekli kuyruklara kısmi yönetici erişimi verme
Konular	"Bazı konulara sınırlı yönetici erişimi verilmesi" sayfa 381 başlıklı konuda açıklandığı gibi, gerekli konulara kısmi yönetici erişimi verme
Kanallar	"Bazı kanallara sınırlı yönetici erişimi verilmesi" sayfa 382 başlıklı konuda açıklandığı gibi, gerekli kanallara kısmi yönetici erişimi verin
Kuyruk yöneticisi	"Bir kuyruk yöneticisine sınırlı denetim erişimi verilmesi" sayfa 384 konusunda açıklandığı gibi kuyruk yöneticisine kısmi yönetici erişimi verme
Süreçler	"Bazı süreçlere sınırlı yönetici erişimi verilmesi" sayfa 385 başlıklı konuda açıklandığı gibi, gerekli süreçlere kısmi yönetici erişimi verin.
Namelistler	"Bazı namelistlere sınırlı yönetici erişimi verilmesi" sayfa 386 başlıklı konuda açıklandığı gibi, gerekli ad bilgilerine kısmi yönetici erişimi verin
Hizmetler	"Bazı hizmetlere sınırlı yönetici erişimi verilmesi" sayfa 387 başlıklı konuda açıklandığı gibi, gerekli hizmetlere kısmi yönetici erişimi verin

Bazı kuyruklara sınırlı denetim erişimi verilmesi

Bir kuyruk yöneticisindeki bazı kuyruklara, iş gereksinimi olan her kullanıcı grubuna kısmi denetim erişimi verin.

Bu görev hakkında

Bazı işlemlere ilişkin bazı kuyruklara sınırlı denetim erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW
  AIX, Linux, and Windows sistemleri için aşağıdaki komutu verin:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName ReqdAction
  ```

- IBM i
  IBM iiçin şu komutu verin:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS z/OSiçin, belirtilen bir kuyruğa erişim vermek üzere aşağıdaki komutları çalıştırın:

```
RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Kullanıcının kuyrukta gerçekleştirebileceği MQSC komutlarını belirtmek için, her MQSC komutu için aşağıdaki komutları verin:

```
RDEFINE MQCMDS QMgrName. ReqdAction. QType UACC(NONE)
PERMIT QMgrName. ReqdAction. QType CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

Kullanıcının DISPLAY QUEUE komutunu kullanmasına izin vermek için aşağıdaki komutları verin:

```
RDEFINE MQCMDS QMgrName.DISPLAY. QType UACC(NONE)
PERMIT QMgrName.DISPLAY. QType CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı.

z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

ObjectProfile
Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

ReqdAction
Grubun gerçekleştirmesine izin verdiğiniz işlem:

- ALW AIX, Linux, and Windows sistemlerinde şu yetkilerin herhangi bir birleşimi: + chg, + clr, + dlt, + dsp. Yetkilendirme + alladm, + chg + clr + dlt + dsp ile eşdeğerdir.
- IBM i IBM iüzerinde, şu yetkilerin herhangi bir birleşimi: *ADMCHG, *ADMCLR, *ADMDLT, *ADMDSP. *ALLADM yetkisi, tüm bu bireysel yetkilere eşdeğerdir.
- z/OS z/OSüzerinde, ALTER, CLEAR, DELETE ya da MOVE değerlerinden biri.

Not: Kuyruklar için + crt verilmesi, kullanıcıyı ya da grubu dolaylı olarak denetimci yapar. Bazı kuyruklara sınırlı yönetim erişimi vermek için + crt yetkisini kullanmayın.

QTür

DISPLAY komutu için, QUEUE, QLOCAL, QALIAS, QMODEL, QREMOTE ya da QCLUSTER değerlerinden biri.

Diğer *ReqdAction*değerleri için, QLOCAL, QALIAS, QMODEL ya da QREMOTE değerlerinden biri.

Bazı konulara sınırlı yönetici erişimi verilmesi

Bir kuyruk yöneticisindeki bazı konulara, iş gereksinimi olan her kullanıcı grubuna kısmi yönetici erişimi verin.

Bu görev hakkında

Bazı işlemlere ilişkin bazı konulara sınırlı yönetici erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW
  AIX, Linux, and Windows sistemleri için aşağıdaki komutu verin:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName ReqdAction
```

- IBM i
  IBM iiçin şu komutu verin:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Bu komutlar, belirtilen konuya erişim izni verir. Kullanıcının konu üzerinde gerçekleştirebileceği MQSC komutlarını saptamak için, her MQSC komutu için aşağıdaki komutları verin:

```
RDEFINE MQCMDS QMgrName. ReqdAction.TOPIC UACC(NONE)
PERMIT QMgrName. ReqdAction.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Kullanıcının DISPLAY TOPIC komutunu kullanmasına izin vermek için aşağıdaki komutları verin:

```
RDEFINE MQCMDS QMgrName.DISPLAY.TOPIC UACC(NONE)
PERMIT QMgrName.DISPLAY.TOPIC CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

  ReqdAction
  Grubun gerçekleştirmesine izin verdiğiniz işlem:

  – ALW AIX, Linux, and Windows sistemlerinde şu yetkilerin herhangi bir birleşimi: + chg, + clr, + crt, + dlt, + dsp. + ctrl. Yetkilendirme + alladm, + chg + clr + dlt + dsp ile eşdeğerdir.

  – IBM i IBM iüzerinde, şu yetkilerin herhangi bir birleşimi: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL. *ALLADM yetkisi, tüm bu bireysel yetkilere eşdeğerdir.

  – z/OS z/OSüzerinde, ALTER, CLEAR, DEFINE, DELETE ya da MOVE değerlerinden biri.

Bazı kanallara sınırlı yönetici erişimi verilmesi

Bir kuyruk yöneticisindeki bazı kanallara, iş gereksinimi olan her bir kullanıcı grubuna kısmi yönetici erişimi verin.

Bu görev hakkında

Bazı işlemlere ilişkin bazı kanallara sınırlı yönetici erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW

  AIX, Linux, and Windows'ta:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName ReqdAction
  ```

- IBM i

  IBM i'ta:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS z/OS'ta:

  ```
  RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
  PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Bu komutlar, belirtilen kanala erişim verir. Kullanıcının kanal üzerinde gerçekleştirebileceği MQSC komutlarını saptamak için, her MQSC komutu için aşağıdaki komutları verin:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.CHANNEL UACC(NONE)
  PERMIT QMgrName. ReqdAction.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Kullanıcının DISPLAY CHANNEL komutunu kullanmasına izin vermek için aşağıdaki komutları verin:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.CHANNEL UACC(NONE)
  PERMIT QMgrName.DISPLAY.CHANNEL CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

  ReqdAction
  Grubun gerçekleştirmesine izin verdiğiniz işlem:

  – ALW AIX, Linux, and Windowsüzerinde, şu yetkilerin herhangi bir birleşimi: + chg, + clr, + crt, + dlt, + dsp. + ctrl, + ctrlx. Yetkilendirme + alladm, + chg + clr + dlt + dsp ile eşdeğerdir.

  – IBM i IBM iişletim sisteminde şu yetkilerin herhangi bir birleşimi bulunur: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLx. *ALLADM yetkisi, tüm bu bireysel yetkilere eşdeğerdir.

– z/OS z/OSüzerinde, ALTER, CLEAR, DEFINE, DELETE ya da MOVE değerlerinden biri.

Bir kuyruk yöneticisine sınırlı denetim erişimi verilmesi

Bir kuyruk yöneticisine, iş gereksinimi olan her bir kullanıcı grubuna kısmi yönetici erişimi verin.

Bu görev hakkında

Kuyruk yöneticisinde bazı işlemler gerçekleştirmek üzere sınırlı denetim erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW

  AIX, Linux, and Windows'ta:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName ReqdAction
  ```

- IBM i

  IBM i'ta:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS

  z/OS'ta:

  Kuyruk yöneticisinde gerçekleştirebileceğiniz MQSC komutlarını saptamak için, her MQSC komutu için aşağıdaki komutları verin:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.QMGR UACC(NONE)
  PERMIT QMgrName. ReqdAction.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Kullanıcının DISPLAY QMGR komutunu kullanmasına izin vermek için aşağıdaki komutları verin:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.QMGR UACC(NONE)
  PERMIT QMgrName.DISPLAY.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

  ReqdAction
  Grubun gerçekleştirmesine izin verdiğiniz işlem:

  – ALW AIX, Linux, and Windowsüzerinde, şu yetkilerin herhangi bir birleşimi: + chg, + clr, + crt, + dlt, + dsp. Yetkilendirme + alladm, + chg + clr + dlt + dsp ile eşdeğerdir.

  + kümesi bir MQI yetkisi olmasına ve normalde denetim olarak kabul edilmemesine rağmen, kuyruk yöneticisinde + kümesi verilmesi dolaylı olarak tam denetim yetkisine yol açabilir. Sıradan kullanıcılara ve uygulamalara + ayarlama izni vermeyin.

– IBM i IBM iüzerinde, şu yetkilerin herhangi bir birleşimi: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. *ALLADM yetkisi, tüm bu bireysel yetkilere eşdeğerdir.

Bazı süreçlere sınırlı yönetici erişimi verilmesi

Bir kuyruk yöneticisindeki bazı süreçlere, iş gereksinimi olan her bir kullanıcı grubuna kısmi yönetici erişimi verin.

Bu görev hakkında

Bazı işlemlere ilişkin bazı işlemlere sınırlı denetim erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW
  AIX, Linux, and Windows'ta:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName ReqdAction
```

- IBM i
  IBM i'ta:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS z/OS'ta:

```
RDEFINE MQADMIN QMgrName.PROCESS. ObjectProfile UACC(NONE)
PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Bu komutlar, belirtilen kanala erişim verir. Kullanıcının kanal üzerinde gerçekleştirebileceği MQSC komutlarını saptamak için, her MQSC komutu için aşağıdaki komutları verin:

```
RDEFINE MQCMDS QMgrName. ReqdAction.PROCESS UACC(NONE)
PERMIT QMgrName. ReqdAction.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Kullanıcının DISPLAY PROCESS komutunu kullanmasına izin vermek için aşağıdaki komutları verin:

```
RDEFINE MQCMDS QMgrName.DISPLAY.PROCESS UACC(NONE)
PERMIT QMgrName.DISPLAY.PROCESS CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

  ReqdAction
  Grubun gerçekleştirmesine izin verdiğiniz işlem:

– ALW AIX, Linux, and Windowsüzerinde, şu yetkilerin herhangi bir birleşimi: + chg, + clr, + crt, + dlt, + dsp. Yetkilendirme + alladm, + chg + clr + dlt + dsp ile eşdeğerdir.

– IBM i IBM iüzerinde, şu yetkilerin herhangi bir birleşimi: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP. *ALLADM yetkisi, tüm bu bireysel yetkilere eşdeğerdir.

– z/OS z/OSüzerinde, ALTER, CLEAR, DEFINE, DELETE ya da MOVE değerlerinden biri.

Bazı namelistlere sınırlı yönetici erişimi verilmesi

Bir kuyruk yöneticisindeki bazı ad bilgilerine, iş gereksinimi olan her kullanıcı grubuna kısmi yönetici erişimi verin.

Bu görev hakkında

Bazı işlemler için bazı ad bilgilerine sınırlı yönetici erişimi vermek üzere işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW

  AIX, Linux, and Windows'ta:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName ReqdAction
  ```

- IBM i

  IBM i'ta:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(ReqdAction) MQMNAME('
  QMgrName ')
  ```

- z/OS z/OS'ta:

  ```
  RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
  PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Bu komutlar, belirtilen ad listesi için erişim verir. Kullanıcının ad listesi üzerinde gerçekleştirebileceği MQSC komutlarını saptamak için, her MQSC komutu için aşağıdaki komutları verin:

  ```
  RDEFINE MQCMDS QMgrName. ReqdAction.NAMELIST UACC(NONE)
  PERMIT QMgrName. ReqdAction.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
  ```

  Kullanıcının DISPLAY NAMELIST komutunu kullanmasına izin vermek için aşağıdaki komutları verin:

  ```
  RDEFINE MQCMDS QMgrName.DISPLAY.NAMELIST UACC(NONE)
  PERMIT QMgrName.DISPLAY.NAMELIST CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
  ```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

ReqdAction
Grubun gerçekleştirmesine izin verdiğiniz işlem:

- ALW AIX, Linux, and Windowsüzerinde, şu yetkilerin herhangi bir birleşimi: + chg, + clr, + crt, + dlt, + ctrl, + ctrlx, + dsp. Yetkilendirme + alladm, + chg + clr + dlt + dsp ile eşdeğerdir.
- IBM i IBM iüzerinde, şu yetkilerin herhangi bir birleşimi vardır: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. *ALLADM yetkisi, tüm bu bireysel yetkilere eşdeğerdir.
- z/OS z/OSüzerinde, ALTER, CLEAR, DEFINE, DELETE ya da MOVE değerlerinden biri.

Bazı hizmetlere sınırlı yönetici erişimi verilmesi

Bir kuyruk yöneticisindeki bazı hizmetlere, iş gereksinimi olan her bir kullanıcı grubuna kısmi yönetici erişimi verin.

Bu görev hakkında

Bazı işlemlere ilişkin bazı hizmetlere sınırlı yönetici erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın. z/OS Hizmet nesnelerinin z/OSüzerinde var olmadığını unutmayın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW
  AIX, Linux, and Windows'ta:

```
setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName ReqdAction
```

- IBM i'ta:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(ReqdAction) MQMNAME('
QMgrName ')
```

- z/OS z/OS'ta:

  Bu komutlar, belirtilen hizmete erişim verir. Kullanıcının hizmet üzerinde gerçekleştirebileceği MQSC komutlarını saptamak için, her MQSC komutu için aşağıdaki komutları verin:

```
RDEFINE MQCMDS QMgrName. ReqdAction.SERVICE UACC(NONE)
PERMIT QMgrName. ReqdAction.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(ALTER)
```

  Kullanıcının DISPLAY SERVICE komutunu kullanmasına izin vermek için aşağıdaki komutları verin:

```
RDEFINE MQCMDS QMgrName.DISPLAY.SERVICE UACC(NONE)
PERMIT QMgrName.DISPLAY.SERVICE CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

ReqdAction
Grubun gerçekleştirmesine izin verdiğiniz işlem:

- ALW AIX, Linux, and Windows sistemlerinde şu yetkilerin herhangi bir birleşimi: + chg, + clr, + crt, + dlt, + ctrl, + ctrlx, + dsp. Yetkilendirme + alladm, + chg + clr + dlt + dsp ile eşdeğerdir.
- IBM i IBM iüzerinde, şu yetkilerin herhangi bir birleşimi vardır: *ADMCHG, *ADMCLR, *ADMCRT, *ADMDLT, *ADMDSP, *CTRL, *CTRLX. *ALLADM yetkisi, tüm bu bireysel yetkilere eşdeğerdir.

Kuyruk yöneticisi kaynaklarının bir alt kümesinde tam denetim erişimi verilmesi

Belirli kullanıcılara bazı kuyruk yöneticisi kaynakları için tam denetim erişimi vermeniz gerekir, ancak tümü için değil. Gerçekleştirmeniz gereken işlemleri saptamak için bu çizelgeleri kullanın.

Çizelge 73. Kuyruk yöneticisi kaynaklarının bir alt kümesine tam denetim erişimi verilmesi

Kullanıcıların bu tipteki nesneleri denetmeleri gerekir	**Bu işlemi gerçekleştir**
Kuyruklar	"Bazı kuyruklara tam denetim erişimi verilmesi" sayfa 388 başlıklı konuda açıklandığı gibi, gerekli kuyruklara tam yönetici erişimi verin.
Konular	"Bazı konulara tam yönetici erişimi verilmesi" sayfa 389 başlıklı konuda açıklandığı gibi, gerekli konulara tam yönetici erişimi verin
Kanallar	"Bazı kanallara tam yönetici erişimi verilmesi" sayfa 390 başlıklı konuda açıklandığı gibi, gerekli kanallara tam yönetici erişimi verin
Kuyruk yöneticisi	"Bir kuyruk yöneticisine tam denetim erişimi verilmesi" sayfa 391 konusunda açıklandığı gibi kuyruk yöneticisine tam yönetici erişimi verin
Süreçler	"Bazı süreçlere tam denetim erişimi verilmesi" sayfa 391 başlıklı konuda açıklandığı gibi, gerekli süreçlere tam yönetici erişimi verin
Namelistler	"Bazı namelistlere tam yönetici erişimi verilmesi" sayfa 392 başlıklı konuda açıklandığı gibi, gerekli ad bilgilerine tam yönetici erişimi verin
Hizmetler	"Bazı hizmetlere tam yönetici erişimi verilmesi" sayfa 393 başlıklı konuda açıklandığı gibi, gerekli hizmetlere tam yönetici erişimi verin

Bazı kuyruklara tam denetim erişimi verilmesi

Bir kuyruk yöneticisindeki bazı kuyruklara, iş gereksinimi olan her bir kullanıcı grubuna tam denetim erişimi verin.

Bu görev hakkında

Bazı kuyruklara tam denetim erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW

  AIX, Linux, and Windows'ta:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +alladm
  ```

- IBM i

  IBM i'ta:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*ALLADM) MQMNAME(' QMgrName
  ')
  ```

- z/OS

  z/OS'ta:

  ```
  RDEFINE MQADMIN QMgrName.QUEUE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.QUEUE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

Bazı konulara tam yönetici erişimi verilmesi

Bir kuyruk yöneticisindeki bazı konulara, iş gereksinimi olan her kullanıcı grubuna tam yönetici erişimi verin.

Bu görev hakkında

Bazı konulara bazı işlemlerle ilgili olarak tam denetim erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW

  AIX, Linux, and Windows'ta:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +alladm
  ```

- IBM i

  IBM i'ta:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

  z/OS'ta:

  ```
  RDEFINE MQADMIN QMgrName.TOPIC. ObjectProfile UACC(NONE)
  PERMIT QMgrName.TOPIC. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

Bazı kanallara tam yönetici erişimi verilmesi

Bir kuyruk yöneticisindeki bazı kanallara, iş gereksinimi olan her bir kullanıcı grubuna tam yönetici erişimi verin.

Bu görev hakkında

Bazı kanallara tam yönetici erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW

  AIX, Linux, and Windows'ta:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t channel -g GroupName +alladm
  ```

- IBM i

  IBM i'ta:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*CHL) USER(GroupName) AUT(ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

  z/OS'ta:

  ```
  RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
  PERMIT QMgrName.CHANNEL. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

Bir kuyruk yöneticisine tam denetim erişimi verilmesi

Bir kuyruk yöneticisine, iş gereksinimi olan her bir kullanıcı grubuna tam denetim erişimi verin.

Bu görev hakkında

Kuyruk yöneticisine tam denetim erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW
  AIX, Linux, and Windows'ta:

```
setmqaut -m QMgrName -t qmgr -g GroupName +alladm
```

- IBM i
  IBM i'ta:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*ALLADM) MQMNAME('
QMgrName ')
```

- z/OS
  z/OS'ta:

```
RDEFINE MQADMIN QMgrName.QMGR UACC(NONE)
PERMIT QMgrName.QMGR CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

Bazı süreçlere tam denetim erişimi verilmesi

Bir kuyruk yöneticisindeki bazı süreçlere, iş gereksinimi olan her kullanıcı grubuna tam denetim erişimi verin.

Bu görev hakkında

Bazı işlemlere tam denetim erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW

  AIX, Linux, and Windows'ta:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +alladm
  ```

- IBM i

  IBM i'ta:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

  z/OS'ta:

  ```
  RDEFINE MQADMIN QMgrName.CHANNEL. ObjectProfile UACC(NONE)
  PERMIT QMgrName.PROCESS. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

Bazı namelistlere tam yönetici erişimi verilmesi

Bir kuyruk yöneticisindeki bazı ad bilgilerine, iş gereksinimi olan her kullanıcı grubuna tam yönetici erişimi verin.

Bu görev hakkında

Bazı ad sahiplerine tam denetim erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW

  AIX, Linux, and Windows'ta:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t namelist -g GroupName +alladm
  ```

- IBM i

  IBM i'ta:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS

z/OS'ta:

```
RDEFINE MQADMIN QMgrName.NAMELIST. ObjectProfile UACC(NONE)
PERMIT QMgrName.NAMELIST. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı.

z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

ObjectProfile
Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

Bazı hizmetlere tam yönetici erişimi verilmesi

Bir kuyruk yöneticisindeki bazı hizmetlere, iş gereksinimi olan her kullanıcı grubuna tam denetim erişimi verin.

Bu görev hakkında

Bazı hizmetlere tam denetim erişimi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW
  AIX, Linux, and Windows'ta:

  ```
  setmqaut -m QMgrName -n ObjectProfile -t service -g GroupName +alladm
  ```

- IBM i
  IBM i'ta:

  ```
  GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*SVC) USER(GroupName) AUT(*ALLADM) MQMNAME('
  QMgrName ')
  ```

- z/OS
  z/OS'ta:

  ```
  RDEFINE MQADMIN QMgrName.SERVICE. ObjectProfile UACC(NONE)
  PERMIT QMgrName.SERVICE. ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
  ```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

Kuyruk yöneticisindeki tüm kaynaklara salt okunur erişim verilmesi

Bir kuyruk yöneticisindeki tüm kaynaklara, iş gereksinimi olan her kullanıcıya ya da kullanıcı grubuna salt okunur erişim verin.

Bu görev hakkında

Rol Tabanlı Yetkiler Ekle sihirbazını ya da işletim sisteminiz için uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yetki ayrıntılarını değiştirdikten sonra, REFRESH SECURITY komutunu kullanarak bir güvenlik yenilemesi gerçekleştirin.

Yordam

- Sihirbazı kullanarak:
  a) IBM MQ Explorer Navigator bölmesinde, kuyruk yöneticisini farenin sağ düğmesiyle tıklatın ve **Nesne Yetkilileri** > **Rol Tabanlı Yetkilendirmeler Ekle** öğesini seçin.
  Role Dayalı Yetkiler Ekle sihirbazı açılır.
- ALW
  AIX, Linux, and Windows sistemleri için aşağıdaki komutları verin:

```
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
```

  SYSTEM.ADMIN.COMMAND.QUEUE ve SYSTEM.MQEXPLORER.REPLY.MODEL yalnızca IBM MQ Explorerkullanmak istiyorsanız gereklidir.
- IBM i
  IBM iiçin aşağıdaki komutları verin:

```
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
  z/OSiçin aşağıdaki komutları verin:

```
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)
```

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı.

z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

GroupName
Erişim verilecek grubun adı.

Bir kuyruk yöneticisindeki tüm kaynaklara tam denetim erişimi verilmesi

Bir kuyruk yöneticisindeki tüm kaynaklara, iş gereksinimi olan her kullanıcıya ya da kullanıcı grubuna tam denetim erişimi verin.

Bu görev hakkında

Rol Tabanlı Yetki Ekle sihirbazını ya da işletim sisteminiz için uygun komutları kullanabilirsiniz.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Notlar: ALW

1. Kuyruk yöneticisini IBM MQ Exploreryerine denetlemek için **runmqsc** kullanıyorsanız, SYSTEM.MQSC.REPLY.QUEUEve SYSTEM.MQEXPLORER.REPLY.MODEL kuyruğu.
2. Bir kullanıcıya bir kuyruk yöneticisindeki tüm kaynaklara erişim verirken, kullanıcının `qm.ini` dosyasına okuma erişimi olmadığı sürece çalıştıramayacağı bazı komutlar vardır. Bunun nedeni, `mqm` dışı kullanıcıların `qm.ini` dosyasını okuyabilmelerine ilişkin kısıtlamalardır.

   Kullanıcıya qm.ini dosyası için okuma erişimi vermediğiniz sürece kullanıcı aşağıdaki komutları veremez:

   - TLS kullanacak şekilde yapılandırılmış bir kanal tanımlama
   - `qm.ini` içinde tanımlanan otomatik yapılandırma ekleme değişkenlerini kullanarak bir kanal tanımlama

Yordam

- Sihirbazı kullanıyorsanız, IBM MQ Explorer Navigator bölmesinde kuyruk yöneticisini sağ tıklatın ve **Nesne Yetkileri** > **Rol Tabanlı Yetkilendirmeler Ekle**seçeneğini belirleyin.

  Role Dayalı Yetkiler Ekle sihirbazı açılır.

- Linux AIX

  AIX and Linux sistemleri için aşağıdaki komutları verin:

```
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
```

`@class` ile ilgili daha fazla bilgi için bkz. **setmqaut**

- Windows

  Windows sistemleri için, AIX and Linux sistemleriyle aynı komutları verin, ancak @classyerine @CLASS profil adını kullanın.

- IBM i

  IBM iiçin şu komutu verin:

```
GRTMQMAUT OBJ(*ALL) OBJTYPE(*ALL) USER(' GroupName ') AUT(*ALLADM) MQMNAME(' QMgrName ')
```

- z/OS

  z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQADMIN QMgrName.*.** UACC(NONE)
PERMIT QMgrName.*.** CLASS(MQADMIN) ID(GroupName) ACCESS(ALTER)
```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  GroupName
  Erişim verilecek grubun adı.

Kuyruk yöneticisine bağlanırlığın kaldırılması

Kullanıcı uygulamalarının kuyruk yöneticinizle bağlantı kurmasını istemiyorsanız, bu uygulamaya bağlanma yetkilerini kaldırın.

Bu görev hakkında

İşletim sisteminiz için uygun komutu kullanarak tüm kullanıcıların kuyruk yöneticisine bağlanma yetkisini iptal edin.

Çoklu platformlarişletim sistemlerinde DELETE AUTHREC komutunu da kullanabilirsiniz.

Not: IBM MQ Appliance üzerinde yalnızca **DELETE AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW

  AIX, Linux, and Windows sistemleri için aşağıdaki komutu verin:

```
setmqaut -m QMgrName -t qmgr -g GroupName -connect
```

- IBM i

  IBM iiçin şu komutu verin:

```
RVKMQMAUT OBJ ('QMgrName') OBJTYPE(*MQM) USER(*ALL) AUT(*CONNECT)
```

- z/OS

  z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
RDEFINE MQCONN QMgrName.CHIN UACC(NONE)
RDEFINE MQCONN QMgrName.CICS UACC(NONE)
RDEFINE MQCONN QMgrName.IMS UACC(NONE)
```

  Herhangi bir PERMIT komutu vermeyin.

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı.

  z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  GroupName
  Erişimin reddedileceği grubun adı.

Kullanıcı uygulamalarının kuyruk yöneticinizle bağlantı kurmasına izin verme

Kullanıcı uygulamasının kuyruk yöneticinize bağlanmasına izin vermek istiyorsunuz. Hangi işlemlerin yapılacağını belirlemek için bu konudaki tabloları kullanın.

Önce, istemci uygulamalarının kuyruk yöneticinize bağlanıp bağlanmayacağını belirleyin.

Kuyruk yöneticinizle bağlantı kuracak uygulamaların hiçbiri istemci uygulamaları değilse, “Kuyruk yöneticisine uzaktan erişimin devre dışı bırakılması” sayfa 404başlıklı konuda açıklandığı gibi uzak erişimi geçersiz kılın.

Kuyruk yöneticinizle bağlantı kuracak uygulamalardan biri ya da daha fazlası istemci uygulamalarıysa, “Kuyruk yöneticisine uzak bağlanırlığın güvenliğinin sağlanması” sayfa 397başlıklı konuda açıklandığı gibi uzak bağlanırlığı güvenceye alın.

Her iki durumda da, “Bağlantı güvenliğinin ayarlanması” sayfa 405 içinde açıklandığı gibi bağlantı güvenliğini ayarlayın

Kuyruk yöneticisine bağlanan her kullanıcı için kaynaklara erişimi denetlemek istiyorsanız aşağıdaki çizelgeye bakın. Birinci kolondaki deyim true ise, ikinci kolonda listelenen işlemi gerçekleştirin.

Deyim	Bu işlemi gerçekleştir
Kuyrukları kullanan uygulamalarınız var	Bkz. “Kuyruklara kullanıcı erişiminin denetlenmesi” sayfa 406
Konuları kullanan uygulamalarınız var	Bkz. “Konulara kullanıcı erişiminin denetlenmesi” sayfa 411.
Kuyruk yöneticisi nesnesinde soran uygulamalarınız var	Bkz. “Bir kuyruk yöneticisinde sorma yetkisi verilmesi” sayfa 412.
Süreç nesnelerini kullanan uygulamalarınız var	Bkz. “Süreçlere erişim yetkisi verilmesi” sayfa 413
Namelistleri kullanan uygulamalarınız var	Bkz. “Ad sahiplerine erişim yetkisi verilmesi” sayfa 414

Kuyruk yöneticisine uzak bağlanırlığın güvenliğinin sağlanması

TLS, güvenlik çıkışı, kanal kimlik doğrulama kayıtları ya da bu yöntemlerin bir birleşimini kullanarak kuyruk yöneticisine uzak bağlanırlığı güvenceye alabilirsiniz.

Bu görev hakkında

İstemci iş istasyonundaki bir istemci bağlantısı kanalını ve sunucudaki bir sunucu bağlantısı kanalını kullanarak bir istemciyi kuyruk yöneticisine bağlayabilirsiniz. Bu tür bağlantıları aşağıdaki yollardan biriyle güvenceye alın.

Yordam

1. Kanal kimlik doğrulama kayıtlarıyla TLS kullanılıyor:
   a) Tüm DN ' leri USERSRC (NOACCESS) ile eşlemek için bir SSLPEERMAP kanal kimlik doğrulama kaydı kullanarak, Ayırt Edici Adın (DN) bir kanal açmasını önleyin.
   b) Belirli DN 'lerin ya da DN' lerin USERSRC (KANAL) ile eşlemek için bir SSLPEERMAP kanal kimlik doğrulama kaydını kullanarak bir kanal açmasına izin verin.
2. Güvenlik çıkışıyla TLS kullanılıyor:
   a) Sunucu bağlantısı kanalındaki MCAUSER değerini, ayrıcalıkları olmayan bir kullanıcı kimliğine ayarlayın.
   b) SSLPeerNamePtr ve SSLPeerNameMQCD yapısındaki çıkışa geçirilen uzunluk alanlarında aldığı TLS DN değerine bağlı olarak bir MCAUSER değeri atamak için bir güvenlik çıkışı yazın.
3. Sabit kanal tanımlama değerleriyle TLS kullanılıyor:
   a) Sunucu bağlantısı kanalındaki SSLPEER ' i belirli bir değere ya da dar bir değer aralığına ayarlayın.
   b) Sunucu bağlantısı kanalındaki MCAUSER ' i kanalın çalışması gereken kullanıcı kimliğine ayarlayın.
4. TLS kullanmayan kanallarda kanal kimlik doğrulama kayıtlarını kullanma:
   a) ADDRESS (*) ve USERSRC (NOACCESS) ile bir adres eşleme kanalı kimlik doğrulama kaydı kullanarak herhangi bir IP adresinin kanalları açmasını önleyin.
   b) USERSRC (KANAL) ile bu adresler için adres eşleme kanalı kimlik doğrulama kayıtlarını kullanarak açık kanallara belirli IP adreslerine izin verin.
5. Güvenlik çıkışını kullanarak:
   a) Seçtiğiniz herhangi bir özelliğe (örneğin, kaynak IP adresine) dayalı olarak bağlantıları yetkilendirmek için bir güvenlik çıkışı yazın.
6. Ayrıca, kanal kimlik doğrulama kayıtlarının bir güvenlik çıkışı ile kullanılması ya da özel koşullarınız gerektiriyorsa, üç yöntemin de kullanılması da mümkündür.

Belirli IP adreslerini engelleme

Bir IP adresinden gelen bağlantıyı kabul eden belirli bir kanalı önleyebilir ya da tüm kuyruk yöneticisinin bir kanal kimlik doğrulama kaydı kullanarak IP adresinden erişime izin vermesini önleyebilirsiniz.

Başlamadan önce

Aşağıdaki komutu çalıştırarak kanal kimlik doğrulama kayıtlarını etkinleştirin:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Bu görev hakkında

Belirli kanalların gelen bir bağlantıyı kabul etmesine izin vermemek ve bağlantıların yalnızca doğru kanal adı kullanılırken kabul edildiğinden emin olmak için, IP adreslerini engellemek için bir kural tipi kullanılabilir. Bir IP adresinin tüm kuyruk yöneticisine erişmesine izin vermemek için, normalde bunu kalıcı olarak engellemek için bir güvenlik duvarı kullanırsınız. Ancak, örneğin güvenlik duvarının güncellenmesini beklerken, birkaç adresi geçici olarak engellemenizi sağlamak için başka bir kural tipi kullanılabilir.

Yordam

- IP adreslerinin belirli bir kanalı kullanmasını engellemek için, **SET CHLAUTH**MQSC komutunu ya da **Set Channel Authentication Record**PCF komutunu kullanarak bir kanal kimlik doğrulama kaydı ayarlayın.

```
SET CHLAUTH(generic-channel-name) TYPE(ADDRESSMAP) ADDRESS(generic-ip-address)
USERSRC(NOACCESS)
```

Komutun üç bölümü vardır:

SET CHLAUTH (*soysal-kanal-adı*)
Komutun bu bölümünü, tüm kuyruk yöneticisi, tek kanal ya da kanal aralığı için bir bağlantıyı engellemek isteyip istemediğinizi denetlemek için kullanırsınız. Buraya koyduğun şey hangi alanların kapsandığını belirler.

Örneğin:

- SET CHLAUTH('*') -bir kuyruk yöneticisindeki her kanalı, yani tüm kuyruk yöneticisini engeller
- SET CHLAUTH ('SYSTEM. *')-SYSTEM ile başlayan her kanalı engeller.
- CHLAUTH ('SYSTEM.DEF.SVRCONN')- SYSTEM.DEF.SVRCONN

CHLAUTH kuralının tipi
Komutun tipini belirlemek için komutun bu bölümünü kullanın ve tek bir adres mi, yoksa adres listesi mi girmek istediğinizi belirleyin.

Örneğin:

- TYPE(ADDRESSMAP) -Tek bir adres ya da genel arama adresi sağlamak istiyorsanız ADDRESSMAP komutunu kullanın. Örneğin, ADDRESS('192.168.*'), 192.168ile başlayan bir IP adresinden gelen bağlantıları engeller.

  IP adreslerini kalıplarla süzme hakkında daha fazla bilgi için bkz. Soysal IP adresleri.
- TYPE(BLOCKADDR) -Engellemek için bir adres listesi sağlamak istiyorsanız BLOCKADDR kullanın.

Ek parametreler
Bu değiştirgeler, komutun ikinci kısmında kullandığınız kural tipine bağlıdır:

- TYPE(ADDRESSMAP) için ADDRESS kullanılır
- TYPE(BLOCKADDR) için ADDRLIST kullanılır

İlgili başvurular
CHLAUTH AYARLA

Kuyruk yöneticisi çalışmıyorsa, belirli IP adreslerini geçici olarak engelleme
Kuyruk yöneticisi çalışmadığında belirli IP adreslerini ya da adres aralıklarını engellemek isteyebilirsiniz; bu nedenle MQSC komutları veremezsiniz. `blockaddr.ini` dosyasını değiştirerek IP adreslerini özel bir temelde geçici olarak engelleyebilirsiniz.

Bu görev hakkında

`blockaddr.ini` dosyası, kuyruk yöneticisi tarafından kullanılan BLOCKADDR tanımlarının bir kopyasını içerir. Dinleyici kuyruk yöneticisinden önce başlatıldıysa, bu dosya dinleyici tarafından okunur. Bu durumlarda, dinleyici `blockaddr.ini` dosyasına el ile eklediğiniz değerleri kullanır.

Ancak, kuyruk yöneticisi başlatıldığında, BLOCKADDR tanımları kümesini `blockaddr.ini` dosyasına yazdığını ve el ile düzenlemiş olabileceğiniz herhangi bir düzenlemenin üzerine yazdığını unutmayın. Benzer şekilde, **SET CHLAUTH** komutunu kullanarak her BLOCKADDR tanımı eklediğinizde ya da sildiğinizde `blockaddr.ini` dosyası güncellenir. Bu nedenle, BLOCKADDR tanımlarında kalıcı değişiklikler yapmak için kuyruk yöneticisi çalışırken **SET CHLAUTH** komutunu kullanabilirsiniz.

Yordam

1. `blockaddr.ini` dosyasını bir metin düzenleyicisinde açın.
   Dosya, kuyruk yöneticisinin veri dizininde bulunur.
2. IP adreslerini, anahtar sözcüğün `Addr`olduğu basit anahtar sözcük-değer çiftleri olarak ekleyin.

IP adreslerini kalıplarla süzme hakkında bilgi için bkz. Soysal IP adresleri.

Örneğin:

```
Addr = 192.0.2.0
Addr = 192.0.*
Addr = 192.0.2.1-8
```

İlgili görevler

“Belirli IP adreslerini engelleme” sayfa 398

Bir IP adresinden gelen bağlantıyı kabul eden belirli bir kanalı önleyebilir ya da tüm kuyruk yöneticisinin bir kanal kimlik doğrulama kaydı kullanarak IP adresinden erişime izin vermesini önleyebilirsiniz.

İlgili başvurular

CHLAUTH AYARLA

Belirli kullanıcı kimliklerini engelleme

Bildirildiyse, kanalın sona ermesine neden olan kullanıcı kimliklerini belirterek, belirli kullanıcıların kanal kullanmasını önleyebilirsiniz. Bunu bir kanal kimlik doğrulama kaydı ayarlayarak yapın.

Başlamadan önce

Kanal kimlik doğrulama kayıtlarının aşağıdaki gibi etkinleştirildiğinden emin olun:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Yordam

SET CHLAUTHMQSC komutunu ya da **Set Channel Authentication Record**PCF komutunu kullanarak bir kanal kimlik doğrulama kaydı ayarlayın. Örneğin, MQSC komutunu şu şekilde yayınlayabilirsiniz:

```
SET CHLAUTH(' generic-channel-name ') TYPE(BLOCKUSER) USERLIST(userID1, userID2)
```

soysal-kanal-adı , erişimi denetlemek istediğiniz bir kanalın adı ya da kanal adıyla eşleşen genel arama karakteri olarak yıldız (*) simgesini içeren bir örüntüdür.

TYPE(BLOCKUSER) ürününde sağlanan kullanıcı listesi, kuyruk yöneticisi kanallarına değil, yalnızca SVRCONN kanallarına uygulanır.

userID1 ve *userID2* , kanalı kullanması engellenecek bir kullanıcının kimliğidir. Ayrıcalıklı yönetimle görevli kullanıcılara gönderme yapmak için *MQADMIN özel değerini de belirtebilirsiniz. Ayrıcalıklı kullanıcılar hakkında daha fazla bilgi için bkz. “Ayrıcalıklı kullanıcılar” sayfa 339. *MQADMINhakkında daha fazla bilgi için bkz. SET CHLAUTH.

İlgili başvurular

CHLAUTH AYARLA

Uzak kuyruk yöneticisinin MCAUSER kullanıcı kimliğiyle eşlenmesi

Kanalın bağlandığı kuyruk yöneticisine göre bir kanalın MCAUSER özniteliğini ayarlamak için kanal kimlik doğrulama kaydını kullanabilirsiniz.

Başlamadan önce

Kanal kimlik doğrulama kayıtlarının aşağıdaki gibi etkinleştirildiğinden emin olun:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Bu görev hakkında

İsteğe bağlı olarak, kuralın geçerli olduğu IP adreslerini kısıtlayabilirsiniz.

Bu tekniğin sunucu bağlantısı kanalları için geçerli olmadığını unutmayın. Aşağıdaki komutlarda bir sunucu bağlantısı kanalının adını belirtirseniz, bu bir etki yaratmaz.

Yordam

- **SET CHLAUTH**MQSC komutunu ya da **Set Channel Authentication Record**PCF komutunu kullanarak bir kanal kimlik doğrulama kaydı ayarlayın. Örneğin, MQSC komutunu şu şekilde yayınlayabilirsiniz:

```
SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
) USERSRC(MAP) MCAUSER(user)
```

  soysal-kanal-adı , erişimi denetlemek istediğiniz bir kanalın adı ya da kanal adıyla eşleşen genel arama karakteri olarak yıldız (*) simgesini içeren bir örüntüdür.
  generic-partner-qmgr-name , kuyruk yöneticisinin adıdır ya da kuyruk yöneticisi adıyla eşleşen genel arama karakteri olarak yıldız (*) simgesini içeren bir kalıptır.
  kullanıcı , belirtilen kuyruk yöneticisinden gelen tüm bağlantılar için kullanılacak kullanıcı kimliğidir.
- Bu komutu belirli IP adresleriyle sınırlamak için **ADDRESS** parametresini aşağıdaki gibi ekleyin:

```
SET CHLAUTH(' generic-channel-name ') TYPE (QMGRMAP) QMNAME(generic-partner-qmgr-name
) USERSRC(MAP) MCAUSER(user) ADDRESS(
generic-ip-address)
```

  soysal-kanal-adı , erişimi denetlemek istediğiniz bir kanalın adı ya da kanal adıyla eşleşen genel arama karakteri olarak yıldız (*) simgesini içeren bir örüntüdür.
  generic-ip-address , tek bir adres ya da genel arama karakteri olarak yıldız (*) simgesini içeren bir kalıp ya da adresle eşleşen bir aralığı gösteren kısa çizgi (-). Soysal IP adresleriyle ilgili ek bilgi için Soysal IP adresleribaşlıklı konuya bakın.

İlgili başvurular
CHLAUTH AYARLA

Bir istemci kullanıcı kimliğinin MCAUSER kullanıcı kimliğiyle eşlenmesi
Bir istemciden alınan kullanıcı kimliğine göre, bir sunucu bağlantısı kanalının MCAUSER özniteliğini değiştirmek için kanal kimlik doğrulama kaydını kullanabilirsiniz.

Başlamadan önce
Kanal kimlik doğrulama kayıtlarının aşağıdaki gibi etkinleştirildiğinden emin olun:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Bu görev hakkında
Bu tekniğin yalnızca sunucu bağlantısı kanalları için geçerli olduğunu unutmayın. Diğer kanal tipleri üzerinde etkisi yoktur.

Yordam

SET CHLAUTHMQSC komutunu ya da **Set Channel Authentication Record** PCF komutunu kullanarak bir kanal kimlik doğrulama kaydı ayarlayın. Örneğin, MQSC komutunu şu şekilde yayınlayabilirsiniz:

```
SET CHLAUTH(' generic-channel-name ') TYPE (USERMAP) CLNTUSER(client-user-name) USERSRC(MAP)
```

```
MCAUSER(
user)
```

soysal-kanal-adı , erişimi denetlemek istediğiniz bir kanalın adı ya da kanal adıyla eşleşen genel arama karakteri olarak yıldız (*) simgesini içeren bir örüntüdür.
istemci-kullanıcı-adı , istemci bağlantısıyla ilişkilendirilmiş kullanıcı kimliğidir; değer, istemci uygulaması tarafından, erken benimseme kullanılarak bağlantı kimlik doğrulamasıyla ya da kanal çıkışı yoluyla ayarlanarak değiştirilebilir.
kullanıcı , istemci kullanıcı adı yerine kullanılacak kullanıcı kimliğidir.

İlgili başvurular
CHLAUTH AYARLA
Kanal kısmı öznitelikleri (ChlauthEarly)

Bir SSL ya da TLS Ayırt Edici Adının MCAUSER Kullanıcı Kimliğiyle Eşlenmesi
Alınan ayırt edici ada (DN) göre bir kanalın MCAUSER özniteliğini ayarlamak için bir kanal kimlik doğrulama kaydı kullanabilirsiniz.

Başlamadan önce
Kanal kimlik doğrulama kayıtlarının aşağıdaki gibi etkinleştirildiğinden emin olun:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Yordam

SET CHLAUTHMQSC komutunu ya da **Set Channel Authentication Record**PCF komutunu kullanarak bir kanal kimlik doğrulama kaydı ayarlayın. Örneğin, MQSC komutunu şu şekilde yayınlayabilirsiniz:

```
SET CHLAUTH('generic-channel-name') TYPE (SSLPEERMAP)
SSLPEER(generic-ssl-peer-name) SSLCERTI(generic-issuer-name)
USERSRC(MAP) MCAUSER(user)
```

soysal-kanal-adı , erişimi denetlemek istediğiniz bir kanalın adı ya da kanal adıyla eşleşen genel arama karakteri olarak yıldız (*) simgesini içeren bir örüntüdür.
generic-ssl-peer-name , SSLPEER değerleri için standart IBM MQ kurallarını izleyen bir dizedir. Bkz. IBM MQ SSLPEER değerleri için kurallar.
kullanıcı , belirtilen ayırt edici adı (DN) kullanan tüm bağlantılar için kullanılacak kullanıcı kimliğidir.
soysal-sertifika veren-adı , eşleşecek sertifikanın Sertifika Veren DN 'sine başvurur. Bu parametre isteğe bağlıdır, ancak birden çok sertifika yetkilisi kullanımdaysa, yanlış sertifikanın istenmeyen bir şekilde eşleştirilmesini önlemek için bu parametreyi kullanmanız gerekir.

İlgili başvurular
CHLAUTH AYARLA

Uzak kuyruk yöneticisinden erişimin engellenmesi
Uzak kuyruk yöneticisinin kanalları başlatmasını önlemek için bir kanal kimlik doğrulama kaydı kullanabilirsiniz.

Başlamadan önce
Kanal kimlik doğrulama kayıtlarının aşağıdaki gibi etkinleştirildiğinden emin olun:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Bu görev hakkında

Bu tekniğin sunucu bağlantısı kanalları için geçerli olmadığını unutmayın. Aşağıdaki komutta bir sunucu bağlantısı kanalının adını belirtirseniz, bu bir etki yaratmaz.

Yordam

SET CHLAUTHMQSC komutunu ya da **Set Channel Authentication Record**PCF komutunu kullanarak bir kanal kimlik doğrulama kaydı ayarlayın. Örneğin, MQSC komutunu şu şekilde yayınlayabilirsiniz:

```
SET CHLAUTH(' generic-channel-name ') TYPE(QMGRMAP) QMNAME(' generic-partner-qmgr-name ')
USERSRC(NOACCESS)
```

soysal-kanal-adı , erişimi denetlemek istediğiniz bir kanalın adı ya da kanal adıyla eşleşen genel arama karakteri olarak yıldız (*) simgesini içeren bir örüntüdür.

generic-partner-qmgr-name , kuyruk yöneticisinin adıdır ya da kuyruk yöneticisi adıyla eşleşen genel arama karakteri olarak yıldız (*) simgesini içeren bir kalıptır.

İlgili başvurular

CHLAUTH AYARLA

İstemci kullanıcı kimliği için erişimin engellenmesi

Bir istemci kullanıcı kimliğinin kanal bağlantısı kurmasını önlemek için kanal kimlik doğrulama kaydı kullanabilirsiniz.

Başlamadan önce

Kanal kimlik doğrulama kayıtlarının aşağıdaki gibi etkinleştirildiğinden emin olun:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Bu görev hakkında

Bu tekniğin yalnızca sunucu bağlantısı kanalları için geçerli olduğunu unutmayın. Diğer kanal tipleri üzerinde etkisi yoktur.

Yordam

SET CHLAUTHMQSC komutunu ya da **Set Channel Authentication Record**PCF komutunu kullanarak bir kanal kimlik doğrulama kaydı ayarlayın. Örneğin, MQSC komutunu şu şekilde yayınlayabilirsiniz:

```
SET CHLAUTH(' generic-channel-name ') TYPE(USERMAP) CLNTUSER(' client-user-name ')
USERSRC(NOACCESS)
```

soysal-kanal-adı , erişimi denetlemek istediğiniz bir kanalın adı ya da kanal adıyla eşleşen genel arama karakteri olarak yıldız (*) simgesini içeren bir örüntüdür.

istemci-kullanıcı-adı , istemci bağlantısıyla ilişkilendirilmiş kullanıcı kimliğidir; değer, istemci uygulaması tarafından, erken benimseme kullanılarak bağlantı kimlik doğrulamasıyla ya da kanal çıkışı yoluyla ayarlanarak değiştirilebilir.

İlgili başvurular

CHLAUTH AYARLA

SSL ya da TLS Ayırt Edici Adı için erişimi engelleme

Bir TLS Ayırt Edici Adının (DN) kanalları başlatmasını önlemek için bir kanal kimlik doğrulama kaydı kullanabilirsiniz.

Başlamadan önce

Kanal kimlik doğrulama kayıtlarının aşağıdaki gibi etkinleştirildiğinden emin olun:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Yordam

SET CHLAUTHMQSC komutunu ya da **Set Channel Authentication Record**PCF komutunu kullanarak bir kanal kimlik doğrulama kaydı ayarlayın. Örneğin, MQSC komutunu şu şekilde yayınlayabilirsiniz:

```
SET CHLAUTH('generic-channel-name') TYPE(SSLPEERMAP)
SSLPEER('generic-ssl-peer-name') SSLCERTI(generic-issuer-name)
USERSRC(NOACCESS)
```

soysal-kanal-adı , erişimi denetlemek istediğiniz bir kanalın adı ya da kanal adıyla eşleşen genel arama karakteri olarak yıldız (*) simgesini içeren bir örüntüdür.

generic-ssl-peer-name , SSLPEER değerleri için standart IBM MQ kurallarını izleyen bir dizedir. Bkz. IBM MQ SSLPEER değerleri için kurallar.

soysal-sertifika veren-adı , eşleşecek sertifikanın Sertifika Veren DN 'sine başvurur. Bu parametre isteğe bağlıdır, ancak birden çok sertifika yetkilisi kullanımdaysa, yanlış sertifikanın istenmeyen bir şekilde eşleştirilmesini önlemek için bu parametreyi kullanmanız gerekir.

İlgili başvurular

CHLAUTH AYARLA

Bir IP adresinin MCAUSER kullanıcı kimliğiyle eşlenmesi

Bir kanalın MCAUSER özniteliğini, bağlantının alındığı IP adresine göre ayarlamak için bir kanal kimlik doğrulama kaydı kullanabilirsiniz.

Başlamadan önce

Kanal kimlik doğrulama kayıtlarının aşağıdaki gibi etkinleştirildiğinden emin olun:

```
ALTER QMGR CHLAUTH(ENABLED)
```

Yordam

SET CHLAUTHMQSC komutunu ya da **Set Channel Authentication Record**PCF komutunu kullanarak bir kanal kimlik doğrulama kaydı ayarlayın. Örneğin, MQSC komutunu şu şekilde yayınlayabilirsiniz:

```
SET CHLAUTH(' generic-channel-name ') TYPE(ADDRESSMAP) ADDRESS(' generic-ip-address ')
USERSRC(MAP) MCAUSER(user)
```

soysal-kanal-adı , erişimi denetlemek istediğiniz bir kanalın adı ya da kanal adıyla eşleşen genel arama karakteri olarak yıldız (*) simgesini içeren bir örüntüdür.

kullanıcı , belirtilen ayırt edici adı (DN) kullanan tüm bağlantılar için kullanılacak kullanıcı kimliğidir.

soysal-ip-adresi , bağlantının yapıldığı adres ya da genel arama karakteri olarak yıldız işareti (*) ya da adresle eşleşen bir aralığı belirtmek için kısa çizgi (-) içeren bir kalıptır.

İlgili başvurular

CHLAUTH AYARLA

Kuyruk yöneticisine uzaktan erişimin devre dışı bırakılması

İstemci uygulamalarının kuyruk yöneticinizle bağlantı kurmasını istemiyorsanız, uzak erişimi geçersiz kılın.

Bu görev hakkında

Aşağıdaki yollardan biriyle istemci uygulamalarının kuyruk yöneticisine bağlanmasını engelleyin:

Yordam

- **DELETE CHANNEL**MQSC komutunu kullanarak tüm sunucu bağlantısı kanallarını silin.
- **ALTER CHANNEL**MQSC komutunu kullanarak, kanalın ileti kanalı aracısı kullanıcı kimliğini (MCAUSER) erişim hakları olmayan bir kullanıcı kimliğine ayarlayın.

Bağlantı güvenliğinin ayarlanması

Kuyruk yöneticisine, iş gereksinimi olan her kullanıcıya ya da kullanıcı grubuna bağlanma yetkisi verin.

Bu görev hakkında

Bağlantı güvenliğini ayarlamak için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW

  AIX, Linux, and Windows'ta:

  ```
  setmqaut -m QMgrName -t qmgr -g GroupName +connect
  ```

- IBM i

  IBM i'ta:

  ```
  GRTMQMAUT OBJ('QMgrName') OBJTYPE(*MQM) USER('GroupName') AUT(*CONNECT)
  ```

- z/OS

  z/OS'ta:

  ```
  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)
  ```

  Bu komutlar, toplu iş, CICS, IMS ve kanal başlatıcısı (CHIN) için bağlanma yetkisi verir. Belirli bir bağlantı tipini kullanmıyorsanız, ilgili komutları atlayın.

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

İlgili kavramlar

“Kanal başlatıcısına ilişkin bağlantı güvenliği tanıtımları” sayfa 196

Kanal başlatıcısından gelen bağlantıları denetlemeye ilişkin profiller, kuyruk yöneticisi ya da kuyruk paylaşım grubu adından ve ardından *CHIN*sözcüğünden oluşur. Kanal başlatıcı tarafından başlatılan görev adresi alanı tarafından kullanılan kullanıcı kimliğine bağlantı tanıtımı için OKUMA erişimi verin.

Kuyruklara kullanıcı erişiminin denetlenmesi

Kuyruklara uygulama erişimini denetlemek istiyorsunuz. Hangi işlemlerin yapılacağını belirlemek için bu konuyu kullanın.

Birinci kolondaki her true deyimi için, ikinci kolonda belirtilen işlemi gerçekleştirin.

Deyim	**Eylem**
Uygulama iletileri kuyruktan alır	Bkz. “Kuyruklardan ileti almak için yetki verilmesi” sayfa 406
Uygulama bağlamı ayarlar	Bkz. “Bağlamı ayarlamak için yetki verilmesi” sayfa 407
Uygulama bağlamı geçirir	Bkz. “Bağlamı geçirmek için yetki verilmesi” sayfa 408
Uygulama iletileri kümelenmiş bir kuyruğa koyar	Bkz. “Uzak küme kuyruklarına ileti konmasına yetki verilmesi” sayfa 491
Uygulama iletileri yerel kuyruğa koyar	Bkz. “İletileri yerel kuyruğa koymak için yetki verilmesi” sayfa 409
Uygulama iletileri model kuyruğuna koyar	Bkz. “İletileri model kuyruğuna koymak için yetki verilmesi” sayfa 409
Uygulama iletileri uzak kuyruğa koyar	Bkz. “Uzak küme kuyruğuna ileti koymak için yetki verilmesi” sayfa 410

Kuyruklardan ileti almak için yetki verilmesi

İş gereksinimi olan her kullanıcı grubuna, kuyruktan ya da kuyruk kümesinden ileti alma yetkisi verin.

Bu görev hakkında

Bazı kuyruklardan ileti alma yetkisi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- AIX, Linux, and Windows sistemleri için aşağıdaki komutu verin:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +get
```

- IBM iiçin şu komutu verin:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*GET) MQMNAME(' QMgrName ')
```

- z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

ObjectProfile
Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

Bağlamı ayarlamak için yetki verilmesi
İş gereksinimi olan her bir kullanıcı grubuna, konmakta olan bir ileti için bağlam ayarlama yetkisi verin.

Bu görev hakkında

Bazı kuyruklarda bağlam ayarlama yetkisi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- AIX, Linux, and Windows sistemleri için aşağıdaki komutlardan birini yayınlayın:
  - Yalnızca kimlik bağlamını ayarlamak için:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setid
```

  - Tüm bağlamı ayarlamak için:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +setall
```

  Not: `setid` ya da `setall` yetkisini kullanmak için, hem uygun kuyruk nesnesinde hem de kuyruk yöneticisi nesnesinde yetkilerin verilmesi gerekir.
- IBM iiçin aşağıdaki komutlardan birini verin:
  - Yalnızca kimlik bağlamını ayarlamak için:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETID) MQMNAME('
QMgrName ')
```

  - Tüm bağlamı ayarlamak için:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*SETALL) MQMNAME('
QMgrName ')
```

- z/OSiçin aşağıdaki komut kümelerinden birini yayınlayın:
  - Yalnızca kimlik bağlamını ayarlamak için:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  - Tüm bağlamı ayarlamak için:

```
RDEFINE MQQUEUE QMgrName. ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(CONTROL)
```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

Bağlamı geçirmek için yetki verilmesi
Alınan bir iletideki bağlamı, iş gereksinimi olan her bir kullanıcı grubuna, konmakta olan bir iletiye iletme yetkisi verin.

Bu görev hakkında

Bazı kuyruklara bağlam iletme yetkisi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- ALW

  AIX, Linux, and Windows sistemleri için aşağıdaki komutlardan birini yayınlayın:

  - Yalnızca kimlik bağlamını geçirmek için:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passid
    ```

  - Tüm bağlamı geçirmek için:

    ```
    setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +passall
    ```

- IBM i

  IBM iiçin aşağıdaki komutlardan birini verin:

  - Yalnızca kimlik bağlamını geçirmek için:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSID) MQMNAME('
    QMgrName ')
    ```

  - Tüm bağlamı geçirmek için:

    ```
    GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PASSALL) MQMNAME('
    QMgrName ')
    ```

- z/OS

  z/OSiçin, kimlik bağlamını ya da tüm bağlamı iletmek üzere aşağıdaki komutları verin:

  ```
  RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
  PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
  ```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

İletileri yerel kuyruğa koymak için yetki verilmesi
İletileri yerel bir kuyruğa ya da kuyruk kümesine, iş gereksinimi olan her kullanıcı grubuna koyma yetkisi verin.

Bu görev hakkında

İletileri bazı yerel kuyruklara koyma yetkisi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- AIX, Linux, and Windows sistemleri için aşağıdaki komutu verin:

```
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- IBM iiçin şu komutu verin:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

- z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

  Değişken adları aşağıdaki anlamlara sahiptir:

  QMgrName
  Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

  ObjectProfile
  Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

  GroupName
  Erişim verilecek grubun adı.

İletileri model kuyruğuna koymak için yetki verilmesi
İletileri bir model kuyruğuna ya da model kuyrukları kümesine, iş gereksinimi olan her kullanıcı grubuna koyma yetkisi verin.

Bu görev hakkında

Dinamik kuyruklar yaratmak için model kuyrukları kullanılır. Bu nedenle, hem model hem de dinamik kuyruklar için yetki vermeniz gerekir. Bu yetkilere yetki vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- AIX, Linux, and Windows sistemleri için aşağıdaki komutları verin:

```
setmqaut -m QMgrName -n ModelQueueName -t queue -g GroupName +put
setmqaut -m QMgrName -n ObjectProfile -t queue -g GroupName +put
```

- IBM iiçin aşağıdaki komutları verin:

```
GRTMQMAUT OBJ(' ModelQueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

- z/OSiçin aşağıdaki komutları verin:

```
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)
```

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

ModelQueueAdı
Dinamik kuyrukların dayalı olduğu model kuyruğunun adı.

ObjectProfile
Yetkileri değiştirilecek dinamik kuyruğun ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

Uzak küme kuyruğuna ileti koymak için yetki verilmesi
İletileri uzak bir küme kuyruğuna ya da kuyruk kümesine, iş gereksinimi olan her kullanıcı grubuna yerleştirme yetkisi verin.

Bu görev hakkında

Bir iletiyi uzak küme kuyruğuna koymak için, iletiyi uzak kuyruğun yerel tanımına ya da tam olarak nitelenmiş bir uzak kuyruğa yerleştirebilirsiniz. Uzak bir kuyruk için yerel bir tanımlama kullanıyorsanız, yerel nesne için yetki vermeniz gerekir: bkz. “İletileri yerel kuyruğa koymak için yetki verilmesi” sayfa 409. Tam olarak nitelenmiş bir uzak kuyruk kullanıyorsanız, uzak kuyruğa koyma yetkinizin olması gerekir. İşletim sisteminiz için uygun komutları kullanarak bu yetkiyi verin.

Varsayılan davranış, SYSTEM.CLUSTER.TRANSMIT.QUEUEüzerinde erişim denetimi gerçekleştirilmesidir. Birden çok iletim kuyruğu kullanıyor olsanız da, bu davranışın geçerli olduğunu unutmayın.

Bu konuda açıklanan belirli bir davranış, yalnızca qm.ini dosyasındaki **ClusterQueueAccessControl** özniteliğini, Güvenlik kısmı konusunda açıklandığı gibi *RQMName*olacak şekilde yapılandırdığınızda ve kuyruk yöneticisini yeniden başlattığınızda geçerlidir.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- AIX, Linux, and Windows sistemleri için aşağıdaki komutu verin:

```
setmqaut -m QMgrName -t rqmname -n
ObjectProfile -g GroupName +put
```

rqmname nesnesini yalnızca uzak küme kuyrukları için kullanabileceğinizi unutmayın.

- IBM iiçin şu komutu verin:

```
GRTMQMAUT OBJTYPE(*RMTMQMNAME) OBJ('
ObjectProfile') USER(GroupName) AUT(*PUT) MQMNAME('
QMgrName')
```

RMTMQMNAME nesnesini yalnızca uzak küme kuyrukları için kullanabildiğinizi unutmayın.

- z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQQUEUE QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQQUEUE) ID(GroupName) ACCESS(UPDATE)
```

Uzak kuyruk yöneticisinin (ya da kuyruk paylaşım grubunun) adını yalnızca uzak küme kuyrukları için kullanabildiğinizi unutmayın.

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

ObjectProfile
Yetkileri değiştirilecek uzak kuyruk yöneticisinin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

Konulara kullanıcı erişiminin denetlenmesi

Konulara uygulama erişimini denetlemeniz gerekir. Hangi işlemlerin yapılacağını belirlemek için bu konuyu kullanın.

Birinci kolondaki her true deyimi için, ikinci kolonda belirtilen işlemi gerçekleştirin.

Çizelge 74. Konulara kullanıcı erişiminin denetlenmesi

Deyim	**Eylem**
Uygulama iletileri bir konuya yayınlar	Bkz. "Bir konuya ileti yayınlama yetkisi verilmesi" sayfa 411
Uygulama bir konuya abone olur	Bkz. "Konulara abone olmak için yetki verilmesi" sayfa 412

Bir konuya ileti yayınlama yetkisi verilmesi

Bir konuya ya da konu kümesine, iş gereksinimi olan her kullanıcı grubuna ileti yayınlama yetkisi verin.

Bu görev hakkında

Bazı konulara ileti yayınlama yetkisi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- AIX, Linux, and Windows sistemleri için aşağıdaki komutu verin:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +pub
```

- IBM iiçin şu komutu verin:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*PUB) MQMNAME('
QMgrName ')
```

- z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQTOPIC QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

ObjectProfile
Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

Konulara abone olmak için yetki verilmesi
Bir konuya ya da konu kümesine abone olma yetkisini, iş gereksinimi olan her kullanıcı grubuna verin.

Bu görev hakkında

Bazı konulara abone olma yetkisi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- AIX, Linux, and Windows sistemleri için aşağıdaki komutu verin:

```
setmqaut -m QMgrName -n ObjectProfile -t topic -g GroupName +sub
```

- IBM iiçin şu komutu verin:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*TOPIC) USER(GroupName) AUT(*SUB) MQMNAME('
QMgrName ')
```

- z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQTOPIC QMgrName.SUBSCRIBE.ObjectProfile UACC(NONE)
PERMIT QMgrName.SUBSCRIBE.ObjectProfile CLASS(MQTOPIC) ID(GroupName) ACCESS(UPDATE)
```

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

ObjectProfile
Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

Bir kuyruk yöneticisinde sorma yetkisi verilmesi

İş gereksinimi olan her kullanıcı grubuna bir kuyruk yöneticisinde sorma yetkisi verin.

Bu görev hakkında

Bir kuyruk yöneticisine sorma yetkisi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- AIX, Linux, and Windows sistemleri için aşağıdaki komutu verin:

```
setmqaut -m QMgrName -n ObjectProfile -t qmgr -g GroupName +inq
```

- IBM iiçin şu komutu verin:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*MQM) USER(GroupName) AUT(*INQ) MQMNAME(' QMgrName
')
```

- z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQCMDS QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Bu komutlar, belirtilen kuyruk yöneticisine erişim verir. Kullanıcının MQINQ komutunu kullanmasına izin vermek için aşağıdaki komutları verin:

```
RDEFINE MQCMDS QMgrName.MQINQ.QMGR UACC(NONE)
PERMIT QMgrName.MQINQ.QMGR CLASS(MQCMDS) ID(GroupName) ACCESS(READ)
```

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

ObjectProfile
Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

Süreçlere erişim yetkisi verilmesi

Bir sürece ya da süreç kümesine, bir iş gereksinimi olan her bir kullanıcı grubuna erişim yetkisi verin.

Bu görev hakkında

Bazı işlemlere erişim yetkisi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- AIX, Linux, and Windows sistemleri için aşağıdaki komutu verin:

```
setmqaut -m QMgrName -n ObjectProfile -t process -g GroupName +all
```

- IBM iiçin şu komutu verin:

```
GRTMQMAUT OBJ(' ObjectProfile ') OBJTYPE(*PRC) USER(GroupName) AUT(*ALL) MQMNAME(' QMgrName
')
```

- z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQPROC QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile CLASS(MQPROC) ID(GroupName) ACCESS(READ)
```

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

ObjectProfile
Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

Ad sahiplerine erişim yetkisi verilmesi

İş gereksinimi olan her kullanıcı grubuna bir ad listesi ya da ad listesi kümesine erişim yetkisi verin.

Bu görev hakkında

Bazı ad bilgilerine erişim yetkisi vermek için işletim sisteminize ilişkin uygun komutları kullanın.

Multiplatforms altyapılarında, SET AUTHREC komutunu da kullanabilirsiniz.

Not: MQ Appliance IBM MQ Appliance üzerinde yalnızca **SET AUTHREC** komutunu kullanabilirsiniz.

Yordam

- AIX, Linux, and Windows sistemleri için aşağıdaki komutu verin:

```
setmqaut -m QMgrName -n
ObjectProfile -t namelist -g GroupName
+all
```

- IBM iiçin şu komutu verin:

```
GRTMQMAUT OBJ('ObjectProfile
') OBJTYPE(*NMLIST) USER(GroupName) AUT(*ALL) MQMNAME('
QMgrName')
```

- z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQNLIST
QMgrName.ObjectProfile UACC(NONE)
PERMIT QMgrName.ObjectProfile
CLASS(MQNLIST) ID(GroupName) ACCESS(READ)
```

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

ObjectProfile
Yetkileri değiştirilecek nesnenin ya da soysal tanıtımın adı.

GroupName
Erişim verilecek grubun adı.

ALW AIX, Linux, and Windows üzerinde IBM MQ yönetimi yetkisi

IBM MQ yöneticileri tüm IBM MQ komutlarını kullanabilir ve diğer kullanıcılara yetki verebilir. Denetimciler uzak kuyruk yöneticilerine komut verdiklerinde, uzak kuyruk yöneticisi üzerinde gerekli yetkiye sahip olmaları gerekir. Windows sistemleri için daha fazla dikkat edilmesi gereken noktalar vardır.

IBM MQ denetimcileri tüm IBM MQ komutlarını kullanma yetkisine sahiptir (diğer kullanıcılara IBM MQ yetkilerini verme komutları da içinde olmak üzere).

IBM MQ yöneticisi olmak için **mqm** grubu adı verilen özel bir grubun üyesi olmanız gerekir.

Windows Alternatif olarak, yalnızca Windows sistemlerinde yerel hesaplar, Windows sistemlerinde Administrators (Yöneticiler) grubunun üyesiyse, IBM MQ ' yi yönetebilir.

Uyarı: Azure AD kullanıcıınızı bir yönetici komutunu kullanarak mqm grubuna ekleyebilirsiniz. Örneğin, `net localgroup mqm AzureAD\<your userID> /add`komutunu kullanın. Daha sonra IBM MQ yönetim komutlarını çalıştırın ya da IBM MQ Explorerkomutunu kullanın.

mqm grubu, IBM MQ kurulduğunda otomatik olarak oluşturulur. Yönetim gerçekleştirmelerine izin vermek için gruba daha fazla kullanıcı ekleyebilirsiniz. Bu grubun tüm üyelerinin tüm kaynaklara erişimi vardır. Bu erişim yalnızca **mqm** grubundan bir kullanıcı kaldırılıp **REFRESH SECURITY** komutu verilerek iptal edilebilir.

Yöneticiler IBM MQ' yi yönetmek için denetim komutlarını kullanabilir. Bu denetim komutlarından biri, diğer kullanıcılara IBM MQ kaynaklarına erişmelerini ya da bunları denetlemelerini sağlamak üzere yetki vermek için kullanılan **setmqaut**komutudur. Yetki kayıtlarını yönetmeye ilişkin PCF komutları, kuyruk yöneticisinde dsp ve chg yetkileri verilen denetimci olmayan kullanıcılar tarafından kullanılabilir. Yetkilerin PCF komutlarını kullanarak yönetilmesine ilişkin ek bilgi için Programlanabilir Komut Biçimleribaşlıklı konuya bakın.

Uzak kuyruk yöneticisi tarafından işlenecek MQSC komutları için denetimcilerin gerekli yetkileri olmalıdır. IBM MQ Explorer , denetim görevlerini gerçekleştirmek için PCF komutları verir. Denetimciler, yerel sistemde bir kuyruk yöneticisini denetlemek için IBM MQ Explorer ' yi kullanmak üzere ek yetkilere gereksinim duymaz. IBM MQ Explorer başka bir sistemdeki bir kuyruk yöneticisini denetlemek için kullanıldığında, denetimcilerin PCF komutlarının uzak kuyruk yöneticisi tarafından işlenmesi için gerekli yetkileri olmalıdır.

Uyarı: IBM MQ 8.0' den IBM MQ Script (MQSC) komutlarını veren **runmqsc**denetim komutunu kullanmak için denetimci olmanız gerekmez.

runmqsc uzak bir kuyruk yöneticisine MQSC komutları göndermek için dolaylı kipte kullanıldığında, her MQSC komutu Escape PCF komutunda kapsüllenir.

PCF ve MQSC komutları işlenirken yetki denetimleriyle ilgili ek bilgi için aşağıdaki konulara bakın:

- Kuyruk yöneticileri, kuyruklar, işlemler, ad ve kimlik doğrulama bilgileri nesnelerinde çalışan PCF komutları için IBM MQ nesnelerle çalışma yetkisibaşlıklı konuya bakın. Escape PCF komutlarında kapsüllenmiş eşdeğer MQSC komutları için bu bölüme bakın.
- Kanallar, kanal başlatıcılar, dinleyiciler ve kümeler üzerinde çalışan PCF komutları için bkz. Kanal güvenliği.
- Yetki kayıtlarında çalışan PCF komutları için PCF komutları için yetki denetimi başlıklı konuya bakın.
- z/OS IBM MQ for z/OSüzerinde komut sunucusu tarafından işlenen MQSC komutları için bkz. z/OS üzerinde komut güvenliği ve komut kaynağı güvenliği .

Ayrıca, Windows sistemlerinde SYSTEM hesabının IBM MQ kaynaklarına tam erişimi vardır.

AIX and Linux platformlarında, yalnızca ürün tarafından kullanılmak üzere özel bir **mqm** kullanıcı kimliği de oluşturulur. Ayrıcalıklı olmayan kullanıcılar tarafından hiçbir zaman kullanılabilir olmamalıdır. Tüm IBM MQ nesnelerinin iyesi **mqm**kullanıcı kimliği olur.

Windows sistemlerinde, Administrators (Yöneticiler) grubunun üyeleri, SYSTEM hesabı gibi herhangi bir kuyruk yöneticisini de yönetebilir. Etki alanı denetleyicisinde, etki alanı içinde etkin olan tüm ayrıcalıklı kullanıcı kimliklerini içeren bir etki alanı **mqm** grubu oluşturabilir ve bunu yerel **mqm** grubuna ekleyebilirsiniz. Bazı komutlar (örneğin, **crtmqm**), IBM MQ nesneleri üzerindeki yetkileri işler ve bu nedenle bu nesnelerle çalışmak için yetki gerekir (aşağıdaki bölümlerde açıklandığı gibi). **mqm** grubunun üyeleri tüm nesnelerle çalışma yetkisine sahiptir, ancak yerel bir kullanıcıysanız ve aynı ada sahip etki alanı kimliği doğrulanmış bir kullanıcıysanız, Windows sistemlerinde yetki verilmediğinde bazı durumlar olabilir. Bu, “AIX, Linux, and Windows üzerindeki asıl adlar ve gruplar” sayfa 419içinde açıklanmıştır.

Kullanıcı Hesabı Denetimi (UAC) özelliğine sahip Windows sürümleri, Yöneticiler grubunun üyesi olsalar da kullanıcıların belirli işletim sistemi olanakları üzerinde gerçekleştirebileceği işlemleri sınırlar. Kullanıcı kimliğiniz Administrators (Yöneticiler) grubundaysa, ancak **mqm** grubundaysa, **crtmqm**gibi IBM MQ yönetim komutlarını vermek için yükseltilmiş bir komut istemi kullanmalısınız; tersi durumda, `AMQ7077: İstenen işlemi gerçekleştirme yetkiniz yok` oluşturulur. Yükseltilmiş bir komut istemi açmak için, komut istemi için Başlat menü öğesini ya da simgesini sağ tıklatın ve **Yönetici olarak çalıştır**seçeneğini belirleyin.

Aşağıdaki işlemleri gerçekleştirmek için **mqm** grubunun üyesi olmanıza gerek yoktur:

- Komutlar kanal başlatıcılarını işlemediği sürece, PCF komutlarını ya da bir Escape PCF komutunda MQSC komutlarını veren bir uygulama programından komut verin. (Bu komutlar "Kanal başlatıcı tanımlarının korunması" sayfa 114 içinde açıklanmıştır).
- Bir uygulama programından MQI çağrıları yayınlayın (`MQCONNX` çağrısında hızlı yol bağ tanımlarını kullanmak istemiyorsanız).
- Veri tipi yapılarında veri dönüştürme gerçekleştiren bir kod parçası yaratmak için `crtmqcvx` komutunu kullanın.
- Kuyruk yöneticilerini görüntülemek için `dspmq` komutunu kullanın.
- IBM MQ biçimli izleme çıkışını görüntülemek için `dspmqtrc` komutunu kullanın.

12 karakter sınırlaması hem grup kimlikleri hem de kullanıcı kimlikleri için geçerlidir.

UNIX and Linux altyapıları genellikle kullanıcı kimliğinin uzunluğunu 12 karakterle sınırlar. AIX 5.3 bu sınırı artırdı, ancak IBM MQ tüm UNIX and Linux platformlarında 12 karakterlik bir kısıtlamayı gözlemlemeye devam ediyor. 12 karakterden uzun bir kullanıcı kimliği kullanırsanız, IBM MQ bu kullanıcı kimliğini `UNKNOWN`değeriyle değiştirir. `UNKNOWN`değeriyle bir kullanıcı kimliği tanımlamayın.

ALW AIX, Linux, and Windows üzerinde mqm grubunu yönetme

mqm grubundaki kullanıcılara IBM MQüzerinde tam denetim ayrıcalıkları verilir. Bu nedenle, mqm grubuna uygulamaları ve sıradan kullanıcıları kaydetmemelisiniz. mqm grubu yalnızca IBM MQ yöneticilerinin hesaplarını içermelidir.

Bu görevler aşağıda açıklanmıştır:

- Windows Windows üzerinde grup yaratılması ve yönetilmesi
- AIX AIX üzerinde grup yaratılması ve yönetilmesi
- Linux Linux üzerinde grup yaratılması ve yönetilmesi

Windows Etki alanı denetleyiciniz Windows 2000 ya da Windows 2003 ya da sonraki bir yayın düzeylerinde çalışıyorsa, etki alanı yöneticinizin IBM MQ ' in kullanması için özel bir hesap oluşturması gerekebilir. Daha fazla bilgi için bkz. IBM MQ ürününü Prepare IBM MQ Wizard ile yapılandırma ve IBM MQiçin Windows etki alanı hesaplarını oluşturma ve ayarlama.

ALW AIX, Linux, and Windows üzerindeki IBM MQ nesneleriyle çalışma yetkisi

Tüm nesneler IBM MQtarafından korunur ve asıl adlara bunlara erişmek için uygun yetki verilmelidir. Farklı birincil kullanıcılar farklı nesnelere farklı erişim hakları gerektir.

Kuyruk yöneticileri, kuyruklar, süreç tanımlamaları, ad alanları, kanallar, istemci bağlantısı kanalları, dinleyiciler, hizmetler ve kimlik doğrulama bilgileri nesnelerinin tümüne MQI çağrıları ya da PCF komutları kullanan uygulamalardan erişilir. Bu kaynakların tümü IBM MQtarafından korunur ve uygulamalara erişmeleri için izin verilmesi gerekir. İsteği yapan varlık bir kullanıcı, MQI çağrısı yapan bir uygulama programı ya da PCF komutu veren bir denetim programı olabilir. İstekte bulunanın tanıtıcısı *asıl ad*olarak adlandırılır.

Farklı birincil kullanıcı gruplarına aynı nesne için farklı erişim yetkisi tipleri atanabilir. Örneğin, belirli bir kuyruk için, bir grubun hem koyma hem de alma işlemlerini gerçekleştirmesine izin verilebilir; başka bir gruba yalnızca kuyruğa göz atma (göz atma seçeneğiyle `MQGET`) izni verilebilir. Benzer şekilde, bazı gruplar bir kuyruğa koyma ve alma yetkisine sahip olabilir, ancak kuyruğun özniteliklerini değiştirme ya da silme izni verilmez.

Bazı işlemler özellikle hassastır ve ayrıcalıklı kullanıcılarla sınırlı olmalıdır. Örneğin:

- İletim kuyrukları ya da komut kuyruğu SYSTEM.ADMIN.COMMAND.QUEUE

- Tam MQI bağlamı seçeneklerini kullanan programları çalıştırma
- Uygulama kuyruklarının yaratılması ve silinmesi

Nesneyi yaratan kullanıcı kimliğine ve mqm grubunun tüm üyelerine (ve Windows sistemlerinde yerel denetimciler grubunun üyelerine) otomatik olarak bir nesneye tam erişim izni verilir.

İlgili kavramlar

"AIX, Linux, and Windows üzerinde IBM MQ yönetimi yetkisi" sayfa 414

IBM MQ yöneticileri tüm IBM MQ komutlarını kullanabilir ve diğer kullanıcılara yetki verebilir. Denetimciler uzak kuyruk yöneticilerine komut verdiklerinde, uzak kuyruk yöneticisi üzerinde gerekli yetkiye sahip olmaları gerekir. Windows sistemleri için daha fazla dikkat edilmesi gereken noktalar vardır.

ALW AIX, Linux, and Windows üzerinde güvenlik denetimleri yapıldığında

Güvenlik denetimleri genellikle bir kuyruk yöneticisine bağlanma, nesneleri açma ya da kapatma ve ileti koyma ya da alma üzerinde yapılır.

Tipik bir uygulama için yapılan güvenlik denetimleri şunlardır:

Kuyruk yöneticisiyle bağlantı kuruluyor (MQCONN ya da MQCONNX çağrıları)
Bu, uygulamanın belirli bir kuyruk yöneticisiyle ilk kez ilişkilendirildiği zamandır. Kuyruk yöneticisi, uygulamayla ilişkili kullanıcı kimliğini keşfetmek için işletim ortamını sorguluyor. IBM MQ daha sonra, kullanıcı kimliğinin kuyruk yöneticisine bağlanma yetkisi olduğunu doğrular ve daha sonra yapılacak denetimler için kullanıcı kimliğini korur.

Kullanıcıların IBM MQ' da oturum açmaları gerekmez; IBM MQ , kullanıcıların temel işletim sisteminde oturum açtılarını ve bu işletim sistemi tarafından kimlik doğrulandığını varsayar.

Nesnenin açılması (MQOPEN ya da MQPUT1 çağrıları)
IBM MQ nesnelerine, nesne açılarak ve nesneye ilişkin komutlar verilerek erişilir. Tüm kaynak denetimleri, nesneye gerçekten erişildiğinde değil, nesne açıldığında gerçekleştirilir. Bu, **MQOPEN** isteğinin gereken erişim tipini belirtmesi gerektiği anlamına gelir (örneğin, kullanıcının yalnızca nesneye göz atmak mı, yoksa iletileri kuyruğa koymak gibi bir güncelleme mi gerçekleştirmek istediği).

IBM MQ , **MQOPEN** isteğinde adı belirtilen kaynağı denetler. Bir diğer ad ya da uzak kuyruk nesnesi için kullanılan yetki, diğer adın ya da uzak kuyruğun çözüldüğü kuyruk değil, nesnenin kendisidir. Bu, kullanıcının buna erişmek için izne ihtiyacı olmadığı anlamına gelir. Ayrıcalıklı kullanıcılar için kuyruk yaratma yetkisini sınırlayın. Bunu yapmazsanız, kullanıcılar bir diğer ad yaratarak normal erişim denetimini atlayabilir. Uzak bir kuyruğa hem kuyruk hem de kuyruk yöneticisi adlarıyla belirtik olarak gönderme yaparsanız, uzak kuyruk yöneticisiyle ilişkili iletim kuyruğu denetlenir.

Dinamik bir kuyruk üzerindeki yetki, türetildiği model kuyruğuna dayalıdır, ancak mutlaka aynı değildir. Bu, Not "1" sayfa 132içinde açıklanır.

Kuyruk yöneticisi tarafından erişim denetimleri için kullanılan kullanıcı kimliği, kuyruk yöneticisine bağlı uygulamanın işletim ortamından elde edilen kullanıcı kimliğidir. Uygun yetkili bir uygulama, alternatif bir kullanıcı kimliği belirterek bir **MQOPEN** çağrısı yapabilir; daha sonra diğer kullanıcı kimliğiyle erişim denetimi denetimleri yapılır. Bu, yalnızca erişim denetimi denetimleri için kullanılan uygulamayla ilişkili kullanıcı kimliğini değiştirmez.

İletiler yerleştiriliyor ve alınıyor (MQPUT ya da MQGET çağrıları)
Erişim denetimi denetimi gerçekleştirilmez.

Nesne kapatılıyor (MQCLOSE)
MQCLOSE , dinamik bir kuyruğun silinmesine neden olmadıkça, erişim denetimi denetimi denetimi gerçekleştirilmez. Bu durumda, kullanıcı kimliğinin kuyruğu silme yetkisi olup olmadığını denetleyin.

Konuya abone olma (MQSUB)
Bir uygulama bir konuya abone olduğunda, gerçekleştirmesi gereken işlem tipini belirtir. Yeni bir abonelik yaratıyor, var olan bir aboneliği değiştiriyor ya da var olan bir aboneliği değiştirmeden sürdürüyor. Kuyruk yöneticisi, her işlem tipi için, uygulamayla ilişkili kullanıcı kimliğinin işlemi gerçekleştirme yetkisi olup olmadığını denetler.

Bir uygulama bir konuya abone olduğunda, yetki denetimleri, uygulamanın abone olduğu konu ağacında ya da konu ağacının üstünde bulunan konu nesneleriyle ilgili olarak gerçekleştirilir. Yetki denetimleri birden çok konu nesnesi üzerinde denetimler içerebilir.

Kuyruk yöneticisinin yetki denetimleri için kullandığı kullanıcı kimliği, uygulama kuyruk yöneticisine bağlandığında işletim sisteminden alınan kullanıcı kimliğidir.

Kuyruk yöneticisi, yönetilen kuyruklarda değil, abone kuyruklarında yetki denetimi gerçekleştirir.

ALW Erişim denetiminin AIX, Linux, and Windows üzerinde IBM MQ tarafından nasıl uygulandığı

IBM MQ , nesne yetki yöneticisini kullanarak temel işletim sistemi tarafından sağlanan güvenlik hizmetlerini kullanır. IBM MQ , erişim denetim listeleri oluşturmak ve bunların bakımını yapmak için komutlar sağlar.

Yetkilendirme Hizmeti Arabirimi adı verilen bir erişim denetimi arabirimi, IBM MQ' in bir parçasıdır. IBM MQ , *nesne yetki yöneticisi (OAM)*olarak bilinen bir erişim denetimi yöneticisinin (Yetki Hizmeti Arabirimi 'ne uygun) uygulamasını sağlar. Bu, yarattığınız her kuyruk yöneticisi için otomatik olarak kurulur ve etkinleştirilir; tersi belirtilmedikçe (“AIX, Linux, and Windows sistemlerinde güvenlik erişimi denetimlerinin önlenmesi” sayfa 378 başlıklı konuda açıklandığı gibi). OAM, Yetki Hizmeti Arabirimi 'ne uyan herhangi bir kullanıcı ya da satıcı tarafından yazılan bileşen tarafından değiştirilebilir.

OAM, işletim sistemi kullanıcı ve grup kimliklerini kullanarak temel işletim sisteminin güvenlik özelliklerinden yararlanır. Kullanıcılar IBM MQ nesnelerine yalnızca doğru yetkiye sahiplerse erişebilir. “AIX, Linux, and Windows üzerinde OAM kullanarak nesnelere erişimi denetleme” sayfa 368 içinde bu yetkiye nasıl izin verileceği ve yetkiyi nasıl iptal edeceği açıklanır.

OAM, denetlediğiniz her kaynak için bir erişim denetimi listesi (ACL) sağlar. Yetki verileri, SYSTEM.AUTH.DATA.QUEUE. Bu kuyruğa erişim, mqm grubundaki kullanıcılarla ve ek olarak Windowsüzerindeki kullanıcılarla, Administrators (Yöneticiler) grubundaki kullanıcılarla ve SYSTEM ID ile oturum açan kullanıcılarla sınırlıdır. Kuyruğa kullanıcı erişimi değiştirilemiyor.

IBM MQ , erişim denetim listeleri oluşturmak ve bunların bakımını yapmak için komutlar sağlar. Bu komutlara ilişkin ek bilgi için bkz. “AIX, Linux, and Windows üzerinde OAM kullanarak nesnelere erişimi denetleme” sayfa 368.

IBM MQ , OAM ' ye bir birincil kullanıcı, kaynak adı ve erişim tipi içeren bir istek iletir. OAM, sağladığı EDL ' ye dayalı olarak erişimi verir ya da reddeder. IBM MQ , OAM ' nin kararını izler; OAM bir karar veremezse, IBM MQ erişime izin vermez.

ALW AIX, Linux, and Windows üzerinde kullanıcı kimliğini belirleme

Nesne yetki yöneticisi, bir kaynağa erişim isteyen birincil kullanıcıyı tanıtır. Birincil kullanıcı olarak kullanılan kullanıcı kimliği bağlama göre değişir.

Nesne yetki yöneticisi (OAM), belirli bir kaynağa kimlerin erişim istediğini saptayabilmelidir. IBM MQ , bu tanıtıcıya başvurmak için *birincil kullanıcı* terimini kullanır. Birincil kullanıcı, uygulama kuyruk yöneticisine ilk kez bağlandığında oluşturulur; kuyruk yöneticisi, bağlanan uygulamayla ilişkili kullanıcı kimliğinden saptanır. (Uygulama, kuyruk yöneticisine bağlanmadan XA çağrıları verirse, kuyruk yöneticisi tarafından yetki denetimleri için xa_open çağrısını veren uygulamayla ilişkili kullanıcı kimliği kullanılır.)

AIX and Linux sistemlerinde yetkilendirme yordamları, uygulamayla ilişkili gerçek (logged-in) kullanıcı kimliğini ya da etkin kullanıcı kimliğini denetler. Denetlenen kullanıcı kimliği bağ tanımlama tipine bağımlı olabilir; ayrıntılar için bkz. Kurulabilir hizmetler.

IBM MQ , her iletinin ileti üstbilgisinde (MQMD yapısı) sistemden alınan kullanıcı kimliğini kullanıcının tanıtıcısı olarak kullanır. Bu tanıtıcı, ileti bağlamı bilgilerinin bir parçasıdır ve “AIX, Linux, and Windows üzerinde bağlam yetkisi” sayfa 421içinde açıklanmıştır. Uygulamalar, bağlam bilgilerini değiştirme yetkisine sahip olmadıkları sürece bu bilgileri değiştiremezler.

ALW *AIX, Linux, and Windows üzerindeki asıl adlar ve gruplar*

Birincil kullanıcılar gruplara ait olabilir. Kişilere değil, gruplara kaynak erişimi vererek, gereken yönetim miktarını azaltabilirsiniz. Erişim Denetim Listeleri (EDL) hem gruplara, hem de kullanıcı kimliklerine dayalıdır.

Örneğin, belirli bir uygulamayı çalıştırmak isteyen kullanıcılardan oluşan bir grup tanımlayabilirsiniz. Diğer kullanıcılara, kullanıcı kimliklerini uygun gruba ekleyerek gereksinim duydukları tüm kaynaklara erişim izni verilebilir.

Grupları tanımlama ve yönetme süreci belirli platformlar için açıklanmıştır:

- AIX AIX üzerinde grup yaratılması ve yönetilmesi
- Linux Linux üzerinde grup yaratılması ve yönetilmesi
- Windows Windows üzerinde grup yaratılması ve yönetilmesi

Bir birincil kullanıcı birden çok gruba (grup kümesi) ait olabilir. Grup kümesindeki her gruba verilen tüm yetkilerin toplamını içeriyor. Bu yetkiler önbelleğe alınır; bu nedenle, **REFRESH SECURITY** (ya da PCF eşdeğeri) MQSC komutunu çalıştırmadığınız sürece, birincil kullanıcının grup üyeliğinde yaptığınız değişiklikler kuyruk yöneticisi yeniden başlatılıncaya kadar tanınmaz.

Linux AIX AIX and Linux sistemleri

IBM MQ 8.0'den erişim denetim listeleri (EDL' ler) hem kullanıcı kimliklerini hem de grupları temel alır ve **SecurityPolicy** özniteliğini qm.ini dosyasının hizmet kısmı ve AIX and Linux üzerinde yetkilendirme hizmeti stanzlarının yapılandırılmasıkonusunda açıklandığı gibi uygun değere ayarlayarak yetki için kullanabilirsiniz.

IBM MQ 8.0' den yetkilendirme için *kullanıcı tabanlı modeli* kullanabilirsiniz ve bu, hem kullanıcıları hem de grupları kullanmanıza olanak sağlar. Ancak, setmqaut komutunda bir kullanıcı belirttiğinizde, yeni izinler kullanıcının ait olduğu gruplar için değil, yalnızca o kullanıcı için geçerlidir. Daha fazla bilgi için bkz “AIX and Linux üzerinde OAM kullanıcı tabanlı izinler” sayfa 368.

Yetkilendirme için *grup tabanlı modeli* kullandığınızda, kullanıcı kimliğinin ait olduğu birincil grup EDL ' de yer alır. Tek tek kullanıcı kimliği dahil edilmez ve bu grubun tüm üyelerine yetki verilir. Bu nedenle, aynı gruptaki başka bir birincil kullanıcının yetkisini değiştirerek, bir birincil kullanıcının yetkisini yanlışlıkla değiştirebileceğinizi unutmayın.

Tüm kullanıcılar varsayılan kullanıcı grubuna (no) atanır ve varsayılan olarak bu grup için yetki verilmez. Belirli yetkilere sahip olmayan kullanıcılara IBM MQ kaynaklarına erişim vermek için `kimse` grubundaki yetkilendirmeyi değiştirebilirsiniz.

V 9.3.0 Kimden IBM MQ 9.3.0, işletim sistemi olmayan bir kullanıcı adı yaratmak için **SecurityPolicy** özniteliğinin `UserExternal` seçeneğini kullanabilirsiniz. İşletim sistemi olmayan bir kullanıcı adı oluşturursanız, o kullanıcının `nobody` grubu dışında hiçbir gruba ait olmadığı varsayılır. Bu seçenekle ilgili daha fazla bilgi için bkz. crtmqm ve qm.ini dosyasının hizmet kısmı.

`UNKNOWN`(Bilinmiyor) değeriyle bir kullanıcı kimliği tanımlamayın. Bir kullanıcı kimliği çok uzun olduğunda `UNKNOWN` (Bilinmiyor) değeri kullanılır; bu nedenle, keyfi kullanıcı kimlikleri `UNKNOWN`(Bilinmiyor) erişim yetkilerini kullanır.

LDAP kullanımına ilişkin bilgi için bkz. “Ayar yetkileri” sayfa 427 .

Kullanıcı kimlikleri en çok 12 karakter ve en çok 12 karakter içerebilir.

Windows Windows sistemleri

ACLs, hem kullanıcı kimliklerini hem de grupları temel alır. Denetimler, AIX and Linuxile aynıdır. Aynı kullanıcı kimliğine sahip farklı etki alanlarında farklı kullanıcılar olabilir. IBM MQ , kullanıcı kimliklerinin bir etki alanı adıyla nitelendirilmesine izin verir; böylece, bu kullanıcılara farklı erişim düzeyleri verilebilir.

Grup adı, isteğe bağlı olarak, aşağıdaki biçimlerde belirtilen bir etki alanı adını içerebilir:

```
GroupName@domain domain_name\group_name
```

Genel gruplar yalnızca iki durumda OAM tarafından denetlenir:

1. Kuyruk yöneticisi güvenlik kısmı şu ayarı içerir: `GroupModel=GlobalGroups`. Bkz. Güvenlik.
2. Kuyruk yöneticisi alternatif bir güvenlik erişimi grubu kullanıyor. Bkz. **crtmqm** .

Kullanıcı kimlikleri en çok 20 karakter, etki alanı adları en çok 15 karakter ve grup adları en çok 64 karakter içerebilir.

OAM önce yerel güvenlik veritabanını, ardından birincil etki alanının veritabanını ve son olarak güvenilir etki alanlarının veritabanını denetler. Karşılaşılan ilk kullanıcı kimliği, denetleme için OAM tarafından kullanılır. Bu kullanıcı kimliklerinin her birinin belirli bir bilgisayarda farklı grup üyelikleri olabilir.

Bazı denetim komutları (örneğin, **crtmqm**), nesne yetki yöneticisini (OAM) kullanarak IBM MQ nesnelerine ilişkin yetkileri değiştirir. OAM, belirli bir kullanıcı kimliğine ilişkin yetki haklarını belirlemek için güvenlik veritabanlarını önceki paragrafta belirtilen sırayla arar. Sonuç olarak, OAM tarafından belirlenen yetki, bir kullanıcı kimliğinin yerel mqm grubunun üyesi olduğu gerçeğini geçersiz kılabilir. Örneğin, **crtmqm** komutunu bir genel grup üzerinden yerel mqm grubu üyeliği olan bir etki alanı denetleyicisi tarafından doğrulanmış bir kullanıcı kimliğinden yayınlarsanız, sistemin yerel mqm grubunda olmayan aynı ada sahip bir yerel kullanıcısı varsa komut başarısız olur.

Windowsüzerinde **SecurityPolicy** özniteliğini ayarlama hakkında daha fazla bilgi için bkz. Kurulabilir hizmetler ve Windows 'ta yetkilendirme hizmeti stanzlarını yapılandırma.

Windows *Windows güvenlik tanıtıcıları (SID)*

IBM MQ on Windows , kullanılabilir olduğu yerde SID ' yi kullanır. Windows SID, bir yetkilendirme isteğiyle birlikte sağlanmazsa IBM MQ , kullanıcıyı yalnızca kullanıcı adına dayalı olarak tanımlar, ancak bu, yanlış yetki verilmesine neden olabilir.

Windows sistemlerinde, kullanıcı kimliğini tamamlamak için güvenlik tanıtıcısı (SID) kullanılır. SID, kullanıcının tanımlandığı Windows güvenlik hesabı yöneticisi (SAM) veritabanındaki tam kullanıcı hesabı ayrıntılarını tanımlayan bilgileri içerir. IBM MQ for Windowsüzerinde bir ileti oluşturulduğunda, IBM MQ SID ' yi ileti tanımlayıcısında saklar. IBM MQ on Windows yetki denetimleri gerçekleştirdiğinde, SAM veritabanından tam bilgileri sorgulamak için SID ' yi kullanır. (Bu sorgunun başarılı olması için, kullanıcının tanımlandığı SAM veritabanına erişilir olmalıdır.)

Varsayılan olarak, Windows SID bir yetkilendirme isteğiyle sağlanmazsa IBM MQ , kullanıcıyı yalnızca kullanıcı adına dayalı olarak tanımlar. Bunu, güvenlik veritabanlarını aşağıdaki sırayla arayarak yapar:

1. Yerel güvenlik veritabanı
2. Birincil etki alanının güvenlik veritabanı
3. Güvenilir etki alanlarının güvenlik veritabanı

Kullanıcı adı benzersiz değilse, yanlış IBM MQ yetkisi verilebilir. Bu sorunu önlemek için, her yetki isteğine bir SID ekleyin; SID, IBM MQ tarafından kullanıcı kimlik bilgilerini oluşturmak için kullanılır.

Tüm yetki isteklerinin bir SID içermesi gerektiğini belirtmek için **regedit**kullanın. SecurityPolicy öğesini NTSIDsRequiredolarak ayarlayın.

ALW AIX, Linux, and Windows üzerinde diğer kullanıcı yetkisi

Bir kullanıcı kimliğinin bir IBM MQ nesnesine erişirken başka bir kullanıcının yetkisini kullanabileceğini belirtebilirsiniz. Buna *diğer kullanıcı yetkisi*denir ve bunu herhangi bir IBM MQ nesnesinde kullanabilirsiniz.

Diğer kullanıcı yetkisi, bir sunucu bir programdan istek alırsa ve programın istek için gerekli yetkiye sahip olduğundan emin olmak isterse gereklidir. Sunucu gerekli yetkiye sahip olabilir, ancak programın istediği işlemlere ilişkin yetkiye sahip olup olmadığını bilmesi gerekir.

Örneğin, PAYSERV kullanıcı kimliği altında çalışan bir sunucu programının, USER1kullanıcı kimliği tarafından kuyruğa konan bir kuyruktan bir istek iletisi aldığını varsayın. Sunucu programı istek iletisini aldığında, isteği işler ve yanıtı istek iletisiyle belirlenen yanıt kuyruğuna geri koyar. Yanıt kuyruğunun açılmasına yetki vermek için kendi kullanıcı kimliğini (PAYSERV) kullanmak yerine, sunucu farklı bir kullanıcı kimliği belirtebilir; bu durumda, USER1. Bu örnekte, PAYSERV ' nin yanıt kuyruğunu açarken alternatif kullanıcı kimliği olarak USER1 belirtmesine izin verilip verilmediğini denetlemek için diğer kullanıcı yetkisini kullanabilirsiniz.

Diğer kullanıcı kimliği, nesne tanımlayıcısının **AlternateUserId** alanında belirtilir.

Linux Linux üzerinde belirli grup üyeliği sorunlarının çözülmesi

Bazı sistemler, **getgrent** işletim sistemi API çağrılarının olağan dizisi aracılığıyla grup bilgilerini geri döndürmekte yavaştır ve kuruluşunuzda aranacak binlerce grup varsa, mqm kullanıcısının hangi gruplarda olduğunu ararken yavaş yanıt bir iç kuyruk yöneticisi zamanaşımına neden olabilir. Bu sorunu önlemek için alternatif bir işletim sistemi API 'si vardır.

Daha hızlı olan alternatif API ' yi kullanmak ve tek bir çağrıdan tüm grupları döndürmek için MQS_GETGROUPLIST_API ortam değişkenini ayarlayın.

Kullanıcının ikincil grubuna bağlanma erişimi verirken ve MQS_GETGROUPLIST_API değişkenini etkinleştirirken RC2035 hatası almış olabilirsiniz.

IBM MQ daha sonra **getgrent** API yerine **getgrouplist** API 'sini kullanır.

getgrouplist' i etkinleştirmek için:

1. Kuyruk yöneticisini durdur
2. MQS_GETGROUPLIST_API=1 komutunu dışa aktarma komutunu verin
3. Kuyruk yöneticisini yeniden başlat

Başarısız olan senaryoyu yeniden deneyin ve sorununuz çözüldüyse, bu ortam değişkenini eklemek için .bashrc / .profile kullanıcı için dosyayı mqm değiştirmeyi ya da kuyruk yöneticisini başlatmak için kullandığınız komut dosyasına ortam değişkenini eklemeyi düşünebilirsiniz.

Sisteminiz, işletim sistemine ilişkin kullanıcı ya da grup bilgilerini NIS ya da LDAP gibi birden çok havuzdan birleştirirse, grup ya da kullanıcı kimliğinin, işletim sistemi düzeyi izinlerini kurmak ve ayarlamak için kullanıldığından, yerel havuzlar da içinde olmak üzere tüm havuzlarda tutarlı olduğundan emin olun.

ALW AIX, Linux, and Windows üzerinde bağlam yetkisi

Bağlam, belirli bir ileti için geçerli olan ve iletinin bir parçası olan ileti tanımlayıcısı MQMD ' de bulunan bilgilerdir. Uygulamalar, MQOPEN ya da MQPUT çağrısı yapıldığında bağlam verilerini belirtebilir.

Bağlam bilgileri iki kısımdan oluşur:

Kimlik bölümü
Mesajın geldiği kişi. UserIdentifier, AccountingTokenve ApplIdentityData alanlarından oluşur.

Kaynak bölümü
Mesajın nereden geldiği ve ne zaman kuyruğa konduğu. PutApplType, PutApplName, PutDate, PutTimeve ApplOriginData alanlarından oluşur.

Uygulamalar, MQOPEN ya da MQPUT çağrısı yapıldığında bağlam verilerini belirtebilir. Bu veriler uygulama tarafından oluşturulabilir, başka bir iletiden iletilebilir ya da varsayılan olarak kuyruk yöneticisi tarafından oluşturulabilir. Örneğin, bağlam verileri sunucu programları tarafından istekte bulunanın kimliğini denetlemek ve iletinin yetkili kullanıcı kimliği altında çalışan bir uygulamadan gelip gelmediğini sınamak için kullanılabilir.

Bir sunucu programı, alternatif bir kullanıcının kullanıcı kimliğini belirlemek için UserIdentifier ' i kullanabilir. Bağlam yetkisi, kullanıcının herhangi bir MQOPEN ya da MQPUT1 çağrısında bağlam seçeneklerinden herhangi birini belirtip belirtemeyeceğini denetlemek için kullanılır.

Bağlam seçenekleriyle ilgili bilgi için Bağlam bilgilerini denetleme başlıklı konuya ve bağlamla ilgili ileti tanımlayıcı alanlarının açıklamaları için MQMD-Message descriptor başlıklı konuya bakın.

Güvenlik çıkışlarında erişim denetimi uygulanması

MCAUserIdentifier ya da nesne yetki yöneticisini kullanarak bir güvenlik çıkışında erişim denetimi uygulayabilirsiniz.

MCAUserIdentifier

Yürürlükteki bir kanalın her eşgörünümünün ilişkili bir kanal tanımlama yapısı (MQCD) vardır. MQCD ' deki alanların ilk değerleri, IBM MQ denetimcisi tarafından yaratılan kanal tanımlamasıyla saptanır. Özellikle, alanlardan birinin ilk değeri olan *MCAUserIdentifier*, DEFINE CHANNEL komutundaki MCAUSER parametresinin değerine göre ya da kanal tanımı başka bir şekilde oluşturulduysa, MCAUSER ' in eşdeğerine göre belirlenir.

MQCD yapısı, bir MCA tarafından çağrıldığında kanal çıkış programına geçirilir. Bir güvenlik çıkışı MCA tarafından çağrıldığında, güvenlik çıkışı *MCAUserIdentifier*değerini değiştirerek kanal tanımında belirtilen herhangi bir değeri değiştirebilir.

Multi Çoklu platformlarsistemlerinde, *MCAUserIdentifier* değeri boş değilse, kuyruk yöneticisi *MCAUserIdentifier* değerini, bir MCA kuyruk yöneticisine bağlandıktan sonra kuyruk yöneticisinin kaynaklarına erişmeye çalıştığında yetki denetimi için kullanıcı kimliği olarak kullanır. *MCAUserIdentifier* değeri boşsa, kuyruk yöneticisi MCA ' nın varsayılan kullanıcı kimliğini kullanır. Bu, RCVR, RQSTR, CLUSRCVR ve SVRCONN kanalları için geçerlidir. MCA ' ları göndermek için, *MCAUserIdentifier* değeri boş olmasa da, yetki denetimlerinde varsayılan kullanıcı kimliği her zaman kullanılır.

z/OS z/OSsistemlerinde, kuyruk yöneticisi boş değilse, yetki denetimleri için *MCAUserIdentifier* değerini kullanabilir. MCA 'ları ve sunucu bağlantısı MCA' larını almak için, kuyruk yöneticisinin yetki denetimleri için *MCAUserIdentifier* değerini kullanıp kullanmadığı şuna bağlıdır:

- Kanal tanımındaki PUTAUT parametresinin değeri
- Denetimler için kullanılan RACF profili
- RESLEVEL tanıtımına ilişkin kanal başlatıcı adres alanı kullanıcı kimliğinin erişim düzeyi

MCA ' ları göndermek için aşağıdakilere bağlıdır:

- Gönderen MCA ' nın arayan mı, yoksa yanıt veren mi olduğu
- RESLEVEL tanıtımına ilişkin kanal başlatıcı adres alanı kullanıcı kimliğinin erişim düzeyi

MCAUserIdentifier içindeki bir güvenlik çıkışının depoladığı kullanıcı kimliği çeşitli şekillerde edinilebilir. Bazı örnekler:

- Bir MQI kanalının istemci ucunda güvenlik çıkışı yoksa, istemci uygulaması bir MQCONN çağrısı yayınladığında, IBM MQ istemci uygulamasıyla ilişkilendirilmiş bir kullanıcı kimliği istemci bağlantısı MCA 'dan sunucu bağlantısı MCA' ya akar.Sunucu bağlantısı MCA, bu kullanıcı kimliğini MQCD ' nin kanal tanımlaması yapısındaki *RemoteUserIdentifier* (Uzak Kullanıcı Tanıtıcısı) alanında saklar. *MCAUserIdentifier* değeri şu anda boşsa, MCA aynı kullanıcı kimliğini *MCAUserIdentifier*içinde saklar. MCA, kullanıcı kimliğini *MCAUserIdentifier*içinde saklamazsa, daha sonra *MCAUserIdentifier* değerini *RemoteUserIdentifier*değerine ayarlayarak bir güvenlik çıkışı bunu yapabilir.

  İstemci sistemden akan kullanıcı kimliği yeni bir güvenlik etki alanı giriyorsa ve sunucu sisteminde geçerli değilse, güvenlik çıkışı geçerli olan kullanıcı kimliğini yerine alabilir ve yerine konan kullanıcı kimliğini *MCAUserIdentifier*içinde saklayabilir.
- Kullanıcı kimliği, iş ortağı güvenlik çıkışı tarafından bir güvenlik iletisinde gönderilebilir.

  Bir ileti kanalında, gönderen MCA tarafından çağrılan bir güvenlik çıkışı, gönderen MCA ' nın altında çalıştığı kullanıcı kimliğini gönderebilir. Alıcı MCA tarafından çağrılan bir güvenlik çıkışı, kullanıcı kimliğini *MCAUserIdentifier*içinde saklayabilir. Benzer şekilde, bir MQI kanalında, kanalın istemci ucundaki bir güvenlik çıkışı IBM MQ MQI client uygulamasıyla ilişkili kullanıcı kimliğini gönderebilir. Daha sonra, kanalın sunucu ucundaki bir güvenlik çıkışı kullanıcı kimliğini *MCAUserIdentifier*içinde

saklayabilir. Önceki örnekte olduğu gibi, kullanıcı kimliği hedef sistemde geçerli değilse, güvenlik çıkışı geçerli olan kullanıcı kimliğinin yerine kullanıcı kimliğini alabilir ve yerine konan kullanıcı kimliğini *MCAUserIdentifier*içinde saklayabilir.

Kimlik doğrulama ve kimlik doğrulama hizmetinin bir parçası olarak bir sayısal sertifika alınırsa, güvenlik çıkışı sertifikadaki Ayırt Edici Adı hedef sistemde geçerli olan bir kullanıcı kimliğiyle eşleyebilir. Daha sonra kullanıcı kimliğini *MCAUserIdentifier*içinde saklayabilir.

- Kanalda TLS kullanılırsa, iş ortağının Ayırt Edici Adı (DN) MQCD 'nin SSLPeerNamePtr alanındaki çıkışa iletilir ve bu sertifikayı veren kişinin DN 'si, MQCXP' nin SSLRemCertIssNamePtr alanındaki çıkışa iletilir.

MCAUserIdentifier alanı, kanal tanımlama yapısı, MQCD ve kanal çıkış parametresi yapısı, MQCXP hakkında daha fazla bilgi için bkz. Kanal çıkış çağrıları ve veri yapıları. MQI kanalındaki bir istemci sisteminden akan kullanıcı kimliğiyle ilgili daha fazla bilgi için Erişim denetimibaşlıklı konuya bakın.

Not: IBM WebSphere MQ 7.1 yayın düzeyinden önce oluşturulan güvenlik çıkışı uygulamalarının güncellenmesi gerekebilir. Daha fazla bilgi için bkz. Kanal güvenliği çıkış programları.

IBM MQ nesne yetkisi yöneticisi kullanıcı kimlik doğrulaması

IBM MQ MQI client bağlantılarında, nesne yetkisi yöneticisi (OAM) kullanıcı kimlik doğrulamasında kullanılan MQCSP yapısını değiştirmek ya da yaratmak için güvenlik çıkışları kullanılabilir. Bu, İleti sistemi kanalları için kanal çıkış programları içinde açıklanmıştır.

İleti çıkışlarında erişim denetimi uygulanıyor

Bir kullanıcı kimliğini başka bir kullanıcı kimliğiyle değiştirmek için ileti çıkışını kullanmanız gerekebilir.

Sunucu uygulamasına ileti gönderen bir istemci uygulamasını düşünün. Sunucu uygulaması, kullanıcı kimliğini ileti tanımlayıcısındaki *UserIdentifier* alanından çıkarabilir ve diğer kullanıcı yetkisi olması koşuluyla, kuyruk yöneticisinden istemci adına IBM MQ kaynaklarına eriştiğinde yetki denetimi için bu kullanıcı kimliğini kullanmasını isteyin.

PUTAUT parametresi CTX (ya da z/OS üzerinde ALTMCA) olarak ayarlanırsa Kanal tanımında, her gelen iletinin *UserIdentifier* alanındaki kullanıcı kimliği, MCA hedef kuyruğu açtığında yetki denetimi için kullanılır.

Belirli durumlarda, bir rapor iletisi oluşturulduğunda, rapora neden olan iletinin *UserIdentifier* alanındaki kullanıcı kimliği yetkisi kullanılarak oluşturulur. Özellikle, teslimatta onayla (COD) raporları ve süre sonu raporları her zaman bu yetkiyle birlikte kullanılır.

Bu durumlar nedeniyle, ileti yeni bir güvenlik etki alanı girerken *UserIdentifier* alanında bir kullanıcı kimliğinin yerine başka bir kullanıcı kimliği konması gerekebilir. Bu, kanalın alıcı ucundaki bir ileti çıkışıyla yapılabilir. Diğer bir seçenek olarak, gelen bir iletinin *UserIdentifier* alanındaki kullanıcı kimliğinin yeni güvenlik etki alanında tanımlandığından emin olun.

Gelen bir ileti, iletiyi gönderen uygulamanın kullanıcısı için bir sayısal sertifika içeriyorsa, bir ileti çıkışı sertifikayı doğrulayabilir ve sertifikadaki Ayırt Edici Adı, alan sistemde geçerli olan bir kullanıcı kimliğiyle eşleyebilir. Daha sonra, ileti tanımlayıcısındaki *UserIdentifier* alanını bu kullanıcı kimliğine ayarlayabilir.

Bir ileti çıkışının gelen iletideki *UserIdentifier* alanının değerini değiştirmesi gerekiyorsa, ileti çıkışının iletiyi gönderenin kimliğini aynı anda doğrulaması uygun olabilir. Daha fazla ayrıntı için bkz. “İleti çıkışlarında kimlik eşleme” sayfa 343.

API çıkışında ve API geçişi çıkışında erişim denetimi uygulanıyor

Bir API ya da API geçiş çıkışı, IBM MQtarafından sağlananları tamamlamak için erişim denetimleri sağlayabilir. Özellikle, çıkış ileti düzeyinde erişim denetimi sağlayabilir. Çıkış, bir uygulamanın kuyruğa girmesini ya da yalnızca belirli ölçütlere uyan iletileri kuyruktan almasını sağlar.

Aşağıdaki örnekleri göz önünde bulundurun:

- Bir ileti, bir siparişe ilişkin bilgileri içerir. Bir uygulama bir iletiyi kuyruğa koymayı denediğinde, bir API ya da API geçişi çıkışı, siparişin toplam değerinin, belirtilen bir sınırın altında olup olmadığını denetleyebilir.

- İletiler uzak kuyruk yöneticilerinden hedef kuyruğa ulaşır. Bir uygulama kuyruktan ileti almaya çalıştığında, bir API ya da API geçişi çıkışı, iletiyi gönderenin kuyruğa ileti gönderme yetkisinin olup olmadığını denetleyebilir.

V 9.3.0 Multi Akış kuyrukları güvenliği

Akıtmalı kuyruklar özelliği, bir yöneticinin, özgün kuyruğa bir ileti yerleştirildiğinde, yinelenen iletilerin yerleştirildiği ikincil bir kuyrukla yerel (ya da model) bir kuyruk yapılandırmasını sağlar. Kuyruk akışı yetkilileriyle ilgili göz önünde bulundurulması gereken iki konu vardır.

Akıtmalı yinelenen iletiler için bir kuyruk yapılandırma yetkisi

Yinelenen iletilerin bir kuyruktan ikincil bir kuyruğa akışını etkinleştirmek istiyorsanız, bunu yapmak için izniniz olmalıdır. Bir kuyruğun **STREAMQ** özniteliğini yapılandırma izni için aşağıdaki yetkilere sahip olmanız gerekir:

1. Şunun için **STREAMQ** özniteliğini değiştirdikleri kuyruğun CHG yetkisi
2. Yinelenen iletilerin konmasını istediğiniz kuyruğa ilişkin CHG yetkisi

Konfigürasyon sırasında bu iki yetki denetiminin birleşimi, yalnızca özgün kuyrukta CHG yetkisi olan bir kullanıcının, iletilerin üzerinde izinleri olmayan başka bir kuyruğa konmasına neden olmamasını sağlar.

Kuyruğu ya da kuyrukları açma ve ileti koyma yetkisi

Bir uygulama, ikincil bir kuyrukla yapılandırılmış bir kuyruğu **STREAMQ** özniteliği aracılığıyla açtığında, uygulama kullanıcısının özgün kuyruk üzerinde PUT yetkisi olduğuna ilişkin bir yetki denetimi yapılır.

Not: İkincil kuyruktaki uygulama kullanıcısı için ek yetki denetimi yapılmaz; bu, diğer ad kuyrukları için kullanılan yetki modeline benzer.

Özgün ya da ikincil kuyruktan ileti alan uygulamalar, yalnızca tükettikleri kuyrukta GET ya da BROWSE yetkisi gerektirir.

Koyma ya da alma sırasında ek yetki denetimi yapılmaz.

Örnek

Aşağıdaki örnekte, `admin` kullanıcısının özgün bir kuyruğu (INQUIRIES.QUEUE, yinelenen iletilerini ANALYTICS.QUEUE, ancak `admin` ' ın iletileri 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)
```

Daha sonra `admin` kullanıcısı şu komutu yayınlayabiliyor:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

ancak, aynı kullanıcı aşağıdaki komutu verir:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

INQUIRIES.QUEUE 'i PURCHASES.QUEUE' e yinelenen iletileri koyacak şekilde yapılandırmak için aşağıdaki hatayı alırlar:

```
AMQ8135E Yetki Yok
```

INQUIRIES.QUEUE , iletileri ANALYTICS.QUEUE, aşağıdaki yetki kayıtları, `appuser` kullanıcısı olarak çalışan bir uygulamanın iletileri INQUIRIES.QUEUEve iletileri ANALYTICS.QUEUE:

```
SET AUTHREC PROFILE(INQUIRIES.QUEUE) PRINCIPAL('appuser') AUTHADD(PUT)
```

Not: `appuser` , ANALYTICS.QUEUE. Yinelenen iletiler kuyruk yöneticisi tarafından kuyruğa yerleştirilir.

İlgili kavramlar

Akış kuyrukları

V 9.3.0 z/OS z/OS üzerinde akış kuyrukları güvenliği

Akıtmalı kuyruklar özelliği, bir yöneticinin, özgün kuyruğa bir ileti yerleştirildiğinde, yinelenen iletilerin yerleştirildiği ikincil bir kuyrukla yerel (ya da model) bir kuyruk yapılandırmasını sağlar. Kuyruk akışı yetkilileriyle ilgili göz önünde bulundurulması gereken iki konu vardır.

Akıtmalı yinelenen iletiler için bir kuyruk yapılandırma yetkisi

Yinelenen iletilerin bir kuyruktan ikincil bir kuyruğa akışını etkinleştirmek istiyorsanız, bunu yapmak için izniniz olmalıdır. Bir kuyruğun **STREAMQ** özniteliğini yapılandırma izni için aşağıdaki profillerin ayarlanmış olması gerekir:

1. **STREAMQ** özniteliğini değiştirdikleri kuyruk için MQADMIN ya da MXADMIN için ALTER erişim düzeyi
2. İletileri akışa aktarmak istediğiniz kuyruk için MQADMIN ya da MXADMIN için ALTER erişim düzeyi

Yapılanış sırasında bu güvenlik denetimlerinin birleşimi, özgün kuyrukta yalnızca ALTER erişimi olan bir kullanıcının, iletilerin, izinleri olmayan başka bir kuyruğa konmasına neden olmamasını sağlar.

Kuyruğu ya da kuyrukları açma ve ileti koyma yetkisi

Bir uygulama, ikincil bir kuyrukla yapılandırılmış bir kuyruğu **STREAMQ** özniteliği aracılığıyla açtığında, uygulama kullanıcısının özgün kuyruk üzerinde UPDATE (güncelleme) yetkisine sahip olduğu için bir yetki denetimi yapılır.

Not: İkincil kuyruktaki uygulama kullanıcısı için ek yetki denetimi yapılmaz; bu, diğer ad kuyrukları için kullanılan yetki modeline benzer.

Özgün ya da ikincil kuyruktan ileti alan uygulamalar, yalnızca tükettikleri kuyrukta UPDATE ya da READ yetkisi gerektirir.

Koyma ya da alma sırasında ek yetki denetimi yapılmaz.

Örnek

Aşağıdaki örnekte, ADMIN kullanıcısının RACFkullanarak iletileri yerel kuyruğa aktarmak için INQUIRIES.QUEUEözgün bir kuyruk yapılandırmasına izin verecek şekilde ayarlanan doğru profiller gösterilmektedir: ANALYTICS.QUEUE :

```
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)
```

Daha sonra ADMIN kullanıcısı şu komutu yayınlayabiliyor:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(ANALYTICS.QUEUE)
```

ancak aynı kullanıcı doğru güvenlik profillerini ayarlamadan aşağıdaki komutu verir:

```
ALTER QLOCAL(INQUIRIES.QUEUE) STREAMQ(PURCHASES.QUEUE)
```

INQUIRIES.QUEUE 'i PURCHASES.QUEUE' e yinelenen iletileri koyacak şekilde yapılandırmak için aşağıdaki hatayı alırlar:

```
CSQM166I <QMGR> CSQMAQLC QLOCAL (INQUIRIES.QUEUE) YETKILI DEğIL
```

İlgili kavramlar

Akış kuyrukları

Multi LDAP Yetkilendirmesi

Yerel kullanıcı kimliği gereksinimini kaldırmak için LDAP yetkilendirmesini kullanabilirsiniz.

Desteklenen platformlarda LDAP yetkilendirmesinin kullanılabilirliği

LDAP yetkilendirmesi Multiplatforms üzerinde kullanılabilir:

Uyarı:

Bu işlev, IBM MQ 9.0 genel kullanılabilirliğinden, ister daha önceki bir yayın düzeyinden yeni ya da yeni düzeye geçiş olsun, tüm kuyruk yöneticilerinin kullanımına sunulur.

LDAP yetkilendirmesi hakkında genel bilgiler

LDAP yetkilendirmesiyle, **setmqaut** ve **DISPLAY AUTHREC**gibi yetkilendirme yapılandırmasını işleyen komutlar Ayırt Edici Adları işleyebilir. Daha önce, kullanıcıların kimlik bilgileri, yerel işletim sisteminde kullanıcılar ve gruplar için var olan maksimum kullanılabilir karakterlerle karşılaştırılarak doğrulanmıştı.

Uyarı: DEFINE AUTHINFO komutunu çalıştırdıysanız, kuyruk yöneticisini yeniden başlatmanız gerekir. Kuyruk yöneticisini yeniden başlatmazsanız, setmqaut komutu doğru sonucu döndürmez.

Bir kullanıcı Ayırt Edici Ad yerine bir kullanıcı kimliği sağlarsa, kullanıcı kimliği işlenir. Örneğin, PUTAUT (CTX) içeren bir kanalda gelen bir ileti varsa, kullanıcı kimliğindeki karakterler bir LDAP Ayırt Edici Adıyla eşlenir ve uygun yetki denetimleri yapılır.

DISPLAY CONNgibi diğer komutlar, kullanıcı kimliği yerel işletim sisteminde gerçekten var olmasa da, bu komutlarla çalışmaya devam eder ve kullanıcı kimliği için gerçek değeri gösterir.

Linux AIX LDAP yetkilendirmesi olduğunda kuyruk yöneticisi, `qm.ini` dosyasındaki **SecurityPolicy** özniteliğinden bağımsız olarak her zaman AIX and Linux platformlarında güvenliğin kullanıcı modelini kullanır. Bu nedenle, tek bir kullanıcı için izinlerin ayarlanması, o kullanıcının gruplarından herhangi birine ait olan başka bir kullanıcıyı değil, yalnızca o kullanıcıyı etkiler.

İşletim sistemi modelinde olduğu gibi, bir kullanıcı, kullanıcının ait olduğu hem kişiye hem de tüm gruplara (varsa) atanan birleşik yetkiye sahiptir.

Örneğin, aşağıdaki kayıtların bir LDAP havuzunda tanımlandığını varsayın.

- **inetOrgPerson** sınıfında:

```
dn="cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry"
        email=JohnDoe1@yourcompany.com [longer than 12 characters]
        shortu=jodoe
        Phone=1234567
```

- **groupOfNames** sınıfında:

```
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"
```

Kimlik doğrulama amacıyla, bu LDAP sunucusunu kullanan bir kuyruk yöneticisi, **CONNAUTH** değerinin IDPWLDAPtipindeki bir **AUTHINFO** nesnesini göstermesi ve ilgili ad-çözünürlük öznitelikleri büyük olasılıkla aşağıdaki gibi ayarlanmış olması için tanımlanmış olmalıdır:

```
USRFIELD(email) SHORTUSR(shortu)
BASEDNU(ou=users,o=yourcompany,c=yourcountry) CLASSUSR(inetOrgPerson)
```

Kimlik doğrulama için bu yapılandırma verildiğinde bir uygulama, MQCNO çağrısında kullanılan CSPUserID alanını aşağıdaki değer kümelerinden biriyle tamamlayabilir:

```
" cn=JohnDoe ", " JohnDoe1@yourcompany.com ", " email=JohnDoe1@yourcompany.com "
```

veya

```
" cn=JohnDoe, ou=users, o=ibm, c=uk ", " shortu=jodoe "
```

Her iki durumda da sistem, " jodoe".

Multi Ayar yetkileri

Yetkileri ayarlamak için kısa adı ya da **USRFIELD** kullanın.

"LDAP Yetkilendirmesi" sayfa 426içinde açıklanan birden çok biçimle çalışma yaklaşımı, shortname ya da USRFIELD ' in adsız bir şekilde kullanılabileceği bir uzantıyla yetkilendirme komutlarına devam eder.

Karakter dizgisi, yetkilendirme için kullanıcıları (asıl adlar) adlandırırken LDAP kaydında belirli bir özniteliği belirtir.

Önemli: Bu karakter işletim sistemi kullanıcı kimliklerinde kullanılamayacağı için, karakter dizgisi = karakterini içermemelidir.

Olası bir shortnameyetkisi için OAM ' ye bir birincil kullanıcı adı geçirirseniz, karakter dizgisinin 12 karaktere sığması gerekir. Eşleme algoritması, önce LDAP sorgusunda SHORTUSR özniteliğini kullanarak bir DN ' ye çözmeyi dener.

Bu bir UNKNOWN_ENTITY hatasıyla başarısız olursa ya da belirtilen dizgi bir shortnameolabilirse, LDAP sorgusunu oluşturmak için USRFIELD özniteliği kullanılarak daha fazla girişimde bulunulacaktır.

Uyarı: DEFINE AUTHINFO komutunu çalıştırdıysanız, kuyruk yöneticisini yeniden başlatmanız gerekir. Kuyruk yöneticisini yeniden başlatmazsanız, setmqaut komutu doğru sonucu döndürmez.

Kullanıcı yetkilerini işlemek için aşağıdaki setmqaut komut ayarlarının tümü eşdeğerdir.

Çizelge 75. Kullanıcı yetkilendirme ayarları

Komut	Not
`setmqaut -m QM -t qmgr -p jodoe +connect`	Bu düz, nitelenmemiş bir ad, SHORTUSR ile çözümlendi.
`setmqaut -m QM -t qmgr -pJohnDoe1@yourcompany.com +connect`	Ayrıca, USRFIELD aracılığıyla aynı varlığa çözümlenen düz, nitelenmemiş bir ad.
`setmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com +connect`	Adlandırılmış bir öznitelik kullanılıyor.
`setmqaut -m QM -t qmgr -p "phone=1234567" +connect`	AUTHINFO nesnesinde yapılandırılmış olması gerekmeyen başka bir adlandırılmış öznitelik kullanılıyor.

SET AUTHREC MQSC komutunu **setmqaut** komutuna alternatif olarak kullanabilirsiniz:

```
SET AUTHREC OBJTYPE(QMGR) PRINCIPAL('JohnDoe1@yourcompany.com') AUTHADD(connect)
```

ya da dizgiyi içeren MQCACF_PRINCIPAL_ENTITY_NAMES öğesini içeren Set Authority Record (MQCMD_SET_AUTH_REC) PCF komutu:

```
 "cn=JohnDoe,ou=users,o=yourcompany,c=yourcountry"
```

Grupları işlerken, grup adının herhangi bir biçimini 12 karakterlere sığdırmaya gerek olmadığı için, `shortname` işlemesiyle ilgili belirsizlik yoktur. Bu nedenle, gruplar için SHORTUSR özniteliğinin eşdeğeri yoktur.

Bu, Çizelge 76 sayfa 428 içinde açıklanan sözdizimi örneklerinin geçerli olduğu anlamına gelir; AUTHINFO nesnesini genişletilmiş özniteliklerle yapılandırdığınızı varsayarak ve şu değere ayarladığınızı varsayarak:

```
GRPFIELD(longname)
BASEDNG(ou=groups,o=yourcompany,c=yourcountry ) CLASSGRP(groupOfNames)
```

Çizelge 76. Grup yetkilendirme ayarları

Komut	**Not**
`setmqaut -m QM -t qmgr -g ApplicationGroupA +connect`	Çözümlemek için GRPFIELD kullanılması
`setmqaut -m QM -t qmgr -g longname=ApplicationGroupA +connect`	Tek bir özniteliği adlandırma
`setmqaut -m QM -t qmgr -g "cn=Application Group A,ou=groups,o=yourcompany,c=yourcountry" +connect`	Tam DN ' yi kullanma

SET AUTHREC MQSC komutunu, önceki **setmqaut** komutuna alternatif olarak kullanabilirsiniz:

```
SET AUTHREC OBJTYPE(QMGR) GROUP('ApplicationGroupA')
      AUTHADD(connect)
```

ya da Set Authority Record (MQCMD_SET_AUTH_REC) PCF komutunu şu dizgiyi içeren MQCACF_GROUP_ENTITY_NAMES öğesiyle birlikte kullanın:

```
  "ApplicationGroupA"
```

Önemli:

İster kullanıcı ister grup olsun, bir ada başvurmak için hangi biçimi kullanırsanız kullanın, benzersiz bir DN türetilebilir.

Örneğin, her ikisinde de "`shortu=jodoe`" olan iki ayrı kayıt olmamalıdır.

Tek bir benzersiz DN belirlenemezse, OAM, MQRC_UNKNOWN_ENTITY değerini döndürür.

Multi

Yetkilerin görüntülenmesi

Kullanıcıların ya da grupların yetkilendirmesini görüntülemenin çeşitli yöntemleri.

dspmqaut komutu

Bir kullanıcı ya da grup için kullanılabilecek yetkileri görüntülemek için en basit yöntem, dspmqaut komutunu kullanmaktır.

Bir kullanıcıyı ya da grubu tanımlamak için sözdizimi çeşitlemelerinde sorgu kullanabilirsiniz. Komut çıkışının, kimliği komut satırında belirtilen biçimde yinelediğini unutmayın. Çıktı, tam çözümlenmiş DN ' yi raporlamıyor.

Örneğin:

```
dspmqaut -m QM -t qmgr -p johndoe
```

```
Entity johndoe has the following authorizations for object QM:
    connect
```

veya

```
dspmqaut -m QM -t qmgr -p email=JohnDoe1@yourcompany.com
Entity email=JohnDoe1@yourcompany.com has the following authorizations for object QM:
    connect
```

dmpmqaut ve dmpmqcfg komutları

dmpmqaut komutu ve MQSC ya da PCF eşdeğerleri, birincil kullanıcıyı ya da grubu desteklenen biçimlerin herhangi birinde (“Ayar yetkileri” sayfa 427içinde açıklanan **setmqaut** çizelgeleri gibi) belirtebilir. Ancak, **dspmqaut**'in tersine, **dmpmqaut** komutu her zaman tam DN' yi bildirir.

```
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
```

Benzer şekilde, seçilen kayıtlarda süzme işlemi olmayan dmpmqcfg komutu, tam DN ' yi her zaman daha sonra yeniden yürütülebilir bir biçimde gösterir.

```
dmpmqcfg -m QM -x authrec
-----------------------------------
SET AUTHREC PROFILE(SELF) +
    PRINCIPAL('cn=JohnDoe, ou=users, o=yourcompany, c=yourcountry') +
    OBJTYPE(QMGR)
    AUTHADD(CONNECT)
```

Multi LDAP yetkilendirmesi kullanılırken dikkat edilmesi gereken diğer noktalar

IBM MQ 9.0.0' den LDAP yetkilendirmesini kullanırken bilmeniz gereken İleti Kuyruğu Arabirimi (MQI) ve diğer MQSC ve PCF komutlarında yapılan değişikliklerin kısa bir açıklaması.

ADOPTCTX

Uygulamaların kimlik doğrulama bilgilerini sağlaması ya da ADOPTCTX özniteliğinin YESdeğerine ayarlanması gerekli değildir.

Bir uygulama belirtik olarak kimlik doğrulaması yapmazsa ya da **ADOPTCTX** , etkin CONNAUTH nesnesi için NO değerine ayarlanırsa, uygulamayla ilişkili kimlik bağlamı işletim sistemi kullanıcı kimliğinden alınır.

Yetkilendirmelerin uygulanması gerekiyorsa, bu bağlam setmqaut komutlarıyla aynı kurallar kullanılarak bir LDAP kimliğiyle eşlenir.

MQI çağrılarına ilişkin giriş değiştirgeleri

MQOPEN, MQPUT1ve MQSUB , alternatif bir kullanıcı kimliğinin belirtilmesine izin veren yapılara sahip.

Bu alanlar kullanılırsa, 12 karakterlik kullanıcı kimliği, **setmqaut**, **dmpmqaut**ve **dspmqaut** komutlarıyla aynı kurallar kullanılarak bir DN ile eşlenir.

MQPUT ve MQPUT1 , uygun yetkili programların MQMD UserIdentifier alanını ayarlamasına da izin verir. Bu alanın değeri PUT işlemi sırasında uygulanmaz ve herhangi bir değere ayarlanabilir.

Ancak her zamanki gibi, **UserIdentifier** değeri ileti işlemenin sonraki aşamalarında (örneğin, PUTAUT (CTX) alıcı kanalda tanımlandığında) yetkilendirme için kullanılabilir.

Bu noktada, kimlik, LDAP ya da OS-tabanlı olabilen bu alan kuyruk yöneticisinin yapılanışı kullanılarak yetki denetimi yapılacaktır.

MQI çağrılarına ilişkin çıkış değiştirgeleri

MQI yapısındaki bir programa kullanıcı kimliği sağlandığı her yerde, bağlantıyla ilişkili 12 karakterlik kısa ad sürümüdür.

Örneğin, API Çıkışları için **MQAXC.UserId** değeri, LDAP eşlemesinden döndürülen kısa addır.

Diğer denetim MQSC ve PCF komutları

DISPLAY CONN USERID gibi nesne durumundaki kullanıcı bilgilerini gösteren komutlar, bağlamla ilişkili 12 karakterlik kısa adı döndürür. Tam DN gösterilmez.

Kanallar için CHLAUTH eşleme kuralları ya da MCAUSER değerleri gibi kimliklerin değerlendirmesine izin veren komutlar, bu öznitelikler için tanımlanan uzunluk üst sınırına (şu anda 64 karakter) kadar değer alabilir.

Sözdiziminde değişiklik yok. Bu kimlik için yetki gerekli olduğunda, **setmqaut**, **dmpmqaut**ve **dspmqaut** komutlarıyla aynı kurallar kullanılarak iç olarak bir DN ile eşlenir.

Bu, bir kanal tanımındaki MCAUSER değerinin DISPLAY CHSTATUS ile aynı dizgiyle görüntülenmeyebileceği, ancak aynı kimliğe başvurduğu anlamına gelir.

Örneğin:

```
DEFINE CHL(SV1) CHLTYPE(SVRCONN) MCAUSER('cn=JohnDoe')
DEFINE CHL(SV2) CHLTYPE(SVRCONN) MCAUSER('jodoe')
DEFINE CHL(SV3) CHLTYPE(SVRCONN) MCAUSER('JohnDoe1@yourcompany.com')
```

DISPLAY CHSTATUS (*) ALL, tüm bağlantılar için *MCAUSER (jodoe)* değerini gösterir.

Multi İşletim sistemi ve LDAP yetkilendirme modelleri arasında geçiş

Farklı platformlarda farklı yetkilendirme yöntemleri arasında geçiş yapma.

Kuyruk yöneticisinin CONNAUTH özniteliği bir AUTHINFO nesnesini gösteriyor. Nesne IDPWLDAP tipindeyse, kimlik doğrulaması için bir LDAP havuzu kullanılır.

Artık aynı nesneye bir yetkilendirme yöntemi uygulayabilirsiniz; bu yöntem, işletim sistemi tabanlı yetkilendirmeye devam etmenizi ya da LDAP yetkilendirmesiyle çalışmanızı sağlar.

IBM i, AIX and Linux

Linux IBM i AIX

Kuyruk yöneticisi, işletim sistemi ve LDAP modelleri arasında herhangi bir zamanda değiştirilebiliyor. REFRESH SECURITY TYPE (CONNAUTH) komutunu kullanarak yapılandırmayı değiştirebilir ve yapılandırmayı etkin hale getirebilirsiniz.

Örneğin, bu nesne kimlik doğrulaması için bağlantı bilgileriyle önceden yapılandırıldıysa:

```
ALTER AUTHINFO(MYLDAP) AUTHTYPE(IDPWLDAP) +
        AUTHORMD(SEARCHGRP) +
        BASEDNG('ou=groups,o=ibm,c=uk') +
        <other attributes>
ALTER QMGR CONNAUTH(MYLDAP)
REFRESH SECURITY
```

Windows

Windows

Bir yetki yapılanışı değişikliği, işletim sistemi ve LDAP modelleri arasında geçiş yapılmasını içeriyorsa, değişikliğin yürürlüğe girmesi için kuyruk yöneticisinin yeniden başlatılması gerekir. Tersi durumda, REFRESH SECURITY TYPE (CONNAUTH) komutunu kullanarak değişikliği etkinleştirebilirsiniz.

İşleme kuralları

İşletim sisteminden LDAP yetkilendirmesine geçilirken, ayarlanan var olan işletim sistemi yetki kuralları devre dışı ve görünmez olur.

dmpmqaut gibi komutlar bu işletim sistemi kurallarını görüntülemez. Benzer şekilde, LDAP ' tan işletim sistemine geri dönerken, tanımlanan LDAP yetkileri devre dışı ve görünmez hale gelir ve özgün işletim sistemi kurallarını geri yükler.

dmpmqcfg komutunu kullanarak herhangi bir nedenle bir kuyruk yöneticisi tanımlamalarını yedeklemek istiyorsanız, bu yedekleme işlemi yalnızca, yedekleme sırasında yürürlükte olan yetkilendirme yöntemi için tanımlanan kuralları içerir.

Multi LDAP yönetimi

Her platformun LDAP ' ı nasıl idare ettiğine ilişkin bir genel bakış.

LDAP yetkilendirmesini kullanırken, işletim sistemindeki mqm grubunun (ya da eşdeğerinin) üyeliği o kadar da önemli değildir. Bu grubun üyesi olmak yalnızca belirli komut satırı komutlarının işlenip işlenemeyeceğini denetler.

Özellikle, strmqm ve endmqm komutlarını yayınlamak için o grupta olmanız gerekir.

Kuyruk yöneticisi çalıştıktan sonra, artık tam ayrıcalıklı hesapta sınırlar vardır. **strmqm** komutunu veren kişinin kullanıcı kimliğinin yanı sıra, işletim sistemi mqm (ya da eşdeğeri) grubuna ait diğer kullanıcılar özel ayrıcalıklar almazlar.

Diğer kullanıcıların yetkileri, ait oldukları LDAP gruplarını temel alır. **setmqaut** gibi komutlarda mqm grup adının nitelenmemiş kullanımının herhangi bir LDAP grubuyla eşlenmesine izin verilmez.

AIX and Linux

Linux AIX

Kuyruk yöneticisi çalıştıktan sonra, otomatik olarak tam ayrıcalıklı olan tek hesap, kuyruk yöneticisini başlatan gerçek kullanıcıdır.

Kuyruk yöneticisinin altında çalıştığı etkin kimlik mqm olduğundan, mqm tanıtıcısı hala var ve dosyalar gibi işletim sistemi kaynaklarının sahibi olarak kullanılıyor. Ancak, mqm kullanıcısı OAM tarafından denetlenen yönetim görevlerini otomatik olarak gerçekleştiremez.

Windows

Windows

Windowsişletim sisteminde, otomatik olarak tam ayrıcalıklı hesaplar, kuyruk yöneticisini başlatan işletim sistemi kullanıcısı ve ayrıca, kuyruk yöneticisi bir Windows hizmeti olarak başlatıldıysa, MUSR_MQADMIN gibi temel kuyruk yöneticisi işlemlerini çalıştıran kullanıcıdır.

LDAP yetkilendirme kipinde çalışırken Windows , AIX and Linux platformlarına çok benzer şekilde davranır. 12 karakterlik kısa ad ve tam DN ile ilgilenir.

IBM i

IBM i

IBM i' da, otomatik olarak ayrıcalıklı hesaplar kuyruk yöneticisini ve QMQM tanıtıcısını başlatan hesaplardır.

Kuyruk yöneticisini başlatan kullanıcı kimliği yalnızca sistemi başlatmak için gerekli olduğundan, her iki kimliğe de ihtiyacınız vardır. Çalıştırıldığında, kuyruk yöneticisi işlemleri yalnızca QMQM yetkisine sahip olur.

MQADMIN ayrıcalıkları sağlamak için örnek komut dosyası

Linux AIX

Bir grubun bir kuyruk yöneticisinde tam denetim yapabilmesi yararlı olduğundan, AIX and Linux altyapılarında aşağıdaki gibi örnek bir komut dosyası gönderilir:

```
MQ_INSTALLATION_PATH/samp/bin/amqauthg.sh
```

Bu örnek iki parametre alır:

- Kuyruk yöneticisi adı
- LDAP grubu adı

Örnek, tüm nesneler için tam yetki veren setmqaut komutlarını işler. Bu komut dosyası, yönetim rolleri için IBM MQ Explorer OAM Sihirbazı tarafından oluşturulan komut dosyasıyla aynıdır. Örneğin, kod şu şekilde başlar:

```
setmqaut -t q -m qmgr -n "**" +alladm -g
      groupname
```

İletilerin gizliliği

İletilerin şifrelenmesi, iletilerin içeriğinin gizli kalmasını sağlar. İletileri, gereksinimlerinize bağlı olarak IBM MQ içinde şifrelemenin çeşitli yöntemleri vardır.

Noktadan noktaya ileti sistemi altyapınız için uygulama düzeyinde, uçtan uca veri korumasına gereksinim duyarsanız, iletileri şifrelemek için kullanabilirsiniz Advanced Message Security ya da kendi API çıkış ya da API geçiş çıkışınızı yazabilirsiniz.

En güvenli çözüm, bir iletiyi uygulama tarafından konduğu noktadan, kullanan uygulamanın elde ettiği noktaya kadar şifreleyerek uçtan uca şifreleme sağlamaktır. Bu, “Advanced Message Security planlaması” sayfa 107 (AMS) kullanılarak ya da kendi API çıkışınızı ya da API geçiş çıkışınızı yazarak yapılabilir; daha fazla bilgi için bkz. “Kullanıcı çıkış programlarında gizlilik uygulanması” sayfa 480 .

İletileri yalnızca bir ağ üzerinden taşınırken şifrelemeniz gerekiyorsa, TLS ' yi kullanabilirsiniz; daha fazla bilgi için bkz. “IBM MQ içinde TLS güvenlik iletişim kuralları” sayfa 24 ya da şifreleme gerçekleştirmek için kendi güvenlik çıkışınızı, ileti çıkışınızı yazabilir ya da çıkış programlarını gönderip alabilirsiniz.

z/OS Bir kuyruk yöneticisinde atıl durumdaki iletileri şifrelemeniz gerekiyorsa, o kuyruk yöneticisinde z/OS veri kümesi şifrelemesini kullanabilirsiniz; ek bilgi için bkz. “Veri kümesi şifrelemesiyle IBM MQ for z/OS üzerinde atıl durumdaki veriler için gizlilik” sayfa 481 .

İlgili görevler

TLS kullanarak iki kuyruk yöneticisinin bağlanması

İstemcinin kuyruk yöneticisine güvenli bir şekilde bağlanması

CipherSpecs Özelliğinin Etkinleştirilmesi

DEFINE CHANNEL ya da **ALTER CHANNEL** MQSC komutunda **SSLCIPH** değiştirgesini kullanarak CipherSpec ' i etkinleştirin.

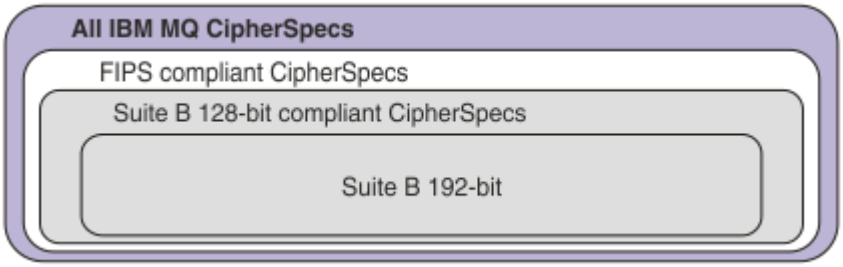
Not: AIX, Linux, and Windowsişletim sistemlerinde IBM MQ , IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, IBM Crypto for C (ICC) sertifikasını görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat

etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu İşlem listesindeki NIST CMVP modüllerindearananarak görüntülenebilir.

IBM MQ ile kullanabileceğiniz CipherSpecs ' in bazıları FIPS uyumludur. TLS_RSA_WITH_AES_256_CBC_SHAgibi diğerleri uyumlu olmasa da, FIPS uyumlu CipherSpecs ' in bazıları da Suite B uyumludur.

Tüm Suite B uyumlu CipherSpecs de FIPS uyumludur. Tüm Suite B uyumlu CipherSpecs iki gruba ayrılır: 128 bit (örneğin, ECDHE_ECDSA_AES_128_GCM_SHA256) ve 192 bit (örneğin, ECDHE_ECDSA_AES_256_GCM_SHA384),

Aşağıdaki çizge, bu altkümeler arasındaki ilişkiyi gösterir:



Ürün, IBM MQ 9.2.0' den tüm platformlarda TLS 1.3 güvenlik iletişim kuralını destekler.

Bu platformların her biri için kullanabileceğiniz CipherSpecs Çizelge 77 sayfa 434içinde listelenir. Bu CipherSpecs' in kullanılmasıyla ilgili bilgi için bkz. "IBM MQ içinde TLS 1.3 kullanılması" sayfa 436 ve "IBM MQ MQI client ve TLS 1.3" sayfa 437.

Yapılandırma ve gelecekteki geçiş kolaylığı için IBM MQ , bir diğer ad kümesi de sağlar CipherSpecs. Var olan güvenlik yapılandırmalarının CipherSpec diğer adını kullanacak şekilde geçirilmesi, gelecekte daha fazla saldırgan yapılandırma değişikliği yapmanıza gerek kalmadan şifreleme eklemelerine ve kullanımdan kaldırmalara uyum sağlayabileceğiniz anlamına gelir. Bu diğer ad CipherSpecs , Çizelge 77 sayfa 434içindeki CipherSpecs bölümünde listelenir. CipherSpecdiğer adını kullanmak üzere geçiş yapılmasına ilişkin ek bilgi için Var olan güvenlik yapılandırmalarının CipherSpecbaşlıklı konuya bakın.

"IBM MQ içinde etkinleştirilen varsayılan CipherSpec değerleri" sayfa 437içinde açıklandığı gibi varsayılan CipherSpecs ' i yapılandırabilirsiniz. Aşağıdaki kanallar üzerinde kullanılmak üzere etkinleştirilen alternatif bir CipherSpecs kümesi de sağlayabilirsiniz:

- Multi IBM MQ for Multiplatforms, "IBM MQ for Multiplatforms üzerinde sipariş edilen ve etkinleştirilen CipherSpecs ' in özel bir listesini sağlama" sayfa 446içinde açıklandığı gibi.
- z/OS IBM MQ for z/OS, "IBM MQ for z/OS üzerinde sipariş edilen ve etkinleştirilen CipherSpecs ' in özel bir listesini sağlama" sayfa 447içinde açıklandığı gibi.

Gerekirse IBM MQ ile kullanmak üzere yeniden etkinleştirebileceğiniz kullanımdan kaldırılmış CipherSpecs "Kullanımdan kaldırılan CipherSpecs" sayfa 448içinde listelenir. Kullanımdan kaldırılan CipherSpecs' in etkinleştirilmesiyle ilgili bilgi için bkz. "IBM MQ for Multiplatforms üzerinde kullanımdan kaldırılan CipherSpecs etkinleştiriliyor" sayfa 451 ya da "z/OS üzerinde kullanımdan kaldırılan CipherSpecs etkinleştiriliyor" sayfa 452.

IBM MQ TLS desteğiyle kullanabileceğiniz CipherSpecs

IBM MQ kuyruk yöneticisiyle otomatik olarak kullanabileceğiniz CipherSpecs (Şifre Belirtimleri) aşağıdaki çizelgede listelenir. Kişisel sertifika isteğinde bulunduğunuzda, genel ve özel anahtar çifti için bir anahtar boyutu belirtirsiniz. TLS anlaşması sırasında kullanılan anahtar boyutu, tabloda belirtildiği şekilde CipherSpectarafından belirlenmedikçe sertifikada saklanan boyuttur.

Çizelge 77. CipherSpecs IBM MQ TLS desteğiyle birlikte kullanabilirsiniz.

Platform desteği "1" sayfa 436	CipherSpec adı	Onaltılık kod	Kullanılan protokol	MAC algoritması	Şifreleme algoritması (şifreleme bitleri)	FIPS "2" sayfa 436	Takım B
Diğer Ad CipherSpecs							
Tümü	ANY_TLS13_OR_HIGHER "3" sayfa 436 "4" sayfa 436	Yok	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler
Tümü	ANY_TLS13 "4" sayfa 436 "5" sayfa 436	Yok	TLS 1.3	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler
Tümü	ANY_TLS12_OR_HIGHER "4" sayfa 436 "6" sayfa 436	Yok	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler
Tümü	ANY_TLS12 "7" sayfa 436	Yok	TLS 1.2	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler
Tümü	ANY "8" sayfa 436	Yok	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler	Anlaşılanet Görüşmeler
CipherSpecs for TLS 1.3							
Tümü	TLS_AES_128_GCM_SHA256	1301	TLS 1.3	GCM	AES-128 (GCM ile) (128)	Evet	Hayır
Tümü	TLS_AES_256_GCM_SHA384	1302	TLS 1.3	GCM	AES-256 (GCM ile) (256)	Evet	Hayır
Tümü	TLS_CHACHA20_POLY1305_SHA256	1303	TLS 1.3	POLY1305	CHACHA20 (256)	Hayır	Hayır
ALW	TLS_AES_128_CCM_SHA256	1304	TLS 1.3	CBC-MAC	AES-128 (CTR ile) (128)	Evet	Hayır
ALW	TLS_AES_128_CCM_8_SHA256 "10" sayfa 436	1305	TLS 1.3	CBC-MAC	AES-128 (CTR ile) (128)	Evet	Hayır
CipherSpecs for TLS 1.2							

Çizelge 77. CipherSpecs IBM MQ TLS desteğiyle birlikte kullanabilirsiniz. (devamı var)

Platform desteği “1” sayfa 436	**CipherSpec adı**	**Onaltılık kod**	**Kullanılan protokol**	**MAC algoritması**	**Şifreleme algoritması (şifreleme bitleri)**	**FIPS** “2” sayfa 436	**Takım B**
Tümü	TLS_RSA_WITH_AES_128_CBC_SHA256 “9” sayfa 436	003C	TLS 1.2	SHA-256	AES (128)	Evet	Hayır
Tümü	TLS_RSA_WITH_AES_256_CBC_SHA256 “9” sayfa 436 “11” sayfa 436	003D	TLS 1.2	SHA-256	AES (256)	Evet	Hayır
Tümü	TLS_RSA_WITH_AES_128_GCM_SHA256 “9” sayfa 436 “12” sayfa 436	009C	TLS 1.2	SHA-256 ve AEAD GCM	AES (128)	Evet	Hayır
Tümü	TLS_RSA_WITH_AES_256_GCM_SHA384 “9” sayfa 436 “11” sayfa 436 “12” sayfa 436	009D	TLS 1.2	SHA-384 ve AEAD GCM	AES (256)	Evet	Hayır
Tümü	ECDHE_ECDSA_AES_128_CBC_SHA256 “9” sayfa 436	C023	TLS 1.2	SHA-256	AES (128)	Evet	Hayır
Tümü	ECDHE_ECDSA_AES_256_CBC_SHA384 “9” sayfa 436 “11” sayfa 436	C024	TLS 1.2	SHA-384	AES (256)	Evet	Hayır
Tümü	ECDHE_RSA_AES_128_CBC_SHA256 “9” sayfa 436	C027	TLS 1.2	SHA-256	AES (128)	Evet	Hayır
Tümü	ECDHE_RSA_AES_256_CBC_SHA384 “9” sayfa 436 “11” sayfa 436	C028	TLS 1.2	SHA-384	AES (256)	Evet	Hayır
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256 “11” sayfa 436 “12” sayfa 436	C02B	TLS 1.2	SHA-256 ve AEAD GCM	AES (SHA384)	Evet	128 bit
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384 “11” sayfa 436 “12” sayfa 436	C02C	TLS 1.2	SHA-384 ve AEAD GCM	AES (SHA384)	Evet	192 bit
Tümü	ECDHE_RSA_AES_128_GCM_SHA256 “12” sayfa 436	C02F	TLS 1.2	SHA-256 ve AEAD GCM	AES (128)	Evet	Hayır
Tümü	ECDHE_RSA_AES_256_GCM_SHA384 “11” sayfa 436 “12” sayfa 436	C030	TLS 1.2	AEAD AES-128 GCM	AES (SHA384)	Evet	Hayır

Çizelge 77. CipherSpecs IBM MQ TLS desteğiyle birlikte kullanabilirsiniz. (devamı var)

Platform desteği "1" sayfa 436	CipherSpec adı	Onaltılı kod	Kullanılan protokol	MAC algoritması	Şifreleme algoritması (şifreleme bitleri)	FIPS "2" sayfa 436	Takım B

Notlar:

1. Her platform simgesinin kapsadığı platformların bir listesi için Ürün belgelerinde kullanılan simgelerbaşlıklı konuya bakın.
2. CipherSpec ' in FIPS onaylı bir platformda FIPS onaylı olup olmadığını belirtir. FIPS ' ye ilişkin açıklamalar için bkz. Federal Information Processing Standards (FIPS) .
3. ALW ANY_TLS13_OR_HIGHER diğer adı CipherSpec , uzak ucun izin vereceği en yüksek güvenlik düzeyini kararlaştırır, ancak yalnızca TLS 1.3 ya da daha yüksek bir iletişim kuralı kullanılarak bağlanır.
4. IBM i TLS 1.3ya da ANY CipherSpec'i kullanmak için IBM i üzerindeki temel işletim sistemi sürümü TLS 1.3' ü desteklemelidir. Daha fazla bilgi için TLSv1.3için Sistem TLS desteği başlıklı konuya bakın.
5. ALW ANY_TLS13 diğer adı CipherSpec , her platform için bu tabloda listelendiği şekilde TLS 1.3 iletişim kuralını kullanan kabul edilebilir CipherSpecs alt kümesini temsil eder.
6. ALW ANY_TLS12_OR_HIGHER diğer adı CipherSpec , uzak ucun izin vereceği en yüksek güvenlik düzeyini kararlaştırır, ancak yalnızca TLS 1.2 ya da daha yüksek bir iletişim kuralı kullanılarak bağlanır.
7. ANY_TLS12 CipherSpec , her platform için bu çizelgede listelendiği şekilde TLS 1.2 protokolünü kullanan kabul edilebilir CipherSpecs alt kümesini temsil eder.
8. ALW ANY diğer adı CipherSpec , uzak ucun izin vereceği en yüksek güvenlik düzeyini kararlaştırır.
9. IBM i Bu CipherSpecs , QSSLCSLCTL Sistem Değeri *OPSSYS olarak ayarlanmış IBM i 7.4 sistemlerinde etkinleştirilmez.
10. ALW Bu CipherSpecs , 16 oktet ICV yerine 8 oktet Bütünlük Denetimi Değeri (ICV) kullanır.
11. Uygun kısıtlanmamış ilke dosyaları Explorer tarafından kullanılan JRE ' ye uygulanmadıkça, IBM MQ Explorer ile bir kuyruk yöneticisine bağlantı korumak için bu CipherSpec kullanılamaz.
12. ALW GSKittarafından yapılan bir öneriyi takiben TLS 1.2 GCM CipherSpecs ' I ileti ile sınırlama vardır; bu kısıtlama ve üzerinden eleri geçtikten sonra24.5 TLS kayıtları aynı oturum anahtarı kullanılarak gönderildikten sonra bağlantının AMQ9288Eiletisiyle sonlandırıldığı anlamına gelir. Bu GCM kısıtlaması, kullanılmakta olan FIPS kipinden bağımsız olarak etkindir.

    Bu hatanın oluşmasını önlemek için TLS 1.2 GCM şifrelemeleri kullanmaktan kaçının, gizli anahtar sıfırlamasını etkinleştirin ya da IBM MQ kuyruk yöneticinizi ya da istemcinizi `GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE` ortam değişkeniyle başlatın. GSKit kitaplıkları için, bu ortam değişkenini bağlantının her iki tarafına da ayarlamalı ve hem kuyruk yöneticisi bağlantıları için istemciye hem de kuyruk yöneticisi bağlantıları için kuyruk yöneticisine uygulamalısınız. Bu ayarın yönetilmeyen .NET istemcilerini etkilediğini, ancak Java ya da yönetilen .NET istemcilerini etkilemediğini unutmayın. Daha fazla bilgi için bkz. AES-GCM şifre sınırlaması.

    Bu kısıtlama IBM MQ for z/OSiçin geçerli değildir.

IBM MQ içinde TLS 1.3 kullanılması

Ürün, IBM MQ 9.2.0'den tüm platformlarda TLS 1.3 ' ü destekler. IBM MQ 9.2.0'den önce, TLS 1.3 desteği, IBM MQ 9.1.4' den Continuous Delivery için AIX, Linux, and Windows ' de mevcuttu.

IBM MQ 9.2.0 ya da daha sonra oluşturulan kuyruk yöneticileri varsayılan olarak TLS 1.3 ' u destekler. IBM MQ 'un önceki sürümlerinden geçirilen kuyruk yöneticilerinin TLS 1.3 ' e sahip olması

gerekir. **AllowTLSV13**=TRUE özelliğini ayarlayarak, geçirilen kuyruk yöneticilerindeki TLS 1.3 ' ü etkinleştirebilirsiniz:

- Multi IBM MQ for Multiplatforms kuyruk yöneticileri için qm.ini dosyasını düzenleyin ve SSL kısmı altına **AllowTLSV13**=TRUE özelliğini ekleyin (bağlantı

```
SSL:
  AllowTLSV13=TRUE
```

- z/OS IBM MQ for z/OS kuyruk yöneticileri için, kuyruk yöneticisi başlatma JCL ' de belirtilen QMINI veri kümesini düzenleyin ve TransportSecurity kısmı altına **AllowTLSV13**=TRUE özelliğini ekleyin.

```
TransportSecurity:
    AllowTLSV13=TRUE
```

TLS 1.3 etkinleştirildiğinde ve TLS 1.3 belirtimineuygun olarak, IBM MQ içinde etkinleştirilip etkinleştirilmediğine bakılmaksızın zayıf bir CipherSpecile iletişim kurma girişimleri reddedilir. TLS 1.3 'in zayıf olarak gördüğü CipherSpecs , aşağıdaki ölçütlerden birini ya da daha fazlasını karşılayan CipherSpecs ' dir:

- SSL 3.0 iletişim kuralını kullanır.
- Şifreleme algoritması olarak RC4 ya da RC2 kullanır.
- 112 'ye eşit ya da 112 'den küçük bir şifreleme anahtarı boyutu (bit) vardır.

Bu kısıtlamalar, Kullanımdan kaldırılan CipherSpecsçizelgesinin 1 numaralı çizelgesinde Not [3] ile işaretlenir.

Bu CipherSpecsözelliğini kullanmaya devam etmeniz gerekiyorsa, TLS 1.3 kipini devre dışı bırakmanız gerekir:

- ALW Kuyruk yöneticisinin qm.ini dosyasını düzenleyin ve **AllowTLSV13** özelliğinin ayarını aşağıdaki gibi değiştirin:

```
SSL:
  AllowTLSV13=FALSE
```

- z/OS Kuyruk yöneticisinin QMINI veri kümesini düzenleyin ve **AllowTLSV13** özelliğinin ayarını aşağıdaki şekilde değiştirin:

```
TransportSecurity:
    AllowTLSV13=FALSE
```

IBM MQ MQI client ve TLS 1.3

ALW

IBM MQ MQI clientkullanılırken, uygulama tarafından kullanılan mqclient.ini dosyasının SSL kısmına belirtik olarak belirtilmedikçe **AllowTLSV13** değeri çıkarılır.

- Zayıf CipherSpecs etkinleştirildiyse, **AllowTLSV13** FALSE olarak ayarlanır ve TLS 1.3 CipherSpecs kullanılamaz.
- Tersi durumda, **AllowTLSV13** TRUE olarak ayarlanır ve yeni TLS 1.3 CipherSpecs ve diğer ad CipherSpecs kullanılabilir.

IBM MQ içinde etkinleştirilen varsayılan CipherSpec değerleri

Yeni bir IBM MQ kuyruk yöneticisi için varsayılan yapılandırmada IBM MQ , CipherSpecskullanarak TLS 1.2 ve TLS 1.3 iletişim kuralları ve çeşitli şifreleme algoritmaları için destek sağlar. Uyumluluk amacıyla IBM MQ , SSL 3.0 ve TLS 1.0 iletişim kurallarını ve zayıf ya da güvenlik açıklarına duyarlı olduğu bilinen bir

dizi şifreleme algoritmasını kullanacak şekilde yapılandırılabilir. Varsayılan yapılandırmada etkinleştirilen CipherSpecs listesi, bakım uygulanarak değişebilir.

IBM MQ , aşağıdaki denetimleri kullanarak CipherSpecs kullanımını kısıtlayacak ya da izin verecek şekilde yapılandırılabilir:

- Yalnızca SSLFIPS kullanarak FIPS 140-2 uyumlu CipherSpecs ' e izin verin.
- ALW SUITEB kullanarak yalnızca NSA Suite B uyumlu CipherSpecs ' e izin verin.
- Multi **AllowedCipherSpecs**kullanarak özel bir CipherSpecs listesine izin verin.
- ALW **AMQ_ALLOWED_CIPHERS** ortam değişkenini kullanarak özel bir CipherSpecs listesine izin verin.
- ALW **AllowWeakCipher** ya da **AMQ_SSL_WEAK_CIPHER_ENABLE** ortam değişkenini kullanarak kullanımdan kaldırılan CipherSpecs kullanımına izin verin.
- z/OS CHINIT JCL ' de DD deyimleri kullanarak kullanımdan kaldırılmış CipherSpecs kullanımına izin verin.

Not: AllowedCipherSpecs kullanarak özel bir CipherSpecs listesi belirtirseniz ya da **AMQ_ALLOWED_CIPHERS** bu, kullanımdan kaldırılan CipherSpecs' in etkinleştirilmesini geçersiz kılar. Özel bir CipherSpec listesiyle birlikte NSA Suite B ya da FIPS 140-2 kısıtlamaları kullanırken, özel listenin yalnızca Suite B ya da FIPS 140-2 ayarlarının izin verdiği CipherSpecs öğesini içerdiğinden emin olmanız gerektiğini unutmayın.

İlgili kavramlar

“IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu” sayfa 46
Bu konuda, IBM MQiçinde CipherSpecs ile dijital sertifikalar arasındaki ilişkiyi özetleyerek güvenlik ilkeniz için uygun CipherSpecs ve dijital sertifikaların nasıl seçileceğine ilişkin bilgiler sağlanır.

“CipherSpecs ve CipherSuites” sayfa 21
Kriptografik güvenlik protokolleri, güvenli bir bağlantı tarafından kullanılan algoritmalar üzerinde anlaşmalıdır. CipherSpecs ve CipherSuites algoritmaların belirli birleşimlerini tanımlar.

“IBM MQ ürününü Suite B için yapılandırma” sayfa 43
IBM MQ , AIX, Linux, and Windows platformlarında NSA Suite B standardına uygun olarak çalışacak şekilde yapılandırılabilir.

“Federal Bilgi İşleme Standartları (FIPS)” sayfa 33
Bu konuda, ABD Ulusal Standartlar ve Teknoloji Enstitüsü 'nün Federal Bilgi İşleme Standartları (FIPS) Şifreleme Modülü Doğrulama Programı ve TLS kanallarında kullanılabilen şifreleme işlevleri tanıtılmaktadır.

İlgili görevler

Var olan güvenlik yapılandırmalarının CipherSpe diğer adını kullanacak şekilde geçirilmesi

İlgili başvurular

KANAL TANıMLAYıN

KANAL DEğIşTIR

Kanalı Değiştir, Kopyala ve Oluştur

ALW AES-GCM şifre kısıtlaması

TLS Şifreleme için kullanıldığında AES-GCM şifrelemelerine uygulanan kısıtlamalara ilişkin bir kılavuz. Bu kısıtlamalar IETF ve NIST kuruluşları tarafından uygulanır ve AES-GCM şifrelemeleri kullanılırken aynı oturum anahtarının 2 'den fazla$^{24.5}$ TLS kaydını güvenli bir şekilde aktarmak için kullanılmamasını gerektirir.

Bu kısıtlamalarla ilgili daha fazla bilgi için bkz. RFC 9325 Section 4.4 Limits on Key Usage ve RFC 8446 section 5.5.

IBM MQ , doğrudan şifreleme işlevselliğini uygulamaz. Bunun yerine, TLS ve Advanced Message Security işlevselliğini sağlamak için birkaç farklı şifreleme kitaplığı kullanılır. Windows, Linuxve AIX işletim sistemlerinde, IBM MQ 'in kullandığı şifreleme kitaplığı IBM Global Security Kit (GSKit)' dir. Uygulamalar için, C ve yönetilmeyen .NET kitaplıkları şifreleme işlevi için GSKit kullanır. AES-GCM şifreleme algoritmalarının GSKit tarafından uygulanması, standartlar grubu tarafından belirtilen kısıtlamaları içerir. Ayrıca, bu kısıtlamalar varsayılan olarak etkinleştirilir. Bu nedenle IBM MQ TLS iletişimi, AES-GCM şifrelerini kullanırken, aynı oturum anahtarı kullanılarak 2 'den fazla$^{24.5}$ TLS kaydı iletildiğinde sona erer.

Not: Farklı şifreleme kitaplıkları kullanıldığından ve bu kitaplıklar aynı kısıtlamayı uygulamadığından, IBM i, IBM Z ya da IBM MQ for HPE NonStop platformları ya da Java/JMS, yönetilen .NET uygulamalarda bu kısıtlama yoktur.

Bir IBM MQ kanalı, 2 'den fazla$^{24.5}$ TLS kaydı aynı oturum anahtarı kullanılarak iletilecek kadar uzun süre çalışır durumda kalırsa, temeldeki şifreleme kitaplığı bağlantıyı sonlandırır. Bu, kanalın sonlandırılmasına ve bir AMQ9288E hata iletisinin oluşturulmasına neden olur. İletişimi bu şekilde sonlandırılan uygulamalar, gerçekleştirilmekte olan IBM MQ işleminden bir `MQRC_CONNECTION_BROKEN` dönüş kodu alır.

Bağlantının sonlandırılması iletişimin her iki ucunda da gerçekleştirilebilir, ancak yalnızca şifreleme işlevi için GSKit kullanan uçlarda gerçekleştirilebilir.

Kısıtlamanın hafifletilmesine ilişkin öneriler

Bu sınırlama nedeniyle sonlandırılan iletişimin nasıl önleneceğine ya da işleneceğine ilişkin bazı seçenekler şunlardır:

Yeniden bağlanabilir istemcileri kullan

Bir bağlantı başarısız olursa, uygulamaların konfigürasyonu otomatik olarak yeniden bağlanma girişiminde bulunacak şekilde tanımlanabilir. Bu, GCM kısıtlaması nedeniyle sonlandırılan bağlantıları içerir. Yeniden bağlantı için yapılandırıldığında, istemci uygulaması herhangi bir hata noktasında otomatik olarak geri yüklenir ve nesneleri açma tanıtıcıları geri yüklenir. Bu, uygulama koduna geri dönülmeden yapılır.

Daha fazla bilgi için bkz. Otomatik istemci yeniden bağlantısı.

Gizli anahtar sıfırlama değeri ayarla

IBM MQ , bir kanal üzerinden yapılandırılabilir bayt sayısı aktarıldıktan sonra oturum anahtarını ilk durumuna getirme isteğinde bulunacak şekilde yapılandırılabilir. Bu sınıra ulaştıktan sonra IBM MQ , şifreleme katmanının oturum anahtarını ilk durumuna getirmesini ister ve yeni bir oturum anahtarıyla sonuçlanır.

Belirtilen değerin, IBM MQtarafından gönderilen iletilerin boyutuyla ilgili olarak aktarılan bayt sayısı olduğunu unutmayın. Kısıtlama, gönderilen TLS kaydı sayısı üzerindedir. TLS kaydı, ağın İletim Birimi Üst Sınırı 'na (MTU) bağlı bayt sayısı üst sınırını gönderebileceği için, ileti baytları ile TLS kayıtları arasında doğrudan eşleme yoktur. Bu değerden büyük olan iletiler birden çok TLS kaydı olarak iletilir. MTU değeri ağlar arasında değişir. Ayrıca, TLS kaydının IBM MQ ileti verilerini iletme dışında gönderilmesi gerekmesinin başka nedenleri de vardır; örneğin, IBM MQ Heartbeat denetimleri, TLS uyarıları, diğer IBM MQ iletişim kuralı iletileri. Bu ek TLS kayıtları, TLS kaydı sayısı üst sınırına doğru sayılır, ancak IBM MQ gizli anahtar sıfırlama değerinde sayılmaz.

Gizli anahtar sıfırlaması kullanılarak bir oturum anahtarının düzenli olarak sıfırlanması, AES-GCM kısıtlaması nedeniyle kanalın sonlandırılmasını önleyebilir.

Daha fazla bilgi için SSL ve TLS gizli anahtarlarını sıfırlamabaşlıklı konuya bakın.

TLS 1.3 şifreleme belirtimlerini kullan

TLS 1.3 iletişim kuralı kullanılırken AES-GCM kısıtlaması hala varken TLS 1.3 iletişim kuralı, TLS iletişimini kesme gereksinimi olmadan otomatik olarak bir oturum anahtarı sıfırlama işleminin gerçekleştirilmesini destekler. Bu, GSKit 'in gerektiğinde oturum anahtarını sıfırlamayı, IBM MQ ' un gizli anahtar sıfırlaması istemesine gerek kalmadan yönetmesini sağlar.

Daha fazla bilgi için bkz. “CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432içinde IBM MQ içinde TLS 1.3 kullanma .

AES-GCM kısıtlamasını devre dışı bırak

Gerekirse, **GSK_ENFORCE_GCM_RESTRICTION=GSK_FALSE** ortam değişkeni AES-GCM kısıtlamasını devre dışı bırakacak şekilde ayarlanarak kısıtlama devre dışı bırakılabilir. Bunu yapmak, aynı oturum anahtarı kullanılarak herhangi bir sayıda TLS kaydının gönderilmesine olanak sağlar. Bu azaltma seçiliyorsa, ortam değişkeni, güvenli iletişim için GSKit kullanan iletişimin her bir ucunda ayarlanmalıdır.

Uyarı: Bu seçenek, 2 'den fazla$^{24.5}$ TLS kaydı gönderildikten sonra, saldırganların kullanılmakta olan oturum anahtarını belirlemek için gönderilen kayıtlar üzerinde analiz gerçekleştirmeleri mümkün olduğu için önerilmez. Oturum anahtarı belirlendikten sonra, bu oturum anahtarını kullanarak var olan ve gelecekteki tüm iletişim tehlikeye atılır.

TLS el sıkışmasında CipherSpec sırası

CipherSpecs sırası, birden çok olası CipherSpecsseçilirken kullanılır; örneğin, ANY* CipherSpecskullanılıyorsa.

TLS anlaşması sırasında bir istemci ve sunucu, CipherSpecs ve destekledikleri protokolleri tercihlerine göre değiştirir. Her iki tarafın da önceliklerini belirlediği ortak bir CipherSpec seçilir ve TLS iletişimi için kullanılır. CipherSpec iletişim kuralı seçilirken, örneğin bir sunucu TLS 1.2 CipherSpecs 'i TLS 1.3 CipherSpecs ' ten önce listelerse, istemci destekleyebildiği ve kullanılabilecek ortak bir TLS 1.3 CipherSpec 'e sahip olduğu sürece TLS 1.3 ' e öncelik tanımaya devam eder.

IBM MQ 9.2.0' den TLS için IBM MQ yapılandırıldığında, CipherSpecs öğesini en çok tercih edilen öğeden en az tercih edilen öğeye kadar aşağıdaki tabloda gösterilen sıraya ayarlar.

Not: Bir CipherSpec **AllowedCipherSpecs** özniteliği aracılığıyla etkinleştirilmezse, TLS anlaşması sırasında kullanılmak üzere yapılandırılmayacaktır.

AllowedCipherSpecs özniteliği belirtilmezse, aşağıdaki çizelgeyle gösterilen varsayılan bir etkin şifrelemeler listesi kullanılır.

Çizelge 78. CipherSpecs sırası: IBM MQ 9.2.0

Hizmet olarak sunulan	CipherSpec	Protokol	Onaltılı kod	Varsayılan olarak etkindir
Tümü	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Evet
Tümü	TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Evet
Tümü	TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Evet
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	1304	Evet
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	1305	Evet
Tümü	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Evet
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	C02C	Evet
Tümü	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Evet

Çizelge 78. CipherSpecs sırası: IBM MQ 9.2.0 (devamı var)				
Hizmet olarak sunulan	**CipherSpec**	**Protokol**	**Onaltılı kod**	**Varsayılan olarak etkindir**
Tümü	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Evet
Tümü	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Evet
Tümü	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Evet
Tümü	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Evet
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	C02B	Evet
Tümü	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Evet
Tümü	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Evet
Tümü	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Evet
Tümü	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Evet
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	C008	Hayır
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	C012	Hayır
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	0005	Hayır
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	C007	Hayır
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	C011	Hayır
Tümü	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Hayır
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	C006	Hayır
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	C010	Hayır

Çizelge 78. CipherSpecs sırası: IBM MQ 9.2.0 (devamı var)				
Hizmet olarak sunulan	**CipherSpec**	**Protokol**	**Onaltılı kod**	**Varsayılan olarak etkindir**
ALW	TLS_RSA_WITH_NULL_NULL (boş değerli)	TLS 1.2	0000	Hayır
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Hayır
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Hayır
IBM i	AES_SHA_US	TLS 1.0	002E	Hayır
Tümü	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Hayır
Tümü	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Hayır
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	0004	Hayır
Tümü	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Hayır
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	0003	Hayır
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	0006	Hayır
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	0002	Hayır
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	0001	Hayır
Tümü	TRIPLE_DES_SHA_US	SSL v3	000A	Hayır
Tümü	RC4_SHA_US	SSL v3	0005	Hayır
Tümü	RC4_MD5_US	SSL v3	0004	Hayır
Tümü	DES_SHA_EXPORT (DışA AKTARMA)	SSL v3	0009	Hayır
Tümü	RC4_MD5_EXPORT	SSL v3	0003	Hayır
Tümü	RC2_MD5_EXPORT	SSL v3	0006	Hayır
Tümü	NULL_SHA	SSL v3	0002	Hayır
Tümü	NULL_MD5	SSL v3	0001	Hayır

Çizelge 78. CipherSpecs sırası: IBM MQ 9.2.0 (devamı var)				
Hizmet olarak sunulan	**CipherSpec**	**Protokol**	**Onaltılı kod**	**Varsayılan olarak etkindir**
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL v3	FEFF	Hayır
ALW	RC4_56_SHA_EXPORT1024	SSL v3	0064	Hayır
ALW	DES_SHA_EXPORT1024	SSL v3	0062	Hayır
ALW	FIPS_WITH_DES_CBC_SHA	SSL v3	FEFE	Hayır

Bu liste, z/OS üzerinde IBM MQ tarafından kullanılan şifreleme kitaplığı tarafından sağlanan varsayılan listeyle birlikte iletişim kuralları sipariş edilerek oluşturulmuştur ve z/OS ve dağıtılmış platformlar arasında tutarlıdır.

sırayı değiştirme

Farklı bir sipariş isteniyorsa, aşağıdaki kurallarla IBM MQ for Multiplatforms z/OS, ya da IBM MQ for z/OSüzerindeki TransportSecurity kısmı, üzerinde SSL ' nin **AllowedCipherSpecs** özniteliği kullanılarak yeni bir CipherSpecs sırası sağlanabilir:

- Listedeki konumlarından bağımsız olarak, daha yüksek protokol sürümleri her zaman kullanılır.
- Listede belirtildiyse, geçersiz kılınmış CipherSpecs yeniden etkinleştirilir.
- TLS sunucusunun liste sırası, TLS istemcisinden daha yüksek önceliğe sahip.
- TLS 1.3 etkinleştirildiğinde, bazı CipherSpecs desteklenmez.

Örneğin, IBM MQ for Multiplatformsüzerinde, kuyruk yöneticisinde aşağıdakiler yapılandırılırsa:

```
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 ve IBM MQ for z/OSüzerinde, kuyruk yöneticisinde aşağıdakiler yapılandırıldıysa:

```
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
```

daha sonra:

- ANY_TLS12 ile bağlanan bir istemci büyük olasılıkla TLS 1.2 CipherSpec TLS_RSA_WITH_AES_128_GCM_SHA256kullanacak.
- ANY_TLS12_OR_HIGHER ile bağlanan bir istemci büyük olasılıkla TLS 1.3 CipherSpec TLS_AES_128_GCM_SHA256 kullanacak (istemcinin TLS 1.3' ü desteklediği varsayılarak).
- TLS 1.0 CipherSpec TLS_RSA_WITH_AES_256_CBC_SHA ile bağlanan bir istemci bu CipherSpec' i kullanır.

Önceki IBM MQ sürümleri

IBM MQ 9.2.0öncesinde aşağıdaki CipherSpecs sırası kullanılıyordu:

Çizelge 79. CipherSpecs sipariş öncesi IBM MQ 9.2.0			
Hizmet olarak sunulan	**CipherSpec**	**Protokol**	**Varsayılan olarak etkindir**
ALW z/OS	TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	Hayır
IBM i	AES_SHA_US	TLS 1.0	Hayır
ALW z/OS	TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	Hayır
Tümü	RC4_SHA_US	SSL v3	Hayır
Tümü	TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	Hayır
Tümü	RC4_MD5_US	SSL v3	Hayır
IBM i	TLS_RSA_WITH_RC4_128_MD5	TLS 1.0	Hayır
Tümü	TRIPLE_DES_SHA_US	SSL v3	Hayır
Tümü	TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	Hayır
ALW	DES_SHA_EXPORT1024	SSL v3	Hayır
Tümü	RC4_56_SHA_EXPORT1024	SSL v3	Hayır
Tümü	RC4_MD5_EXPORT	SSL v3	Hayır
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5	TLS 1.0	Hayır
Tümü	RC2_MD5_EXPORT	SSL v3	Hayır
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5	TLS 1.0	Hayır
Tümü	DES_SHA_EXPORT (DışAAKTARMA)	SSL v3	Hayır
Tümü	TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	Hayır
Tümü	NULL_SHA	SSL v3	Hayır
IBM i	TLS_RSA_WITH_NULL_SHA	TLS 1.0	Hayır
Tümü	NULL_MD5	SSL v3	Hayır
IBM i	TLS_RSA_WITH_NULL_MD5	TLS 1.0	Hayır
ALW	FIPS_WITH_DES_CBC_SHA	SSL v3	Hayır

Çizelge 79. CipherSpecs sipariş öncesi IBM MQ 9.2.0 (devamı var)			
Hizmet olarak sunulan	**CipherSpec**	**Protokol**	**Varsayılan olarak etkindir**
ALW	FIPS_WITH_3DES_EDE_CBC_SHA	SSL v3	Hayır
Tümü	TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	Evet
Tümü	TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	Evet
Tümü	TLS_RSA_WITH_NULL_SHA256	TLS 1.2	Hayır
Tümü	TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	Evet
Tümü	TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	Evet
ALW	ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	Hayır
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	Hayır
Multi	ECDHE_RSA_RC4_128_SHA256	TLS 1.2	Hayır
Multi	ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	Hayır
Tümü	ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	Evet
Tümü	ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	Evet
Tümü	ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	Evet
Tümü	ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	Evet
Multi	ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	Evet
Multi	ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	Evet
Tümü	ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	Evet
Tümü	ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	Evet
Multi	ECDHE_RSA_NULL_SHA256	TLS 1.2	Hayır
ALW	ECDHE_ECDSA_NULL_SHA256	TLS 1.2	Hayır

Çizelge 79. CipherSpecs sipariş öncesi IBM MQ 9.2.0 (devamı var)			
Hizmet olarak sunulan	**CipherSpec**	**Protokol**	**Varsayılan olarak etkindir**
ALW	TLS_RSA_WITH_NULL_NULL (boş değerli)	TLS 1.2	Hayır
ALW	TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	Hayır
Multi	TLS_AES_128_GCM_SHA256	TLS 1.3	Evet
Multi	TLS_AES_256_GCM_SHA384	TLS 1.3	Evet
Multi	TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	Evet
ALW	TLS_AES_128_CCM_SHA256	TLS 1.3	Evet
ALW	TLS_AES_128_CCM_8_SHA256	TLS 1.3	Evet

Önemli: 23rd Temmuz 2020 itibariyle, aşağıdaki AllowedCipherSpecs özniteliği yalnızca varsayılan olarak etkinleştirilmiş CipherSpecs özelliğini etkinleştirir. Ancak, bu tarihten bu yana kullanımdan kaldırılan CipherSpecs öğesinin yanlışlıkla yeniden etkinleştirilmediğinden emin olmak için, aşağıdaki AllowedCipherSpecs özniteliğiyle etkinleştirilen CipherSpecs öğesini doğrulamanız gerekir.

Bu CipherSpecssiparişine geri dönmeniz gerekirse, aşağıdaki **AllowedCipherSpecs** SSL/TransportSecurity stanza öznitelik değerini kullanarak bunu yapabilirsiniz:

```
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
```

IBM MQ for Multiplatforms üzerinde sipariş edilen ve etkinleştirilen CipherSpecs ' in özel bir listesini sağlama

Multi

ALW **AMQ_ALLOWED_CIPHERS** ortam değişkeni ' ni ya da `.ini` dosyasının **AllowedCipherSpecs** SSL kısmı özniteliğini kullanarak IBM MQ kanallarıyla kullanmak için, etkinleştirilen alternatif bir CipherSpecs kümesi ve tercih sıranıza göre sağlayabilirsiniz. Aşağıdaki nedenlerden biri için bu ayarı kullanmak isteyebilirsiniz:

- IBM MQ dinleyicilerinin CipherSpecsadlı öğelerden birini kullanmadıkları sürece, gelen kanal başlatma isteklerini kabul etmelerini engellemek için.
- TLS el sıkışmasında kullanılan CipherSpecs öncelik sırasını değiştirmek için.

Bu işlev, ANY* CipherSpecsiçinde bulunan CipherSpecs öğesini denetlemek için kullanılabilir.

AMQ_ALLOWED_CIPHERS ortam değişkeni ya da **AllowedCipherSpecs** SSL kısmı özniteliği aşağıdakileri kabul eder:

- Tek bir CipherSpec adı.
- Yeniden etkinleştirilecek CipherSpec adlarının virgülle ayrılmış listesi.
- Tüm CipherSpecs' i gösteren ALLözel değeri.

Not: SSL 3.0 ve TLS 1.0 iletişim kurallarını ve çok sayıda zayıf şifreleme algoritmasını etkinleştireceği için **ALL** CipherSpecsözelliğini etkinleştirmemelisiniz.

Bu ayar yapılandırılırsa, varsayılan CipherSpec listesini geçersiz kılar ve IBM MQ ' in zayıf şifreleme kullanımdan kaldırma ayarlarını yoksaymasına neden olur (aşağıya bakın):

- IBM MQ dinleyiciler yalnızca CipherSpecsadlı belirtimlerden birini kullanan SSL/TLS önerilerini kabul eder.
- IBM MQ kanalları yalnızca boş bir SSLCIPH değerine ya da CipherSpecsadlı değerlerden birine izin verir.
- **runmqsc** SSLCIPH değerlerinin sekme ile tamamlanması, tamamlanma değerlerini CipherSpecsadından biriyle sınırlar.

Örneğin, yalnızca kanalların tanımlanmasına/değiştirilmesine ve dinleyicilerin ECDHE_RSA_AES_128_GCM_SHA256 ya da ECDHE_ECDSA_AES_256_GCM_SHA384 kabul etmesine izin vermek istiyorsanız, `qm.ini` dosyasında aşağıdakileri ayarlayabilirsiniz:

```
SSL:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256, ECDHE_ECDSA_AES_256_GCM_SHA384
```

Buna ek olarak, TLS anlaşması sırasında kullanılan CipherSpecs ' in önceliğini belirlemek için bu listedeki CipherSpecs kullanılır. Örneğin, bir `TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256` listesi belirtirseniz, tokalaşma sırasında, bir istemci bunların her ikisini de CipherSpecs(yani ANY_TLS12ile bağlantı kuran bir istemci) belirterek bağlantı kurarsa, TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec üzerinden TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec seçilir.

AMQP ya da MQTT kanalları tarafından kullanılan şifrelemelerin java.security dosya ayarları kullanılarak kısıtlanabileceğini unutmayın.

IBM MQ for z/OS üzerinde sipariş edilen ve etkinleştirilen CipherSpecs ' in özel bir listesini sağlama

z/OS

QMINI veri kümesinin **AllowedCipherSpecs** TransportSecurity stanza özniteliğini kullanarak IBM MQ kanallarıyla kullanmak için, etkinleştirilen ve sizin tercih sıranıza göre, alternatif bir CipherSpecs kümesi sağlayabilirsiniz. Bunu aşağıdakilerden biri nedeniyle yapmak isteyebilirsiniz:

- IBM MQ dinleyicilerinin CipherSpecsadlı öğelerden birini kullanmadıkları sürece, gelen kanal başlatma isteklerini kabul etmelerini engellemek için.
- TLS el sıkışmasında kullanılan CipherSpecs öncelik sırasını değiştirmek için.

Bu işlevi, ANY* CipherSpecsiçinde bulunan CipherSpecs öğesini denetlemek için kullanabilirsiniz. **AllowedCipherSpecs** özniteliği aşağıdakileri kabul eder:

- Tek bir CipherSpec adı.
- Yeniden etkinleştirilecek CipherSpec adlarının virgülle ayrılmış listesi.
- Tüm CipherSpecs' i gösteren `ALL`özel değeri.

  Not: SSL 3.0 ve TLS 1.0 iletişim kurallarını ve çok sayıda zayıf şifreleme algoritmasını etkinleştireceği için **ALL** CipherSpecsözelliğini etkinleştirmemelisiniz. Bu ayarı yapılandırırsanız, varsayılan CipherSpec listesini geçersiz kılar ve IBM MQ ' in zayıf şifre kullanımdan kaldırma ayarlarını yoksaymasına neden olur; bkz. “z/OS üzerinde kullanımdan kaldırılan CipherSpecs etkinleştiriliyor” sayfa 452.

IBM MQ dinleyicileri yalnızca, adı belirtilen CipherSpecs ve IBM MQ kanallarından birini kullanan SSL/TLS önerilerini kabul eder ve yalnızca boş bir SSLCIPH değerine ya da adı belirtilen CipherSpecsdeğerlerinden birine izin verir.

Örneğin, yalnızca kanalların tanımlanmasına/değiştirilmesine izin vermek istiyorsanız ve dinleyicilerin ECDHE_RSA_AES_128_GCM_SHA256 ya da ECDHE_RSA_AES_256_GCM_SHA384 kabul etmesini istiyorsanız aşağıdakileri ayarlayabilirsiniz:

```
TransportSecurity:
  AllowedCipherSpecs=ECDHE_RSA_AES_128_GCM_SHA256,
                     ECDHE_RSA_AES_256_GCM_SHA384
```

Buna ek olarak, TLS anlaşması sırasında kullanılan CipherSpecs önceliğini belirlemek için bu listedeki CipherSpecs kullanılır. For example, if you specify a list of TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA256 it is likely that, during the handshake, the TLS_RSA_WITH_AES_128_CBC_SHA256 CipherSpec will be chosen over the TLS_RSA_WITH_AES_256_CBC_SHA256 CipherSpec if a client connects specifying both of these CipherSpecs, that is, a client connecting with ANY_TLS12.

Deprecated Kullanımdan kaldırılan CipherSpecs

Gerekirse IBM MQ ile birlikte kullanabileceğiniz kullanımdan kaldırılmış CipherSpecs listesi.

Not: AIX, Linux, and Windowsişletim sistemlerinde IBM MQ , IBM Crypto for C (ICC) şifreleme modülü aracılığıyla FIPS 140-2 uyumluluğu sağlar. Bu modüle ilişkin sertifika Geçmiş durumuna taşındı. Müşteriler, IBM Crypto for C (ICC) sertifikasını görüntüleyip NIST tarafından sağlanan tüm önerilere dikkat etmelidir. Yeni bir FIPS 140-3 modülü şu anda devam ediyor ve durumu İşlem listesindeki NIST CMVP modüllerindearanarak görüntülenebilir.

Kullanımdan kaldırılan CipherSpecsözelliğinin etkinleştirilmesiyle ilgili bilgi için bkz. “IBM MQ for Multiplatforms üzerinde kullanımdan kaldırılan CipherSpecs etkinleştiriliyor” sayfa 451 ya da “z/OS üzerinde kullanımdan kaldırılan CipherSpecs etkinleştiriliyor” sayfa 452.

IBM MQ TLS desteğiyle birlikte kullanabileceğiniz kullanımdan kaldırılmış CipherSpecs aşağıdaki tabloda listelenmiştir.

Çizelge 80. IBM MQ ile kullanmak üzere yeniden etkinleştirebileceğiniz kullanımdan kaldırılmış CipherSpecs

Platform desteği “1” sayfa 451	CipherSpec adı	Onaltılı kod	Kullanılan protokol	Veri bütünlüğü	Şifreleme algoritması (şifreleme bitleri)	FIPS “2” sayfa 451	Takım B	Kullanımdan kaldırıldığında güncelle
CipherSpecs for SSL 3.0								
IBM i	AES_SHA_US “3” sayfa 451	002F	SSL 3.0	SHA-1	AES (128)	Hayır	Hayır	9.0.0.0
Tümü	DES_SHA_EXPORT “3” sayfa 451 “4” sayfa 451 “5” sayfa 451	0009	SSL 3.0	SHA-1	DES (56)	Hayır	Hayır	9.0.0.0
ALW	DES_SHA_EXPORT1024 “3” sayfa 451 “6” sayfa 451	0062	SSL 3.0	SHA-1	DES (56)	Hayır	Hayır	9.0.0.0
ALW	FIPS_WITH_DES_CBC_SHA “3” sayfa 451	FEFE	SSL 3.0	SHA-1	DES (56)	Hayır “7” sayfa 451	Hayır	9.0.0.0
ALW	FIPS_WITH_3DES_EDE_CBC_SHA “3” sayfa 451	AYdaki ka ve,	SSL 3.0	SHA-1	3DES (168)	Hayır “8” sayfa 451	Hayır	9.0.0.1 ve 9.0.1
Tümü	NULL_MD5 “3” sayfa 451	0001	SSL 3.0	MD5	Yok	Hayır	Hayır	9.0.0.1
Tümü	NULL_SHA “3” sayfa 451	0002	SSL 3.0	SHA-1	Yok	Hayır	Hayır	9.0.0.1

Çizelge 80. IBM MQ ile kullanmak üzere yeniden etkinleştirebileceğiniz kullanımdan kaldırılmış CipherSpecs (devamı var)

Platform desteği "1" sayfa 451	CipherSpec adı	Onaltılı kod	Kullanılan protokol	Veri bütünlüğü	Şifreleme algoritması (şifreleme bitleri)	FIPS "2" sayfa 451	Takım B	Kullanımdan kaldırıldığında güncelle
Tümü	RC2_MD5_EXPORT "3" sayfa 451 "4" sayfa 451 "5" sayfa 451	0006	SSL 3.0	MD5	RC2 (40)	Hayır	Hayır	9.0.0.0
Tümü	RC4_MD5_EXPORT "4" sayfa 451 "3" sayfa 451	0003	SSL 3.0	MD5	RC4 (40)	Hayır	Hayır	9.0.0.0
Tümü	RC4_MD5_US "3" sayfa 451	0004	SSL 3.0	MD5	RC4 (128)	Hayır	Hayır	9.0.0.0
Tümü	RC4_SHA_US "3" sayfa 451 "5" sayfa 451	0005	SSL 3.0	SHA-1	RC4 (128)	Hayır	Hayır	9.0.0.0
ALW	RC4_56_SHA_EXPORT1024 "3" sayfa 451 "6" sayfa 451	0064	SSL 3.0	SHA-1	RC4 (56)	Hayır	Hayır	9.0.0.0
Tümü	TRIPLE_DES_SHA_US "3" sayfa 451 "5" sayfa 451	000A	SSL 3.0	SHA-1	3DES (168)	Hayır	Hayır	9.0.0.1 ve 9.0.1
CipherSpecs for TLS 1.0								
IBM i	TLS_RSA_EXPORT_WITH_RC2_40_MD5 "3" sayfa 451	0006	TLS 1.0	MD5	RC2 (40)	Hayır	Hayır	9.0.0.0
IBM i	TLS_RSA_EXPORT_WITH_RC4_40_MD5 "3" sayfa 451 "4" sayfa 451	0003	TLS 1.0	MD5	RC4 (40)	Hayır	Hayır	9.0.0.0
Tümü	TLS_RSA_WITH_DES_CBC_SHA "3" sayfa 451	0009	TLS 1.0	SHA-1	DES (56)	Hayır "9" sayfa 451	Hayır	9.0.0.0
IBM i	TLS_RSA_WITH_NULL_MD5 "3" sayfa 451	0001	TLS 1.0	MD5	Yok	Hayır	Hayır	9.0.0.1
IBM i	TLS_RSA_WITH_NULL_SHA "3" sayfa 451	0002	TLS 1.0	SHA-1	Yok	Hayır	Hayır	9.0.0.1
IBM i	TLS_RSA_WITH_RC4_128_MD5 "3" sayfa 451	0004	TLS 1.0	MD5	RC4 (128)	Hayır	Hayır	9.0.0.0
z/OS ALW	TLS_RSA_WITH_AES_128_CBC_SHA "10" sayfa 451	002F	TLS 1.0	SHA-1	AES (128)	Evet	Hayır	9.0.5
z/OS ALW	TLS_RSA_WITH_AES_256_CBC_SHA "6" sayfa 451 "10" sayfa 451	0035	TLS 1.0	SHA-1	AES (256)	Evet	Hayır	9.0.5
Tümü	TLS_RSA_WITH_3DES_EDE_CBC_SHA	000A	TLS 1.0	SHA-1	3DES (168)	Evet	Hayır	9.0.0.1 ve 9.0.1
CipherSpecs for TLS 1.2								

Çizelge 80. IBM MQ ile kullanmak üzere yeniden etkinleştirebileceğiniz kullanımdan kaldırılmış CipherSpecs (devamı var)

Platform desteği "1" sayfa 451	CipherSpec adı	Onaltılı kod	Kullanılan protokol	Veri bütünlüğü	Şifreleme algoritması (şifreleme bitleri)	FIPS "2" sayfa 451	Takım B	Kullanımdan kaldırıldığında güncelle
ALW	ECDHE_ECDSA_NULL_SHA256 "3" sayfa 451	C006	TLS 1.2	SHA-1	Yok	Hayır	Hayır	9.0.0.1
ALW	ECDHE_ECDSA_RC4_128_SHA256 "3" sayfa 451	C007	TLS 1.2	SHA-1	RC4 (128)	Hayır	Hayır	9.0.0.0
ALW IBM i	ECDHE_RSA_NULL_SHA256 "3" sayfa 451	C010	TLS 1.2	SHA-1	Yok	Hayır	Hayır	9.0.0.1
ALW IBM i	ECDHE_RSA_RC4_128_SHA256 "3" sayfa 451	C011	TLS 1.2	SHA-1	RC4 (128)	Hayır	Hayır	9.0.0.0
ALW	TLS_RSA_WITH_NULL_NULL "3" sayfa 451	0000	TLS 1.2	Yok	Yok	Hayır	Hayır	9.0.0.1
Tümü	TLS_RSA_WITH_NULL_SHA256 "3" sayfa 451	003B	TLS 1.2	SHA-256	Yok	Hayır	Hayır	9.0.0.1
ALW	TLS_RSA_WITH_RC4_128_SHA256 "3" sayfa 451	0005	TLS 1.2	SHA-1	RC4 (128)	Hayır	Hayır	9.0.0.0
ALW	ECDHE_ECDSA_3DES_EDE_CBC_SHA256	C0008	TLS 1.2	SHA-1	3DES (168)	Evet	Hayır	9.0.0.1 ve 9.0.1
ALW IBM i	ECDHE_RSA_3DES_EDE_CBC_SHA256	C012	TLS 1.2	SHA-1	3DES (168)	Evet	Hayır	9.0.0.1 ve 9.0.1

Çizelge 80. IBM MQ ile kullanmak üzere yeniden etkinleştirebileceğiniz kullanımdan kaldırılmış CipherSpecs (devamı var)

Platform desteği "1" sayfa 451	CipherSpec adı	Onaltılı kod	Kullanılan protokol	Veri bütünlüğü	Şifreleme algoritması (şifreleme bitleri)	FIPS "2" sayfa 451	Takım B	Kullanımdan kaldırıldığında güncelle

Notlar:

1. Her platform simgesinin kapsadığı platformların bir listesi için Ürün belgelerinde kullanılan simgelerbaşlıklı konuya bakın.
2. CipherSpec ' in FIPS onaylı bir platformda FIPS onaylı olup olmadığını belirtir. FIPS ' ye ilişkin açıklamalar için bkz. Federal Information Processing Standards (FIPS) .
3. ALW Bu CipherSpecs , TLS 1.3 etkinleştirildiğinde (`qm.ini`içindeki AllowTLSV13 özelliği aracılığıyla) devre dışı bırakılır.

   z/OS IBM MQ for z/OS 9.2.0 ya da daha sonra oluşturulan kuyruk yöneticileri varsayılan olarak TLS 1.3 'u etkinleştirerek bu CipherSpecs' i devre dışı bırakır. Gerekirse, TLS V1.3'i kapatarak bu CipherSpecs' i etkinleştirebilirsiniz. Bu, kuyruk yöneticisi JCL ' deki QMINI veri kümesinin TransportSecurity kısmına **`AllowTLSV13`**=*FALSE* eklenerek yapılır. Daha önceki bir sürümden IBM MQ for z/OS 9.2.0 ' e geçirilen kuyruk yöneticilerinin varsayılan olarak TLS 1.3 özelliği etkinleştirilmez ve bu CipherSpecs özelliği etkinleştirilir.
4. El sıkışma anahtarı büyüklüğü üst sınırı 512 bittir. SSL anlaşması sırasında değiş tokuş edilen sertifikalardan birinin anahtar boyutu 512 bitten fazlaysa, el sıkışması sırasında kullanılmak üzere geçici bir 512 bitlik anahtar oluşturulur.
5. Bu CipherSpecs artık IBM MQ classes for Java ya da IBM MQ classes for JMStarafından desteklenmez. Daha fazla bilgi için bkz. IBM MQ classes for Java içinde SSL/TLS CipherSpecs ve CipherSuites ya da IBM MQ classes for JMS içinde SSL/TLS CipherSpecs ve CipherSuites.
6. El sıkışma anahtarı boyutu 1024 bittir.
7. Deprecated Bu CipherSpec , 19 Mayıs 2007 'den önce FIPS 140-2 sertifikalıydı. `FIPS_WITH_DES_CBC_SHA` adı geçmiştir ve bu CipherSpec ' in önceden (ancak artık) FIPS uyumlu olmadığı gerçeğini yansıtır. Bu CipherSpec kullanımdan kaldırılmıştır ve kullanılması önerilmez.
8. Deprecated `FIPS_WITH_3DES_EDE_CBC_SHA` adı geçmiştir ve bu CipherSpec ' in önceden (ancak artık) FIPS uyumlu olmadığı gerçeğini yansıtır. Bu CipherSpec kullanımı kullanımdan kaldırılmıştır.
9. Bu CipherSpec , 19 Mayıs 2007 'den önce FIPS 140-2 sertifikalıydı.
10. z/OS Yalnızca bu CipherSpecs belirtimleri yeniden etkinleştirilmek için CSQXWEAK DD deyimi kullanılması gerekmez.

IBM MQ for Multiplatforms üzerinde kullanımdan kaldırılan CipherSpecs etkinleştiriliyor

Multi

Varsayılan olarak, bir kanal tanımlamasında kullanımdan kaldırılmış bir CipherSpec belirtmenize izin verilmez. IBM MQ for Multiplatformsüzerinde kullanımdan kaldırılmış bir CipherSpec belirtmeyi denerseniz, AMQ8242iletisi alırsınız: SSLCIPH tanımı yanlış ve PCF `MQRCCF_SSL_CIPHER_SPEC_ERROR`değerini döndürür.

Bir kanalı kullanımdan kaldırılmış bir CipherSpecile başlatamazsınız. Bunu, kullanımdan kaldırılmış bir CipherSpecile yapmayı denerseniz, sistem istemciye `MQCC_FAILED (2)`öğesini **Reason** ile birlikte `MQRC_SSL_INITIALIZATION_ERROR (2393)` öğesini döndürür.

AMQ_SSL_WEAK_CIPHER_ENABLEortam değişkenini ayarlayarak, sunucuda yürütme sırasında kanalları tanımlamak için kullanımdan kaldırılan CipherSpecs değişkenlerinden birini ya da daha fazlasını yeniden etkinleştirebilirsiniz.

AMQ_SSL_WEAK_CIPHER_ENABLE ortam değişkeni aşağıdakileri kabul eder:

- Tek bir CipherSpec adı ya da
- Yeniden etkinleştirilecek CipherSpec adlarının virgülle ayrılmış listesi ya da
- Tüm CipherSpecs' i gösteren `ALL`özel değeri.

Uyarı: `ALL` geçerli bir seçenek olsa da, `ALL` CipherSpecs özelliğinin SSL 3.0 ve TLS 1.0 iletişim kurallarının yanı sıra çok sayıda zayıf şifreleme algoritması da etkinleştirildiğinden, bunu kuruluşunuzun gerektirdiği belirli bir durumda **yalnızca** kullanmanız gerekir.

Örneğin, ECDHE_RSA_RC4_128_SHA256değişkenini yeniden etkinleştirmek istiyorsanız aşağıdaki ortam değişkenini ayarlayın:

```
export AMQ_SSL_WEAK_CIPHER_ENABLE=ECDHE_RSA_RC4_128_SHA256
```

ya da aşağıdaki ayarları yaparak `qm.ini` dosyasındaki SSL kısmı değerini değiştirin:

```
SSL:
  AllowTLSV1=Y
  AllowWeakCipherSpec=ECDHE_RSA_RC4_128_SHA256
```

z/OS üzerinde kullanımdan kaldırılan CipherSpecs etkinleştiriliyor

z/OS

Varsayılan olarak, bir kanal tanımlamasında kullanımdan kaldırılmış bir CipherSpec belirtmenize izin verilmez. z/OSüzerinde kullanımdan kaldırılmış bir CipherSpec belirtmeyi denerseniz, CSQM102E, CSQX616Eya da CSQX674Eiletisi alırsınız.

Bu iletilerden herhangi birini alırsanız bu bölümde listelenen yönergeleri izleyin ve işletmenizin zayıf CipherSpecskullanımını yeniden etkinleştirmesi gerekir.

Uyarı: Aşağıdaki yönergelerde, kukla tanımlama (DD) deyimlerinin yürürlüğe girmesi için SSLTASKS sıfır dışında bir değer olmalıdır. Bu, SSLTASKS ' de bir değişiklik yapılmasını gerektiriyorsa, kanal başlatıcısını geri dönüştürmeniz gerekir.

IBM MQ for z/OSişletim sistemlerinde, zayıf ya da bozuk CipherSpecs ' i denetleme yöntemi aşağıdaki gibidir:

- Zayıf CipherSpecskullanımını yeniden etkinleştirmek istiyorsanız, bunu kanal başlatıcısı JCL ' ye CSQXWEAK adlı bir sahte veri tanımlaması (DD) deyimi ekleyerek yaparsınız. Tek başına belirtilirse, bu yalnızca TLS 1.2 iletişim kuralıyla ilişkili zayıf CipherSpecs ' i etkinleştirir; örneğin:

  ```
  //CSQXWEAK DD DUMMY
  ```

  Not: Kullanımdan kaldırılan tüm CipherSpecs , bu DD deyiminin kullanılmasını gerektirmez; önceki çizelgede not 10 'a bakın.
- SSLv3 CipherSpecskullanımını yeniden etkinleştirmek istiyorsanız, bunu kanal başlatıcı JCL ' ye CSQXSSL3 adlı bir kukla DD deyimi ekleyerek de yaparsınız. Tüm SSLv3 CipherSpecs **Zayıf**olarak değerlendirilir; bu nedenle CSQXWEAKbelirtmeniz gerekir:

  ```
  //CSQXSSL3 DD DUMMY
  ```

- Kullanımdan kaldırılan TLS V1 CipherSpecs'i yeniden etkinleştirmek istiyorsanız, kanal başlatıcı JCL' ye TLS10ON (TLS V1.0 ' ı AÇIK) adlı bir kukla DD deyimi ekleyerek bunu yaparsınız. Tek başına belirtilirse, TLS 1.0 protokolüyle ilişkilendirilmiş Strong CipherSpecs etkinleştirilir:

  ```
  //TLS10ON DD DUMMY
  ```

  CSQXWEAK ile birlikte belirtilirse bu, TLS 1.0ile ilişkilendirilmiş **Weak** CipherSpecs ' i de etkinleştirir.
- Kullanımdan kaldırılan TLS V1 CipherSpecs'i belirtik olarak kapatmak istiyorsanız, kanal başlatıcı JCL' ye TLS10OFF (TLS V1.0 ' ı kapat) adlı bir kukla DD deyimi ekleyerek bunu yaparsınız; örneğin:

  ```
  //TLS10OFF DD DUMMY
  ```

Yalnızca **System SSL** varsayılan şifre belirtimi listesinde listelenen şifre belirtimlerini kullanarak dinleyiciyle anlaşmak istiyorsanız, CHINIT JCL ' de aşağıdaki DD deyimini tanımlamanız gerekir:

```
JCL: //GSKDCIPS DD DUMMY
```

Önemli: IBM MQ for z/OS 9.2.0 ve sonraki yayın düzeylerinde, kanal başlatıcısı başlatılırken hangi iletişim kurallarının etkinleştirilip etkinleştirilmediğini belirtmek için iletiler görüntülenirken, daha önce listelenen DD kartları ve **AllowTLSV13** değeri dikkate alınır. Bu nedenle, daha önce listelenen DD kartlarından biri belirtilse bile, bu ayarların bir birleşimi nedeniyle, belirli bir protokolün başka bir protokolle etkinleştirilemeyeceği anlamına gelebilir. Örneğin, TLS 1.3 etkinleştirildiyse, SSL 3.0 protokolüne izin verilmez.

Veri Tanımı değişikliği uygun değilse, zayıf CipherSpecsve SSLv3 desteğini zorla yeniden etkinleştirmek için kullanılabilecek alternatif mekanizmalar vardır. Daha fazla bilgi için IBM Hizmet bölümüyle iletişim kurun.

İlgili kavramlar

“IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu” sayfa 46
Bu konuda, IBM MQiçinde CipherSpecs ile dijital sertifikalar arasındaki ilişkiyi özetleyerek güvenlik ilkeniz için uygun CipherSpecs ve dijital sertifikaların nasıl seçileceğine ilişkin bilgiler sağlanır.

İlgili başvurular

KANAL TANıMLAYıN

KANAL DEğIşTIR

Diğer ad CipherSpec ayarları arasındaki ilişki

Bu bilgiler, istemci ve sunucu yapılandırmalarında farklı CipherSpecs diğer adı birleşimleriyle beklenen davranışı açıklar. Burada istemci, bir istemci uygulaması ya da kuyruk yöneticisi gönderen kanalı gibi iletişimi başlatan varlığı ve sunucu, istemciden iletişimi alan varlığı (örneğin, bir sunucu bağlantısı kanalı ya da alıcı kanalı) belirtir.

Minimum iletişim kuralı ve sabit iletişim kuralı CipherSpecs

IBM MQ , iki farklı CipherSpecstipini destekler:

Protokol alt sınırı
: CipherSpecs iletişim kuralı alt sınırı, üst sınır belirlemeyenlerdir; örneğin, ANY, ANY_TLS12_OR_HIGHER ya da ANY_TLS13_OR_HIGHER.

Sabit iletişim kuralı
: Değişmez protokol CipherSpecs , belirli bir protokolü (örneğin, ANY_TLS12 ve ANY_TLS13) ya da ECDHE_ECDSA_3DES_EDE_CBC_SHA256gibi belirli bir algoritmayı tanıtan protokollerdir.

IBM MQ 9.2.0' den CipherSpecs alt sınır ve sabit iletişim kuralı tüm platformlarda desteklenir.

Güvenliği sağlarken yapılandırmanın basitliğini en üst düzeye çıkarmak için, kanalın her iki tarafında **minimum iletişim kuralı** CipherSpecs kullanılması önerilir. Bu, her iki taraf da her iki tarafın yapılandırmasını değiştirmeye gerek kalmadan yeni bir sürümü desteklediğinde iletişiminizin daha yüksek bir TLS iletişim kuralı sürümünü otomatik olarak desteklemesini ve kullanmasını sağlar.

Başlangıç tarafında **minimum iletişim kuralı** CipherSpec kullanılması, ancak alma tarafında **sabit iletişim kuralı** CipherSpec , bağlantının reddedilmesine neden olabilir ve

- Multi Yayınlanmakta olan iletiler AMQ9631 ve AMQ9641 .
- z/OS İletiler CSQX631E ve CSQX641E yayınlanıyor.

Aşağıdaki çizelgelerde, farklı diğer ad CipherSpec ayarları ile beklenen sonuç arasındaki ilişki gösterilir. Çizelge 81 sayfa 454 , TLS 1.3 istemci, sunucu ya da her ikisinde de etkinleştirilmediğinde beklenen davranışı gösterir. Çizelge 82 sayfa 454 , hem istemcide hem de sunucuda TLS 1.3 etkinleştirildiğinde beklenen davranışı gösterir. Her iki durumda da, istemciye ilişkin CipherSpecs çizelgenin Y ekseninde gösterilir ve sunucunun CipherSpecs değeri çizelgenin X ekseninde gösterilir.

Not: Aşağıdaki tablolarda, *Başarısız olma olasılığı* işaretli hücreler, bir bağlantının bir parçası için **minimum iletişim kuralı** CipherSpec ve başka bir parça için belirli bir (**sabit iletişim kuralı**) CipherSpec belirttiğinizde çakışma olasılığını gösterir.

Örneğin, istemci ve sunucunun ANY CipherSpeckullanacak şekilde ayarlandığı ve sunucu kanalının belirli bir CipherSpeckullanacak şekilde ayarlandığı varsayalım:

- Hem istemci hem de sunucu için desteklenen en güçlü CipherSpec , kanalda yapılandırılan belirli CipherSpec ile eşleşirse, TLS anlaşması başarıyla çözülür.
- Ancak, hem istemci hem de Sunucu desteği olan daha güçlü bir CipherSpec varsa, TLS el sıkışması, kanalda belirtilen CipherSpec ile eşleşmese de bunu kullanmaya çözülür ve TLS el sıkışması başarısız olur.

Çizelge 81. İstemcide, sunucuda ya da her ikisinde TLS 1.3 etkinleştirilmediğinde beklenen davranış

	Sunucu			
Müşteri	**Belirli TLS 1.2 CipherSpec**	**Fark Etmez**	**ANY_ TLS12**	**ANY_TLS12_ YA DA DAHA YüKSEK**
Belirli TLS 1.2 CipherSpec	Bağlanmalar	Bağlanmalar	Bağlanmalar	Bağlanmalar
Fark Etmez	*Başarısız olma olasılığı*	Bağlanmalar	Bağlanmalar	Bağlanmalar
ANY_ TLS12	*Başarısız olma olasılığı*	Bağlanmalar	Bağlanmalar	Bağlanmalar
ANY_TLS12_ YA DA DAHA YüKSEK	*Başarısız olma olasılığı*	Bağlanmalar	Bağlanmalar	Bağlanmalar

Çizelge 82. İstemcide ve sunucuda TLS 1.3 etkinleştirildiğinde beklenen davranış

	Sunucu						
Müşteri	**Belirli TLS 1.2 CipherSpec**	**Belirli TLS 1.3 CipherSpec**	**Fark Etmez**	**ANY_TLS 12**	**ANY_TLS 13**	**ANY_TLS12_ OR_YüKSEK**	**ANY_TLS13_ OR_YüKSEK**
Belirli TLS 1.2 CipherSpec	Bağlanmalar	**Başarısız**	Bağlanmalar	Bağlanmalar	**Başarısız**	Bağlanmalar	**Başarısız**
Belirli TLS 1.3 CipherSpec	**Başarısız**	Bağlanmalar	Bağlanmalar	**Başarısız**	Bağlanmalar	Bağlanmalar	Bağlanmalar
Fark Etmez	**Başarısız**	*Başarısız olma olasılığı*	Bağlanmalar	**Başarısız**	Bağlanmalar	Bağlanmalar	Bağlanmalar

Çizelge 82. İstemcide ve sunucuda TLS 1.3 etkinleştirildiğinde beklenen davranış (devamı var)							
	Sunucu						
Müşteri	**Belirli TLS 1.2 CipherSpec**	**Belirli TLS 1.3 CipherSpec**	**Fark Etmez**	**ANY_TLS 12**	**ANY_TLS 13**	**ANY_TLS12_ OR_YüKSEK**	**ANY_TLS13_ OR_YüKSEK**
ANY_TLS12	*Başarısız olma olasılığı*	**Başarısız**	Bağlanmalar	Bağlanmalar	**Başarısız**	Bağlanmalar	**Başarısız**
ANY_TLS13	**Başarısız**	*Başarısız olma olasılığı*	Bağlanmalar	**Başarısız**	Bağlanmalar	Bağlanmalar	Bağlanmalar
ANY_TLS12_ OR_üST	**Başarısız**	*Başarısız olma olasılığı*	Bağlanmalar	**Başarısız**	Bağlanmalar	Bağlanmalar	Bağlanmalar
ANY_TLS13_ OR_üST	**Başarısız**	*Başarısız olma olasılığı*	Bağlanmalar	**Başarısız**	Bağlanmalar	Bağlanmalar	Bağlanmalar

İlgili kavramlar

“IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu” sayfa 46
Bu konuda, IBM MQiçinde CipherSpecs ile dijital sertifikalar arasındaki ilişkiyi özetleyerek güvenlik ilkeniz için uygun CipherSpecs ve dijital sertifikaların nasıl seçileceğine ilişkin bilgiler sağlanır.

“CipherSpecs ve CipherSuites” sayfa 21
Kriptografik güvenlik protokolleri, güvenli bir bağlantı tarafından kullanılan algoritmalar üzerinde anlaşmalıdır. CipherSpecs ve CipherSuites algoritmaların belirli birleşimlerini tanımlar.

“CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432
DEFINE CHANNEL ya da **ALTER CHANNEL** MQSC komutunda **SSLCIPH** değiştirgesini kullanarak CipherSpec ' i etkinleştirin.

İlgili görevler

Var olan güvenlik yapılandırmalarının ANY_TLS12_OR_HIGHER CipherSpec ' i kullanacak şekilde geçirilmesi

IBM MQ Explorer kullanarak CipherSpecs hakkında bilgi alınması

CipherSpecstanımlamalarını görüntülemek için IBM MQ Explorer komutunu kullanabilirsiniz.

“CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432içindeki CipherSpecs ile ilgili bilgi edinmek için aşağıdaki yordamı kullanın:

1. IBM MQ Explorer dosyasını açın ve **Kuyruk Yöneticileri** klasörünü genişletin.
2. Kuyruk yöneticinizi başlattığınızdan emin olun.
3. Çalışmak istediğiniz kuyruk yöneticisini seçin ve **Kanalları**' nı tıklatın.
4. Çalışmak istediğiniz kanalı sağ tıklatın ve **Özellikler**seçeneğini belirleyin.
5. **SSL** özellik sayfasını seçin.
6. Çalışmak istediğiniz CipherSpec öğesini listeden seçin. Listenin altındaki pencerede bir tanımlama görüntülenir.

z/OS IBM i CipherSpecs belirtilerek ilgili alternatifler

İşletim sisteminin TLS desteğini sağladığı platformlarda, sisteminiz “CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432içinde bulunmayan yeni CipherSpecs özelliğini destekleyebilir.

SSLCIPH parametresiyle yeni bir CipherSpec belirtebilirsiniz, ancak sağladığınız değer altyapınıza bağlıdır. Her durumda belirtim, sisteminizin çalıştırdığı TLS sürümü tarafından hem geçerli hem de desteklenen bir TLS CipherSpec ' e karşılık gelmelidir.

Not: CipherSpecs IBM MQ ürünüyle birlikte sağlandığından bu bölüm AIX, Linux, and Windows sistemleri için geçerli değildir, bu nedenle yeni CipherSpecs teslimattan sonra kullanılamaz.

IBM i

Onaltılı bir değeri gösteren iki karakterli bir dizgi.

İzin verilen değerlerle ilgili daha fazla bilgi için, Güvenli oturum için karakter bilgileri ayarlanması' nin Kullanım Notları bölümündeki üçüncü noktaya bakın.

Uyarı: Hangi şifrenin kullanılacağı ve hangi protokolün kullanılacağı belirsiz olduğundan **SSLCIPH**içinde onaltılı şifre değerlerini belirtmemelisiniz. Onaltılı şifre değerlerinin kullanılması, CipherSpec yanlış eşleşme hatalarına neden olabilir.

Değeri belirtmek için **CHGMQMCHL** ya da **CRTMQMCHL** komutunu kullanabilirsiniz; örneğin:

```
CRTMQMCHL CHLNAME(' channel name ') SSLCIPH(' hexadecimal value ')
```

SSLCIPH değiştirgesini ayarlamak için **ALTER QMGR** MQSC komutunu da kullanabilirsiniz.

z/OS

Onaltılı bir değeri gösteren dört karakterli bir dizilim. Onaltılı kodlar, TLS iletişim kuralında tanımlanan değerlere karşılık gelir.

Daha fazla bilgi için, 4 basamaklı onaltılı kodlar biçiminde desteklenen tüm TLS 1.0, TLS 1.2ve TLS 1.3 şifre belirtimlerinin bir listesinin bulunduğu Cipher Suite Tanımları konusuna bakın.

Not: Deprecated SSL V3.0 ya da TLS 1.0gibi kullanımdan kaldırılmış bir protokole ait zayıf bir CipherSpecya da CipherSpec kullanmak için, kanal başlatıcı başlatma JCL ' de ilgili DD kartını belirtmeniz gerekir. Ek bilgi için bkz. “Kullanımdan kaldırılan CipherSpecs” sayfa 448 .

IBM MQ kümeleri için dikkat edilecek noktalar

IBM MQ kümeleriyle, “CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432içindeki CipherSpec adlarını kullanmak en güvenlidir. Alternatif bir belirtim kullanıyorsanız, belirtimin diğer platformlarda geçerli olmayabileceğini unutmayın. Daha fazla bilgi için bkz. “SSL/TLS ve kümeler” sayfa 494.

IBM MQ MQI client için CipherSpec belirtilmesi

IBM MQ MQI clientiçin CipherSpec belirtmek üzere üç seçeneğiniz vardır.

Bu seçenekler şunlardır:

- Kanal tanımlama çizelgesinin kullanılması
- MQCD yapısındaki SSLCipherSpec alanını kullanarak, bir MQCONNX çağrısında MQCD_VERSION_7 ya da üstü.
- Active Directory ' yi kullanma (Active Directory desteği olan Windows sistemlerinde)

IBM MQ classes for Java ve IBM MQ classes for JMS ile bir CipherSuite belirtme

IBM MQ classes for Java ve IBM MQ classes for JMS , CipherSuites ' i diğer platformlardan farklı olarak belirtin.

IBM MQ classes for Javaile bir CipherSuite belirtme hakkında bilgi için bkz. Transport Layer Security (TLS) support for Java

IBM MQ classes for JMSile bir CipherSuite belirtme hakkında bilgi için bkz. IBM MQ classes for JMSile Transport Layer Security (TLS) kullanma

IBM MQ.NET için CipherSpec belirtilmesi

IBM MQ.NET için, CipherSpec öğesini MQEnvironment sınıfını kullanarak ya da bağlantı özelliklerinin HASH çizelgesindeki MQC.SSL_CIPHER_SPEC_PROPERTY MQC.SSL_CIPHER_SPEC_PROPERTY öğesini kullanarak belirtebilirsiniz.

.NET yönetilmeyen istemci için CipherSpec belirtilmesiyle ilgili bilgi için Yönetilmeyen .NET istemci için TLS ' nin etkinleştirilmesi başlıklı konuya bakın.

.NET yönetilen istemcisi için CipherSpec belirtilmesiyle ilgili bilgi için CipherSpec support for the managed .NET client başlıklı konuya bakın.

z/OS IBM MQ for z/OS ile AT-TLS kullanımı

Application Transparent Transport Layer Security (AT-TLS), TLS desteğini uygulamak zorunda olmayan z/OS uygulamaları için TLS desteği sağlar ya da TLS ' nin kullanıldığından emin olun. AT-TLS yalnızca z/OSüzerinde kullanılabilir.

AT-TLS, IBM MQ for z/OS' nin tüm sürümleriyle kullanılabilir.

IBM MQ for z/OSile AT-TLS ' yi kullanmadan önce, ilgili “Sınırlamalar” sayfa 460 öğesini anladığınızdan emin olun.

Uygulama Şeffaf İletim Katmanı Güvenliği 'ni kullanmak için, hangi TCP/IP bağlantılarının TLS' nin saydam olarak etkinleştirildiğine karar vermek üzere z/OS Communications Server tarafından kullanılan kurallar kümesini içeren ilke deyimlerini tanımlarsınız.

IBM MQ for z/OS , kanalların desteklenen bir CipherSpecile yapılandırılmış SSLCIPH parametresine sahip olmasını gerektiren kendi TLS uygulamasına sahiptir.

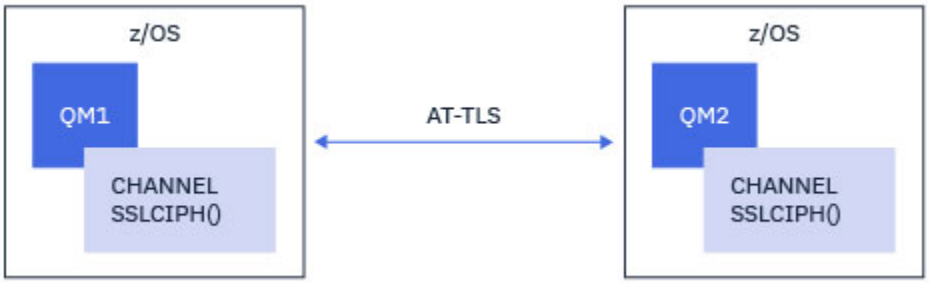
Bir kanalda TLS ' yi etkinleştirmeye karar verirken IBM MQ yöneticisi AT-TLS ya da IBM MQ TLS kullanmaya karar verebilir. Karar genellikle AT-TLS ' nin diğer ara katman yazılımları için mi yoksa performans etkileri nedeniyle mi kullanıldığına bağlıdır. AT-TLS ve IBM MQ TLS performansının temel karşılaştırması için bkz. MP16: IBM MQ for z/OSiçin Kapasite Planlama ve Ayarlama.

Senaryolar

IBM MQ ile AT-TLS kullanımı aşağıdaki senaryolarda desteklenir:

1. senaryo

Kanalın her iki tarafının AT-TLS kullandığı iki IBM MQ for z/OS kuyruk yöneticisi arasında. Yani, hiçbir kanal SSLCIPH özniteliğini belirtmez. Bu yaklaşım herhangi bir ileti kanalında kullanılabilir.



Bu senaryonun uygulanması, kanalın her bir tarafı için bir tane olmak üzere iki AT-TLS ilkesinin tanımlanmasından oluşur. Bu ilkeler, Senaryo 3 ya da Senaryo 4ile kullanılanlar ile aynıdır.

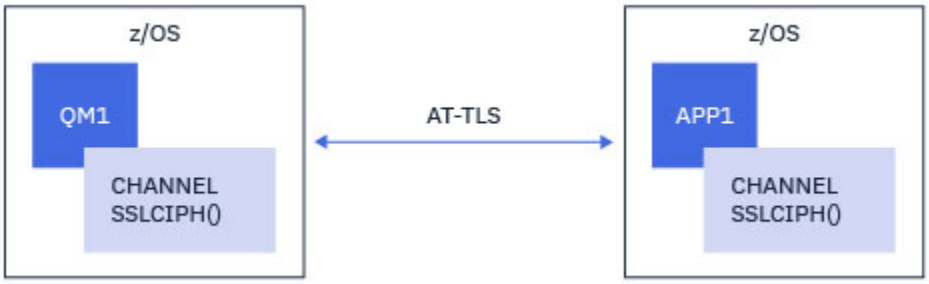
Örneğin, kanal tek bir CipherSpec kullanımından AT-TLS kullanımına değiştiriliyorsa, giden kanal “CipherSpec adlı tek bir kuyruk yöneticisini kullanarak giden kanalda AT-TLS ' nin IBM MQ for Multiplatforms kuyruk yöneticisine yapılandırılması” sayfa 461 ilkesini kullanır ve gelen kanal “CipherSpec

adlı tek bir kuyruk yöneticisini kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması” sayfa 470ilkesini kullanır.

Kanal CipherSpec diğer adını kullanarak AT-TLS ' yi kullanmaya değiştiriliyorsa, giden kanal “Giden kanalda AT-TLS ' nin CipherSpecs diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisine yapılandırılması” sayfa 465 ilkesini kullanır ve gelen kanal “CipherSpec diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması” sayfa 474ilkesini kullanır.

2. senaryo

Kanalın her iki tarafının AT-TLS kullandığı z/OS üzerinde çalışan bir IBM MQ for z/OS kuyruk yöneticisi ile bir IBM MQ Java istemci uygulaması arasında. Yani, ne sunucu-bağlantı kanalı ne de istemci-bağlantı kanalı SSLCIPH özniteliğini belirtmez.



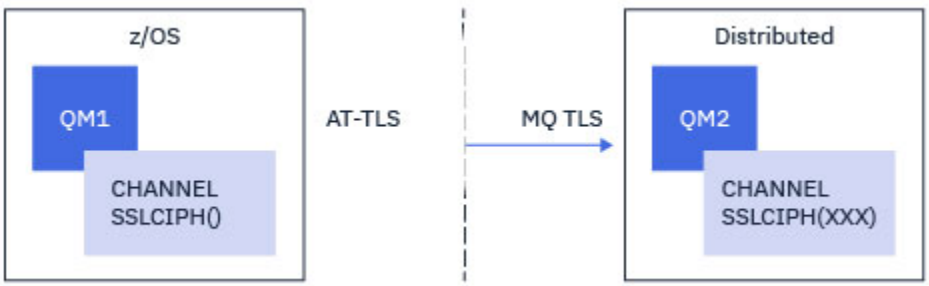
Bu senaryonun uygulanması, kanalın her bir tarafı için bir tane olmak üzere iki AT-TLS ilkesinin tanımlanmasından oluşur. Bu ilkeler, Senaryo 3 ya da Senaryo 4ile kullanılanlar ile aynıdır.

Örneğin, kanal tek bir CipherSpec kullanımından AT-TLS 'ye değiştiriliyorsa, istemci-bağlantı kanalı “CipherSpec adlı tek bir kuyruk yöneticisini kullanarak giden kanalda AT-TLS ' nin IBM MQ for Multiplatforms kuyruk yöneticisine yapılandırılması” sayfa 461 ilkesini kullanır ve sunucu bağlantısı kanalı “CipherSpec adlı tek bir kuyruk yöneticisini kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması” sayfa 470' daki ilkeyi kullanır.

Kanal CipherSpec diğer adını kullanarak AT-TLS ' yi kullanmaya değiştiriliyorsa, istemci-bağlantı kanalı “Giden kanalda AT-TLS ' nin CipherSpecs diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisine yapılandırılması” sayfa 465 ilkesini kullanır ve sunucu bağlantısı kanalı “CipherSpec diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması” sayfa 474ilkesini kullanır.

3. senaryo

Bir IBM MQ for z/OS kuyruk yöneticisi ile IBM MQ for Multiplatformsüzerinde çalışan bir kuyruk yöneticisi arasında, IBM MQ for z/OS kuyruk yöneticisinin AT-TLS kullandığı ve IBM MQ for Multiplatforms kuyruk yöneticisinin tek bir CipherSpecadlı SSLCIPH özniteliğini belirterek IBM MQ TLS ' yi kullandığı bir kuyruk yöneticisi. Bu, küme gönderen ve küme alıcısı dışındaki tüm ileti kanalı tipleri için geçerlidir.

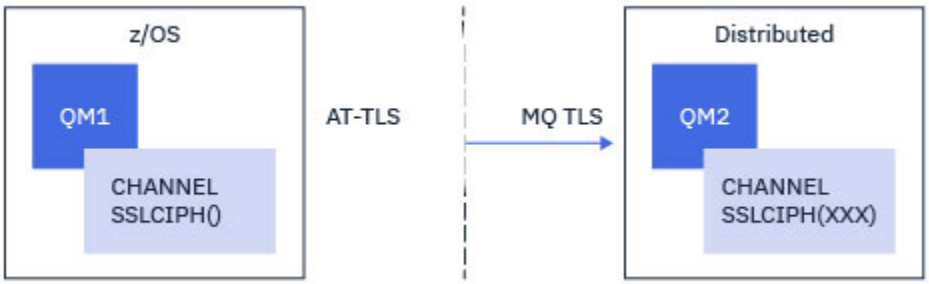


See “CipherSpec adlı tek bir kuyruk yöneticisini kullanarak giden kanalda AT-TLS ' nin IBM MQ for Multiplatforms kuyruk yöneticisine yapılandırılması” sayfa 461 for an example AT-TLS configuration for outbound channels from the IBM MQ for z/OS queue manager to the IBM MQ for Multiplatforms queue manager, and “CipherSpec adlı tek bir kuyruk yöneticisini kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması” sayfa 470 for an example AT-TLS configuration for inbound channels from the IBM MQ for Multiplatforms queue manager to the IBM MQ for z/OS queue manager.

Aynı AT-TLS yapılandırması, her iki kuyruk yöneticisi de z/OSüzerinde olduğunda kullanılabilir, ancak sağ taraftaki kuyruk yöneticisi AT-TLS kullanacak şekilde yapılandırılmadı.

4. senaryo

Bir IBM MQ for z/OS kuyruk yöneticisi ile IBM MQ for Multiplatformsüzerinde çalışan bir kuyruk yöneticisi arasında; burada IBM MQ for z/OS kuyruk yöneticisi AT-TLS kullanır ve IBM MQ for Multiplatforms kuyruk yöneticisi, CipherSpecdiğer adıyla SSLCIPH özniteliğini belirterek IBM MQ TLS kullanır. Bu, küme gönderen ve küme alıcısı dışındaki tüm ileti kanalı tipleri için geçerlidir.

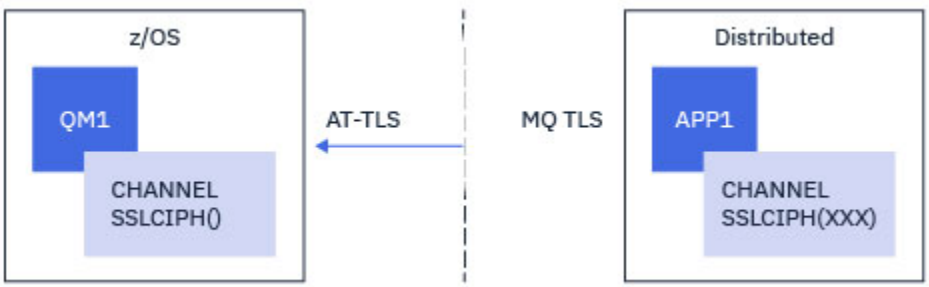


See “Giden kanalda AT-TLS ' nin CipherSpecs diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisine yapılandırılması” sayfa 465 for an example AT-TLS configuration for outbound channels from the IBM MQ for z/OS queue manager to the IBM MQ for Multiplatforms queue manager, and “CipherSpec diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması” sayfa 474, and “CipherSpec diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması” sayfa 474 for an example AT-TLS configuration for inbound channels from the IBM MQ for Multiplatforms queue manager to the IBM MQ for z/OS queue manager.

Aynı AT-TLS yapılandırması, her iki kuyruk yöneticisi de z/OSüzerinde olduğunda kullanılabilir, ancak sağ taraftaki kuyruk yöneticisi AT-TLS kullanacak şekilde yapılandırılmadı.

5. senaryo

IBM MQ for z/OS kuyruk yöneticisinin AT-TLS kullandığı ve istemci uygulamasının tek bir CipherSpecadlı SSLCIPH özniteliğini belirterek IBM MQ TLS kullandığı IBM MQ for Multiplatformsüzerinde çalışan bir IBM MQ for z/OS kuyruk yöneticisi ve istemci uygulaması arasında.

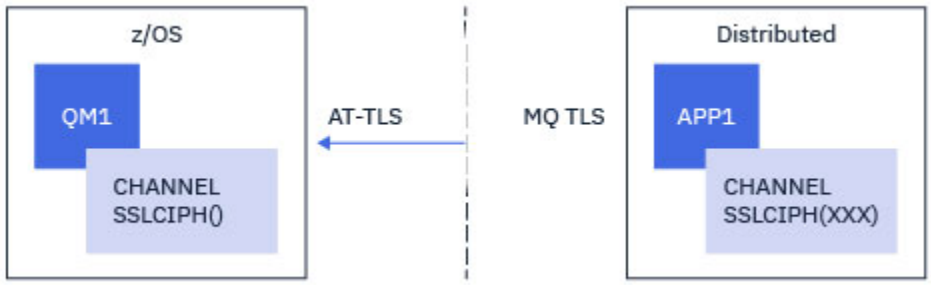


Bu senaryo, bir gelen ileti kanalı tarafından kullanılanlar ile aynı gereksinimleri karşılayan tek bir AT-TLS ilkesi gerektirir; bkz. “CipherSpec adlı tek bir kuyruk yöneticisini kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması” sayfa 470.

İstemci uygulaması bir Java uygulaması olduğunda ve z/OSüzerinde çalıştığında aynı AT-TLS yapılandırması kullanılabilir, ancak AT-TLS kullanacak şekilde yapılandırılmamış.

6. senaryo

IBM MQ for z/OS kuyruk yöneticisi ile IBM MQ for Multiplatformsüzerinde çalışan bir istemci uygulaması arasında, IBM MQ for z/OS kuyruk yöneticisi AT-TLS 'yi kullanır ve istemci uygulaması, CipherSpecdiğer adıyla SSLCIPH özniteliğini belirterek IBM MQ TLS' yi kullanır.



Bu senaryo, bir gelen ileti kanalı tarafından kullanılanlar ile aynı gereksinimleri karşılayan tek bir AT-TLS ilkesi gerektirir; bkz. “CipherSpec diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması” sayfa 474.

İstemci uygulaması bir Java uygulaması olduğunda ve z/OSüzerinde çalıştığında aynı AT-TLS yapılandırması kullanılabilir, ancak AT-TLS kullanacak şekilde yapılandırılmamış.

Sınırlamalar

IBM MQ for z/OS , AT-TLS ' nin farkında olmadığından, önceki senaryolarda geçerli olan bazı kısıtlamalar vardır:

- IBM MQ TLS ile birlikte AT-TLS, küme gönderen ve küme alıcı kanallarıyla çalışmaz.
- IBM MQ for z/OS kuyruk yöneticileri AT-TLS kullandıklarının farkında değildir ve iş ortağı kuyruk yöneticisinden ya da istemcisinden sertifika bilgisi almazlar. Bu nedenle, aşağıdaki özniteliklerin AT-TLS kullanan bir kanalın z/OS tarafında bir etkisi yoktur:
  - SSLCAUTH ve SSLPEER kanal öznitelikleri
  - SSLRKEYC kuyruk yöneticisi özniteliği
  - CHLAUTH kurallarının SSLPEERMAP öznitelikleri
- TLS gizli anahtar yeniden anlaşması kullanımı, kanalın her iki tarafının da IBM MQ TLS kullanmasını gerektirir. Bu nedenle, AT-TLS kullanarak bir IBM MQ for z/OS kuyruk yöneticisine bağlanıyorsanız, IBM MQ for Multiplatforms kuyruk yöneticisi ya da istemcisinde TLS gizli anahtar yeniden anlaşması etkinleştirilmemelidir.

  Bir kuyruk yöneticisine ilişkin TLS gizli anahtar yenileme anlaşmasını devre dışı bırakmak için kuyruk yöneticisi SSLRKEYC parametresini `0`olarak ayarlayın. Bir istemci için, ilgili değiştirgeyi istemci tipine bağlı olarak `0` olarak ayarlayın. Bunun nasıl yapılacağını öğrenmek için bkz. “SSL ve TLS gizli anahtarlarını sıfırlama” sayfa 478.

AT-TLS yapılandırma bildirimleri

AT-TLS, bir deyim kümesi kullanılarak yapılandırılır. Bu konuda belgelenen senaryolarda kullanılıyorsa:

TTLSRule
Bir TCP/IP bağlantısını TLS yapılandırmasıyla eşleştirmeye ilişkin bir ölçüt kümesini belirtir. Bu da diğer deyim tiplerini gösterir.

TTLSGroupAction
Başvuran `TTLSRule` öğesinin etkinleştirilip etkinleştirilmediğini belirtir.

TTLSEnvironmentAction
Başvuran `TTLSRule` için ayrıntılı yapılandırmayı belirtir ve diğer bazı deyimlere başvurur.

TTLSKeyringParms
AT-TLS tarafından kullanılacak anahtar halkasına başvurur.

TTLSCipherParms
Kullanılacak şifreleme takımlarını tanımlar.

TTLSEnvironmentAdvancedParms
Hangi TLS ya da SSL iletişim kurallarının etkinleştirildiğini tanımlar.

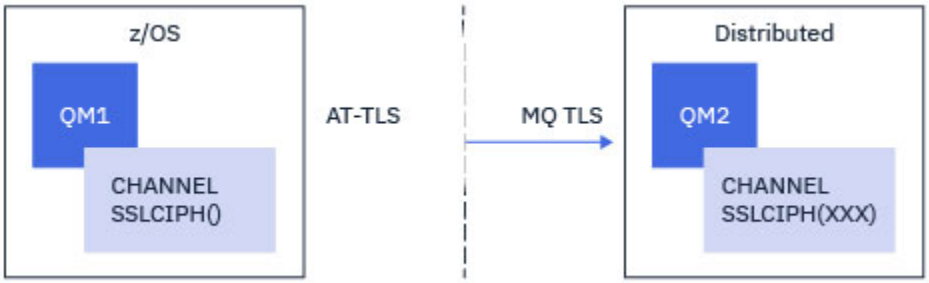
Uyarı: Burada belgelenmemiş AT-TLS içeren başka bir AT-TLS ilke bildirimleri vardır ve gereksinimine bağlı olarak IBM MQ ile birlikte kullanılabilir. Ancak, IBM MQ yalnızca bu konuda açıklanan ilkelerle sınanmıştır.

CipherSpec adlı tek bir kuyruk yöneticisini kullanarak giden kanalda AT-TLS ' nin IBM MQ for Multiplatforms kuyruk yöneticisine yapılandırılması

IBM MQ for z/OS kuyruk yöneticisinden IBM MQ for Multiplatforms kuyruk yöneticisine giden bir kanalda AT-TLS ' yi ayarlama. Bu durumda, z/OS kuyruk yöneticisindeki kanal, SSLCIPH özniteliği ayarlanmamış bir gönderen kanaldır vez/OS kuyruk yöneticisindeki kanal, SSLCIPH özniteliği tek bir CipherSpecolarak ayarlanmış bir alıcı kanaldır.

CipherSpecdiğer adını kullanan bir örnek için bkz. “Giden kanalda AT-TLS ' nin CipherSpecs diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisine yapılandırılması” sayfa 465 .

Bu örnekte, TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec kullanan var olan bir gönderici-alıcı kanal çifti IBM MQ TLS yerine AT-TLS kullanacak şekilde ayarlanacak.



Yapılandırmada küçük ayarlamalar yapılarak diğer TLS iletişim kuralları ve CipherSpecs kullanılabilir. Küme gönderen ve küme alıcı kanalları dışında, AT-TLS yapılandırmasında herhangi bir değişiklik olmadan diğer ileti kanalı tipleri de kullanılabilir.

Yordam

Adım 1: Kanalın durdurulması

Adım 2: AT-TLS ilkesi oluşturma ve uygulama

Bu senaryo için aşağıdaki AT-TLS deyimlerini oluşturmanız gerekir:

1. Kanal başlatıcı adres alanındaki giden bağlantıları hedef alıcı kanalının IP adresi ve kapı numarasıyla eşleştirmek için bir TTLSRule deyimi. Bu değerler, gönderen kanalının CONNAME içinde kullanılan bilgilerle eşleşmelidir. Burada, belirli bir kanal başlatıcı iş adıyla eşleşmesi için daha fazla süzgeç eklenmiştir.

```
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
}
```

Önceki kural, CSQ1CHIN işinden 1414 numaralı kapıdaki 123.456.78.9 IP adresine giden bağlantılarla eşleşir.

Daha ileri düzey süzgeç uygulama seçenekleri için TTLSRulekonusuna bakın.

2. Kuralı etkinleştiren bir TTLSGroupAction deyimi. TTLSRule , **TTLSGroupActionRef** özelliğini kullanarak TTLSGroupAction ' a başvurur.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. **TTLSEnvironmentActionRef** özelliği tarafından TTLSRule ile ilişkili bir TTLSEnvironmentAction deyimi. TTLSEnvironmentAction , TLS Ortamını yapılandırır ve hangi anahtarın kullanılacağını belirtir.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. **TTLSKeyringParmsRef** özelliği tarafından TTLSEnvironmentAction ile ilişkili bir TTLSKeyringParms deyimi ve AT-TLS tarafından kullanılan anahtar halkasını tanımlar.

   Anahtarlık,z/OS dışı uzak kuyruk yöneticisi tarafından güvenilen sertifikaları içermelidir. Bu anahtarlık, kanal başlatıcısı tarafından kullanılan bir anahtarlık ile aynı şekilde tanımlanabilir; bkz. “z/OS sisteminizi TLS kullanacak şekilde yapılandırma” sayfa 251.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** özelliği tarafından TTLSEnvironmentAction ile ilişkili bir TTLSCipherParms deyimi.

   Bu deyim, hedef alıcı kanalında kullanılan IBM MQ CipherSpec adının eşdeğeri olması gereken tek bir şifreleme takımı adı içermelidir.

   Not: AT-TLS şifreleme takımı adlarının IBM MQ CipherSpec adlarıyla eşleşmesi gerekmez. Ancak, aşağıdaki tabloda IBM MQ CipherSpec adını bularak ve onaltılı kod sütununu TTLSCipherParms deyimi başlığındaki Tablo 2 'deki genişletilmiş karakter sütunuyla çapraz başvuruda bulunarak IBM MQ CipherSpec adıyla eşleşen AT-TLS şifre grubu adı bulunabilir.

Çizelge 83. CipherSpecs on z/OS (IBM MQ for z/OS 9.2.0)

CipherSpec	Protokol	Onaltılı kod	Varsayılan olarak etkindir
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Evet
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Evet
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Evet
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Evet
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Evet
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Evet
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Evet
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Evet
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Evet
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Evet
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Evet
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Evet
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Evet
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Hayır
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Hayır
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Hayır
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Hayır
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Hayır
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Hayır
TRIPLE_DES_SHA_US	SSL v3	000A	Hayır
RC4_SHA_US	SSL v3	0005	Hayır
RC4_MD5_US	SSL v3	0004	Hayır

Çizelge 83. CipherSpecs on z/OS (IBM MQ for z/OS 9.2.0) (devamı var)			
CipherSpec	**Protokol**	**Onaltılı kod**	**Varsayılan olarak etkindir**
DES_SHA_EXPORT (DışA AKTARMA)	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Hayır
RC2_MD5_EXPORT	SSL v3	0006	Hayır
NULL_SHA	SSL v3	0002	Hayır
NULL_MD5	SSL v3	0001	Hayır

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. TTLSEnvironmentAdvancedParms deyimi, **TTLSEnvironmentAdvancedParmsRef** özelliği tarafından TTLSEnvironmentAction ile ilişkilendirilir.

   Bu deyim, hangi SSL ve TLS iletişim kurallarının etkinleştirildiğini belirtmek için kullanılabilir. IBM MQ ile yalnızca, TTLSCipherParms deyiminde kullanılan şifre takımı adıyla eşleşen tek iletişim kuralını etkinleştirmeniz gerekir.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

Tüm deyim kümesi aşağıdaki gibidir ve ilke aracısına uygulanmalıdır:

```
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
}
```

Adım 3: z/OS kanalından SSLCIPH ' nin kaldırılması

Aşağıdaki komutu kullanarak z/OS kanalından CipherSpec ' i kaldırın:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Adım 4: Kanalın başlatılması

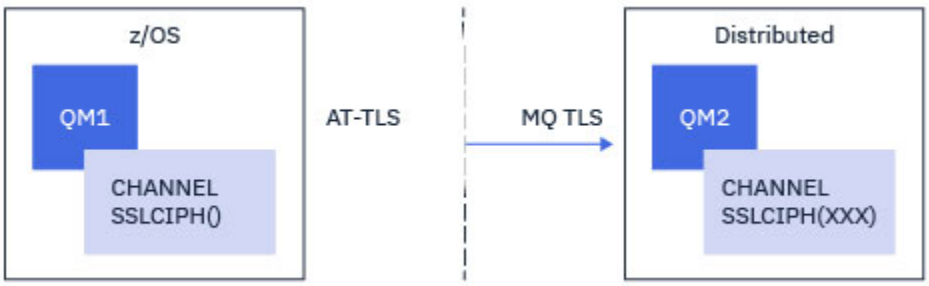
Kanal başlatıldıktan sonra AT-TLS ve IBM MQ TLS birleşimini kullanacak.

Uyarı: Önceki AT-TLS bildirimleri çok az bir yapılandırmadır. Burada belgelenmemiş AT-TLS ile birlikte başka AT-TLS ilke bildirimleri vardır ve gereksinimine bağlı olarak IBM MQ ile birlikte kullanılabilir. Ancak, IBM MQ yalnızca açıklanan ilkelerle sınanmıştır.

Giden kanalda AT-TLS ' nin CipherSpecs diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisine yapılandırılması

IBM MQ for z/OS kuyruk yöneticisinden IBM MQ for Multiplatforms kuyruk yöneticisine giden bir kanalda AT-TLS ' yi ayarlama. Bu durumda, z/OS kuyruk yöneticisindeki kanal, SSLCIPH özniteliği ayarlanmamış bir gönderen kanaldır vez/OS dışı kuyruk yöneticisindeki kanal, SSLCIPH özniteliği CipherSpec diğer adına ayarlanmış bir alıcı kanaldır.

Bu örnekte, ANY_TLS13 diğer adını CipherSpec kullanan var olan bir gönderici-alıcı kanal çifti, gönderen kanalının IBM MQ TLS yerine AT-TLS kullanması için ayarlanacak.



Yapılandırmada küçük ayarlamalar yapılarak diğer TLS iletişim kuralları ve CiperSpecs kullanılabilir. Küme gönderen ve küme alıcı kanalları dışında, AT-TLS yapılandırmasında herhangi bir değişiklik olmadan diğer ileti kanalı tipleri de kullanılabilir.

Yordam

Adım 1: Kanalın durdurulması

Adım 2: AT-TLS ilkesi oluşturma ve uygulama

Bu senaryo için aşağıdaki AT-TLS deyimlerini oluşturmanız gerekir:

1. Kanal başlatıcı adres alanındaki giden bağlantıları hedef alıcı kanalının IP adresi ve kapı numarasıyla eşleştirmek için bir TTLSRule deyimi. Bu değerler, gönderen kanalının CONNAME içinde kullanılan bilgilerle eşleşmelidir. Burada, belirli bir kanal başlatıcı iş adıyla eşleşmesi için daha fazla süzgeç eklenmiştir.

```
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
}
```

   Önceki kural, CSQ1CHIN işinden 1414 numaralı kapıdaki 123.456.78.9 IP adresine giden bağlantılarla eşleşir.

   Daha ileri düzey süzgeç uygulama seçenekleri için TTLSRulekonusuna bakın.
2. Kuralı etkinleştiren bir TTLSGroupAction deyimi. TTLSRule , **TTLSGroupActionRef** özelliğini kullanarak TTLSGroupAction ' a başvurur.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. **TTLSEnvironmentActionRef** özelliği tarafından TTLSRule ile ilişkili bir TTLSEnvironmentAction deyimi. TTLSEnvironmentAction , TLS Ortamını yapılandırır ve hangi anahtarın kullanılacağını belirtir.

```
TTLSEnvironmentAction               CSQ1-OUTBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                     CLIENT
  TTLSKeyringParmsRef               CSQ1-KEYRING
  TTLSCipherParmsRef                CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef   CSQ1-ENVIRONMENT-ADVANCED
}
```

4. **TTLSKeyringParmsRef** özelliği tarafından TTLSEnvironmentAction ile ilişkili bir TTLSKeyringParms deyimi ve AT-TLS tarafından kullanılan anahtar halkasını tanımlar.

   Anahtarlık,z/OS dışı uzak kuyruk yöneticisi tarafından güvenilen sertifikaları içermelidir. Bu anahtarlık, kanal başlatıcısı tarafından kullanılan bir anahtarlık ile aynı şekilde tanımlanabilir; bkz. “z/OS sisteminizi TLS kullanacak şekilde yapılandırma” sayfa 251.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** özelliği tarafından TTLSEnvironmentAction ile ilişkili bir TTLSCipherParms deyimi.

   Bu deyim, hedef alıcı kanalında kullanılan CipherSpec diğer adının örtük olarak belirttiği CipherSpecs kümesiyle uyumlu olması gereken bir ya da daha çok şifreleme takımı adı içermelidir.

   Not: AT-TLS şifreleme takımı adlarının IBM MQ CipherSpec adlarıyla eşleşmesi gerekmez. Ancak, aşağıdaki tabloda IBM MQ CipherSpec adını bularak ve TTLSCipherParms konusunda Tablo 2 'deki genişletilmiş karakter sütunuyla onaltılı kod sütununa çapraz başvurarak IBM MQ CipherSpec adıyla eşleşen AT-TLS şifre grubu adı bulunabilir.

Çizelge 84. CipherSpecs on z/OS (IBM MQ for z/OS 9.2.0)

CipherSpec	Protokol	Onaltılı kod	Varsayılan olarak etkindir
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Evet
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Evet
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Evet
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Evet
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Evet
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Evet
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Evet
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Evet
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Evet
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Evet

Çizelge 84. CipherSpecs on z/OS (IBM MQ for z/OS 9.2.0) (devamı var)			
CipherSpec	**Protokol**	**Onaltılı kod**	**Varsayılan olarak etkindir**
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Evet
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Evet
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Evet
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Hayır
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Hayır
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Hayır
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Hayır
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Hayır
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Hayır
TRIPLE_DES_SHA_US	SSL v3	000A	Hayır
RC4_SHA_US	SSL v3	0005	Hayır
RC4_MD5_US	SSL v3	0004	Hayır
DES_SHA_EXPORT (DışA AKTARMA)	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Hayır
RC2_MD5_EXPORT	SSL v3	0006	Hayır
NULL_SHA	SSL v3	0002	Hayır
NULL_MD5	SSL v3	0001	Hayır

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

Uyarı: Hem kuyruk yöneticisi hem de AT-TLS ilkesi TLS 1.3' ü desteklerse, kanalın başlatılmasına yalnızca en az bir TLS 1.3 CipherSpec içeren CipherSpecs diğer adı izin verir. Örneğin, ANY_TLS12 'nin kullanılması, TTLSCipherParms TLS 1.2 CipherSpecs' i içerse de, ancak ANY_TLS12_OR_HIGHER ya da ANY_TLS13 ' un kullanılması kanalın başlatılmasına izin verse de, kanalın başlatılamamasına neden olur. Açıklama için bkz. “Diğer ad CipherSpec ayarları arasındaki ilişki” sayfa 453 .

6. TTLSEnvironmentAdvancedParms deyimi, **`TTLSEnvironmentAdvancedParmsRef`** özelliği tarafından `TTLSEnvironmentAction` ile ilişkilendirilir.

Bu deyim, hangi SSL ve TLS iletişim kurallarının etkinleştirildiğini belirtmek için kullanılabilir ve TTLSCipherParms deyimindeki şifreleme takımlarıyla tutarlı olmalıdır.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

Tüm deyim kümesi aşağıdaki gibidir ve ilke aracısına uygulanmalıdır:

```
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
}
```

Adım 3: z/OS kanalından SSLCIPH ' nin kaldırılması

Aşağıdaki komutu kullanarak z/OS kanalından CipherSpec ' i kaldırın:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Adım 4: Kanalın başlatılması

Kanal başlatıldıktan sonra AT-TLS ve IBM MQ TLS birleşimini kullanacak.

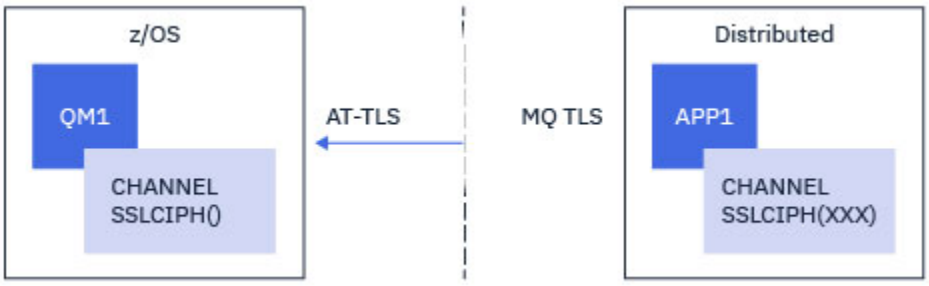
Uyarı: Önceki AT-TLS bildirimleri çok az bir yapılandırmadır. Burada belgelenmemiş AT-TLS ile birlikte başka AT-TLS ilke bildirimleri vardır ve gereksinimine bağlı olarak IBM MQ ile birlikte kullanılabilir. Ancak, IBM MQ yalnızca açıklanan ilkelerle sınanmıştır.

CipherSpec adlı tek bir kuyruk yöneticisini kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması

IBM MQ for Multiplatforms kuyruk yöneticisinden IBM MQ for z/OS kuyruk yöneticisine gelen bir kanalda AT-TLS ' yi ayarlama. Bu durumda, z/OS kuyruk yöneticisindeki kanal SSLCIPH özniteliği ayarlanmamış bir alıcı kanaldır vez/OS dışı kuyruk yöneticisindeki kanal, SSLCIPH özniteliği tek bir CipherSpecolarak ayarlanmış bir gönderen kanaldır.

CipherSpecdiğer adını kullanan bir örnek için bkz. “CipherSpec diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması” sayfa 474 .

Bu örnekte, TLS 1.3 TLS_AES_256_GCM_SHA384 CipherSpec kullanan var olan bir gönderici-alıcı kanal çifti, alıcı kanalının IBM MQ TLS yerine AT-TLS kullanması için ayarlanacak.



Yapılandırmada küçük ayarlamalar yapılarak diğer TLS iletişim kuralları ve CipherSpecs kullanılabilir. Küme gönderen ve küme alıcı kanalları dışında, AT-TLS yapılandırmasında herhangi bir değişiklik olmadan diğer ileti kanalı tipleri de kullanılabilir.

Yordam

Adım 1: Kanalın durdurulması

Adım 2: AT-TLS ilkesi oluşturma ve uygulama

Bu senaryo için aşağıdaki AT-TLS deyimlerini oluşturmanız gerekir:

1. Gönderen kanalın IP adresinden kanal başlatıcı adres alanıyla gelen bağlantıları eşleştirmek için bir TTLSRule deyimi. Burada, belirli bir kanal başlatıcı iş adıyla eşleşmesi için daha fazla süzgeç eklenmiştir.

```
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
}
```

   Önceki kural, 123.456.78.9uzak IP adresinden 1414 numaralı yerel kapıdaki CSQ1CHIN işine gelen bağlantılarla eşleşir.

   Daha ileri düzey süzgeç uygulama seçenekleri için TTLSRulekonusuna bakın.

2. Kuralı etkinleştiren bir TTLSGroupAction deyimi. TTLSRule , **TTLSGroupActionRef** özelliğini kullanarak TTLSGroupAction ' a başvurur.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. TTLSEnvironmentAction deyimi, **TTLSEnvironmentActionRef** özelliği tarafından TTLSRule ile ilişkilendirilir. TTLSEnvironmentAction , TLS Ortamını yapılandırır ve hangi anahtarın kullanılacağını belirtir.

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS, SSLCAUTH kanal özniteliğini kullanmaya eşdeğer olan karşılıklı kimlik doğrulaması sağlama yeteneği sağlar. Bu, gelen TTLSEnvironmentAction deyimi için **HandshakeRole** değeri *ServerWithClientAuth* olan bir TTLSEnvironmentAction deyimi ile yapılır.

4. TTLSKeyringParms deyimi, **TTLSKeyringParmsRef** özelliği tarafından TTLSEnvironmentAction ile ilişkilendirilir ve AT-TLS tarafından kullanılan anahtar halkasını tanımlar.

   Anahtarlık,z/OS dışı uzak kuyruk yöneticisi tarafından güvenilen sertifikaları içermelidir. Bu anahtarlık, kanal başlatıcısı tarafından kullanılan bir anahtarlık ile aynı şekilde tanımlanabilir; bkz. “z/OS sisteminizi TLS kullanacak şekilde yapılandırma” sayfa 251.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** özelliği tarafından TTLSEnvironmentAction ile ilişkili bir TTLSCipherParms deyimi.

   Bu deyim, uzak gönderen kanalında kullanılan IBM MQ CipherSpec adının eşdeğeri olması gereken tek bir şifreleme takımı adı içermelidir.

   Not: AT-TLS şifreleme takımı adlarının IBM MQ CipherSpec adlarıyla eşleşmesi gerekmez. Ancak, aşağıdaki tabloda IBM MQ CipherSpec adını bularak ve onaltılı kod sütununu TTLSCipherParms deyimi başlığındaki Tablo 2 'deki genişletilmiş karakter sütunuyla çapraz başvuruda bulunarak IBM MQ CipherSpec adıyla eşleşen AT-TLS şifre grubu adı bulunabilir.

Çizelge 85. CipherSpecs on z/OS (IBM MQ for z/OS 9.2.0)

CipherSpec	Protokol	Onaltılı kod	Varsayılan olarak etkindir
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Evet
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Evet
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Evet
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Evet
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Evet

Çizelge 85. CipherSpecs on z/OS (IBM MQ for z/OS 9.2.0) (devamı var)			
CipherSpec	**Protokol**	**Onaltılı kod**	**Varsayılan olarak etkindir**
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Evet
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Evet
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Evet
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Evet
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Evet
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Evet
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Evet
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Evet
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Hayır
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Hayır
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Hayır
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Hayır
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Hayır
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Hayır
TRIPLE_DES_SHA_US	SSL v3	000A	Hayır
RC4_SHA_US	SSL v3	0005	Hayır
RC4_MD5_US	SSL v3	0004	Hayır
DES_SHA_EXPORT (DışA AKTARMA)	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Hayır
RC2_MD5_EXPORT	SSL v3	0006	Hayır
NULL_SHA	SSL v3	0002	Hayır
NULL_MD5	SSL v3	0001	Hayır

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_AES_256_GCM_SHA384
}
```

6. TTLSEnvironmentAdvancedParms deyimi, **TTLSEnvironmentAdvancedParmsRef** özelliği tarafından `TTLSEnvironmentAction` ile ilişkilendirilir.

   Bu deyim, hangi SSL ve TLS iletişim kurallarının etkinleştirildiğini belirtmek için kullanılabilir. IBM MQ ile yalnızca, `TTLSCipherParms` deyiminde kullanılan şifre takımı adıyla eşleşen tek iletişim kuralını etkinleştirmeniz gerekir.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3          OFF
  TLSv1          OFF
  TLSv1.1        OFF
  SecondaryMap   OFF
  TLSv1.2        OFF
  TLSv1.3        ON
}
```

Tüm deyim kümesi aşağıdaki gibidir ve ilke aracısına uygulanmalıdır:

```
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
}
```

Adım 3: z/OS kanalından SSLCIPH ' nin kaldırılması

Aşağıdaki komutu kullanarak z/OS kanalından CipherSpec ' i kaldırın:

```
ALTER CHANNEL(channel-name) CHLTYPE(RCVR) SSLCIPH(' ')
```

Adım 4: Kanalın başlatılması

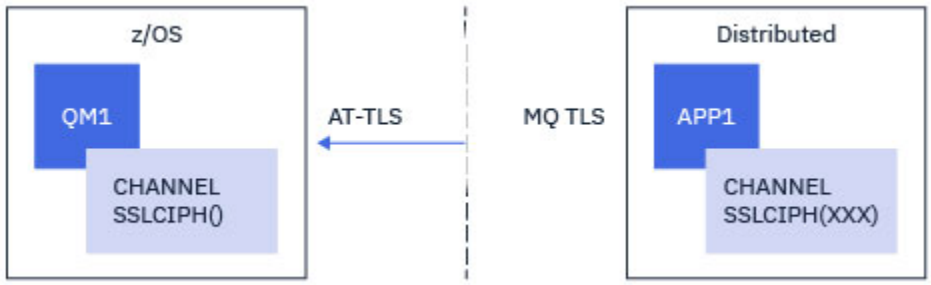
Kanal başlatıldıktan sonra AT-TLS ve IBM MQ TLS birleşimini kullanacak.

Uyarı: Önceki AT-TLS bildirimleri çok az bir yapılandırmadır. Burada belgelenmemiş AT-TLS ile birlikte başka AT-TLS ilke bildirimleri vardır ve gereksinimine bağlı olarak IBM MQ ile birlikte kullanılabilir. Ancak, IBM MQ yalnızca açıklanan ilkelerle sınanmıştır.

CipherSpec diğer adını kullanarak IBM MQ for Multiplatforms kuyruk yöneticisinden gelen kanalda AT-TLS ' nin yapılandırılması

IBM MQ for Multiplatforms kuyruk yöneticisinden IBM MQ for z/OS kuyruk yöneticisine gelen bir kanalda AT-TLS ' yi ayarlama. Bu durumda, z/OS kuyruk yöneticisindeki kanal, SSLCIPH özniteliği ayarlanmamış bir alıcı kanaldır vez/OS dışı kuyruk yöneticisindeki kanal, SSLCIPH özniteliği CipherSpecdiğer adına ayarlanmış bir gönderen kanaldır.

Bu örnekte, herhangi bir TLS 1.3 CipherSpec kullanan var olan bir gönderici-alıcı kanal çifti, alıcı kanalının IBM MQ TLS yerine AT-TLS kullanması için ayarlanacak.



Yapılandırmada küçük ayarlamalar yapılarak diğer TLS iletişim kuralları ve CiperSpecs kullanılabilir. Küme gönderen ve küme alıcı kanalları dışında, AT-TLS yapılandırmasında herhangi bir değişiklik olmadan diğer ileti kanalı tipleri de kullanılabilir.

Yordam

Adım 1: Kanalın durdurulması

Adım 2: AT-TLS ilkesi oluşturma ve uygulama

Bu senaryo için aşağıdaki AT-TLS deyimlerini oluşturmanız gerekir:

1. Gönderen kanalın IP adresinden kanal başlatıcı adres alanıyla gelen bağlantıları eşleştirmek için bir TTLSRule deyimi. Burada, belirli bir kanal başlatıcı iş adıyla eşleşmesi için daha fazla süzgeç eklenmiştir.

```
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
}
```

Önceki kural, 123.456.78.9uzak IP adresinden 1414 numaralı yerel kapıdaki CSQ1CHIN işine gelen bağlantılarla eşleşir.

Daha ileri düzey süzgeç uygulama seçenekleri için TTLSRulekonusuna bakın.

2. Kuralı etkinleştiren bir TTLSGroupAction deyimi. TTLSRule , **TTLSGroupActionRef** özelliğini kullanarak TTLSGroupAction ' a başvurur.

```
TTLSGroupAction             CSQ1-GROUP-ACTION
{
  TTLSEnabled               ON
}
```

3. TTLSEnvironmentAction deyimi, **TTLSEnvironmentActionRef** özelliği tarafından TTLSRule ile ilişkilendirilir. TTLSEnvironmentAction , TLS Ortamını yapılandırır ve hangi anahtarın kullanılacağını belirtir.

```
TTLSEnvironmentAction                 CSQ1-INBOUND-ENVIRONMENT-ACTION
{
  HandshakeRole                       SERVER
  TTLSKeyringParmsRef                 CSQ1-KEYRING
  TTLSCipherParmsRef                  CSQ1-CIPHERPARM
  TTLSEnvironmentAdvancedParmsRef     CSQ1-ENVIRONMENT-ADVANCED
}
```

AT-TLS, SSLCAUTH kanal özniteliğini kullanmaya eşdeğer olan karşılıklı kimlik doğrulaması sağlama yeteneği sağlar. Bu, gelen TTLSEnvironmentAction deyimi için **HandshakeRole** değeri *ServerWithClientAuth* olan bir TTLSEnvironmentAction deyimi ile yapılır.

4. TTLSKeyringParms deyimi, **TTLSKeyringParmsRef** özelliği tarafından TTLSEnvironmentAction ile ilişkilendirilir ve AT-TLS tarafından kullanılan anahtar halkasını tanımlar.

Anahtarlık,z/OS dışı uzak kuyruk yöneticisi tarafından güvenilen sertifikaları içermelidir. Bu anahtarlık, kanal başlatıcısı tarafından kullanılan bir anahtarlık ile aynı şekilde tanımlanabilir; bkz. “z/OS sisteminizi TLS kullanacak şekilde yapılandırma” sayfa 251.

```
TTLSKeyringParms            CSQ1-KEYRING
{
  Keyring                   MQCHIN/CSQ1RING
}
```

5. **TTLSCipherParmsRef** özelliği tarafından TTLSEnvironmentAction ile ilişkili bir TTLSCipherParms deyimi.

Bu deyim, uzak gönderen kanalında ayarlanan CipherSpec diğer adında bulunan en az bir şifreleme takımı adı içermelidir.

Not: AT-TLS şifreleme takımı adlarının IBM MQ CipherSpec adlarıyla eşleşmesi gerekmez. Ancak, aşağıdaki tabloda IBM MQ CipherSpec adını bularak ve onaltılı kod sütununu TTLSCipherParms deyimi başlığındaki Tablo 2 'deki genişletilmiş karakter sütunuyla çapraz başvuruda bulunarak IBM MQ CipherSpec adıyla eşleşen AT-TLS şifre grubu adı bulunabilir.

Çizelge 86. CipherSpecs on z/OS (IBM MQ for z/OS 9.2.0)

CipherSpec	Protokol	Onaltılı kod	Varsayılan olarak etkindir
TLS_CHACHA20_POLY1305_SHA256	TLS 1.3	1303	Evet
TLS_AES_256_GCM_SHA384	TLS 1.3	1302	Evet

Çizelge 86. CipherSpecs on z/OS (IBM MQ for z/OS 9.2.0) (devamı var)

CipherSpec	Protokol	Onaltılı kod	Varsayılan olarak etkindir
TLS_AES_128_GCM_SHA256	TLS 1.3	1301	Evet
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	009D	Evet
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	C030	Evet
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	003D	Evet
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	C024	Evet
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	C028	Evet
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	009C	Evet
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	C02F	Evet
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	003C	Evet
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	C023	Evet
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	C027	Evet
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	003B	Hayır
TLS_RSA_WITH_AES_256_CBC_SHA	TLS 1.0	0035	Hayır
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	002F	Hayır
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.0	000A	Hayır
TLS_RSA_WITH_RC4_128_SHA	TLS 1.0	0005	Hayır
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	0009	Hayır
TRIPLE_DES_SHA_US	SSL v3	000A	Hayır
RC4_SHA_US	SSL v3	0005	Hayır
RC4_MD5_US	SSL v3	0004	Hayır
DES_SHA_EXPORT (DışA AKTARMA)	SSL v3	0009	N
RC4_MD5_EXPORT	SSL v3	0003	Hayır

Çizelge 86. CipherSpecs on z/OS (IBM MQ for z/OS 9.2.0) (devamı var)			
CipherSpec	**Protokol**	**Onaltılı kod**	**Varsayılan olarak etkindir**
RC2_MD5_EXPORT	SSL v3	0006	Hayır
NULL_SHA	SSL v3	0002	Hayır
NULL_MD5	SSL v3	0001	Hayır

```
TTLSCipherParms             CSQ1-CIPHERPARM
{
  V3CipherSuites            TLS_CHACHA20_POLY1305_SHA256
  V3CipherSuites            TLS_AES_256_GCM_SHA384
  V3CipherSuites            TLS_AES_128_GCM_SHA256
}
```

Uyarı: Hem kuyruk yöneticisi hem de AT-TLS ilkesi TLS 1.3' ü desteklerse, kanalın başlatılmasına yalnızca en az bir TLS 1.3 CipherSpec içeren CipherSpecs diğer adı izin verir. Örneğin, ANY_TLS12 'nin kullanılması, TTLSCipherParms TLS 1.2 CipherSpecs' i içerse de, ancak ANY_TLS12_OR_HIGHER ya da ANY_TLS13 ' un kullanılması kanalın başlatılmasına izin verse de, kanalın başlatılamamasına neden olur. Açıklama için bkz. “Diğer ad CipherSpec ayarları arasındaki ilişki” sayfa 453 .

6. TTLSEnvironmentAdvancedParms deyimi, **TTLSEnvironmentAdvancedParmsRef** özelliği tarafından TTLSEnvironmentAction ile ilişkilendirilir.

   Bu deyim, hangi SSL ve TLS iletişim kurallarının etkinleştirildiğini belirtmek için kullanılabilir ve TTLSCipherParms deyimindeki şifreleme takımlarıyla tutarlı olmalıdır.

```
TTLSEnvironmentAdvancedParms CSQ1-ENVIRONMENT-ADVANCED
{
  SSLv3           OFF
  TLSv1           OFF
  TLSv1.1         OFF
  SecondaryMap    OFF
  TLSv1.2         OFF
  TLSv1.3         ON
}
```

Tüm deyim kümesi aşağıdaki gibidir ve ilke aracısına uygulanmalıdır:

```
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
}
```

Adım 3: z/OS kanalından SSLCIPH ' nin kaldırılması

Aşağıdaki komutu kullanarak z/OS kanalından CipherSpec ' i kaldırın:

```
ALTER CHANNEL(channel-name) CHLTYPE(SDR) SSLCIPH(' ')
```

Adım 4: Kanalın başlatılması

Kanal başlatıldıktan sonra AT-TLS ve IBM MQ TLS birleşimini kullanacak.

Uyarı: Önceki AT-TLS bildirimleri çok az bir yapılandırmadır. Burada belgelenmemiş AT-TLS ile birlikte başka AT-TLS ilke bildirimleri vardır ve gereksinimine bağlı olarak IBM MQ ile birlikte kullanılabilir. Ancak, IBM MQ yalnızca açıklanan ilkelerle sınanmıştır.

SSL ve TLS gizli anahtarlarını sıfırlama

IBM MQ , kuyruk yöneticilerinde ve istemcilerde gizli anahtarların ilk durumuna getirilmesini destekler.

Belirtilen sayıda şifrelenmiş veri kanal boyunca aktığında gizli anahtarlar sıfırlanır. Kanal sağlıklı işletim bildirimleri etkinleştirilirse, kanal sağlıklı işletim bildirimi gönderilmeden ya da alınmadan önce gizli anahtar sıfırlanır.

Anahtar ilk duruma getirme değeri her zaman IBM MQ kanalının başlangıç tarafından ayarlanır.

Kuyruk yöneticisi

Bir kuyruk yöneticisi için, anahtar yeniden anlaşması sırasında kullanılan değerleri ayarlamak üzere **SSLRKEYC** parametresiyle birlikte **ALTER QMGR** komutunu kullanın.

IBM i IBM isistemlerinde **CHGMQM** değiştirgesini **SSLRSTCNT** değiştirgesiyle kullanın.

MQI istemcisi

Varsayılan olarak, MQI istemcileri gizli anahtarı yeniden anlaşmaz. Bir MQI istemcisinin anahtarı üç şekilde yeniden görüşmesini sağlayabilirsiniz. Aşağıdaki listede, yöntemler öncelik sırasına göre gösterilir. Birden çok değer belirtirseniz, en yüksek öncelik değeri kullanılır.

1. MQCONNX çağrısında MQSCO yapısındaki KeyResetCount (Anahtar Sıfırlama Sayısı) alanını kullanarak.
2. **MQSSLRESET**ortam değişkenini kullanarak.
3. İstemci yapılanış kütüğünün SSL kısmına **SSLKeyResetCount** özniteliğini ayarlayarak.

Bu değişkenler, TLS gizli anahtarı yeniden anlaşılmadan önce TLS etkileşimi içinde gönderilen ve alınan şifrelenmemiş bayt sayısını gösteren, 0-999 999 999 aralığında bir tamsayıya ayarlanabilir. 0 değerinin belirtilmesi, TLS gizli anahtarlarının hiçbir zaman yeniden anlaşılmadığını gösterir. 1-32 KB aralığında bir TLS gizli anahtar sıfırlama sayısı belirtirseniz, TLS kanalları 32 KB gizli anahtar sıfırlama sayısını kullanır. Bu, küçük TLS gizli anahtar sıfırlama değerleri için ortaya çıkabilecek aşırı anahtar sıfırlamalarını önlemektir.

Sıfırdan büyük bir değer belirtilirse ve kanal için kanal sağlıklı işletim bildirimleri etkinleştirilirse, ileti verileri bir kanal sağlıklı işletim bildirimi sonrasında gönderilmeden ya da alınmadan önce gizli anahtar da yeniden belirlenir.

Her başarılı yeniden anlaşma sonrasında bir sonraki gizli anahtar yeniden anlaşması sıfırlanıncaya kadar bayt sayısı.

Java

IBM MQ classes for Javaiçin bir uygulama gizli anahtarı aşağıdaki yollardan biriyle sıfırlayabilir:

- MQEnvironment sınıfında sslResetSayı alanını ayarlayarak.
- Bir Hashtable nesnesinde MQC.SSL_RESET_COUNT_PROPERTY ortam özelliğini ayarlayarak. Daha sonra uygulama, MQEnvironment sınıfındaki `properties` alanına Hashtable 'ı atar ya da Hashtable 'ı oluşturucusundaki bir MQQueueManager nesnesine geçirir.

Uygulama bu yöntemlerden birden fazlasını kullanıyorsa, olağan öncelik kuralları geçerlidir. Öncelik kuralları için bkz. Class com.ibm.mq.MQEnvironment .

sslResetSayı alanı ya da ortam özelliği MQC.SSL_RESET_COUNT_PROPERTY değeri, gizli anahtar yeniden anlaşılmadan önce IBM MQ classes for Java istemci kodu tarafından gönderilen ve alınan toplam bayt sayısını gösterir. Gönderilen bayt sayısı, şifrelemeden önceki sayıdır ve alınan bayt sayısı, şifre çözmeden sonraki sayıdır. Bayt sayısı, IBM MQ classes for Java istemcisi tarafından gönderilen ve alınan denetim bilgilerini de içerir.

Sıfırlama sayısı sıfırsa (varsayılan değer), gizli anahtar hiçbir zaman yeniden anlaşılmaz. CipherSuite belirtilmezse sıfırlama sayısı yoksayılır.

JMS

IBM MQ classes for JMSiçin SSLRESETCOUNT özelliği, şifreleme için kullanılan gizli anahtar yeniden anlaşılmadan önce bir bağlantı tarafından gönderilen ve alınan toplam bayt sayısını gösterir. Gönderilen bayt sayısı, şifrelemeden önceki sayıdır ve alınan bayt sayısı, şifre çözmeden sonraki sayıdır. Bayt sayısı, IBM MQ classes for JMStarafından gönderilen ve alınan denetim bilgilerini de içerir. Örneğin, 4 MB veri aktıktan sonra yeniden görüşülen bir gizli anahtarla TLS etkin bir MQI kanalı üzerinden bağlantı

oluşturmak için kullanılabilecek bir ConnectionFactory nesnesi yapılandırmak için JMSAdmin 'e şu komutu verin:

```
ALTER CF(my.cf) SSLRESETCOUNT(4194304)
```

Varsayılan değer olan SSLRESETCOUNT değeri sıfırsa, gizli anahtar hiçbir zaman yeniden anlaşılmaz. SSLCIPHERSUITE ayarlanmazsa, SSLRESETCOUNT özelliği yoksayılır.

.NET

.NET yönetilmeyen istemciler için, **SSLKeyResetCount** tamsayı özelliği, gizli anahtar yeniden anlaşılmadan önce TLS etkileşimi içinde gönderilen ve alınan şifrelenmemiş bayt sayısını gösterir. IBM MQ classes for .NETiçinde nesne özelliklerinin kullanımı hakkında daha fazla bilgi için bkz. Öznitelik değerlerini alma ve ayarlama.

.NET yönetilen istemciler için, SSLStream sınıfı gizli anahtar yeniden ayarlamayı/yeniden anlaşmayı desteklemez. Ancak, diğer IBM MQ istemcileriyle tutarlı olması için IBM MQ yönetilen .NET istemcisi, uygulamaların **SSLKeyResetCount**öğesini ayarlamasına izin verir. Daha fazla bilgi için bkz. Gizli anahtar sıfırlama ya da yeniden anlaşma.

XMS .NET

XMS .NET yönetilmeyen istemciler için IBM MQ kuyruk yöneticisine güvenli bağlantılarbaşlıklı konuya bakın.

İlgili başvurular
ALTER QMGR
DISPLAYQMGR
İleti Kuyruğu Yöneticisini Değiştir (CHGMQM)
İleti Kuyruğu Yöneticisini Görüntüle (DSPMQM)

Kullanıcı çıkış programlarında gizlilik uygulanması

Güvenlik çıkışlarında gizliliğin uygulanması

Güvenlik çıkışları, kanalda akan verileri şifrelemek ve verilerin şifresini çözmek için simetrik anahtarı üreterek ve dağıtarak gizlilik hizmetinde bir rol oynayabilir. Bunu yapmak için yaygın bir teknik PKI teknolojisini kullanır.

Bir güvenlik çıkışı rasgele bir veri değeri oluşturur, bunu, ortak güvenlik çıkışının temsil ettiği kuyruk yöneticisinin ya da kullanıcının genel anahtarıyla şifreler ve şifrelenmiş verileri bir güvenlik iletisinde iş ortağına gönderir. Ortak güvenlik çıkışı, temsil ettiği kuyruk yöneticisinin ya da kullanıcının özel anahtarıyla rasgele veri değerinin şifresini çözer. Her güvenlik çıkışı artık her ikisi tarafından da bilinen bir algoritmayı kullanarak simetrik anahtarı birbirinden bağımsız olarak türetmek için rasgele veri değerini kullanabilir. Diğer bir seçenek olarak, rasgele veri değerini anahtar olarak kullanabilirler.

İlk güvenlik çıkışı bu zamana kadar ortağının kimliğini doğrulamadıysa, iş ortağı tarafından gönderilen bir sonraki güvenlik iletisi, simetrik anahtarla şifrelenmiş beklenen bir değer içerebilir. İlk güvenlik çıkışı, iş ortağı güvenlik çıkışının beklenen değeri doğru şekilde şifrelediğini denetleyerek iş ortağının kimliğini doğrulayabilir.

Güvenlik çıkışları, birden fazla algoritma kullanılabiliyorsa, kanalda akan verileri şifrelemek ve verilerin şifresini çözmek için algoritmayı kabul etmek için de bu fırsatı kullanabilir.

İleti çıkışlarında gizlilik uygulanıyor

Bir kanalın gönderme ucundaki bir ileti çıkışı, bir iletideki uygulama verilerini şifreleyebilir ve kanalın alıcı ucundaki başka bir ileti çıkışı, verilerin şifresini çözebilir. Performans nedenleriyle, normalde bu amaç için

bir simetrik anahtar algoritması kullanılır. Simetrik anahtarın nasıl oluşturulabileceği ve dağıtılabileceği hakkında daha fazla bilgi için bkz. “Kullanıcı çıkış programlarında gizlilik uygulanması” sayfa 480.

Yerleşik ileti tanımlayıcısını içeren iletim kuyruğu üstbilgisi MQXQH gibi bir iletideki üstbilgiler bir ileti çıkışıyla şifrelenmemelidir. Bunun nedeni, ileti üstbilgilerinin veri dönüştürmesinin, gönderen uçta bir ileti çıkışı çağrıldıktan sonra ya da alıcı uçta bir ileti çıkışı çağrılmadan önce gerçekleşmesi olabilir. Üstbilgiler şifrelenirse, veri dönüştürme başarısız olur ve kanal durur.

Gönderme ve alma çıkışlarında gizlilik uygulanıyor

Bir kanalda akan verileri şifrelemek ve verilerin şifresini çözmek için gönderme ve alma çıkışları kullanılabilir. Bunlar, aşağıdaki nedenlerden ötürü bu hizmeti sağlamak için ileti çıkışlarından daha uygundur:

- Bir ileti kanalında, ileti üstbilgileri ve iletilerdeki uygulama verileri şifrelenebilir.
- Gönderme ve alma çıkışları, ileti kanallarının yanı sıra MQI kanallarında da kullanılabilir. MQI çağrılarına ilişkin değiştirgeler, MQI kanalında akarken korunması gereken duyarlı uygulama verileri içerebilir. Bu nedenle her iki kanalda da aynı gönderme ve alma çıkışlarını kullanabilirsiniz.

API çıkışında ve API geçişi çıkışında gizlilik uygulanıyor

Bir iletideki uygulama verileri, ileti gönderen uygulama tarafından konduğunda ve ileti alan uygulama tarafından alındığında ikinci bir çıkış tarafından şifresi çözüldüğünde bir API veya API geçiş çıkışı tarafından şifrelenebilir. Performans nedenleriyle, simetrik anahtar algoritması genellikle bu amaç için kullanılır. Ancak, birçok kullanıcının birbirine ileti gönderebildiği uygulama düzeyinde, sorun, iletinin yalnızca hedeflenen alıcısının iletinin şifresini çözebilmesini nasıl sağlayabileceğidir. Bir çözüm, birbirine ileti gönderen her kullanıcı çifti için farklı bir simetrik anahtar kullanmaktır. Ancak, özellikle kullanıcılar farklı kuruluşlara aitse, bu çözümün yönetilmesi zor ve zaman alıcı olabilir. Bu sorunu çözmenin standart bir yolu *dijital zarflama* olarak bilinir ve PKI teknolojisini kullanır.

Bir uygulama bir iletiyi kuyruğa koyduğunda, API ya da API geçişi çıkışı rasgele bir simetrik anahtar oluşturur ve iletideki uygulama verilerini şifrelemek için anahtarı kullanır. Çıkış, simetrik anahtarı hedeflenen alıcının genel anahtarıyla şifreler. Daha sonra, iletideki uygulama verilerini şifrelenmiş uygulama verileri ve şifrelenmiş simetrik anahtarla değiştirir. Bu şekilde, yalnızca istenen alıcı simetrik anahtarın şifresini çözebilir ve bu nedenle uygulama verilerinin şifresini çözebilir. Şifrelenmiş bir iletinin birden fazla olası hedef alıcısı varsa, çıkış, amaçlanan her alıcı için simetrik anahtarın bir kopyasını şifreleyebilir.

Uygulama verilerinin şifrelenmesi ve şifrelerinin çözülmesi için farklı algoritmalar kullanılabiliyorsa, çıkış, kullandığı algoritmanın adını içerebilir.

z/OS Veri kümesi şifrelemesiyle IBM MQ for z/OS üzerinde atıl durumdaki veriler için gizlilik

IBM MQ for z/OS , verileri etkin günlük veri kümelerine, arşiv günlüğü veri kümelerine, sayfa kümelerine, önyükleme şeridi veri kümelerine (BSDS) ve paylaşılan ileti veri kümelerine (SMDS) yazarak müşteri ve yapılandırma verilerini güçleyebilir.

z/OS , veri kümelerinin verimli, ilke tabanlı şifrelemesini sağlar. IBM MQ for z/OS , aşağıdakiler için z/OS veri kümesi şifrelemesini destekler:

- Etkin günlük veri kümeleri; nota bakın “1” sayfa 482
- Günlük veri kümelerini arşivle; bkz. not “2” sayfa 482
- Sayfa kümeleri; bkz. not “1” sayfa 482
- BSDS; bkz. not “2” sayfa 482
- CSQINP* veri kümeleri; bkz. not “2” sayfa 482
- SMDS; bkz. not “1” sayfa 482

Bu, tek bir z/OS kuyruk yöneticisinde kalan verilerin gizliliğini sağlar.

Notlar:

1. IBM MQ for z/OS 9.2.0' den etkin günlükler için z/OS veri kümesi şifrelemesi. sayfa kümeleri ve SMDS desteklenir.
2. Arşiv günlükleri, BSDS ve CSQINP* veri kümeleri için veri kümesi şifrelemesi tüm IBM MQ for z/OSsürümlerinde desteklenir.
3. IBM MQ Advanced Message Security , durmakta olan verileri korumak için alternatif bir mekanizma sağlar. Buna ek olarak AMS , bellekteki ve hareket halinde olan verileri de korur

z/OS veri kümesi şifrelemesiyle ilgili daha fazla bilgi için z/OS veri kümesi şifreleme geliştirmelerinin kullanılması başlıklı konuya bakın.

z/OS veri kümesi şifrelemesinin yapılandırması, IBM MQ for z/OSdenetiminin dışındadır. Şifreleme ayarları, veri kümesi oluşturulduğunda geçerli olur.

Bu, yeni bir veri kümesi şifreleme ilkesinin kullanılabilmesi için var olan veri kümelerinin yeniden oluşturulması gerektiği anlamına gelir.

IBM MQ for z/OS , şifrelenmiş ve şifrelenmemiş veri kümelerinin bir karışımıyla çalışabilir, ancak standart bir yapılandırma, kullanılan veri kümelerinin tümünü veya hiçbirini şifrelemez.

z/OS IBM MQ for z/OS veri kümesini şifreleme adımlarına genel bakış

Bir IBM MQ for z/OS veri kümesini nasıl şifrelediğinizi.

Başlamadan önce

z/OS veri kümesi şifrelemesini kuruluşunuzda doğru şekilde yapılandırdığınızdan emin olmanız gerekir. Bir kuyruk paylaşım grubunda veri kümesi şifrelemesi ayarlıyorsanız, veri paylaşımı için z/OS veri kümesi şifrelemesini yapılandırmanız gerekir.

Not: z/OS şifrelenmiş veri kümesi, genişletilmiş bir biçim veri kümesi olmalıdır.

Yordam

1. Veri kümesini şifrelemek için kullanılacak şifreleme anahtarını ve RACF içinde `key-label` ' yi ayarlayın.
2. RACF CSFKEYS sınıfında `key-label` için bir profil oluşturun.
3. Kuyruk yöneticisinin kullanıcı kimliğine ve şifrelenmiş verilere erişmesi gereken diğer kullanıcı kimliğine OKUMA erişimi verin.

   Bu, veri kümesine karşı yazdırma yardımcı programlarını çalıştırmak için kullanılan kullanıcı kimliklerini içerebilir. Örneğin, CSQUTIL SKOPI işlemini çalıştıran kullanıcının ilgili sayfa kümesinin şifresini çözmesi gerekir.
4. `key-label` şifrelemesini veri kümesi adıyla ilişkilendirin.

   Bunu, veri kümesi adı ya da üst düzey niteleyici için bir SMS veri sınıfı ya da RACF DFP kesimi kullanarak yapabilirsiniz.

   Veri kümesi ayrıldığında `key-label` öğesini veri kümesiyle de ilişkilendirebilirsiniz.
5. IDCAMS ALTER deyimini kullanarak var olan veri kümesini yeniden adlandırın.
6. Veri kümesini uygun özniteliklerle yeniden ayırın.
7. Yeniden adlandırılan veri kümesinin içeriğini IDCAMS REPRO kullanarak yeni veri kümesine kopyalayın.

   Veriler, veri kümesine kopyalama işlemiyle şifrelenir.
8. Şifrelenmesi gereken diğer veri kümeleri için “4” sayfa 482 - “6” sayfa 482 arasındaki adımları yineleyin.

z/OS Kuyruk yöneticisi etkin günlüklerinin nasıl şifreleneceğine ilişkin örnek

Aşağıdaki konular, var olan etkin günlüklerde veri kümesi şifrelemesini etkinleştirme işleminde size yol gösterir.

Not: Diğer veri kümeleri için işlem, etkin günlüklere benzer.

Bu örnekte:

- CSQ1 kuyruk yöneticisi QMCSQ1kullanıcısı altında çalıştırılır ve etkin günlük veri kümeleri CSQ1.LOGS.LOGCOPY1.DS001, CSQ1.LOGS.LOGCOPY1.DS002, vb.
- Donanım ve yazılım ortamı, z/OS veri kümesi şifrelemesini kullanabilir
- RACF , SAF olarak kullanılır
- Kuyruk yöneticisi durduruldu

Yordamı aşağıdaki sırayla gerçekleştirin:

1. “Kuyruk yöneticisi için veri kümesi şifreleme anahtarının yapılandırılması” sayfa 483
2. “Günlük veri kümeleri için veri kümesi şifrelemesini yapılandırma” sayfa 484

z/OS Kuyruk yöneticisi için veri kümesi şifreleme anahtarının yapılandırılması

Bir kuyruk yöneticisi için veri kümesi şifreleme anahtarını yapılandırma.

Bu görev hakkında

Bu görev, “Günlük veri kümeleri için veri kümesi şifrelemesini yapılandırma” sayfa 484için bir önkoşuldur.

Yordam

1. z/OS anahtar oluşturucu yardımcı programını (KGUP)kullanarak AES-256 bit şifreleme DATA anahtarını bir etiketle ayarlayın; örneğin, CSQ1DSKY.
2. Aşağıdaki komutu vererek CSQ1DSKY şifreleme anahtarı için RACF CSFKEYS profilini tanımlayın:

   ```
   RDEFINE CSFKEYS CSQ1DSKY UACC(NONE)
   ```

3. Aşağıdaki komutu vererek, tanıtımın ICSF bölümünü, anahtarın korunan anahtar olarak kullanılmasına izin verecek şekilde yapılandırın:

   ```
   RALTER CSFKEYS CSQ1DSKY ICSF(SYMCPACFWRAP(YES) SYMCPACFRET(YES))
   ```

4. Aşağıdaki komutu vererek, kuyruk yöneticisinin tanıtım için QMCSQ1 READ erişimi vererek şifreleme anahtarını kullanmasına izin verin:

   ```
   PERMIT CSQ1DSKY CLASS(CSFKEYS) ID(QMCSQ1) ACCESS(READ)
   ```

   Şifrelenmiş veri kümesini okuması ya da yazması gereken herhangi bir yönetici kullanıcıya aynı erişimi verin.
5. Aşağıdaki komutu vererek CSFKEYS sınıfını yenileyin.

   ```
   SETROPTS RACLIST(CSFKEYS) REFRESH
   ```

Sonraki adım

“Günlük veri kümeleri için veri kümesi şifrelemesini yapılandırma” sayfa 484 içinde açıklandığı gibi veri kümeleri için veri kümesi şifrelemesini yapılandırma

z/OS Günlük veri kümeleri için veri kümesi şifrelemesini yapılandırma

Günlük veri kümelerinde şifrelemeyi yapılandırma.

Başlamadan önce

Aşağıdakileri okuduğunuzdan emin olun:

Bir IBM MQ for z/OS veri kümesini şifreleme adımlarına genel bakışve yordamı
“Kuyruk yöneticisi için veri kümesi şifreleme anahtarının yapılandırılması” sayfa 483

Bu görev hakkında

Bu yöntem, RACF soysal profilinin DFP bölümünü kullanır; böylece, profille eşleşen tüm yeni veri kümeleri için şifreleme anahtarını kullanabilirsiniz.

Diğer bir seçenek olarak, bir SMS veri sınıfı yapılandırabilir ve kullanabilirsiniz ya da anahtar etiketi veri kümesi ayrılırken doğrudan belirtilebilir.

Daha önce açıklandığı gibi, bu örnekte CSQ1 kuyruk yöneticisi QMCSQ1kullanıcısı altında çalıştırılır ve CSQ1.LOGS.LOGCOPY1.DS001, CSQ1.LOGS.LOGCOPY1.DS002, vb.

Yordam

1. Yoksa, aşağıdaki komutu vererek soysal tanıtımı yaratın:

   ```
   ADDSD 'CSQ1.LOGS.*' UACC(NONE)
   ```

2. Aşağıdaki komutu vererek, kuyruk yöneticisi kullanıcısının profilde erişimi değiştirmesine izin verin:

   ```
   PERMIT 'CSQ1.LOGS.*' ID(QMCSQ1) ACCESS(ALTER)
   ```

   Ayrıca, herhangi bir yönetici kullanıcı için gereken uygun erişime izin verin.

3. Aşağıdaki komutu vererek DFP bölümünü şifreleme anahtarı etiketiyle ekleyin:

   ```
   ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
   ```

   Not: Kuyruk yöneticisi için veri kümesi şifreleme anahtarını yapılandırmadakullandığınız şifreleme anahtarını kullanmanız gerekir.

4. Aşağıdaki komutu vererek genel veri kümesi profillerini yenileyin:

   ```
   SETROPTS GENERIC(DATASET) REFRESH
   ```

5. Her günlük veri kümesini yedek olarak yeniden adlandırın ve IDCAMS kullanarak verileri yeniden yaratın ve geri yükleyin. Aşağıdaki JCL parçası CSQ1.LOGS.LOGCOPY1.DS001:

   a) Veri kümesini yedek olarak yeniden adlandır

   ```
   //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) Veri kümesini yeniden tanımlayın.

   Yeni veri kümesi, RACF profili nedeniyle şifrelenecek.

   Not: + + EXTDCLASS + + veri kümesi için kullanmak istediğiniz genişletilmiş biçim veri sınıfının adıyla değiştirin.

   ```
   //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) Yedekten verileri yeniden oluşturulan veri kümesine kopyalayın.

Bu adım verileri şifreler:

```
//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)
```

Sonraki adım

Tüm etkin günlük veri kümeleri için “5” sayfa 484 numaralı adımı yineleyin.

Yalnızca tek bir şifreleme anahtarı gereklidir ve tüm veri kümeleri aynı anahtar etiketiyle ilişkilendirilebilir.

CSQ1kuyruk yöneticisini yeniden başlatın. Günlük veri kümelerinin şifrelendiğini doğrulamak için DISPLAY LOG komutunun çıkışını kullanın.

z/OS Bir kuyruk paylaşım grubundaki z/OS veri kümesi şifrelemesiyle ilgili dikkat edilecek noktalar

Bir kuyruk paylaşım grubundaki (QSG) her kuyruk yöneticisi, QSG 'deki diğer her kuyruk yöneticisinin günlüklerini, BSDS' yi ve paylaşılan ileti veri kümelerini (SMDS) okuyabilmelidir.

Bu, QSG 'nin bir üyesinin çalışabileceği her bir sistemin z/OS veri kümesi şifrelemesi gereksinimlerini karşılaması ve QSG' deki her bir kuyruk yöneticisine ilişkin veri kümelerini korumak için kullanılan tüm anahtar etiketlerinin ve şifreleme anahtarlarının her bir sistemde kullanılabilir olması gerektiği anlamına gelir.

IBM MQ for z/OS 9.1.4 öncesinde bir kuyruk yöneticisi şifrelenmiş bir etkin günlük veri kümesine erişemez.

IBM MQ for z/OS 9.1.5 öncesinde bir kuyruk yöneticisi şifrelenmiş bir SMDS ' ye erişemez.

z/OS veri kümesi şifrelemesini kullanmadan önce, QSG 'deki tüm kuyruk yöneticilerini en az IBM MQ for z/OS 9.1.5' a geçirmeniz gerekir.

QSG 'deki bir kuyruk yöneticisi herhangi bir şifrelenmiş etkin günlük veri kümesiyle başlatılırsa ve QSG' deki başka bir kuyruk yöneticisi başlatıldıysa, ancak en son şifrelenmiş etkin günlükleri destekleyen bir IBM MQ for z/OS sürümüyle başlatılmamışsa, şifrelenmiş etkin günlüğü olan kuyruk yöneticisi olağandışı sona erer 5C6-00F50033.

Bir QSG 'yi şifrelenmiş etkin günlükleri ve SMDS' yi tam bir kesinti olmadan kullanmak üzere aşağıdaki işlemleri yaparak dönüştürebilirsiniz:

1. Her kuyruk yöneticisi en az IBM MQ for z/OS 9.1.5 ' a geçiriliyor.
2. Her kuyruk yöneticisi için etkin günlükleri şifrelenmiş veri kümelerine dönüştürme. Bu, kuyruk yöneticisinin kapatılmasını ve yeniden başlatılmasını gerektirir.

   Aynı zamanda, şifrelenmiş veri kümeleri için de sayfa kümeleri ve arşiv günlükleri etkinleştirilebilir, ancak bu QSG geçişini etkilemez.

Her bir veri kümesini dönüştürme yordamı “Kuyruk yöneticisi etkin günlüklerinin nasıl şifreleneceğine ilişkin örnek” sayfa 483 içinde açıklanmıştır.

3. SMDS ' yi her bir CF yapısı için şifrelenmiş veri kümelerine dönüştürme:
   a. SMDS ' ye kuyruk yöneticisi erişimini askıya almak için RESET SMDS (*) ACCESS (DISABLED) CFSTRUCT (structure-name) komutunu verin.

      Bu süre içinde, SMDS ile ilişkili paylaşılan kuyruklardaki verilerin geçici olarak kullanılamadığını unutmayın.
   b. “Kuyruk yöneticisi etkin günlüklerinin nasıl şifreleneceğine ilişkin örnek” sayfa 483içinde açıklanan yordamı kullanarak SMDS ' yi şifrelenmiş veri kümelerine dönüştüren her veri kümesi.
   c. SMDS ' ye kuyruk yöneticisi erişimini sürdürmek için RESET SMDS (*) ACCESS (ENABLED) CFSTRUCT (structure-name) komutunu verin.

Uyarı: Günlükleri dönüştürmeden önce kuyruk yöneticisini düzgün bir şekilde kapatmanız gerekir ve etkin günlük veri kümeleri geçici olarak kullanılamayacağı için, dönüştürme sırasında bağlaşım olanağı yapısı kurtarma mümkün olmayabilir.

z/OS z/OS veri kümesi şifrelemesi kullanılırken geriye doğru geçişle ilgili önemli noktalar

Bir ya da daha fazla şifrelenmiş veri kümesi olan bir kuyruk yöneticisini geriye doğru geçirirken aşağıdakileri göz önünde bulundurmanız gerekir.

z/OS veri kümesi şifrelemesi aşağıdaki IBM MQ for z/OS veri kümelerinde desteklenir:

- Etkin günlük veri kümeleri
- Günlük veri kümelerini arşivle
- Sayfa kümeleri
- BSDS (BDS)
- KOBİ ' LER
- CSQINP* veri kümeleri

BSDS, arşiv günlüğü ya da CSINP* veri kümeleri için geriye dönük geçiş konuları yoktur.

Bununla birlikte,

- KOBİ ' LER
- Sayfa kümesi ve
- Etkin günlük

veri kümelerinin z/OS veri kümesi şifrelemesiyle birlikte kullanılması IBM MQ for z/OS 9.1.0ve daha önceki uzun süreli destek yayınlarında desteklenmez.

Geriye doğru geçişten önce, SMDS, sayfa kümesi ve etkin günlük veri kümeleri için tüm şifreleme ilkelerinin kaldırılması ve verilerin şifresinin çözülmesi gerekir. Bu işlem “Veri kümesi şifrelemesini veri kümesinden kaldırma” sayfa 486içinde açıklanmıştır.

Uyarı: Geri geçirilecek kuyruk yöneticisi bir kuyruk paylaşım grubunun (QSG) parçasıysa, önce “Kuyruk paylaşım grubu ile ilgili önemli noktalar” sayfa 488 bölümünü okuyun.

Veri kümesi şifrelemesini veri kümesinden kaldırma

Bu örnek, CSQ1.LOGS.LOGCOPY1.DS001. SMDS ve sayfa kümeleri için eşdeğer bir işlem kullanabilirsiniz.

Örnek şunları varsayar:

- RACF SAF ' tır
- Veri kümesini kullanan kuyruk yöneticisi durduruldu
- Şifreleme anahtarı etiketi, soysal RACF tanıtımı CSQ1.LOGS.*

Aşağıdaki yordamı gerçekleştirin:

1. Verileri veri kümesinden bir yedek veri kümesine kopyalayın.

   a. Bir şifreleme anahtarı etiketiyle ilişkilendirilmemiş bir yedek veri kümesi tanımlayın.

      Not: + + EXTDCLASS + + veri kümesi için kullanmak istediğiniz genişletilmiş biçim veri sınıfının adıyla değiştirin.

      ```
      //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. Verileri özgün veri kümesinden yedeğe kopyalayın. Bu adım, verilerin şifresini çözer.

      ```
      //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. Özgün veri kümesini sil

      ```
      //DELETE   EXEC PGM=IDCAMS,REGION=0M
      //SYSPRINT DD SYSOUT=*
      //SYSIN    DD *
        /*-------------------------------------------------------------*/
        /* DELETE ORIGINAL                                             */
        /*-------------------------------------------------------------*/
        DELETE ('CSQ1.LOGS.LOGCOPY1.DS001')
      /*
      ```

   d. Yedeği özgün veri kümesi adıyla yeniden adlandırın. Veriler şifresiz kalır

      ```
      //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. İsteğe bağlı olarak, CSQ1.LOGS.* genel tanıtım.
3. İsteğe bağlı olarak, CSQ1.LOGS.* soysal tanıtımın şifresi çözüldü, soysal tanıtımla ilişkili DATAKEY ' yi kaldırmak için aşağıdaki komutu verin:

   ```
   ALTDSD 'CSQ1.LOGS.*' DFP(RESOWNER(QMCSQ1) DATAKEY(CSQ1DSKY))
   ```

4. Aşağıdaki komutu vererek genel veri kümesi profillerini yenileyin:

   ```
   SETROPTS GENERIC(DATASET) REFRESH
   ```

5. Kuyruk yöneticisini yeniden başlatın.

6. Şifreleme anahtarı artık gerekmezse, anahtarı silin ve ilişkili RACF profilini CSFKEYS sınıfından silin.

Kuyruk paylaşım grubu ile ilgili önemli noktalar

Bir kuyruk paylaşım grubunun parçası olan bir kuyruk yöneticisi, veri kümesi şifrelemesini desteklemeyen bir IBM MQ for z/OS sürümüne geri geçirilecekse, QSG 'deki tüm kuyruk yöneticilerinin tüm etkin günlük veri kümelerinin ve SMDS' lerinin veri kümesi şifreleme ilkelerinin kaldırılması ve verilerinin şifresinin çözülmesi gerekir.

Bu, QSG 'nin tek bir üyesinin geriye doğru ya da QSG' nin tüm üyelerine ait olup olmadığına bakılmaksızın geçerlidir.

Aşağıdakiler aracılığıyla tam bir QSG kesintisi olmadan şifreleme ilkelerinin kaldırılmasını ve verilerin şifresinin çözülmesini elde edebilirsiniz:

1. “Veri kümesi şifrelemesini veri kümesinden kaldırma” sayfa 486içinde açıklanan işlemi kullanarak QSG ' deki her kuyruk yöneticisini kapatma, şifreleme ilkelerini kaldırma ve etkin günlüklerinden verilerin şifresini çözme.

   Kuyruk yöneticisinin geriye doğru geçişi yapılacaksa, sayfa kümesinin şifresi de şu anda çözülmelidir. Daha sonra kuyruk yöneticisini yeniden başlatın.
2. Şifreleme ilkelerinin kaldırılması ve her bir CF yapısının SMDS 'si için verilerin şifresinin çözülmesi:
   a. Komutun verilmesi

      ```
      RESET SMDS(*) ACCESS(DISABLED) CFSTRUCT(structure-name)
      ```

      SMDS ' ye kuyruk yöneticisi erişimini askıya almak için. Bu süre içinde, SMDS ile ilişkili paylaşılan kuyruklardaki veriler geçici olarak kullanılamaz.
   b. SMDS ' yi oluşturan her veri kümesi için “Veri kümesi şifrelemesini veri kümesinden kaldırma” sayfa 486 içindeki işlemi izleyin.
   c. Komutun verilmesi

      ```
      RESET SMDS(*) ACCESS(ENABLED) CFSTRUCT(structure-name)
      ```

      SMDS ' ye kuyruk yöneticisi erişimini sürdürmek için.

z/OS veri kümesi şifrelemesini, onu desteklemeyen bir kuyruk yöneticisiyle kullanma

Yanlışlıkla bir kuyruk yöneticisini veri kümesi şifrelemesini desteklemeyen bir IBM MQ for z/OS sürümüne geçirir ve şifreleme ilkelerini kaldırmayı unutursanız ve kuyruk yöneticisi veri kümesine erişmeye çalıştığında bir hata elde ettiğinizde verilerin şifresini çözersiniz.

Hata, veri kümesi tipine bağlıdır ve aşağıdaki çizelgede gösterilir.

Not: Bu hatalardan biri ya da birkaçı ortaya çıkarsa, etkilenen veri kümesi için “Veri kümesi şifrelemesini veri kümesinden kaldırma” sayfa 486 içinde açıklanan işlemleri izlemeniz gerekir. Bunlar, IBM MQ for z/OSsürümü değiştirilmeden gerçekleştirilebilir.

Veri kümesi	Kuyruk yöneticisi z/OS veri kümesi şifrelemesini desteklemiyorsa hata oluştu
Sayfa kümesi 0	Kuyruk yöneticisi başlangıcında 5C6-00C91400 olağandışı bitti
Sayfa kümeleri 1-99	MQRC 2193 Sayfa kümesine erişilirken "Sayfa kümesi hatası"; örneğin, MQPUT
Etkin günlük	Kuyruk yöneticisi başlangıcında 5C6-00E80084 olağandışı bitti
KOBİ ' LER	IEC161I-122 iletisi "Veri kümesinin KEYLABEL değeri var, ancak kullanıcı uygulamanın şifrelemeyi işleyebileceğini belirtmedi".

Veri kümesi	Kuyruk yöneticisi z/OS veri kümesi şifrelemesini desteklemiyorsa hata oluştu
	SMDS, AVAIL olarak işaretlendi (HATA).

İletilerin veri bütünlüğü

Veri bütünlüğünü korumak için, iletileriniz için ileti özetleri ya da dijital imzalar sağlamak üzere çeşitli kullanıcı çıkış programı türlerini kullanabilirsiniz.

Veri bütünlüğü

İletilerde veri bütünlüğünü uygulama

TLS kullandığınızda, CipherSpec seçeneğiniz işletmede veri bütünlüğü düzeyini belirler. IBM MQ Advanced Message Service (AMS) olanağını kullanıyorsanız, benzersiz bir iletinin bütünlüğünü belirtebilirsiniz.

İleti çıkışlarında veri bütünlüğünün uygulanması

Bir ileti, bir kanalın gönderme sonundaki bir ileti çıkışı tarafından dijital olarak imzalanabilir. Daha sonra dijital imza, iletinin kasıtlı olarak değiştirilip değiştirilmediğini saptamak için bir kanalın alıcı ucundaki bir ileti çıkışı tarafından kontrol edilebilir.

Dijital imza yerine ileti özeti kullanılarak bazı koruma sağlanabilir. Bir mesaj özeti gündelik veya ayrım gözetmeyen kurcalamaya karşı etkili olabilir, ancak daha bilgili bireyin mesajı değiştirmesini veya değiştirmesini ve bunun için tamamen yeni bir özet oluşturmasını engellemez. Bu özellikle, ileti özetini oluşturmak için kullanılan algoritma iyi bilinen bir algoritmaysa doğrudur.

Gönderme ve alma çıkışlarında veri bütünlüğünün uygulanması

Bir ileti kanalında, bir ileti çıkışının tüm bir iletiye erişimi olduğundan, ileti çıkışları bu hizmeti sağlamak için daha uygundur. Bir MQI kanalında, MQI çağrılarına ilişkin değiştirgeler korunması gereken uygulama verilerini içerebilir ve yalnızca gönderme ve alma çıkışları bu korumayı sağlayabilir.

API çıkışında ya da API-geçiş çıkışında veri bütünlüğünü uygulama

Bir ileti, gönderen uygulama tarafından konduğunda bir API ya da API geçiş çıkışı tarafından dijital olarak imzalanabilir. Dijital imza, iletiyi alan uygulama tarafından alındığında iletinin kasıtlı olarak değiştirilip değiştirilmediğini saptamak için ikinci bir çıkış tarafından kontrol edilebilir.

Dijital imza yerine ileti özeti kullanılarak bazı koruma sağlanabilir. Bir mesaj özeti gündelik veya ayrım gözetmeyen kurcalamaya karşı etkili olabilir, ancak daha bilgili bireyin mesajı değiştirmesini veya değiştirmesini ve bunun için tamamen yeni bir özet oluşturmasını engellemez. Bu özellikle, ileti özetini oluşturmak için kullanılan algoritma iyi bilinen bir algoritmaysa doğrudur.

Daha fazla bilgi

Veri bütünlüğünün sağlanmasına ilişkin daha fazla bilgi için “CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432 başlıklı bölüme bakın.

İlgili görevler

TLS kullanarak iki kuyruk yöneticisinin bağlanması

İstemcinin kuyruk yöneticisine güvenli bir şekilde bağlanması

Denetleme

Olay iletilerini kullanarak güvenlik izinsiz girişlerini ya da izinsiz giriş girişlerini denetleyebilirsiniz. IBM MQ Explorerkomutunu kullanarak sisteminizin güvenliğini de denetleyebilirsiniz.

Kuyruk yöneticisine bağlanma ya da kuyruğa ileti koyma gibi yetkisiz işlemleri gerçekleştirme girişimlerini saptamak için, kuyruk yöneticileriniz tarafından üretilen olay iletilerini, özellikle de yetki olayı iletilerini

inceleyin. Kuyruk yöneticisi olay iletileriyle ilgili ek bilgi için Kuyruk yöneticisi olaylarıkonusuna bakın ve genel olarak olay izlemeyle ilgili ek bilgi için Olay izlemebaşlıklı konuya bakın.

Kümelerin güvenliğini sağlama

Kuyruk yöneticilerinin kümelere katılmasını ya da küme kuyruklarına ileti koymasını yetkilendirin ya da önleyin. Bir kuyruk yöneticisini kümeden ayrılmaya zorlayın. TLS ' yi kümeler için yapılandırırken dikkat edilmesi gereken bazı ek noktaları göz önünde bulundurun.

Yetkisiz kuyruk yöneticilerinin ileti göndermesini durdurma

Yetkisiz kuyruk yöneticilerinin kanal güvenliği çıkışını kullanarak kuyruk yöneticinize ileti göndermesini önleyin.

Başlamadan önce

Kümelemenin güvenlik çıkışlarının çalışma şekli üzerinde bir etkisi yoktur. Bir kuyruk yöneticisine erişimi, dağıtılmış bir kuyruğa alma ortamındaki gibi sınırlayabilirsiniz.

Bu görev hakkında

Seçilen kuyruk yöneticilerinin kuyruk yöneticinize ileti göndermesini engelle:

Yordam

1. CLUSRCVR kanal tanımında bir kanal güvenliği çıkış programı tanımlayın.
2. Küme-alıcı kanalınızda ileti göndermeye çalışan kuyruk yöneticilerinin kimliğini doğrulayan bir program yazın ve yetkili değillerse bu kullanıcılara erişimi reddedin.

Sonraki adım

MCA başlatma ve sonlandırma sırasında kanal güvenliği çıkış programları çağrılır.

Kuyruklarınıza ileti yerleştirerek yetkisiz kuyruk yöneticilerinin durdurulması

İletileri kuyruklarınıza koyacak yetkisiz kuyruk yöneticilerini durdurmak için küme-alıcı kanalındaki kanal koyma yetkisi özniteliğini kullanın. z/OSüzerinde RACF ya da diğer altyapılarda OAM kullanarak iletideki kullanıcı kimliğini denetleyerek uzak bir kuyruk yöneticisine yetki verin.

Bu görev hakkında

Kuyruklara erişimi denetlemek için bir platformun güvenlik olanaklarını ve IBM MQ içindeki erişim denetimi mekanizmasını kullanın.

Yordam

1. Belirli kuyruk yöneticilerinin iletileri kuyruğa koymalarını önlemek için altyapınızda bulunan güvenlik olanaklarını kullanın.

   Örneğin:

   - RACF ya da IBM MQ for z/OS üzerinde diğer dış güvenlik yöneticileri
   - Diğer altyapılarda nesne yetkisi yöneticisi (OAM).
2. CLUSRCVR kanal tanımlamasında PUTAUTözniteliğini kullanın.

   PUTAUT özniteliği, bir iletiyi kuyruğa koymak için yetki oluşturmak üzere hangi kullanıcı tanıtıcılarının kullanılacağını belirtmenizi sağlar.

PUTAUT özniteliğindeki seçenekler şunlardır:

DEF
Varsayılan kullanıcı kimliğini kullanın. z/OSişletim sistemlerinde, denetim hem ağdan alınan kullanıcı kimliğinin hem de MCAUSER' den türetilen kullanıcı kimliğinin kullanılmasını içerebilir.

CTX
İletiyle ilişkili bağlam bilgilerinde kullanıcı kimliğini kullanın. z/OS sistemlerinde denetim, ağdan alınan kullanıcı kimliğinin ya da MCAUSER' den türetilen kullanıcı kimliğinin ya da her ikisinin kullanılmasını içerebilir. Bağlantı güvenilir ve doğrulanmış ise bu seçeneği kullanın.

ONLYMCA (yalnızca z/OS)
DEF için, ancak ağdan alınan kullanıcı kimliği kullanılmaz. Bağlantı güvenilir değilse bu seçeneği kullanın. Üzerinde yalnızca, MCAUSERiçin tanımlanan belirli bir işlem kümesine izin vermek istiyorsunuz.

ALTMCA (yalnızca z/OS)
CTXile ilgili olarak, ağdan alınan kullanıcı kimliği kullanılmaz.

Uzak küme kuyruklarına ileti konmasına yetki verilmesi

z/OS ' ta RACFkullanarak bir küme kuyruğuna koymak için yetki ayarlayın. Diğer altyapılarda, erişimi kuyruk yöneticilerine bağlanma ve bu kuyruk yöneticilerindeki kuyruklara koyma yetkisi verin.

Bu görev hakkında

Varsayılan davranış, SYSTEM.CLUSTER.TRANSMIT.QUEUEüzerinde erişim denetimi gerçekleştirilmesidir. Birden çok iletim kuyruğu kullanıyor olsanız da, bu davranışın geçerli olduğunu unutmayın.

Bu konuda açıklanan belirli bir davranış, yalnızca qm.ini dosyasındaki **ClusterQueueAccessControl** özniteliğini, Güvenlik kısmı konusunda açıklandığı gibi *RQMName*olacak şekilde yapılandırdığınızda ve kuyruk yöneticisini yeniden başlattığınızda geçerlidir.

Yordam

- z/OSiçin aşağıdaki komutları verin:

```
RDEFINE MQQUEUE QMgrName.QUEUE. QueueName UACC(NONE)
PERMIT QMgrName.QUEUE. QueueName CLASS(MQADMIN) ID(GroupName) ACCESS(UPDATE)
```

- AIX, Linux, and Windows sistemleri için aşağıdaki komutları verin:

```
setmqaut -m QMgrName -t qmgr -g GroupName +connect
setmqaut -m QMgrName -t queue -n QueueName -g GroupName -all +put
```

- IBM iiçin aşağıdaki komutları verin:

```
GRTMQMAUT OBJ(' QMgrName ') OBJTYPE(*MQM) USER(GroupName) AUT(*CONNECT)
GRTMQMAUT OBJ(' QueueName ') OBJTYPE(*Q) USER(GroupName) AUT(*PUT) MQMNAME(' QMgrName ')
```

Kullanıcı, iletileri yalnızca belirlenen küme kuyruğuna yerleştirebilir ve başka küme kuyruğu koyamaz.

Değişken adları aşağıdaki anlamlara sahiptir:

QMgrName
Kuyruk yöneticisinin adı. z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.

GroupName
Erişim verilecek grubun adı.

QueueName
Yetkileri değiştirilecek kuyruğun ya da soysal tanıtımın adı.

Sonraki adım

Bir iletiyi küme kuyruğuna koyarken bir yanıt kuyruğu belirlerseniz, kullanan uygulamanın yanıtı gönderme yetkisi olmalıdır. “Uzak küme kuyruğuna ileti koymak için yetki verilmesi” sayfa 410içindeki yönergeleri izleyerek bu yetkiyi ayarlayın.

İlgili kavramlar

qm.ini içindeki güvenlik kısmı

Kuyruk yöneticilerinin bir kümeye katılmasını önleme

Bir düzenbaz kuyruk yöneticisi bir kümeye katılırsa, kümenin almasını istemediğiniz iletileri almasını önlemek zordur.

Yordam

Yalnızca belirli yetkili kuyruk yöneticilerinin bir kümeye katıldığından emin olmak istiyorsanız, üç teknik seçeneğiniz vardır:

- Kanal kimlik doğrulama kayıtlarını kullanarak, uzak IP adresine, uzak kuyruk yöneticisi adına ya da uzak sistem tarafından sağlanan TLS Ayırt Edici Adına dayalı olarak küme kanalı bağlantısını engelleyebilirsiniz.
- Yetkisiz kuyruk yöneticilerinin SYSTEM.CLUSTER.COMMAND.QUEUEdosyasına yazmalarını önlemek için bir çıkış programı yazın. SYSTEM.CLUSTER.COMMAND.QUEUE ' e erişimi, hiçbir kuyruk yöneticisi yazamayacak şekilde sınırlamayın ya da herhangi bir kuyruk yöneticisinin kümeye katılmasını engelleyin.
- CLUSRCVR kanal tanımında bir güvenlik çıkış programı.

Küme kanallarında güvenlik çıkışları

Küme kanallarında güvenlik çıkışları kullanılırken dikkat edilmesi gereken ek noktalar.

Bu görev hakkında

Bir kümeyi gönderen kanal ilk kez başlatıldığında, sistem yöneticisi tarafından el ile tanımlanan öznitelikleri kullanır. Kanal durdurulup yeniden başlatıldığında, ilgili küme alıcı kanal tanımından öznitelikleri alır. SecurityExit özniteliği de içinde olmak üzere, yeni özniteliklerle özgün küme gönderen kanal tanımlamasının üzerine yazılır.

Yordam

1. Hem küme gönderen ucunda hem de bir kanalın küme alıcı ucunda bir güvenlik çıkışı tanımlamanız gerekir.

   Güvenlik çıkışı adı küme-alıcı tanımından gönderilse de, ilk bağlantı bir güvenlik çıkışı el sıkışmasıyla yapılmalıdır.
2. Güvenlik çıkışındaki MQCXP yapısında PartnerName geçerliliğini denetleyin.

   Çıkış, yalnızca ortak kuyruk yöneticisi yetkilendirildiyse kanalın başlatılmasına izin vermelidir
3. Başlatılacak günlük nesnesi tanımlamasında güvenlik çıkışını tasarlayın.
4. Bunu gönderen tarafından başlatılan olarak tasarlarsanız, güvenlik çıkışı olmayan yetkisiz bir kuyruk yöneticisi kümeye katılabilir, çünkü güvenlik denetimi gerçekleştirilmez.

   Kanal durdurulup yeniden başlatılıncaya kadar, SCYEXIT adı küme alıcısı tanımlamasından ve tam güvenlik denetimlerinden gönderilebilir.
5. Kullanılmakta olan küme gönderen kanal tanımını görüntülemek için şu komutu kullanın:

```
DISPLAY CLUSQMGR( queue manager ) ALL
```

Komut, küme günlük nesnesi tanımlamasında gönderilen öznitelikleri görüntüler.

6. Özgün tanımlamayı görüntülemek için şu komutu kullanın:

```
DISPLAY CHANNEL( channel name ) ALL
```

7. Kuyruk yöneticileri farklı platformlardaysa, kümeyi gönderen kuyruk yöneticisinde bir kanal otomatik tanımlama çıkışı (CHADEXIT) tanımlamanız gerekebilir.

   SecurityExit özniteliğini hedef altyapı için uygun bir biçime ayarlamak için kanal otomatik tanımlama çıkışını kullanın.

8. Güvenlik çıkışını devreye alın ve yapılandırın.

   z/OS z/OS

   Güvenlik çıkışı yükleme modülü, kanal başlatıcı adres alanı yordamının CSQXLIB DD deyiminde belirtilen veri kümesinde olmalıdır.

   ALW AIX, Linux, and Windows sistemleri

   - Güvenlik çıkışı dinamik bağlantı kitaplığı, kanal tanımlamasının SCYEXIT özniteliğinde belirtilen yolda olmalıdır.
   - Kanal otomatik tanımlama çıkışı dinamik bağlantı kitaplığı, kuyruk yöneticisi tanımlamasının CHADEXIT özniteliğinde belirtilen yolda olmalıdır.

İstenmeyen kuyruk yöneticilerini kümeden ayrılmaya zorlama

İstenmeyen bir kuyruk yöneticisini, tam havuz kuyruğu yöneticisinde RESET CLUSTER komutunu vererek kümeden ayrılmaya zorlar.

Bu görev hakkında

İstenmeyen bir kuyruk yöneticisini kümeden ayrılmaya zorlayabilirsiniz. Örneğin, bir kuyruk yöneticisi silinir, ancak küme alıcı kanalları kümede tanımlanmaya devam eder. Toparlansan iyi olur.

Yalnızca tam havuz kuyruğu yöneticilerinin bir kümeden kuyruk yöneticisini çıkarma yetkisi vardır.

Not: RESET CLUSTER komutu kullanıldığında bir kuyruk yöneticisi kümeden zorla kaldırılmasına rağmen, RESET CLUSTER ' in tek başına kullanılması kuyruk yöneticisinin daha sonra kümeye yeniden katılmasını engellemez. Kuyruk yöneticisinin kümeye yeniden katılmamasını sağlamak için “Kuyruk yöneticilerinin bir kümeye katılmasını önleme” sayfa 492içinde açıklanan adımları izleyin.

OSLO Kümeden NORWAYkuyruk yöneticisini çıkarmak için aşağıdaki yordamı izleyin:

Yordam

1. Tam havuz kuyruğu yöneticisinde şu komutu verin:

```
RESET CLUSTER(NORWAY) QMNAME(OSLO) ACTION(FORCEREMOVE)
```

2. Diğer bir seçenek olarak, komutta QMNAME yerine QMID kullanın:

```
RESET CLUSTER(NORWAY) QMID(qmid) ACTION(FORCEREMOVE)
```

   Not: QMID bir dizedir; bu nedenle qmid değeri tek tırnak işareti ile çevrelenmelidir; örneğin, QMID('FR01_2019-07-15_14.42.42').

Sonuçlar

Zorlamalı olarak kaldırılan kuyruk yöneticisi değişmez; yerel küme tanımlamaları kümede olduğunu gösterir. Diğer tüm kuyruk yöneticilerindeki tanımlamalar kümede gösterilmez.

Kuyruk yöneticilerinin ileti almasını engelleme

Bir küme kuyruk yöneticisinin, alma yetkisi olmayan iletileri çıkış programlarını kullanarak almasını önleyebilirsiniz.

Bu görev hakkında

Bir kümenin üyesi olan bir kuyruk yöneticisinin kuyruk tanımlamasını durdurması zordur. Bir düzenbaz kuyruk yöneticisinin bir kümeye katılması ve kümedeki kuyruklardan birinin kendi eşgörünümünü tanımlaması tehlikesi vardır. Artık alma yetkisi olmayan iletileri alabilir. Bir kuyruk yöneticisinin ileti almasını önlemek için, yordamda belirtilen seçeneklerden birini kullanın.

Yordam

- Her bir küme gönderen kanalında bir kanal çıkış programı. Çıkış programı, iletilerin gönderileceği hedef kuyruk yöneticisinin uygunluğunu saptamak için bağlantı adını kullanır.
- İletilerin gönderileceği hedef kuyruğun ve kuyruk yöneticisinin uygunluğunu belirlemek için hedef kayıtları kullanan bir küme iş yükü çıkış programı.

SSL/TLS ve kümeler

Kümeler için TLS yapılandırılırken, bir CLUSRCVR kanal tanımlamasının otomatik olarak tanımlanan CLUSSDR kanalı olarak diğer kuyruk yöneticilerine yayıldığını unutmayın. Bir CLUSRCVR kanalı TLS kullanıyorsa, kanal aracılığıyla iletişim kuran tüm kuyruk yöneticilerinde TLS ' yi yapılandırmanız gerekir.

TLS hakkında daha fazla bilgi için bkz. “IBM MQ içinde TLS güvenlik iletişim kuralları” sayfa 24. Bu öneri genellikle küme kanalları için geçerlidir, ancak aşağıdakilere özel bir önem vermek isteyebilirsiniz:

Bir IBM MQ kümesinde belirli bir CLUSRCVR kanal tanımlaması, otomatik olarak tanımlanan CLUSSDR' a dönüştürüldüğü diğer birçok kuyruk yöneticisine sık sık yayılır. Daha sonra, CLUSRCVRkanalını başlatmak için otomatik olarak tanımlanan CLUSSDR kullanılır. CLUSRCVR TLS bağlanırlığı için yapılandırıldıysa, aşağıdaki noktalar geçerlidir:

- Bu CLUSRCVR ile iletişim kurmak isteyen tüm kuyruk yöneticilerinin TLS desteğine erişimi olmalıdır. Bu TLS yetkilendirmesi, kanal için CipherSpec ' i desteklemelidir.
- Otomatik olarak tanımlanan küme gönderen kanallarının yayıldığı farklı kuyruk yöneticilerinin her birinin farklı bir ayırt edici adı vardır. CLUSRCVR üzerinde ayırt edici ad eşdüzey denetimi kullanılacaksa, alınabilecek tüm ayırt edici adlar başarıyla eşleştirilecek şekilde ayarlanmalıdır.

  Örneğin, belirli bir CLUSRCVR' e bağlanacak küme gönderen kanallarını barındıracak tüm kuyruk yöneticilerinin ilişkili sertifikaları olduğunu varsayalım. Ayrıca, bu sertifikaların tümündeki ayırt edici adların ülkeyi İngiltere, kuruluşu IBM, kuruluş birimini IBM MQ Geliştirme olarak tanımladığını ve hepsinin DEVT.QMnnnbiçiminde ortak adları olduğunu varsayalım; burada nnn sayısal.

  Bu durumda CLUSRCVR üzerindeki SSLPEER değeri `C=UK, O=IBM, OU=IBM MQ Development, CN=DEVT.QM*` , gerekli tüm küme gönderen kanallarının başarıyla bağlanmasına izin verir, ancak istenmeyen küme gönderen kanallarının bağlanmasını önler.
- Özel CipherSpec dizgileri kullanılıyorsa, özel dizgi biçimlerine tüm platformlarda izin verilmediğini unutmayın. Buna bir örnek olarak, CipherSpec string RC4_SHA_US in IBM i üzerinde 05 değeri vardır, ancak AIX, Linux, and Windows sistemlerinde geçerli bir belirtim değildir. Bu nedenle, bir CLUSRCVRüzerinde özel SSLCIPH parametreleri kullanılırsa, sonuçta ortaya çıkan tüm otomatik tanımlı küme gönderen kanalları, temel TLS desteğinin bu CipherSpec ' i uyguladığı ve özel değerle belirtilebileceği platformlarda bulunmalıdır. Kümenizde anlaşılacak SSLCIPH parametresi için bir değer seçemezseniz, kullanılmakta olan platformların anlayacağı bir değere değiştirmek için bir kanal otomatik tanımlama çıkışına gereksinim duyarsınız. Olanaklı olduğu yerlerde metinli CipherSpec dizgilerini kullanın (örneğin, TLS_RSA_WITH_AES_128_CBC_SHA).

SSLCRLNL parametresi tek bir kuyruk yöneticisi için geçerlidir ve bir küme içindeki diğer kuyruk yöneticilerine yayılmaz.

Kümelenmiş kuyruk yöneticileri ve kanalları SSL/TLS ' ye yükseltme

CLUSSDR kanallarından önce tüm CLUSRCVR kanallarını değiştirerek küme kanallarını birer birer büyütün.

Başlamadan önce

Bir küme için CipherSpec seçeneğini etkileyebileceğinden, aşağıdaki noktaları göz önünde bulundurun:

- Bazı CipherSpecs tüm platformlarda kullanılamaz. Kümedeki tüm kuyruk yöneticileri tarafından desteklenen bir CipherSpec seçin.
- Bazı CipherSpecs , geçerli IBM MQ yayınında yeni olabilir ve daha eski yayınlarda desteklenmez. Farklı MQ yayınlarında çalışan kuyruk yöneticilerini içeren bir küme, yalnızca her yayın düzeyinde desteklenen CipherSpecs özelliğini kullanabilir.

  Bir küme içinde yeni bir CipherSpec kullanmak için, önce tüm küme kuyruğu yöneticilerini yürürlükteki yayına geçirmeniz gerekir.
- Bazı CipherSpecs , özellikle de Eliptik Eğri Şifrelemesi kullananlar için belirli bir sayısal sertifika kullanılmasını gerektirir.

Uyarı: Bir kümenin parçası olarak birleştirmek istediğiniz kuyruk yöneticilerindeki Elliptic Curve imzalı sertifikalarla RSA imzalı sertifikaların karışımı kullanılamaz.

Bir kümedeki kuyruk yöneticilerinin tümünün RSA imzalı sertifikaları kullanması ya da hepsinin her ikisinin bir karışımını değil, EC imzalı sertifikaları kullanması gerekir.

Ek bilgi için bkz. “IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu” sayfa 46 .

Bu düzeylerde değillerse, kümedeki tüm kuyruk yöneticilerini IBM MQ V8 ya da üstüne yükseltin. TLS ' nin her birinden çalışması için sertifikaları ve anahtarları dağıtın.

CipherSpecs (ANY_TLS13, ANY_TLS13_OR_HIGHER, ANY_TLS12, ANY_TLS12_OR_HIGHERvb.) diğer adlarından herhangi birini yükseltmek ya da kullanmak istiyorsanız, kümedeki tüm IBM MQ for Multiplatforms kuyruk yöneticilerini IBM MQ 9.1.4 ya da daha sonraki bir sürüme ve kümedeki tüm IBM MQ for z/OS kuyruk yöneticilerini IBM MQ for z/OS 9.2.0 ya da sonraki bir düzeye yükseltmeniz gerekir.

Bu görev hakkında

CLUSSDR kanallarından önce CLUSRCVR kanallarını değiştirin.

Yordam

1. CLUSRCVR kanallarını istediğiniz herhangi bir sırada TLS 'ye çevirin, bir CLUSRCVR ' yi aynı anda değiştirin ve bir sonrakini değiştirmeden önce değişikliklerin kümede akmasına izin verin.

   Önemli: Yürürlükteki kanala ilişkin değişiklikler kümeye dağıtılıncaya kadar ters yolu değiştirmediğinizden emin olun.
2. İsteğe bağlı: Tüm el ile CLUSSDR kanallarını TLS ' ye çevirir.

   REFRESH CLUSTER komutunu REPOS(YES) seçeneğiyle kullanmadığınız sürece, bu işlemin kümenin çalışması üzerinde herhangi bir etkisi olmaz.

   Not: Büyük kümeler için, **REFRESH CLUSTER** komutunun kullanımı, devam ederken kümede kesintiye neden olabilir ve bundan sonra küme nesneleri otomatik olarak ilgili tüm kuyruk yöneticilerine durum güncellemeleri gönderdiğinde, 27 günlük aralıklarla kümede kesintiye neden olabilir. Bkz. Büyük bir kümede yenilenmesi, kümenin performansını ve kullanılabilirliğini etkileyebilir.
3. Yeni güvenlik yapılandırmasının kümede yayıldığından emin olmak için DISPLAY CLUSQMGR komutunu kullanın.
4. TLS kullanmak için kanalları yeniden başlatın ve REFRESH SECURITY (SSL) komutunu çalıştırın.

İlgili kavramlar

“CipherSpecs Özelliğinin Etkinleştirilmesi” sayfa 432

DEFINE CHANNEL ya da **ALTER CHANNEL** MQSC komutunda **SSLCIPH** değiştirgesini kullanarak CipherSpec ' i etkinleştirin.

“IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu” sayfa 46
Bu konuda, IBM MQiçinde CipherSpecs ile dijital sertifikalar arasındaki ilişkiyi özetleyerek güvenlik ilkeniz için uygun CipherSpecs ve dijital sertifikaların nasıl seçileceğine ilişkin bilgiler sağlanır.

İlgili bilgiler
Kümeleme: REFRESH CLUSTER en iyi uygulamalarını kullanma

Kümelenmiş kuyruk yöneticileri ve kanallarında SSL/TLS ' nin devre dışı bırakılması

TLS ' yi kapatmak için SSLCIPH parametresini ' ' olarak ayarlayın. Küme gönderen kanallarından önce tüm küme alıcı kanallarını değiştirerek küme kanallarında TLS ' yi tek tek devre dışı bırakın.

Bu görev hakkında

Bir defada bir küme alıcı kanalını değiştirin ve sonraki kanalı değiştirmeden önce değişikliklerin kümede akmasına izin verin.

Önemli: Yürürlükteki kanala ilişkin değişiklikler kümeye dağıtılıncaya kadar ters yolu değiştirmediğinizden emin olun.

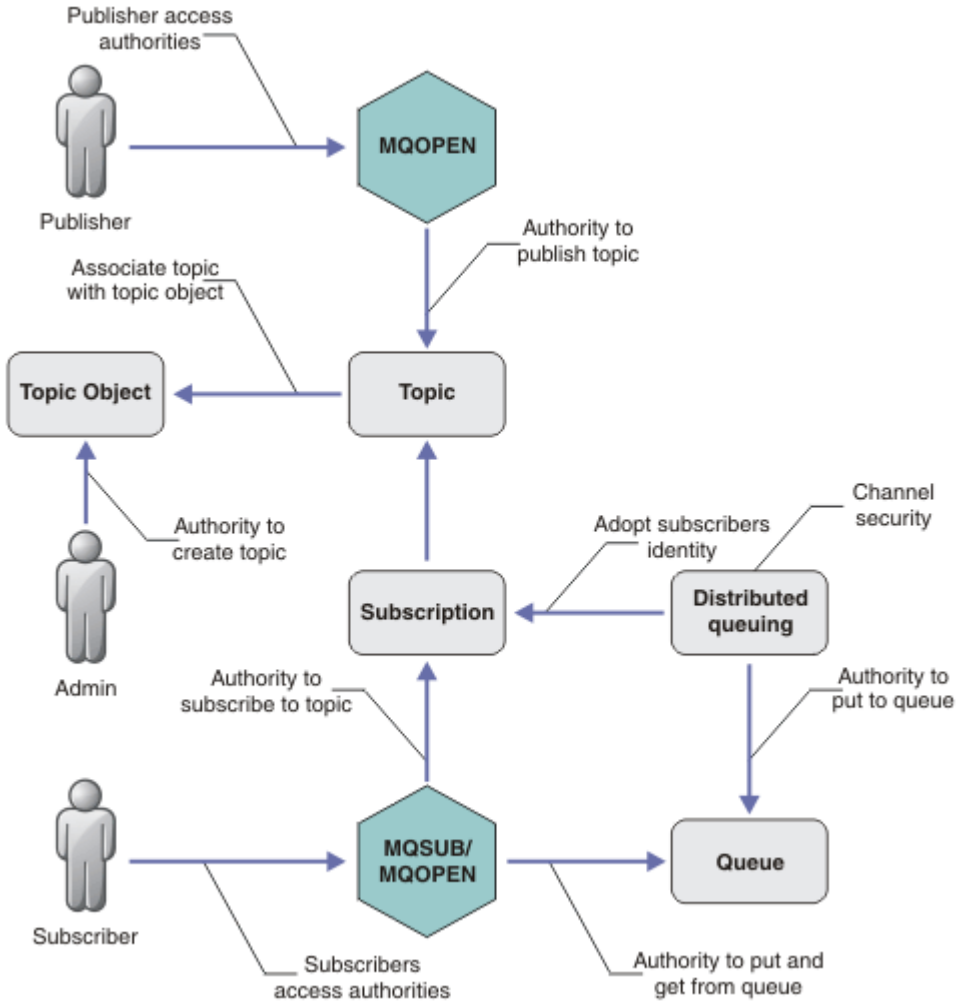
Yordam

1. SSLCIPH parametresinin değerini ' ' olarak ayarlayın; tek tırnak işareti içinde boş bir dizgi
   IBM i, ya da IBM i üzerinde *NONE .
   İstediğiniz herhangi bir sırada küme alıcı kanallarında TLS ' yi kapatabilirsiniz.
   Değişikliklerin TLS ' yi etkin olarak bırakdığınız kanallar üzerinde ters yönde akmaya devam ettiği unutulmamalıdır.
2. **DISPLAY CLUSQMGR(*)** ALLkomutunu kullanarak, yeni değerin diğer tüm kuyruk yöneticilerine yansıtıldığını denetleyin.
3. Tüm manuel küme gönderen kanallarında TLS ' yi kapatın.
   REFRESH CLUSTER komutunu REPOS （YES） seçeneğiyle kullanmadığınız sürece, bu işlemin kümenin çalışması üzerinde herhangi bir etkisi olmaz.
   Büyük kümeler için, **REFRESH CLUSTER** komutunun kullanımı, devam ederken kümede kesintiye neden olabilir ve bundan sonra küme nesneleri otomatik olarak ilgili tüm kuyruk yöneticilerine durum güncellemeleri gönderdiğinde, düzenli aralıklarla kümede kesintiye neden olabilir. Daha fazla bilgi için bkz. Büyük bir kümede yenilenme, kümenin performansını ve kullanılabilirliğini etkileyebilir .
4. Küme gönderen kanallarını durdurun ve yeniden başlatın.

Yayınlama/abone olma güvenliği

Yayınlama/abone olma sürecine dahil olan bileşenler ve etkileşimler, takip eden daha ayrıntılı açıklamalara ve örneklere giriş olarak tanımlanır.

Bir konuyu yayınlama ve bu konuya abone olma ile ilgili birçok bileşen vardır. Aralarındaki güvenlik ilişkilerinden bazıları Şekil 22 sayfa 497 içinde gösterilmiştir ve aşağıdaki örnekte açıklanmıştır.



Şekil 22. Çeşitli bileşenler arasındaki güvenlik ilişkilerini yayınla/abone ol

Konular

Konular, konu dizgileriyle tanımlanır ve genellikle ağaçlar halinde düzenlenir, bkz. Konu ağaçları. Konuya erişimi denetlemek için bir konuyu bir konu nesnesiyle ilişkilendirmeniz gerekir. “Konu güvenlik modeli” sayfa 499 , konu nesnelerini kullanarak konuların güvenliğini nasıl sağladığınızı açıklar.

Denetim konusu nesneleri

Bir konuya kimlerin erişebileceğini ve hangi amaçla **setmqaut** komutunu yönetimle ilgili konu nesnelerinin listesiyle kullanarak denetleyebilirsiniz. “Bir kullanıcıya bir konuya abone olması için erişim verme” sayfa 503 ve “Bir kullanıcıya bir konuyu yayınlaması için erişim verme” sayfa 510örneklerine bakın. z/OS z/OSüzerindeki konu nesnelerine erişimi denetlemek için Konu güvenliği için profillerbaşlıklı konuya bakın.

Abonelikler

Yayınların konu dizgileriyle eşleşecek genel arama karakterleri içerebilen bir konu dizgisi sağlayan bir abonelik oluşturarak bir ya da daha fazla konuya abone olun. Daha fazla ayrıntı için bkz:

Konu nesnesi kullanarak abone ol

“Konu nesnesi adını kullanarak abone olma” sayfa 500

Konu kullanarak abone ol

“Konu düğümünün var olmadığı bir konu dizesini kullanarak abone olma” sayfa 501

Genel arama karakterleriyle bir konuyu kullanarak abone ol

"Genel arama karakterleri içeren bir konu dizgisi kullanarak abone olma" sayfa 501

Abonelik, abonenin kimliğine ve yayınların yerleştirileceği hedef kuyruğun kimliğine ilişkin bilgileri içerir. Ayrıca, yayının hedef kuyruğa nasıl yerleştirileceğine ilişkin bilgileri de içerir.

Hangi abonelerin belirli konulara abone olma yetkisine sahip olduğunu tanımlamanın yanı sıra, abonelikleri tek bir abone tarafından kullanılmasıyla sınırlandırabilirsiniz. Yayınlar hedef kuyruğa yerleştirildiğinde, kuyruk yöneticisi tarafından aboneye ilişkin hangi bilgilerin kullanılacağını da denetleyebilirsiniz. Bkz. "Abonelik güvenliği" sayfa 516.

Kuyruklar

Hedef kuyruk, güvenli kılmak için önemli bir kuyruktur. Abonenin yereldir ve abonelikle eşleşen yayınlar üzerine yerleştirilir. Hedef kuyruğa iki perspektiften erişmeyi göz önünde bulundurmanız gerekir:

1. Hedef kuyruğa bir yayın konuyor.
2. Yayın hedef kuyruktan alınıyor.

Kuyruk yöneticisi, abone tarafından sağlanan bir kimliği kullanarak bir yayını hedef kuyruğa koyar. Abone ya da yayınları alma görevi için yetkilendirilen bir program, iletileri kuyruktan alır. Bkz. "Hedef kuyruklar için yetki" sayfa 501.

Konu nesnesi diğer adı yok, ancak konu nesnesinin diğer adı olarak bir diğer ad kuyruğu kullanabilirsiniz. Yayınlama ya da abone olma için konuyu kullanma yetkisini denetleyenin yanı sıra bunu yaparsanız, kuyruk yöneticisi kuyruğu kullanma yetkisini denetler.

"Kuyruk yöneticileri arasında yayınlama/abone olma güvenliği" sayfa 517

Bir konuyu yayınlama ya da bir konuya abone olma izniniz, yerel kimlikler ve yetkiler kullanılarak yerel kuyruk yöneticisinde denetlenir. Yetkilendirme, konunun tanımlanıp tanımlanmadığına ya da nerede tanımlandığına bağlı değildir. Sonuç olarak, kümelenmiş konular kullanıldığında kümelenmiş bir kümedeki her kuyruk yöneticisinde konu yetkilendirmesi gerçekleştirmeniz gerekir.

Not: Konuların güvenlik modeli, kuyruklara ilişkin güvenlik modelinden farklıdır. Her kümelenmiş kuyruk için yerel olarak bir kuyruk diğer adı tanımlayarak kuyruklar için aynı sonucu elde edebilirsiniz.

Kuyruk yöneticileri bir kümedeki abonelikleri değiştirir. Çoğu IBM MQ küme yapılandırmasında kanallar, kanal işleminin yetkisini kullanarak iletileri hedef kuyruklara yerleştirmek için PUTAUT=DEF ile yapılandırılır. Kanal yapılanışını, abone olan kullanıcının bir aboneliği kümedeki başka bir kuyruk yöneticisine yayma yetkisine sahip olmasını gerektirecek şekilde PUTAUT=CTX kullanacak şekilde değiştirebilirsiniz.

"Kuyruk yöneticileri arasında yayınlama/abone olma güvenliği" sayfa 517 içinde, kimlerin abonelikleri kümedeki diğer sunuculara yaymasına izin verildiğini denetlemek için kanal tanımlarınızın nasıl değiştirileceği açıklanır.

Yetkilendirme

Konu nesnelerine, kuyruklar ve diğer nesneler gibi yetki uygulayabilirsiniz. Yalnızca konulara uygulayabileceğiniz `pub`, `sub`ve `resume` olmak üzere üç yetkilendirme işlemi vardır. Ayrıntılar Farklı nesne tipleri için yetkilerin belirlenmesibaşlıklı konuda açıklanmaktadır.

İşlev çağrıları

Yayınlama ve abone olma programlarında, kuyruğa alınan programlarda olduğu gibi, nesneler açıldığında, yaratıldığında, değiştirildiğinde ya da silindiğinde yetki denetimleri yapılır. Yayınları koymak ve almak için MQPUT ya da MQGET MQI çağrıları yapıldığında denetimler yapılmaz.

Bir konuyu yayınlamak için, yetki denetimlerini gerçekleştiren konu üzerinde bir MQOPEN işlemi gerçekleştirin. Yetki denetimi gerçekleştirmeyen MQPUT komutunu kullanarak konu tanıtıcısına ileti yayınlayın.

Bir konuya abone olmak için, genellikle aboneliği oluşturmak ya da sürdürmek ve yayınları almak üzere hedef kuyruğu açmak için bir MQSUB komutu gerçekleştirmeniz gerekir. Diğer bir seçenek olarak,

hedef kuyruğu açmak için ayrı bir MQOPEN gerçekleştirin ve aboneliği oluşturmak ya da sürdürmek için MQSUB işlemini gerçekleştirin.

Hangi çağrıyı kullanırsanız kullanın, kuyruk yöneticisi konuya abone olup olamadığınızı denetler ve sonuçtaki yayınları hedef kuyruktan alır. Hedef kuyruk yönetilmezse, kuyruk yöneticisinin yayınları hedef kuyruğa yerleştirebildiğine ilişkin yetki denetimleri de yapılır. Eşleşen bir abonelikten benimsediği kimliği kullanır. Kuyruk yöneticisinin, yayınları her zaman yönetilen hedef kuyruklarına yerleştirebileceği varsayılır.

Roller

Kullanıcılar, yayınlama/abone olma uygulamalarını çalıştırmada dört rolde yer almaktadır:

1. Yayıncı
2. Abone
3. Konu yöneticisi
4. IBM MQ Yönetici-grubun üyesi mqm

Yayınlama, abone olma ve konu yönetimi rolleriyle ilgili uygun yetkilere sahip grupları tanımlayın. Daha sonra bu gruplara, belirli yayınlama ve abone olma görevlerini gerçekleştirme yetkisi veren birincil kullanıcılar atayabilirsiniz.

Ayrıca, yayınları ve abonelikleri taşımaktan sorumlu kuyrukların ve kanalların yöneticisine yönetim işlemleri yetkilerini de genişletmeniz gerekir.

Konu güvenlik modeli

Yalnızca tanımlı konu nesnelerinin ilişkili güvenlik öznitelikleri olabilir. Konu nesnelerine ilişkin açıklamalar için Denetim konusu nesnelerikonusuna bakın. Güvenlik öznitelikleri, belirtilen bir kullanıcı kimliğinin ya da güvenlik grubunun her konu nesnesi üzerinde bir abone olma ya da yayınlama işlemi gerçekleştirmesine izin verilip verilmediğini belirtir.

Güvenlik öznitelikleri, konu ağacındaki uygun yönetim düğümüyle ilişkilendirilir. Bir abone olma ya da yayınlama işlemi sırasında belirli bir kullanıcı kimliği için yetki denetimi yapıldığında, verilen yetki, ilişkili konu ağacı düğümünün güvenlik özniteliklerine dayalıdır.

Güvenlik öznitelikleri, belirli bir işletim sistemi kullanıcı kimliğinin ya da güvenlik grubunun konu nesnesi üzerindeki yetkisini gösteren bir erişim denetimi listesidir.

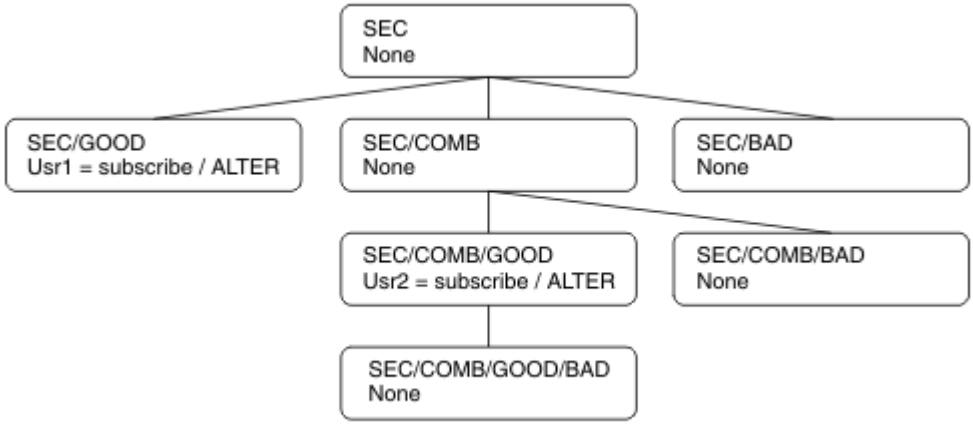
Aşağıdaki örnekte, konu nesnelerinin güvenlik öznitelikleriyle ya da gösterilen yetkililerle tanımlanmış olduğunu göz önünde bulundurun:

Çizelge 87. Örnek konu nesnesi yetkileri

Konu adı	Konu dizgisi	Yetkiler- z/OS değil	z/OS yetkileri
SECROOT	SEC	Yok	Yok
SECGOOD	SEC/GOOD	usr1+subscribe	ALTER HLQ.SUBSCRIBE.SECGOOD
SECBAD	SEC/BAD	Yok	Yok HLQ.SUBSCRIBE.SECBAD
SECCOMB	SEC/COMB	Yok	Yok HLQ.SUBSCRIBE.SECCOMB
SECCOMBB	SEC/COMB/ GOOD/BAD	Yok	Yok HLQ.SUBSCRIBE.SECCOMBB

Çizelge 87. Örnek konu nesnesi yetkileri (devamı var)			
Konu adı	**Konu dizgisi**	**Yetkiler- z/OS değil**	**z/OS yetkileri**
SECCOMBG	SEC/COMB/GOOD	usr2+subscribe	ALTER HLQ.SUBSCRIBE.SECCOMBG
SECCOMBN	SEC/COMB/BAD	Yok	Yok HLQ.SUBSCRIBE.SECCOMBN

Her düğümde ilişkili güvenlik özniteliklerine sahip konu ağacı aşağıdaki gibi gösterilebilir:



Listelenen örnekler aşağıdaki yetkileri verir:

- /SECağacının kök düğümünde, o düğümde kullanıcının yetkisi yoktur.
- usr1 nesneye abone olma yetkisi verildi /SEC/GOOD
- usr2 nesneye abone olma yetkisi verildi /SEC/COMB/GOOD

Konu nesnesi adını kullanarak abone olma

MQCHAR48 adını belirterek bir konu nesnesine abone olurken, konu ağacında ilgili düğüm bulunur. Düğümle ilişkili güvenlik öznitelikleri kullanıcının abone olma yetkisi olduğunu gösteriyorsa, erişim verilir.

Kullanıcıya erişim verilmezse, ağaçtaki üst düğüm kullanıcının üst düğüm düzeyinde abone olma yetkisine sahip olup olmadığını belirler. Bu durumda erişim verilir. Değilse, o düğümün üst düğümü dikkate alınır. Özyineleme, kullanıcıya abone olma yetkisi veren bir düğüm bulununcaya kadar devam eder. Yetki verilmeden kök düğüm dikkate alındığında özyineleme durur. İkinci durumda, erişim reddedilir.

Kısacası, yoldaki herhangi bir düğüm söz konusu kullanıcıya ya da uygulamaya abone olma yetkisi verirse, abonenin o düğümde ya da konu ağacında o düğümün altında herhangi bir yerde abone olmasına izin verilir.

Örnekteki kök düğüm: SEC.

Erişim denetim listesi kullanıcı kimliğinin kendisinin yetkisi olduğunu ya da kullanıcı kimliğinin üyesi olduğu bir işletim sistemi güvenlik grubunun yetkisi olduğunu gösterirse, kullanıcıya abone olma yetkisi verilir.

Yani, örneğin:

- usr1 , SEC/GOODkonu dizgisini kullanarak abone olmayı denerse, kullanıcı kimliğinin o konuyla ilişkili düğüme erişimi olduğu için aboneliğe izin verilir. Ancak, usr1 konu dizgisi SEC/COMB/GOOD kullanılarak abone olmaya çalışırsa, kullanıcı kimliğinin ilişkili düğüme erişimi olmadığı için aboneliğe izin verilmez.

- usr2 abone olmayı denerse, kullanıcı kimliğinin konuyla ilişkili düğüme erişimi olduğu için, SEC/COMB/GOOD konu dizgisi kullanılarak aboneliğe izin verilir. Ancak, usr2 SEC/GOOD ' a abone olmaya çalışırsa, kullanıcı kimliğinin ilişkili düğüme erişimi olmadığı için aboneliğe izin verilmez.
- usr2 , SEC/COMB/GOOD/BAD konu dizgisini kullanarak abone olmaya çalışırsa, kullanıcı kimliğinin üst düğüme erişimi olduğu için aboneliğe izin verilir SEC/COMB/GOOD.
- usr1 ya da usr2 , /SEC/COMB/BADkonu dizgisini kullanarak abone olmaya çalışırsa, konu düğümüyle ilişkilendirilmiş konu düğümüne ya da o konunun üst düğümlerine erişimleri olmadığı için bunlara izin verilmez.

Var olmayan bir konu nesnesinin adını belirten bir abone olma işlemi, MQRC_UNKNOWN_OBJECT_NAME hatasıyla sonuçlanır.

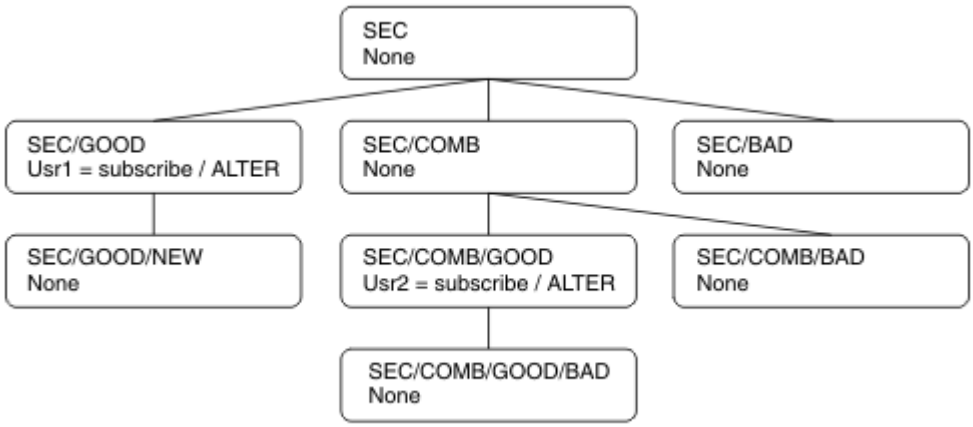
Konu düğümünün var olduğu bir konu dizgisi kullanılarak abone olunması

Bu davranış, konunun MQCHAR48 nesne adıyla belirtilmesiyle aynıdır.

Konu düğümünün var olmadığı bir konu dizesini kullanarak abone olma

Şu anda konu ağacında bulunmayan bir konu düğümünü gösteren bir konu dizgisi belirterek, bir uygulamaya abone olma olasılığını göz önünde bulundurun. Yetki denetimi, önceki bölümde belirtildiği şekilde gerçekleştirilir. Denetim, konu dizgisinin gösterdiği üst düğümle başlar. Yetki verilirse, konu ağacında konu dizgisini gösteren yeni bir düğüm yaratılır.

Örneğin, usr1 bir konuya abone olmayı dener SEC/GOOD/NEW. usr1 üst düğüme SEC/GOODerişimi olduğu için yetki verilir. Aşağıdaki çizgede gösterildiği gibi ağaçta yeni bir konu düğümü yaratılır. Yeni konu düğümü, doğrudan ilişkili güvenlik öznitelikleri olmayan bir konu nesnesi değil; öznitelikler üst öğesinden edinilir.



Genel arama karakterleri içeren bir konu dizgisi kullanarak abone olma

Genel arama karakteri içeren bir konu dizgisi kullanarak abone olma durumunu göz önünde bulundurun. Konu ağacında, konu dizgisinin tam olarak nitelenmiş bölümüyle eşleşen düğüm için yetki denetimi yapılır.

Bu nedenle, bir uygulama SEC/COMB/GOOD/*' a abone olursa, konu ağacında SEC/COMB/GOOD düğümdeki önceki iki kısımda özetlendiği gibi bir yetki denetimi gerçekleştirilir.

Benzer şekilde, bir uygulamanın SEC/COMB/*/GOOD' e abone olması gerekiyorsa, SEC/COMBdüğümünde bir yetki denetimi gerçekleştirilir.

Hedef kuyruklar için yetki

Bir konuya abone olurken, değiştirgelerden biri, yayınları almak üzere çıkış için açılmış bir kuyruğun hobj tanıtıcısıdır.

hobj belirtilmezse, ancak boşsa, aşağıdaki koşullar geçerse yönetilen bir kuyruk yaratılır:

- MQSO_MANAGED seçeneği belirtildi.
- Abonelik yok.
- Yaratma belirtildi.

hobj boşsa ve var olan bir aboneliği değiştirip sürdürürseniz, önceden sağlanan hedef kuyruk yönetilebilir ya da yönetilmez.

MQSUB isteğinde bulunan uygulamanın ya da kullanıcının, iletileri, sağladığı hedef kuyruğa koyma yetkisi olmalıdır; yürürlükteki yetkiyle, o kuyruğa konan iletileri yayınlama yetkisi olmalıdır. Yetki denetimi, kuyruk güvenliği denetimine ilişkin var olan kuralları izler.

Güvenlik denetimi, gerektiğinde diğer kullanıcı kimliği ve bağlam güvenliği denetimlerini içerir. Kimlik bağlamı alanlarından herhangi birini ayarlayabilmek için MQSO_CREATE ya da MQSO_ALTER seçeneğinin yanı sıra MQSO_SET_IDENTITY_CONTEXT seçeneğini de belirtmeniz gerekir. Bir MQSO_RESUME isteğinde Kimlik bağlamı alanlarından herhangi birini ayarlayamazsınız.

Hedef yönetilen bir kuyruksa, yönetilen hedef için güvenlik denetimi gerçekleştirilmez. Bir konuya abone olmanıza izin verilirse, yönetilen hedefleri kullanabileceğiniz varsayılır.

Konu düğümünün bulunduğu konu adı ya da konu dizgisi kullanılarak yayınlama

Yayınlamaya ilişkin güvenlik modeli, genel arama karakterleri dışında, abone olmak için kullanılan güvenlik modeliyle aynıdır. Yayınlar joker karakter içermez; bu nedenle dikkate alınacak joker karakter içeren bir konu dizesi yoktur.

Yayınlamak ve abone olmak için yetkiler farklıdır. Bir kullanıcı ya da grup, diğerini yapmak zorunda kalmadan birini yapma yetkisine sahip olabilir.

MQCHAR48 adını ya da konu dizgisini belirterek bir konu nesnesine yayınlama sırasında, konu ağacında ilgili düğüm bulunur. Konu düğümüyle ilişkili güvenlik öznitelikleri kullanıcının yayınlama yetkisi olduğunu gösteriyorsa, erişim verilir.

Erişim verilmezse, ağaçtaki üst düğüm kullanıcının o düzeyde yayınlama yetkisine sahip olup olmadığını belirler. Bu durumda erişim verilir. Değilse, kullanıcıya yayınlama yetkisi veren bir düğüm bulununcaya kadar özyineleme devam eder. Yetki verilmeden kök düğüm dikkate alındığında özyineleme durur. İkinci durumda, erişim reddedilir.

Kısacası, yoldaki herhangi bir düğüm o kullanıcıya ya da uygulamaya yayınlama yetkisi verirse, yayınlayıcının o düğümde ya da konu ağacında o düğümün altında herhangi bir yerde yayınlama yetkisine sahip olmasına izin verilir.

Konu düğümünün var olmadığı konu adı ya da konu dizgisi kullanılarak yayınlama

Abone olma işleminde olduğu gibi, bir uygulama yayınlandığında, konu ağacında var olmayan bir konu düğümünü gösteren bir konu dizgisi belirtilirken, yetki denetimi, konu dizgisi tarafından gösterilen düğümün üst öğesinden başlayarak gerçekleştirilir. Yetki verilirse, konu ağacında konu dizgisini gösteren yeni bir düğüm yaratılır.

Konu nesnesine çözülen bir diğer ad kuyruğunu kullanarak yayınlama

Bir konu nesnesine çözülen bir diğer ad kuyruğunu kullanarak yayınlarsanız, hem diğer ad kuyruğunda hem de çözümleneceği temel konuda güvenlik denetimi gerçekleştirilir.

Diğer ad kuyruğundaki güvenlik denetimi, kullanıcının o diğer ad kuyruğuna ileti koyma yetkisi olduğunu doğrular ve konu üzerindeki güvenlik denetimi, kullanıcının bu konuda yayınlama yapabildiğini doğrular. Bir diğer ad kuyruğu başka bir kuyruğa çözüldüğünde, temel kuyrukta denetimler yapılmaz. Konular ve kuyruklar için yetki denetimi farklı gerçekleştirilir.

Aboneliğin kapatılması

Aboneliği bu tanıtıcı altında yaratmadıysanız, MQCO_REMOVE_SUB seçeneğini kullanarak bir aboneliği kapatırsanız ek güvenlik denetimi olur.

İşlem, aboneliğin kaldırılmasına neden olduğundan, bunu yapmak için doğru yetkiye sahip olduğunuzdan emin olmak üzere bir güvenlik denetimi gerçekleştirilir. Konu düğümüyle ilişkili güvenlik öznitelikleri kullanıcının yetkisi olduğunu gösteriyorsa, erişim verilir. Değilse, kullanıcının aboneliği kapatma yetkisi olup olmadığını saptamak için ağaçtaki üst düğüm dikkate alınır. Özyineleme, yetki verilinceye ya da kök düğüme ulaşılıncaya kadar devam eder.

Abonelik tanımlama, değiştirme ve silme

MQSUB API isteği kullanmak yerine, bir abonelik yönetimsel olarak oluşturulduğunda abone olma güvenlik denetimleri gerçekleştirilmez. Yöneticiye komut aracılığıyla bu yetki önceden verildi.

Yayınların, abonelikle ilişkili hedef kuyruğa yerleştirilebilmesini sağlamak için güvenlik denetimleri gerçekleştirilir. Denetimler, MQSUB isteğiyle aynı şekilde gerçekleştirilir.

Bu güvenlik denetimleri için kullanılan kullanıcı kimliği, verilmekte olan komuta bağlıdır. **SUBUSER** parametresi belirtilirse, Çizelge 88 sayfa 503içinde gösterildiği gibi, denetime ilişkin gerçekleştirilme şeklini etkiler:

Çizelge 88. Komutlara ilişkin güvenlik denetimleri için kullanılan kullanıcı kimlikleri

Komut	SUBUSER belirlendi ve boş	SUBUSER belirlendi ve tamamlandı	SUBUSER belirtilmedi
	Yönetici kimliğini kullan		LIKE aboneliğinden kullanıcı kimliğini kullan
	Yönetici kimliğini kullan		SYSTEM.DEFAULT.SUB aboneliği-boşsa, denetimci kimliğini kullanın
	Yönetici kimliğini kullan		Var olan abonelikten kullanıcı kimliğini kullan

DELETE SUB komutu kullanılarak abonelikler silinirken yalnızca güvenlik denetimi gerçekleştirilir.

Örnek yayınlama/abone olma güvenlik ayarı

Bu bölümde, konular üzerinde, güvenlik denetiminin gerektiği gibi uygulanmasına olanak sağlayacak şekilde ayarlanmış erişim denetimine sahip bir senaryo açıklanmaktadır.

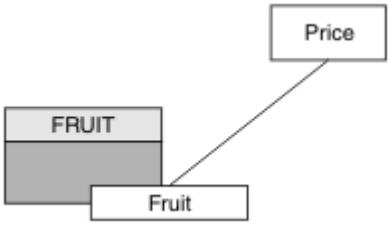
Bir kullanıcıya bir konuya abone olması için erişim verme

Bu konu, birden çok kullanıcı tarafından konulara nasıl erişim verileceğini size bildiren görevler listesindeki ilk konudur.

Bu görev hakkında

Bu görev, abonelik ya da yayın için herhangi bir denetim konusu nesnesi tanımlanmadığını ya da herhangi bir tanıtım tanımlanmadığını varsayar. Uygulamalar, var olanları sürdürmek yerine yeni abonelikler oluşturuyor ve bunu yalnızca konu dizgisini kullanarak yapıyor.

Bir uygulama, bir konu nesnesi, konu dizgisi ya da her ikisinin bir birleşimini sağlayarak abonelik yapabilir. Uygulamanın hangi yolu seçerse seçsin, sonuç, konu ağacında belirli bir noktada abonelik oluşturmaktır. Konu ağacındaki bu nokta bir denetim konusu nesnesiyle gösteriliyorsa, o konu nesnesinin adına dayalı olarak bir güvenlik profili denetlenir.



Şekil 23. Konu nesnesi erişimi örneği

Çizelge 89. Örnek konu nesnesi erişimi

Konu	Abone olma erişimi gerekli	Konu nesnesi
Fiyat	Kullanıcı yok	Yok
Fiyat/Meyve	USER1	MEYVE

Aşağıdaki gibi yeni bir konu nesnesi tanımlayın:

Yordam

1. `DEF TOPIC(FRUIT) TOPICSTR('Price/Fruit')`MQSC komutunu verin.
2. Şu şekilde erişim ver:
   - z/OS **z/OS** :

     Kullanıcıya `hlq.SUBSCRIBE.FRUIT` profili için erişim vererek `"Price/Fruit"` konusuna abone olmak için USER1 erişimi verin. Aşağıdaki RACF komutlarını kullanarak bunu yapın:

     ```
     RDEFINE MXTOPIC hlq.SUBSCRIBE.FRUIT UACC(NONE)
     PERMIT hlq.SUBSCRIBE.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
     ```

   - Diğer platformlar:

     Kullanıcıya FRUIT nesnesi için erişim vererek `"Price/Fruit"` konusuna abone olmak için USER1 erişimi verin. Bunu, altyapıya ilişkin yetkilendirme komutunu kullanarak yapın:

     ALW **AIX, Linux, and Windows sistemleri**

     ```
     setmqaut -t topic -n FRUIT -p USER1 +sub
     ```

     IBM i **IBM i**

     ```
     GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
     ```

Sonuçlar

USER1 konuya abone olmayı denediğinde `"Price/Fruit"` sonuç başarılı olur.

USER2 `"Price/Fruit"` konusuna abone olmayı denediğinde, sonuç bir `MQRC_NOT_AUTHORIZED` iletisiyle birlikte başarısız olur:

- z/OS z/OS' ta, konsolda görüntülenen ve denenen konu ağacında tam güvenlik yolunu gösteren aşağıdaki iletiler:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW Diğer platformlarda, aşağıdaki yetkilendirme olayı:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- IBM i IBMi üzerinde aşağıdaki yetkilendirme olayı:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

Bunun, tüm alanların değil, gördüklerinin bir resmi olduğunu unutmayın.

Bir kullanıcıya ağacın daha derinliklerinde bir konuya abone olması için erişim verme

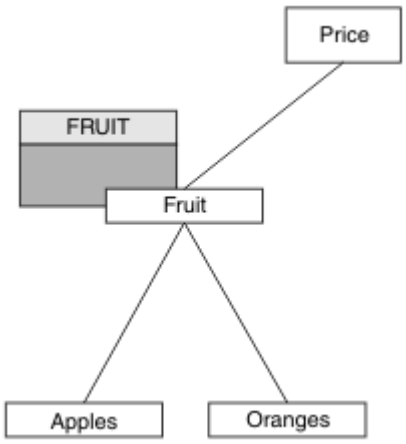
Bu konu, birden çok kullanıcı tarafından konulara nasıl erişim verileceğini size bildiren görevler listesinde yer alan ikinci konudur.

Başlamadan önce

Bu konuda, “Bir kullanıcıya bir konuya abone olması için erişim verme” sayfa 503içinde açıklanan kurulum kullanılır.

Bu görev hakkında

Konu ağacında uygulamanın aboneliği yaptığı nokta bir yönetim konusu nesnesiyle gösterilmezse, en yakın üst yönetim konusu nesnesi bulununcaya kadar ağacı yukarı taşıyın. Güvenlik profili, o konu nesnesinin adına dayalı olarak denetlenir.



Şekil 24. Konu ağacındaki bir konuya erişim izni verme örneği

Çizelge 90. Örnek konular ve konu nesneleri için erişim gereksinimleri		
Konu	**Abone olma erişimi gerekli**	**Konu nesnesi**
Fiyat	Kullanıcı yok	Yok
Fiyat/Meyve	USER1	MEYVE
Fiyat/Meyve/Elmalar	USER1	
Fiyat/Meyve/Portakal	USER1	

Önceki görevde USER1 , z/OS üzerindeki `hlq.SUBSCRIBE.FRUIT` profiline erişim ve diğer platformlardaki FRUIT profiline abone olma erişimi vererek `"Price/Fruit"` konusuna abone olma erişimi verildi. Bu tek profil, `"Price/Fruit/Apples"`, `"Price/Fruit/Oranges"` ve `"Price/Fruit/#"`ürününe abone olmak için USER1 erişimi de verir.

USER1 konuya abone olmayı denediğinde `"Price/Fruit/Apples"` sonuç başarılı olur.

USER2 `"Price/Fruit/Apples"` konusuna abone olmayı denediğinde, sonuç bir `MQRC_NOT_AUTHORIZED` iletisiyle birlikte başarısız olur:

- z/OS' ta, konsolda görüntülenen ve denenen konu ağacında tam güvenlik yolunu gösteren aşağıdaki iletiler:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- Diğer platformlarda, aşağıdaki yetkilendirme olayı:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Apples"
```

Aşağıdakileri unutmayın:

- z/OS üzerinde aldığınız iletiler, aynı konu nesneleri ve tanıtımları erişimi denetlediğinden, önceki görevde alınanlarla aynıdır.
- Diğer platformlarda aldığınız olay iletisi, önceki görevde alınana benzer, ancak gerçek konu dizgisi farklı.

Ağacın derinliklerinde yalnızca konuya abone olmak için başka bir kullanıcıya erişim ver

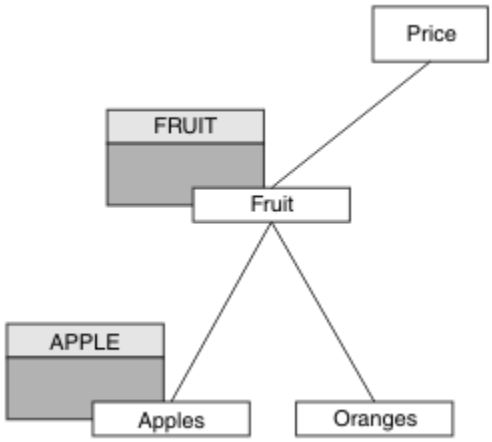
Bu konu, birden çok kullanıcı tarafından konulara abone olmak için erişim verilmesini bildiren görevler listesinin üçüncü konusudur.

Başlamadan önce

Bu konuda, “Bir kullanıcıya ağacın daha derinliklerinde bir konuya abone olması için erişim verme” sayfa 505içinde açıklanan kurulum kullanılır.

Bu görev hakkında

Önceki görevde USER2 , `"Price/Fruit/Apples"`konusuna erişimi reddedildi. Bu konuda, diğer konulara değil, bu konuya nasıl erişim verileceği açıklanır.



Şekil 25. Konu ağacındaki belirli konulara erişim verilmesi

Çizelge 91. Örnek konular ve konu nesneleri için erişim gereksinimleri		
Konu	**Abone olma erişimi gerekli**	**Konu nesnesi**
Fiyat	Kullanıcı yok	Yok
Fiyat/Meyve	USER1	MEYVE
Fiyat/Meyve/Elmalar	USER1 ve USER2	Elma
Fiyat/Meyve/Portakal	USER1	

Aşağıdaki gibi yeni bir konu nesnesi tanımlayın:

Yordam

1. `DEF TOPIC(APPLE) TOPICSTR('Price/Fruit/Apples')`MQSC komutunu verin.
2. Şu şekilde erişim ver:

- z/OS **z/OS** :

  In the previous task USER1 was granted access to subscribe to topic `"Price/Fruit/Apples"` by granting the user access to the `hlq.SUBSCRIBE.FRUIT` profile.

  Bu tek profil, `"Price/Fruit/Oranges"` `"Price/Fruit/#"` 'a abone olmak için USER1 erişimi de verdi ve bu erişim, yeni konu nesnesinin ve onunla ilişkili profillerin eklenmesiyle bile devam eder.

  Kullanıcıya `hlq.SUBSCRIBE.APPLE` profili için erişim vererek `"Price/Fruit/Apples"` konusuna abone olmak için USER2 erişimi verin. Aşağıdaki RACF komutlarını kullanarak bunu yapın:

  ```
  RDEFINE MXTOPIC hlq.SUBSCRIBE.APPLE UACC(NONE)
  PERMIT hlq.SUBSCRIBE.FRUIT APPLE(MXTOPIC) ID(USER2) ACCESS(ALTER)
  ```

- Diğer platformlar:

  In the previous task USER1 was granted access to subscribe to topic `"Price/Fruit/Apples"` by granting the user subscribe access to the `FRUIT` profile.

  Bu tek profil, `"Price/Fruit/Oranges"` ve `"Price/Fruit/#"`ürününe abone olmak için USER1 erişimi de verdi ve bu erişim, yeni konu nesnesinin ve bununla ilişkili profillerin eklenmesiyle birlikte kalır.

Kullanıcıya APPLE profili için abone olma erişimi vererek "Price/Fruit/Apples" konusuna abone olmak için USER2 erişimi verin. Bunu, altyapıya ilişkin yetkilendirme komutunu kullanarak yapın:

AIX, Linux, and Windows sistemleri

```
setmqaut -t topic -n APPLE -p USER2 +sub
```

IBM i

```
GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER2) AUT(*SUB)
```

Sonuçlar

z/OSsistemlerinde USER1 , "Price/Fruit/Apples" konusuna abone olmayı denediğinde hlq.SUBSCRIBE.APPLE profilindeki ilk güvenlik denetimi başarısız olur, ancak ağacın yukarı taşındığında hlq.SUBSCRIBE.FRUIT profili USER1 ' in abone olmasına izin verir; bu nedenle abonelik başarılı olur ve MQSUB çağrısına dönüş kodu gönderilmez. Ancak, ilk denetim için bir RACF ICH iletisi oluşturulur:

```
ICH408I USER(USER1   ) ...
  hlq.SUBSCRIBE.APPLE ...
```

USER2 konuya abone olmayı denediğinde "Price/Fruit/Apples" , güvenlik denetimi ilk profilde başarılı olduğu için sonuç başarılı olur.

USER2 "Price/Fruit/Oranges" konusuna abone olmayı denediğinde, sonuç bir MQRC_NOT_AUTHORIZED iletisiyle birlikte başarısız olur:

- z/OS z/OS' ta, konsolda görüntülenen ve denenen konu ağacında tam güvenlik yolunu gösteren aşağıdaki iletiler:

  ```
  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.FRUIT ...

  ICH408I USER(USER2   ) ...
    hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
  ```

- ALW AIX, Linux, and Windows platformlarında aşağıdaki yetkilendirme olayı:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

- IBM i IBMi üzerinde aşağıdaki yetkilendirme olayı:

  ```
  MQRC_NOT_AUTHORIZED
  ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
  UserIdentifier        USER2
  AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
  TopicString           "Price/Fruit/Oranges"
  ```

Bu ayarın dezavantajı, z/OSişletim sisteminde konsolda ek ICH iletileri almaktır. Konu ağacının güvenliğini farklı bir şekilde sağlarsanız bunu önleyebilirsiniz.

Ek iletileri önlemek için erişim denetimini değiştir

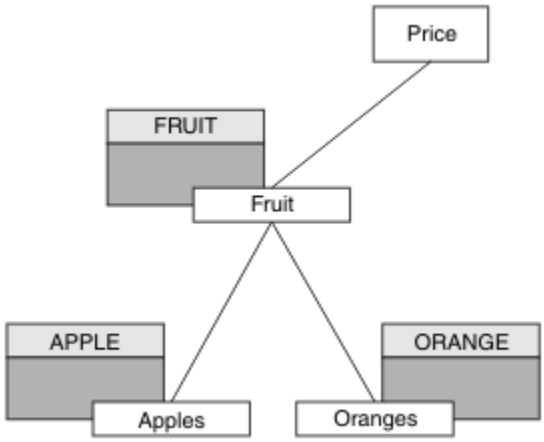
Bu konu, birden çok kullanıcı tarafından konulara abone olma izni verilmesini ve z/OSüzerinde ek RACF ICH408I iletilerinin gönderilmesini önlemek için size nasıl erişim verileceğini bildiren görevler listesinin dördüncü konusudur.

Başlamadan önce

Bu konu, ek hata iletilerini önlemek için “Ağacın derinliklerinde yalnızca konuya abone olmak için başka bir kullanıcıya erişim ver” sayfa 506 içinde açıklanan ayarları geliştirir.

Bu görev hakkında

Bu konuda, ağacın derinliklerindeki konulara nasıl erişim verileceği ve hiçbir kullanıcı gerekmediğinde ağacın aşağısındaki konuya erişimin nasıl kaldırılacağı açıklanır.



Şekil 26. Ek iletileri önlemek için erişim denetimi verme örneği.

Aşağıdaki gibi yeni bir konu nesnesi tanımlayın:

Yordam

1. `DEF TOPIC(ORANGE) TOPICSTR('Price/Fruit/Oranges')`MQSC komutunu verin.
2. Şu şekilde erişim ver:

   - z/OS **z/OS** :

     Yeni bir profil tanımlayın ve o profile ve var olan profillere erişim ekleyin. Aşağıdaki RACF komutlarını kullanarak bunu yapın:

     ```
     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)
     ```

   - Diğer platformlar:

     Platform için yetkilendirme komutlarını kullanarak eşdeğer erişimi ayarlayın:

     ALW **AIX, Linux, and Windows sistemleri**

     ```
     setmqaut -t topic -n ORANGE -p USER1 +sub
     setmqaut -t topic -n APPLE -p USER1 +sub
     ```

IBM i

```
GRTMQAUT OBJ(ORANGE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
GRTMQAUT OBJ(APPLE) OBJTYPE(*TOPIC) USER(USER1) AUT(*SUB)
```

Sonuçlar

z/OSüzerinde, USER1 `"Price/Fruit/Apples"` konusuna abone olmayı denediğinde `hlq.SUBSCRIBE.APPLE` profilindeki ilk güvenlik denetimi başarılı olur.

Benzer şekilde, USER2 konuya abone olmayı denediğinde `"Price/Fruit/Apples"` güvenlik denetimi ilk profili geçtiği için sonuç başarılı olur.

USER2 `"Price/Fruit/Oranges"` konusuna abone olmayı denediğinde, sonuç bir `MQRC_NOT_AUTHORIZED` iletisiyle birlikte başarısız olur:

- z/OS z/OS' ta, konsolda görüntülenen ve denenen konu ağacında tam güvenlik yolunu gösteren aşağıdaki iletiler:

```
ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.ORANGE ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.SUBSCRIBE.SYSTEM.BASE.TOPIC ...
```

- ALW Diğer platformlarda, aşağıdaki yetkilendirme olayı:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

- IBM i IBMi üzerinde aşağıdaki yetkilendirme olayı:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_SUB_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      ORANGE, FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit/Oranges"
```

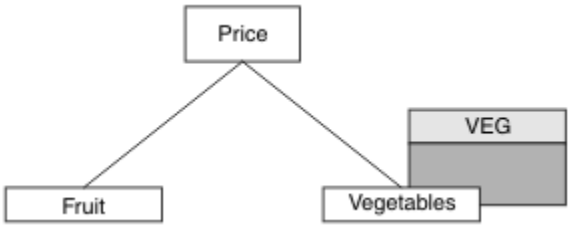
Bir kullanıcıya bir konuyu yayınlaması için erişim verme

Bu konu, birden çok kullanıcı tarafından yayınlama konularına nasıl erişim verileceğini size bildiren görevler listesindeki ilk konudur.

Bu görev hakkında

Bu görev, konu ağacının sağ tarafında hiçbir yönetim konusu nesnesinin bulunmadığını ve yayınlanmak üzere herhangi bir tanıtım tanımlanmadığını varsayar. Kullanılan varsayım, yayıncıların yalnızca konu dizgisini kullanmış olduklarıdır.

Bir uygulama, bir konu nesnesi, konu dizgisi ya da her ikisinin bir birleşimini sağlayarak bir konuya yayınlama yapabilir. Uygulamanın hangi yolu seçerse seçsin, sonuç, konu ağacında belirli bir noktada yayınlanmasıdır. Konu ağacındaki bu nokta bir denetim konusu nesnesiyle gösteriliyorsa, o konu nesnesinin adına dayalı olarak bir güvenlik profili denetlenir. Örneğin:



Şekil 27. Bir konuya yayınlama erişimi verilmesi

Çizelge 92. Örnek yayınlama erişimi gereksinimleri		
Konu	**Yayınlama erişimi gerekli**	**Konu nesnesi**
Fiyat	Kullanıcı yok	Yok
Fiyat/Sebze	USER1	VG

Aşağıdaki gibi yeni bir konu nesnesi tanımlayın:

Yordam

1. `DEF TOPIC(VEG) TOPICSTR('Price/Vegetables')` MQSC komutunu verin.
2. Şu şekilde erişim ver:

   - z/OS **z/OS** :

     Grant access to USER1 to publish to topic `"Price/Vegetables"` by granting the user access to the `hlq.PUBLISH.VEG` profile. Aşağıdaki RACF komutlarını kullanarak bunu yapın:

     ```
     RDEFINE MXTOPIC hlq.PUBLISH.VEG UACC(NONE)
     PERMIT hlq.PUBLISH.VEG CLASS(MXTOPIC) ID(USER1) ACCESS(UPDATE)
     ```

   - Diğer platformlar:

     Grant access to USER1 to publish to topic `"Price/Vegetables"` by granting the user access to the VEG profile. Bunu, altyapıya ilişkin yetkilendirme komutunu kullanarak yapın:

     ALW **AIX, Linux, and Windows sistemleri**

     ```
     setmqaut -t topic -n VEG -p USER1 +pub
     ```

     IBM i **IBM i**

     ```
     GRTMQAUT OBJ(VEG) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
     ```

Sonuçlar

USER1 Konu `"Price/Vegetables"` ' da yayınlama girişiminde bulunulduğunda sonuç başarılı olur; yani, MQOPEN çağrısı başarılı olur.

USER2 `"Price/Vegetables"` konusuna yayınlama girişiminde bulunduğunda MQOPEN çağrısı bir `MQRC_NOT_AUTHORIZED` iletisiyle başarısız olur:

- z/OS z/OS' ta, konsolda görüntülenen ve denenen konu ağacında tam güvenlik yolunu gösteren aşağıdaki iletiler:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- ALW Diğer platformlarda, aşağıdaki yetkilendirme olayı:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

- IBM i IBMi üzerinde aşağıdaki yetkilendirme olayı:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables"
```

Bunun, tüm alanların değil, gördüklerinin bir resmi olduğunu unutmayın.

Bir kullanıcıya, ağacın daha derinliklerinde bir konuya yayınlama izni verme

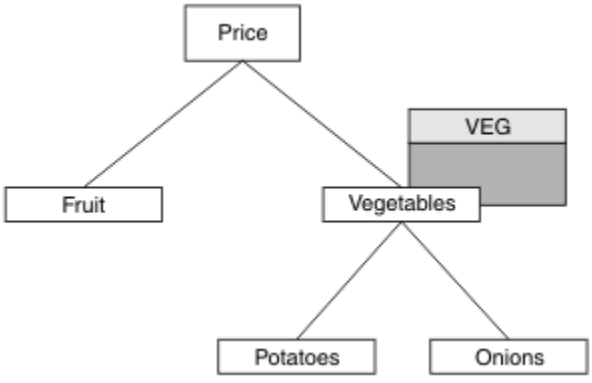
Bu konu, birden çok kullanıcı tarafından konulara yayınlama erişimi verilmesini bildiren görevler listesinde yer alan ikinci konudur.

Başlamadan önce

Bu konuda, “Bir kullanıcıya bir konuyu yayınlaması için erişim verme” sayfa 510içinde açıklanan kurulum kullanılır.

Bu görev hakkında

Konu ağacında uygulamanın yayınlandığı nokta bir yönetim konusu nesnesiyle gösterilmezse, en yakın üst yönetim konusu nesnesi bulununcaya kadar ağacı yukarı taşıyın. Güvenlik profili, o konu nesnesinin adına dayalı olarak denetlenir.



Şekil 28. Konu ağacı içindeki bir konuya yayınlama erişimi verilmesi

Çizelge 93. Örnek yayınlama erişimi gereksinimleri

Konu	Abone olma erişimi gerekli	Konu nesnesi
Fiyat	Kullanıcı yok	Yok
Fiyat/Sebze	USER1	VG
Fiyat/Sebze/Patates	USER1	
Fiyat/Sebze/Soğan	USER1	

Önceki görevde USER1 , z/OS üzerindeki `hlq.PUBLISH.VEG` profiline erişim ya da diğer platformlardaki VEG profiline yayınlama erişimi vererek `"Price/Vegetables/Potatoes"` konusunu yayınlama erişimi verildi. Bu tek profil ayrıca, `"Price/Vegetables/Onions"`. adresinde yayınlamak için USER1 erişimi de verir.

USER1 `"Price/Vegetables/Potatoes"` konusunda yayınlama girişiminde bulunduğunda sonuç başarılı olur; MQOPEN çağrısı başarılı olur.

USER2 `"Price/Vegetables/Potatoes"` konusuna abone olmayı denediğinde sonuç başarısız olur; yani, MQOPEN çağrısı bir `MQRC_NOT_AUTHORIZED` iletisiyle başarısız olur:

- z/OS' ta, konsolda görüntülenen ve denenen konu ağacında tam güvenlik yolunu gösteren aşağıdaki iletiler:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.VEG ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- Diğer platformlarda, aşağıdaki yetkilendirme olayı:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      VEG, SYSTEM.BASE.TOPIC
TopicString           "Price/Vegetables/Potatoes"
```

Aşağıdakileri unutmayın:

- z/OS üzerinde aldığınız iletiler, aynı konu nesneleri ve tanıtımları erişimi denetlediğinden, önceki görevde alınanlarla aynıdır.
- Diğer platformlarda aldığınız olay iletisi, önceki görevde alınana benzer, ancak gerçek konu dizgisi farklı.

Yayınlama ve abone olma için erişim ver

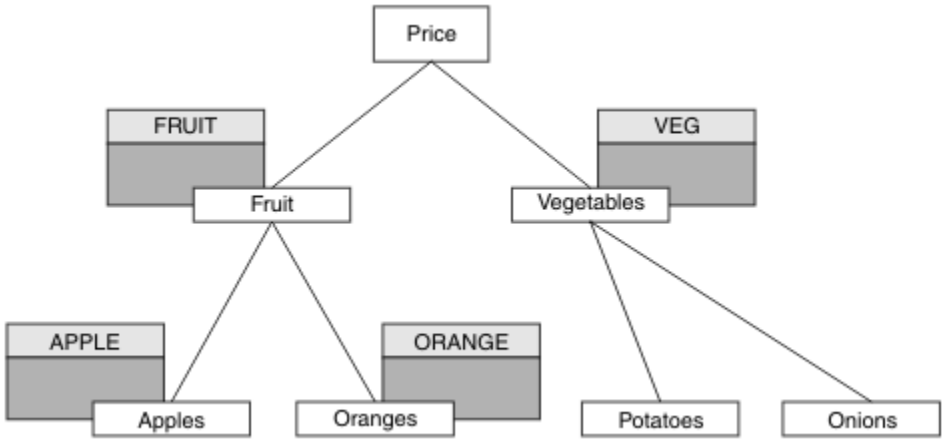
Bu konu, birden çok kullanıcı tarafından yayınlama ve konulara abone olma için nasıl erişim verileceğini size bildiren görevler listesinin sonuncusudur.

Başlamadan önce

Bu konuda, “Bir kullanıcıya, ağacın daha derinliklerinde bir konuya yayınlama izni verme” sayfa 512içinde açıklanan kurulum kullanılır.

Bu görev hakkında

Önceki bir görevde USER1 ' e `"Price/Fruit"`konusuna abone olma erişimi verildi. Bu konuda, söz konusu kullanıcıya o konuyu yayınlamak üzere nasıl erişim verileceği anlatılıyor.



Şekil 29. Yayınlama ve abone olma için erişim verilmesi

Çizelge 94. Örnek yayınlama ve erişim gereksinimlerine abone olma

Konu	Abone olma erişimi gerekli	Yayınlama erişimi gerekli	Konu nesnesi
Fiyat	Kullanıcı yok	Kullanıcı yok	Yok
Fiyat/Meyve	USER1	USER1	MEYVE
Fiyat/Meyve/Elmalar	USER1 ve USER2		Elma
Fiyat/Meyve/Portakal	USER1		Turuncu

Yordam

Şu şekilde erişim ver:

- z/OS **z/OS** :

  In an earlier task USER1 was granted access to subscribe to topic `"Price/Fruit"` by granting the user access to the `hlq.SUBSCRIBE.FRUIT` profile.

  `"Price/Fruit"` konusunda yayınlama yapmak için `hlq.PUBLISH.FRUIT` profiline USER1 erişimi verin. Aşağıdaki RACF komutlarını kullanarak bunu yapın:

  ```
  RDEFINE MXTOPIC hlq.PUBLISH.FRUIT UACC(NONE)
  PERMIT hlq.PUBLISH.FRUIT CLASS(MXTOPIC) ID(USER1) ACCESS(ALTER)
  ```

- Diğer platformlar:

  Grant access to USER1 to publish to topic `"Price/Fruit"` by granting the user publish access to the `FRUIT` profile. Bunu, altyapıya ilişkin yetkilendirme komutunu kullanarak yapın:

  ALW **AIX, Linux, and Windows sistemleri**

  ```
  setmqaut -t topic -n FRUIT -p USER1 +pub
  ```

IBM i

```
GRTMQAUT OBJ(FRUIT) OBJTYPE(*TOPIC) USER(USER1) AUT(*PUB)
```

Sonuçlar

z/OSüzerinde, USER1 "Price/Fruit" konusuna yayınlama girişiminde bulunduğunda, MQOPEN çağrısındaki güvenlik denetimi başarılı olur.

USER2 "Price/Fruit" konusunda yayınlama girişiminde bulunduğunda, sonuç bir MQRC_NOT_AUTHORIZED iletisiyle birlikte başarısız olur:

- z/OS z/OS' ta, konsolda görüntülenen ve denenen konu ağacında tam güvenlik yolunu gösteren aşağıdaki iletiler:

```
ICH408I USER(USER2   ) ...
  hlq.PUBLISH.FRUIT ...

ICH408I USER(USER2   ) ...
  hlq.PUBLISH.SYSTEM.BASE.TOPIC ...
```

- ALW AIX, Linux, and Windows platformlarında aşağıdaki yetkilendirme olayı:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

- IBM i IBM üzerinde aşağıdaki yetkilendirme olayı:

```
MQRC_NOT_AUTHORIZED
ReasonQualifier      MQRQ_OPEN_NOT_AUTHORIZED
UserIdentifier        USER2
AdminTopicNames      FRUIT, SYSTEM.BASE.TOPIC
TopicString           "Price/Fruit"
```

Bu görevlerin eksiksiz bir kümesini izleyerek, USER1 ve USER2 ' e listelenen konuları yayınlamak ve bunlara abone olmak için aşağıdaki erişim yetkilerini verir:

Çizelge 95. Güvenlik örneklerinden kaynaklanan erişim yetkililerinin tam listesi

Konu	Abone olma erişimi gerekli	Yayınlama erişimi gerekli	Konu nesnesi
Fiyat	Kullanıcı yok	Kullanıcı yok	Yok
Fiyat/Meyve	USER1	USER1	MEYVE
Fiyat/Meyve/Elmalar	USER1 ve USER2		Elma
Fiyat/Meyve/Portakal	USER1		Turuncu
Fiyat/Sebze		USER1	VG
Fiyat/Sebze/Patates			

Çizelge 95. Güvenlik örneklerinden kaynaklanan erişim yetkililerinin tam listesi (devamı var)			
Konu	**Abone olma erişimi gerekli**	**Yayınlama erişimi gerekli**	**Konu nesnesi**
Fiyat/Sebze/ Soğan			

Konu ağacında farklı düzeylerde güvenlik erişimi için farklı gereksinimleriniz varsa, dikkatli planlama, z/OS konsol günlüğünde dış güvenlik uyarıları almamanızı sağlar. Ağaç içinde güvenliğin doğru düzeyde ayarlanması, yanıltıcı güvenlik iletilerini önler.

Abonelik güvenliği

MQSO_ALTERNATE_USER_AUTHORITY

AlternateUser(Alternatif Kullanıcı) alanı, bu MQSUB çağrısının geçerliliğini denetlemek için kullanılacak bir kullanıcı kimliği içerir. Arama, yalnızca bu AlternateUserkimliği belirtilen erişim seçenekleriyle konuya abone olma yetkisine sahip ise, uygulamanın çalıştırıldığı kullanıcı kimliğinin bunu yapma yetkisi olup olmadığına bakılmaksızın başarılı olabilir.

MQSO_SET_IDENTITY_CONTEXT

Abonelik, PubAccountingToken ve PubApplIdentityData alanlarında sağlanan muhasebe simgesini ve uygulama kimlik verilerini kullanmaktır.

Bu seçenek belirtilirse, MQSO_MANAGED seçeneğinin de kullanıldığı durumlar dışında, hedef kuyruğa MQOO_SET_IDENTITY_CONTEXT ile MQOPEN çağrısı kullanılarak erişildiği gibi aynı yetki denetimi gerçekleştirilir; bu durumda hedef kuyrukta yetki denetimi yoktur.

Bu seçenek belirtilmezse, bu aboneye gönderilen yayınlarla ilişkili varsayılan bağlam bilgileri aşağıdaki gibi olur:

Çizelge 96. Varsayılan yayın bağlamı bilgileri	
MQMD ' deki Alan	**Kullanılan değer**
UserIdentifier	Yayın yapıldığında, abonelikle ilişkili kullanıcı kimliği (DISPLAY SBSTATUS ekranındaki SUBUSER alanına bakın).
AccountingToken	Olanaklıysa ortamdan saptanır; tersi durumda MQACT_NONE olarak ayarlanır.
ApplIdentityVerileri	Boşluk olarak ayarlayın.

Bu seçenek yalnızca MQSO_CREATE ve MQSO_ALTER ile geçerlidir. MQSO_RESUME ile kullanılırsa, PubAccountingSimgesi ve PubApplIdentityData alanları yoksayılır, bu nedenle bu seçeneğin bir etkisi yoktur.

Abonelik, daha önce aboneliğin sağladığı kimlik bağlamı bilgilerinin bulunduğu bu seçenek kullanılmadan değiştirilirse, değiştirilen abonelik için varsayılan bağlam bilgileri oluşturulur.

Farklı kullanıcı kimliklerinin MQSO_ANY_USERID seçeneğiyle kullanılmasına izin veren bir abonelik farklı bir kullanıcı kimliğiyle sürdürüldüğünde, aboneliğe sahip yeni kullanıcı kimliği için varsayılan kimlik bağlamı oluşturulur ve yeni kimlik bağlamını içeren sonraki yayınlar teslim edilir.

AlternateSecurityTanıtıcısı

Bu, uygun yetki denetimlerinin gerçekleştirilmesine izin vermek için yetkilendirme hizmetine AlternateUserkimliği ile iletilen bir güvenlik tanıtıcısıdır. AlternateSecurityTanıtıcısı yalnızca MQSO_ALTERNATE_USER_AUTHORITY belirtildiyse ve AlternateUserId alanı ilk boş karaktere ya da alanın sonuna kadar tamamen boş değilse kullanılır.

MQSO_ANY_USERID abonelik seçeneği

MQSO_ANY_USERID belirtildiğinde, abonenin kimliği tek bir kullanıcı kimliğiyle sınırlı değildir. Bu, herhangi bir kullanıcının uygun yetkiye sahip olduğunda aboneliği değiştirmesini ya da sürdürmesini sağlar. Aynı anda yalnızca tek bir kullanıcı aboneliğe sahip olabilir. Başka bir uygulama tarafından kullanılmakta olan bir aboneliğin kullanımını sürdürme girişimi, çağrı MQRC_SUBSCRIPTION_IN_USE ile başarısız olur.

Bu seçeneği var olan bir aboneliğe eklemek için MQSUB çağrısı (MQSO_ALTER kullanılarak), özgün abonelikle aynı kullanıcı kimliğinden gelmelidir.

Bir MQSUB çağrısı MQSO_ANY_USERID ile var olan bir aboneliğe gönderme yapıyorsa ve kullanıcı kimliği özgün abonelikten farklıysa, çağrı ancak yeni kullanıcı kimliğinin konuya abone olma yetkisi varsa başarılı olur. İşlem başarılı bir şekilde tamamlandıktan sonra, bu abonenin gelecekteki yayınları, yayında yeni kullanıcı kimliği ayarlanmış olarak abonenin kuyruğuna yerleştirilir.

MQSO_FIXED_USERID

MQSO_FIXED_USERID belirtildiğinde, abonelik yalnızca sahip olan tek bir kullanıcı kimliği tarafından değiştirilebilir ya da sürdürülebilir. Bu kullanıcı kimliği, bu seçeneği ayarlanan aboneliği değiştiren son kullanıcı kimliğidir; böylece MQSO_ANY_USERID seçeneği kaldırılır ya da herhangi bir değişiklik yapılmazsa, aboneliği yaratan kullanıcı kimliği olur.

Bir MQSUB komutu MQSO_ANY_USERID ile var olan bir aboneliğe gönderme yaparsa ve MQSO_FIXED_USERID seçeneğini kullanmak için aboneliği (MQSO_ALTER kullanılarak) değiştirirse, aboneliğin kullanıcı kimliği artık bu yeni kullanıcı kimliğinde düzeltilmiştir. Arama, yalnızca yeni kullanıcı kimliğinin konuya abone olma yetkisi varsa başarılı olur.

Abonelik sahibi olarak kaydedilenden başka bir kullanıcı kimliği bir MQSO_FIXED_USERID aboneliğini sürdürme ya da değiştirme girişiminde bulunursa, çağrı MQRC_IDENTITY_MISMATCH ile başarısız olur. Bir aboneliğin sahip olan kullanıcı kimliği, DISPLAY SBSTATUS komutu kullanılarak görüntülenebilir.

MQSO_ANY_USERID ya da MQSO_FIXED_USERID belirtilmezse, varsayılan MQSO_FIXED_USERID olur.

Kuyruk yöneticileri arasında yayınlama/abone olma güvenliği

Yetkili sunucu abonelikleri ve yayınları gibi iç iletileri yayınlama/abone olma, normal kanal güvenlik kuralları kullanılarak sistem kuyruklarını yayınlama/abone olma işlemleri için kullanılır. Bu konudaki bilgi ve çizgeler, bu iletilerin teslim edilmesinde yer alan çeşitli süreçleri ve kullanıcı kimliklerini vurgular.

Yerel erişim denetimi

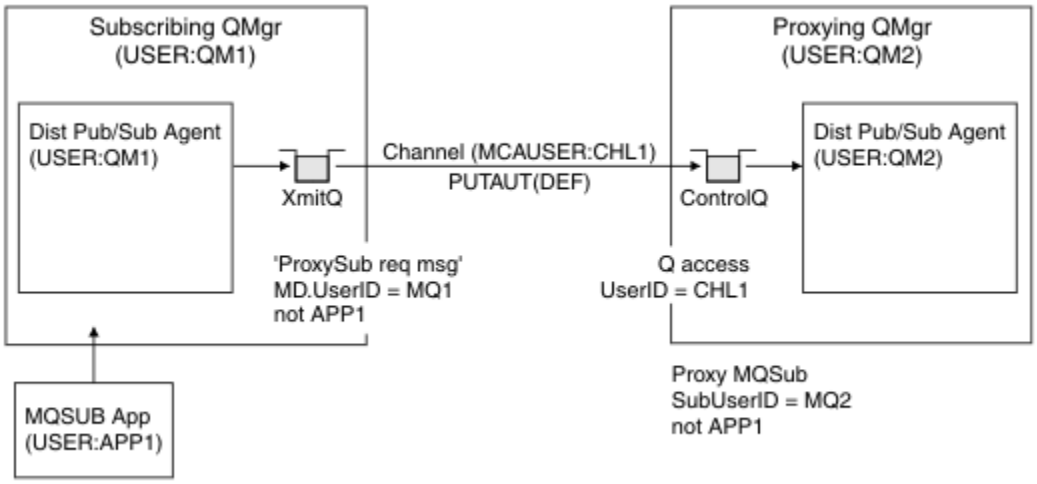
Yayın ve aboneliklere ilişkin konulara erişim, Yayınla/abone ol güvenliğindeaçıklanan yerel güvenlik tanımları ve kuralları tarafından yönetilir. z/OSişletim sistemi üzerinde, erişim denetimi oluşturmak için yerel konu nesnesi gerekmez. Diğer platformlarda da erişim denetimi için yerel konu gerekmez. Yöneticiler, kümelenmiş konu nesnelerine henüz kümede olup olmadıklarından bağımsız olarak erişim denetimini uygulamayı seçebilir.

Sistem yöneticileri, kendi yerel sistemlerinde erişim denetiminden sorumludur. Erişim denetimi ilkelerinden sorumlu olmak için sıradüzeninin ya da küme toplayıcılarının diğer üyelerinin yöneticilerine güvenmeleri gerekir. Erişim denetimi her bir ayrı makine için tanımlandığından, ince düzey denetimi gerekirse bu büyük olasılıkla yük olacaktır. Herhangi bir erişim denetimi zorunlu olmayabilir ya da erişim denetimi, konu ağacındaki üst düzey nesnelerde tanımlanabilir. Konu ad alanının her alt bölümü için ince düzey erişim denetimi tanımlanabilir.

Yetkili sunucu aboneliği yapılması

Bir kuruluşun kuyruk yöneticisini kuyruk yöneticinize bağlayacağına ilişkin güven, normal kanal kimlik doğrulaması yöntemleriyle onaylanır. Bu güvenilir kuruluşun da dağıtılmış yayınlama/abone olma izni varsa, bir yetki denetimi yapılır. Kanal, dağıtılmış bir yayınlama/abone olma kuyruğuna ileti koyduğunda denetim yapılır. Örneğin, bir ileti `SYSTEM.INTER.QMGR.CONTROL` kuyruğuna konursa. Kuyruk yetki denetiminin kullanıcı kimliği, alan kanalın `PUTAUT` değerlerine bağlıdır. Örneğin, kanalın kullanıcı kimliği `MCAUSER`, değere ve platforma bağlı olarak ileti bağlamı. Kanal güvenliği hakkında daha fazla bilgi için bkz. Kanal güvenliği.

Yetkili sunucu abonelikleri, uzak kuyruk yöneticisinde dağıtılmış yayınlama/abone olma aracısının kullanıcı kimliğiyle yapılır. Örneğin, Şekil 30 sayfa 518içinde `QM2` . Kullanıcı kimliği sistemde tanımlı olduğundan ve bu nedenle etki alanı çakışması olmadığından, kullanıcıya yerel konu nesnesi tanıtımlarına kolayca erişim izni verilir.



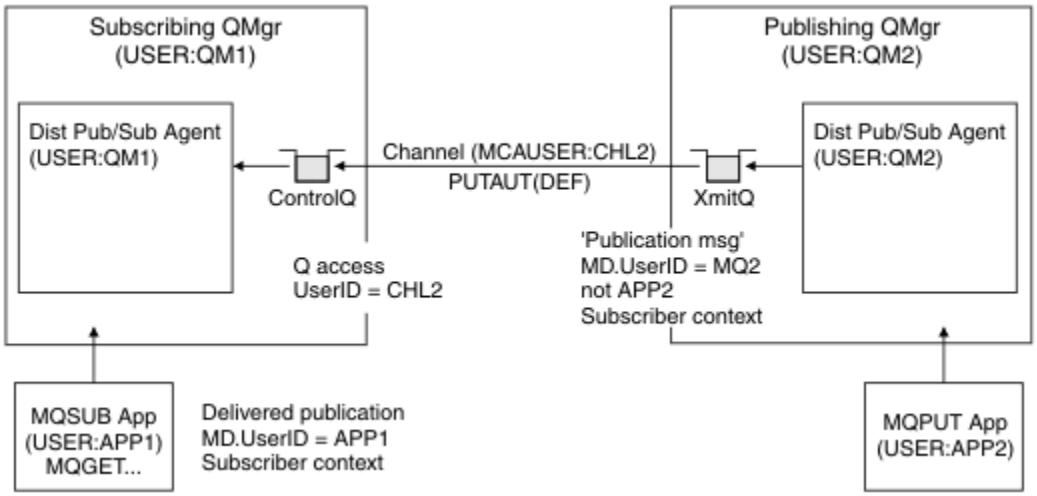
Şekil 30. Yetkili sunucu abonelik güvenliği, abonelik yapma

Uzak yayınların geri gönderilmesi

Yayınlama kuyruğu yöneticisinde bir yayın yaratıldığında, yetkili sunucu aboneliği için yayının bir kopyası yaratılır. Kopyalanan yayının bağlamı, aboneliği yapan kullanıcı kimliğinin bağlamını içerir; Şekil 31 sayfa 519içinde `QM2` . Yetkili sunucu aboneliği, uzak kuyruk olan bir hedef kuyrukla yaratılır ve yayınlama iletisi bir iletim kuyruğuna çözülür.

Bir kuruluşun kuyruk yöneticisini (`QM2`) başka bir kuyruk yöneticisine (`QM1`) bağlayacağına ilişkin güven, normal kanal kimlik doğrulaması yöntemleriyle onaylanır. Bu güvenilir kuruluşun dağıtılmış yayınlama/abone olma işlemi yapmasına izin verilirse, kanal yayın iletisini dağıtılmış yayınlama/abone olma yayın kuyruğuna yerleştirdiğinde bir yetki denetimi gerçekleştirilir `SYSTEM.INTER.QMGR.PUBS`. Kuyruk yetki denetiminin kullanıcı kimliği, alan kanalın `PUTAUT` değerine bağlıdır (örneğin, kanalın kullanıcı kimliği, `MCAUSER`, ileti bağlamı ve değere ve platforma bağlı olarak diğerleri). Kanal güvenliği hakkında daha fazla bilgi için bkz. Kanal güvenliği.

Yayınlama iletisi abone olan kuyruk yöneticisine ulaştığında, o kuyruk yöneticisinin yetkisi altında başka bir MQPUT gerçekleştirilir ve iletiye ilişkin bağlam, her biri ileti verildiği için yerel abonelerin her birinin bağlamıyla değiştirilir.



Şekil 31. Yetkili sunucu aboneliği güvenliği, yayınları iletme

Güvenlikle ilgili olarak çok az şey dikkate alınan bir sistemde, dağıtılmış yayınlama/abone olma işlemlerinin `mqm` grubundaki bir kullanıcı kimliği altında çalışması olasıdır, bir kanaldaki `MCAUSER` parametresi boştur (varsayılan) ve iletiler gerektiğinde çeşitli sistem kuyruklarına teslim edilir. Güvenli olmayan sistem, dağıtılmış yayınlama/abone olma özelliğini göstermek için bir kavram kanıtı oluşturmanızı kolaylaştırır.

Güvenliğin daha ciddi olarak değerlendirildiği bir sistemde, bu iç iletiler kanal üzerinden giden herhangi bir iletiyle aynı güvenlik denetimlerine tabidir.

Kanal, boş olmayan bir `MCAUSER` ve `MCAUSER` değerinin işaretlenmesi gerektiğini belirten bir `PUTAUT` değeriyle ayarlanırsa, söz konusu `MCAUSER` ' e `SYSTEM.INTER.QMGR.*` kuyruklarına erişim yetkisi verilmelidir. Farklı `MCAUSER` kimlikleri altında çalışan kanallar olan birden çok farklı uzak kuyruk yöneticisi varsa, tüm bu kullanıcı kimliklerine `SYSTEM.INTER.QMGR.*` kuyruklarına erişim verilmesi gerekir. Farklı `MCAUSER` tanıtıcıları altında çalışan kanallar oluşabilir; örneğin, tek bir kuyruk yöneticisinde birden çok sıradüzenli bağlantı yapılandırıldığında.

Kanal, iletinin bağlamının kullanıldığını belirten bir `PUTAUT` değeriyle ayarlanırsa, iç iletideki kullanıcı kimliğine dayalı olarak `SYSTEM.INTER.QMGR.*` kuyruklarına erişim denetlenir. Tüm bu iletiler, iç iletiyi gönderen kuyruk yöneticisinden dağıtılmış yayınlama/abone olma aracısının kullanıcı kimliğiyle ya da yayınlama iletisiyle (bkz. Şekil 31 sayfa 519) yerleştirildiğinden, dağıtımlı yayınlama/abone olma güvenliğinizi bu şekilde ayarlamak istiyorsanız, çeşitli sistem kuyruklarına (uzak kuyruk yöneticisi başına bir tane) erişim vermek için çok büyük bir kullanıcı kimliği kümesi değildir. Kanal bağlamı güvenliğinin her zaman sahip olduğu sorunlarla aynı sorunlara sahiptir; farklı kullanıcı kimliği etki alanları ve iletideki kullanıcı kimliğinin alıcı sistemde tanımlanmamış olması. Ancak bu, gerekirse koşmanın son derece kabul edilebilir bir yoludur.

z/OS Sistem kuyruğu güvenliği , dağıtılmış yayınlama/abone olma ortamınızı güvenli bir şekilde kurmak için gereken kuyrukların ve erişimin bir listesini sağlar. Güvenlik ihlalleri nedeniyle herhangi bir iç ileti ya da yayın konamazsa, kanal günlüğe normal şekilde bir ileti yazar ve iletiler normal kanal hata işlemesine göre gönderilebilir.

Dağıtılmış yayınlama/abone olma amacıyla tüm kuyruklar arası yönetici ileti sistemi normal kanal güvenliği kullanılarak çalıştırılır.

Konu düzeyinde yayınları ve yetkili sunucu aboneliklerini kısıtlama hakkında bilgi için bkz. Güvenliğin yayınlanması/abone olunması.

Kuyruk yöneticisi sıradüzeniyle varsayılan kullanıcı kimliklerinin kullanılması

Farklı altyapılarda çalışan kuyruk yöneticilerinden oluşan bir sıradüzeniniz varsa ve varsayılan kullanıcı kimliklerini kullanıyorsanız, bu varsayılan kullanıcı kimliklerinin platformlar arasında farklılık gösterdiğini ve hedef altyapıda bilinmeyebileceğini unutmayın. Sonuç olarak, bir altyapıda çalışan bir kuyruk

yöneticisi, diğer altyapılardaki kuyruk yöneticilerinden alınan iletileri MQRC_NOT_AUTHORIZEDneden koduyla reddeder.

İletilerin reddedilmesini önlemek için, diğer altyapılarda kullanılan varsayılan kullanıcı kimliklerine en az aşağıdaki yetkilerin eklenmesi gerekir:

- SYSTEM.BROKER. Kuyruklar
- *PUB SYSTEM.BROKER. Konular
- SYSTEM.BROKER.CONTROL.QUEUE (KUYRUK).

Kuyruk yöneticisi sıradüzenine sahip varsayılan kullanıcı kimlikleri şunlardır:

Hizmet olarak sunulan	Varsayılan kullanıcı kimliği
Windows	mqm
AIX and Linux sistemleri	mqm
IBM i	QMQM
z/OS	Kanal başlatıcı adres alanı kullanıcı kimliği

z/OS, AIX, Linux, and Windows altyapılarında kuyruk yöneticileri için IBM i üzerinde bir kuyruk yöneticisine sıradüzensel olarak bağlıysa, 'qmqm' kullanıcı kimliği yaratın ve bu kullanıcı kimliğine erişim verin.

IBM i ve z/OS platformlarındaki kuyruk yöneticileri için, AIX, Linux, and Windows üzerindeki bir kuyruk yöneticisine sıradüzensel olarak bağlıysa, 'mqm' kullanıcı kimliğini yaratın ve bu kullanıcı kimliğine erişim verin.

Çoklu platformlarüzerinde Kuyruk Yöneticileri için z/OS üzerinde bir kuyruk yöneticisine sıradüzensel olarak bağlıysa, z/OS kanal başlatıcısı adres alanı kullanıcı kimliği oluşturun ve bu kullanıcıya erişim verin.

Kullanıcı kimlikleri büyük ve küçük harfe duyarlı olabilir. Kaynak kuyruk yöneticisi (Çoklu platformlarüzerindeyse), kullanıcı kimliğini büyük harfli olmaya zorlar. Alıcı kuyruk yöneticisi (AIX, Linux, and Windowsise), kullanıcı kimliğini küçük harfli olmaya zorlar. Bu nedenle, AIX and Linux sistemlerinde oluşturulan tüm kullanıcı kimliklerinin küçük harf biçiminde oluşturulması gerekir. Bir ileti çıkışı kurulduysa, kullanıcı kimliğini büyük ya da küçük harfe zorlama işlemi gerçekleşmez. İleti çıkışının kullanıcı kimliğini nasıl işlediğini anlamak için dikkatli olunmalıdır.

Kullanıcı kimliklerinin dönüştürülmesiyle ilgili olası sorunları önlemek için:

- AIX, Linux, and Windows sistemlerinde kullanıcı kimliklerinin küçük harfle belirtildiğinden emin olun.
- IBM i ve z/OSüzerinde, kullanıcı kimliklerinin büyük harfle belirtildiğinden emin olun.

IBM MQ Console ve REST API güvenliği

IBM MQ Console ve REST API için güvenlik, mqwebuser.xml dosyasında mqweb sunucusu yapılandırması düzenlenerek yapılandırılır.

Bu görev hakkında

Kullanıcı işlemlerini izleyebilir ve mqweb sunucusunun günlük dosyalarını inceleyerek IBM MQ Console ve REST API kullanımını denetleyebilirsiniz.

IBM MQ Console ve REST API kullanıcılarının kimlikleri aşağıdaki kullanılarak doğrulanabilir:

- Temel kayıt dosyası
- LDAP kaydı
- Yerel işletim sistemi kaydı
- z/OS üzerinde SAF
- WebSphere Liberty tarafından desteklenen başka bir kayıt dosyası tipi

Roller, IBM MQ Console kullanıcılarına ve REST API kullanıcılarına IBM MQ nesnelerine hangi erişim düzeyinin verildiğini belirlemek için atanabilir. Örneğin, ileti alışverişi gerçekleştirmek için kullanıcılara `MQWebUser` rolü atanmalıdır. Kullanılabilir roller hakkında daha fazla bilgi için bkz. “IBM MQ Console ve REST API üzerindeki roller” sayfa 532.

Bir kullanıcıya bir rol atandıktan sonra, kullanıcının kimliğini doğrulamak için kullanılabilecek birçok yöntem vardır. IBM MQ Consoleile kullanıcılar bir kullanıcı adı ve parolayla oturum açabilir ya da istemci sertifikası kimlik doğrulamasını kullanabilir. REST APIile kullanıcılar temel HTTP kimlik doğrulamasını, belirteç tabanlı kimlik doğrulamasını ya da istemci sertifikası kimlik doğrulamasını kullanabilir.

Yordam

1. Kullanıcıların kimliğini doğrulamak için kullanıcı kayıt defterini tanımlayın ve her bir kullanıcıya ya da gruba IBM MQ Console ya da REST APIürününü kullanmak üzere kullanıcıları ve grupları yetkilendirmek için bir rol atayın. Daha fazla bilg için bkz. “Kullanıcıları ve rolleri yapılandırma” sayfa 522
2. IBM MQ Console kullanıcılarının mqweb sunucusu ile nasıl kimlik doğrulaması yapacaklarını seçin. Tüm kullanıcılar için aynı yöntemi kullanmanız gerekmez:
   - Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda bir kullanıcı, IBM MQ Console oturum açma ekranına bir kullanıcı kimliği ve parola girer. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Bu kimlik doğrulama seçeneğini kullanmak için başka bir yapılandırma gerekmez, ancak isteğe bağlı olarak LTPA simgesi için süre bitimini yapılandırabilirsiniz. Daha fazla bilgi için LTPA belirteci süre bitim aralığını yapılandırmabaşlıklı konuya bakın.
   - Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı IBM MQ Console' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.
3. REST API kullanıcılarının mqweb sunucusu ile nasıl kimlik doğrulaması yapacaklarını seçin. Tüm kullanıcılar için aynı yöntemi kullanmanız gerekmez:
   - Kullanıcıların HTTP temel kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı adı ve parola kodlanır, ancak şifrelenmez ve kullanıcının kimliğini doğrulamak ve bu istek için yetki vermek üzere her REST API isteğiyle birlikte gönderilir. Bu kimlik doğrulamasının güvenli olması için güvenli bir bağlantı kullanmanız gerekir. Yani, HTTPSkullanmalısınız. Daha fazla bilgi için bkz “REST API ile HTTP temel kimlik doğrulamasını kullanma” sayfa 541.
   - Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı REST API `login` kaynağı için HTTP POST yöntemiyle bir kullanıcı kimliği ve parola sağlar. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Daha fazla bilgi için bkz “REST API ile belirteç tabanlı kimlik doğrulamasını kullanma” sayfa 542.

     Bu kimlik doğrulamasının güvenli olması için güvenli bir bağlantı kullanmanız gerekir. Yani, HTTPSkullanmalısınız. Ancak, HTTP bağlantılarını etkinleştirdiyseniz, bir HTTP bağlantısı için HTTPS bağlantısı için yayınlanan bir LTPA belirtecinin kullanılmasına izin verebilirsiniz. Daha fazla bilgi için LTPA simgesini yapılandırmabaşlıklı konuya bakın.
   - Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı REST API' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.
4. İsteğe bağlı: REST APIiçin Köken Arası Kaynak Paylaşımını yapılandırın.

   Varsayılan olarak, bir web tarayıcısı, komut dosyası REST APIile aynı kökenden olmadığında JavaScriptgibi komut dosyalarının REST API ' ı çağırmasına izin vermez. Yani, kökler arası istekler etkinleştirilmez. Kökler Arası Kaynak Paylaşımını (CORS), belirtilen URL adreslerinden gelen kökler

arası isteklere izin verecek şekilde yapılandırabilirsiniz. Daha fazla bilgi için bkz “REST API için CORS ' un yapılandırılması” sayfa 545.

5. İsteğe bağlı: IBM MQ Console ve REST APIiçin anasistem üstbilgisi doğrulamasını yapılandırın.

   Anasistem üstbilgisi geçerlilik denetimini yapılandırabilir ve yalnızca belirli anasistem üstbilgilerini içeren isteklerin IBM MQ Console ve REST APItarafından işlendiğinden emin olmak için izin verilen anasistem adları ve kapıları listesi oluşturabilirsiniz. Daha fazla bilgi için bkz “IBM MQ Console ve REST API için anasistem üstbilgisi doğrulamasını yapılandırma” sayfa 546.

Kullanıcıları ve rolleri yapılandırma

IBM MQ Console ya da REST APIöğesini kullanmak için kullanıcıların mqweb sunucusunda tanımlı bir kullanıcı kaydına karşı kimlik doğrulaması yapması gerekir.

Bu görev hakkında

Kimliği doğrulanmış kullanıcıların, IBM MQ Console ve REST APIyeteneklerine erişim yetkisi veren gruplardan birinin üyesi olması gerekir. Varsayılan olarak, kullanıcı kaydı herhangi bir kullanıcı içermez; bunların `mqwebuser.xml` dosyası düzenlenerek eklenmesi gerekir.

Kullanıcıları ve grupları yapılandırırken, öncelikle kullanıcıların ve grupların kimliklerini doğrulamak için bir kullanıcı kaydı yapılandırırsınız. Bu kullanıcı kaydı, IBM MQ Console ile REST APIarasında paylaşılır. Kullanıcılarınız ve gruplarınız için rolleri yapılandırırken kullanıcıların ve grupların IBM MQ Console, REST APIya da her ikisine de erişip erişemeyeceğini denetleyebilirsiniz.

Kullanıcı kayıt defterini yapılandırdıktan sonra, kullanıcılara ve gruplara yetki vermek üzere rolleri yapılandırırsınız. Managed File Transferiçin REST API kullanımına özgü roller de içinde olmak üzere birçok rol vardır. Her rol farklı bir erişim düzeyi verir. Daha fazla bilgi için bkz “IBM MQ Console ve REST API üzerindeki roller” sayfa 532.

Kullanıcıların ve grupların yapılandırmasını basitleştirmek için mqweb sunucusu ile birlikte bir dizi örnek XML dosyası sağlanır. WebSphere Liberty (WLP) içinde güvenliği yapılandırmaya aşina olan kullanıcılar, örnekleri kullanmamayı tercih edebilir. WLP, burada belgelenenlere ek olarak başka yetkilendirme yetenekleri de sağlar.

Yordam

- `basic_registry.xml` dosyasını kullanarak temel bir kayıt dosyasıyla kullanıcıları ve grupları yapılandırın.

  Kayıt defterindeki kullanıcı adları ve parolalar, IBM MQ Console ve REST APIkullanıcılarını doğrulamak ve yetkilendirmek için kullanılır.

  `basic_registry.xml` örnek dosyasını kullanarak temel bir kayıt dosyası yapılandırmak için bkz. “IBM MQ Console ve REST API için temel kayıt dosyası yapılandırılması” sayfa 523.

- `ldap_registry.xml` dosyasını kullanarak LDAP kayıt defteriyle kullanıcıları ve grupları yapılandırın.

  LDAP kayıt defterindeki kullanıcı adları ve parolalar, IBM MQ Console ve REST APIkimliklerini doğrulamak ve bunlara yetki vermek için kullanılır.

  `ldap_registry.xml` örnek dosyasını kullanarak bir LDAP kaydını yapılandırmak için bkz. “IBM MQ Console ve REST API için LDAP kaydı yapılandırılması” sayfa 528.

- ALW

  `local_os_registry.xml` dosyasını kullanarak kullanıcıları ve grupları yerel işletim sistemi kayıt defteriyle yapılandırın.

  İşletim sistemi kayıt defterindeki kullanıcı adları ve parolalar, IBM MQ Console ve REST APIkullanıcılarını doğrulamak ve yetkilendirmek için kullanılır.

  `local_os_registry.xml` örnek dosyasını kullanarak yerel bir OS kaydı yapılandırmak için bkz. “IBM MQ Console ve REST API için yerel bir işletim sistemi kayıt defterinin yapılandırılması” sayfa 526.

- z/OS
  `zos_saf_registry.xml` dosyasını kullanarak z/OS üzerinde Sistem yetkilendirme olanağı (SAF) arabirimiyle kullanıcıları ve grupları yapılandırın.

  RACFya da başka bir güvenlik ürünü olan profiller, kullanıcılara ve gruplara rollere erişim vermek için kullanılır. RACF veritabanındaki kullanıcı adları ve parolalar, IBM MQ Console ve REST APIkullanıcılarını doğrulamak ve yetkilendirmek için kullanılır.

  `zos_saf_registry.xml` örnek dosyasını kullanarak SAF arabirimini yapılandırmak için bkz. “IBM MQ Console ve REST API için SAF kaydının yapılandırılması” sayfa 530.
- `no_security.xml` dosyasını kullanarak HTTPSkullanarak IBM MQ Consoleya da REST API' e erişme yeteneği de dahil olmak üzere güvenliği devre dışı bırakın.

Sonraki adım

Kullanıcıların kimlik doğrulamasını seçin:

IBM MQ Console Kimlik Doğrulaması Seçenekleri

- Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda bir kullanıcı, IBM MQ Console oturum açma ekranına bir kullanıcı kimliği ve parola girer. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Bu kimlik doğrulama seçeneğini kullanmak için başka bir yapılandırma gerekmez, ancak isteğe bağlı olarak LTPA belirteci için süre bitimi aralığını yapılandırabilirsiniz. Daha fazla bilgi için LTPA belirteci süre bitim aralığını yapılandırmabaşlıklı konuya bakın.
- Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı IBM MQ Console' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.

REST API Kimlik Doğrulaması Seçenekleri

- Kullanıcıların HTTP temel kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı adı ve parola kodlanır, ancak şifrelenmez ve kullanıcının kimliğini doğrulamak ve bu istek için yetki vermek üzere her REST API isteğiyle birlikte gönderilir.
  Bu kimlik doğrulamasının güvenli olması için güvenli bir bağlantı kullanmanız gerekir. Yani, HTTPSkullanmalısınız. Daha fazla bilgi için bkz “REST API ile HTTP temel kimlik doğrulamasını kullanma” sayfa 541.
- Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı REST API `login` kaynağı için HTTP POST yöntemiyle bir kullanıcı kimliği ve parola sağlar. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Daha fazla bilgi için bkz “REST API ile belirteç tabanlı kimlik doğrulamasını kullanma” sayfa 542. LTPA belirteci için süre bitim aralığını yapılandırabilirsiniz. Daha fazla bilgi için LTPA simgesini yapılandırmabaşlıklı konuya bakın.
- Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı REST API' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.

IBM MQ Console ve REST API için temel kayıt dosyası yapılandırılması

`mqwebuser.xml` dosyasında temel bir kayıt dosyası yapılandırabilirsiniz. IBM MQ Console ve REST APIkullanıcılarını doğrulamak ve yetkilendirmek için XML dosyasındaki kullanıcı adları, parolalar ve roller kullanılır.

Başlamadan önce

- Temel kayıt dosyasında kullanıcıları yapılandırırken, her kullanıcıya bir rol atamanız gerekir. Her rol, IBM MQ Console ve REST API' e erişmek için farklı ayrıcalık düzeyleri sağlar ve izin verilen bir

işlem denendiğinde kullanılan güvenlik bağlamını belirler. Temel kaydı yapılandırmadan önce bu rolleri anlamanız gerekir. Rollerin her biri hakkında daha fazla bilgi için bkz. "IBM MQ Console ve REST API üzerindeki roller" sayfa 532.

- Bu görevi tamamlamak için, `mqwebuser.xml` dosyasını düzenlemek üzere yeterli ayrıcalığa sahip bir kullanıcı olmanız gerekir:
  - z/OS z/OS' da `mqwebuser.xml` dosyasına yazma erişiminiz olmalıdır.
  - Multi Diğer tüm işletim sistemlerinde ayrıcalıklı kullanıcıolmanız gerekir.
  - V 9.3.5 Linux mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki `mqwebuser.xml` dosyasına yazma erişiminiz olmalıdır.

Yordam

1. `basic_registry.xml` örnek XML dosyasını aşağıdaki yollardan birinden kopyalayın:
   - IBM MQ kuruluşunda:
     - ALW AIX, Linux, and Windowssistemlerinde: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
     - z/OS z/OSsistemlerinde: `PathPrefix /web/mq/samp/configuration`

       Burada `PathPrefix` , IBM MQ for z/OS UNIX System Services Components kuruluş yoludur.
   - V 9.3.5 Linux Bağımsız bir IBM MQ Web Server kuruluşunda: *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     Burada *MQWEB_INSTALLATION_PATH* , IBM MQ Web Server kuruluş dosyasının açıldığı dizindir.
2. Örnek dosyayı uygun dizine yerleştirin:
   - IBM MQ kuruluşunda:
     - Linux AIX AIX ya da Linuxsistemlerinde: `/var/mqm/web/installations/`*installationName*`/servers/mqweb`
     - Windows Windowsüzerinde: *MQ_DATA_PATH*`\web\installations\`*installationName*`\servers\mqweb`; burada *MQ_DATA_PATH* , IBM MQ veri yoludur. Bu yol, IBM MQkuruluşu sırasında seçilen veri yoludur. Varsayılan olarak bu yol `C:\ProgramData\IBM\MQ`olur.
     - z/OS z/OSişletim tarihinde: *WLP_user_directory*`/servers/mqweb`

       Burada *WLP_user_directory* , **crtmqweb** komut dosyası mqweb sunucusu tanımlamasını oluşturmak için çalıştırıldığında belirtilen dizindir.
   - V 9.3.5 Linux Bağımsız bir IBM MQ Web Server kuruluşunda: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     Burada *MQ_OVERRIDE_DATA_PATH* , **MQ_OVERRIDE_DATA_PATH** ortam değişkeninin gösterdiği IBM MQ Web Server veri dizinidir.
3. İsteğe bağlı: `mqwebuser.xml`içinde herhangi bir yapılandırma ayarını değiştirdiyseniz, bunları örnek dosyaya kopyalayın.
4. Var olan `mqwebuser.xml` dosyasını silin ve örnek dosyayı `mqwebuser.xml`olarak yeniden adlandırın.
5. **basicRegistry** etiketleri içinde kullanıcılar ve gruplar eklemek için yeni `mqwebuser.xml` dosyasını düzenleyin.

   `MQWebUser` rolüne sahip herhangi bir kullanıcının yalnızca, kuyruk yöneticisinde gerçekleştirmesi için kullanıcı kimliği verilen işlemleri gerçekleştirebileceğini unutmayın. Bu nedenle, kayıt dosyasında tanımlanan kullanıcı kimliğinin, IBM MQ ' in kurulu olduğu sistemde aynı bir kullanıcı kimliğine

sahip olması gerekir. Bu kullanıcı kimlikleri aynı durumda olmalıdır; tersi durumda, kullanıcı kimlikleri arasındaki eşleme başarısız olabilir.

Temel kullanıcı kayıt defterlerini yapılandırma hakkında daha fazla bilgi için WebSphere Liberty belgelerinde Configuring a basic user registry for Liberty başlıklı konuya bakın.

6. `mqwebuser.xml` dosyasını düzenleyerek kullanıcılara ve gruplara roller atayın:

   Kullanıcılara ve gruplara IBM MQ Consoleve REST APIürününü kullanma yetkisi veren birkaç rol vardır. Her rol farklı bir erişim düzeyi verir. Daha fazla bilgi için bkz "IBM MQ Console ve REST API üzerindeki roller" sayfa 532.

   - Rolleri atamak ve IBM MQ Consoleerişimi vermek için kullanıcılarınızı ve gruplarınızı **<enterpriseApplication id="com.ibm.mq.console">** etiketleri içindeki uygun **security-role** etiketleri arasına ekleyin.
   - Rolleri atamak ve REST APIerişimi vermek için kullanıcılarınızı ve gruplarınızı **<enterpriseApplication id="com.ibm.mq.rest">** etiketleri içindeki uygun **security-role** etiketleri arasına ekleyin.

   security-role etiketleri içindeki kullanıcı ve grup bilgilerinin biçimiyle ilgili yardım için bkz. örnekler.
7. `mqwebuser.xml`içinde kullanıcılar için parola sağladıysanız, bu parolaları kodlayarak WebSphere Libertytarafından sağlanan **securityUtility encoding** komutunu kullanarak parolaları daha güvenli hale getirmeniz gerekir. Daha fazla bilgi için WebSphere Liberty ürün belgelerinde Liberty:securityUtility command başlıklı konuya bakın.

Örnek

Aşağıdaki örnekte, `MQWebAdminGroup` grubuna `MQWebAdmin`rolüyle IBM MQ Console erişimi verilir. `reader`adlı kullanıcıya `MQWebAdminRO`rolüyle erişim verilir ve `guest` adlı kullanıcıya `MQWebUser`rolü ile erişim verilir:

```
<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>
```

Aşağıdaki örnekte, `reader` ve `guest` kullanıcılarına IBM MQ Consoleerişimi verilir. `user` kullanıcısına REST APIerişimi verilir ve `MQAdmin` grubu içindeki kullanıcılara IBM MQ Console ve REST APIiçin erişim verilir. `mftadmin` kullanıcısına MFT için REST API erişimi verilir:

```
<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>
```

Sonraki adım

Kullanıcıların kimlik doğrulamasını seçin:

IBM MQ Console Kimlik Doğrulaması Seçenekleri

- Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda bir kullanıcı, IBM MQ Console oturum açma ekranına bir kullanıcı kimliği ve parola girer. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Bu kimlik doğrulama seçeneğini kullanmak için başka bir yapılandırma gerekmez, ancak isteğe bağlı olarak LTPA belirteci için süre bitimi aralığını yapılandırabilirsiniz. Daha fazla bilgi için LTPA belirteci süre bitim aralığını yapılandırmabaşlıklı konuya bakın.
- Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı IBM MQ Console' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.

REST API Kimlik Doğrulaması Seçenekleri

- Kullanıcıların HTTP temel kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı adı ve parola kodlanır, ancak şifrelenmez ve kullanıcının kimliğini doğrulamak ve bu istek için yetki vermek üzere her REST API isteğiyle birlikte gönderilir. Bu kimlik doğrulamasının güvenli olması için güvenli bir bağlantı kullanmanız gerekir. Yani, HTTPSkullanmalısınız. Daha fazla bilgi için bkz “REST API ile HTTP temel kimlik doğrulamasını kullanma” sayfa 541.
- Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı REST API `login` kaynağı için HTTP POST yöntemiyle bir kullanıcı kimliği ve parola sağlar. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Daha fazla bilgi için bkz “REST API ile belirteç tabanlı kimlik doğrulamasını kullanma” sayfa 542. LTPA belirteci için süre bitim aralığını yapılandırabilirsiniz. Daha fazla bilgi için LTPA simgesini yapılandırmabaşlıklı konuya bakın.
- Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı REST API' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.

ALW IBM MQ Console ve REST API için yerel bir işletim sistemi kayıt defterinin yapılandırılması

`mqwebuser.xml` dosyasında yerel bir işletim sistemi kaydı yapılandırabilirsiniz. Yerel işletim sistemindeki kullanıcı adları ve parolalar, IBM MQ Console ve REST APIkullanıcılarını doğrulamak ve yetkilendirmek için kullanılır.

Başlamadan önce

- Yerel işletim sistemi kimlik doğrulama özelliğiyle istemci sertifikası kimlik doğrulaması için, kullanıcı kimliği, istemci sertifikasının ayırt edici adından (DN) alınan ortak addır (CN). Kullanıcı kimliği işletim sistemi kullanıcısı olarak yoksa, istemci sertifikasında oturum açma başarısız olur ve parola tabanlı kimlik doğrulamasına geri döner.
- Bu görevi tamamlamak için, `mqwebuser.xml` dosyasını düzenlemek üzere yeterli ayrıcalığa sahip bir kullanıcı olmanız gerekir:

– V 9.3.5 Linux mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki `mqwebuser.xml` dosyasına yazma erişiminiz olmalıdır.
– mqweb sunucusu bir IBM MQ kuruluşunun parçasıysa, bir ayrıcalıklı kullanıcıolmanız gerekir.

Bu görev hakkında

Yerel bir işletim sistemi kayıt defteriyle, kullanıcılara ve gruplara otomatik olarak bir rol atanır:

- IBM i'mqm' grubunun ya da 'QMQMADM' grubunun bir parçası olan herhangi bir kullanıcıya `MQWebAdmin` ve `MFTWebAdmin` rolleri verilir.
- Diğer tüm kullanıcılara `MQWebUser` rolü verilir.

Bu rollerle ilgili daha fazla bilgi için bkz. “IBM MQ Console ve REST API üzerindeki roller” sayfa 532.

Yerel bir işletim sistemi kaydı yalnızca AIX, Linux, and Windowsüzerinde kullanılabilir. z/OS üzerinde bir SAF kayıt dosyası yapılandırılarak eşdeğer işlev sağlanır. Daha fazla bilgi için bkz “IBM MQ Console ve REST API için SAF kaydının yapılandırılması” sayfa 530.

Yordam

1. `local_os_registry.xml` örnek XML dosyasını aşağıdaki yollardan birinden kopyalayın:
   - V 9.3.5 Linux Bağımsız bir IBM MQ Web Server kuruluşunda: *MQWEB_INSTALLATION_PATH*`/web/mq/samp/configuration`

     Burada *MQWEB_INSTALLATION_PATH* , IBM MQ Web Server kuruluş dosyasının açıldığı dizindir.
   - IBM MQ kuruluşunda: *MQ_INSTALLATION_PATH*`/web/mq/samp/configuration`
2. Örnek dosyayı aşağıdaki dizinlerden birine yerleştirin:
   - V 9.3.5 Linux Bağımsız bir IBM MQ Web Server kuruluşunda: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb`

     Burada *MQ_OVERRIDE_DATA_PATH* , **MQ_OVERRIDE_DATA_PATH** ortam değişkeninin gösterdiği IBM MQ Web Server veri dizinidir.
   - IBM MQ kuruluşunda: *MQ_DATA_PATH*`/web/installations/`*installationName*`/servers/mqweb`
3. İsteğe bağlı: `mqwebuser.xml`içinde herhangi bir yapılandırma ayarını değiştirdiyseniz, bunları örnek dosyaya kopyalayın.
4. Var olan `mqwebuser.xml` dosyasını silin ve örnek dosyayı `mqwebuser.xml`olarak yeniden adlandırın.

Sonraki adım

Kullanıcıların kimlik doğrulamasını seçin:

IBM MQ Console Kimlik Doğrulaması Seçenekleri

- Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda bir kullanıcı, IBM MQ Console oturum açma ekranına bir kullanıcı kimliği ve parola girer. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Bu kimlik doğrulama seçeneğini kullanmak için başka bir yapılandırma gerekmez, ancak isteğe bağlı olarak LTPA belirteci için süre bitimi aralığını yapılandırabilirsiniz. Daha fazla bilgi için LTPA belirteci süre bitim aralığını yapılandırmabaşlıklı konuya bakın.
- Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı IBM MQ Console' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.

REST API Kimlik Doğrulaması Seçenekleri

- Kullanıcıların HTTP temel kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı adı ve parola kodlanır, ancak şifrelenmez ve kullanıcının kimliğini doğrulamak ve bu istek için yetki vermek üzere her REST API isteğiyle birlikte gönderilir. Bu kimlik doğrulamasının güvenli olması için güvenli bir bağlantı kullanmanız gerekir. Yani, HTTPSkullanmalısınız. Daha fazla bilgi için bkz “REST API ile HTTP temel kimlik doğrulamasını kullanma” sayfa 541.
- Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı REST API `login` kaynağı için HTTP POST yöntemiyle bir kullanıcı kimliği ve parola sağlar. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Daha fazla bilgi için bkz “REST API ile belirteç tabanlı kimlik doğrulamasını kullanma” sayfa 542. LTPA belirteci için süre bitim aralığını yapılandırabilirsiniz. Daha fazla bilgi için LTPA simgesini yapılandırmabaşlıklı konuya bakın.
- Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı REST API' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.

IBM MQ Console ve REST API için LDAP kaydı yapılandırılması

`mqwebuser.xml` dosyasında bir LDAP kaydı yapılandırabilirsiniz. LDAP kaydındaki kullanıcı adları ve parolalar, IBM MQ Console ve REST APIkullanıcılarını doğrulamak ve yetkilendirmek için kullanılır.

Başlamadan önce

- Bir LDAP kaydını yapılandırdığınızda, her kullanıcıya bir rol atamanız gerekir. Her rol, IBM MQ Console ve REST API' e erişmek için farklı ayrıcalık düzeyleri sağlar ve izin verilen bir işlem denendiğinde kullanılan güvenlik bağlamını belirler. Kaydı yapılandırmadan önce bu rolleri anlamanız gerekir. Rollerin her biri hakkında daha fazla bilgi için bkz. “IBM MQ Console ve REST API üzerindeki roller” sayfa 532.

  `MQWebUser` rolüne sahip herhangi bir kullanıcının yalnızca, kuyruk yöneticisinde gerçekleştirmesi için kullanıcı kimliği verilen işlemleri gerçekleştirebileceğini unutmayın. Bu nedenle, LDAP sunucusunda tanımlanan kullanıcı kimliği, IBM MQ ' in kurulu olduğu sistemde aynı kullanıcı kimliğine sahip olmalıdır. Bu kullanıcı kimlikleri aynı durumda olmalıdır; tersi durumda, kullanıcı kimlikleri arasındaki eşleme başarısız olabilir.
- Bu görevi tamamlamak için, `mqwebuser.xml` dosyasını düzenlemek üzere yeterli ayrıcalığa sahip bir kullanıcı olmanız gerekir:
  - z/OS z/OS' da `mqwebuser.xml` dosyasına yazma erişiminiz olmalıdır.
  - Multi Diğer tüm işletim sistemlerinde ayrıcalıklı kullanıcıolmanız gerekir.
  - V 9.3.5 Linux mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki `mqwebuser.xml` dosyasına yazma erişiminiz olmalıdır.

Yordam

1. `ldap_registry.xml` örnek XML dosyasını aşağıdaki yollardan birinden kopyalayın:
   - IBM MQ kuruluşunda:
     - ALW AIX, Linux, and Windowssistemlerinde: *MQ_INSTALLATION_PATH* `/web/mq/samp/configuration`
     - z/OS z/OSsistemlerinde: `PathPrefix /web/mq/samp/configuration`

       Burada `PathPrefix` , IBM MQ for z/OS UNIX System Services Components kuruluş yoludur.

- V 9.3.5 Linux Bağımsız bir IBM MQ Web Server kuruluşunda: *MQWEB_INSTALLATION_PATH*/web/mq/samp/configuration

  Burada *MQWEB_INSTALLATION_PATH* , IBM MQ Web Server kuruluş dosyasının açıldığı dizindir.

2. Örnek dosyayı uygun dizine yerleştirin:
   - IBM MQ kuruluşunda:
     - Linux AIX AIX ya da Linuxsistemlerinde: /var/mqm/web/installations/*installationName*/servers/mqweb
     - Windows Windowsüzerinde: *MQ_DATA_PATH*\web\installations*installationName*\servers\mqweb; burada *MQ_DATA_PATH* , IBM MQ veri yoludur. Bu yol, IBM MQkuruluşu sırasında seçilen veri yoludur. Varsayılan olarak bu yol C:\ProgramData\IBM\MQolur.
     - z/OS z/OSişletim tarihinde: *WLP_user_directory*/servers/mqweb

       Burada *WLP_user_directory* , **crtmqweb** komut dosyası mqweb sunucusu tanımlamasını oluşturmak için çalıştırıldığında belirtilen dizindir.
   - V 9.3.5 Linux Bağımsız bir IBM MQ Web Server kuruluşunda: *MQ_OVERRIDE_DATA_PATH*/web/installations/MQWEBINST/servers/mqweb

     Burada *MQ_OVERRIDE_DATA_PATH* , **MQ_OVERRIDE_DATA_PATH** ortam değişkeninin gösterdiği IBM MQ Web Server veri dizinidir.
3. İsteğe bağlı: mqwebuser.xmliçinde herhangi bir yapılandırma ayarını değiştirdiyseniz, bunları örnek dosyaya kopyalayın.
4. Var olan mqwebuser.xml dosyasını silin ve örnek dosyayı mqwebuser.xmlolarak yeniden adlandırın.
5. **ldapRegistry** ve **idsLdapFilterProperties** etiketleri içindeki LDAP kayıt ayarlarını değiştirmek için yeni mqwebuser.xml dosyasını düzenleyin.

   LDAP kayıtlarını yapılandırma hakkında daha fazla bilgi için WebSphere Liberty belgelerinde Configuring LDAP user registries in Liberty başlıklı konuya bakın.
6. mqwebuser.xml dosyasını düzenleyerek kullanıcılara ve gruplara roller atayın:

   Kullanıcılara ve gruplara IBM MQ Consoleve REST APIürününü kullanma yetkisi veren birkaç rol vardır. Her rol farklı bir erişim düzeyi verir. Daha fazla bilgi için bkz “IBM MQ Console ve REST API üzerindeki roller” sayfa 532.

   - Rolleri atamak ve IBM MQ Consoleerişimi vermek için kullanıcılarınızı ve gruplarınızı **<enterpriseApplication id="com.ibm.mq.console">** etiketleri içindeki uygun **security-role** etiketleri arasına ekleyin.
   - Rolleri atamak ve REST APIerişimi vermek için kullanıcılarınızı ve gruplarınızı **<enterpriseApplication id="com.ibm.mq.rest">** etiketleri içindeki uygun **security-role** etiketleri arasına ekleyin.

Sonraki adım

Kullanıcıların kimlik doğrulamasını seçin:

IBM MQ Console Kimlik Doğrulaması Seçenekleri

- Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda bir kullanıcı, IBM MQ Console oturum açma ekranına bir kullanıcı kimliği ve parola girer. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Bu kimlik doğrulama seçeneğini kullanmak için başka bir yapılandırma gerekmez, ancak isteğe bağlı olarak LTPA belirteci için süre bitimi aralığını yapılandırabilirsiniz. Daha fazla bilgi için LTPA belirteci süre bitim aralığını yapılandırmabaşlıklı konuya bakın.
- Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı IBM MQ Console' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz,

ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.

REST API Kimlik Doğrulaması Seçenekleri

- Kullanıcıların HTTP temel kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı adı ve parola kodlanır, ancak şifrelenmez ve kullanıcının kimliğini doğrulamak ve bu istek için yetki vermek üzere her REST API isteğiyle birlikte gönderilir. Bu kimlik doğrulamasının güvenli olması için güvenli bir bağlantı kullanmanız gerekir. Yani, HTTPSkullanmalısınız. Daha fazla bilgi için bkz “REST API ile HTTP temel kimlik doğrulamasını kullanma” sayfa 541.
- Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı REST API `login` kaynağı için HTTP POST yöntemiyle bir kullanıcı kimliği ve parola sağlar. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Daha fazla bilgi için bkz “REST API ile belirteç tabanlı kimlik doğrulamasını kullanma” sayfa 542. LTPA belirteci için süre bitim aralığını yapılandırabilirsiniz. Daha fazla bilgi için LTPA simgesini yapılandırmabaşlıklı konuya bakın.
- Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı REST API' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.

z/OS IBM MQ Console ve REST API için SAF kaydının yapılandırılması

Sistem Yetkilendirme Olanağı (SAF) arabirimi, mqweb sunucusunun kimlik doğrulama ve yetkilendirme denetimi için dış güvenlik yöneticisini çağırmasına olanak sağlar. Daha sonra bir kullanıcı, z/OS kullanıcı kimliği ve parolasıyla IBM MQ Console ve REST API içinde oturum açabilir.

Başlamadan önce

- Bir SAF kaydını yapılandırdığınızda, kullanıcılara bir rol atamanız gerekir. Her rol, IBM MQ Console ve REST API' e erişmek için farklı ayrıcalık düzeyleri sağlar ve izin verilen bir işlem denendiğinde kullanılan güvenlik bağlamını belirler. Kaydı yapılandırmadan önce bu rolleri anlamanız gerekir. Rollerin her biri hakkında daha fazla bilgi için bkz. “IBM MQ Console ve REST API üzerindeki roller” sayfa 532.
- SSF ' ye ilişkin yetkili arabirimi kullanmak için çalışan WebSphere Liberty Angel işlemi gerekir. Ek bilgi için bkz. Liberty üzerinde z/OS yetkili hizmetlerinin z/OS için etkinleştirilmesi .
- Bu görevi tamamlamak için, `mqwebuser.xml` kütüğüne yazma erişiminiz ve güvenlik yöneticisi tanıtımlarını tanımlama yetkinizin olması gerekir.

Not: V 9.3.5 V 9.3.0.20 IBM MQ 9.3.5 for Continuous Delivery ve IBM MQ 9.3.0 Fix Pack 20 for Long Term Supportürününden `zos_saf_registry.xml` örnek yapılandırma dosyası, yinelenen bir safAuthorization girişini kaldıracak şekilde güncellenmiştir.

Bu güncelleme, ICH408I hatasının IBM MQ Console on z/OS , WebSphere Liberty Profile 22.0.0.12 ya da sonraki bir düzeyle yükseltildiğinde ortaya çıkabileceği bir sorunu düzeltir: Long Term Support için IBM MQ 9.3.0 Fix Pack 2 ve Continuous Deliveryiçin IBM MQ 9.3.1 CSU 1 ve IBM MQ 9.3.2 . Birden çok safAuthorization deyiminin olması desteklenmez ve MQWebAdmin ya da MQWebAdminRO rollerinde olmayan kullanıcılar IBM MQ Consolearacılığıyla z/OS kuyruk yöneticisine erişmeye çalıştığında bir ICH408I hatasına neden olabilir.

Günlüğe kaydetme girişimi tiplerini belirten **racRouteLog**için varsayılan değer NONE' dır. Güvenlik denetimi için ek bir rapora ya da kayda gereksinim duyarsanız, daha fazla bilgi için SAF Yetkisi (safAuthorization) başlıklı konuya bakın.

Bu görev hakkında

SAF arabirimi, mqweb sunucusunun IBM MQ Console ve REST APIiçin kimlik doğrulama ve yetkilendirme denetimi için dış güvenlik yöneticisini çağırmasına olanak sağlar.

Yordam

1. mqweb sunucunuza z/OS yetkili hizmetlerini kullanma yetkisi vermek için Liberty üzerinde z/OS yetkili hizmetlerinin z/OS için etkinleştirilmesi içindeki adımları izleyin.

   Melek işlemini başlatmak için örnek JCL USS_ROOT/web/templates/zos/procs/bbgzangl.jcldizininde bulunur; burada USS_ROOT, z/OS UNIX bileşenlerinin kurulu olduğu z/OS UNIX System Services (z/OS UNIX) içindeki yoldur.

   bbgzangl.jcliçinde, SET ROOT deyimini USS_ROOT/webdeğerini gösterecek şekilde değiştirin; örneğin, /usr/lpp/mqm/V9R2M0/web.

   Melek sürecini durdurma ve başlatma hakkında daha fazla bilgi için bkz. Administering Liberty on z/OS .
2. Libertyiçin gerekli olan kimliği doğrulanmamış kullanıcıyı oluşturmak için Liberty: System Authorization Facility (SAF) kimliği doğrulanmamış kullanıcısını ayarlama başlıklı konudaki adımları izleyin.
3. zos_saf_registry.xml dosyasını şu yoldan kopyalayın: PathPrefix /web/mq/samp/configuration ; burada PathPrefix , z/OS UNIX Components kuruluş yoludur.
4. Örnek dosyayı *WLP_user_directory*/servers/mqweb dizinine yerleştirin; burada *WLP_user_directory* , **crtmqweb** komut dosyası mqweb sunucusu tanımlamasını oluşturmak için çalıştırıldığında belirtilen dizindir.
5. İsteğe bağlı: Daha önce mqwebuser.xmliçinde herhangi bir yapılandırma ayarını değiştirdiyseniz, bunları örnek dosyaya kopyalayın.
6. Var olan mqwebuser.xml dosyasını silin ve örnek dosyayı mqwebuser.xmlolarak yeniden adlandırın.
7. mqwebuser.xmliçindeki **safCredentials** öğesini özelleştirin.
   a. **profilePrefix** ' ı Liberty sunucunuz için benzersiz bir ada ayarlayın. Tek bir sistemde çalışan birden çok mqweb sunucunuz varsa, her sunucu için farklı bir ad seçmeniz gerekir; örneğin, MQWEB920 ve MQWEB915.
   b. **unauthenticatedUser** ayarını, "2" sayfa 531. adımda oluşturulan kimliği doğrulanmamış kullanıcının adına ayarlayın.
8. mqweb sunucusu APPLID değerini RACFolarak tanımlayın.

   APPLID kaynak adı, "7" sayfa 531. adımda **profilePrefix** özniteliğinde belirttiğiniz değerdir. Aşağıdaki örnek, RACFiçinde mqweb sunucusu APPLID değerini tanımlar:

   ```
   RDEFINE APPL profilePrefix UACC(NONE)
   ```
9. APPL sınıfındaki mqweb sunucusu APPLID için IBM MQ Console ya da REST API okuma erişimi için kimliği doğrulanacak tüm kullanıcılara ya da gruplara yetki verin.

   Bunu, "2" sayfa 531. adımda tanımlanan kimliği doğrulanmamış kullanıcı için de yapmalısınız. Aşağıdaki örnek, RACFiçinde bir kullanıcıya mqweb sunucusu APPLID 'sine okuma erişimi verir:

   ```
   PERMIT profilePrefix CLASS(APPL) ACCESS(READ) ID(userID)
   ```
10. Depolama alanı içindeki RACLISTed APPL sınıf profillerini yenilemek için **SETROPTS** RACF komutunu kullanın:

    ```
    SETROPTS RACLIST(APPL) REFRESH
    ```
11. Kullanıcılara IBM MQ Console ve REST APIiçindeki rollere erişim vermek için gereken EJBROLE sınıfındaki tanıtımları tanımlayın.

    Aşağıdaki örnek, RACFiçindeki profilleri tanımlar; burada **profilePrefix** , "7" sayfa 531adımında **profilePrefix** özniteliği için belirtilen değerdir.

```
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. Kullanıcılara IBM MQ Console ve REST APIiçindeki rollere erişim yetkisi verin.

    Bunu yapmak için, kullanıcılara ya da gruplara “11” sayfa 531. adımda oluşturulan EBJROLE sınıfındaki bir ya da daha fazla tanıtıma okuma erişimi verin. Roller hakkında daha fazla bilgi için bkz. “IBM MQ Console ve REST API üzerindeki roller” sayfa 532.

    Aşağıdaki örnek, RACFiçinde REST API için MQWebAdmin rolüne kullanıcı erişimi verir; burada **profilePrefix** , “7” sayfa 531adımında **profilePrefix** özniteliği için belirtilen değerdir.

```
PERMIT profilePrefix.com.ibm.mq.rest.MQWebAdmin CLASS(EJBROLE) ACCESS(READ) ID(userID)
```

Sonuçlar

IBM MQ Console ve REST APIiçin SAF kimlik doğrulamasını ayarladınız.

Sonraki adım

Kullanıcıların kimlik doğrulamasını seçin:

IBM MQ Console Kimlik Doğrulaması Seçenekleri

- Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda bir kullanıcı, IBM MQ Console oturum açma ekranına bir kullanıcı kimliği ve parola girer. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Bu kimlik doğrulama seçeneğini kullanmak için başka bir yapılandırma gerekmez, ancak isteğe bağlı olarak LTPA belirteci için süre bitimi aralığını yapılandırabilirsiniz. Daha fazla bilgi için LTPA belirteci süre bitim aralığını yapılandırmabaşlıklı konuya bakın.
- Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı IBM MQ Console' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.

REST API Kimlik Doğrulaması Seçenekleri

- Kullanıcıların HTTP temel kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı adı ve parola kodlanır, ancak şifrelenmez ve kullanıcının kimliğini doğrulamak ve bu istek için yetki vermek üzere her REST API isteğiyle birlikte gönderilir. Bu kimlik doğrulamasının güvenli olması için güvenli bir bağlantı kullanmanız gerekir. Yani, HTTPSkullanmalısınız. Daha fazla bilgi için bkz “REST API ile HTTP temel kimlik doğrulamasını kullanma” sayfa 541.
- Kullanıcıların belirteç kimlik doğrulamasını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, bir kullanıcı REST API `login` kaynağı için HTTP POST yöntemiyle bir kullanıcı kimliği ve parola sağlar. Kullanıcının belirli bir süre boyunca oturum açmasını ve yetkili kalmasını sağlayan bir LTPA belirteci oluşturulur. Daha fazla bilgi için bkz “REST API ile belirteç tabanlı kimlik doğrulamasını kullanma” sayfa 542. LTPA belirteci için süre bitim aralığını yapılandırabilirsiniz. Daha fazla bilgi için LTPA simgesini yapılandırmabaşlıklı konuya bakın.
- Kullanıcıların istemci sertifikalarını kullanarak kimlik doğrulaması yapmalarına izin verin. Bu durumda, kullanıcı REST API' da oturum açmak için kullanıcı kimliği ya da parola kullanmaz, ancak bunun yerine istemci sertifikasını kullanır. Daha fazla bilgi için bkz “REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma” sayfa 537.

IBM MQ Console ve REST API üzerindeki roller

Kullanıcılara ve gruplara IBM MQ Console ya da REST APIürününü kullanma yetkisi verdiğinizde, kullanıcılara ve gruplara şu rollerden birini atamanız gerekir: **MQWebAdmin**, **MQWebAdminRO**, **MQWebUser**, **MFTWebAdmin**ve **MFTWebAdminRO**. Her rol, IBM MQ Console ve REST API' e erişmek için farklı ayrıcalık düzeyleri sağlar ve izin verilen bir işlem denendiğinde kullanılan güvenlik bağlamını belirler.

Not: **MQWebUser** rolü dışında, kullanıcı kimliği büyük ve küçük harfe duyarlı değildir. Bu role ilişkin belirli gereksinimler için bkz. “MQWebUser” sayfa 533 .

MQWebAdmin
Bu role atanmış bir kullanıcı ya da grup, tüm denetim işlemlerini gerçekleştirebilir ve mqweb sunucusunu başlatmak için kullanılan işletim sistemi kullanıcı kimliğinin güvenlik bağlamı altında çalışır.

Bu role sahip bir kullanıcının ya da grubun şu REST hizmetlerine erişimi yok:

- MFTiçin REST API . Bu hizmetleri kullanmak için kullanıcı ya da gruba **MFTWebAdmin** ya da **MFTWebAdminRO** rolü de atanmalıdır.
- messaging REST API. messaging REST API' ı kullanmak için kullanıcıya **MQWebUser** rolü atanmalıdır.

MQWebAdminRO
Bu rol, IBM MQ Console ya da REST APIiçin salt okunur erişim verir. Bu role atanmış bir kullanıcı ya da grup aşağıdaki işlemleri gerçekleştirebilir:

- Kuyruklar ve kanallar gibi IBM MQ nesnelerindeki işlemleri görüntüleyin ve sorgulayın.
- Kuyruklardaki iletilere göz atın.

Bu role atanan bir kullanıcı ya da grup, mqweb sunucusunu başlatmak için kullanılan işletim sistemi kullanıcı kimliğinin güvenlik bağlamı altında çalışır.

Bu role sahip bir kullanıcının ya da grubun şu REST hizmetlerine erişimi yok:

- MFTiçin REST API . Bu hizmetleri kullanmak için kullanıcı ya da gruba **MFTWebAdmin** ya da **MFTWebAdminRO** rolü de atanmalıdır.
- messaging REST API. messaging REST API' ı kullanmak için kullanıcıya **MQWebUser** rolü atanmalıdır.

MQWebUser
Bu role atanan bir kullanıcı ya da grup, kullanıcı kimliğinin kuyruk yöneticisinde gerçekleştirmesine izin verilen herhangi bir işlemi gerçekleştirebilir. Örneğin:

- Kanallar gibi IBM MQ nesnelerinde işlemleri başlatın ve durdurun.
- Kuyruklar ve kanallar gibi IBM MQ nesneleri üzerinde işlemleri tanımlayın ve ayarlayın.
- Kuyruklar ve kanallar gibi IBM MQ nesnelerindeki işlemleri görüntüleyin ve sorgulayın.
- messaging REST APIkullanarak iletileri koyun ve alın.

Bu role atanan bir kullanıcı ya da grup, birincil kullanıcının güvenlik bağlamı altında çalışır ve yalnızca, kuyruk yöneticisinde gerçekleştirmek için kullanıcı kimliğinin verildiği işlemleri gerçekleştirebilir.

Bu nedenle, kullanıcının herhangi bir işlem gerçekleştirebilmesi için, mqweb kullanıcı kaydında tanımlanan kullanıcıya ya da gruba IBM MQ içinde yetki verilmesi gerekir. Bu rolü kullanarak, hangi kullanıcıların IBM MQ Console ve REST APIkaynaklarını kullanırken belirli IBM MQ kaynaklarına hangi tip erişiminin kullanılacağını ayrıntılı olarak denetleyebilirsiniz.

Not:

- Bu rolün atandığı kullanıcı kimliği uzunluğu üst sınırı 12 karakterdir.
- Kullanıcı kimliğinin durumu, mqweb kullanıcı kaydında ve IBM MQ sisteminde aynı olmalıdır. Kullanıcı kimliğinin durumu farklıysa, kullanıcının kimliği IBM MQ Console ve REST API tarafından doğrulanabilir, ancak IBM MQ kaynaklarını kullanma yetkisi olmayabilir.

MFTWebAdmin
Bu role atanmış bir kullanıcı ya da grup, tüm MFT REST işlemlerini gerçekleştirebilir ve mqweb sunucusunu başlatmak için kullanılan işletim sistemi kullanıcı kimliğinin güvenlik bağlamı altında çalışır.

Bu role sahip bir kullanıcının ya da grubun IBM MQ REST API hizmetlerinin hiçbirine erişimi yoktur. Bu hizmetleri kullanmak için kullanıcıya ya da gruba **MQWebAdmin**, **MQWebAdminRO**ya da **MQWebUser** rolü de atanmalıdır.

MFTWebAdminRO

Bu rol, MFT için REST API ' e salt okunur erişim verir. Bu role atanan bir kullanıcı ya da grup, liste aktarımı ve liste aracıları gibi salt okunur işlemleri (GET istekleri) gerçekleştirebilir.

Bu role atanan bir kullanıcı ya da grup, mqweb sunucusunu başlatmak için kullanılan işletim sistemi kullanıcı kimliğinin güvenlik bağlamı altında çalışır.

Bu role sahip bir kullanıcının ya da grubun IBM MQ REST API hizmetlerinin hiçbirine erişimi yoktur. Bu hizmetleri kullanmak için kullanıcıya ya da gruba **MQWebAdmin**, **MQWebAdminRO**ya da **MQWebUser** rolü de atanmalıdır.

Kullanıcıları ve grupları bu rolleri kullanacak şekilde yapılandırma hakkında daha fazla bilgi için bkz. “Kullanıcıları ve rolleri yapılandırma” sayfa 522.

Çakışan roller

Bir kullanıcıya ya da gruba birden çok rol atanabilir. Bir kullanıcı bu durumda bir işlem gerçekleştirdiğinde, işlem için geçerli olan en yüksek ayrıcalık rolü kullanılır. Örneğin, **MQWebAdminRO** ve **MQWebUser** rollerine sahip bir kullanıcı bir sorgu kuyruğu işlemi gerçekleştirirse, **MQWebAdminRO** rolü kullanılır ve işlem, web sunucusunu başlatan sistem kullanıcı kimliği bağlamı altında denenir. Aynı kullanıcı bir tanımlama işlemi gerçekleştirirse, **MQWebUser** rolü kullanılır ve işlem birincil kullanıcının bağlamı altında denenir.

ALW IBM MQ Console tarafından sağlanan sertifikayı tarayıcınızla değiştirme

IBM MQ Console ' i kimlik doğrulama amacıyla kendi CA imzalı sertifikanızı sunacak şekilde yapılandırabilirsiniz. Bunu yaptığınızda, IBM MQ Console konsoluna erişirken bir web tarayıcısı tarafından sunulan kendinden onaylı sertifika uyarısı kaldırılır.

Başlamadan önce

IBM MQ Consoleürününü kullanma yetkisi olacak kullanıcıları, grupları ve rolleri yapılandırın. Daha fazla bilgi için bkz “Kullanıcıları ve rolleri yapılandırma” sayfa 522.

Bu görev hakkında

Konsol güvenliği, IBM MQ kuruluşunuz tarafından kullanılan bir IBM WebSphere Application Server Liberty tarafından sağlanır.

Bu sunucu tarafından tarayıcınıza sunulan sertifikayı değiştirmek için aşağıdakileri yapmanız gerekir:

1. Web sunucusu anahtar deposuna sunmak istediğiniz sertifikayı ekleyin.
2. Sertifikayı etiketleyin.
3. Varsayılan güvenlik yapılandırmasını kapatmak için `mqwebuser.xml` dosyasını düzenleyin.
4. `mqwebuser.xml` dosyasında kendi güvenlik yapılandırmanızı açın ve sunmak istediğiniz sertifikayı belirtin.

Yordamda aşağıdakiler olduğu varsayılır:

- AIX, Linux, and Windows sisteminin kullanılması.
- Ayrıcalıklı kullanıcı.

Notlar:

- Aşağıdaki örnek, bir Linux makinesinde verilen komutları kullanarak kendinden onaylı bir sertifika oluşturur ve kullanır; yani, bir Windows makinesinde kullanılan **dir** yerine **ls**.
- Bu size kavramı gösterir, ancak tarayıcı uyarısını kaldırmaz.
- Tarayıcı uyarısını kaldırmak için CA imzalı bir sertifika sağlamanız gerekir.

Yordam

1. Liberty sunucusu çalışıyorsa, komut satırına **endmqweb** komutunu girerek sunucuyu durdurun.
2. Sertifikanızı, Liberty uygulama sunucusunun kullandığı anahtar deposuna ekleyin; böylece sertifikayı bulup web tarayıcınıza sunabilir.
   a) Aşağıdaki komutu vererek anahtar deposu yerine gidin ve çıkışı listeleyin:

```
cd /var/mqm/web/installations/Installation1/servers/mqweb/resources/security
ls
```

   Örneğin, `key.jks`adlı anahtar deposunu görüntüleyen aşağıdaki çıkışı görürsünüz:

```
/var/mqm/web/installations/Installation1/servers/mqweb/resources/security$
ls key.jks ltpa.keys
```

   b) Kendinden onaylı sertifika yarat:

   `password`parolasıyla `key.jks` ' e eklenen, eğitim amacıyla kendinden imzalı bir sertifika oluşturmak için aşağıdaki komutu verin:

```
runmqckm -cert -create -db key.jks -pw password -dn
"cn=QueueManager,o=IBM,c=UK" -label myowncertificate
```

   -dn işareti, sertifikanız üzerinde görüntülenen değerleri belirtmenizi sağlar.

   c) Aşağıdaki komutu vererek sertifikayı başarıyla eklediğinizi doğrulayın:

```
runmqckm -cert -list -db key.jks -pw password
```

   Örneğin, sertifikanın etiketiyle birlikte sunucunun kullanmakta olduğu `default` etiketli sertifikayla birlikte eklendiğini gösteren aşağıdaki çıkışı görürsünüz:

```
/var/mqm/web/installations/Installation1/servers/mqweb/resources/security
$ runmqckm -cert -list -db key.jks -pw password
Certificates in database /var/mqm/web/installations/Installation1/servers/mqweb/resources/
security/key.jks
   default
   myown certificate
```

3. Sunucunun yeni sertifikayı sağlamasını sağlamak için `mqwebuser.xml` dosyasını düzenleyin.
   a) `mqwebuser.xml` dosyasının konumuna gidin ve bu dosyayı düzenlemek için seçtiğiniz bir metin düzenleyicisinde açın; bu durumda *nano*

```
cd /var/mqm/web/installations/Installation1/servers/mqweb
nano mqwebuser.xml
```

   b) Varsayılan güvenlik yapılandırmasını kapatın.

   Kod satırının başına `<!--` ve kod satırının sonuna `-->` ekleyerek aşağıdaki satırı açıklama satırı yapın:

```
<!--
<sslDefault sslRef="mqDefaultSSLConfig"/>
-->
```

   c) Kendi yapılandırmanızı etkinleştirin ve belirtin.

   Bunu yapmak için aşağıdaki yordamı gerçekleştirin:

   i) Kod öbeğinin başından `<!--` ve kod öbeğinin sonundan `-->` öğesini kaldırarak aşağıdaki kod satırlarını açıklama satırı olmaktan kaldırın.

```
<!--
<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"
```

```
serverKeyAlias="default" trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"/>
<sslDefault sslRef="thisSSLConfig"/>
-->
```

ii) **Kod öbeğinin ilk satırını değiştirmeyin** ; bu satır, konsolun kişisel sertifikalarını saklamak için kullandığı anahtar deposunu belirtir.

iii) **Kod öbeğinin ikinci satırını açıklama satırı yapın**; bu satır, konsolun istemci sertifikalarını arayacağı güvenli depoyu belirtir. Simge kimlik doğrulamasını kullandığınızda, bir güvenli depo oluşturmadığınız için, kod satırının açık bırakılması konsol başlatıldığında bir hataya neden olur.

iv) **serverKeyAlias= "default" değerini, kod öbeğinin üçüncü satırında serverKeyAlias= "myowncertificate"** olarak değiştirin ve diğer her şeyi aynı bırakın.

v) **Kod öbeğinin son satırını değiştirmeyin** ; bu, sunucuya az önce belirttiğiniz yapılandırmayı kullanmasını bildirir.

Kod bloğu şu şekilde görünüyor:

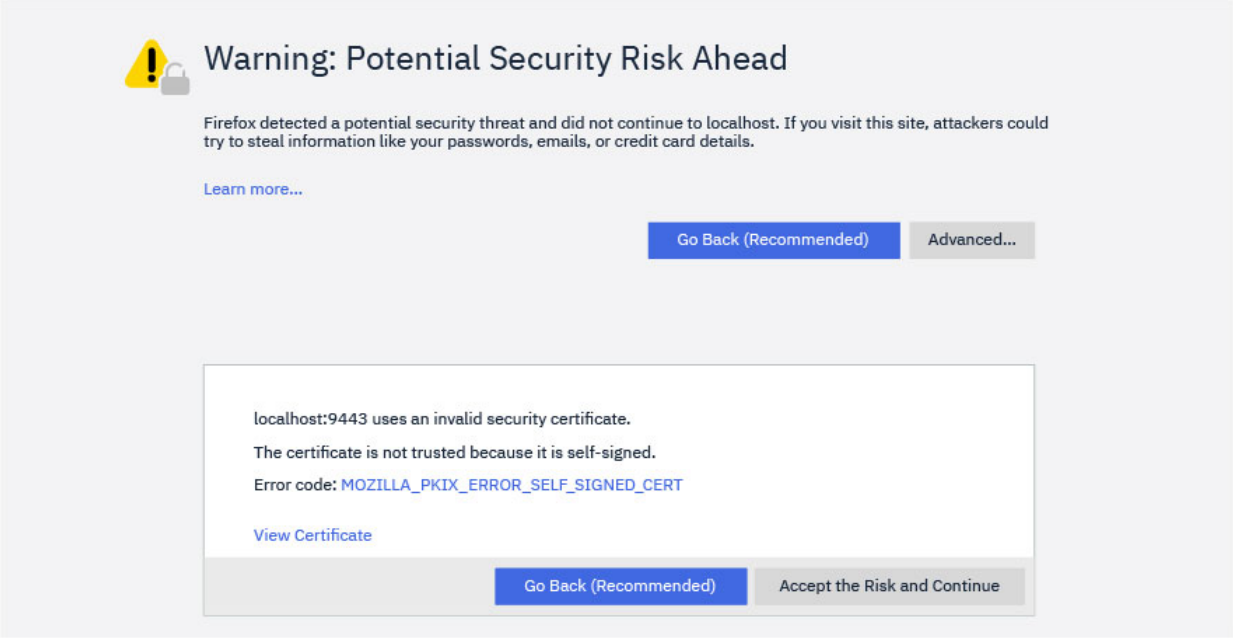
```
<keyStore id="defaultKeyStore" location="key.jks" type="JKS" password="password"/>
<!-- Commenting out the defaultTrustStore as otherwise we get errors (viewable in the messages.log file
in the logs folder) j
<keyStore id="defaultTrustStore" location="trust.jks" type="JKS" password="password"/>
-->
<ssl id="thisSSLConfig" clientAuthenticationSupported="true" keyStoreRef="defaultKeyStore"
serverKeyAlias="myowncertificate" trustStoreRef="defaultTrustStore" sslProtocol="TLSv1.2"/>
<sslDefault sslRef="thisSSLConfig"/>
```

4. **strmqweb** komutunu kullanarak web sunucusunu yeniden başlatın.

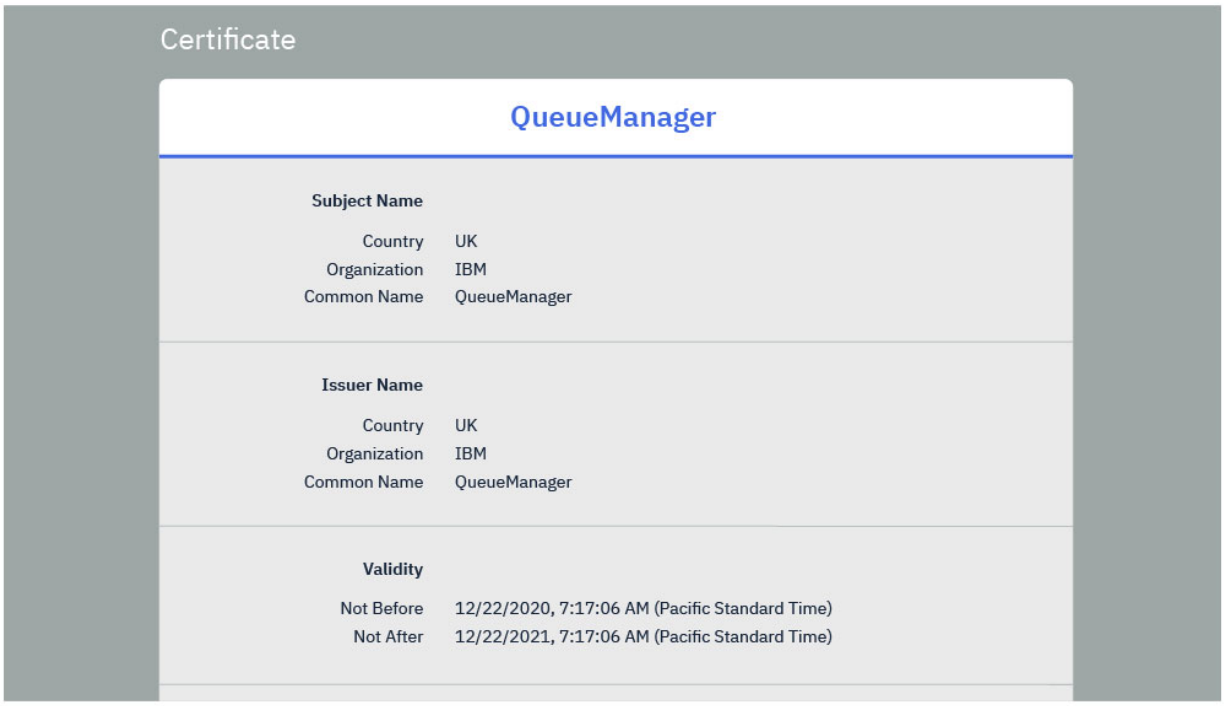
Sonuçlar

Web sunucusu başladığında, IBM MQ Console ' e göz atın ve yenileyin. Oluşturduğunuz kendinden onaylı bir sertifika kullanıyorsanız, “2” sayfa 535 ve “3” sayfa 535adımlarında önceki metinde açıklanan yordamı kullanarak bir güvenlik uyarısı görürsünüz.

Bu uyarının biçiminin kullandığınız tarayıcıya bağlı olduğunu unutmayın.



Sertifikayı Görüntüledüğmesini tıklatırsanız, “2.b” sayfa 535. adımda sertifikayı oluşturduğunuzda **-dn** işaretinde sağladığınız ayrıntılara sahip olduğunu görürsünüz.



Ancak, CA imzalı bir sertifika kullanıyorsanız, aşağıdaki komutu vererek eklediğiniz tarayıcınız güvenir:

```
runmqckm -cert -add -db key.jks -pw password -label myCACertificate
```

Burada `myCACertificate` , doğrudan oturum açma sayfasına yönlendirildiğiniz CA sertifikanızı içeren dosyanın dosya yoludur.

Uyarı: CA imzalı bir sertifika kullanıyorsanız ve sertifika kuruluşu (CA) sertifikası bir sertifika zincirinin parçasıysa, kök sertifika kuruluşu (CA) sertifikasıyla başlayarak tüm sertifikaları zincire eklemeniz gerekir. Ek bilgi için bkz. “AIX, Linux, and Windows üzerindeki bir anahtar havuzuna CA sertifikası ya da kendinden onaylı bir sertifikanın genel kısmını ekleme” sayfa 314 .

ALW REST API ve IBM MQ Console ile istemci sertifikası kimlik doğrulamasını kullanma

IBM MQ Console ve REST API kullanıcılarının kimliğini doğrulamak için istemci sertifikalarını birincil kullanıcılarla eşleyebilirsiniz.

Başlamadan önce

- Kullanıcıları, grupları ve rolleri, IBM MQ Console ve REST APIürününü kullanma yetkisine sahip olacak şekilde yapılandırın. Daha fazla bilgi için bkz “Kullanıcıları ve rolleri yapılandırma” sayfa 522.
- REST APIkomutunu kullandığınızda, `login` kaynağında HTTP GET yöntemini kullanarak geçerli kullanıcının kimlik bilgilerini sorgulayabilirsiniz; bu, isteğin kimliğini doğrulamak için istemci sertifikasını sağlar. Bu istek, kullanıcı adı ve kullanıcının atandığı rollerle ilgili bilgileri döndürür. Daha fazla bilgi için bkz. GET `/login`.
- Kullanıcıların kimliğini doğrulamak için istemci sertifikalarını birincil kullanıcılarla eşlediğinizde, istemci sertifikasının ayırt edici adı, yapılandırılan kullanıcı kaydındaki kullanıcılarla eşleşmek için kullanılır:
  - Temel bir kayıt için, Ortak Ad (CN) kullanıcıyla eşleştirilir. Örneğin, `CN=Fred, O=IBM, C=GB` , `Fred`kullanıcı adıyla eşleştirilir.

– Bir LDAP kaydı için, varsayılan olarak tam ayırt edici ad LDAP ile eşleştirilir. Eşleştirmeyi özelleştirmek için süzgeçler ve eşleme ayarlayabilirsiniz. Daha fazla bilgi için WebSphere Liberty belgelerinde Liberty :LDAP certificate map mode başlıklı konuya bakın.

Bu görev hakkında

Bir kullanıcı istemci sertifikası kullanarak kimlik doğrulaması yaptığında, sertifika kullanıcı adı ve parola yerine kullanılır. REST APIiçin istemci sertifikası, kullanıcının kimliğini doğrulamak için her REST isteğiyle birlikte sağlanır. IBM MQ Consoleiçin, bir kullanıcı bir sertifikayla oturum açtığında kullanıcının oturumu kapatılamaz.

Yordam aşağıdaki bilgileri alır:

- `mqwebuser.xml` dosyanız aşağıdaki örneklerden birini temel alır:
  - `basic_registry.xml`
  - `local_os_registry.xml`
  - `ldap_registry.xml`
- Bir AIX, Linux, and Windows sistemi kullandığınızı.
- Ayrıcalıklı kullanıcısınız.

İstemci sertifikası kimlik doğrulamasını z/OSüzerinde bir RACF anahtarlığı ile yapılandırmak için “z/OS üzerinde REST API ve IBM MQ Console için TLS ' nin yapılandırılması” sayfa 550başlıklı konudaki yordamı izleyin.

Not: Aşağıdaki yordamda, IBM MQ Console ve REST APIile istemci sertifikalarını kullanmak için gerekli adımlar açıklanmaktadır. Geliştirici kolaylığı için, adımlarda kendinden imzalı sertifikaların nasıl oluşturulacağı ve kullanılacağı ayrıntılarıyla açıklanmıştır. Ancak, üretim için bir sertifika yetkilisinden alınan sertifikaları kullanın.

Yordam

1. Komut satırında **strmqweb** komutunu girerek mqweb sunucusunu başlatın.
2. İstemci sertifikası yarat:
   a) PKCS#12 anahtar deposu yaratın:
      i) Komut satırına **strmqikm** komutunu girerek IBM Key Management aracını açın.
      ii) IBM Key Management aracındaki **Key Database File** (Anahtar Veritabanı Dosyası) menüsünden **New**(Yeni) düğmesini tıklatın.
      iii) **Anahtar veritabanı tipi** listesinde **PKCS12** öğesini seçin.
      iv) Anahtar deposunu saklamak için bir yer seçin ve **Dosya Adı** alanına uygun bir ad girin. Örneğin, `user.p12`
      v) İstendiğinde bir parola belirleyin.
   b) Kendinden onaylı bir sertifika yaratarak ya da bir sertifika yetkilisinden sertifika alarak sertifikayı yaratın:
      - Kendinden onaylı sertifika yarat:
        i) **Kendinden Onaylı Yeni**seçeneğini tıklatın.
        ii) **Anahtar Etiketi** alanına `user` girin.
        iii) Temel kullanıcı kaydı kullanıyorsanız, **Ortak Ad** alanına kullanıcı kaydınızdaki bir kullanıcının adını girin. Örneğin, `mqadmin`. Bir LDAP kullanıcı kaydı için, sertifikanın ayırt edici adının LDAP kaydındaki ayırt edici adla eşleştiğinden emin olun.
        iv) **Tamam**'ı tıklatın.
      - Sertifika yetkilisinden bir sertifika alın. CA sertifikası, ayırt edici ad (DN) alanının ortak adı (CN) içinde uygun kullanıcı adını içermelidir:

i) Yeni bir sertifika isteyin. **Oluştur** menüsünden **Yeni Sertifika İsteği**seçeneğini tıklatın.

ii) **Anahtar Etiketi** alanına sertifika etiketini girin.

iii) Temel bir kullanıcı kaydı kullanıyorsanız, **Ortak Ad** alanına sertifikanın ait olduğu kullanıcının kullanıcı adını girin.

Yerel bir işletim sistemi kaydı kullanıyorsanız, **Ortak Ad** alanının yerel işletim sistemi kullanıcı kimliğiyle eşleşmesi gerekir.

Bir LDAP kullanıcı kaydı için, sertifikanın ayırt edici adının LDAP kaydındaki ayırt edici adla eşleştiğinden emin olun.

iv) Uygun olduğu şekilde, kalan alanlar için değer yazın ya da seçin.

v) Sertifika isteğinin nereye kaydedileceğini ve sertifika isteğinin dosya adını seçin ve **Tamam**' ı tıklatın.

vi) Sertifika isteği dosyasını bir sertifika yetkilisine (CA) gönderin.

vii) CA ' dan sertifikanız varsa, komut satırına **strmqikm** komutunu girerek IBM Key Management aracını açın.

viii) IBM Key Management aracındaki **Key Database File** (Anahtar Veritabanı Dosyası) menüsünden **Open**(Aç) seçeneğini tıklatın.

ix) İstemci sertifikasını tutan PKCS#12 anahtar deposunu seçin. Örneğin, `user.p12`

x) **Al**'ı tıklatın, uygun sertifikayı seçin ve **Tamam**' ı tıklatın.

3. İstemci sertifikasının genel kısmını çıkarın:
   a) Komut satırına **strmqikm** komutunu girerek IBM Key Management aracını açın.
   b) IBM Key Management aracındaki **Key Database File** (Anahtar Veritabanı Dosyası) menüsünden **Open**(Aç) seçeneğini tıklatın.
   c) İstemci sertifikasını tutan PKCS#12 anahtar deposunu seçin. Örneğin, `user.p12`
   d) IBM Key Management aracındaki sertifika listesinden istemci sertifikasını seçin.
   e) **Sertifikayı Çek**düğmesini tıklatın.
   f) Sertifikayı kaydetmek için bir konum seçin ve **Sertifika dosyası adı** alanına uygun bir dosya adı girin. Örneğin, `user.arm`.
4. Sunucunun istemci sertifikasını doğrulayabilmesi için istemci sertifikasının genel kısmını imzalayıcı sertifikası olarak mqweb sunucusu güven anahtar deposuna aktarın:
   a) Önceden yoksa, mqweb sunucusu tarafından kullanılmak üzere bir `trust.jks` anahtar deposu yaratın:
      i) IBM Key Management aracındaki **Key Database File** (Anahtar Veritabanı Dosyası) menüsünden **New**(Yeni) düğmesini tıklatın.
      ii) **Anahtar veritabanı tipi** listesinden **JKS** seçeneğini belirleyin.
      iii) **Göz At** düğmesini tıklatın ve şu sayfaya gidin: `MQ_DATA_DIRECTORY/web/installations/installationName/servers/mqweb/resources/security`.

      Bu dizin zaten bir `key.jks` dosyası içermelidir. Bir `trust.jks` dosyası zaten varsa, üzerine yazmak yerine var olan dosyayı açın.

      iv) **Dosya Adı** alanına `trust.jks` girin.
      v) İstendiğinde bir parola belirleyin.
   b) Açılan menüden **İmzalayıcı Sertifikaları**' nı seçin.
   c) **Ekle**'yi tıklatın.
   d) Uygun kol dosyasını seçin ve **Tamam**düğmesini tıklatın. Örneğin, `user.arm`seçeneğini belirleyin.
   e) Sertifika için bir etiket girin.
5. mqweb sunucusu anahtar deposunun parolasını değiştirin:
   a) **Anahtar Veritabanı Dosyası** menüsünden **Aç**' ı tıklatın.

b) **Anahtar veritabanı tipi** listesinden **JKS** seçeneğini belirleyin.

c) **Göz At** düğmesini tıklatın ve *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb/resources/security seçeneğine gidin

d) key.jks anahtar deposunu seçin ve **Aç**' ı tıklatın.

e) İstendiğinde parolayı girin. Varsayılan parola: password.

f) **Anahtar Veritabanı Dosyası** menüsünden **Parolayı Değiştir**seçeneğini tıklatın.

g) Anahtar deposu için yeni bir parola girin.

6. mqwebuser.xml dosyasında istemci sertifikası kimlik doğrulamasını etkinleştir:

   mqwebuser.xml dosyası şu yolda bulunabilir: *MQ_DATA_PATH*/web/installations/*installationName*/servers/mqweb

   a) mqwebuser.xml dosyasında istemci sertifikası kimlik doğrulamasını etkinleştiren bölümü açıklama satırı olmaktan kaldırın. Bölüm aşağıdaki metni içerir:

   ```
   <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"/>
   ```

   b) **serverKeyDiğer Adı** değerinin, sunucu sertifikasının adıyla eşleştiğini doğrulayın. Varsayılan sunucu sertifikasını kullanıyorsanız, değer doğrudur.

   c) defaultKeyStore için **password** değerini, key.jks anahtar deposuna ilişkin parolanın kodlanmış bir sürümüyle değiştirin:

      i) *MQ_INSTALLATION_PATH*/web/bin dizininde, komut satırında şu komutu girin:

      ```
      securityUtility encode password
      ```

      ii) Bu komutun çıkışını defaultKeyStoreiçin **parola** alanına yerleştirin.

   d) defaultTrustStore için **parola** değerini, trust.jks anahtar deposunun parolasıyla eşleşecek şekilde değiştirin:

      i) *MQ_INSTALLATION_PATH*/web/bin dizininde, komut satırında şu komutu girin:

      ```
      securityUtility encode password
      ```

      ii) Bu komutun çıkışını defaultTrustStoreiçin **parola** alanına yerleştirin.

   e) mqwebuser.xml dosyasından aşağıdaki satırı kaldırın ya da çıkarın:

   ```
   <sslDefault sslRef="mqDefaultSSLConfig"/>
   ```

7. Komut satırında **endmqweb** komutunu girerek mqweb sunucusunu durdurun.
8. Komut satırında **strmqweb** komutunu girerek mqweb sunucusunu başlatın.
9. Kimlik doğrulamak için istemci sertifikasını kullanın:

   - İstemci sertifikasını IBM MQ Consoleile kullanmak için istemci sertifikasını IBM MQ Console' e erişmek için kullanılan web tarayıcısına kurun. Örneğin, user.p12 istemci sertifikasını kişisel sertifika olarak kurun.
   - İstemci sertifikasını REST APIile kullanmak için istemci sertifikasını her REST isteğiyle birlikte sağlayın. HTTP POST, PATCH ya da DELETE yöntemlerini kullandığınızda, siteler arası istek sahteciliği saldırılarını önlemek için istemci sertifikasıyla ek kimlik doğrulaması sağlamanız gerekir. Yani, isteği doğrulamak için kullanılan kimlik bilgilerinin kimlik bilgilerinin kimlik bilgileri sahibi tarafından kullanıldığını doğrulamak için ek kimlik doğrulaması kullanılır.

     Bu ek kimlik doğrulaması ibm-mq-rest-csrf-token HTTP üstbilgisi tarafından sağlanır. ibm-mq-csrf-token üstbilgisinin değerini boşluk da içinde olmak üzere herhangi bir değere ayarlayın ve isteği gönderin.

Örnek

Önemli: Örnekte, tüm cURL uygulamaları kendinden imzalı sertifikaları desteklemez, bu nedenle bunu yapan bir cURL uygulaması kullanmanız gerekir.

Aşağıdaki cURL örneği, istemci sertifikası kimlik doğrulamasıyla `QM1`kuyruk yöneticisinde `Q1`yeni bir kuyruğun nasıl yaratılacağını göstermektedir. Bu cURL komutunun tam yapılandırması, cURL ' nin oluşturulduğu kitaplıklara bağlıdır. Bu örnek, OpenSSL' ye karşı oluşturulan cURL ile bir Windows sistemine dayalıdır.

- Kuyruk kaynağıyla HTTP POST yöntemini kullanın, istemci sertifikasıyla kimlik doğrulaması gerçekleştirin ve isteğe bağlı bir değere sahip `ibm-mq-rest-csrf-token` HTTP üstbilgisini dahil edin. Bu değer, boşluk da içinde olmak üzere herhangi bir değer olabilir. `--cert-type` işareti, sertifikanın bir PKCS#12 sertifikası olduğunu belirtir. `--cert` işareti, sertifikanın yerini ve ardından iki nokta üst üste,: ve sonra sertifikaya ilişkin parolayı belirtir:

```
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\"}"
```

REST API ile HTTP temel kimlik doğrulamasını kullanma

REST API kullanıcıları, HTTP üstbilgisinde kullanıcı kimliklerini ve parolalarını sağlayarak kimlik doğrulayabilir. Bu kimlik doğrulama yöntemini POST, PATCH ve DELETE gibi HTTP yöntemleriyle kullanmak için `ibm-mq-rest-csrf-token` HTTP üstbilgisinin yanı sıra bir kullanıcı kimliği ve parola da sağlanmalıdır.

Başlamadan önce

- REST APIürününü kullanma yetkisi olacak kullanıcıları, grupları ve rolleri yapılandırın. Daha fazla bilgi için bkz “Kullanıcıları ve rolleri yapılandırma” sayfa 522.
- HTTP temel kimlik doğrulamasının etkinleştirildiğinden emin olun. Aşağıdaki XML ' in `mqwebuser.xml` dosyasında var olduğunu ve yorum yapılmadığını doğrulayın. Bu XML, `<featureManager>` etiketleri içinde olmalıdır:

```
<feature>basicAuthenticationMQ-1.0</feature>
```

z/OS z/OSüzerinde, bu dosyayı düzenlemek için `mqwebuser.xml` ' e yazma erişimi olan bir kullanıcı olmanız gerekir.

Multi Diğer tüm işletim sistemlerinde, `mqwebuser.xml` dosyasını düzenlemek için ayrıcalıklı kullanıcı olmanız gerekir.

- REST istekleri gönderirken güvenli bir bağlantı kullandığınızdan emin olun. Kullanıcı adı ve parola birleşimi kodlandığından, ancak şifrelenmediğinden, REST APIile HTTP temel kimlik doğrulamasını kullanırken güvenli bir bağlantı (HTTPS) kullanmanız gerekir.
- `login` kaynağında HTTP GET yöntemini kullanarak, isteği doğrulamak için temel kimlik doğrulama bilgilerini sağlayarak geçerli kullanıcının kimlik bilgilerini sorgulayabilirsiniz. Bu istek, kullanıcı adı ve kullanıcının atandığı rollerle ilgili bilgileri döndürür. Daha fazla bilgi için bkz. `GET /login`.

Yordam

1. Kullanıcı adını iki nokta ile ve parolayla birleştirin. Kullanıcı adının büyük ve küçük harfe duyarlı olduğunu unutmayın.

   Örneğin, admin kullanıcı adı ve admin parolası şu dizgi olur:

   ```
   admin:admin
   ```

2. Bu kullanıcı adı ve parola dizgisini base64 kodlamasında kodlayın.

3. Bu kodlanmış kullanıcı adını ve parolayı HTTP `Authorization: Basic` üstbilgisine ekleyin.

   Örneğin, kodlanmış bir yönetici kullanıcı adı ve bir yönetici parolası ile aşağıdaki üstbilgi oluşturulur:

   ```
   Authorization: Basic YWRtaW46YWRtaW4=
   ```

4. HTTP POST, PATCH ya da DELETE yöntemlerini kullandığınızda, kullanıcı adı ve parolanın yanı sıra ek kimlik doğrulaması da sağlamanız gerekir.

   Bu ek kimlik doğrulaması `ibm-mq-rest-csrf-token` HTTP üstbilgisi tarafından sağlanır. `ibm-mq-rest-csrf-token` HTTP üstbilgisi istekte bulunmalıdır, ancak değeri boşluk da içinde olmak üzere herhangi bir şey olabilir.

5. REST isteğinizi uygun üstbilgilerle IBM MQ adresine gönderin.

Örnek

Aşağıdaki örnek, Windows sistemlerinde temel kimlik doğrulamasıyla QM1kuyruk yöneticisinde yeni bir kuyruk Q1yaratılmasını göstermektedir. Örnekte cURLkullanılmıştır:

- Kuyruk kaynağıyla birlikte HTTP POST yöntemini kullanın, temel kimlik doğrulamasıyla kimlik doğrulaması gerçekleştirin ve isteğe bağlı bir değere sahip `ibm-mq-rest-csrf-token` HTTP üstbilgisini dahil edin. Bu değer, boşluk da içinde olmak üzere herhangi bir değer olabilir:

  ```
  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\"}"
  ```

REST API ile belirteç tabanlı kimlik doğrulamasını kullanma

REST API kullanıcıları, HTTP POST yöntemiyle REST API `login` kaynağına bir kullanıcı kimliği ve parola sağlayarak kimlik doğrulayabilir. Kullanıcının gelecekteki istekleri doğrulamasını sağlayan bir LTPA belirteci oluşturulur. Bu LTPA simgesi `LtpaToken2`önekine sahiptir. Kullanıcı, HTTP DELETE yöntemini kullanarak oturumu kapatabilir ve yürürlükteki kullanıcının oturum açma bilgilerini HTTP GET yöntemiyle sorgulayabilir.

Başlamadan önce

- REST APIürününü kullanma yetkisi olacak kullanıcıları, grupları ve rolleri yapılandırın. Daha fazla bilgi için bkz “Kullanıcıları ve rolleri yapılandırma” sayfa 522.
- Varsayılan olarak, LTPA simgesini içeren tanımlama bilgisinin adı `LtpaToken2`ile başlar ve mqweb sunucusu yeniden başlatıldığında değişebilecek bir sonek içerir. Bu rasgele tanımlama bilgisi adı, aynı sistemde birden çok mqweb sunucusunun çalışmasına izin verir. Ancak tanımlama bilgisi adının tutarlı bir değer olarak kalmasını istiyorsanız, **setmqweb** komutunu kullanarak tanımlama bilgisinin sahip olduğu adı belirtebilirsiniz. Daha fazla bilgi için LTPA simgesini yapılandırmabaşlıklı konuya bakın.
- Varsayılan olarak, LTPA belirteci tanımlama bilgisinin süresi 120 dakika sonra dolar. **setmqweb** komutunu kullanarak LTPA belirteci tanımlama bilgisinin süre bitim zamanını yapılandırabilirsiniz. Daha fazla bilgi için LTPA simgesini yapılandırmabaşlıklı konuya bakın.
- REST istekleri gönderirken güvenli bir bağlantı kullandığınızdan emin olun. `login` kaynağında HTTP POST yöntemini kullandığınızda, istekle gönderilen kullanıcı adı ve parola birleşimi şifrelenmez. Bu nedenle, REST APIile belirteç tabanlı kimlik doğrulamasını kullanırken güvenli bir bağlantı (HTTPS) kullanmanız gerekir. Varsayılan olarak, LTPA belirteci kimlik doğrulamasıyla HTTP kullanamazsınız. **secureLTPA** öğesini `False`olarak ayarlayarak, LTPA simgesini güvenli olmayan HTTP bağlantıları tarafından kullanılacak şekilde etkinleştirebilirsiniz. Daha fazla bilgi için LTPA simgesini yapılandırmabaşlıklı konuya bakın.
- `login` kaynağında HTTP GET yöntemini kullanarak, isteği doğrulamak için LTPA simgesini sağlayarak geçerli kullanıcının kimlik bilgilerini sorgulayabilirsiniz. Bu istek, kullanıcı adı ve kullanıcının atandığı rollerle ilgili bilgileri döndürür. Daha fazla bilgi için bkz. GET `/login`.

Yordam

1. Kullanıcı oturumu aç:
   a) `login` kaynağında HTTP POST yöntemini kullanın:

   ```
   https://host:port/ibmmq/rest/v1/login
   ```

   Kullanıcı adını ve parolayı JSON isteğinin gövdesine aşağıdaki biçimde ekleyin:

   ```
   {
       "username" : name,
       "password" : password
   }
   ```

   b) İsteğin döndürdüğü LTPA simgesini yerel tanımlama bilgisi deposunda saklayın. Varsayılan olarak, bu LTPA simgesi `LtpaToken2`önekine sahiptir.
2. REST isteklerini, her istekle birlikte tanımlama bilgisi olarak saklanan LTPA belirteciyle doğrulayın.

   HTTP PUT, PATCH ya da DELETE yöntemlerini kullanan istekler için bir `ibm-mq-rest-csrf-token` üstbilgisi ekleyin. Bu üstbilginin değeri boşluk da içinde olmak üzere herhangi bir şey olabilir.
3. Bir kullanıcının oturumunuzu kapatın:
   a) `login` kaynağında HTTP DELETE yöntemini kullanın:

   ```
   https://host:9443/ibmmq/rest/v1/login
   ```

   İsteği doğrulamak için tanımlama bilgisi olarak LTPA belirtecini sağlamanız ve bir `ibm-mq-rest-csrf-token` üstbilgisi eklemeniz gerekir. Bu üstbilginin değeri boşluk da içinde olmak üzere herhangi bir şey olabilir

   b) LTPA simgesini yerel tanımlama bilgisi deposundan silme yönergesini işleyin.

   Not: Yönerge işlenmezse ve LTPA simgesi yerel tanımlama bilgisi deposunda kalırsa, gelecekteki REST isteklerini doğrulamak için LTPA belirteci kullanılabilir. Yani, oturum sona erdirildikten sonra kullanıcı LTPA simgesiyle kimlik doğrulaması yapmaya çalıştığında, var olan simgeyi kullanan yeni bir oturum yaratılır.

Örnek

Aşağıdaki cURL örneği, Windows sistemlerinde simgeye dayalı kimlik doğrulamasıyla QM1kuyruk yöneticisinde Q1yeni bir kuyruğun nasıl yaratılacağını göstermektedir:

- Oturum açın ve LTPA simgesini `LtpaToken2`önekiyle yerel tanımlama bilgisi deposuna ekleyin. Kullanıcı adı ve parola bilgileri JSON gövdesine eklenir. `-c` işareti, simgenin saklanacağı dosyanın yerini belirtir:

  ```
  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
  ```

- Bir kuyruk oluşturun. HTTP POST yöntemini kuyruk kaynağıyla birlikte, LTPA simgesiyle kimlik doğrulamasını kullanarak kullanın. `LtpaToken2` önekine sahip LTPA simgesi, `-b` işareti kullanılarak `cookiejar.txt` dosyasından alınır. CSRF koruması, `ibm-mq-rest-csrf-token` HTTP üstbilgisinin varlığıyla sağlanır:

  ```
  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\"}"
  ```

- Oturumu kapatın ve yerel tanımlama bilgisi deposundan LTPA simgesini silin. LTPA simgesi, `-b` işareti kullanılarak `cookiejar.txt` dosyasından alınır. CSRF koruması, `ibm-mq-rest-csrf-token`

HTTP üstbilgisinin varlığıyla sağlanır. `cookiejar.txt` dosyasının konumu, LTPA simgesinin dosyadan silinmesi için `-c` işaretiyle belirtilir:

```
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
```

İlgili başvurular
POST /login
GET /login
SİL /login

IBM MQ Console ' nin IFrame 'e yerleştirilmesi

HTML < iframe> öğesi, bir Web sayfasını bir İç Çerçeve (IFrame) kullanılarak başka bir web sayfasına yerleştirmek için kullanılabilir. Güvenlik nedeniyle, IBM MQ Console varsayılan olarak bir IFrame içine yerleştirilemez. Ancak, mqweb sunucusunda **mqConsoleFrameAncestors** yapılandırma özelliğini kullanarak bir IFrame 'i etkinleştirebilirsiniz.

Bu görev hakkında

mqweb sunucusu, IFrame kullanarak IBM MQ Console ' i yerleştirebilen web sayfalarının kökenlerinin bir izin listesini sağlar. Başlangıç noktası, URL şeması, etki alanı ve kapı birleşimidir; örneğin, `https://example.com:1234`.

Listedeki girişleri belirtmek için mqweb sunucusundaki **mqConsoleFrameAncestors** yapılandırma özelliğini kullanabilirsiniz.

Varsayılan olarak **mqConsoleFrameAncestors** boştur; bu, IBM MQ Console öğesinin bir IFrame içine yerleştirilemeyecek anlamına gelir.

Yordam

Aşağıdaki komutu girerek IBM MQ Console ' i bir IFrame 'e yerleştirebilecek web sayfalarının kökenlerinin bir listesini belirtin:

```
setmqweb properties -k mqConsoleFrameAncestors -v allowedOrigins
```

Burada *allowedOrigins* , kökenlerin virgülle ayrılmış bir listesidir. Her kaynak aşağıdakilerden oluşmalıdır:

- Anasistem adı ya da IP adresi
- İsteğe bağlı bir URL şeması
- İsteğe bağlı bir kapı numarası

Anasistem adının (*) genel arama karakteriyle başlayabileceğini ve kapı numarasının (*) genel arama karakterini de kullanabildiğini unutmayın.
Örnek kökenler şunlardır:

```
https://example.com:1234
```

Bu, `https://example.com:1234` tarafından sunulan herhangi bir web sayfasının IBM MQ Console öğesini bir IFrame 'e yerleştirmesini sağlar.

```
https://*.example.com:*
```

Bu, anasistem adı `example.com`ile biten ve herhangi bir kapıyı kullanan HTTPS web sayfasının IBM MQ Console öğesini bir IFrame ürününe yerleştirmesini sağlar.

Örnek

Aşağıdaki örnek, IBM MQ Console ' in `https://site2.example.com:1234` ya da `https://site2.example.com:1235`tarafından sunulan web sayfalarından bir IFrame 'e yerleştirilmesini sağlar:

```
setmqweb properties -k mqConsoleFrameAncestors -v
https://site2.example.com:1234,https://site2.example.com:1235
```

REST API için CORS ' un yapılandırılması

Varsayılan olarak, bir web tarayıcısı, komut dosyası REST APIile aynı kökenden olmadığında JavaScriptgibi komut dosyalarının REST API ' ı çağırmasına izin vermez. Yani, kökler arası istekler etkinleştirilmez. Kökler Arası Kaynak Paylaşımını (CORS), belirtilen kökenlerden gelen başlangıç noktaları arası isteklere izin verecek şekilde yapılandırabilirsiniz.

Bu görev hakkında

REST API ' e bir web tarayıcısı aracılığıyla (örneğin, bir komut dosyası aracılığıyla) erişebilirsiniz. Bu istekler farklı bir kökenden REST API' e geldiğinden, web tarayıcısı isteği reddeder çünkü bu istek kökler arası bir istektir. Etki alanı, kapı ya da şema aynı değilse, kaynak farklı olur.

Örneğin, `http://localhost:1999/` adresinde barındırılan bir komut dosyeniz varsa, `https://localhost:9443/`adresinde bulunan bir web sitesinde HTTP GET komutunu yayınlarsanız, başlangıç noktaları arası bir istekte bulunurunuz. Kapı numaraları ve şema (HTTP) farklı olduğundan, bu istek başlangıç noktaları arası bir istektir.

CORS 'yi yapılandırarak ve REST API' a erişmesine izin verilen kaynakları belirterek kökler arası istekleri etkinleştirebilirsiniz.

CRS hakkında daha fazla bilgi için bkz. https://www.w3.org/TR/cors/ ve https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS.

Yordam

1. Aşağıdaki komutu girerek geçerli yapılandırmayı görüntüleyin:

   ```
   dspmqweb properties -a
   ```

   `mqRestCorsAllowedOrigins` girdisi izin verilen kökenleri belirtir. `mqRestCorsMaxAgeInSeconds` girdisi, web tarayıcısının herhangi bir CORS uçuş öncesi denetiminin sonuçlarını önbelleğe alabileceği süreyi saniye cinsinden belirtir.
2. Aşağıdaki komutu girerek REST API ' e erişmesine izin verilen kökenleri belirtin:

   ```
   setmqweb properties -k mqRestCorsAllowedOrigins -v allowedOrigins
   ```

   Burada *allowedOrigins* , başlangıç noktaları arası isteklere izin vermek istediğiniz kaynağı belirtir. Tüm başlangıç noktaları arası isteklere izin vermek için çift tırnak işareti ("*") içine alınmış bir yıldız işareti kullanabilirsiniz. Virgülle ayrılmış bir listede çift tırnak içine alınmış birden çok köken girebilirsiniz. Başlangıç noktaları arası isteklere izin vermemek için, *allowedOrigins*değeri olarak boş tırnak işaretleri girin.
3. Aşağıdaki komutu girerek, bir web tarayıcısının CORS ön uçuş denetimlerinin sonuçlarını önbelleğe almasına izin vermek istediğiniz süreyi saniye cinsinden belirtin:

   ```
   setmqweb properties -k mqRestCorsMaxAgeInSeconds -v time
   ```

Örnek

Aşağıdaki örnekte, `http://localhost:9883`, `https://localhost:1999`ve `https://localhost:9663`için etkinleştirilmiş başlangıç noktaları arası istekler gösterilmektedir. Herhangi bir CORS uçuş öncesi denetiminin önbelleğe alınan sonuç yaşı üst sınırı 90 saniye olarak ayarlanır:

```
setmqweb properties -k mqRestCorsAllowedOrigins -v "http://localhost:9883,https://
localhost:1999,https://localhost:9663"
setmqweb properties -k mqRestCorsMaxAgeInSeconds -v 90
```

IBM MQ Console ve REST API için anasistem üstbilgisi doğrulamasını yapılandırma

mqweb sunucusunu, yalnızca belirli bir izin listesiyle eşleşen bir anasistem üstbilgisiyle gönderilen istekler işlenecek şekilde IBM MQ Console ve REST API ' e erişimi kısıtlayacak şekilde yapılandırabilirsiniz. İzin listesinde olmayan bir anasistem üstbilgisi değeri kullanılırsa bir hata döndürülür.

Bu görev hakkında

mqweb sunucusu, kabul edilebilir anasistem üstbilgilerinin izin verilen listesini tanımlamak için sanal anasistemleri kullanır. Sanal anasistemlerle ilgili daha fazla bilgi için WebSphere Liberty belgelerine bakın: https://www.ibm.com/docs/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/cwlp_virtual_hosts.html

Bu görevi tamamlamak için, `mqwebuser.xml` dosyasını düzenlemek üzere yeterli ayrıcalığa sahip bir kullanıcı olmanız gerekir:

- z/OS z/OS' da `mqwebuser.xml` dosyasına yazma erişiminiz olmalıdır.
- Multi Diğer tüm işletim sistemlerinde ayrıcalıklı kullanıcıolmanız gerekir.
- V 9.3.5 Linux mqweb sunucusu bağımsız bir IBM MQ Web Server kuruluşunun parçasıysa, IBM MQ Web Server veri dizinindeki `mqwebuser.xml` dosyasına yazma erişiminiz olmalıdır.

Yordam

1. `mqwebuser.xml` dosyasını açın. Bu dosya aşağıdaki konumlardan birinde:
   - IBM MQ kuruluşunda:
     - Linux AIX AIX ya da Linuxsistemlerinde: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb`
     - Windows Windowsüzerinde: *`MQ_DATA_PATH`*`\web\installations\`*`installationName`*`\servers\mqweb`; burada *MQ_DATA_PATH* , IBM MQ veri yoludur. Bu yol, IBM MQkuruluşu sırasında seçilen veri yoludur. Varsayılan olarak bu yol `C:\ProgramData\IBM\MQ`olur.
     - z/OS z/OSişletim tarihinde: *`WLP_user_directory`*`/servers/mqweb`

       Burada *WLP_user_directory* , **crtmqweb** komutu mqweb sunucusu tanımlamasını oluşturmak için çalıştırıldığında belirtilen dizindir.
   - V 9.3.5 Linux Bağımsız bir IBM MQ Web Server kuruluşunda: *`MQ_OVERRIDE_DATA_PATH`*`/web/installations/MQWEBINST/servers/mqweb`

     Burada *MQ_OVERRIDE_DATA_PATH* , **MQ_OVERRIDE_DATA_PATH** ortam değişkeninin gösterdiği IBM MQ Web Server veri dizinidir.
2. `mqwebuser.xml` dosyasına şu kodu ekleyin ya da bu kodun açıklamasını kaldırın:

```
<virtualHost allowFromEndpointRef="defaultHttpEndpoint" id="default_host">
        <hostAlias>localhost:9080</hostAlias>
</virtualHost>
```

3. İzin vermek istediğiniz anasistem adı ve kapı birleşimini ekleyerek **<hostAlias>** alanını düzenleyin.

   Bu birleşim, mqweb sunucusunun yapılandırmasında kullandığınız anasistem adı ve kapı adı olabilir. Örneğin, `localhost:9443`varsayılan yapılandırmasını kullanıyorsanız, **<hostAlias>** alanında `localhost:9443` komutunu kullanmak isteyebilirsiniz.

   Gerekirse, daha fazla anasistem adı ve kapı birleşimine izin vermek için **<virtualHost>** etiketlerine birden çok **<hostAlias>** alanı ekleyebilirsiniz. Örneğin, HTTPS kapısını kullanan anasistem üstbilgilerinin yanı sıra HTTP kapısını kullanan anasistem üstbilgilerine izin vermek için.

Denetleme

IBM MQ Console ve REST API içinde gerçekleştirilen işlemlerin denetim kayıtları, kuyruk yöneticisi komutu ve yapılandırma olayları etkinleştirilerek üretilebilir ve AIX, Linux, and Windows üzerinde önemli durum değişiklikleri, mqweb sunucusunun günlük dosyalarına kaydedilir.

Önemli durum değişiklikleri

ALW

AIX, Linux, and Windows' da IBM MQ Console , önemli durum değişikliklerini mqweb sunucusunun günlüklerinde ileti olarak kaydeder. Her ileti, işlemi isteyen kimliği doğrulanmış birincil kullanıcı adını gösterir.

Kuyruk yöneticilerinin ne zaman yaratıldığı, başlatıldığı, sona erdirildiği ya da silindiği gibi önemli durum değişiklikleri, [AUDIT] günlük kaydı düzeyinde mqweb sunucusuna `messages.log` ve `console.log` dosyalarına kaydedilir. Her günlük girişi, işlemi isteyen kimliği doğrulanmış birincil kullanıcı adını gösterir.

`messages.log` ve `console.log` dosyalarını aşağıdaki konumda bulabilirsiniz:

- IBM MQ kuruluşunda:
  - Linux AIX AIX ya da Linuxsistemlerinde: `/var/mqm/web/installations/`*`installationName`*`/servers/mqweb/logs`
  - Windows Windowsüzerinde: *MQ_DATA_PATH*`\web\installations\`*`installationName`*`\servers\mqweb\logs`; burada *MQ_DATA_PATH* , IBM MQ veri yoludur. Bu yol, IBM MQkuruluşu sırasında seçilen veri yoludur. Varsayılan olarak bu yol `C:\ProgramData\IBM\MQ`olur.
- V 9.3.5 Linux Bağımsız bir IBM MQ Web Server kuruluşunda: *MQ_OVERRIDE_DATA_PATH*`/web/installations/MQWEBINST/servers/mqweb/logs`

  Burada *MQ_OVERRIDE_DATA_PATH* , **MQ_OVERRIDE_DATA_PATH** ortam değişkeninin gösterdiği IBM MQ Web Server veri dizinidir.

mqweb sunucusu günlük kaydı düzeylerini yapılandırma hakkında daha fazla bilgi için bkz. Günlük kaydını yapılandırma.

Komut ve yapılandırma olayları

İsteğe bağlı olarak, çoğu IBM MQ Console ve REST API etkinliği hakkında bilgi sağlamak için kuyruk yöneticisinde komut ve yapılandırma olaylarını etkinleştirebilirsiniz. Örneğin, kanalların oluşturulması ve kuyrukların sorgulanması komut ve yapılandırma olayları oluşturur. Komut ve yapılandırma olaylarını etkinleştirme hakkında daha fazla bilgi için bkz. Yapılandırma, komut ve kaydedici olaylarını denetleme.

Bu komut ve yapılandırma olayı iletileri için **MQIACF_EVENT_ORIGIN** alanı `MQEVO_REST` olarak ayarlanır ve **MQCACF_EVENT_APPL_IDENTITY** alanı, kimliği doğrulanmış birincil kullanıcı adının ilk 32 karakterini bildirir. Bir kullanıcı `MQWebAdmin` ya da `MQWebAdminRO` rolüne sahipse, **MQCACF_EVENT_USER_ID** alanı komutu yayınlayan birincil kullanıcının kullanıcı adını değil, mqweb sunucusu kullanıcı kimliğini bildirir. Ancak, kullanıcı `MQWebUser` rolüne sahipse, **MQCACF_EVENT_USER_ID** komutu yayınlayan birincil kullanıcının kullanıcı adını bildirir.

İlgili kavramlar

“Denetleme” sayfa 489

Olay iletilerini kullanarak güvenlik izinsiz girişlerini ya da izinsiz giriş girişlerini denetleyebilirsiniz. IBM MQ Explorerkomutunu kullanarak sisteminizin güvenliğini de denetleyebilirsiniz.

z/OS üzerinde IBM MQ Console ve REST API için güvenlikle ilgili önemli noktalar

IBM MQ Console ve REST API , bir kullanıcının komutları yayınlayabileceğini, görüntüleyebileceğini ya da değiştirebileceğini denetleyen güvenlik özelliklerine sahiptir. Daha sonra komutlar kuyruk yöneticisine geçirilir ve kuyruk yöneticisi güvenliği, kullanıcının komutu o kuyruk yöneticisine vermesine izin verilip verilmediğini denetlemek için kullanılır.

Yordam

1. mqweb sunucusu tarafından başlatılan görev kullanıcı kimliğinin belirli PCF komutlarını yayınlamak ve belirli kuyruklara erişmek için uygun yetkilere sahip olduğundan emin olun. Daha fazla bilgi için bkz “mqweb sunucusu tarafından başlatılan görev kullanıcı kimliğinin gerektirdiği yetki” sayfa 548.
2. `MQWebUser` rolü verilen kullanıcıların uygun yetkilere sahip olduğundan emin olun.

   `MQWebUser` rolüne atanan IBM MQ Console ve REST API kullanıcıları, birincil kullanıcının güvenlik bağlamı altında çalışır. Bu kullanıcı kimlikleri yalnızca kuyruk yöneticisinde gerçekleştirmek için kullanıcı kimliği verilen işlemleri gerçekleştirebilir ve mqweb sunucusu adres alanıyla aynı sistem kuyruklarına erişim yetkisi verilmesi gerekir.

   mqweb sunucusu tarafından başlatılan görev kullanıcı kimliğine, `MQWebUser` rolüne atanmış tüm kullanıcılar için alternatif kullanıcı erişimi verilmelidir.

   `MQWebUser` rolüne sahip kullanıcılara uygun yetkiler verme hakkında daha fazla bilgi için bkz. “IBM MQ Console ya da REST API kullanmak için gerekli IBM MQ kaynaklarına erişim” sayfa 549.
3. İsteğe bağlı: IBM MQ Console ve REST APIiçin TLS ' yi yapılandırın. Daha fazla bilgi için bkz “z/OS üzerinde REST API ve IBM MQ Console için TLS ' nin yapılandırılması” sayfa 550.

z/OS mqweb sunucusu tarafından başlatılan görev kullanıcı kimliğinin gerektirdiği yetki

z/OSüzerinde, mqweb sunucusu tarafından başlatılan görev kullanıcı kimliği, PCF komutlarını yayınlamak ve sistem kaynaklarına erişmek için belirli yetkilerin kullanılmasını gerektirir.

mqweb sunucusu tarafından başlatılan görev kullanıcı kimliği gerekiyor:

- z/OS UNIX System Serviceskullanabilecek bir z/OS UNIX kullanıcı kimliği (UID).
- IBM MQ kuruluşundaki `hlq.SCSQAUTH` ve `hlq.SCSQANL*` veri kümelerine erişim.
- z/OS UNIX System Servicesiçindeki IBM MQ kuruluş dosyalarına okuma erişimi.
- **`crtmqweb`** komut dosyası tarafından oluşturulan Liberty kullanıcı dizinine okuma ve yazma erişimi.
- Kuyruk yöneticisine bağlanma yetkisi. MQCONN sınıfındaki `hlq.BATCH` tanıtımı için, mqweb sunucusu tarafından başlatılan görev kullanıcı kimliğine *READ* erişimi verin.
- IBM MQ komutlarını verme ve belirli kuyruklara erişme yetkisi. Bu ayrıntılar “IBM MQ Console -gerekli komut güvenliği tanıtımları” sayfa 226, “Sistem kuyruğu güvenliği” sayfa 203ve “Bağlam güvenliğine ilişkin profiller” sayfa 213içinde açıklanmaktadır.
- MFTiçin REST API ' yi kullanmak üzere SYSTEM.FTE konusuna abone olma yetkisi. MXTOPIC sınıfındaki `hlq.SUBSCRIBE.SYSTEM.FTE` profiline mqweb sunucusu tarafından başlatılan görev kullanıcı kimliği *ALTER* erişimi verin.
- Bir SAF kaydı yapılandırıyorsanız, çeşitli güvenlik profillerine erişim. Ek bilgi için bkz. “IBM MQ Console ve REST API için SAF kaydının yapılandırılması” sayfa 530 .

Bağlantı kimlik doğrulaması

Kuyruk yöneticiniz, tüm toplu iş uygulamalarının geçerli bir kullanıcı kimliği ve parola sağlamasını gerektirecek şekilde yapılandırıldıysa, CHKLOCL (REQUIRED) ayarını tanımlayarak, MQCONN sınıfındaki `hlq.BATCH` tanıtımı için mqweb sunucusunun başlattığı görev kullanıcı kimliğine *UPDATE* erişim vermeniz gerekir.

Bu yetki, mqweb sunucusu tarafından başlatılan görev kullanıcı kimliği için bağlantı doğrulamanın CHKLOCL (İSTEğE BAğLı) kipinde çalışmasına neden olur.

Kuyruk yöneticisini, tüm toplu iş uygulamalarının geçerli bir kullanıcı kimliği ve parola sağlamasını gerektirecek şekilde yapılandırmadıysanız, MQCONN sınıfındaki `hlq.BATCH` tanıtımı için *READ* mqweb sunucusu görevini başlatan kullanıcı kimliğine erişim vermek yeterlidir.

CHCKLOCL hakkında daha fazla bilgi için bkz. "CHCKLOCL ' un yerel olarak bağlı uygulamalarda kullanılması" sayfa 193.

IBM MQ Console ya da REST API kullanmak için gerekli IBM MQ kaynaklarına erişim

IBM MQ Consoleya da REST APIiçinde `MQWebUser` rolündeki bir kullanıcı tarafından gerçekleştirilen işlemler, kullanıcının güvenlik bağlamı altında gerçekleşir.

Bu görev hakkında

IBM MQ Console ve REST APIiçindeki rollerle ilgili daha fazla bilgi için bkz. "IBM MQ Console ve REST API üzerindeki roller" sayfa 532 .

Bir kullanıcıya `MQWebUser` rolünde IBM MQ Console ya da REST APIkullanmak için gereken kuyruk yöneticisi kaynaklarına erişim vermek için aşağıdaki yordamı kullanın.

Yordam

1. `mqweb server started task` kullanıcı kimliğine, `MQWebUser` rolündeki her bir kullanıcı kimliği için alternatif kullanıcı erişimi verin.

   Bunu, kullanıcıların IBM MQ Console ya da REST APIaracılığıyla yöneteceği her kuyruk yöneticisinde yapın.

   `mqweb server started task` kullanıcı kimliğine `MQWebUser` rolündeki bir kullanıcıya alternatif kullanıcı erişimi vermek için aşağıdaki örnek RACF komutlarını kullanabilirsiniz:

   ```
   RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
   PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
   SETROPTS RACLIST(MQADMIN) REFRESH
   ```

   Burada:

   `hlq`
   Tanıtım öneki, kuyruk yöneticisi adı ya da kuyruk paylaşım grubu adı olabilir

   `userId`
   Kullanıcı `MQWebUser` rolünde mi

   `mqwebUserId`
   `mqweb server started task` kullanıcı kimliği

   Not: Büyük ve küçük harf karma güvenlik kullanıyorsanız, MQADMIN sınıfı yerine MXADMIN sınıfını kullanın.
2. `MQWebUser` rolündeki her bir kullanıcıya IBM MQ Console ve REST APIöğelerini kullanmak için gerekli olan sistem kuyruklarına erişim yetkisi verin.

   Bunu yapmak için, her iki SYSTEM.ADMIN.COMMAND.QUEUE ve SYSTEM.REST.REPLY.QUEUE, karışık büyük/küçük harf güvenliğinin kullanılıp kullanılmadığına bağlı olarak, her kullanıcıya MQQUEUE ya da MXQUEUE sınıfları için UPDATE erişimi verin.

Bunu, administrative REST API ağ geçidiaracılığıyla denetlenen uzak kuyruk yöneticileri de içinde olmak üzere, kullanıcının REST APIaracılığıyla yöneteceği her kuyruk yöneticisinde yapmanız gerekir.

3. `MQWebUser` rolündeki bir kullanıcının uzak kuyruk yöneticilerini yönetmesine izin vermek için kullanıcıya MQQUEUE ya da MXQUEUE sınıfındaki profil için UPDATE erişimi verin ve uzak kuyruk yöneticisine komut göndermek için kullanılan iletim kuyruğunu korur. Kullanıcıya ağ geçidi kuyruk yöneticisinde UPDATE erişimi vermeniz gerektiğini unutmayın.

   Uzak kuyruk yöneticisinde, komut yanıt iletilerini ağ geçidi kuyruk yöneticisine geri göndermek için kullanılan iletim kuyruğuna koymak üzere aynı kullanıcı için erişim verin.
4. `MQWebUser` rolündeki kullanıcılara, IBM MQ Console ve REST APItarafından desteklenen işlemleri gerçekleştirmek için gereken diğer kaynaklara erişim verin.

   Aşağıdakiler için gereken erişim:

   - REST APIiçinde işlemlerin gerçekleştirilmesi, tek tek REST API kaynaklarının *Güvenlik gereksinimleri bölümlerinde açıklanmaktadır.*
   - IBM MQ Console tarafından verilen komutlar için bkz. “IBM MQ Console -gerekli komut güvenliği tanıtımları” sayfa 226

z/OS z/OS üzerinde REST API ve IBM MQ Console için TLS ' nin yapılandırılması

z/OS' da, mqweb sunucusunu TLS ile güvenli bağlantılar ve istemci sertifikası kimlik doğrulaması için sertifikaları depolamak üzere bir RACF anahtar halkası kullanacak şekilde yapılandırabilirsiniz.

Başlamadan önce

Bu yordamı tamamlamak için `mqwebuser.xml` kütüğüne yazma erişimi olan bir kullanıcı ve SAF anahtarlarıyla çalışma yetkisi olmanız gerekir.

Bu görev hakkında

Varsayılan mqweb sunucusu yapılandırması, sunucu ve güvenilir sertifikalar için Java anahtar depolarını kullanır. z/OSüzerinde, mqweb sunucusunu Java anahtar depoları yerine bir RACF anahtar halkası kullanacak şekilde yapılandırabilirsiniz. Sunucu, kullanıcıların bir istemci sertifikası kullanarak kimlik doğrulamasını sağlayacak şekilde de yapılandırılabilir.

Libertyiçinde RACF anahtar halkalarını kullanma hakkında bilgi için bkz. Liberty: Anahtar depoları .

mqweb sunucusunu bir RACF anahtarlığı kullanacak şekilde yapılandırmak ve isteğe bağlı olarak istemci sertifikası kimlik doğrulamasını yapılandırmak için bu yordamı izleyin. Bu yordam, kendi sertifika yetkilisi (CA) sertifikalarınızla imzalanmış sertifikaları yaratmak ve kullanmak için gerekli adımları açıklar. Üretim için, bir dış sertifika yetkilisinden alınan sertifikaları kullanmayı tercih edebilirsiniz.

Yordam

1. Sunucu sertifikasını imzalamak için kullanılacak bir sertifika yetkilisi (CA) sertifikası yaratın. Örneğin, şu RACF komutunu girin:

```
RACDCERT GENCERT -
  CERTAUTH -
  SUBJECTSDN(CN('mqweb Certification Authority') -
    O('IBM') -
    OU('MQ')) -
  SIZE(2048) -
  WITHLABEL('mqwebCertauth')
```

2. Aşağıdaki komutu girerek, adım 1 'de oluşturulan CA sertifikasıyla imzalanmış bir sunucu sertifikası oluşturun:

```
RACDCERT ID(mqwebUserId) GENCERT -
  SUBJECTSDN(CN('hostname') -
   O('IBM') -
```

```
   OU('MQ')) -
   SIZE(2048) -
   SIGNWITH (CERTAUTH LABEL('mqwebCertauth')) -
  WITHLABEL('mqwebServerCert')
```

Burada *mqwebUserId* , mqweb sunucusu tarafından başlatılan görev kullanıcı kimliği ve *hostname* , mqweb sunucusunun anasistem adıdır.

3. Aşağıdaki komutları girerek CA sertifikasını ve sunucu sertifikasını bir SAF anahtar halkasına bağlayın:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebCertauth') CERTAUTH)
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebServerCert'))
```

Burada *mqwebUserId* , mqweb sunucusu tarafından başlatılan görev kullanıcı kimliği ve *anahtarlık* , kullanmak istediğiniz anahtarlık adıdır.

4. Aşağıdaki komutu girerek CA sertifikasını bir CER dosyasına aktarın:

```
RACDCERT CERTAUTH EXPORT(LABEL('mqwebCertauth')) -
        DSN('hlq.CERT.MQWEBCA') -
        FORMAT(CERTDER) -
        PASSWORD('password')
```

5. Dışa aktarılan CA sertifikasını ikili olarak iş istasyonunuza FTP ile aktarın ve sertifika yetkilisi sertifikası olarak tarayıcınıza aktarın.
6. İsteğe bağlı: İstemci sertifikası kimlik doğrulamasını yapılandırmak istiyorsanız, bir istemci sertifikası yaratın ve dışa aktarın.
   a) İstemci sertifikasını imzalamak için kullanılacak bir sertifika yetkilisi (CA) sertifikası yaratın. Örneğin, şu RACF komutunu girin:

```
RACDCERT GENCERT -
  CERTAUTH -
  SUBJECTSDN(CN('mqweb User CA') -
   O('IBM') -
   OU('MQ')) -
   SIZE(2048) -
  WITHLABEL('mqwebUserCertauth')
```

   b) Aşağıdaki komutu girerek CA sertifikasını bir SAF anahtar halkasına bağlayın:

```
RACDCERT ID(mqwebUserId) CONNECT(RING(keyring) LABEL('mqwebUserCertauth') CERTAUTH)
```

   Burada *mqwebUserId* , mqweb sunucusu tarafından başlatılan görev kullanıcı kimliği ve *anahtarlık* , kullanmak istediğiniz anahtarlık adıdır.

   c) CA sertifikasıyla imzalanmış bir istemci sertifikası yaratın. Örneğin, şu komutu girin.

```
RACDCERT ID(clientUserId) GENCERT -
  SUBJECTSDN(CN('clientUserId') -
     O('IBM') -
     OU('MQ')) -
   SIZE(2048) -
   SIGNWITH (CERTAUTH LABEL('mqwebUserCertauth')) -
  WITHLABEL('userCertLabel')
```

   Burada *clientUserId* , kullanıcı adıdır.

   Bir sertifikayı bir birincil kullanıcıyla eşlemek için kullanılan yöntem, yapılandırılan kullanıcı kayıt dosyası tipine bağlıdır:

   - Temel bir kayıt dosyası kullanıyorsanız, sertifikadaki Ortak Ad alanı, kayıttaki kullanıcıyla eşleştirilir.
   - Bir SAF kaydı kullanıyorsanız ve sertifika RACF veritabanıysa, sertifikayı oluştururken **ID** parametresiyle belirtilen sertifika sahibi kullanılır.
   - Bir LDAP kaydı kullanıyorsanız, sertifikadaki tam ayırt edici ad LDAP kaydıyla eşleştirilir.

   d) Aşağıdaki komutu girerek istemci sertifikasını bir PKCS #12 dosyasına aktarın:

```
RACDCERT ID(mqwebUserId) EXPORT(LABEL('userCertLabel')) -
         PASSWORD('password') DSN('hlq.USER.CERT')
```

e) Dışa aktarılan sertifikayı ikili olarak iş istasyonunuza FTP ile göndermenizi sağlar. İstemci sertifikasını IBM MQ Consoleile kullanmak için, bunu IBM MQ Console ' a kişisel sertifika olarak erişmek için kullanılan web tarayıcısına aktarın.

7. *WLP_user_directory*/servers/mqweb/mqwebuser.xmldosyasını düzenleyin; burada *WLP_user_directory* , **crtmqweb** komut dosyası mqweb sunucusu tanımlamasını oluşturmak için çalıştırıldığında belirtilen dizindir.

   mqweb sunucusunu bir RACF anahtarlığı kullanacak şekilde yapılandırmak için aşağıdaki değişiklikleri yapın:

   a) Aşağıdaki satırı kaldırın ya da açıklama satırı yapın:

   ```
   <sslDefault sslRef="mqDefaultSSLConfig"/>
   ```

   b) Aşağıdaki deyimleri ekleyin:

   ```
   <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"/>
   ```

   Burada:

   - *mqwebUserId* , mqweb sunucusu tarafından başlatılan görev kullanıcı kimliğidir.
   - *anahtarlık* , RACF anahtarlık adıdır.
   - *mqwebServerCert* , mqweb sunucusu sertifikasının etiketidir.

   Notlar: keyStore password değeri yoksayıldı.

8. mqweb sunucusunun başlattığı görevi durdurup yeniden başlatarak mqweb sunucusunu yeniden başlatın.
9. İsteğe bağlı: Kimlik doğrulamak için istemci sertifikasını kullanın:

   - İstemci sertifikasını IBM MQ Consoleile kullanmak için istemci sertifikasını kurduğunuz web tarayıcısında IBM MQ Console için URL girin.
   - İstemci sertifikasını REST API ile kullanmak için, istemci sertifikasını her REST isteğiyle birlikte sağlayın.

   Notlar:

   a. IBM MQ Consolekimlik doğrulaması için yalnızca sertifikaları kullanıyorsanız, tarayıcı içinden seçim yapabileceğiniz sertifikaların bir listesini görüntüleyebilir.

   b. Farklı bir sertifika kullanmak istiyorsanız, tarayıcınızı kapatıp yeniden başlatmanız gerekebilir.

   c. RACF veritabanında olmayan istemci sertifikalarını kullanıyorsanız, sertifika özniteliklerini bir kullanıcı kimliğiyle eşlemek için RACF sertifika adı süzgecini kullanabilirsiniz. Örneğin:

   ```
   RACDCERT ID(DEPT3USR) MAP SDNFILTER(OU=DEPT1.C=US)
   ```

   OU=DEPT1 ve C=US içeren bir konu ayırt edici adına sahip sertifikaları DEPT3USRkullanıcı kimliğiyle eşler.

Sonuçlar

IBM MQ Console ve REST APIiçin bir TLS arabirimi ayarladınız.

ALW AIX, Linux, and Windows üzerinde anahtarları ve sertifikaları yönetme

AIX, Linux, and Windowssistemlerinde, anahtarları, sertifikaları ve sertifika isteklerini yönetmek için **runmqckm** ve **runmqakm** komutlarını kullanın.

Bu görev hakkında

runmqckm komutu, **iKeyman**, komutuna benzer işlevler sağlar ve **runmqakm** komutu, **gskitcapicmd**komutuna benzer işlevler sağlar. **runmqckm** ya da **runmqakm**komutunu kullanmadan önce, **setmqenv** komutunu çalıştırarak sistem ortam değişkenlerinin doğru yapılandırıldığından emin olun.

runmqckm komutu, IBM MQ JRE bileşeninin kurulmasını gerektirir. Bu bileşen kurulu değilse, bunun yerine **runmqakm** komutunu kullanabilirsiniz.

TLS sertifikalarını FIPS uyumlu bir şekilde yönetmeniz gerekiyorsa, **runmqckm** komutu yerine **runmqakm** komutunu kullanın. Bunun nedeni, **runmqakm** komutunun daha güçlü şifrelemeyi desteklemesi olabilir.

Yordam

- Aşağıdakileri gerçekleştirmek için **runmqckm** ve **runmqakm** komutlarını kullanın:
  - V 9.3.0 IBM MQ ' in gerektirdiği CMS ya da PKCS#12 anahtar veritabanı dosyalarının tipini yaratın.
  - Sertifika istekleri yarat
  - Kişisel sertifikaları içe aktar
  - CA sertifikalarını içe aktar
  - Kendinden imzalı sertifikaları yönet

İlgili görevler

“strmqikm kullanıcı arabiriminin kullanılması” sayfa 304
strmqikm (iKeyman) kullanarak kişisel sertifika yaratabilirsiniz. Grafik kullanıcı arabirimi.

İlgili başvurular

IBM **strmqikm** (iKeyman) GUI 'sinin çağrılması

İlgili bilgiler

Anahtar Aracı

ALW AIX, Linux, and Windows üzerinde runmqckm ve runmqakm komutları

Bu bölümde, komutun nesnesine göre **runmqckm** ve **runmqakm** komutları açıklanmaktadır.

İki komut arasındaki temel farklılıklar şunlardır:

- **runmqckm**
  - **iKeycmd** ile benzer işlevler sağlar
  - JKS ve JCEKS anahtar havuzu dosya biçimlerini destekler
- **runmqakm**
  - **gskitcapicmd** ile benzer işlevler sağlar
  - **runmqckm** komutu, Elliptic Curve ortak anahtarlarıyla sertifika ve sertifika isteklerinin oluşturulmasını desteklerken,
  - Anahtar havuzu dosyasının **-strong** parametresiyle **runmqckm** komutundan daha güçlü şifrelenmesini destekler

– FIPS 140-2 uyumlu olarak onaylanmıştır ve **-fips** parametresi kullanılarak FIPS uyumlu bir şekilde çalışacak şekilde yapılandırılabilir

Uyarı: runmqckm komutu, IBM MQ Java runtime environment (JRE) özelliğinin kurulmasını gerektirir.

Her komut en az bir *nesne*belirtir. PKCS #11 aygıt işlemlerine ilişkin komutlar ek nesneler belirtebilir. Anahtar veritabanı, sertifika ve sertifika isteği nesnelerine ilişkin komutlar bir *işlem*de belirtir. Nesne aşağıdakilerden biri olabilir:

-keydb
İşlemler bir anahtar veritabanı için geçerlidir

-cert
İşlemler bir sertifikaya uygulanır

-certreq.
İşlemler bir sertifika isteği için geçerlidir

-help
yardımı görüntüler

-version
Sürüm bilgilerini görüntüler

Aşağıdaki alt konularda, anahtar veritabanı, sertifika ve sertifika isteği nesneleri üzerinde gerçekleştirebileceğiniz işlemler açıklanır; bu komutlara ilişkin seçeneklerin açıklaması için bkz. "AIX, Linux, and Windows üzerinde runmqckm ve runmqakm seçenekleri" sayfa 565 .

ALW AIX, Linux, and Windows üzerinde CMS ya da PKCS#12 anahtar veritabanlarına ilişkin komutlar

Bir CMS anahtar veritabanına ya da PKCS#12 anahtar veritabanına ilişkin anahtarları ve sertifikaları yönetmek için **runmqckm** ve **runmqakm** komutlarını kullanın.

Not: IBM MQ , SHA-3 ya da SHA-5 algoritmalarını desteklemez. Her iki algoritma da SHA-2 ailesinin üyesi olduğundan, SHA384WithRSA ve SHA512WithRSA dijital imza algoritması adlarını kullanabilirsiniz.

Deprecated Sayısal imza algoritması adları SHA3WithRSA ve SHA5WithRSA , sırasıyla SHA384WithRSA ve SHA512WithRSA kısaltması oldukları için kullanımdan kaldırılmıştır.

-keydb -changepw
Anahtar veritabanına ilişkin parolayı değiştirin:

runmqckm komutunu kullanarak:

```
-keydb -changepw -db filename -pw password -new_pw new_password -expire days
```

runmqakm komutunu kullanarak:

```
-keydb -changepw -db filename -pw password -new_pw new_password -expire days
-fips -strong
```

-keydb -convert
runmqckm komutu için anahtar veritabanını bir biçimden diğerine dönüştürün:

```
-keydb -convert -db filename -pw password
-old_format cms | pkcs12 -new_format cms
```

runmqakm komutunu kullanarak, eski bir CMS anahtar veritabanını yeni CMS anahtar veritabanına dönüştürün:

```
-keydb -convert -db filename -pw password
-new_db filename -new_pw password -strong -fips
```

-keydb -create
Anahtar veritabanı yarat:

runmqckm komutunu kullanarak:

```
V 9.3.0 -keydb -create -db filename -pw password -type cms
| pkcs12
```

runmqakm komutunu kullanarak:

```
V 9.3.0 -keydb -create -db filename -pw password -type  cms
/ p12 -fips -strong
```

-keydb -delete
Anahtar veritabanını sil:

Şu komutlardan birini kullanarak:

```
-keydb -delete -db filename -pw password
```

-keydb -list
Şu anda desteklenen anahtar veritabanı tiplerini listele:

runmqckm komutunu kullanarak:

```
-keydb -list
```

runmqakm komutunu kullanarak:

```
-keydb -list -fips
```

-cert -add
Anahtar veritabanına bir dosyadan sertifika ekleyin:

runmqckm komutunu kullanarak:

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

runmqakm komutunu kullanarak:

```
-cert -add -db filename -pw password -label label -file filename
-format ascii | binary -fips
```

-cert -create
Kendinden onaylı sertifika yarat:

runmqckm komutunu kullanarak:

```
-cert -create -db filename -pw password -label label
-dn distinguished_name -size 1024 | 512 -x509version 3 | 1 | 2
-expire days -sig_alg MD2_WITH_RSA | MD2WithRSA |
MD5_WITH_RSA | MD5WithRSA |
SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

runmqakm komutunu kullanarak:

```
-cert -create -db filename -pw password -label label
-dn distinguished_name -size 2048 | 1024 | 512 -x509version 3 | 1 | 2
-expire days -fips -sig_alg md5 |
MD5_WITH_RSA | SHA_WITH_DSA |
```

```
SHA_WITH_RSA | sha1 |
SHA1WithDSA | SHA1WithECDSA |
SHA1WithRSA | sha224 |
SHA224_WITH_RSA | SHA224WithDSA |
SHA224WithECDSA | SHA224WithRSA |
sha256 | SHA256_WITH_RSA |
SHA256WithDSA | SHA256WithECDSA |
SHA256WithRSA | SHA2WithRSA |
sha384 | SHA384_WITH_RSA |
SHA384WithECDSA | SHA384WithRSA |
sha512 | SHA512_WITH_RSA |
SHA512WithECDSA | SHA512WithRSA |
SHAWithDSA | SHAWithRSA |
EC_ecdsa_with_SHA1 | EC_ecdsa_with_SHA224 |
EC_ecdsa_with_SHA256 | EC_ecdsa_with_SHA384 |
EC_ecdsa_with_SHA512
```

-cert -delete
Bir sertifikayı sil:

runmqckm komutunu kullanarak:

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

runmqakm komutunu kullanarak:

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

-cert -details
Belirli bir sertifikaya ilişkin ayrıntılı bilgileri listeleyin:

runmqckm komutunu kullanarak:

```
-cert -details -db filename -pw password -label label
```

runmqakm komutunu kullanarak:

```
-cert -details -db filename -pw password -label label -fips
```

-cert -export
Kişisel sertifikayı ve ilişkili özel anahtarını bir anahtar veritabanından PKCS#12 dosyasına ya da başka bir anahtar veritabanına aktarın:

runmqckm komutunu kullanarak:

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

runmqakm komutunu kullanarak:

```
-cert -export -db filename -pw password -label label -type cms | pkcs12
-target filename -target_pw password -target_type  cms | pkcs12
-encryption strong | weak -fips
```

-cert -extract
Bir sertifikayı anahtar veritabanından çek:

runmqckm komutunu kullanarak:

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

runmqakm komutunu kullanarak:

```
-cert -extract -db filename -pw password -label label -target filename
-format ascii | binary -fips
```

-cert -import
Anahtar veritabanından kişisel sertifika al:

runmqckm komutunu kullanarak:

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

runmqakm komutunu kullanarak:

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

Her iki komut için:

- `-label` seçeneği gereklidir ve kaynak anahtar veritabanından içe aktarılacak sertifikanın etiketini belirtir.
- Ayrıca, `-new_label` seçeneğini de kullanabilirsiniz. Bu, içe aktarılan sertifikaya hedef anahtar veritabanında kaynak veritabanındaki etiketten farklı bir etiket verilmesini sağlar.

-cert -list

Bir anahtar veritabanındaki tüm sertifikaları listele:

runmqckm komutunu kullanarak:

```
-cert -list all | personal | CA -db filename -pw password
```

runmqakm komutunu kullanarak:

```
-cert -list all | personal | CA -db filename -pw password -fips
```

-cert -receive

Dosyadan sertifika al:

runmqckm komutunu kullanarak:

```
-cert -receive -file filename -db filename -pw password
-format ascii | binary -default_cert yes | no
```

runmqakm komutunu kullanarak:

```
-cert -receive -file filename -db filename -pw password
-format ascii | binary -default_cert yes | no -fips
```

-cert -sign

Bir sertifikayı imzala:

runmqckm komutunu kullanarak:

```
-cert -sign -db filename -file filename -pw password
-label label -target filename -format ascii | binary -expire days
-sig_alg MD2_WITH_RSA | MD2WithRSA | MD5_WITH_RSA |
MD5WithRSA | SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

runmqakm komutunu kullanarak:

```
-cert -sign -db filename -file filename -pw password
-label label -target filename -format ascii | binary -expire days -fips
-sig_alg md5 | MD5_WITH_RSA | SHA_WITH_DSA |
SHA_WITH_RSA | sha1 | SHA1WithDSA |
SHA1WithECDSA | SHA1WithRSA | sha224 |
SHA224_WITH_RSA | SHA224WithDSA |
SHA224WithECDSA | SHA224WithRSA | sha256 |
SHA256_WITH_RSA | SHA256WithDSA |
SHA256WithECDSA | SHA256WithRSA |
SHA2WithRSA | sha384 | SHA384_WITH_RSA |
```

```
SHA384WithECDSA | SHA384WithRSA |
sha512 | SHA512_WITH_RSA |
SHA512WithECDSA | SHA512WithRSA |
SHAWithDSA | SHAWithRSA |
EC_ecdsa_with_SHA1 | EC_ecdsa_with_SHA224 |
EC_ecdsa_with_SHA256 | EC_ecdsa_with_SHA384 |
EC_ecdsa_with_SHA512
```

-certreq -create
Sertifika isteği yarat:

runmqckm komutunu kullanarak:

```
-certreq -create -db filename -pw password -label label -dn distinguished_name
-size 1024 | 512 -file filename
-sig_alg MD2_WITH_RSA | MD2WithRSA |
MD5_WITH_RSA | MD5WithRSA |
SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

runmqakm komutunu kullanarak:

```
-certreq -create -db filename -pw password -label label -dn distinguished_name
-size 2048 | 1024 | 512 -file filename -fips
-sig_alg md5 | MD5_WITH_RSA | SHA_WITH_DSA |
SHA_WITH_RSA | sha1 | SHA1WithDSA |
SHA1WithECDSA | SHA1WithRSA | sha224 |
SHA224_WITH_RSA | SHA224WithDSA |
SHA224WithECDSA | SHA224WithRSA | sha256 |
SHA256_WITH_RSA | SHA256WithDSA |
SHA256WithECDSA | SHA256WithRSA |
SHA2WithRSA | sha384 | SHA384_WITH_RSA |
SHA384WithECDSA | SHA384WithRSA |
sha512 | SHA512_WITH_RSA |
SHA512WithECDSA | SHA512WithRSA |
SHAWithDSA | SHAWithRSA |
EC_ecdsa_with_SHA1 | EC_ecdsa_with_SHA224 |
EC_ecdsa_with_SHA256 | EC_ecdsa_with_SHA384 |
EC_ecdsa_with_SHA512
```

-certreq -delete
Sertifika silme isteği:

runmqckm komutunu kullanarak:

```
-certreq -delete -db filename -pw password -label  label
```

runmqakm komutunu kullanarak:

```
-certreq -delete -db filename -pw password -label  label -fips
```

-certreq -details
Belirli bir sertifika isteğine ilişkin ayrıntılı bilgileri listeleyin:

runmqckm komutunu kullanarak:

```
-certreq -details -db filename -pw password -label  label
```

runmqakm komutunu kullanarak:

```
-certreq -details -db filename -pw password -label  label -fips
```

Bir sertifika isteğiyle ilgili ayrıntılı bilgileri listeleyin ve tam sertifika isteğini gösterin:

runmqckm komutunu kullanarak:

```
-certreq -details -showOID -db filename -pw password -label label
```

runmqakm komutunu kullanarak:

```
-certreq -details -showOID -db filename -pw password -label label -fips
```

-certreq -extract
Sertifika isteği veritabanından bir sertifika isteğini bir dosyaya çek:

runmqckm komutu için:

```
-certreq -extract -db filename -pw password -label label -target filename
```

runmqakm komutunu kullanarak:

```
-certreq -extract -db filename -pw password -label label -target filename -fips
```

-certreq -list
Sertifika isteği veritabanındaki tüm sertifika isteklerini listele:

runmqckm komutunu kullanarak:

```
-certreq -list -db filename -pw password
```

runmqakm komutunu kullanarak:

```
-certreq -list -db filename -pw password -fips
```

-certreq -rekreate
Sertifika isteğini yeniden yarat:

runmqckm komutunu kullanarak:

```
-certreq -recreate -db filename -pw password -label label -target filename
```

runmqakm komutunu kullanarak:

```
-certreq -recreate -db filename -pw password -label label -target filename -fips
```

ALW AIX, Linux, and Windows üzerinde şifreleme aygıtı işlemlerine ilişkin komutlar

Şifreleme aygıtı işlemlerine ilişkin anahtarları ve sertifikaları yönetmek için **runmqckm** (iKeycmd) ve **runmqakm** komutlarını kullanabilirsiniz.

Not: IBM MQ , SHA-3 ya da SHA-5 algoritmalarını desteklemez. Her iki algoritma da SHA-2 ailesinin üyesi olduğundan, SHA384WithRSA ve SHA512WithRSA dijital imza algoritması adlarını kullanabilirsiniz.

Deprecated Sayısal imza algoritması adları SHA3WithRSA ve SHA5WithRSA , sırasıyla SHA384WithRSA ve SHA512WithRSA kısaltması oldukları için kullanımdan kaldırılmıştır.

-keydb -changepw
Şifreleme aygıtına ilişkin parolayı değiştirin:

runmqckm komutunu kullanarak:

```
-keydb -changepw -crypto module_name -tokenlabel token_label
-pw password -new_pw new_password
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller

64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-keydb -changepw -db filename -crypto module_name -tokenlabel  token_label
-pw password -new_pw new_password -fips -strong
```

-keydb -list

Şu anda desteklenen anahtar veritabanı tiplerini listele:

runmqckm komutunu kullanarak:

```
-keydb -list
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-keydb -list -fips
```

-cert -add

Bir dosyadan şifreleme aygıtına sertifika ekleyin:

runmqckm komutunu kullanarak:

```
-cert -add -crypto module_name -tokenlabel token_label -pw password
-label label -file filename -format ascii | binary
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-cert -add -crypto module_name -tokenlabel token_label -pw password
-label label -file filename -format ascii | binary -fips
```

-cert -create

Şifreleme aygıtında kendinden onaylı bir sertifika yaratın:

runmqckm komutunu kullanarak:

```
-cert -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 1024 | 512 -x509version 3 | 1 | 2
-default_cert no | yes -expire days
-sig_alg MD2_WITH_RSA | MD2WithRSA |
MD5_WITH_RSA | MD5WithRSA |
SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11

kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-cert -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 2048 | 1024 | 512 -x509version 3 | 1 | 2
-default_cert no | yes -expire days
-fips -sig_alg md5 | MD5_WITH_RSA | SHA_WITH_DSA |
SHA_WITH_RSA | sha1 | SHA1WithDSA |
SHA1WithECDSA | SHA1WithRSA |
sha224 | SHA224_WITH_RSA |
SHA224WithDSA | SHA224WithECDSA |
SHA224WithRSA | sha256 |
SHA256_WITH_RSA | SHA256WithDSA |
SHA256WithECDSA | SHA256WithRSA |
SHA2WithRSA | sha384 | SHA384_WITH_RSA |
SHA384WithECDSA | SHA384WithRSA |
sha512 | SHA512_WITH_RSA |
SHA512WithECDSA | SHA512WithRSA |
SHAWithDSA | SHAWithRSA |
EC_ecdsa_with_SHA1 | EC_ecdsa_with_SHA224 |
EC_ecdsa_with_SHA256 | EC_ecdsa_with_SHA384 |
EC_ecdsa_with_SHA512
```

-cert -delete

Şifreleme aygıtındaki bir sertifikayı sil:

runmqckm komutunu kullanarak:

```
-cert -delete -crypto module_name -tokenlabel token_label -pw password -label label
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-cert -delete -crypto module_name -tokenlabel token_label -pw password -label label -fips
```

-cert -details

Şifreleme aygıtındaki belirli bir sertifikaya ilişkin ayrıntılı bilgileri listeleyin:

runmqckm komutunu kullanarak:

```
-cert -details -crypto module_name -tokenlabel token_label
-pw password -label label
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-cert -details -crypto module_name -tokenlabel token_label
-pw password -label label -fips
```

Ayrıntılı bilgileri listeleyin ve bir şifreleme aygıtında belirli bir sertifikaya ilişkin tam sertifikayı gösterin:

runmqckm komutunu kullanarak:

```
-cert -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-cert -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label -fips
```

-cert -extract

Bir sertifikayı anahtar veritabanından çek:

runmqckm komutunu kullanarak:

```
-cert -extract -crypto module_name -tokenlabel token_label -pw password
-label label -target filename -format ascii | binary
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-cert -extract -crypto module_name -tokenlabel token_label -pw password
-label label -target filename -format ascii | binary -fips
```

-cert -import

İkincil anahtar veritabanı desteği olan bir şifreleme aygıtına sertifika alın:

runmqckm komutunu kullanarak:

```
-cert -import -db filename -pw password -label label -type cms
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-cert -import -db filename -pw password -label label -type cms
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password -fips
```

Bir PKCS #12 sertifikasını ikincil anahtar veritabanı desteği olan bir şifreleme aygıtına aktar:

runmqckm komutunu kullanarak:

```
-cert -import -file filename -pw password -type pkcs12
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-cert -import -file filename -pw password -type pkcs12
-crypto module_name -tokenlabel token_label -pw  password
-secondaryDB filename -secondaryDBpw password -fips
```

-cert -list
Bir şifreleme aygıtındaki tüm sertifikaları listele:

runmqckm komutunu kullanarak:

```
-cert -list all | personal | CA -crypto module_name
-tokenlabel token_label -pw  password
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-cert -list all | personal | CA -crypto module_name
-tokenlabel token_label -pw  password -fips
```

-cert -receive
İkincil anahtar veritabanı desteği ile bir dosyadan şifreleme aygıtına sertifika alın:

runmqckm komutunu kullanarak:

```
-cert -receive -file filename -crypto module_name -tokenlabel  token_label
-pw password -default_cert yes | no -secondaryDB filename
-secondaryDBpw password -format  ascii | binary
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-cert -receive -file filename -crypto module_name -tokenlabel  token_label
-pw password -default_cert yes | no -secondaryDB filename
-secondaryDBpw password -format  ascii | binary -fips
```

-certreq -create
Şifreleme aygıtında sertifika isteği yarat:

runmqckm komutunu kullanarak:

```
-certreq -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 1024 | 512 -file filename
-sig_alg MD2_WITH_RSA | MD2WithRSA | MD5_WITH_RSA |
MD5WithRSA | SHA1WithDSA | SHA1WithRSA |
SHA256_WITH_RSA | SHA256WithRSA |
SHA2WithRSA | SHA384_WITH_RSA |
SHA384WithRSA | SHA512_WITH_RSA |
SHA512WithRSA | SHA_WITH_DSA |
SHA_WITH_RSA | SHAWithDSA |
SHAWithRSA
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11

kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-certreq -create -crypto module_name -tokenlabel token_label
-pw password -label label -dn distinguished_name
-size 2048 | 1024 | 512 -file filename -fips
-sig_alg md5 | MD5_WITH_RSA | SHA_WITH_DSA |
SHA_WITH_RA | sha1 | SHA1WithDSA |
SHA1WithECDSA | SHA1WithRSA |
sha224 | SHA224_WITH_RSA | SHA224WithDSA |
SHA224WithECDSA | SHA224WithRSA |
sha256 | SHA256_WITH_RSA | SHA256WithDSA |
SHA256WithECDSA | SHA256WithRSA |
SHA2WithRSA | sha384 | SHA384_WITH_RSA |
SHA384WithECDSA | SHA384WithRSA |
sha512 | SHA512_WITH_RSA |
SHA512WithECDSA | SHA512WithRSA |
SHAWithDSA | SHAWithRSA |
EC_ecdsa_with_SHA1 | EC_ecdsa_with_SHA224 |
EC_ecdsa_with_SHA256 | EC_ecdsa_with_SHA384 |
EC_ecdsa_with_SHA512
```

-certreq -delete

Şifreleme aygıtından sertifika silme isteği:

runmqckm komutunu kullanarak:

```
-certreq -delete -crypto module_name -tokenlabel token_label
-pw password -label label
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-certreq -delete -crypto module_name -tokenlabel token_label
-pw password -label label -fips
```

-certreq -details

Şifreleme aygıtındaki belirli bir sertifika isteğine ilişkin ayrıntılı bilgileri listeleyin:

runmqckm komutunu kullanarak:

```
-certreq -details -crypto module_name -tokenlabel token_label
-pw password -label label
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-certreq -details -crypto module_name -tokenlabel token_label
-pw password -label label -fips
```

Bir sertifika isteğiyle ilgili ayrıntılı bilgileri listeleyin ve şifreleme aygıtında tam sertifika isteğini gösterin:

runmqckm komutunu kullanarak:

```
-certreq -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-certreq -details -showOID -crypto module_name -tokenlabel  token_label
-pw password -label label -fips
```

-certreq -extract
Şifreleme aygıtındaki bir sertifika isteği veritabanından bir dosyaya sertifika isteği çıkarın:

runmqckm komutunu kullanarak:

```
-certreq -extract -crypto module_name -tokenlabel token_label
-pw password -label label -target filename
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-certreq -extract -crypto module_name -tokenlabel token_label
-pw password -label label -target filename -fips
```

-certreq -list
Şifreleme aygıtındaki sertifika isteği veritabanındaki tüm sertifika isteklerini listele:

runmqckm komutunu kullanarak:

```
-certreq -list -crypto module_name -tokenlabel token_label
-pw password
```

PKCS#11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' in 64 bitlik programlar olduğunu unutmayın. PKCS#11 desteği için gerekli dış modüller 64 bitlik bir işleme yüklenecek, bu nedenle şifreleme donanımının yönetimi için 64 bitlik bir PKCS#11 kitaplığınızın kurulu olması gerekir. **strmqikm** ve **runmqckm** programları bu platformlarda 32 bit olduğu için, Windows ve Linux x86 32 bit platformları tek istisnadır.

runmqakm komutunu kullanarak:

```
-certreq -list -crypto module_name -tokenlabel token_label
-pw password -fips
```

ALW AIX, Linux, and Windows üzerinde runmqckm ve runmqakm seçenekleri

Anahtarları, sertifikaları ve sertifika isteklerini yönetmek için **runmqckm** ve **runmqakm** komut satırı seçeneklerini kullanabilirsiniz. **runmqckm** , **iKeycmd**işlevlerine benzer işlevler sağlar ve **runmqakm** , **gskitcapicmd**işlevlerine benzer işlevler sağlar.

Not: IBM MQ , SHA-3 ya da SHA-5 algoritmalarını desteklemez. Her iki algoritma da SHA-2 ailesinin üyesi olduğundan, SHA384WithRSA ve SHA512WithRSA dijital imza algoritması adlarını kullanabilirsiniz.

Deprecated Sayısal imza algoritması adları SHA3WithRSA ve SHA5WithRSA , sırasıyla SHA384WithRSA ve SHA512WithRSA kısaltması oldukları için kullanımdan kaldırılmıştır.

Bir seçeneğin anlamı, komutta belirtilen nesneye ve işleme bağlı olabilir.

*Çizelge 97. **runmqckm** ve **runmqakm** ile kullanılabilen seçenekler*

Parametre	Açıklama
`-create`	Anahtar veritabanı yaratma seçeneği.
`-crypto`	PKCS #11 şifreleme aygıtını yönetmek için kullanılan modülün adı. Özellikler dosyasında modül adını belirtirseniz, **`-crypto`** ' den sonraki değer isteğe bağlıdır. PKCS #11 şifreleme donanımında saklanan sertifikaları ya da anahtarları kullanıyorsanız, **runmqckm** ve **strmqikm** ' nin IBM MQ kuruluşuyla birlikte sağlanan Java sanal makinesi (JVM) kullanılarak çalıştırıldığına dikkat edin. PKCS #11 desteği için gereken dış modüller JVM işlemine yüklenecek; bu nedenle, JVM ' nin bit değeriyle eşleşen şifreleme donanımının yönetimi için bir PKCS #11 kitaplığınız kurulu olmalıdır ve bu kitaplığı **runmqckm** ya da **strmqikm**olarak belirtmeniz gerekir.
-db	Anahtar veritabanının tam olarak nitelenmiş yol adı.
-default_cert	Bir sertifikayı varsayılan sertifika olarak ayarlar. Değer `evet` ya da `hayır`olabilir. Varsayılan değer no' dur.
-dn	X.500 ayırt edici adı. Değer, çift tırnak içine alınmış bir dizedir; örneğin, `"CN=John Smith,O=IBM,OU=Test,C=GB"`. Yalnızca O ve C özniteliklerinin gerekli olduğunu unutmayın. Ortak ad (CN) belirtilmesi isteğe bağlıdır.
-encryption	Sertifika dışa aktarma komutunda kullanılan şifreleme gücü. Değer `güçlü` ya da `zayıf`olabilir. Varsayılan değer `strong`(güçlü) değeridir.
-expire	Bir sertifikanın ya da veritabanı parolasının son kullanma tarihi (gün). Varsayılan değer, bir sertifika parolası için 365 gündür. Veritabanı parolası için varsayılan zaman yoktur: Veritabanı parolası süre sonunu belirtik olarak ayarlamak için **`-expire`** değiştirgesini kullanın.
-file	Sertifika ya da sertifika isteğinin dosya adı.
-fips	komutun FIPS kipinde çalıştırılacağını belirtir. FIPS kipindeyken, IBM Crypto for C (ICC) bileşeni FIPS 140-2 doğrulanmış algoritmaları kullanır. ICC bileşeni FIPS kipinde başlatılmazsa, **runmqakm** komutu başarısız olur.
-format	Sertifikanın biçimi. Değer, Base64_encoded ASCII için `ascii` ya da İkili DER verileri için `binary` olabilir. Varsayılan değer `ascii`' dir.
-label	Bir sertifika ya da sertifika isteğine eklenen etiket. Sertifika, bir IBM MQ istemci uygulamasını ya da kuyruk yöneticisini tanımlamak için kullanılan kişisel bir sertifikaysa, etiketin IBM MQ sertifika etiketi (CERTLABL) ayarına karşılık gelmesi gerekir; ek bilgi için bkz. “Dijital sertifika etiketleri, gereksinimlerin anlaşılması” sayfa 26.
-new_format	Anahtar veritabanının yeni biçimi.
-new_label	Bir sertifika içe aktarma komutunda kullanılan bu seçenek, bir sertifikanın kaynak anahtar veritabanında sahip olduğu etiketten farklı bir etiketle içe aktarılmasına olanak sağlar. Sertifika, bir IBM MQ istemci uygulamasını ya da kuyruk yöneticisini tanımlamak için kullanılan kişisel bir sertifikaysa, etiketin IBM MQ sertifika etiketi (CERTLABL) ayarına karşılık gelmesi gerekir; ek bilgi için bkz. “Dijital sertifika etiketleri, gereksinimlerin anlaşılması” sayfa 26.

*Çizelge 97. **runmqckm** ve **runmqakm** ile kullanılabilen seçenekler (devamı var)*	
Parametre	**Açıklama**
-new_pw	Yeni veritabanı parolası.
-old_format	Anahtar veritabanının eski biçimi.
-pw	Anahtar veritabanı ya da PKCS #12 dosyasının parolası.
-secondaryDB	PKCS #11 aygıt işlemleri için ikincil anahtar veritabanının adı.
-secondaryDBpw	PKCS #11 aygıt işlemlerine ilişkin ikincil anahtar veritabanının parolası.
runmqakm **-secretKey** -add -create -extract	 Bir gizli anahtar ekleyin. Rasgele bir gizli anahtar oluştur Anahtar veritabanından gizli bir anahtarı çıkar
runmqckm **-secKey** -create -list -export	 Rasgele bir gizli anahtar oluşturun. Gizli anahtarları listele Gizli anahtarları dışa aktar
-showOID	Tam sertifika ya da sertifika isteğini görüntüler.
-sig_alg	Bir sertifika isteği, kendinden onaylı bir sertifika ya da bir sertifikanın imzalanması sırasında kullanılan hash algoritması. Bu hash algoritması, yeni yaratılan sertifika ya da sertifika isteğiyle ilişkili imzayı yaratmak için kullanılır. **runmqckm**için değer MD2_WITH_RSA, MD2WithRSA, MD5_WITH_RSA, MD5WithRSA, SHA1WithDSA, SHA1WithECDSA, SHA1WithRSA, SHA2/ECDSA, SHA224WithECDSA, SHA256_WITH_RSA, SHA256WithECDSA, SHA256WithRSA, SHA2WithECDSA, SHA3/ECDSA, SHA384_WITH_RSA, SHA384WithECDSA, SHA384WithRSA, SHA3WithECDSA, SHA5/ECDSA, SHA512_WITH_RSA, SHA512WithECDSA, SHA512WithRSA, SHA5WithECDSA, SHA_WITH_DSA, SHA_WITH_RSA, SHAWithDSA, SHAWithRSA. Varsayılan değer SHA1WithRSAdeğeridir. **runmqakm**için değer 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_SHA384ya da EC_ecdsa_with_SHA512. Varsayılan değer SHA1WithRSAdeğeridir.

*Çizelge 97. **runmqckm** ve **runmqakm** ile kullanılabilen seçenekler (devamı var)*	
Parametre	**Açıklama**
-size	Anahtar boyutu. **runmqckm**için değer 512, 1024 ya da 2048 olabilir. Varsayılan değer 1024 bittir. **runmqakm**için değer, imza algoritmasına bağlıdır: • RSA imza algoritmaları için (**-sig_alg** belirtilmezse kullanılan varsayılan algoritma), değer 512, 1024, 2048 ya da 4096 olabilir. **-fips** parametresi etkinleştirildiyse 512 bitlik RSA anahtarı boyutuna izin verilmez. Varsayılan RSA anahtarı boyutu 2048 bittir. • Eliptik Eğri algoritmaları için değer 256, 384 ya da 512 olabilir. Varsayılan Eliptik Eğri anahtar boyutu imza algoritmasına bağlıdır. SHA256için 256; SHA384için 384; SHA512için 512 'dir.
-stash	Anahtar veritabanı parolasını bir dosyada saklamanızı sağlar. Yalnızca CMS ve PKCS12tipindeki veritabanları için geçerlidir. **Not: -stash** , **-keydb** -create komutlarında **runmqckm**/**runmqakm** ' e parolayı içeren bir parola saklama dosyası oluşturmasını söylemek için geçerlidir. `$ runmqakm -help` komutu verilirken, yalnızca üst düzey yardım parametreleri listelenir.
-stashed	Anahtar veritabanına ilişkin parolayı ya da PKCS #12 dosyasının bir parola saklama dosyasında olduğunu gösterir. **Not: -stashed** seçeneği, **-keydb** -create komutları dışındaki çağrılarda geçerlidir. Bu seçeneği belirlemezseniz, **-pw**komutunu kullanarak parolayı girmeniz gerekir. Ayrıca, yalnızca komuta ne tür bir işlem gerçekleştirmekte olduğunuzu bildirdiğinizde, **-stashed** ' u gösteren ayrıntılı yardım görüntülenir.
-stashpw	Anahtar veritabanı parolasını bir dosyada saklamanızı sağlar. Yalnızca CMS ve PKCS12tipi veritabanları için geçerlidir.
-target	Hedef dosya ya da veritabanı.
-target_pw	**-target** bir anahtar veritabanı belirtiyorsa, anahtar veritabanının parolası.
-target_type	**-target** işleneni tarafından belirtilen veritabanı tipi. İzin verilen değerler için bkz. **-type** parametresi.
-tokenLabel	PKCS #11 şifreleme aygıtının etiketi.
-trust	Bir CA sertifikasının güven durumu. Değer `enable` ya da `disable`olabilir. Varsayılan değer `enable`' dır.
-type	Veritabanı tipi. Değer, aşağıdaki değerlerden herhangi biri olabilir: • CMS anahtar veritabanı için `cms` • Bir PKCS #12 dosyası için `pkcs12` .
-x509version	Yaratılacak X.509 sertifikasının sürümü. Değer 1, 2 ya da 3 olabilir. Varsayılan 3'tür.

*Çizelge 97. **runmqckm** ve **runmqakm** ile kullanılabilen seçenekler (devamı var)*	
Parametre	**Açıklama**

-rfc3339	Bu parametreyi, aşağıdaki biçimde olan `runmqakm -cert -details` komutu için RFC 3339 biçiminde tarih çıkışı yapmak için kullanın:
	`Not Before : 2015-08-26T08:53:37Z`
	`Not After : 2016-08-26T08:53:37Z`
	-rfc3339 değiştirgesinin ek değiştirgelerden sonra komutta görünmesi gerektiğini unutmayın:
	`runmqakm -cert -details -db exampleDB -stashed -label certficateLabel -rfc3339`

ALW AIX, Linux, and Windows üzerinde runmqakm hata kodları

runmqakmtarafından verilen sayısal hata kodlarını ve bunların ne anlama yaptıklarını içeren bir çizelge.

Hata kodu	Hata İletisi
0	Başarı
1	Bilinmeyen hata oluştu
2	Bir ASN.1 kodlama/kod çözme hatası oluştu.
3	ASN.1 kodlayıcısı/kod çözücü kullanıma hazırlanırken bir hata oluştu.
4	Aralık dışı bir dizin ya da var olmayan isteğe bağlı bir alan nedeniyle ASN.1 kodlama/kod çözme hatası oluştu.
5	Bir veritabanı hatası oluştu.
6	Veritabanı dosyası açılırken bir hata oluştu, dosyanın var olup olmadığını ve izin olup olmadığını denetleyin.
7	Veritabanı dosyası yeniden açılırken bir hata oluştu.
8	Veritabanı yaratılması başarısız oldu.
9	Veritabanı zaten var.
10	Veritabanı dosyası silinirken bir hata oluştu.
11	Veritabanı açılamadı.
12	Veritabanı dosyası okunurken bir hata oluştu.
13	Veri taban kt § ne veri yazlrken hata ortaya ‡kt.
14	Veritabanı geçerlilik denetimi hatası oluştu.
15.000	Geçersiz bir veritabanı sürümüyle karşılaşıldı.
16	Geçersiz bir veritabanı parolasıyla karşılaşıldı.
17	Geçersiz bir veritabanı dosyası tipiyle karşılaşıldı.
18	Belirtilen veritabanı bozulmuş.

Hata kodu	Hata İletisi
19	Geçersiz bir parola sağlandı ya da anahtar veritabanı kurcalanmış ya da bozulmuş.
20	Veritabanı anahtarı giriş bütünlüğü hatası oluştu.
21	Veritabanında yinelenen bir sertifika var.
22	Veritabanında yinelenen bir anahtar var (Kayıt Tanıtıcısı).
23	Anahtar veritabanında aynı etikete sahip bir sertifika zaten var.
24	Veritabanında yinelenen bir anahtar var (İmza).
25	Veritabanında yinelenen bir anahtar var (İmzalanmamış Sertifika).
26	Veritabanında yinelenen bir anahtar zaten var (Veren ve Seri Numarası).
27	Veritabanında yinelenen bir anahtar zaten var (Subject Public Key Info).
28	Veritabanında yinelenen bir anahtar zaten var (İmzalanmamış CRL).
29	Etiket veritabanında kullanıldı.
30	Parola şifreleme hatası oluştu.
31	LDAP ile ilgili bir hata oluştu. (LDAP bu program tarafından desteklenmiyor)
24	Şifreleme hatası oluştu.
33	Şifreleme/şifre çözme hatası oluştu.
34	Geçersiz bir şifreleme algoritması bulundu.
%35	Veriler imzalanırken bir hata oluştu.
36	Veriler doğrulanırken hata oluştu.
37	Verilerin özeti hesaplanırken bir hata oluştu.
38	Geçersiz bir şifreleme parametresi bulundu.
39	Desteklenmeyen bir şifreleme algoritmasıyla karşılaşıldı.
40	Belirtilen giriş büyüklüğü, desteklenen modülo büyüklüğünden fazla.
41	Desteklenmeyen bir modül boyutu bulundu.
42	Veritabanı geçerlilik denetimi hatası oluştu.
43	Anahtar girişi geçerlilik denetimi başarısızlıkla sonuçlandı.
44	Yinelenen bir uzantı alanı var.
45	Anahtarın sürümü yanlış.
46	Gerekli bir uzantı alanı yok.

Hata kodu	Hata İletisi
47	Geçerlilik süresi bugünü içermiyor ya da sertifika verenin geçerlilik süresi içinde yer almıyor
48	Geçerlilik süresi bugünü içermiyor ya da sertifika verenin geçerlilik süresi içinde yer almıyor.
49	Özel anahtar kullanım uzantısının geçerliliği denetlenirken bir hata oluştu.
50	Anahtarı veren bulunamadı.
51	Gerekli bir sertifika uzantısı eksik.
52	Geçersiz bir temel kısıt uzantısı bulundu.
53	Anahtar imzası geçerlilik denetimi başarısızlıkla sonuçlandı.
54	Anahtarın kök anahtarı güvenilir değil.
55	Anahtar iptal edildi.
56	Yetki anahtarı tanıtıcısı uzantısının geçerliliği denetlenirken hata oluştu.
57	Özel anahtar kullanım uzantısının geçerliliği denetlenirken bir hata oluştu.
58	Konu alternatif ad uzantısının geçerliliği denetlenirken bir hata oluştu.
59	Sertifika veren diğer ad uzantısının geçerliliği denetlenirken hata oluştu.
60	Anahtar kullanım uzantısının geçerliliği denetlenirken bir hata oluştu.
61	Bilinmeyen bir kritik uzantı bulundu.
62	Anahtar çifti girişleri doğrulanırken bir hata oluştu.
63	CRL doğrulanırken bir hata oluştu.
64	Muteks hatası oluştu.
65	Geçersiz bir parametre bulundu.
86	Boş değerli bir parametre ya da bellek ayırma hatasıyla karşılaşıldı.
%67	Sayı ya da boyut çok büyük ya da çok küçük.
75	Eski parola geçersiz.
75	Yeni parola geçersiz.
70	Parolanın süresi doldu.
77	İş parçacığıyla ilgili bir hata oluştu.
68	İş parçacıkları yaratılırken hata oluştu.
73	Bir iş parçacığı çıkmayı beklerken hata oluştu.
74	Bir G/Ç hatası oluştu.
75	CMSyüklenirken bir hata oluştu.

Hata kodu	Hata İletisi
76	Şifreleme donanımıyla ilgili bir hata oluştu.
77	Kitaplık kullanıma hazırlama yordamı başarıyla çağrılamadı.
65	İç veritabanı tanıtıcısı çizelgesi bozuk.
65	Bellek ayırma hatası oluştu.
80	Tanınmayan bir seçenek bulundu.
81	Saat bilgileri alınırken hata oluştu.
82	Muteks oluşturma hatası oluştu.
76	İleti kataloğu açılırken hata oluştu.
84	Hata iletisi kataloğu açılırken hata oluştu
85	Boş değerli bir dosya adı bulundu.
69	Dosyalar açılırken bir hata oluştu, dosyanın varlığını ve izinlerini denetleyin.
87	Okunacak dosyalar açılırken bir hata oluştu.
88	Yazılacak dosyalar açılırken bir hata oluştu.
89	Böyle bir dosya yok.
90	İzin ayarı nedeniyle dosya açılamıyor.
91	Dosyalara veri yazılırken hata oluştu.
92	Dosyalar silinirken bir hata oluştu.
93	Geçersiz Base64-encoded veri bulundu.
94	Geçersiz bir Base64 ileti tipi bulundu.
95	Veriler Base64 kodlama kuralıyla kodlanırken bir hata oluştu.
96	Base64-encoded verilerin kodu çözülürken bir hata oluştu.
97	Ayırt edici ad etiketi alınırken hata oluştu.
98	Gerekli ortak ad alanı boş.
99	Gerekli ülke ya da bölge adı alanı boş.
100	Geçersiz bir veritabanı tanıtıcısı bulundu.
101	Anahtar veritabanı yok.
102	İstek anahtarı çifti veritabanı yok.
103	Parola dosyası yok.
104	Yeni parola eskisiyle aynı.
105	Anahtar veritabanında anahtar bulunamadı.
106	İstek anahtarı bulunamadı.
107	Güvenilir bir CA bulunamadı.
108	Sertifika için istek anahtarı bulunamadı.

Hata kodu	Hata İletisi
109	Anahtar veritabanında özel anahtar yok.
110	Anahtar veritabanında varsayılan anahtar yok.
111	Anahtar kaydında özel anahtar yok.
112	Anahtar kaydında sertifika yok.
113	CRL girişi yok.
114	Geçersiz bir anahtar veritabanı dosyası adı bulundu.
115	Tanınmayan bir özel anahtar tipi bulundu.
116	Geçersiz bir ayırt edici ad girişi bulundu.
117	Belirtilen anahtar etiketine sahip bir anahtar girişi bulunamadı.
118	Anahtar etiketi listesi bozulmuş.
119	Giriş verileri geçerli PKCS12 verileri değil.
120	Parola geçersiz ya da PKCS12 verileri bozulmuş ya da sonraki PKCS12 sürümüyle yaratılmış.
121	Tanınmayan bir anahtar dışa aktarma tipi bulundu.
122	Desteklenmeyen bir parola tabanlı şifreleme algoritması bulundu.
123	Anahtarlık dosyası CMS anahtar veritabanına dönüştürülürken bir hata oluştu.
124	CMS anahtar veritabanı bir anahtarlık dosyasına dönüştürülürken hata oluştu.
125	Sertifika isteği için sertifika yaratılırken hata oluştu.
126	Tam bir sertifika veren zinciri oluşturulamıyor.
127	Geçersiz WEBDB verileri bulundu.
128	Anahtarlık dosyasına yazılacak veri yok.
129	Girdiğiniz gün sayısı, izin verilen geçerlilik süresini aşıyor.
130	Parola çok kısa; en az {0} karakterden oluşmalıdır.
131	Bir parola en az bir sayısal sayı içermelidir.
132	Paroladaki tüm karakterler alfabetik ya da sayısal karakterlerdir.
133	Tanınmayan ya da desteklenmeyen bir imza algoritması belirtildi.
134	Geçersiz bir veritabanı tipiyle karşılaşıldı.
135	Belirtilen ikincil anahtar veritabanı başka bir PKCS#11 aygıtı tarafından kullanılıyor.
136	İkincil anahtar veritabanı belirtilmedi.
137	Etiket PKCS#11 aygıtında yok.

Hata kodu	Hata İletisi
138	PKCS#11 aygıtına erişmek için parola gerekli.
139	PKCS#11 aygıtına erişmek için parola gerekli değil.
140	Şifreleme kitaplığı yüklenemiyor.
141	PKCS#11 bu işlem için desteklenmiyor.
142	PKCS#11 aygıtındaki bir işlem başarısız oldu.
143	LDAP kullanıcısı geçerli bir kullanıcı değil. (LDAP bu program tarafından desteklenmiyor)
144	LDAP kullanıcısı geçerli bir kullanıcı değil. (LDAP bu program tarafından desteklenmiyor)
145	LDAP sorgusu başarısız oldu. (LDAP bu program tarafından desteklenmiyor)
146	Geçersiz bir sertifika zinciri bulundu.
147	Kök sertifika güvenilir değil.
148	İptal edilen bir sertifikayla karşılaşıldı.
149	Bir şifreleme nesnesi işlevi başarısız oldu.
150	Kullanılabilir sertifika iptal listesi veri kaynağı yok.
151	Kullanılabilir şifreleme aygıtı yok.
152	FIPS kipi kullanılamıyor.
153	FIPS kipi ayarlarıyla bir çakışma var.
154	Girilen parola, gerekli güvenlik düzeyi alt sınırını karşılamıyor.
200	Program başlatılırken bir hata oluştu.
201	Runmqakm Programına geçirilen bağımsız değişkenlerin bölümlenmesi başarısız oldu.
202	Komutta belirtilen nesne tanınan bir nesne değil.
203	Geçirilen işlem bilinen bir -keydb işlemi değil.
204	Geçirilen işlem bilinen bir -cert işlemi değil.
205	Geçirilen işlem, bilinen bir -certreq işlemi değil.
206	İstenen komut için bir etiket eksik.
207	-version etiketiyle geçirilen değer, tanınan bir değer değil.
208	-size etiketiyle geçirilen değer tanınan bir değer değil.
209	-dn etiketiyle geçirilen değer doğru biçimde değil.
210	-format etiketiyle geçirilen değer tanınan bir değer değil.
211	Dosya açılırken bir hata oluştu.
212	PKCS12 bu aşamada desteklenmiyor.

Hata kodu	Hata İletisi
213	Parolasını değiştirmeye çalıştığınız şifreleme simgesi parola korumalı değil.
214	PKCS12 bu aşamada desteklenmiyor.
215	Girilen parola, gerekli güvenlik düzeyi alt sınırını karşılamıyor.
216	FIPS kipi kullanılamıyor.
217	Süre bitim tarihi olarak girdiğiniz gün sayısı, izin verilen aralığın dışında.
218	Parola güvenlik düzeyi, minimum gereksinimleri karşılayamadı.
219	İstenen anahtar veritabanında Varsayılan sertifika bulunamadı.
220	Geçersiz bir güven durumuyla karşılaşıldı.
221	Desteklenmeyen bir imza algoritmasıyla karşılaşıldı. Bu aşamada yalnızca Deprecated MD5 ve Deprecated SHA1 desteklenir.
222	PCKS11 belirli bir işlem için desteklenmiyor.
223	Geçirilen işlem bilinen bir rasgele işlem değil.
224	Sıfırdan küçük bir uzunluğa izin verilmez.
225	-strong etiketi kullanırken, parola uzunluğu alt sınırı 14 karakterdir.
226	-strong etiketi kullanırken, parola uzunluğu üst sınırı 300 karakterdir.
227	MD5 algoritması FIPS kipindeyken desteklenmez.
228	-cert -list komutu için site etiketi desteklenmiyor. Bu öznitelik, geriye dönük uyumluluk ve gelecekteki olası geliştirme için eklenir.
229	-ca etiketiyle ilişkili değer tanınmıyor. Değer 'true' ya da 'false ' olmalıdır.
230	-type etiketiyle geçirilen değer geçerli değil.
231	-expire etiketiyle geçirilen değer, izin verilen aralığın altında.
232	Kullanılan ya da istenen şifreleme algoritması desteklenmiyor.
233	Hedef zaten var.

IBM MQ bileşeni yapılandırma dosyalarındaki parolaları koruma

IBM MQ'in belirli özelliklerini kullanmak için, parolaların doğrudan IBM MQ ' e ya da özelliğin okuduğu yapılandırma dosyalarında sağlanması gerekebilir. IBM MQ 9.2.0' den, bu yapılandırma dosyalarındaki parolaları koruyan bir parola koruma sistemi uygulanır.

Yapılandırma dosyalarındaki parolalar şifrelenmelidir. Aşağıdaki listede, her bileşen için kullanılan ortak terminoloji açıklanmaktadır:

İlk anahtar
Parolayı korumak için kullanılan şifreleme anahtarı.

Listelenen her bileşen için, ilgili bileşenin yapılandırmasında saklanan parolaları korumak için kullanılan benzersiz bir başlangıç anahtarı sağlayın. Parolanın şifresinin çözülebilmesi için aynı başlangıç anahtarının bileşen tarafından da kullanılabilir kılınması gerekir.

Çoğu bileşen, ilk anahtarın bir dosyada sağlanmasını gerektirir. İlk anahtar dosyası:

- **En az bir karakterden oluşan tek bir satır içerir.**
- İşletim sistemi izinlerini kullanarak yeterli koruma sağlayın.

İlk anahtarın uzunluğuna ya da belirtilebilir karakterlere ilişkin herhangi bir gereksinim yoktur. Ancak, yeterli güvenlik için, en az 16 karakter uzunluğunda bir başlangıç anahtarı belirtmeniz gerekir. Örneğin, ilk anahtar dosyanız şunları içerebilir:

```
Th1sIs@n3Ncypt|onK$y
```

Varsayılan başlangıç anahtarı
Verileri şifrelerken bir ilk anahtar belirtmezseniz, kullanılan varsayılan şifreleme anahtarı. Ancak, şifrelenmiş verileri yeterince korumadığı için varsayılan başlangıç anahtarını **kullanmamanız** gerekir.

Düz metin dizgisi
Şifrelenen dizgi, genellikle bir parola.

Şifrelenmiş parola dizgisi
Şifrelenmiş parolayı IBM MQ ' in anladığı biçimde içeren bir dize.

Önemli: Bir bileşenle kullanmak üzere oluşturacağınız şifrelenmiş parola dizgileri, başka bir bileşenin yapılandırma dosyasına kopyalanamaz. Her bileşene ilişkin her parola, bileşene özgü yardımcı program kullanılarak korunmalıdır.

Parola korumayı destekleyen her IBM MQ bileşeni için parolaların nasıl korunacağına ilişkin ayrıntılar aşağıdaki bölümlerde listelenmiştir:

- Gelişmiş İleti Güvenliği
- “Managed File Transfer” sayfa 577
- “IBM MQ Internet Pass-Thru” sayfa 578
- Deprecated “IBM MQ Bridge to blockchain” sayfa 579
- Deprecated “IBM MQ Bridge to Salesforce” sayfa 579
- V 9.3.0 “Şifreleme donanımını kullanan IBM MQ clients” sayfa 580
- “IBM MQ Kuyruk Yöneticisi” sayfa 581
- V 9.3.0 “IBM MQ C istemci uygulamaları” sayfa 581
- V 9.3.2 “Yerel HA yapılandırmaları” sayfa 581
- V 9.3.4 “IBM MQ kuyruk yöneticisi (qm.ini dosyasındaAuthToken kısmı)” sayfa 582

Advanced Message Security

Advanced Message Security (AMS) Java istemcilerinin, iletiyi korumak için özel anahtarlar içeren bir anahtar deposuna erişmeleri gerekir.

V 9.3.0 Advanced Message Security (AMS) MCA müdahalesi gerçekleştirmek üzere yapılandırılan MQI istemcileri ya da kuyruk yöneticileri, iletileri korumak için özel anahtarlar içeren PKCS#11 şifreleme donanımına ya da PEM dosyalarına erişim gerektirebilir.

Bu dosyalara erişmek için, `keystore.conf`olarak adlandırılan AMS yapılandırma dosyasında bir parola sağlanmalıdır. `keystore.conf` dosyasında bulunan hassas bilgileri korumak için **runamscred** komutunu kullanın. Örneğin,

```
runamscred -f <keystore configuration file>
```

runamscred komutu, **-f** işaretini kullanarak belirtilen dosya içindeki hassas parametreleri korur.

V 9.3.0 IBM MQ kuruluşuna iki **runamscred** programı eklenir:

- `<IBM MQ installation root>/bin` içinde bulunan bir MQI **runamscred** programı
- `<IBM MQ installation root>/java/bin` içinde bulunan bir Java **runamscred** programı

Uyarı: Uyumluluğu sağlamak için,

1. V 9.3.0 MQI AMS istemcileriyle kullanılacak yapılandırma dosyalarını korumak üzere Java AMS istemcileri ve MQI **runamscred** programıyla kullanılacak yapılandırma dosyalarını korumak için Java **runamscred** programını kullanın.
2. **runamscred**çalıştırıldıktan sonra gerekli tüm hassas bilgilerin korunduğunu doğrulayın.
3. Korumalı dosyayı, AMS etkin uygulamalara normal şekilde sağlayın.

AMS uygulamalarının çalıştırma zamanında kullanılacak ilk anahtar dosyasını geçersiz kılmak ya da sağlamak için ya da **runamscred**komutunu kullanarak bir anahtar deposu yapılandırma dosyasını koruduğunuzda, öncelik sırasına göre aşağıdaki dört mekanizmalardan birini kullanın:

1. **-sf** parametresi (yalnızca**runamscred**)
2. **MQS_AMSCRED_KEYFILE** ortam değişkeni
3. `keystore.conf` yapılandırma dosyasındaki **amscred.keyfile** parametresi
4. Önceki seçeneklerin hiçbiri belirtilmezse, varsayılan ilk anahtar dosyası.

Uyarı: V 9.3.0 Varsayılan başlangıç anahtarını kullanmayın.

IBM MQ 9.2' den önce, AMS Java yapılandırma dosyalarındaki parolaları korumak için farklı bir parola koruma sistemi kullanılırdı.

Varsayılan olarak, **runamscred** programı yeni sistemi kullanarak parolaları korur. Bu, yeni yapılandırma dosyalarının eski AMS Javasürümleriyle uyumlu olmadığı anlamına gelir. Eski parola koruma sistemiyle yapılandırma dosyalarını korumak için **-sp** *0* işaretini kullanın.

Managed File Transfer

Managed File Transfer (MFT), birden çok XML özellik dosyasında kuyruk yöneticilerine ya da diğer kaynaklara erişmek için gereken kimlik bilgilerini saklar:

- `MQMFTCredentials.xml` -Güvenli iletişim için anahtar depolarına bağlanmak üzere aracı, eşgüdüm ve komut kuyruğu yöneticilerine ve parolalara bağlanmak için kullanılan kimlik bilgileri.
- `ProtocolBridgeCredentials.xml` -FTP/SFTP/FTPS gibi Protokol Sunucularına bağlanma kimlik bilgileri.
- `ConnectDirectCredentials.xml` - Connect:Direct aracısının bir Connect:Direct düğümüne bağlanması için kimlik bilgileri.

Daha fazla bilgi için bkz “MFT içinde saklanan kimlik bilgilerini şifreleme” sayfa 585.

Bu dosyalarda saklanan duyarlı bilgileri korumak için, **-f** işaretini kullanarak belirtilen dosyada fteObfuscate komutunu kullanın; örneğin:

```
fteObfuscate -f <File to protect>
```

MFT yapılandırmalarınızın korunması sırasında kullanılacak ilk anahtar dosyasını sağlamak için **-sf** işaretini kullanın:

```
fteObfuscate -f <File to protect> -sf <initial key file>
```

İlk anahtarı belirtmezseniz, hassas bilgileri korumak için varsayılan bir anahtar kullanılır, ancak bu seçeneği kullanmamanız gerekir.

Uyarı:

1. **fteObfuscate**çalıştırıldıktan sonra gerekli tüm hassas bilgilerin korunduğunu doğrulayın.
2. Korumalı dosyayı MFT' e normal şekilde sağlayın.

Çalıştırma zamanında, aşağıdaki üç mekanizma aracılığıyla kullanılacak ilk anahtar dosyasını sağlayın, öncelik sırasına göre:

1. Java sistem özelliğini kullanarak.
   - V 9.3.0.10 V 9.3.1 IBM MQ 9.3.1 ve IBM MQ 9.3.0 Fix Pack 10öncesinde, bu Java sistem özelliğinin adı ürün kodunda `com.ibm.wqmfte.cred.keyfile`olarak yanlış yazılıyordu. IBM MQ 9.3.1 ve IBM MQ 9.3.0 Fix Pack 10' den özellik adının yazımı `com.ibm.wmqfte.cred.keyfile`olarak düzeltilir. Managed File Transfer , bir kullanıcının kimlik bilgilerini şifrelemek ve kimlik bilgilerinin şifresini çözmek için kullanılacak ilk anahtarı içeren bir dosya belirtip belirtmediğini denetlerken Java sistem özelliğinin her iki sürümünü de kullanır. Bu, eski yanlış yazılmış adla önceki bir sürümle uyumluluğu korurken özellik adının doğru yazımının kullanılmasını sağlar. Her iki Java sistem özelliği de ayarlanırsa, doğru yazılmış `com.ibm.wmqfte.cred.keyfile` özelliğinin değeri kullanılır.
   - IBM MQ 9.3.1 ve IBM MQ 9.3.0 Fix Pack 10' den önce `com.ibm.wqmfte.cred.keyfile`özelliğini kullanın.
2. Aracı, kaydedici, komutlar ve koordinasyon özellik dosyalarında.
3. `installation.properties` dosyasında.

IBM MQ 9.2' den önce, MFT yapılandırma dosyalarındaki kimlik bilgilerini korumak için farklı bir kimlik bilgileri koruma sistemi kullanılırdı.

Varsayılan olarak **fteObfuscate** , yeni sistemi kullanarak kimlik bilgilerini korur; bu, yapılandırma dosyalarının eski MFTsürümleriyle uyumlu olmadığı anlamına gelir.

Eski kimlik bilgileri koruma sistemiyle yapılandırma dosyalarını korumak için **-sp** *0* parametresini kullanın.

IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru (MQIPT) yapılandırma dosyası, çeşitli kaynaklara ve MQIPT yönetim parolasına erişmek için parola içerebilir.

Bu parolaları, MQIPTile verilen **mqiptPW** komutunu kullanarak koruyabilirsiniz.

```
mqiptPW
```

Bir parolayı belirli bir başlangıç anahtarıyla korumak için **-sf** işaretini sağlayın:

```
mqiptPW -sf <intial key file>
```

Daha fazla bilgi için bkz. Parola şifreleme anahtarının belirtilmesi.

İlk anahtarı belirtmezseniz, hassas bilgileri korumak için varsayılan bir anahtar kullanılır, ancak bu seçeneği kullanmamanız gerekir.

mqiptPW , korumak için güvenli bir parola girmenizi ister ve MQIPT yapılandırma dosyasına kopyalanması gereken bir dizgi döndürür.

Çalıştırma zamanında, aşağıdaki dört mekanizma aracılığıyla kullanılacak ilk anahtar dosyasını sağlayın. Öncelik sırasına göre şunlar:

1. MQIPT başlatıldığında **-sf** parametresiyle.
2. MQS_MQIPTCRED_KEYFILE ortam değişkeninde.
3. **com.ibm.mq.ipt.cred.keyfile** Java özelliğinde.
4. MQIPT ana dizininde `mqipt_cred.key` adlı bir dosyada, MQIPT yapılandırma ve günlük dosyalarını ve diğerlerini içeren dizin bulunur.

IBM MQ 9.2' den önce, MQIPT yapılandırma dosyalarındaki kimlik bilgilerini korumak için farklı bir kimlik bilgileri koruma sistemi kullanılırdı.

Varsayılan olarak **mqiptPW** , yeni sistemi kullanan kimlik bilgilerini korur; bu, yapılandırma dosyalarının eski MQIPTsürümleriyle uyumlu olmadığı anlamına gelir.

Eski kimlik bilgileri koruma sistemini kullanan anahtar deposu parolalarını korumak için, IBM MQ 9.2öncesi sürümlerde desteklenen **mqiptPW** komut sözdizimini kullanın.

IBM MQ Bridge to blockchain

Deprecated

Bridge to blockchain yapılandırmaları, **runmqbcb** komutuyla oluşturulabilen dosyalarda saklanır. Bu komutu çalıştırdığınızda, parolaları ve kullanılacak ilk anahtar dosyasının konumunu güvenli bir şekilde sağlamanız istenecektir.

Çalıştırma zamanı ya da yapılandırma kipi sırasında kullanılacak ilk anahtar dosyasını geçersiz kılmak için **-sf** işaretini kullanın. Örneğin, belirli bir ilk anahtar dosyasıyla bir yapılandırma oluşturun:

```
runmqbcb -o <output file> -sf <initial key file>
```

Ya da yürütme sırasında belirli bir ilk anahtar dosyasını kullanmak için:

```
runmqbcb -f <config file> -sf <initial key file>
```

IBM MQ 9.2' den önce, Bridge to blockchain yapılandırma dosyalarındaki kimlik bilgilerini korumak için farklı bir kimlik bilgileri koruma sistemi kullanılırdı.

Varsayılan olarak **runmqbcb** , yeni sistemi kullanarak kimlik bilgilerini korur; bu, yapılandırma dosyalarının eski Bridge to blockchainsürümleriyle uyumlu olmadığı anlamına gelir.

Yapılandırma dosyalarını eski kimlik bilgileri koruma sistemiyle korumak için **-sp** *0* işaretini kullanın.

Önemli:

- Deprecated IBM MQ Bridge to blockchain , 22 Kasım 2022 'deki tüm yayınlarda kullanımdan kaldırılmıştır (bkz. ABD Duyurusu mektubu 222-341). Blockchain bağlanırlığı, IBM App Connect ile ya da IBM Cloud Pak for Integrationile sağlanan App Connect yetenekleriyle elde edilebilir.
- Continuous Deliveryiçin IBM MQ Bridge to blockchain , IBM MQ 9.3.2adresindeki üründen kaldırılır.

IBM MQ Bridge to Salesforce

Deprecated

Bridge to Salesforce yapılandırmaları, **runmqsfb** komutuyla oluşturulabilen dosyalarda saklanır. Bu komutu çalıştırırken, parolaları ve kullanılacak ilk anahtar dosyasının konumunu güvenli bir şekilde sağlamanız isteniyor.

Çalıştırma zamanı ya da yapılandırma kipi sırasında kullanılacak ilk anahtar dosyasını geçersiz kılmak için **-sf** işaretini kullanın. Örneğin, belirli bir ilk anahtar dosyasıyla yapılandırma oluşturmak için:

```
runmqsfb -o <output file> -sf <initial key file>
```

Ya da yürütme sırasında belirli bir ilk anahtar dosyasını kullanmak için:

```
runmqsfb -f <config file> -sf <initial key file>
```

IBM MQ 9.2' den önce, Bridge to Salesforce yapılandırma dosyalarındaki kimlik bilgilerini korumak için farklı bir kimlik bilgileri koruma sistemi kullanılırdı.

Varsayılan olarak **runmqfsb** , yeni sistemi kullanarak kimlik bilgilerini korur; bu, yapılandırma dosyalarının eski Bridge to Salesforcesürümleriyle uyumlu olmadığı anlamına gelir.

Yapılandırma dosyalarını eski kimlik bilgileri koruma sistemiyle korumak için **-sp** *0* işaretini kullanın.

Önemli: IBM MQ Bridge to Salesforce , 22 Kasım 2022 'deki tüm yayınlarda kullanımdan kaldırılmıştır (bkz. ABD Duyuru mektubu 222-341).

Şifreleme donanımını kullanan IBM MQ clients

V 9.3.0

TLS iletişimlerinde kullanılan özel anahtarları ve sertifikaları depolamak için IBM MQ istemcilerini PKCS #11 şifreleme donanımını kullanacak şekilde yapılandırabilirsiniz. PKCS #11 aygıtlarına erişmek için, IBM MQ client' a sağlanan yapılandırma dizgisinin bir parçası olarak bir parola sağlamanız gerekir.

Önemli: MQCSO yapısında **CryptoHardware** alanı kullanılarak sağlanan parolalar ya da kuyruk yöneticisi **SSLCRYP** özniteliği bu düzenek kullanılarak korunamaz.

Bu parolayı, IBM MQ kuruluş dizinindeki `bin` klasöründe bulunan **runp11cred** komutunu kullanarak koruyabilirsiniz.

runp11cred komutu, parolanın şifrelenip şifrelenmeyeceğini sorar ve şifrelenmiş parolayı döndürür. Şifrelenmiş parola, şifreleme donanımı yapılandırma dizesine kopyalanmalıdır.

Örneğin, şifreleme donanımı yapılandırma dizeniz aşağıdaysa:

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;Passw0rd;SYMMETRIC_CIPHER_ON
```

runp11cred komutu parolayı girmenizi istediğinde `Passw0rd`girin. Komut, aşağıdakine benzer bir dizgi döndürür:

```
<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHmSNF0/ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==
```

Şifrelenmiş parolayı içeren şu dizgiyi vermek için şifreleme donanımı yapılandırma dizgisindeki parolayı **runp11cred** komutu tarafından döndürülen dizgiyle değiştirin:

```
GSK_PKCS11=/usr/lib/pkcs11/PKCS11_API.so;tokenlabel;<P11>!2!0TyDxrRaS6JUsjON9zfK6S4wEHm SNF0/
ZsOdCaTD2dc=!MdpCoxGnFqPtZ1dTLQ58kg==;SYMMETRIC_CIPHER_ON
```

Şifrelenmiş parolayı içeren şifreleme donanımı yapılandırma dizgisini, istemci yapılandırma dosyasının SSL kısmına ilişkin **SSLCryptoHardware** özniteliğinde ya da **MQSSLCRYP** ortam değişkenine saklayın.

Varsayılan olarak, **runp11cred** komutu bir parolayı varsayılan başlangıç anahtarıyla şifreler. Bir parolayı kendi ilk anahtarınızla korumak için, öncelik sırasına göre aşağıdaki mekanizmalardan birini kullanarak ilk anahtarı içeren dosyanın adını belirtin:

1. **runp11cred** komutuna ilişkin **-sf** parametresi.
2. **MQS_SSLCRYP_KEYFILE** ortam değişkeni.

DİKKAT: Parolaları güvenli bir şekilde korumadığı için parolaları şifrelemek için varsayılan başlangıç anahtarını kullanmayın.

Parola şifrelendiğinde bir ilk anahtar dosyası belirtilirse, IBM MQ client çalıştırıldığında ilk anahtarı içeren dosyanın adını da belirtmeniz gerekir. İlk anahtar dosyası adını, öncelik sırasına göre aşağıdaki mekanizmalardan birini kullanarak belirleyin:

1. **MQS_SSLCRYP_KEYFILE** ortam değişkeni.
2. İstemci yapılandırma dosyasının **SSL** kısmına ilişkin **SSLCryptoHardwareKeyFile** özniteliği.

IBM MQ Kuyruk Yöneticisi

IBM MQ kuyruk yöneticisi, parolaları çeşitli özniteliklerde (örneğin, kuyruk yöneticisi **KEYRPWD**) depolar. IBM MQ , parolayı diskteki dosyalarda saklamadan önce otomatik olarak şifreler.

Anahtar deposu parolası, IBM MQ parola koruma sistemi ya da bir anahtar deposu parola saklama dosyası kullanılarak korunabilir. Bu iki yöntem hakkında daha fazla bilgi için bkz. “AIX, Linux, and Windows üzerinde anahtar havuzu parolalarını şifreleme” sayfa 293.

Kuyruk yöneticisi bir parolayı şifrelediğinde, kuyruk yöneticisi nesnesinde **INITKEY** özniteliği kullanılarak alternatif bir anahtar belirtilmedikçe varsayılan ilk anahtar kullanılır. Şifrelenecek parolaları sağlamadan önce benzersiz, güçlü bir anahtar ayarlayın.

Uyarı: Anahtar havuzu parolasını sağladıktan sonra ilk anahtarın değiştirilmesi, anahtar havuzu geçiş tümceciğinin yeni başlangıç anahtarıyla şifrelenmesine neden olmaz. Bu nedenle, anahtar havuzu parolası yeniden sağlanmadan ilk anahtarın değiştirilmesi, IBM MQ ' in anahtar havuzu parolası şifresini çözememesine ve bu nedenle anahtar havuzuna erişememesine neden olur.

Daha fazla bilgi için bkz. INITKEY.

IBM MQ C istemci uygulamaları

V 9.3.0

IBM MQ C istemcisi kitaplıkları, belirli güvenli kaynaklara (örneğin, kuyruk yöneticisine bağlanmak için TLS kullanan uygulamalar için TLS anahtar deposu) erişmek için parola gerektirir.

Anahtar deposu parolası, IBM MQ parola koruma sistemi ya da bir anahtar deposu parola saklama dosyası kullanılarak korunabilir. Bu iki yöntem hakkında daha fazla bilgi için bkz. “AIX, Linux, and Windows üzerinde anahtar havuzu parolalarını şifreleme” sayfa 293.

Parolaları IBM MQ parola koruma sistemiyle korumak için **runmqicred** komutunu kullanın. Komut, *MQ_INSTALLATION_PATH*/bin dizininde bulunur.

runmqicred komutu, şifrelenecek parolayı girmenizi ister ve düz metin parolası yerine kullanılabilecek şifrelenmiş parolayı döndürür.

Örneğin, *MQKEYRPWD* ortam değişkenini kullanarak bir TLS anahtar deposu parolası belirtmeyi seçerseniz ve TLS anahtar deposu parolanız Passw0rdise. **runmqicred**komutunu çalıştırdığınızda, istendiğinde Passw0rd girin. Komut, aşağıdakine benzer bir dizgi döndürür:

```
<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w==
```

Bu dizgiyi *MQKEYRPWD* ortam değişkeninin değeri olarak ayarlayın:

```
export MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
set MQKEYRPWD="<MQI>!2!G4lRxBuiNfJ3uOeYTD3lG1hrL5NvVZLAlgZCX3Tn6d8=!pUDOErDfDi9+JFVa0usS7w=="
```

Varsayılan olarak, **runmqicred** komutu bir parolayı varsayılan başlangıç anahtarıyla şifreler. Bir parolayı kendi ilk anahtarınızla korumak için, öncelik sırasına göre anahtarı içeren dosyanın adını belirtmek üzere aşağıdaki mekanizmalardan birini kullanın:

1. **runmqicred** komutuna ilişkin **-sf** parametresi.
2. **MQS_MQI_KEYFILE** ortam değişkeni.

DİKKAT: Parolaları güvenli bir şekilde korumadığı için parolaları şifrelemek için varsayılan başlangıç anahtarını kullanmayın.

Daha fazla bilgi için bkz “AIX, Linux, and Windows üzerinde IBM MQ MQI client için anahtar havuzu parolası sağlanması” sayfa 299.

Yerel HA yapılandırmaları

V 9.3.2

Eşgörünümler arasındaki yerel HA günlük eşleme trafiği TLS kullanılarak şifrelenebilir. Günlük eşleme trafiğinin güvenliğini sağlamak için kullanılan sertifikalar, `qm.ini` dosyasının **NativeHALocalInstance** kısmı içinde belirtilen bir anahtar deposunda saklanır.

Anahtar deposu parolası, IBM MQ parola koruma sistemi ya da bir anahtar deposu parola saklama dosyası kullanılarak korunabilir. Bu iki yöntem hakkında daha fazla bilgi için bkz. "AIX, Linux, and Windows üzerinde anahtar havuzu parolalarını şifreleme" sayfa 293.

Yerli HA anahtar deposu parolasını IBM MQ parola koruma sistemiyle korumak için **runmqicred** komutunu kullanın.

runmqicred komutu, şifrelenecek parolanın girilmesini ister ve düz metin parolası yerine kullanılması gereken şifrelenmiş parolayı döndürür. `qm.ini` dosyasının **NativeHALocalInstance** kısmına ilişkin **KeyRepositoryPassword** özniteliğinin değerini, komutun döndürdüğü şifrelenmiş parolaya ayarlayın.

Varsayılan olarak, **runmqicred** komutu bir parolayı varsayılan başlangıç anahtarıyla şifreler. Bir parolayı kendi ilk anahtarınızla korumak için, öncelik sırasına göre anahtarı içeren dosyanın adını belirtmek üzere aşağıdaki mekanizmalardan birini kullanın:

1. **runmqicred** komutuna ilişkin **-sf** parametresi.
2. *MQS_MQI_KEYFILE* ortam değişkeni.

DİKKAT: Parolaları güvenli bir şekilde korumadığı için parolaları şifrelemek için varsayılan başlangıç anahtarını kullanmayın.

Anahtar deposu parolasını kendi ilk anahtarınızla şifrelediyseniz, `qm.ini` dosyasının **NativeHALocalInstance** kısmına ilişkin **InitialKeyFile** özniteliğini kullanarak aynı ilk anahtar dosyasını da belirtmeniz gerekir.

Daha fazla bilgi için bkz. NativeHALocal qm.ini dosyasının eşgörünüm kısmı.

IBM MQ kuyruk yöneticisi (`qm.ini` dosyasındaAuthToken kısmı)

IBM MQ 9.3.4' den AIX ya da Linux sistemlerinde çalışan IBM MQ kuyruk yöneticilerine bağlanan IBM MQ MQI clients , kuyruk yöneticisiyle kimlik doğrulaması yapmak için kimlik doğrulama belirteçlerini kullanabilir. Kuyruk yöneticisi, kimlik doğrulama belirteçlerini kabul edecek ve belirteç yayıncısının genel anahtar sertifikasına ya da simgeyi imzalamak için kullanılan gizli anahtara erişebilecek şekilde yapılandırılmalıdır. Güvenilen sertifika verenin genel anahtar sertifikalarını ya da gizli anahtarlarını içeren anahtar deposu bir parolayla korunmuştur.

Anahtar deposu parolası, IBM MQ parola koruma sistemi ya da bir anahtar deposu parola saklama dosyası kullanılarak korunabilir. Bu iki yöntem hakkında daha fazla bilgi için bkz. "AIX, Linux, and Windows üzerinde anahtar havuzu parolalarını şifreleme" sayfa 293.

IBM MQ parola koruma sistemiyle kimlik doğrulama simgesi anahtar deposu parolasını korumak için **runqmcred** komutunu kullanarak parolayı şifreleyin.

Parolayı belirli bir başlangıç anahtarıyla şifrelemek için, başlangıç anahtarını içeren dosyanın yolunu belirtmek üzere **-sf** parametresini kullanın. İlk anahtarı sağlamazsanız, varsayılan bir başlangıç anahtarı kullanılır.

DİKKAT: Parolaları güvenli bir şekilde korumadığı için parolaları şifrelemek için varsayılan başlangıç anahtarını kullanmayın.

Önemli: Şifreleme anahtarını içeren bir ilk anahtar dosyası belirtirseniz, kuyruk yöneticisinin parolanın şifresini çözebilmesi için kuyruk yöneticisi **INITKEY** özniteliğinde aynı ilk anahtar belirtilmelidir. Kuyruk yöneticisi **INITKEY** özniteliği önceden ayarlandıysa, **runqmcred** komutunu çalıştırırken aynı ilk anahtarı kullanın. Kuyruk yöneticisi **INITKEY** özniteliğine ilişkin ek bilgi için INITKEYbaşlıklı konuya bakın.

Örneğin, `/home/initial.key`dosyasındaki ilk anahtarı kullanarak kimlik doğrulama simgesi anahtar deposu parolalarını şifrelemek için aşağıdaki komutu verin:

```
runqmcred -sf /home/initial.key
```

İstendiğinde, şifrelemek istediğiniz parolayı girin.

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Enter password:
*************
<QM>!2!UnH/9hRXEGA0cenLVSGCW9a0s5A2vHDkTiA7vRv8ogc=!yhlsHFw7MIh48SvaYeTwRQ==
```

Şifrelenmiş parola son satırda çıktı olarak gösterilir. Şifrelenmiş parolayı bir dosyaya kopyalayın ve `qm.ini` dosyasındaki **AuthToken** öğesinin **KeyStorePwdFile** özniteliğinde dosyanın yolunu ekleyin.

Daha fazla bilgi için bkz "Bir kuyruk yöneticisinin kimlik doğrulama belirteçlerini kabul edecek şekilde yapılandırılması" sayfa 349.

Parola şifreleme yoluyla koruma sınırları

IBM MQ , çeşitli yapılandırma dosyalarında saklanan parolalar için AES-128 şifrelemesini destekler. IBM MQ yapılandırmasında parolaları korumak için Gelişmiş Şifreleme Standardı (Advanced Encryption Standard; AES) şifrelemesini kullandığınızda, bu şifrelemenin sağladığı korumanın sınırlarını anlamanız gerekir.

IBM MQ yapılandırma dosyalarında bir parolanın şifrelenmesi, parolanın güvenli ya da korumalı olduğu anlamına gelmez. Yalnızca şifrelenmiş parolaya erişebilen ancak şifreleme anahtarını bilmeyen biri tarafından kolayca kurtarılmasını önler. IBM MQ işlemleri, kullanılacak açık metin parolasını elde etmek için hem şifrelenmiş parolaya hem de şifre çözme anahtarına erişim gerektirir. Bu veri öğelerinin her ikisi de IBM MQ' in erişebileceği bir konumda dosya sisteminde saklanmalıdır. Bir yapılandırma dosyasına yerleştirilen bir parolayı şifreleyen herkesin şifreleme anahtarına da erişmesi gerekir. Bir saldırganın IBM MQile aynı dosya kümesine erişimi varsa, parolaya AES şifrelemesi uygulanarak en düşük düzeyde koruma sağlanır.

Bununla birlikte, şifrelerin yanlışlıkla ifşa edilmesini önlediğinden ve şifre çözme anahtarı da paylaşılmıyorsa yapılandırma dosyalarının paylaşılmasını sağladığından, atıl durumdaki parolaların şifrelenmesi dikkate alınması önemlidir.

Şifre çözme anahtarını içeren dosyanın paylaşılmamasını sağlamanın yanı sıra, dosyanın sistemdeki diğer kullanıcılardan korunduğundan emin olun. IBM MQ yapılanış kütüklerine tüm kullanıcılar erişebilirken, şifre çözme anahtarını içeren kütüğe ilişkin izinleri gereken alt sınırla sınırlayın. IBM MQ ' in işlediği kullanıcı kimliklerine, şifre çözme anahtarını içeren dosyayı okuma erişimi verilmelidir. Ancak, dosyayı bir gruba ya da sistemdeki tüm kullanıcılara okuma erişimi vermek gerekmez.

Veritabanı kimlik doğrulama ayrıntılarının korunması

Veritabanı yöneticisine bağlanmak için kullanıcı adı ve parola kimlik doğrulaması kullanıyorsanız, parolayı `qm.ini` dosyasında düz metinde saklamayı önlemek için bunları MQ XA kimlik bilgileri deposunda saklayabilirsiniz.

Kaynak yöneticisi için XAOpenString ' i güncelle

Kimlik bilgileri deposunu kullanmak için `qm.ini` dosyasında XAOpenString ' i değiştirmeniz gerekir. Dizgi, veritabanı yöneticisine bağlanmak için kullanılır. XAOpenString dizgisinde kullanıcı adı ve parolanın yerine konan yeri belirlemek için değiştirilebilir alanları belirtirsiniz.

- +USER+ alanı, XACredentials deposunda saklanan kullanıcı adı değeriyle değiştirilir.
- +PASSWORD+ alanı, XACredentials deposunda saklanan parola değeriyle değiştirilir.

Aşağıdaki örneklerde, veritabanına bağlanmak için kimlik bilgileri dosyasını kullanmak üzere bir XAOpenString ' in nasıl değiştirileceği gösterilmiştir.

Db2 veritabanına bağlanma

```
XAResourceManager:
  Name=mydb2
  SwitchFile=db2swit
```

```
  XAOpenString=db=mydbname,uid=+USER+,pwd=+PASSWORD+,toc=t
  ThreadOfControl=THREAD
```

Oracle veritabanına bağlanma

```
XAResourceManager:
  Name=myoracle
  SwitchFile=oraswit
  XAOpenString=Oracle_XA+Acc=P/+USER+/+PASSWORD++SesTm=35
            +LogDir=/tmp+threads=true
  ThreadOfControl=THREAD
```

MQ XA kimlik bilgileri deposunda veritabanına ilişkin kimlik bilgileriyle çalışma

`qm.ini` dosyasını değiştirilebilir kimlik bilgisi dizgileriyle güncelledikten sonra, **setmqxacred** komutunu kullanarak kullanıcı adını ve parolayı MQ kimlik bilgileri deposuna eklemeniz gerekir. **setmqxacred** ' yi var olan kimlik bilgilerini değiştirmek, kimlik bilgilerini silmek ya da listelemek için de kullanabilirsiniz. Aşağıdaki örnekler bazı tipik kullanım senaryolarını verir:

Kimlik bilgileri ekleniyor

Aşağıdaki komut, `mqdb2kaynağı` için `QM1` kuyruk yöneticisine ilişkin kullanıcı adını ve parolayı güvenli bir şekilde kaydeder.

```
setmqxacred -m QM1 -x mydb2 -u user1 -p Password2
```

Kimlik bilgileri güncelleniyor

Bir veritabanına bağlanmak için kullanılan kullanıcı adını ve parolayı güncellemek için, **setmqxacred** komutunu yeni kullanıcı adı ve parolayla yeniden verin:

```
setmqxacred -m QM1 -x mydb2 -u user3 -p Password4
```

Değişikliklerin yürürlüğe girmesi için kuyruk yöneticisini yeniden başlatmanız gerekir.

Kimlik bilgileri siliniyor

Aşağıdaki komut kimlik bilgilerini siler:

```
setmqxacred -m QM1 -x mydb2 -d
```

Kimlik bilgileri listeleniyor

Aşağıdaki komut kimlik bilgilerini listeler:

```
setmqxacred -m QM1 -l
```

İlgili başvurular

setmqxacred

güvenlikManaged File Transfer

Kuruluştan hemen sonra ve değişiklik yapılmadan Managed File Transfer , korumalı bir ortamda test ya da değerlendirme amaçlarına uygun bir güvenlik düzeyine sahiptir. Ancak, bir üretim ortamında, dosya aktarma işlemlerini kimlerin başlatabileceğini, aktarılmakta olan dosyaları kimlerin okuyabileceğini ve yazabileceğini ve dosyaların bütünlüğünün nasıl korunacağını uygun şekilde denetlemeyi düşünmeniz gerekir.

İlgili görevler

MFT' ye özgü kaynaklar için grup yetkilerinin kısıtlanması

MFT' ye özgü kaynaklara ilişkin yetkilerin yönetilmesi

“Advanced Message Security ' yi Managed File Transfer ile kullanma” sayfa 648

Bu senaryoda, bir Managed File Transferaracılığıyla gönderilen veriler için ileti gizliliği sağlamak üzere Advanced Message Security ' in nasıl yapılandırılacağı açıklanır.

İlgili başvurular

MFT ' nin dosya sistemlerine erişmesine ilişkin yetkiler

commandPath MFT özelliği

MFT Agent 'ların günlüğü ve durum iletilerini yayınlama yetkisi

MFT içinde saklanan kimlik bilgilerini şifreleme

Managed File Transfer (MFT), iki XML dosyasında saklanan birkaç kullanıcı kimliği ve kimlik bilgisi gerektirir ve bunları **fteObfuscate** komutunu kullanarak gizleyebilirsiniz. IBM MQ 9.2.0' den bu komut, depolanan kimlik bilgilerinin gelişmiş bir şekilde korunmasını sağlar.

Kimlik bilgileri dosyaları

MQMFTCredentials.xml
Bu dosya, aracılara ve koordinasyon ve komut kuyruğu yöneticilerine bağlanmak için kullanıcı kimliğini ve kimlik bilgilerini içerir. Kuyruk yöneticilerine güvenli bağlantılar için anahtar depolarına erişmek üzere kullanılan kimlik bilgileri de aynı dosyada saklanır.

`MQMFTCredentials.xml` dosyasının konumunu tanımlayan özellik değerlerinin ayrıntıları için bkz. “MFT ve IBM MQ bağlantı doğrulaması” sayfa 588 .

ProtocolBridgeCredentials.xml
Bu dosya, protokol sunucularına bağlanmak için kullanılan kullanıcı kimliğini ve kimlik bilgilerini içerir.

fteObfuscate komutunu kullanarak kimlik bilgilerini şifreleme

IBM MQ 9.2.0' den **fteObfuscate** komutu aşağıdaki parametreleri kabul eder:

- **-f** *credentials_file_name* (gerekli)

  Not: Deprecated Bu değiştirge, IBM MQ 9.2.0' den kullanımdan kaldırılan **-credentialsFile** değiştirgesinin yerine geçer.
- **-sp** *koruma_kipi*
- **-sf** *credentials_key_file*
- **-o** *çıkış_dosyası_adı*

Parametrelerin ayrıntıları için bkz. **fteObfuscate** .

Koruma kipini ya da kimlik bilgileri anahtar dosyasını belirtmezseniz, komut varsayılan koruma kipini kullanır ve en son algoritmayı kullanır, ancak kimlik bilgilerini şifrelemek için sabit bir anahtarla birlikte kullanılır.

0koruma kipini belirtirseniz ve bir kimlik bilgileri anahtar dosyası belirtmezseniz, komut ürünün önceki yayınlarında olduğu gibi çalışır. Konsolda, kullanımdan kaldırılan korumanın kullanımını belirten bir uyarı iletisi alırsınız.

0koruma kipini belirtirseniz ve bir kimlik bilgileri anahtar dosyası belirtirseniz, konsolda 0koruma kipini kullanırken anahtar dosyasını belirtmenin geçerli olmadığını belirten bir hata çıkışı alırsınız.

1koruma kipini belirtirseniz ve bir kimlik bilgileri anahtar dosyası belirtmezseniz, komut en son algoritmayı kullanır, ancak kimlik bilgilerini şifrelemek için sabit bir anahtarla birlikte kullanılır.

1koruma kipini belirtirseniz ve bir kimlik bilgileri anahtar dosyası belirtirseniz, komut kimlik bilgilerini en son algoritmayla şifreler.

1koruma kipini belirtirseniz ya da koruma kipini belirtmezseniz ve var olmayan bir kimlik bilgileri anahtar dosyası belirtirseniz, konsolda dosyanın var olmadığını belirten bir hata görüntülenir.

1koruma kipini belirtirseniz ya da koruma kipini belirtmezseniz ve okunabilir olmayan bir kimlik bilgileri anahtar dosyası belirtirseniz, konsolda dosyanın okunabilir olmadığını belirten bir hata görüntülenir.

V 9.3.0 2koruma kipini belirtirseniz ve bir kimlik bilgileri anahtar dosyası belirtmezseniz, komut en son algoritmayı ve şifrelenecek sabit bir anahtarı kullanarak kimlik bilgilerini şifrelemek için koruma kipi 2 'yi kullanır.

V 9.3.0 2koruma kipini belirtirseniz ve bir kimlik bilgileri anahtar dosyası belirtirseniz, komut en son algoritmayı kullanarak kimlik bilgilerini şifrelemek için koruma kipi 2 'yi ve şifrelemek için kullanıcı tarafından belirtilen bir anahtarı kullanır.

V 9.3.0 2koruma kipini belirtirseniz ya da koruma kipini belirtmezseniz ve var olmayan bir kimlik bilgileri anahtar dosyası belirtirseniz, konsolda dosyanın var olmadığını belirten bir hata görüntülenir.

V 9.3.0 2koruma kipini belirtirseniz ya da koruma kipini belirtmezseniz ve okunabilir olmayan bir kimlik bilgileri anahtar dosyası belirtirseniz, konsolda dosyanın okunabilir olmadığını belirten bir hata görüntülenir.

Kimlik bilgilerinin şifresi çözülüyor

İlk anahtar dosyasının yolunu çeşitli yerlerde belirtebilirsiniz. Varsayılan anahtar dışında bir ilk anahtar kullanılarak şifrelenen kimlik bilgilerinin şifresini çözmek için, ilk anahtarı içeren dosyanın adı, bu öncelik sırasına göre MFT ' e aşağıdaki yollardan biriyle sağlanmalıdır:

1. Java sistem özelliğini kullanarak, örneğin:

   ```
   -Dcom.ibm.wmqfte.cred.keyfile=/usr/hime/credkeyfile.key
   ```

   Not:

   - V 9.3.1 IBM MQ 9.3.1' den önce, bu Java sistem özelliğinin adı ürün kodunda `com.ibm.wqmfte.cred.keyfile`olarak yanlış yazılıyordu. IBM MQ 9.3.1' den özellik adının yazımı `com.ibm.wmqfte.cred.keyfile`olarak düzeltilir. Managed File Transfer , bir kullanıcının kimlik bilgilerini şifrelemek ve kimlik bilgilerinin şifresini çözmek için kullanılması gereken ilk anahtarı içeren bir dosya belirtip belirtmediğini denetlerken Java sistem özelliğinin her iki sürümünü de kullanır. Bu, eski yanlış yazılmış adla geriye doğru uyumluluğu korurken özellik adının doğru yazımını kullanmanızı sağlar. Her iki Java sistem özelliği de ayarlanırsa, doğru yazılmış `com.ibm.wmqfte.cred.keyfile` özelliğinin değeri kullanılır.
   - IBM MQ 9.3.1' den önce `com.ibm.wqmfte.cred.keyfile`özelliğini kullanın.

   .
2. Bir aracı, komut, eşgüdüm ya da günlük kaydedici özellikler dosyasında bir özellik ayarlayarak. Özellikler dosyasının adı ve içinde ayarlanması gereken özellik aşağıdaki tabloda gösterilir:

Özellik Dosyası	**Özellik adı**
agent.properties	agentCredentialsKeyFile
command.properties	commandCredentialsKeyFile
coordination.properties	coordinationCredentialsKeyFile
logger.properties	loggerCredentialsKeyFile

3. installation.properties dosyasında.

   Tek tek özellikler dosyalarına özellikler eklemek yerine, aracı, kaydedici ve komutların aynı özelliği kullanabilmesi için **`commonCredentialsKeyFile`** özelliğini var olan ortak `installation.properties` dosyasına ekleyebilirsiniz.

Birden çok konumda çeşitli **`CredentialsKeyFile`** özelliklerini tanımladıysanız:

- Aracı ve kaydedici için kullanılan kimlik bilgileri anahtar dosyasının yolu, o aracı ya da kaydedici için `output0.log` dosyasına kaydedilir.

- Komutlar için kullanılan kimlik bilgileri anahtar dosyasının yolu konsolda görüntülenir.

Java Sistem özelliği **com.ibm.wmqfte.cred.keyfile** , diğer tüm özellikleri geçersiz kılar. Sistem özelliği ayarlanmazsa, aracı `agent.properties` dosyasını ve ardından ilk anahtar dosyası için `installation.properties` dosyasını arar.

İlk anahtar dosyası hala bulunamazsa ve **fteObfuscate** komutunda koruma kipini `1`olarak ayarladıysanız, aracı `output0.log` dosyasına bir hata iletisi kaydeder.

fteObfuscate komutunda koruma kipini `0` olarak ayarladıysanız, kullanımdan kaldırmayı belirten bir uyarı iletisi günlüğe kaydedilir.

Kaydedici ve komutlar, ilk anahtar dosyasını bulmak için aynı adımları izler.

İletişim Kuralı Köprüsü ve Connect:Direct Köprüsü

Protocol Bridge, FTP, SFTP ve FTPS sunucularına bağlanmak için `ProtocolBridgeProperties.xml`adlı bir özellikler dosyası kullanır. Bu özellikler dosyası, bu sunuculara bağlanmak için gereken bağlantı özniteliklerini içerir.

`ProtocolBridgeProperties.xml` dosyasındaki **credentialsFile** ya da **credentialsKeyFile** özniteliklerinin değerini değiştirirseniz köprü aracısının yeniden başlatılması gerekir.

Özniteliklerden biri **credentialsFile**ve değer, bu sunuculara bağlanmak için gerekli olan UID ya da PWD ya da Anahtar içeren bir XML dosyasının yolunu içerir. Özniteliğin varsayılan değeri *ProtocolBridgeCredentials.xml* ' dir ve dosya `MQMFTCredentials.xml` dosyası gibi ana dizininizdedir.

```
<tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />
```

*MQMFTCredentails.xml*gibi, *ProtocolBridgeCredentials.xml* ' i **fteObfuscate** komutuyla şifreleyebilirsiniz. Şifre çözme amacıyla, aşağıdaki metinde gösterildiği gibi **credentialsKeyFile** ek öğesini kullanarak bir kimlik bilgileri anahtar dosyasının gerekli yolunu belirtebilirsiniz. Yol ortam değişkenlerini içerebilir.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

Not: `installation.properties`içinde ya da **com.ibm.wqmfte.cred.keyfile**sistem özelliği aracılığıyla **agentCredentialsKeyFile** aracı özelliği, **commonCredentialsKeyFile** özelliği için bir değer belirtilmesi, **credentialsKeyFile** özniteliği için belirtilen değer üzerinde herhangi bir etkiye sahip değildir.

Benzer şekilde Connect:Direct Bridge, Connect:Direct sunucusuna bağlanmak için *ConnectDirectNodeProperties.xml* dosyasını kullanır. XML dosyası, kimlik bilgileri XML dosyasının yolunu tanımlayan bir öznitelikle birlikte gerekli bağlantı bilgilerini içerir. Bu kimlik bilgileri XML dosyası, UID ya da PWD bilgilerini ve Connect:Direct sunucusuna bağlanmak için gereken ek bilgileri içerir.

```
<tns:credentialsFile path="$HOME/ ConnectDirectCredentials.xml" />
```

ProtocolBridgeCredentials.xml dosyası gibi, *ConnectDirectCredentials.xml* dosyasını **fteObfuscate** komutuyla şifreleyebilirsiniz. Şifre çözme amacıyla, aşağıdaki metinde gösterildiği gibi **credentialsKeyFile** ek öğesini kullanarak bir kimlik bilgileri anahtar dosyasının gerekli yolunu belirtebilirsiniz. Yol ortam değişkenlerini içerebilir.

```
<tns:credentialsKeyFile path="$HOME/CredKey.key"/>
```

Not: `installation.properties`içinde ya da **com.ibm.wqmfte.cred.keyfile** sistem özelliği aracılığıyla **agentCredentialsKeyFile** aracı özelliği, **commonCredentialsKeyFile** özelliği için bir değer belirtilmesi, **credentialsKeyFile** özniteliği için belirtilen değer üzerinde herhangi bir etkiye sahip değildir.

credentialsKeyFile öğesini, *ProtocolBridgeProperties.xml* dosyasında **credentialsFile** öğesini belirtmeden belirtebilirsiniz.

credentialsFile öğesini belirtmezseniz, protokol köprüsü aracısı tarafından varsayılan kimlik bilgileri dosyası *ProtocolBridgeCredentials.xml* kullanılır ve **credentialsKeyFile** özniteliğinde belirtilen anahtar dosyasının değeri kimlik bilgileri dosyasının şifresini çözmek için kullanılır.

Benzer şekilde, *ConnectDirectNodeProperties.xml* dosyasında **credentialsFile** öğesini belirtmeden **credentialsKeyFile** öğesini belirtebilirsiniz.

credentialsFile öğesini belirtmezseniz, varsayılan kimlik bilgileri dosyası *ConnectDirectCredentials.xml* Connect:Direct köprüsü tarafından kullanılır ve **credentialsKeyFile** özniteliğinde belirtilen anahtar dosyasının değeri kimlik bilgileri dosyasının şifresini çözmek için kullanılır.

z/OS üzerindeki veri kümesindeki anahtarı kullanma

z/OS

z/OSüzerinde, **MQMFTCredentials** değerini belirtebilir ve bir PDSE kullanarak kimlik bilgileri anahtar dosyasını sağlayabilirsiniz. Bkz. “z/OS üzerinde MQMFTCredentials.xml ' ın yapılandırılması” sayfa 590.

İlgili başvurular

Hangi MFT komutunun hangi kuyruk yöneticisine bağlandığı

MFT kimlik bilgileri dosya biçimi

fteObfuscate (hassas verileri şifreleyin)

MFT ve IBM MQ bağlantı doğrulaması

Bağlantı kimlik doğrulaması, bir kuyruk yöneticisinin, sağlanan bir kullanıcı kimliği ve parola kullanarak uygulamaların kimliğini doğrulayabilmesi için yapılandırılmasına olanak sağlar. İlişkili kuyruk yöneticisinde güvenlik etkinleştirildiyse ve kimlik bilgileri ayrıntıları (kullanıcı kimliği ve parola) gerektiriyorsa, bir kuyruk yöneticisiyle başarılı bir bağlantı kurulmadan önce bağlantı kimlik doğrulama özelliği etkinleştirilmelidir. Bağlantı kimlik doğrulaması uyumluluk kipinde ya da MQCSP kimlik doğrulama kipinde çalıştırılabilir.

Kimlik bilgisi ayrıntılarının sağlanmasıyla ilgili yöntemler

Birçok Managed File Transfer komutu, kimlik bilgileri ayrıntılarını sağlamak için aşağıdaki yöntemleri destekler:

Komut satırı bağımsız değişkenleri tarafından sağlanan ayrıntılar.
Kimlik bilgileri ayrıntıları, **-mquserid** ve **-mqpassword** parametreleri kullanılarak belirtilebilir. **-mqpassword** sağlanmazsa, kullanıcıdan girişin görüntülenmediği parolayı sorulur.

Bir kimlik bilgileri dosyasından sağlanan ayrıntılar: MQMFTCredentials.xml.
Kimlik bilgileri ayrıntıları, `MQMFTCredentials.xml` dosyasında açık metin ya da gizlenmiş metin olarak önceden tanımlanabilir.

IBM MQ for Multiplatforms üzerinde MQMFTCredentials.xml dosyası ayarlama hakkında bilgi için bkz. “MQMFTCredentials.xml ' in Multiplatforms üzerinde yapılandırılması” sayfa 589.

IBM MQ for z/OS üzerinde MQMFTCredentials.xml dosyası ayarlama hakkında bilgi için bkz. “z/OS üzerinde MQMFTCredentials.xml ' ın yapılandırılması” sayfa 590.

Öncelik

Kimlik bilgileri ayrıntılarının belirlenmesinin önceliği:

1. Komut satırı bağımsız değişkeni.
2. `MQMFTCredentials.xml` dizini, ilişkili kuyruk yöneticisi ve komutu çalıştıran kullanıcı tarafından.
3. İlişkili kuyruk yöneticisine göre `MQMFTCredentials.xml` dizini.
4. Önceki IBM MQya da IBM WebSphere MQ yayınlarıyla uyumluluğu sağlamak için kimlik bilgisi ayrıntılarının sağlanmadığı varsayılan geriye doğru uyumluluk kipi

Notlar:

- **fteStartAgent** ve **fteStartLogger** komutları, **-mquserid**ya da **-mqpassword**komut satırı bağımsız değişkenini desteklemez ve kimlik bilgileri ayrıntıları yalnızca `MQMFTCredentials.xml` dosyasıyla belirtilebilir.
- z/OS

  z/OS' da, kullanıcı parolasının küçük harfleri olsa bile, parola büyük harfli olmalıdır. Örneğin, kullanıcının parolası "password" ise, "PASSWORD" olarak girilmelidir.

İlgili başvurular

Hangi MFT komutunun hangi kuyruk yöneticisine bağlandığı

MFT kimlik bilgileri dosya biçimi

MQMFTCredentials.xml ' in Multiplatforms üzerinde yapılandırılması

Managed File Transfer (MFT) güvenlik etkinleştirilmiş olarak yapılandırıldıysa, bağlantı kimlik doğrulaması, kullanıcı kimliği ve parola kimlik bilgilerini sağlamak için bir kuyruk yöneticisine bağlanan tüm MFT komutlarını gerektirir. Benzer şekilde, MFT kaydedicilerinin bir veritabanına bağlanırken kullanıcı kimliği ve parola belirtmeleri gerekebilir. Bu kimlik bilgileri MFT kimlik bilgileri dosyasında saklanabilir.

Bu görev hakkında

`MQMFTCredentials.xml` dosyasındaki öğeler, `MQMFTCredentials.xsd` şemasına uygun olmalıdır. `MQMFTCredentials.xml`biçimi hakkında bilgi için bkz. MFT kimlik bilgileri dosyası biçimi.

`MQ_INSTALLATION_PATH/mqft/samples/credentials` dizininde örnek bir kimlik bilgileri dosyası bulabilirsiniz.

Eşgüdüm kuyruğu yöneticisi için bir MFT kimlik bilgileri dosyası, komut kuyruğu yöneticisi için bir kimlik bilgileri, her aracı için bir kimlik bilgileri dosyası ve her günlük kaydedici için bir kimlik bilgileri dosyası olabilir. Alternatif olarak, topolojinizdeki her şey tarafından kullanılan bir dosyanız olabilir.

MFT kimlik bilgileri dosyasının varsayılan konumu şöyledir:

Linux AIX **AIX and Linux**

$HOME

Windows **Windows**

%USERPROFILE% veya %HOMEDRIVE%%HOMEPATH%

Kimlik bilgileri dosyası farklı bir konumda saklanıyorsa, komutların arayacağı yeri belirtmek için aşağıdaki özellikleri kullanabilirsiniz:

Çizelge 98. : Çeşitli komutlar için MQMFTCredentials.xml dosyasının yerini tanımlayan özellikler.

Komut tipi	Özellik Dosyası	Özellik adı
Koordinasyon kuyruk yöneticisine bağlanan komut	coordination.properties	coordinationQMgrAuthenticationCredentialsDosyası
Komut kuyruğu yöneticisine bağlanan komut	connection.properties	connectionQMgrAuthenticationCredentialsDosyası
Bir aracı işlemine bağlanan komut	agent.properties	agentQMgrAuthenticationCredentialsDosyası
Günlüğe kaydedici işlemine bağlanan komut	logger.properties	loggerQMgrAuthenticationCredentialsDosyası

Çizelge 99. : Aracılar ve kaydedici işlemleri için MQMFTCredentials.xml dosyasının yerini tanımlayan özellikler.		
Komut tipi	**Özellik Dosyası**	**Özellik adı**
MFT aracılar	agent.properties	agentQMgrAuthenticationCredentialsDosyası
MFT Günlüğe kaydediciler	logger.properties	loggerQMgrAuthenticationCredentialsDosyası

Hangi komutların ve işlemlerin hangi kuyruk yöneticisine bağlandığı hakkında ayrıntılı bilgi için Hangi MFT komutlarının ve işlemlerinin hangi kuyruk yöneticisine bağlandığıkonusuna bakın.

Tek tek özellikler dosyalarına özellikler eklemek yerine, **commonCredentialsKeyFile** özelliğini var olan ortak `installation.properties` dosyasına ekleyebilirsiniz; böylece aracı, kaydedici ve komutlar aynı özelliği kullanabilir.

Kimlik bilgileri dosyası kullanıcı kimliği ve parola bilgilerini içerdiğinden, dosyaya yetkisiz erişimi önlemek için özel izinler gerekir:

Linux AIX **AIX and Linux**

```
chown <agent owner userid>
chmod 600
```

Windows **Windows**

Üstten edinmenin etkinleştirilmediğinden emin olun ve kimlik bilgileri dosyasını kullanacak aracı ya da kaydediciyi çalıştıryanlar dışında tüm kullanıcı kimliklerini kaldırın.

IBM MQ Explorer Managed File Transfer eklentisindeki bir MFT eşgüdüm kuyruk yöneticisine bağlanmak için kullanılan kimlik bilgileri ayrıntıları, yapılandırmanın tipine bağlıdır:

Genel (yerel diskteki konfigürasyon)
Genel bir yapılandırma, koordinasyon ve komut özelliklerinde belirtilen kimlik bilgileri dosyasını kullanır.

Yerel (IBM MQ Exploreriçinde tanımlanır):
Yerel bir yapılandırma, IBM MQ Exploreriçindeki ilişkili kuyruk yöneticisinin bağlantı ayrıntılarının özelliklerini kullanır.

İlgili görevler

“MFT için bağlantı kimlik doğrulamasını etkinleştirme” sayfa 592
Bir koordinasyon kuyruğu yöneticisiyle ya da komut kuyruğu yöneticisiyle bağlantı kuran IBM MQ Explorer MFT Plugin 'in bağlantı kimlik doğrulaması ve bir koordinasyon kuyruğu yöneticisiyle ya da komut kuyruğu yöneticisiyle bağlantı kuran bir Managed File Transfer aracısı için bağlantı kimlik doğrulaması uyumluluk kipinde ya da MQCSP kimlik doğrulama kipinde çalıştırılabilir.

IBM MQ Dosya Aktarımı Yapısının Yaratılması

İlgili başvurular

MFT kimlik bilgileri dosya biçimi

MFT içinde saklanan kimlik bilgilerini şifreleme

fteObfuscate: hassas verileri şifrele

z/OS z/OS üzerinde MQMFTCredentials.xml 'ın yapılandırılması

Managed File Transfer (MFT) güvenlik etkinleştirilmiş olarak yapılandırıldıysa, bağlantı kimlik doğrulaması, kullanıcı kimliği ve parola kimlik bilgilerini sağlamak için tüm MFT aracılarını ve bir kuyruk yöneticisine bağlanan komutları gerektirir.

Benzer şekilde, MFT kaydedicilerinin bir veritabanına bağlanırken kullanıcı kimliği ve parola belirtmeleri gerekebilir.

Bu kimlik bilgileri MFT kimlik bilgileri dosyasında saklanabilir. Kimlik bilgileri dosyalarının isteğe bağlı olduğunu, ancak ortamı özelleştirmeden önce gerek duyduğunuz dosyayı ya da dosyaları tanımlamanın daha kolay olduğunu unutmayın.

Buna ek olarak, kimlik bilgileri dosyalarınız varsa, daha az uyarı iletisi alırsınız. Uyarı iletileri, MFT ' in kuyruk yöneticisi güvenliğinin kapalı olduğunu ve bu nedenle kimlik doğrulama ayrıntılarını sağlamadığınızı belirtmiştir.

`MQ_INSTALLATION_PATH/mqft/samples/credentials` dizininde örnek bir kimlik bilgileri dosyası bulabilirsiniz.

Aşağıda bir `MQMFTCredentials.xml`dosyası örneği verilmiştir:

```
<?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>
```

ADMIN kullanıcı kimlikli bir işin kuyruk yöneticisi MQPH ' ye bağlanması gerektiğinde, *JOHNDOEH* kullanıcı kimliğini geçirir ve *cXXXX*parolasını kullanır.

İş başka bir kullanıcı kimliği tarafından çalıştırılırsa ve MQPH ile bağlantı kurarsa, bu iş *NONEH* kullanıcı kimliğini ve *yXXXX*parolasını geçirir.

`MQMFTCredentials.xml` dosyasının varsayılan konumu, kullanıcının z/OS UNIX System Services (USS) üzerindeki ana dizinidir. Ayrıca, dosyayı USS ' de farklı bir yerde ya da bölümlenmiş bir veri kümesindeki bir üyede saklayabilirsiniz.

Kimlik bilgileri dosyası farklı bir konumda saklandıysa, komutların bu dosyayı nerede arayacağını belirtmek için aşağıdaki özellikleri kullanabilirsiniz:

Çizelge 100. : Çeşitli komutlar için MQMFTCredentials.xml dosyasının yerini tanımlayan özellikler.

Komut tipi	**Özellik Dosyası**	**Özellik adı**
Koordinasyon kuyruk yöneticisine bağlanan komut	coordination.properties	coordinationQMgrAuthenticationCredentialsDosyası
Komut kuyruğu yöneticisine bağlanan komut	connection.properties	connectionQMgrAuthenticationCredentialsDosyası
Bir aracı işlemine bağlanan komut	agent.properties	agentQMgrAuthenticationCredentialsDosyası
Bir kaydedici işlemine bağlanan komut	logger.properties	loggerQMgrAuthenticationCredentialsDosyası

Çizelge 101. : Aracılar ve kaydedici işlemleri için MQMFTCredentials.xml dosyasının yerini tanımlayan özellikler.

Komut tipi	**Özellik Dosyası**	**Özellik adı**
MFT aracılar	agent.properties	agentQMgrAuthenticationCredentialsDosyası
MFT Günlüğe kaydediciler	logger.properties	loggerQMgrAuthenticationCredentialsDosyası

Hangi komutların ve işlemlerin hangi kuyruk yöneticisine bağlanacağı hakkında ayrıntılı bilgi için Hangi MFT komutlarının ve işlemlerinin hangi kuyruk yöneticisine bağlandığıkonusuna bakın.

Bölümlenmiş bir veri kümesi içinde kimlik bilgileri dosyası oluşturmak için aşağıdaki adımları gerçekleştirin:

- VB ve mantıksal kayıt uzunluğu (Lrecl) 200 biçimindeki bir PDSE yaratın.
- Veri kümesi içinde bir üye oluşturun, veri kümesi ve üyeyi not edin ve üyeye aşağıdaki kodu ekleyin:

```
<?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>
```

Kimlik bilgileri dosyasını bir güvenlik ürünü kullanarak koruyabilirsiniz; örneğin, RACF, ancak Managed File Transfer komutlarını çalıştıran kullanıcı kimlikleri ve aracı ve günlük kaydedici işlemlerini yönetme, bu dosyaya okuma erişimi gerekir.

Bu dosyadaki bilgileri BFGCROBS üyesindeki JCL ' yi kullanarak gizleyebilirsiniz. Bu işlem dosyayı alır ve IBM MQ kullanıcı kimliğini ve parolasını şifreler. Örneğin, BFGCROBS üyesi satırı alır

```
<tns:qmgr name="MQPI" user="JOHNDOE2" mqUserId="JOHNDOE1" mqPassword="yXXXX" />
```

ve yaratır

```
<tns:qmgr mqPasswordCipher="e977c61e9b9c363c" mqUserIdCipher="c394c5887867157c"
name="MQPI" user="JOHNDOE2"/>
```

Kullanıcı kimliğini IBM MQ kullanıcı kimliği eşlemesine alıkoymak istiyorsanız, dosyaya açıklamalar ekleyebilirsiniz. Örnek:

```
<!--      name="MQPI" user="ADMIN"    mqUserId="JOHNDOE1 -->
```

Bu yorumlar, gizleme süreci tarafından değiştirilmez.

İçeriğin gizlendiğini, güçlü bir şekilde şifrelenmediğini unutmayın. Dosyaya erişebilecek kullanıcı kimliklerini sınırlandırmanız gerekir.

İlgili görevler

“MQMFTCredentials.xml ' in Multiplatforms üzerinde yapılandırılması” sayfa 589

Managed File Transfer (MFT) güvenlik etkinleştirilmiş olarak yapılandırıldıysa, bağlantı kimlik doğrulaması, kullanıcı kimliği ve parola kimlik bilgilerini sağlamak için bir kuyruk yöneticisine bağlanan tüm MFT komutlarını gerektirir. Benzer şekilde, MFT kaydedicilerinin bir veritabanına bağlanırken kullanıcı kimliği ve parola belirtmeleri gerekebilir. Bu kimlik bilgileri MFT kimlik bilgileri dosyasında saklanabilir.

MFT için bağlantı kimlik doğrulamasını etkinleştirme

Bir koordinasyon kuyruğu yöneticisiyle ya da komut kuyruğu yöneticisiyle bağlantı kuran IBM MQ Explorer MFT Plugin 'in bağlantı kimlik doğrulaması ve bir koordinasyon kuyruğu yöneticisiyle ya da komut kuyruğu yöneticisiyle bağlantı kuran bir Managed File Transfer aracısı için bağlantı kimlik doğrulaması uyumluluk kipinde ya da MQCSP kimlik doğrulama kipinde çalıştırılabilir.

Bu görev hakkında

IBM MQ 9.2.0' den önce uyumluluk kipi, bağlantı kimlik doğrulaması için varsayılan ayardır. Ancak, varsayılan uyumluluk kipini geçersiz kılabilir ve MQCSP kimlik doğrulama kipini etkinleştirebilirsiniz.

IBM MQ 9.2.0' den MQCSP kimlik doğrulama kipi varsayılan kiptir.

IBM MQ Explorer Managed File Transfer eklentisi ya da CLIENT iletimini kullanarak bir kuyruk yöneticisine bağlanan Managed File Transfer aracıları için 12 karakterden uzun parolalar yalnızca MQCSP kimlik doğrulama kipi için desteklenir. Uyumluluk kipini kullanırken 12 karakterden uzun bir parola belirtirseniz, bir hata ortaya çıkar ve aracı kuyruk yöneticisiyle kimlik doğrulaması yapmaz. Tanılama iletileri: BFGAG0001 - BFGAG9999içindeki BFGAG0187E iletisine bakın.

Yordam

- IBM MQ Explorericinde bir eşgüdüm kuyruğu yöneticisi ya da komut kuyruğu yöneticisi için bağlantı kimlik doğrulama kipini seçmek üzere aşağıdaki adımları tamamlayın:
  a) Bağlanmak istediğiniz kuyruk yöneticisini seçin.
  b) Sağ tıklatın ve beliren menüden **Bağlantı Ayrıntıları-> Özellikler** seçeneklerini belirleyin.
  c) **Kullanıcı kimliği** sekmesini tıklatın.
  d) Kullanmak istediğiniz bağlantı kimlik doğrulama kipine ilişkin onay kutusunun seçili olduğundan emin olun:
     – IBM MQ 9.1.0' dan varsayılan olarak **Kullanıcı kimliği uyumluluk kipi** onay kutusunun işareti kaldırılır. Bu, **Kullanıcı kimliğini etkinleştir** onay kutusu seçilirse, IBM MQ Explorer ' in kuyruk yöneticisine bağlanırken MQCSP kimlik doğrulamasını kullanacağı anlamına gelir. IBM MQ Explorer ' in MQCSP kimlik doğrulaması yerine uyumluluk kipini kullanarak kuyruk yöneticisine bağlanması gerekiyorsa, **Kullanıcı kimliğini etkinleştir** ve **Kullanıcı kimliği uyumluluk kipi** onay kutularının seçili olmasına dikkat edin.
     – IBM MQ 9.1.0' den önce, varsayılan olarak **User identification compatibility mode** (Kullanıcı kimliği uyumluluk kipi) onay kutusu seçilidir. Bu, **Kullanıcı kimliğini etkinleştir** onay kutusu seçiliyse, IBM MQ Explorer ' in kuyruk yöneticisine bağlanırken uyumluluk kipini kullanacağı anlamına gelir. IBM MQ Explorer ' in MQCSP kimlik doğrulamasını kullanarak kuyruk yöneticisine bağlanması gerekiyorsa, **Kullanıcı kimliğini etkinleştir** onay kutusunun seçili olmasına ve **Kullanıcı kimliği uyumluluk kipi** onay kutusunun seçili olmamasına dikkat edin.
- `MQMFTCredentials.xml` dosyasını kullanarak bir Managed File Transfer aracısı için MQCSP kimlik doğrulama kipini etkinleştirmek ya da devre dışı bırakmak için ilgili kullanıcıya ilişkin `MQMFTCredentials.xml` dosyasına **`useMQCSPAuthentication`** parametresini ekleyin.

  `useMQCSPAuthentication` değiştirgesi aşağıdaki değerleri içerir:

  doğru
  MQCSP kimlik doğrulama kipi, kuyruk yöneticisiyle kullanıcının kimliğini doğrulamak için kullanılır.

  IBM MQ 9.2.0' dan `true` varsayılan değerdir. **`useMQCSPAuthentication`** değiştirgesi belirtilmezse, varsayılan olarak `true` değerine ayarlanır ve kuyruk yöneticisiyle kullanıcının kimliğini doğrulamak için MQCSP kimlik doğrulama kipi kullanılır.

  yanlış
  Kuyruk yöneticisiyle kullanıcının kimliğini doğrulamak için uyumluluk kipi kullanılır.

  IBM MQ 9.2.0öncesinde, **`useMQCSPAuthentication`** parametresi belirtilmezse, varsayılan olarak `false` değerine ayarlanır ve kullanıcının kuyruk yöneticisiyle kimliğini doğrulamak için uyumluluk kipi kullanılır.

  Aşağıdaki örnek, `MQMFTCredentials.xml` dosyasında **`useMQCSPAuthentication`** parametresinin nasıl ayarlanacağını göstermektedir:

```
<tns:qmgr name="CoordQueueMgr" user="ernest" mqUserId="ernest"
    mqPassword="AveryL0ngPassw0rd2135" useMQCSPAuthentication="true"/>
```

İlgili kavramlar

“MQCSP parola koruması” sayfa 31
MQCSP yapısında belirtilen kimlik doğrulama kimlik bilgileri, IBM MQ MQCSP parola koruma özelliği kullanılarak korunabilir ya da TLS şifrelemesi kullanılarak şifrelenebilir.

İlgili başvurular

“MFT ve IBM MQ bağlantı doğrulaması” sayfa 588
Bağlantı kimlik doğrulaması, bir kuyruk yöneticisinin, sağlanan bir kullanıcı kimliği ve parola kullanarak uygulamaların kimliğini doğrulayabilmesi için yapılandırılmasına olanak sağlar. İlişkili kuyruk yöneticisinde güvenlik etkinleştirildiyse ve kimlik bilgileri ayrıntıları (kullanıcı kimliği ve parola) gerektiriyorsa, bir kuyruk yöneticisiyle başarılı bir bağlantı kurulmadan önce bağlantı kimlik doğrulama özelliği etkinleştirilmelidir. Bağlantı kimlik doğrulaması uyumluluk kipinde ya da MQCSP kimlik doğrulama kipinde çalıştırılabilir.

MFT kimlik bilgileri dosya biçimi

MFT sandboxes

Dosya sisteminin, aracının aktarımın bir parçası olarak erişebileceği alanını sınırlayabilirsiniz. Aracının sınırlı olduğu alana kum havuzu adı verilir. Aracıya ya da aktarım isteyen kullanıcıya kısıtlamalar uygulayabilirsiniz.

Aracı bir protokol köprüsü aracısı ya da bir Connect:Direct köprü aracısı olduğunda kum havuzları desteklenmez. IBM MQ kuyruklarına/kuyruklarından aktarılması gereken aracılar için aracı çalışma yeri kullanımını kullanamazsınız.

İlgili başvurular

“MFT Agent kum havuzlarıyla çalışma” sayfa 594
Managed File Transfer' e ek bir güvenlik düzeyi eklemek için, bir aracının erişebileceği dosya sistemi alanını sınırlayabilirsiniz.

“MFT kullanıcı kum havuzlarıyla çalışma” sayfa 595
Dosyaların aktarılabileceği dosya sistemi alanını, aktarma isteğinde bulunan MQMD kullanıcı adına dayalı olarak sınırlandırabilirsiniz.

MFT Agent kum havuzlarıyla çalışma

Managed File Transfer' e ek bir güvenlik düzeyi eklemek için, bir aracının erişebileceği dosya sistemi alanını sınırlayabilirsiniz.

IBM MQ kuyruklarına ya da kuyruklarından aktarılan aracılar için aracı çalışma yeri kullanımını kullanamazsınız. Kum havuzu kullanımı olan IBM MQ kuyruklarına erişimin kısıtlanması, kum havuzu kullanımı gereksinimleri için önerilen çözüm olan kullanıcı kum havuzu kullanımı kullanılarak gerçekleştirilebilir. Kullanıcı kum havuzu kullanımı hakkında daha fazla bilgi için bkz. “MFT kullanıcı kum havuzlarıyla çalışma” sayfa 595

Aracı çalışma yeri kullanımını etkinleştirmek için, kısıtlamak istediğiniz aracının `agent.properties` dosyasına aşağıdaki özelliği ekleyin:

```
sandboxRoot=[!]restricted_directory_nameseparator...separator[!]restricted_directory_name
```

Burada:

- *restricted_directory_name* , izin verilecek ya da reddedilecek bir dizin yoludur.
- `!` isteğe bağlıdır ve *restricted_directory_name* için aşağıdaki değerin reddedildiğini (dışlandığını) belirtir. `!` belirtilmezse *restricted_directory_name* , izin verilen (içerilen) bir yoldur.
- *separator* , platforma özgü ayırıcıdır.

Örneğin, AGENT1 'in sahip olduğu erişimi yalnızca `/tmp` diziniyle sınırlamak istiyor, ancak `private` alt dizinine erişilmesine izin vermiyorsanız, özelliği AGENT1: `sandboxRoot=/tmp:!/tmp/private`' e ait `agent.properties` dosyasında şu şekilde ayarlayın.

sandboxRoot özelliği, Gelişmiş aracı özellikleriikısmında açıklanır.

Hem aracı hem de kullanıcı kum havuzu kullanımı, protokol köprüsü aracılarında ya da Connect:Direct köprü aracılarında desteklenmez.

AIX, Linux, and Windows platformlarında bir kum havuzunda çalışma

ALW AIX, Linux, and Windows platformlarında, çalışma yeri kullanımı bir Managed File Transfer Agent ' in hangi dizinlerden okuyabileceğini ve hangi dizinlere yazabileceğini sınırlar. Kum havuzu kullanımı etkinleştirildiğinde, Managed File Transfer Agent izin verildiği şekilde belirtilen dizinleri ve alt dizinler sandboxRootreddedildiği gibi belirtilmedikçe, belirtilen dizinlerin içerdiği tüm alt dizinleri okuyabilir ve bu dizinlere yazabilir. Managed File Transfer kum havuzu kullanımı, işletim sistemi

güvenliğinden daha öncelikli değildir. Managed File Transfer Agent ' i başlatan kullanıcının, dizinden okuyabilmesi ya da dizine yazabilmesi için herhangi bir dizine uygun işletim sistemi düzeyinde erişimi olmalıdır. Bağlantı verilen dizin belirtilen sandboxRoot dizinlerinin (ve alt dizinlerin) dışındaysa, bir dizine yönelik simgesel bir bağlantı izlenmez.

z/OS üzerinde bir kum havuzunda çalışma

z/OS z/OSüzerinde, çalışma yeri kullanımı, Managed File Transfer Agent ' in okuyabileceği ve yazabileceği veri kümesi adı niteleyicilerini sınırlar. Managed File Transfer Agent ' i başlatan kullanıcının, ilgili veri kümeleri için doğru işletim sistemi yetkileri olmalıdır. Bir sandboxRoot veri kümesi adı niteleyicisi değerini çift tırnak işareti içine aldıysanız, değer normal z/OS kuralını izler ve tam olarak nitelenmiş olarak işlenir. Çift tırnak işaretlerini çıkarırsanız, sandboxRoot öğesinin başına yürürlükteki kullanıcı kimliği eklenir. Örneğin, sandboxRoot özelliğini şu değere ayarlarsanız: `sandboxRoot=//test`, aracı şu veri kümelerine erişebilir (standart z/OS gösteriminde) *//username*`.test.**` Çalıştırma zamanında, tam olarak çözümlenen veri kümesi adının ilk düzeyleri sandboxRootile eşleşmiyorsa, aktarma isteği reddedilir.

IBM i sistemlerinde kum havuzunda çalışma

IBM i IBM i sistemlerinde tümleşik dosya sistemindeki dosyalar için, çalışma yeri kullanımı bir Managed File Transfer Agent ' in hangi dizinlerden okuyabileceğini ve hangi dizinlere yazabileceğini sınırlar. Kum havuzu kullanımı etkinleştirildiğinde, Managed File Transfer Agent izin verildiği şekilde belirtilen dizinleri ve alt dizinler sandboxRootreddedildiği gibi belirtilmedikçe, belirtilen dizinlerin içerdiği tüm alt dizinleri okuyabilir ve bu dizinlere yazabilir. Managed File Transfer kum havuzu kullanımı, işletim sistemi güvenliğinden daha öncelikli değildir. Managed File Transfer Agent ' i başlatan kullanıcının, dizinden okuyabilmesi ya da dizine yazabilmesi için herhangi bir dizine uygun işletim sistemi düzeyinde erişimi olmalıdır. Bağlantı verilen dizin belirtilen sandboxRoot dizinlerinin (ve alt dizinlerin) dışındaysa, bir dizine yönelik simgesel bir bağlantı izlenmez.

İlgili başvurular

“Joker karakter aktarımları için ek denetimler” sayfa 598
Bir aracı, aracının dosyaları aktarabileceği konumları kısıtlamak için bir kullanıcı ya da aracı korumalı alanı ile yapılandırıldıysa, o aracı için genel arama karakteri aktarımlarında ek denetimler yapılabileceğini belirtebilirsiniz.

“MFT Agent kum havuzlarıyla çalışma” sayfa 594
Managed File Transfer' e ek bir güvenlik düzeyi eklemek için, bir aracının erişebileceği dosya sistemi alanını sınırlayabilirsiniz.

MFT `agent.properties` dosyası

MFT kullanıcı kum havuzlarıyla çalışma

Dosyaların aktarılabileceği dosya sistemi alanını, aktarma isteğinde bulunan MQMD kullanıcı adına dayalı olarak sınırlandırabilirsiniz.

Aracı bir iletişim kuralı köprüsü aracısı ya da bir Connect:Direct köprü aracısı olduğunda kullanıcı kum havuzları desteklenmez.

Kullanıcı korumalı alanı kullanımını etkinleştirmek için, kısıtlamak istediğiniz aracının `agent.properties` dosyasına aşağıdaki özelliği ekleyin:

```
userSandboxes=true
```

Bu özellik var olduğunda ve true değerine ayarlandığında aracı, dosya sisteminin hangi bölümlerine erişebileceğini belirlemek için *MQ_DATA_PATH*`/mqft/config/`*coordination_qmgr_name*`/agents/`*agent_name*`/UserSandboxes.xml` dosyasındaki bilgileri kullanır.

`UserSandboxes.xml` XML, sıfır ya da daha fazla `<sandbox>` öğesi içeren bir `<agent>` öğesinden oluşur. Bu öğeler, hangi kullanıcıların hangi kurallara uygulandığını açıklar. `<sandbox>` öğesinin `user` özniteliği, isteğin MQMD kullanıcısıyla eşleşmek için kullanılan bir kalıptır.

`UserSandboxes.xml` dosyası aracı tarafından düzenli olarak yeniden yüklenir ve dosyada yapılan geçerli değişiklikler aracının davranışını etkiler. Varsayılan yeniden yükleme aralığı 30 saniyedir. Bu aralık, `agent.properties` dosyasında xmlConfigReloadInterval aracı özelliği belirtilerek değiştirilebilir.

`userPattern="regex"` özniteliğini ya da değerini belirtirseniz, `user` özniteliği Java düzenli ifadesi olarak yorumlanır. Daha fazla bilgi için bkz. MFT tarafından kullanılan düzenli ifadeler.

`userPattern="regex"` özniteliğini ya da değerini belirtmezseniz, `user` özniteliği aşağıdaki genel arama karakterlerini içeren bir kalıp olarak yorumlanır:

- Sıfır ya da daha fazla karakteri gösteren yıldız işareti (*)
- soru imi (?), tam olarak bir karakteri temsil eder

Eşleşmeler, <sandbox> öğelerinin dosyada listelendiği sırayla gerçekleştirilir. Yalnızca ilk eşleşme kullanılır, dosyadaki aşağıdaki olası eşleşmelerin tümü yoksayılır. Dosyada belirtilen <sandbox> öğelerinin hiçbiri, aktarma isteği iletisiyle ilişkilendirilmiş MQMD kullanıcısıyla eşleşmiyorsa, aktarma dosya sistemine erişemez. MQMD kullanıcı adı ile `user` özniteliği arasında bir eşleşme bulunduğunda, eşleşme, aktarmaya uygulanan bir <sandbox> öğesi içinde bir kural kümesini tanıtır. Bu kural kümesi, hangi dosyalarınya da veri kümelerinin aktarımın bir parçası olarak okunabileceğini ya da yazılabileceğini belirlemek için kullanılır.

Her kural kümesi, hangi dosyaların okunabileceğini tanımlayan bir <read> öğesini ve hangi dosyaların yazılabileceğini tanımlayan bir <write> öğesini belirtebilir. <read> ya da <write> öğelerini bir kural kümesinden çıkarırsanız, bu kural kümesiyle ilişkilendirilmiş kullanıcının herhangi bir okuma ya da yazma işlemi gerçekleştirmesine izin verilmediği varsayılır.

Not: <read> öğesi <write> öğesinden önce, <include> öğesi `UserSandboxes.xml` dosyasında <exclude> öğesinden önce olmalıdır.

Her <read> ya da <write> öğesi, bir dosyanın kum havuzunda olup olmadığını ve aktarılıp aktarılamayacağını belirlemek için kullanılan bir ya da daha fazla kalıp içerir. <include> ve <exclude> öğelerini kullanarak bu kalıpları belirtin. <include> ya da <exclude> öğesinin `name` özniteliği, eşleştirilecek örüntüyü belirtir. İsteğe bağlı bir `type` özniteliği, ad değerinin bir dosya mı, yoksa kuyruk kalıbı mı olduğunu belirtir. `type` özniteliği belirtilmezse, aracı örüntüyü bir dosya ya da dizin yolu örüntüsü olarak işler. Örneğin:

```
<tns:read>
    <tns:include name="/home/user/**"/>
    <tns:include name="USER.**" type="queue"/>
    <tns:exclude name="/home/user/private/**"/>
</tns:read>
```

<include> ve <exclude> `name` kalıpları, aracı tarafından dosyaların, veri kümelerinin ya da kuyrukların okunup okunamayacağını ya da yazılıp yazılamayacağını belirlemek için kullanılır. Kurallı dosya yolu, veri kümesi ya da kuyruk adı, dahil edilen kalıplardan en az biriyle ve kapsam dışı bırakılan kalıplardan tam olarak sıfır ile eşleşiyorsa bir işleme izin verilir. <include> ve <exclude> öğelerinin `name` özniteliği kullanılarak belirtilen örüntüler, aracının çalıştığı platforma uygun yol ayırıcıları ve kuralları kullanır. Göreli dosya yolları belirtirseniz, yollar aracının `transferRoot` özelliğiyle göreli olarak çözümlenir.

Bir kuyruk kısıtlaması belirtilirken, QUEUE@QUEUEMANAGER sözdizimi aşağıdaki kurallarla desteklenir:

- Girişte at karakteri (@) yoksa, örüntü, herhangi bir kuyruk yöneticisinde erişilebilen bir kuyruk adı olarak kabul edilir. Örneğin, örüntü `name` ise, `name@**`ile aynı şekilde işlenir.
- Girişteki ilk karakter (@) ise, örüntü bir kuyruk yöneticisi adı olarak işlenir ve kuyruk yöneticisindeki tüm kuyruklara erişilebilir. Örneğin, örüntü `@name` ise, `**@name`ile aynı şekilde işlenir.

Aşağıdaki genel arama karakterleri, <include> ve <exclude> öğelerinin `name` özniteliğinin bir parçası olarak belirtildiğinde özel anlam taşır:

*****

Tek bir yıldız işareti, dizin adında ya da veri kümesi adı ya da kuyruk adı niteleyicisinde sıfır ya da daha fazla karakterle eşleşir.

?
Soru işareti, bir dizin adında ya da veri kümesi adının ya da kuyruk adının niteleyicisinde tam olarak bir karakterle eşleşir.

******
İki yıldız işareti sıfır ya da daha fazla dizin adıyla ya da bir veri kümesi adında ya da kuyruk adında sıfır ya da daha fazla niteleyiciyle eşleşir. Ayrıca, yol ayırıcısıyla biten yolların sonuna örtük bir "**" eklenmiş olur. `/home/user/` , `/home/user/**`ile aynıdır.

Örneğin:

- `/**/test/**` , yolunda `test` dizini olan herhangi bir dosyayla eşleşir
- `/test/file?` , `/test` dizininde `file` dizgisiyle başlayan ve ardından herhangi bir tek karakterin izlediği herhangi bir dosyayla eşleşir
- `c:\test\*.txt` , `c:\test` dizinindeki herhangi bir dosyayla `.txt` uzantısıyla eşleşir
- `c:\test\**\*.txt` , `'c:\test` dizinindeki herhangi bir dosyayla ya da `.txt` uzantısına sahip alt dizinlerinden biriyle eşleşir
- z/OS `//'TEST.*.DATA'` , ilk niteleyicisi TESTolan, ikinci niteleyicisi ve üçüncü niteleyicisi DATAolan herhangi bir veri kümesiyle eşleşir.
- *@QM1 , tek bir niteleyicisi olan QM1 kuyruk yöneticisindeki herhangi bir kuyruğuyla eşleşir.
- `TEST.*.QUEUE@QM1` , TESTilk niteleyicisine sahip QM1 kuyruk yöneticisindeki herhangi bir kuyruğuyla eşleşir, herhangi bir ikinci niteleyiciye ve QUEUEüçüncü niteleyicisine sahiptir.
- **@QM1 , QM1kuyruk yöneticisindeki herhangi bir kuyrukla eşleşir.

Simgesel bağlantılar

`<include>` ve `<exclude>` öğelerinde sabit bağlantılar belirterek, `UserSandboxes.xml` dosyasındaki dosya yollarında kullandığınız simgesel bağlantıları tam olarak çözmeniz gerekir. Örneğin, `/var`'un `/SYSTEM/var`ile eşleştiği sembolik bir bağlanmanız varsa, bu yolu `<tns:include name="/SYSTEM/var"/>`olarak belirtmeniz gerekir; tersi durumda, istenen aktarma bir kullanıcı korumalı alanı güvenlik hatasıyla başarısız olur.

Örnek

Bu örnek, `guest` MQMD kullanıcı adına sahip kullanıcının AGENT_JUPITER aracının çalıştığı sistemdeki `/home/user/public` dizininden ya da alt dizinlerinden herhangi birini aktarmasına AGENT_JUPITER yapılanış dizinindeki `UserSandboxes.xml` dosyasına aşağıdaki `<sandbox>` öğesini ekleyerek nasıl izin verileceğini gösterir:

```
<?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>
```

Örnek

Bu örnek, MQMD kullanıcı adı `account` ve ardından tek bir basamak (örneğin, `account4`) olan herhangi bir kullanıcının aşağıdaki işlemleri tamamlamasına nasıl izin verileceğini gösterir:

- AGENT_SATÜRN aracısının çalıştığı sistemdeki /home/account/private dizini dışında, herhangi bir dosyayı /home/account dizininden ya da alt dizinlerinden aktarın.
- Herhangi bir dosyayı, AGENT_SATÜRN aracısının çalıştığı sistemdeki /home/account/output dizinine ya da alt dizinlerinden herhangi birine aktarın.
- Read messages from queues on the local queue manager starting with the prefix ACCOUNT. unless it starts with ACCOUNT.PRIVATE. (that is has PRIVATE at the second level).
- Herhangi bir kuyruk yöneticisinde ACCOUNT.OUTPUT. önekiyle başlayan kuyruklara veri aktarın.

account MQMD kullanıcı adına sahip bir kullanıcının bu işlemleri tamamlamasına izin vermek için, AGENT_SATÜRN yapılanış dizininde UserSandboxes.xmlkütüğüne şu <sandbox> öğesini ekleyin:

```
<?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>
```

İlgili başvurular

“Joker karakter aktarımları için ek denetimler” sayfa 598
Bir aracı, aracının dosyaları aktarabileceği konumları kısıtlamak için bir kullanıcı ya da aracı korumalı alanı ile yapılandırıldıysa, o aracı için genel arama karakteri aktarımlarında ek denetimler yapılabileceğini belirtebilirsiniz.

MFT agent.properties dosyası

Joker karakter aktarımları için ek denetimler

Bir aracı, aracının dosyaları aktarabileceği konumları kısıtlamak için bir kullanıcı ya da aracı korumalı alanı ile yapılandırıldıysa, o aracı için genel arama karakteri aktarımlarında ek denetimler yapılabileceğini belirtebilirsiniz.

additionalWildcardSandboxChecking özelliği

Joker karakter aktarımlarına ilişkin ek denetlemeyi etkinleştirmek için, denetlemek istediğiniz aracıya ilişkin agent.properties dosyasına aşağıdaki özelliği ekleyin.

```
additionalWildcardSandboxChecking=true
```

Bu özellik true olarak ayarlandığında ve aracı, genel arama karakterinin dosya eşleştirmesi için tanımlanan kum havuzunun dışındaki bir konumu okumaya çalışan bir aktarma isteğinde bulunduğunda, aktarma başarısız olur. Bir aktarma isteği içinde birden çok aktarım varsa ve bu isteklerden biri, kum havuzunun dışındaki bir konumu okuma girişimi nedeniyle başarısız olursa, tüm aktarma başarısız olur. Denetleme başarısız olursa, hata nedeni bir hata iletisinde verilir.

additionalWildcardSandboxChecking özelliği bir aracının agent.properties dosyasında atlanırsa ya da false olarak ayarlanırsa, o aracı için genel arama karakteri aktarımlarında ek denetimler yapılmaz.

Joker karakter denetimine ilişkin hata iletileri

Yapılandırılmış bir kum havuzu konumu dışındaki bir konuma joker karakter aktarma isteği yapıldığında bildirilen iletiler aşağıdaki gibidir.

Bir aktarma isteğindeki genel arama karakteri dosya yolu, kısıtlı çalışma yerinin dışında bulunduğunda aşağıdaki ileti oluşur:

```
BFGSS0077E: Dosya yolunu okuma girişimi: yol reddedildi.
Dosya yolu, kısıtlı aktarma korumalı alanının dışında bulunuyor.
```

Birden çok aktarma isteği içindeki bir aktarma, yolun sınırlı çalışma yerinin dışında bulunduğu bir genel arama karakteri aktarma isteği içerdiğinde aşağıdaki ileti görüntülenir:

```
BFGSS0078E: Dosya yolunu okuma girişimi: yol başka bir aktarma olarak yoksayıldı
Yönetilen aktarımda bulunan öğe, sınırlı aktarım korumalı alanının dışında okuma girişiminde
bulundu.
```

Bir dosya sınırlı korumalı alanın dışında bulunduğunda aşağıdaki ileti oluşur:

```
BFGSS0079E: dosya yolunu okuma girişimi reddedildi.
Dosya, kısıtlı aktarma korumalı alanının dışında bulunuyor.
```

Başka bir genel arama karakteri aktarma isteğinin bu iletinin yoksayılmasına neden olduğu birden çok aktarma isteğinde aşağıdaki ileti görüntülenir:

```
BFGSS0080E: Dosyayı okuma girişimi: dosya yolu başka bir aktarma olarak yoksayıldı
Yönetilen aktarımda bulunan öğe, sınırlı aktarım korumalı alanının dışında okuma girişiminde
bulundu.
```

Joker karakter içermeyen tek dosya aktarımları söz konusu olduğunda, aktarma işlemi kum havuzunun dışında bulunan bir dosyayı içerdiğinde bildirilen ileti önceki yayın düzeylerinden değiştirilmez:

```
BFGIO0056E: "FILE" dosyasını okuma girişimi reddedildi.
Dosya, kısıtlı aktarma korumalı alanının dışında bulunuyor.
```

İlgili başvurular

“MFT kullanıcı kum havuzlarıyla çalışma” sayfa 595
Dosyaların aktarılabileceği dosya sistemi alanını, aktarma isteğinde bulunan MQMD kullanıcı adına dayalı olarak sınırlandırabilirsiniz.

“MFT Agent kum havuzlarıyla çalışma” sayfa 594
Managed File Transfer' e ek bir güvenlik düzeyi eklemek için, bir aracının erişebileceği dosya sistemi alanını sınırlayabilirsiniz.

MFT agent.properties dosyası

MFT için SSL ya da TLS şifrelemesini yapılandırma

Aracılar ve aracı kuyruk yöneticileri, komutlar ve bağlı oldukları kuyruk yöneticileri ile topolojiniz içindeki çeşitli kuyruk yöneticisi bağlantıları arasındaki iletişimi korumak için IBM MQ Managed File Transfer ile SSL ya da TLS kullanabilirsiniz.

Başlamadan önce

IBM MQ Managed File Transfer topolojisinden akan iletileri şifrelemek için SSL ya da TLS şifrelemesini kullanabilirsiniz. Bunlar arasında aşağıdakiler yer alır:

- Bir aracı ile aracı kuyruk yöneticisi arasında geçen iletiler.
- Bağlı oldukları komutlara ve kuyruk yöneticilerine ilişkin iletiler.
- Topoloji içindeki aracı kuyruk yöneticileri, komut kuyruğu yöneticileri ve koordinasyon kuyruk yöneticisi arasında akan iç iletiler.

Bu görev hakkında

IBM MQile SSL kullanma hakkında genel bilgi için bkz. “SSL/TLS ile çalışma” sayfa 271. IBM MQ terimlerinde Managed File Transfer , standart bir Java istemci uygulamasıdır.

Managed File Transferile SSL kullanmak için aşağıdaki adımları izleyin:

Yordam

1. Bir güvenli depo dosyası ve isteğe bağlı olarak bir anahtar deposu dosyası yaratın (bu dosyalar aynı dosya olabilir). İstemci kimlik doğrulamasına (kanallarda SSLCAUTH=ISTEĞE bağlı) gerek duymuyorsanız, anahtar deposu belirtmeniz gerekmez. Yalnızca kuyruk yöneticisinin sertifikasını doğrulamak için bir güvenli depo gerekir.

   IBM MQile çalışmak için güvenli depo ve anahtar depoları için sertifika yaratmak üzere kullanılan anahtar algoritması RSA olmalıdır.
2. IBM MQ kuyruk yöneticinizi SSL kullanacak şekilde ayarlayın.

   IBM MQ Explorer komutunu kullanarak bir kuyruk yöneticisinin SSL kullanacak şekilde ayarlanmasıyla ilgili bilgi için Kuyruk yöneticisinde SSL yapılandırılmasıbaşlıklı konuya bakın.
3. Güvenli depo dosyasını ve anahtar deposu dosyasını (varsa) uygun bir konuma kaydedin. Önerilen konum, *config_directory*/*coordination_qmgr*/agents/*agent_name* dizinidir.
4. SSL özelliklerini, uygun Managed File Transfer özellikler dosyasındaki her SSL etkin kuyruk yöneticisi için gerektiği şekilde ayarlayın. Her bir özellik kümesi ayrı bir kuyruk yöneticisine (aracı, eşgüdüm ve komut) gönderme yapar, ancak bir kuyruk yöneticisi bu rollerden iki ya da daha fazlasını gerçekleştirebilir.

   CipherSpec ya da **CipherSuite** özelliklerinden biri gereklidir; tersi durumda, istemci SSL olmadan bağlanmayı dener. IBM MQ ve Javaarasındaki terminoloji farklılıkları nedeniyle hem **CipherSpec** hem de **CipherSuite** özellikleri sağlanır. Managed File Transfer , her iki özelliği de kabul eder ve gerekli dönüştürmeyi yapar; bu nedenle, her iki özelliği de ayarlamanız gerekmez. Hem **CipherSpec** hem de **CipherSuite** özelliklerini belirtirseniz, **CipherSpec** öncelikli olur.

   PeerName özelliği isteğe bağlıdır. Özelliği, bağlanmak istediğiniz kuyruk yöneticisinin Ayırt Edici Adı olarak ayarlayabilirsiniz. Managed File Transfer , Ayırt Edici Adı eşleşmeyen yanlış bir SSL sunucusuna yönelik bağlantıları reddeder.

   SslTrustStore ve **SslKeyStore** özelliklerini, güvenli depo ve anahtar deposu dosyalarını işaret eden dosya adlarına ayarlayın. Çalışmakta olan bir aracı için bu özellikleri ayarlıyorsa, SSL kipinde yeniden bağlanmak için aracıyı durdurun ve yeniden başlatın.

   Özellikler dosyaları düz metin parolaları içerir, bu nedenle uygun dosya sistemi izinlerini ayarlamayı göz önünde bulundurun.

   SSL özellikleriyle ilgili daha fazla bilgi için bkz. “MFT için SSL/TLS özellikleri” sayfa 600.
5. Bir aracı kuyruk yöneticisi SSL kullanıyorsa, aracıyı oluştururken gerekli ayrıntıları sağlayamazsınız. Aracıyı oluşturmak için aşağıdaki adımları kullanın:
   a) **fteCreateAgent** komutunu kullanarak aracıyı oluşturun. Aracının varlığını eşgüdüm kuyruk yöneticisine yayınlayamıyorsunuz uyarısı alırsınız.
   b) SSL bilgilerini eklemek için önceki adımla oluşturulan agent.properties dosyasını düzenleyin. Aracı başarıyla başlatıldığında, yayınlama yeniden denenir.
6. agent.properties dosyasında ya da coordination.properties dosyasında SSL özellikleri değiştirilirken IBM MQ Explorer aracıları ya da eşgörünümleri çalışıyorsa, aracıyı ya da IBM MQ Explorer' yi yeniden başlatmanız gerekir.

İlgili başvurular

MFT agent.properties dosyası

MFT için SSL/TLS özellikleri

Bazı MFT özellik dosyaları SSL ve TLS özelliklerini içerir. Aracılar ve kuyruk yöneticileri arasında yetkisiz bağlantıları önlemek ve aracılar ve kuyruk yöneticileri arasındaki ileti trafiğini şifrelemek için IBM MQ ve Managed File Transfer ile SSL ya da TLS kullanabilirsiniz.

Aşağıdaki MFT özellikler dosyaları SSL özelliklerini içerir:

- MFT agent.properties dosyası için SSL/TLS özellikleri
- MFT coordination.properties dosyası için SSL/TLS özellikleri

- MFT `command.properties` dosyası için SSL/TLS özellikleri
- MFT `logger.properties` dosyası için SSL/TLS özellikleri

Managed File Transferile SSL ya da TLS kullanma hakkında bilgi için bkz. "MFT için SSL ya da TLS şifrelemesini yapılandırma" sayfa 599.

IBM WebSphere MQ 7.5' den dosya ya da dizin konumlarını gösteren bazı Managed File Transfer özelliklerinde ortam değişkenlerini kullanabilirsiniz. Bu, ürünün parçalarını çalıştırırken kullanılan dosyaların ya da dizinlerin konumlarının, işlemi hangi kullanıcının çalıştırdığı gibi ortam değişikliklerine bağlı olarak değişmesine olanak sağlar. Daha fazla bilgi için bkz. MFT özelliklerinde ortam değişkenlerinin kullanımı.

İlgili kavramlar

Çoklu platformlarda MFT yapılandırma seçenekleri

İlgili başvurular

MFT özelliklerinde ortam değişkenlerinin kullanımı

Kanal kimlik doğrulamasıyla istemci kipinde bir kuyruk yöneticisine bağlanma

IBM MQ , kanal düzeyinde daha kesin erişimi denetlemek için kanal kimlik doğrulama kayıtlarını kullanır. Bu, varsayılan olarak yeni yaratılan kuyruk yöneticilerinin Managed File Transfer bileşeninden istemci bağlantılarını reddettiği anlamına gelir.

Kanal kimlik doğrulamasıyla ilgili daha fazla bilgi için bkz. "Kanal kimlik doğrulama kayıtları" sayfa 50.

Managed File Transfer tarafından kullanılan SVRCONN için kanal kimlik doğrulama yapılandırması ayrıcalıklı olmayan bir MCAUSER kimliğini belirtiyorsa, Managed File Transfer Agent ve komutlarının doğru çalışması için kuyruk yöneticisi, kuyruklar ve konular için belirli yetki kayıtları vermeniz gerekir. Kanal kimlik doğrulama kayıtlarını yaratmak, değiştirmek ya da kaldırmak için SET CHLAUTH MQSC komutunu ya da Set Channel Authentication Record PCF komutunu kullanın. IBM MQ kuyruk yöneticisine bağlanmak istediğiniz tüm Managed File Transfer aracıları için, tüm aracılarınız için kullanılacak bir MCAUSER kimliği ayarlayabilir ya da her aracı için ayrı bir MCAUSER tanıtıcısı ayarlayabilirsiniz.

Her bir MCAUSER Kimliğine aşağıdaki izinleri verin:

- Kuyruk yöneticisi için gereken yetki kayıtları:
  - bağlan
  - SETID
  - inq
- Kuyruklar için yetki kayıtları gerekiyor.

  Aşağıdaki listede *agent_name* ile biten, aracıya özgü tüm kuyruklar için, bir istemci bağlantısı kullanarak IBM MQ kuyruk yöneticisine bağlanmak istediğiniz her aracı için bu kuyruk yetkisi kayıtlarını oluşturmanız gerekir.
  - koyma, alma, dsp (SYSTEM.DEFAULT.MODEL.QUEUE)
  - koyma, alma, setid, göz atma (SYSTEM.FTE.COMMAND.*aracı_adı*)
  - koyma, bkz. (SYSTEM.FTE.DATA.*aracı_adı*)
  - koyma, bkz. (SYSTEM.FTE.REPLY.*aracı_adı*)
  - koyma, alma, inq, göz atma (SYSTEM.FTE.STATE.*aracı_adı*)
  - koyma, alma, göz atma (SYSTEM.FTE.EVENT.*aracı_adı*)
  - koyma, bkz. (SYSTEM.FTE)
- Konular için gerekli yetki kayıtları:
  - alt, pub (SYSTEM.FTE)
- Dosya aktarımları için gerekli yetki kayıtları.

Kaynak ve hedef aracı için ayrı MCAUSER kimlikleriniz varsa, hem kaynak hem de hedefte aracıların kuyruklarında yetki kayıtları oluşturun.

Örneğin, kaynak aracının MCAUSER kimliği **user1** ve hedef aracı MCAUSER kimliği **user2**ise, aracı kullanıcıları için aşağıdaki yetkileri ayarlayın:

AGENT kullanıcısı	Kuyruk	Yetki gerekli
user1	SYSTEM.FTE.DATA.*hedef_aracı_adı*	put
user1	SYSTEM.FTE.COMMAND.*hedef_aracı_adı*	put
user2	SYSTEM.FTE.REPLY.*kaynak_aracı_adı*	put
user2	SYSTEM.FTE.COMMAND.*kaynak_aracı_adı*	put

Connect:Direct köprü aracısı ile Connect:Direct düğümü arasında SSL ya da TLS ' nin yapılandırılması

Bir anahtar deposu ve güvenilir depo yaratarak ve Connect:Direct köprü aracısı özellikler dosyasında özellikleri ayarlayarak, Connect:Direct köprü aracısını ve Connect:Direct düğümünü SSL protokolü aracılığıyla birbirine bağlanacak şekilde yapılandırın.

Bu görev hakkında

Bu adımlar, bir sertifika yetkilisi tarafından imzalanmış anahtarlarınızı almaya ilişkin yönergeleri içerir. Sertifika yetkilisi kullanmıyorsanız, kendinden onaylı bir sertifika oluşturabilirsiniz. Kendinden onaylı sertifika oluşturma hakkında daha fazla bilgi için bkz. “AIX, Linux, and Windows üzerinde SSL/TLS ile çalışma” sayfa 289.

Bu adımlar, Connect:Direct köprü aracısı için yeni bir anahtar deposu ve güvenilir depo yaratılmasına ilişkin yönergeleri içerir. Connect:Direct Bridge aracısının IBM MQ kuyruk yöneticilerine güvenli bir şekilde bağlanmak için kullandığı bir anahtar deposu ve güvenilir deposu varsa, Connect:Direct düğümüne güvenli bir şekilde bağlanırken var olan anahtar deposunu ve güvenilir depoyu kullanabilirsiniz. Daha fazla bilgi için bkz “MFT için SSL ya da TLS şifrelemesini yapılandırma” sayfa 599.

Yordam

Connect:Direct düğümü için aşağıdaki adımları tamamlayın:

1. Connect:Direct düğümü için bir anahtar ve imzalı sertifika oluşturun.

   Bunu, IBM MQile birlikte gönderilen IBM Key Management aracını kullanarak yapabilirsiniz. Daha fazla bilgi için bkz “SSL/TLS ile çalışma” sayfa 271.
2. Anahtarın imzalanması için sertifika yetkilisine bir istek gönderin. Karşılığında bir sertifika alırsınız.
3. Sertifika yetkilinizin genel anahtarını içeren bir metin dosyası oluşturun; örneğin, `/test/ssl/certs/CAcert`.
4. Connect:Direct düğümüne Secure + Option ürününü kurun.

   Düğüm zaten varsa, kuruluş programını yeniden çalıştırarak, var olan kuruluşun konumunu belirterek ve yalnızca Güvenli + Seçenek 'i kurmayı seçerek Güvenli + Seçeneği 'ni kurabilirsiniz.
5. Yeni bir metin dosyası oluşturun; örneğin, `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt`.
6. Sertifika yetkilinizden aldığınız sertifikayı ve `/test/ssl/cd/privateKeys/`*`node_name`*`.key`içinde bulunan özel anahtarı metin dosyasına kopyalayın.

   `/test/ssl/cd/keyCertFile/`*`node_name`*`.txt` içeriği aşağıdaki biçimde olmalıdır:

```
-----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. Güvenli + Yönetim Aracı 'nı başlatın.
   - AIX and Linux sistemlerinde **spadmin.sh**komutunu çalıştırın.
   - Windows sistemlerinde **Start** > **Programs** > **Sterling Commerce Connect:Direct** > **CD Secure + Admin Tool** (Programları Başlat) seçeneğini tıklatın.

   CD Secure + Admin Tool başlatılır.
8. CD Secure + Admin Tool 'da **öğesini çift tıklatın.Ana SSL ya da TLS ayarlarını düzenlemek için yerel** hattı.
   a) Kullandığınız protokole bağlı olarak **Enable SSL Protocol** (SSL İletişim Kuralını Etkinleştir) ya da **Enable TLS Protocol**(TLS İletişim Kuralını Etkinleştir) seçeneğini belirleyin.
   b) **Geçersiz Kılmayı Geçersiz Kıl**seçeneğini belirleyin.
   c) En az bir şifreleme takımı seçin.
   d) İki yönlü kimlik doğrulaması istiyorsanız, **İstemci Kimlik Doğrulamasını Etkinleştir** değerini Yesolarak değiştirin.
   e) **Güvenilir Kök Sertifika** alanında, sertifika yetkilinizin genel sertifika dosyasının yolunu girin (`/test/ssl/certs/CAcert`).
   f) **Anahtar Sertifika Dosyası** alanında, oluşturduğunuz dosyanın yolunu girin (`/test/ssl/cd/keyCertFile/`*`node_name`*`.txt`).
9. **simgesini çift tıklatın.Ana SSL ya da TLS ayarlarını düzenlemek için İstemci** hattı.
   a) Kullandığınız protokole bağlı olarak **Enable SSL Protocol** (SSL İletişim Kuralını Etkinleştir) ya da **Enable TLS Protocol**(TLS İletişim Kuralını Etkinleştir) seçeneğini belirleyin.
   b) **Geçersiz Kılmayı Geçersiz Kıl**seçeneğini belirleyin.

Connect:Direct köprü aracısı için aşağıdaki adımları gerçekleştirin:

10. Bir güvenli depo oluşturun. Bunu, kukla bir anahtar yaratıp kukla anahtarı silerek yapabilirsiniz.

    Aşağıdaki komutları kullanabilirsiniz:

```
keytool -genkey -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

```
keytool -delete -alias dummy -keystore /test/ssl/fte/stores/truststore.jks
```

11. Sertifika yetkilisinin genel sertifikasını güvenli depoya aktarın.

Aşağıdaki komutu kullanabilirsiniz:

```
keytool -import -trustcacerts -alias myCA
        -file /test/ssl/certs/CAcert
        -keystore /test/ssl/fte/stores/truststore.jks
```

12. Connect:Direct köprü aracısı özellikler dosyasını düzenleyin.

    Dosyanın herhangi bir yerinde aşağıdaki satırları ekleyin:

    ```
    cdNodeProtocol=protocol
    cdNodeTruststore=/test/ssl/fte/stores/truststore.jks
    cdNodeTruststorePassword=password
    ```

    Bu adımdaki örnekte *iletişim kuralı* , kullandığınız iletişim kuralıdır (SSL ya da TLS) ve *parola* , güvenli depoyu oluşturduğunuzda belirttiğiniz paroladır.
13. İki yönlü kimlik doğrulaması istiyorsanız, Connect:Direct köprü aracısı için bir anahtar ve sertifika oluşturun.
    a) Bir anahtar deposu ve anahtar oluşturun.

       Aşağıdaki komutu kullanabilirsiniz:

       ```
       keytool -genkey -keyalg RSA -alias agent_name
               -keystore /test/ssl/fte/stores/keystore.jks
               -storepass password -validity 365
       ```

    b) Bir imzalama isteği oluşturun.

       Aşağıdaki komutu kullanabilirsiniz:

       ```
       keytool -certreq -v -alias agent_name
               -keystore /test/ssl/fte/stores/keystore.jks -storepass password
               -file /test/ssl/fte/requests/agent_name.request
       ```

    c) Önceki adımdan aldığınız sertifikayı anahtar deposuna aktarın. Sertifika x.509 biçiminde olmalıdır.

       Aşağıdaki komutu kullanabilirsiniz:

       ```
       keytool -import -keystore /test/ssl/fte/stores/keystore.jks
               -storepass password -file certificate_file_path
       ```

    d) Connect:Direct köprü aracısı özellikler dosyasını düzenleyin.

       Dosyanın herhangi bir yerinde aşağıdaki satırları ekleyin:

       ```
       cdNodeKeystore=/test/ssl/fte/stores/keystore.jks
       cdNodeKeystorePassword=password
       ```

       Bu adımdaki örnekte *parola* , anahtar deposunu yaratırken belirttiğiniz paroladır.

İlgili görevler

Connect:Direct köprüsünün yapılandırılması

ALW AMQP istemcilerinin güvenliğinin sağlanması

AMQP istemcilerinden gelen bağlantıların güvenliğini sağlamak ve verilerin ağ üzerinde uygun şekilde korunduğundan emin olmak için çeşitli güvenlik mekanizmaları kullanabilirsiniz. MQ Light uygulamalarınızda güvenlik oluşturabilirsiniz. IBM MQ ' in var olan güvenlik özelliklerini AMQP istemcileriyle birlikte, özelliklerin diğer uygulamalar için kullanılmasıyla aynı şekilde kullanabilirsiniz.

Kanal kimlik doğrulama kuralları (CHLAUTH)

TCP bağlantılarını bir kuyruk yöneticisiyle sınırlamak için kanal kimlik doğrulama kurallarını kullanabilirsiniz. AMQP kanalları, kuyruk yöneticiniz için yapılandırdığınız kanal kimlik doğrulama kurallarının kullanılmasını destekler. Kanal kimlik doğrulama kuralları, kuyruk yöneticinizdeki AMQP kanallarıyla eşleşen bir tanıtımla tanımlanırsa, bu kurallar bu kanallara uygulanır. Varsayılan olarak, yeni IBM® MQ kuyruk yöneticileriyle kanal kimlik doğrulaması etkinleştirilir; bu nedenle, AMQP kanalını kullanmadan önce en az bir yapılandırmayı tamamlamanız gerekir.

Kanal kimlik doğrulama kurallarının kuyruk yöneticinizle AMQP bağlantılarına izin verecek şekilde nasıl yapılandırılacağı hakkında daha fazla bilgi için bkz. AMQP kanalları oluşturma ve kullanma.

Bağlantı kimlik doğrulaması (CONNAUTH)

Bir kuyruk yöneticisine yönelik bağlantıların kimliğini doğrulamak için bağlantı kimlik doğrulamasını kullanabilirsiniz. AMQP kanalları, AMQP uygulamalarından kuyruk yöneticisine erişimi denetlemek için bağlantı kimlik doğrulamasının kullanılmasını destekler.

AMQP protokolü, bir bağlantının nasıl doğrulandığını belirtmek için SASL (Basit Kimlik Doğrulama ve Güvenlik Katmanı) çerçevesini kullanır. Çeşitli SASL mekanizmaları vardır ve IBM MQ iki SASL mekanizmasını destekler: ANONYMOUS ve PLAIN.

ANONYMOUS durumunda, kimlik doğrulaması için istemciden kuyruk yöneticisine kimlik bilgileri iletilmez. CONNAUTH özniteliğinde belirtilen MQ AUTHINFO nesnesinin CHCKCLNT değeri REQUIRED ya da REQDADM (denetimci kullanıcı olarak bağlanıyorsa) ise, bağlantı reddedilmiştir. CHCKCLNT değeri NONE ya da OPTIONAL ise, bağlantı kabul edilir.

PLAIN durumunda, kimlik doğrulaması için istemciden kuyruk yöneticisine bir kullanıcı adı ve parola geçirilir. CONNAUTH özniteliğinde belirtilen MQ AUTHINFO nesnesi NONE değerine sahipse, bağlantı reddedilmiştir. CHCKCLNT değeri OPTIONAL, REQUIRED ya da REQDADM (denetimci kullanıcı olarak bağlanıyorsa) ise, kullanıcı adı ve parola kuyruk yöneticisi tarafından denetlenir. Kuyruk yöneticisi işletim sistemini (AUTHINFO nesnesi IDPWOS tipindeyse) ya da bir LDAP havuzunu (AUTHINFO nesnesi IDPWLDAP tipindeyse) denetler.

Aşağıdaki tablo bu kimlik doğrulama davranışını özetler:

Çizelge 102. SASL mekanizmalarının ve bağlantı kimlik doğrulamasının özeti

SASL mekanizması	**Kimlik bilgileri istemciden kuyruk yöneticisine iletilsin mi?**	**CHKCLNT değeri**
anonim	Hayır	REQUIRED ya da REQDADM-bağlantı reddedildi NONE ya da OPTIONAL-bağlantı kabul edildi
Düz	Evet, kullanıcı adı ve parola	REQUIRED, REQDADM ya da OPTIONAL-kuyruk yöneticisi tarafından denetlenen kullanıcı adı ve parola NONE-bağlantı reddedildi

Bir MQ Light istemcisi kullanıyorsanız, bunları bağlandığınız AMQP adresine ekleyerek kimlik bilgilerini belirtebilirsiniz; örneğin:

```
amqp://mwhitehead:mYp4ssw0rd@localhost:5672/sports/football
```

Kanaldaki MCAUSER ayarı

AMQP kanallarının bir MCAUSER özniteliği vardır; bu öznitelik, söz konusu kanalla kurulan tüm bağlantıların yetkili olduğu IBM MQ kullanıcı kimliğini ayarlamak için kullanılabilir. AMQP istemcilerinden o kanala yapılan tüm bağlantılar, yapılandırdığınız MCAUSER kimliğini benimser. Bu kullanıcı kimliği, farklı konularda ileti sistemi yetkisi için kullanılır.

Kuyruk yöneticilerine güvenli bağlantılar sağlamak için kanal kimlik doğrulamasını (CHLAUTH) kullanmanız önerilir. Kanal kimlik doğrulamasını kullanıyorsanız, MCAUSER değerini ayrıcalıklı olmayan bir kullanıcıya yapılandırmanız önerilir. Bu, bir kanal bağlantısı bir CHLAUTH kuralıyla eşleşmezse, bağlantının kuyruk yöneticisinde herhangi bir ileti alışverişi gerçekleştirme yetkisi olmamasını sağlar.

Not: Windows Windowsişletim sisteminde, IBM MQ 9.2' den önce, MCAUSER kullanıcı kimliği ayarı yalnızca 12 karaktere kadar olan kullanıcı kimlikleri için desteklenir. IBM MQ 9.2 Long Term Supportolanağından, 12 karakter sınırı kaldırılır.

SSL/TLS desteği

AMQP kanalları, kuyruk yöneticiniz için yapılandırılan anahtar havuzundaki anahtarları kullanarak SSL/TLS şifrelemesini destekler. SSL/TLS şifrelemesi için AMQP kanal yapılandırma seçenekleri, diğer MQ kanalı tipleriyle aynı seçenekleri destekler; bir şifre belirtimi belirtebilir ve kuyruk yöneticisinin AMQP istemci bağlantılarından sertifikalar gerektirip gerektirmediğini belirleyebilirsiniz.

Kuyruk yöneticisinin FIPS özniteliklerini kullanarak, AMQP istemcilerinden gelen bağlantıların güvenliğini sağlamak için kullanabileceğiniz SSL/TLS şifreleme takımlarını denetleyebilirsiniz.

Kuyruk yöneticisi için anahtar havuzu ayarlama hakkında bilgi için bkz. “AIX, Linux, and Windows üzerinde SSL/TLS ile çalışma” sayfa 289.

Bir AMQP istemci bağlantısı için SSL/TLS desteğinin nasıl yapılandırılacağına ilişkin bilgi için AMQP kanalları oluşturma ve kullanmabaşlıklı konuya bakın.

Java Kimlik Doğrulama ve Yetkilendirme Hizmeti (JAAS)

İsteğe bağlı olarak AMQP kanallarını, bir AMQP istemcisi tarafından sağlanan kullanıcı adını ve parolayı denetleyebilen bir JAAS oturum açma modülüyle yapılandırabilirsiniz. Bkz. “JAAS ' ı AMQP kanalları için yapılandırma” sayfa 607.

İlgili görevler

AMQP istemci uygulamalarının geliştirilmesi

AMQP kanallarının oluşturulması ve kullanılması

ALW AMQP istemcisi devralma kısıtlanıyor

Var olan AMQP istemci bağlantısıyla aynı istemci tanıtıcısına sahip bir AMQP istemci bağlantısı kurulduğunda, varsayılan olarak var olan istemci bağlantısı kesilir. Ancak, kuyruk yöneticisini istemci devralma davranışını, yalnızca belirli ölçütler karşılandığında devralınabilecek şekilde kısıtlayacak şekilde yapılandırabilirsiniz.

Örneğin, farklı ekipler tarafından geliştirilen AMQP uygulamaları varsa ve aynı istemci tanıtıcısını kullanıyorlarsa, var olan istemci bağlantısının kesilmesi uygun olmayabilir. Bu sorunu ele almak için, kullanılmakta olan AMQP kanalının adına, istemcinin IP adresine ve istemci kullanıcı kimliğine (SASL kimlik doğrulaması etkinleştirildiğinde) dayalı olarak istemci devralma özelliğini kısıtlayabilirsiniz.

Aşağıdaki çizelgede ayrıntılı olarak açıklandığı gibi, gereken istemci devralma kısıtlaması düzeyini belirtmek için kuyruk yöneticisi öznitelikleri **AdoptNewMCA** ve **AdoptNewMCACheck** ayarlarını kullanın:

*Çizelge 103. İstemci devralma işlemini kısıtlamak için **AdoptNewMCA** ve **AdoptNewMCACheck** ayarları*		
AdoptNewMCA	**AdoptNewMCACheck**	**İstemci devralma işlemine izin verilmeden önce denetlenen ölçütler**
NO ya da tanımsız	Geçerli değildir	Yok. Kimliği doğrulanan tüm istemci bağlantıları için istemci devralma işlemine izin verilir ve tüm CHLAUTH kurallarını geçirir.
ALL (ya da NO dışında bir değer)	QM ya da tanımsız	Yok. Kimliği doğrulanan tüm istemci bağlantıları için istemci devralma işlemine izin verilir ve tüm CHLAUTH kurallarını geçirir.
ALL (ya da NO dışında bir değer)	AD	Kullanıcı Kimliği (SASL etkinleştirildiğinde) Kanal adı
ALL (ya da NO dışında bir değer)	ADRES	Kullanıcı Kimliği (SASL etkinleştirildiğinde) IP adresi
ALL (ya da NO dışında bir değer)	TÜMÜ	Kullanıcı Kimliği (SASL etkinleştirildiğinde) Kanal adı IP adresi

Kuyruk yöneticisi öznitelikleri **AdoptNewMCA** ve **AdoptNewMCACheck** , KANAL kısmında tanımlanan kuyruk yöneticisi yapılanışının bir parçasıdır. IBM MQ for Windows ve IBM MQ for Linux x86-64 sistemlerinde, yapılandırma bilgilerini IBM MQ Explorerkullanarak değiştirin. Diğer sistemlerde, `qm.ini` yapılandırma dosyasını düzenleyerek bilgileri değiştirin. Kuyruk yöneticisi kanalları bilgilerinin nasıl değiştirileceğine ilişkin bilgi için Kanalların özniteliklericibaşlıklı konuya bakın.

İlgili görevler

AMQP istemci uygulamalarının geliştirilmesi

AMQP kanallarının oluşturulması ve kullanılması

ALW JAAS ' ı AMQP kanalları için yapılandırma

Java Authentication and Authorization Service (JAAS) özel modülleri, bağlandığında AMQP istemcisi tarafından bir AMQP kanalına iletilen kullanıcı adı ve parola kimlik bilgilerini doğrulamak için kullanılabilir.

Bu görev hakkında

Diğer Javatabanlı sistemlerde kimlik doğrulaması için JAAS modüllerini kullanıyorsanız ve MQile AMQP bağlantılarını doğrulamak için bu modülleri yeniden kullanmak istiyorsanız özel bir JAAS modülü kullanmak isteyebilirsiniz. Diğer bir seçenek olarak, MQ ' da yerleşik kimlik doğrulama özellikleri kullanmak istediğiniz kimlik doğrulama mekanizmasını desteklemiyorsa özel bir JAAS modülü yazmak isteyebilirsiniz.

AMQP kanalları için JAAS birimlerinin yapılandırılması, kuyruk yöneticisi düzeyinde gerçekleştirilir. Bu, kuyruk yöneticisine AMQP bağlantılarını doğrulamak için bir JAAS modülü yapılandırırsanız, modülün tüm AMQP kanallarına uygulanacağı anlamına gelir. JAAS modülünü çağıran kanalın adı modüle geçirilerek farklı kanallar için farklı JAAS oturum açma davranışını kodlayın.

Diğer bilgiler JAAS modülüne de geçirilir:

- Kimlik doğrulamayı deneyen AMQP istemcisinin istemci tanıtıcısı.
- AMQP istemcisinin ağ adresi.
- JAAS modülünü çağıran kanalın adı.

Yordam

Aşağıdaki adımları tamamlayarak AMQP kanalları için bir JAAS yapılandırma modülü yapılandırırsınız:

1. Bir ya da daha çok JAAS modülü yapılandırma kısmı içeren bir `jaas.config` dosyası tanımlayın. Kıta, JAAS `javax.security.auth.spi.LoginModule` arabirimini gerçekleştiren Java sınıfının tam olarak nitelenmiş adını belirtmelidir.
   - Varsayılan bir `jaas.config` kütüğü ürünle birlikte gönderilir ve *QM_data_directory*`/amqp/jaas.config`dizininde bulunur.
   - `MQXRConfig` adlı önceden yapılandırılmış bir kısmı varsayılan `jaas.config` dosyasında zaten tanımlı.
2. AMQP kanalları için kullanılacak kıta adını belirtin.
   - Linux AIX `amqp_unix.properties` dosyasına bir özellik ekleyin.
   - Windows `amqp_win.properties` dosyasına bir özellik ekleyin.

   Özellik şu formu içerir:

   ```
   com.ibm.mq.MQXR.JAASConfig=JAAS_stanza_name
   ```

   Örneğin:

   ```
   com.ibm.mq.MQXR.JAASConfig=MQXRConfig
   ```
3. Kuyruk yöneticisi ortamını özel modülün sınıfını içerecek şekilde yapılandırın. AMQP hizmetinin, JAAS yapılandırma alanında yapılandırılan Java sınıfına erişimi olmalıdır.

   Bunu, yolu JAAS sınıfına MQ `service.env` kütüğüne ekleyerek yaparsınız. CLASSPATH değişkenini JAAS birim sınıfının yerine ayarlamak için MQ yapılanış dizinindeki (*MQ_config_directory*) ya da kuyruk yöneticisi yapılanış dizinindeki (*QM_config_directory*) `service.env` dosyasını düzenleyin.

Sonraki adım

mq_installation_directory`/amqp/samples` dizininde ürünle birlikte örnek bir JAAS oturum açma modülü gönderilir. Örnek JAAS oturum açma modülü, istemcinin bağlandığı kullanıcı adı ya da paroladan bağımsız olarak tüm istemci bağlantılarını doğrular.

Örneğin kaynak kodunu değiştirebilir ve yeniden derleyerek yalnızca belirli bir parolaya sahip belirli kullanıcıların kimliğini doğrulamasını deneyebilirsiniz. UNIX sisteminde AMQP kanalını ürünle birlikte gönderilen örnek JAAS oturum açma modülünü kullanacak şekilde yapılandırmak için:

1. `/var/mqm/qmgrs/QMNAME/amqp/amqp_unix.properties` dosyasını düzenleyin ve `com.ibm.mq.MQXR.JAASConfig=MQXRConfig`özelliğini ayarlayın.
2. /var/mqm/service.env dosyasını düzenleyin ve CLASSPATH=*mq_installation_location*`/amqp/samples` özelliğini ayarlayın

`jaas.config` dosyası, oturum açma modülü sınıfı olarak `samples.JAASLoginModule` örnek sınıfını belirten `MQXRConfig` adlı bir bölümü zaten içeriyor. Örnek modülü denemeden önce `jaas.config` üzerinde herhangi bir değişiklik yapılması gerekmez.

İlgili görevler

AMQP istemci uygulamalarının geliştirilmesi

AMQP kanallarının oluşturulması ve kullanılması

Advanced Message Security

Advanced Message Security (AMS), son uygulamaları etkilemediği halde IBM MQ ağı üzerinden akan hassas veriler için yüksek düzeyde koruma sağlayan bir IBM MQ bileşenidir.

Advanced Message Security ürününe genel bakış

IBM MQ uygulamaları, yüksek değerli finansal işlemler ve kişisel bilgiler gibi hassas verileri, bir açık anahtar şifreleme modeli kullanarak farklı koruma düzeyleriyle göndermek için Advanced Message Security ' u kullanabilir.

İlgili kavramlar

"Message Channel Agent (MCA) Müdahalesi ve AMS" sayfa 658

MCA kesilmesi, IBM MQ altında çalışan bir kuyruk yöneticisinin, ilkelerin sunucu bağlantısı kanalları için uygulanmasını seçmeli olarak etkinleştirmesini sağlar.

İlgili başvurular

AMS iletilerinde kullanılan IBM Global Security Kit (GSKit) dönüş kodları

Advanced Message Security ürününün özellikleri ve işlevleri

Advanced Message Security , ileti düzeyinde veri imzalama ve şifreleme sağlamak için IBM MQ güvenlik hizmetlerini genişletir. Genişletilmiş hizmetler, ileti verilerinin başlangıçta bir kuyruğa yerleştirildiği zaman ile alındığı zaman arasında değiştirilmediğini garanti eder. Buna ek olarak AMS , ileti verilerini gönderen bir kullanıcının imzalı iletileri hedef kuyruğa yerleştirme yetkisinin olduğunu doğrular.

AMS aşağıdaki işlevleri sağlar:

- IBM MQtarafından işlenen hassas ya da yüksek değerli işlemlerin güvenliğini sağlar.
- Alan bir uygulama tarafından işlenmeden önce, sahte ya da yetkisiz iletileri algılar ve kaldırır.
- Kuyruktan kuyruğa taşıma sırasında iletilerin değiştirilmediğini doğrular.
- Verileri yalnızca ağ boyunca değil, aynı zamanda bir kuyruğa yerleştirildiğinde de korur.
- IBM MQiçin var olan patentli ve müşteri tarafından yazılan uygulamaların güvenliğini sağlar.
- z/OS IBM MQ 9.1.3içinden IBM MQ for z/OS , isteğe bağlı olarak, ağda akan iletilerden AMS korumasını kaldırma ve ekleme yeteneği sağlar. Bu, *Sunucudan Sunucuya İleti Kanal Aracısı (MCA) Aracısı*olarak bilinir.
- ALW IBM MQ 9.1.4 ve IBM MQ 9.1.0 Fix Pack 4işletim sistemlerinde, müşterinin uygulama programı içinde çalışan IBM MQ kitaplık koduna bir denetim eklenir. Denetim, *AMQ_AMS_FIPS_OFF* ortam değişkeninin değerini okumak için kullanıma hazırlamada erken çalışır ve herhangi bir değere ayarlanırsa, IBM Global Security Kit (GSKit) kodu o uygulamada FIPS dışı kipte çalıştırılır.

AMS ile sağlanan koruma nitelikleri

Advanced Message Security, `Integrity`, `Privacy`ve `Confidentiality`için üç koruma özelliği vardır.

`Integrity` koruması, iletiyi kimin oluşturduğunu ve iletinin değiştirilmediğini ya da değiştirilmediğini garanti eden dijital imzalama ile sağlanır.

`Privacy` koruması, dijital imzalama ve şifreleme birleşimiyle sağlanır. Şifreleme, ileti verilerinin yalnızca istenen alıcı ya da alıcılar tarafından görüntülenebilmesini sağlar. Yetkisiz alıcılar şifrelenmiş ileti verilerinin bir kopyasını alsalar bile, gerçek ileti verilerini görüntüleyemezler.

`Confidentiality` koruması yalnızca isteğe bağlı anahtar yeniden kullanımıyla şifrelemeyle sağlanır.

Performans üzerindeki etkisi

AMS , dijital imzalama ve şifreleme sağlamak için simetrik ve asimetrik şifreleme yordamlarının bir birleşimini kullanır. Simetrik anahtar işlemleri, CPU yoğunluğu olan asimetrik anahtar işlemleriyle

karşılaştırıldığında çok hızlı olduğu için, bu da AMSile çok sayıda iletiyi koruma maliyetleri üzerinde önemli bir etkiye sahip olabilir.

Asimetrik şifreleme yordamları
Örneğin, imzalı bir ileti konurken, ileti hash değeri asimetrik bir anahtar işlemi kullanılarak imzalanır.

İmzalı bir ileti alınırken, imzalı hash 'i doğrulamak için başka bir asimetrik anahtar işlemi kullanılır.

Bu nedenle, ileti verilerini imzalamak ve doğrulamak için ileti başına en az iki asimetrik anahtar işlemi gerekir.

Asimetrik ve simetrik şifreleme rutinleri
Şifrelenmiş bir ileti yerleştirilirken, bir simetrik anahtar oluşturulur ve daha sonra, iletinin amaçlanan her alıcısı için asimetrik bir anahtar işlemi kullanılarak şifrelenir.

İleti verileri daha sonra simetrik anahtarla şifrelenir. Şifreli ileti alınırken, hedeflenen alıcının ileti için kullanılan simetrik anahtarı keşfetmek için asimetrik bir anahtar işlemi kullanması gerekir.

Bu nedenle, koruma niteliklerinin üçü de, CPU yoğunluklu asimetrik anahtar işlemlerinin çeşitli unsurlarını içerir ve bu, uygulamalar için ileti koyma ve alma için maksimum ulaşılabilir mesajlaşma oranını önemli ölçüde etkileyecektir.

Ancak `Confidentiality` ilkeleri, bir ileti dizisi üzerinde simetrik anahtarın yeniden kullanılmasına izin verir. Simetrik anahtar yeniden kullanımı yoluyla `Confidentiality` ilkeleriyle önemli CPU maliyet tasarrufları yapılabilir. Bu işlem kipi, simetrik şifreleme anahtarını paylaşmak için PKCS#7 biçimini kullanmaya devam eder. Ancak, ileti başına asimetrik anahtar işlemlerinden bazılarını ortadan kaldıran dijital imza yoktur. Simetrik anahtarın yine de her alıcı için asimetrik anahtar işlemleriyle şifrelenmesi gerekir, ancak simetrik anahtar isteğe bağlı olarak aynı alıcılar için yönlendirilen birden çok ileti üzerinden yeniden kullanılabilir. İlke tarafından anahtarın yeniden kullanılmasına izin veriliyorsa, yalnızca ilk ileti asimetrik anahtar işlemleri gerektirir. Sonraki iletilerin yalnızca simetrik anahtar işlemlerini kullanması gerekir.

Anahtar yeniden kullanımı

`Confidentiality` ilkeleriyle, aynı kuyruğa konan ve aynı alıcıya ya da alıcıya yönelik bir dizi iletiyi şifrelemeye ilişkin maliyetleri önemli ölçüde azaltmak için simetrik anahtar yeniden kullanım yaklaşımını kullanabilirsiniz.

Örneğin, aynı alıcı kümesine 10 şifrelenmiş ileti koyarken, bir simetrik anahtar oluşturulur ve daha sonra ilk ileti için şifrelenir, iletinin amaçlanan her alıcısı için asimetrik bir anahtar işlemi kullanılır.

İlke denetimli sınırlara dayalı olarak, şifrelenmiş simetrik anahtar daha sonra aynı alıcılar için tasarlanan sonraki iletiler tarafından yeniden kullanılabilir. Simetrik anahtarın sonraki iletiler tarafından yeniden kullanılmasına izin vermek için, uygulamanın kuyruğa bir ileti koyduktan sonra kuyruğu açık tutması gerekir. Simetrik anahtar MQPUT1 işlemleri tarafından yeniden kullanılamaz. Şifrelenmiş iletileri alan bir uygulama, simetrik bir anahtarın ne zaman değişmediğini saptayabilmesi ve simetrik anahtarı alma harcamasından kaçınması için aynı optimizasyonu uygulayabilir.

Bu örnekte asimetrik anahtar işlemlerinin %90 'ı, aynı anahtar yeniden kullanılarak hem koyma hem de alma uygulamaları tarafından önlenebilir.

Anahtarın yeniden kullanılmasıyla ilgili daha fazla bilgi için aşağıdaki başlara bakın:

- MQSC komutu SET POLICY
- setmqspl denetim komutu
- IBM i IBM i komut SETMQMSPL

AMS içindeki temel kavramlar

Aracın nasıl çalıştığını ve etkin bir şekilde nasıl yönetileceğini anlamak için Advanced Message Security içindeki temel kavramlar hakkında bilgi edinin.

Genel anahtar altyapısı ve Advanced Message Security

Açık anahtar altyapısı (PKI), güvenli iletişim elde etmek için açık anahtar şifrelemesi kullanımını destekleyen bir tesis, politika ve hizmet sistemidir.

Bir genel anahtar altyapısının bileşenlerini tanımlayan tek bir standart yoktur, ancak PKI genellikle açık anahtar sertifikalarının kullanımını içerir ve aşağıdaki hizmetleri sağlayan sertifika yetkililerini (CA) ve diğer kayıt yetkililerini (RA) içerir:

- Dijital sertifikaları verme
- Dijital sertifikaların geçerliliği denetleniyor
- Dijital sertifikaları iptal etme
- Sertifikaları dağıtma

Kullanıcıların ve uygulamaların kimliği, imzalanmış ya da şifrelenmiş iletilerle ilişkilendirilmiş bir sertifikada **ayırt edici ad (DN)** alanıyla gösterilir. Advanced Message Security , bir kullanıcıyı ya da uygulamayı göstermek için bu kimliği kullanır. Bu kimliği doğrulamak için, kullanıcı ya da uygulamanın sertifikanın ve ilişkili özel anahtarın saklandığı anahtar deposuna erişimi olmalıdır. Her sertifika, anahtar deposundaki bir etiketle gösterilir.

İlgili kavramlar

“Anahtar depolarının ve sertifikaların AMS ile kullanılması” sayfa 652

IBM MQ uygulamalarına şeffaf şifreleme koruması sağlamak için Advanced Message Security , genel anahtar sertifikalarının ve özel anahtarın saklandığı anahtar deposu dosyasını kullanır. z/OSüzerinde, anahtar deposu dosyası yerine SAF anahtar halkası kullanılır.

AMS içinde dijital sertifikalar

Advanced Message Security , kullanıcıları ve uygulamaları X.509 standart dijital sertifikalarıyla ilişkilendirir. X.509 sertifikaları genellikle güvenilir sertifika yetkilisi (CA) tarafından imzalanır ve şifreleme ve şifre çözme için kullanılan özel ve genel anahtarları içerir.

Dijital sertifikalar, sahibin bir birey, kuyruk yöneticisi ya da başka bir varlık olması fark etmez, genel anahtarı sahibine bağlayarak kimliğine bürünmeye karşı koruma sağlar. Dijital sertifikalar genel anahtar sertifikaları olarak da bilinir, çünkü asimetrik bir anahtar şeması kullandığınızda açık anahtarın sahipliği konusunda size güvence verirler. Bu şema, bir uygulama için genel anahtar ve özel anahtar oluşturulmasını gerektirir. Genel anahtarla şifrelenen verilerin şifresi yalnızca karşılık gelen özel anahtar kullanılarak çözülebilirken, özel anahtarla şifrelenen verilerin şifresi yalnızca ilgili genel anahtar kullanılarak çözülebilir. Özel anahtar, parola korumalı bir anahtar veritabanı dosyasında saklanır. Yalnızca sahibinin, karşılık gelen genel anahtar kullanılarak şifrelenen iletilerin şifresini çözmek için kullanılan özel anahtara erişimi vardır.

Ortak anahtarlar doğrudan sahipleri tarafından başka bir varlığa gönderilirse, iletinin engellenebilme ve genel anahtarın başka bir varlıkla değiştirilme riski vardır. Bu "ortadaki adam" saldırısı olarak bilinir. Çözüm, genel anahtarın iletişim kurmakta olduğunuz varlığa ait olduğuna dair kullanıcıya güçlü bir güvence vererek, güvenilir bir üçüncü kişi aracılığıyla açık anahtarların değiştirilmesini sağlamaktır. Genel anahtarınızı doğrudan göndermek yerine, güvenilir bir üçüncü kişiden bunu dijital bir sertifikaya dahil etmesini isteyin. Dijital sertifikaları veren güvenilir üçüncü kişiye sertifika yetkilisi (CA) denir.

Dijital sertifikalar hakkında daha fazla bilgi için bkz. Dijital sertifikada ne var.

Sayısal sertifika, bir varlığın genel anahtarını içerir ve genel anahtarın bu varlığa ait olduğunu belirtir:

- Bir sertifika tek bir varlığa ilişkin olduğunda, buna *kişisel sertifika* ya da *kullanıcı sertifikası*denir.
- Bir sertifika bir sertifika yetkilisine ilişkin olduğunda, sertifikaya *CA sertifikası* ya da *imzalayıcı sertifikası*adı verilir.

Not: Advanced Message Security , hem Java hem de yerel uygulamalarda kendinden imzalı sertifikaları destekler

İlgili kavramlar

“Şifreleme” sayfa 11

Şifreleme, *düz metin*adı verilen okunabilir metin ile *şifreli metin*adı verilen okunamayan bir form arasında dönüştürme işlecidir.

Multi *Nesne yetkisi yöneticisi ve AMS*

Multiplatforms üzerinde, Object Authority Manager (OAM), IBM MQ ürünleriyle verilen yetkilendirme hizmeti bileşenidir.

Advanced Message Security varlıklarına erişim, IBM MQ kullanıcı grupları ve OAM aracılığıyla denetlenir. Yöneticiler, yetkileri gerektiği şekilde vermek ya da iptal etmek için komut satırı arabirimini kullanabilir. Farklı kullanıcı grupları, aynı nesneler için farklı türde erişim yetkisine sahip olabilir. Örneğin, bir grup belirli bir kuyruk için hem PUT hem de GET işlemlerini gerçekleştirebilirken, başka bir gruba yalnızca kuyruğa göz atma izni verilebilir. Benzer şekilde, bazı grupların bir kuyruk için GET ve PUT yetkileri olabilir, ancak kuyruğu değiştirmelerine ya da silmelerine izin verilmez.

OAM aracılığıyla aşağıdakileri denetleyebilirsiniz:

- Advanced Message Security nesnelerine İleti Kuyruğu Arabirimi (MQI) aracılığıyla erişim. Bir uygulama programı nesnelere erişmeye çalıştığında, OAM, isteği yapan kullanıcı tanıtımının istenen işleme ilişkin yetkiye sahip olup olmadığını denetler. Bu, kuyrukların ve kuyruklardaki iletilerin yetkisiz erişimden korunabileceği anlamına gelir.
- PCF ve MQSC komutlarını kullanma izni.

İlgili kavramlar

Nesne yetki yöneticisi

İleti Kuyruğu Arabirimine Genel Bakış

Advanced Message Security tarafından desteklenen teknoloji

Advanced Message Security , bir güvenlik altyapısı sağlamak için çeşitli teknoloji bileşenlerine bağlıdır.

Advanced Message Security , aşağıdaki IBM MQ uygulama programlama arabirimlerini (API ' ler) destekler:

- İleti kuyruğu arabirimi (MQI)
- IBM MQ Java Message Service (JMS) 1.0.2 ve 1.1.
- IBM MQ Temel Sınıflar- Java
- Yönetilmeyen kipte. Net için IBM MQ sınıfları

Not: Advanced Message Security , X.509 uyumlu sertifika yetkililerini destekler.

Bilinen AMS sınırlamaları

Advanced Message Securityiçin desteklenmeyen ya da sınırlamaları olan birçok IBM MQ seçeneği vardır.

- Aşağıdaki IBM MQ seçenekleri desteklenmez ya da sınırlamaları vardır:

  Yayınla/abone ol

  Bir yayınlama/abone olma ileti modelinin noktadan noktaya iletişim üzerinden başlıca avantajlarından biri, verilerin gönderilmesi ve alınması için gönderen ve alan uygulamaların birbirleri hakkında hiçbir şey bilmesine gerek olmamasıdır. Bu avantaj, amaçlanan alıcıları ya da yetkili imzalayıcıları tanımlaması gereken Advanced Message Security ilkelerinin kullanımıyla olumsuzdur. Bir uygulamanın bir ilke tarafından korunan bir diğer ad kuyruğu tanımı aracılığıyla bir konuya yayınlanması mümkündür; abone olan bir uygulamanın ilke korumalı bir kuyruktan ileti alabilmesi de mümkündür. Bir ilke doğrudan bir konu dizgisine atanamaz, ilkeler yalnızca kuyruk tanımlamalarına atanabilir.

  Kanal verilerini dönüştürme

  Advanced Message Security korumalı bir iletinin korunan bilgi yükü ikili biçim kullanılarak iletilir; bu, uygulamalar arasında bir kanalda veri dönüştürmenin ileti özetini geçersiz kılmamasını sağlar. İletileri ilke korumalı bir kuyruktan alan uygulamalar veri dönüştürme isteğinde bulunmalıdır; korunan bilgi yükünü dönüştürme işlemi, iletiler başarıyla doğrulandıktan ve korumasız kaldıktan sonra denenir.

Dağıtım listeleri
Listedeki her hedef kuyruğun aynı ilkeye sahip olması koşuluyla, dağıtım listelerine ileti koyan uygulamalar korunurken Advanced Message Security ilkeleri kullanılabilir. Uygulama bir dağıtım listesini açtığında tutarsız ilkeler saptanırsa, açma işlemi başarısız olur ve uygulamaya bir güvenlik hatası döndürülür.

Uygulama iletisi bölümlemesi
İlke korumalı iletilerin boyutu artar ve uygulamaların bir iletinin bölüm sınırlarını doğru bir şekilde belirtmesi mümkün değildir.

Yönetilen kipte IBM MQ classes for .NET kullanan uygulamalar (istemci bağlantıları)
Yönetilen kipte (istemci bağlantıları) IBM MQ classes for .NET kullanan uygulamalar desteklenmez.

Not: MCA kesmesi, desteklenmeyen istemcilerin AMS' i kullanmasına izin vermek için kullanılabilir.

Yönetilen kipte .NET (XMS) uygulamaları için İleti Hizmeti istemcisi
Yönetilen kipteki .NET (XMS) uygulamaları için İleti Hizmeti istemcisi desteklenmez.

Not: MCA kesmesi, desteklenmeyen istemcilerin AMS kullanmasına izin vermek için kullanılabilir.

IMS köprüsü tarafından işlenen IBM MQ kuyrukları
IMS köprüsü tarafından işlenen IBM MQ kuyrukları desteklenmez.

Not: AMS , CICS köprü kuyruklarında desteklenir. CICS köprü kuyruklarında MQPUT (şifreleme) ve MQGET (şifre çözme) için aynı kullanıcı kimliğini kullanmalısınız.

Bekleyen alıcıya koy
Alıcı bekleme yöntemine koyma, AMS ilkeleri tanımlanmış kuyruklara karşı alıcı uygulamaları için desteklenmez.

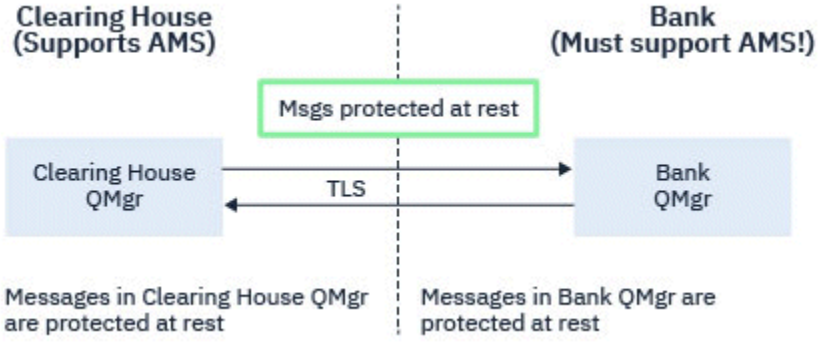
Sunucudan sunucuya MCA kesmesi
IBM MQ for z/OS 9.1.3' den sunucuya MCA kesmesi yalnızca gönderen, sunucu, alıcı ve istekte bulunan kanal tipleri için desteklenir.

- Bir iletiyi korurken hangi sertifikanın kullanılacağı belirlenmediği için, kullanıcılar aynı Ayırt Edici Ada sahip birden fazla sertifikayı tek bir anahtar deposu dosyasına koymaktan kaçınmalıdır.
- **WMQ_PROVIDER_VERSION** özelliği 6 olarak ayarlanırsa, JMS ' da AMS desteklenmez.
- AMS kesici AMQP ya da MQTT kanalları için desteklenmez.

z/OS Advanced Message Security ileti kanallarında araya girme

z/OSişletim sisteminde Advanced Message Security (AMS), gönderene, sunucuya, alıcıya ve istekte bulunan kanallara ek bir güvenlik ilkesi koruması (SPLPROT) seçeneği sağlayarak AMS 'u desteklemenize ve AMS' i desteklemeyen çözüm ortaklarıyla iletişim kurmanıza olanak tanır.

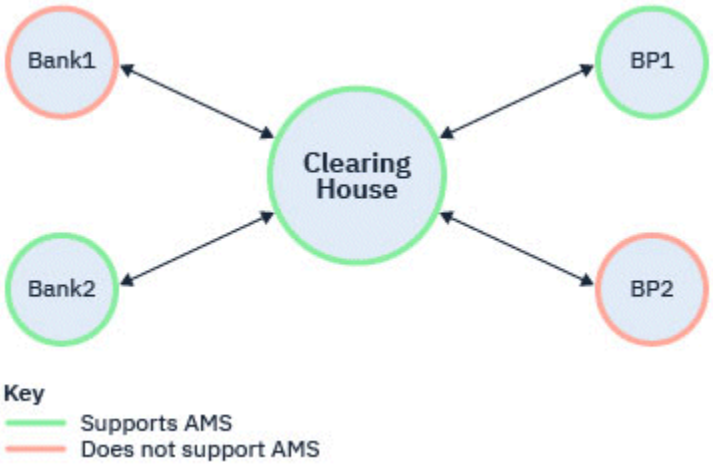
Şekil 1 , bir banka ile iletişim kuran bir temizleme evi örneğine göre, AMS müdahalesi olmadan, sistemin her iki tarafının da desteklemesi gerektiğini gösterir AMS.



Şekil 32. AMS ' nin AMS müdahalesi olmadan kullanımı

AMS Interception seçeneğinin temel avantajlarından biri, kuruluşunuz AMS yapılandırmışsa ve tüm çözüm ortaklarınız desteklemiyorsa AMS, giden iletilere karşı korumayı kaldırabilir ve kanallardaki gelen iletileri AMS' u desteklemeyen çözüm ortaklarından koruyabilirsiniz.

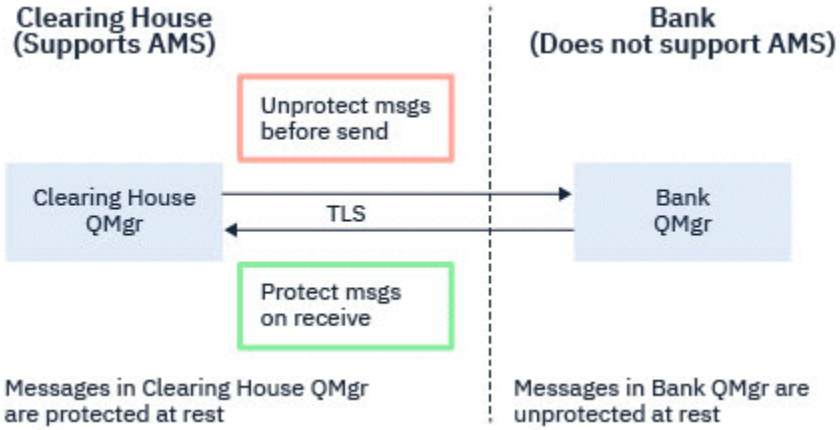
Bu senaryo, temizleme evi ve bankaları örneği kullanılarak Şekil 2'de gösterilir; burada, bazı kurumların AMS' e sahip olduğu ve diğerlerinin sahip olmadığı temizleme evi, bankalar ve çözüm ortakları arasında bir ileti akışı vardır.



Şekil 33. Bazı ortaklar AMS ' yi destekler, bazıları ise desteklemez

Genellikle kanallar TLS etkindir.

Ancak, bazı bankaların ve çözüm ortaklarının AMS' i desteklemediği ve tüm bankalar ve çözüm ortakları arasında ileti alışverişi yapma zorunluluğu olduğu bir durum olabilir. Bu senaryo Şekil 3 içinde gösterilmektedir



Şekil 34. Çözüm ortakları arasındaki ileti akışı

İlgili görevler

Sunucudan sunucuya ileti kanalı engelleme örneği yapılandırmaları

z/OS *Sunucudan sunucuya ileti kanallarında AMS kesmesi*

Sunucudan sunucuya ileti kanalı kesmesi, iletilere uygulanabilir Advanced Message Security (AMS) ilkelerinin uygulanıp uygulanmayacağını, gönderen tipi ileti kanalı aracılarının iletim kuyruklarından ileti alıp almadığını ve alıcı tipi ileti kanalı aracılarının hedef kuyruklara ileti gönderip göndermediklerini denetlemek için bir yol sağlar.

Bu, AMS özelliği etkinleştirilmemiş bir kuyruk yöneticisiyle iletişim kurarken, gönderen, sunucu, alıcı ve istekte bulunan sunucular arasında ileti kanallarını kullanarak AMS korumasının bir kuyruk yöneticisinde etkinleştirilmesini sağlar.

Yani, AMS etkin kuyruk yöneticilerindeki AMS korumalı iletiler,AMS etkin olmayan kuyruk yöneticilerine gönderilmeden önce korumasız olabilir veAMS etkin olmayan kuyruk yöneticilerinden alınan korunmayan iletiler, uygulanabilir AMS ilkeleriyle AMS etkin kuyruk yöneticileriyle korunabilir.

Sunucudan sunucuya ileti kanalı alıkonması yapılandırılıyor

Sunucudan sunucuya ileti kanalı kesmesi, kanal tipi gönderen, sunucu, alıcı ya da istekte bulunan kanallarda SPLPROT özniteliğiyle yapılandırılır. Davranışı yapılandırmak için kullanılabilecek seçenekler, belirtilen kanal tipine bağlıdır:

Passthru
Bu kanal için ileti kanalı aracısı tarafından gönderilen ya da alınan iletileri geçirin, değiştirmeden.

This value is valid for channels with a channel type (**CHLTYPE**) of SDR, SVR, RCVR, or RQSTR, and is the default value.

KALDIR
İleti kanalı aracısı tarafından iletim kuyruğundan alınan iletilerden AMS korumasını kaldırın ve iletileri iş ortağına gönderin.

İleti kanalı aracısı iletim kuyruğundan bir ileti aldığında, iletim kuyruğu için bir AMS ilkesi tanımlanmışsa, iletiyi kanal üzerinden göndermeden önce iletiden AMS korumasını kaldırmak için bu uygulanır. İletim kuyruğu için bir AMS ilkesi tanımlanmamışsa, ileti olduğu gibi gönderilir.

Bu değer yalnızca SDR ya da SVRkanal tipine sahip kanallar için geçerlidir.

AASİLKE
Hedef kuyruk için tanımlanan ilkeye dayalı olarak, hedef kuyruğa yerleştirmeden önce gelen iletilere AMS korumasını uygulayın.

İleti kanalı aracısı bir gelen iletisi aldığında, hedef kuyruk için bir AMS ilkesi tanımlanmışsa, AMS hedef kuyruğa konan iletiden önce iletiye koruma uygulanır. Hedef kuyruk için bir AMS ilkesi tanımlanmamışsa, ileti hedef kuyruğa olduğu gibi yerleştirilsın " bir yorsanız.

Bu değer yalnızca RCVR ya da RQSTRkanal tipine sahip kanallar için geçerlidir.

İleti kanalı müdahalesi için kullanıcı kimliği

Sunucudan sunucuya ileti kanalı alımıyla kullanılan kullanıcı kimlikleri gereksinimi, var olan AMS etkinleştirilmiş uygulamalarla aynıdır. Çalışan bir kanal için, gönderen ileti kanalı aracısı iletileri bir iletim kuyruğundan alır ve alan ileti kanalı aracısı iletileri hedef kuyruklara koyar. Sunucuda sunucu kanallarına ayarlanan ileti kanalı aracısı kullanıcı kimliği (MCAUSER) alanı, ileti kanalı aracılarının koyma ve alma isteklerini gerçekleştirdikleri kullanıcı kimliğini tanımlar.

Sunucudan sunucuya ileti kanalı alımıyla, diğer AMS etkin uygulamalarda olduğu gibi, alma ve koyma istekleri sırasında AMS işlevleri gerçekleştirilir. Bu nedenle, ileti kanalı aracısı kullanıcı kimlikleri, AMS uygulaması kullanıcı kimlikleriyle aynı gereksinimlere sahiptir.

Koyma ve alma işlemini gerçekleştirmek için kullanılan MCAUSER yapılandırılabilir ve bunun giden ya da gelen bir kanal olmasına bağlıdır. Seçilen kullanıcı kimliğinin ileti kanalı aracısında nasıl işlem gerçekleştirdiğine ilişkin ayrıntılar için bkz. MCAUSER . Bu nedenle, kanal başlatıcısının altında çalıştığı kullanıcı kimliği, sunucudan sunucuya ileti kanalı kesmesi sırasında gerçekleştirilen AMS işlevleri için kullanılacak kullanıcı kimliğidir. Bu nedenle, bu kullanıcı kimlikleri AMS uygulaması kullanıcı kimlikleriyle aynı gereksinimlere sahiptir.

Kimlik doğrulama, PUTAUT yapılandırmasına sahip kanallar için ayrıntılı olarak açıklanan kanal için var olan kurallar kullanılarak gerçekleştirilir. Ek bilgi için kanal başlatıcı tarafından kullanılan kullanıcı kimlikleri konusuna bakın.

Not: Sunucudan sunucuya ileti kanalı kesmesi, PUTAUT kanal özniteliğinin değerini dikkate almıyor.

İleti boyutu ve MAXMSGL

AMS koruması nedeniyle, korunan iletilerin ileti boyutu özgün ileti boyutundan büyük olacaktır.

Korunan iletiler, korunmayan iletilerden büyük. Bu nedenle, hem kuyruklardaki hem de kanallardaki **MAXMSGL** özniteliğinin değerinin, korunan iletilerin boyutunu dikkate alacak şekilde değiştirilmesi gerekebilir.

İlgili başvurular

Sunucudan sunucuya ileti kanalı engelleme örneği yapılandırmaları

AMS için hata işleme

IBM MQ Advanced Message Security , korunmayan iletiler ya da hatalar içeren iletileri yönetmek için bir hata işleme kuyruğu tanımlar.

Kusurlu mesajlar istisnai durumlar olarak ele alınmaktadır. Alınan bir ileti, bulunduğu kuyruğa ilişkin güvenlik gereksinimlerini karşılamıyorsa, örneğin, ileti şifrelenmesi gerektiği zaman imzalandıysa ya da şifre çözme ya da imza doğrulaması başarısız olursa, ileti hata işleme kuyruğuna gönderilir. Aşağıdaki nedenlerden ötürü hata işleme kuyruğuna bir ileti gönderilebilir:

- Koruma kalitesi uyumsuzluğu-alınan ileti ile güvenlik ilkesinde QOP tanımı arasında bir koruma kalitesi (QOP) uyuşmazlığı var.
- Şifre çözme hatası-iletinin şifresi çözülemiyor.
- PDMQ üstbilgisi hatası- Advanced Message Security (AMS) ileti üstbilgisine erişilemiyor.
- Büyüklük uyuşmazlığı-şifre çözdükten sonra iletinin uzunluğu beklenenden farklı.
- Şifreleme algoritması gücü uyumsuzluğu-ileti şifreleme algoritması gerekenden daha zayıf.
- Bilinmeyen hata-beklenmeyen hata oluştu.

AMS , SYSTEM.PROTECTION.ERROR.QUEUE . IBM MQ AMS tarafından SYSTEM.PROTECTION.ERROR.QUEUE üstbilgisinden önce bir MQDLH üstbilgisi gelir.

IBM MQ yöneticiniz SYSTEM.PROTECTION.ERROR.QUEUE .

z/OS IBM MQ 9.1.3' dan IBM MQ for z/OSüzerinde, MCA (Message Channel Agent; İleti Kanal Aracısı) kullanımı kullanıyorsa:

- Daha önce belirtilen nedenlerden biri nedeniyle, IBM MQ AMS iletileri iletim kuyruğundan hata işleme kuyruğuna taşırsa, gönderen MCA, iletim kuyruğundaki sonraki kullanılabilir iletiyi işlemeye devam eder.
- Genel olarak, var olan kanal kuralları aşağıdakiler için geçerlidir:
  - İletileri Teslim Edilemeyen Mektup Kuyruğuna koyma, ve
  - Teslim Edilemeyen Mektup Kuyruğuna konursa yapılacak işlemler başarısız olur.

Belirli senaryolara ilişkin daha fazla bilgi için bkz. “z/OS üzerinde AMS için teslim edilmemiş iletiler” sayfa 616 .

z/OS *z/OS üzerinde AMS için teslim edilmemiş iletiler*

IBM MQ for z/OSüzerinde sunucudan sunucuya Message Channel Agent müdahalesi ile ilgili belirli senaryolar.

IBM MQ 9.1.3' dan IBM MQ for z/OSüzerinde, MCA (Message Channel Agent; İleti Kanal Aracısı) kullanımı kullanıyorsa:

- Bir iletiyi aldıktan ve korumayı kaldırdıktan sonra, örneğin, ileti kanal için çok büyük olduğu için, gönderen MCA bir iletiyi bir nedenden dolayı teslim edemezse, USEDLQ gönderen kanal özniteliği YES olarak ayarlanırsa, gönderen MCA iletiyi yerel DLQ ' ya (Ölü Harf Kuyruğu) taşır.

  SYSTEM.DEAD.LETTER.QUEUE yerel DLQ olarak kullanılıyor, ileti korumasız olarak yerleştirildi.

  Not: IBM MQ AMS , sistem kuyruklarına konan iletilerin korunmasını desteklemez.

  Yerel DLQ olarak adlandırılan bir DLQ kullanılıyorsa, adı belirtilen DLQ ile aynı adı taşıyan bir IBM MQ AMS ilkesi tanımladıysanız ve uygun bir ilke tanımlamadıysanız, ileti korunur.

- Bir ileti herhangi bir nedenle yerel DLQ ' ya konamazsa, kanalın NPMSPEED değeri NORMAL olarak ayarlanırsa ya da ileti kalıcı bir iletiyse, yürürlükteki ileti kümesi geriletiilir ve kanal RETRY durumuna getirilir. Tersi durumda, ileti atılır ve gönderen MCA, iletim kuyruğundaki sonraki iletiyi işlemeye devam eder.
- Güvenlik ilkelerinin SYSTEM.DEAD.LETTER.QUEUEya da “AMS içinde sistem kuyruğu koruması” sayfa 688içinde listelenen diğer SYSTEM kuyrukları üzerinde herhangi bir etkisi olmadığı göz önünde bulundurulursa, SYSTEM.DEAD.LETTER.QUEUE kullanıyorsa, MCA ' lar tarafından bu kuyruğa konan iletiler olduğu gibi yerleştirilir. Başka bir deyişle, iletiler daha önce korunmuşsa, korunmuş olarak yerleştirilirler; tersi durumda, korunmasız olarak yerleştirilirler.

  Kuyruk yöneticisi DEADQ özniteliği bir diğer (sistem dışı) gitmeyen iletiler kuyruğunun adına ayarlandıysa ve aynı ada sahip bir AMS ilkesi yoksa, bu kuyruğa MCA ' lar tarafından konan iletiler olduğu gibi yerleştirilir. Başka bir deyişle, iletiler daha önce korunmuşsa, korunmuş olarak yerleştirilirler; tersi durumda, korunmasız olarak yerleştirilirler.

  Kuyruk yöneticisi DEADQ özniteliği, diğer bir (sistem dışı) gitmeyen iletiler kuyruğunun adına ayarlandıysa ve DLQ ile aynı ada sahip bir AMS ilkesi varsa, ilke, bu kuyruğa MCA ' lar tarafından konan iletileri korumak için kullanılır. İleti önceden korunmuşsa, yeniden korunmaz; bu, çift korumayı önler. Aynı ada sahip bir AMS ilkesi yoksa, iletiler olduğu gibi yerleştirilir.
- setmqspl komutunda tolerate seçeneği kapalı ('**-t** O') olarak ayarlanmış DLQ ile ilgili bir ilke varsa, ileti AMS korumalı değilse ve bu nedenle PDMQ üstbilgisine sahip değilse, DLQ ' ya koyma başarısız olur. İleti, alıcıya PDMQ üstbilgisi olmadan ulaşırsa bu oluşur. Bu, iletinin özgün girişinin hedef için bir ilkesi olmadığı ve alıcının SPLPROT (ASPOLICY) ayarı olmadığı anlamına geliyor.
- DLQ için tanımlanan AMS ilkesi, iletiyi korumak için kanal başlatıcısının altında çalıştığı kullanıcı kimliğine izin vermiyorsa, bir MCA DLQ ' ya ileti koyamayabilir.
- Alıcı kanallar genellikle yerel DLQ ' ya teslim edilmemiş iletiler yerleştirirken, gönderen kanallar genellikle herhangi bir nedenle işlenemeyen iletiler yerleştirir; örneğin, kuyruk için çok büyük ileti ya da hatalı MQXQH üstbilgisi gibi.
- DLQ işleyicileri genellikle ileti bilgi yükünün kendisine değil, yalnızca DLQ üstbilgisine (DLH) bakar. Bu nedenle, ileti bilgi yükünün korunmuş olabileceği gerçeği, işleyicilerin iletinin DLQ ' ya neden yerleştirildiğini belirlemesini engellemez.
- DLQ tanımlanmamışsa, kanal:
  - Kalıcı bir ileti teslim edilemezse, olağan dışı sona erer (ve yeniden denenme durumuna geçer).
  - Kalıcı olmayan teslim edilmemiş bir iletiyi atar ve çalışmaya devam eder.

İlgili kavramlar

“AMS için hata işleme” sayfa 616

IBM MQ Advanced Message Security , korunmayan iletiler ya da hatalar içeren iletileri yönetmek için bir hata işleme kuyruğu tanımlar.

AMS için kullanıcı senaryoları

Advanced Message Securityile hangi iş hedeflerine ulaşabileceğinizi anlamak için olası senaryoları tanıyın.

Windows *Windows platformlarında AMS için Hızlı Başlangıç Kılavuzu*

Windows platformlarında ileti güvenliği sağlamak üzere Advanced Message Security (AMS) ürününü hızlı bir şekilde yapılandırmak için bu kılavuzu kullanın. Bu işlemi tamamladığınızda, kullanıcı kimliklerini doğrulamak için bir anahtar veritabanı ve kuyruk yöneticiniz için tanımlanmış imzalama/şifreleme ilkeleri yaratmış olursunuz.

Başlamadan önce

Sisteminizde en az aşağıdaki özelliklerin kurulu olması gerekir:

- Sunucu
- Development Toolkit (Örnek programlar için)

- Advanced Message Security (AMS)

Ayrıntılar için bkz. Windows sistemleri için IBM MQ özellikleri .

İşletim sistemi tarafından uygun IBM MQ komutlarının bulunması ve yürütülmesi için geçerli ortamı kullanıma hazırlamak üzere **setmqenv** komutunu kullanma hakkında bilgi için bkz. setmqenv (set IBM MQ environment).

1. Kuyruk yöneticisi ve kuyruk yaratılması

Bu görev hakkında

Aşağıdaki örneklerde, uygulamalar arasında ileti geçirmek için TEST.Q adlı bir kuyruk kullanılır. Advanced Message Security , iletileri standart IBM MQ arabirimi aracılığıyla IBM MQ altyapısına girdikleri noktada imzalamak ve şifrelemek için engelleyicileri kullanır. Temel kuruluş IBM MQ içinde yapılır ve aşağıdaki adımlarda yapılandırılır.

Tüm varsayılan sihirbaz ayarlarını kullanarak QM_VERIFY_AMS kuyruk yöneticisini ve TEST.Q adlı yerel kuyruğunu yaratmak için IBM MQ Explorer komutunu kullanabilir ya da `C:\Program Files\IBM\MQ\bin`içinde bulunan komutları kullanabilirsiniz. Aşağıdaki yönetim komutlarını çalıştırmak için `mqm` kullanıcı grubunun bir üyesi olmanız gerektiğini unutmayın.

Yordam

1. Kuyruk yöneticisi yarat

   ```
   crtmqm QM_VERIFY_AMS
   ```

2. Kuyruk yöneticisini başlat

   ```
   strmqm QM_VERIFY_AMS
   ```

3. **runmqsc** for queue manager QM_VERIFY_AMS içine aşağıdaki komutu girerek TEST.Q adlı bir kuyruk oluşturun

   ```
   DEFINE QLOCAL(TEST.Q)
   ```

Sonuçlar

Yordam tamamlandıysa, **runmqsc** içine girilen komut TEST.Qile ilgili ayrıntıları görüntüler:

```
DISPLAY Q(TEST.Q)
```

2. Kullanıcıların oluşturulması ve yetkilendirilmesi

Bu görev hakkında

Bu örnekte görüntülenen iki kullanıcı vardır: `alice`, gönderen ve bob, alıcı. Uygulama kuyruğunu kullanmak için, bu kullanıcılara bu kuyruğu kullanma yetkisi verilmesi gerekir. Ayrıca, bu kullanıcıları tanımlayacağımız koruma ilkelerinin başarıyla kullanılabilmesi için, bazı sistem kuyruklarına erişim yetkisi verilmesi gerekir. **setmqaut** komutuyla ilgili daha fazla bilgi için bkz. **setmqaut**.

Yordam

1. İki kullanıcı oluşturun ve bu iki kullanıcı için HOMEPATH ve HOMEDRIVE değerinin ayarlandığından emin olun.
2. Kullanıcılara kuyruk yöneticisine bağlanma ve kuyrukla çalışma yetkisi verme

```
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. Ayrıca, iki kullanıcının sistem ilkesi kuyruğuna göz atmasına ve iletileri hata kuyruğuna koymasına izin vermelisiniz.

```
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
```

Uyarı: IBM MQ , SYSTEM.PROTECTION.POLICY.QUEUE QUEUE.

IBM MQ , kullanılabilir tüm ilkeleri önbelleğe almaz. Çok sayıda ilke varsa, IBM MQ sınırlı sayıda ilkeye önbelleğe alır. Bu nedenle, kuyruk yöneticisinde tanımlanmış az sayıda ilke varsa, SYSTEM.PROTECTION.POLICY.QUEUE.

Ancak, çok sayıda ilke tanımlanmışsa ya da eski istemcileri kullanıyorsanız, bu kuyruk için göz atma yetkisi vermeniz gerekir. SYSTEM.PROTECTION.ERROR.QUEUE , AMS kodu tarafından oluşturulan hata iletilerini koymak için kullanılır. Bu kuyruk için koyma yetkisi, yalnızca kuyruğa bir hata iletisi koyma girişiminde bulunduğunuzda denetlenir. AMS korumalı kuyruğuna ileti koyma ya da alma girişiminde bulunduğunuzda, kuyruk için koyma yetkiniz denetlenmez.

Sonuçlar

Kullanıcılar şimdi oluşturulur ve kendilerine gerekli yetkiler verilir.

Sonraki adım

Adımların doğru gerçekleştirilip gerçekleştirilmediğini doğrulamak için `amqsput` ve `amqsget` örneklerini “7. Kurulumun sınanması” sayfa 622bölümünde açıklandığı gibi kullanın.

3. Anahtar veritabanı ve sertifikalarının oluşturulması

Bu görev hakkında

Interceptor, iletiyi şifrelemek için gönderen kullanıcıların genel anahtarını gerektirir. Bu nedenle, genel ve özel anahtarlarla eşlenen kullanıcı kimliklerinin anahtar veritabanı yaratılmalıdır. Kullanıcıların ve uygulamaların birkaç bilgisayara dağıtıldığı gerçek sistemde, her kullanıcının kendi özel anahtar deposu vardır. Benzer şekilde, bu kılavuzda `alice` ve bob için anahtar veritabanları oluşturuyoruz ve bunlar arasında kullanıcı sertifikalarını paylaşıyoruz.

Not: Bu kılavuzda, yerel bağ tanımlarını kullanarak C dilinde yazılmış örnek uygulamaları kullanıyoruz. İstemci bağ tanımlarını kullanarak Java uygulamalarını kullanmayı planlıyorsanız, JRE ' nin bir parçası olan **keytool** komutunu kullanarak bir JKS anahtar deposu ve sertifikaları yaratmanız gerekir (daha fazla ayrıntı için bkz. “Java istemcileriyle AMS için Hızlı Başlangıç Kılavuzu” sayfa 639). Diğer tüm diller ve yerel bağ tanımlarını kullanan Java uygulamaları için bu kılavuzdaki adımlar doğrudur.

Yordam

1. IBM Key Management GUI 'sini kullan (`strmqikm.exe`) `alice`kullanıcısı için yeni bir anahtar veritabanı oluşturmak için.

```
Type:       CMS
```

```
Filename:   alicekey.kdb
Location:   C:/Documents and Settings/alice/AMS
```

Not:

- Veritabanının güvenliğini sağlamak için güçlü bir parola kullanılması önerilir.
- **Parolayı bir dosyaya yazın** onay kutusunun seçili olduğundan emin olun.

2. Anahtar veritabanı içeriği görünümünü **Kişisel Sertifikalar**olarak değiştirin.
3. **Yeni Kendinden Onaylı** seçeneğini belirleyin; Bu senaryoda kendinden imzalı sertifikalar kullanılır.
4. Bu alanları kullanarak, `alice` kullanıcısını şifrelemede kullanmak üzere tanımlayan bir sertifika oluşturun:

```
Key label: Alice_Cert
Common Name: alice
Organisation: IBM
Country: GB
```

Not:

- Bu kılavuzun amacı doğrultusunda, bir Sertifika Yetkilisi kullanılmadan oluşturulabilen kendinden imzalı sertifika kullanıyoruz. Üretim sistemleri için, kendinden imzalı sertifikaların kullanılmaması, bunun yerine bir Sertifika Yetkilisi tarafından imzalanmış sertifikalara güvenilmesi önerilir.
- **Key label** parametresi, gerekli bilgileri almak için kesicilerin arayacağı sertifikanın adını belirtir.
- **Common Name** ve isteğe bağlı parametreler, her kullanıcı için benzersiz olması gereken **Ayırt Edici Ad** (DN) ayrıntılarını belirtir.

5. Kullanıcı için 1-4 arasındaki adımları yineleyin bob

Sonuçlar

İki kullanıcının `alice` ve bob artık her birinin kendinden onaylı bir sertifikası vardır.

4. keystore.conf dosyasının oluşturulması

Bu görev hakkında

Advanced Message Security kesicilerini anahtar veritabanlarının ve sertifikalarının bulunduğu dizine göstermeniz gerekir. Bu, bu bilgileri düz metin biçiminde tutan `keystore.conf` dosyası aracılığıyla yapılır. Her kullanıcının `.mqs` klasöründe ayrı bir `keystore.conf` dosyası olmalıdır. Bu adım hem `alice` hem de bobiçin yapılmalıdır.

`keystore.conf` içeriği şu biçimde olmalıdır:

```
cms.keystore = dir/keystore_file
cms.certificate = certificate_label
```

Örnek

Bu senaryoda, `keystore.conf` içeriği aşağıdaki gibidir:

```
cms.keystore = C:/Documents and Settings/alice/AMS/alicekey
cms.certificate = Alice_Cert
```

Not:

- Anahtar deposu dosyasının yolu, dosya uzantısı olmadan sağlanmalıdır.
- Sertifika etiketi boşluk içerebilir, bu nedenle "Alice_Cert" ve "Alice_Cert" (ucunda boşluk olan), iki farklı sertifikayı içeren etiketler olarak tanınır. Ancak karışıklığı önlemek için, etiket adında boşluk kullanmamanız daha iyi olur.

- Şu anahtar deposu biçimleri vardır: CMS (Cryptographic Message Sözdizimi), JKS (Java Keystore) ve JCEKS (Java Cryptographic Extension Keystore). Daha fazla bilgi için bkz. “AMS için anahtar deposu yapılandırma dosyasının (keystore.conf) yapısı” sayfa 653.
- *%HOMEDRIVE%\%HOMEPATH%*\.mqs\keystore.conf (örn. C:\Documents and Settings\alice\.mqs\keystore.conf), Advanced Message Security ' un keystore.conf dosyasını aradığı varsayılan konumdur. keystore.confiçin varsayılan olmayan bir yerin nasıl kullanılacağına ilişkin bilgi için bkz. “Anahtar depolarının ve sertifikaların AMS ile kullanılması” sayfa 652.
- .mqs dizinini oluşturmak için komut istemini kullanmanız gerekir.

5. Sertifikaları Paylaşma

Bu görev hakkında

Her kullanıcının diğerini başarıyla tanıyabilmesi için sertifikaları iki anahtar veritabanı arasında paylaşın. Bu, her kullanıcının genel sertifikası bir dosyaya açılarak yapılır ve daha sonra diğer kullanıcının anahtar veritabanına eklenir.

Not: *export* (dışa aktarma) seçeneğini değil, *extract* (extract) seçeneğini kullanmaya dikkat edin. *Ayıkla* seçeneği kullanıcının genel anahtarını alırken, *dışa aktarma* seçeneği hem genel hem de özel anahtarı alır. *Dışa aktarma* özelliğinin yanlışlıkla kullanılması, özel anahtarını aktararak uygulamanızı tamamen tehlikeye atar.

Yordam

1. alice ' i tanıtan sertifikayı bir dış dosyaya çıkarın:

```
runmqakm -cert -extract -db "C:/Documents and Settings/alice/AMS/alicekey.kdb" -pw passw0rd
-label Alice_Cert -target alice_public.arm
```

2. Sertifikayı bob ' s anahtar deposuna ekleyin:

```
runmqakm -cert -add -db "C:/Documents and Settings/bob/AMS/bobkey.kdb" -pw passw0rd -label
Alice_Cert -file alice_public.arm
```

3. bobiçin adımları yineleyin:

```
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
```

Sonuçlar

İki kullanıcı alice ve bob , kendinden onaylı sertifikalar oluşturup paylaşarak birbirlerini başarıyla tanımlayabilirler.

Sonraki adım

GUI ' yi kullanarak bir sertifikayı tarayarak ya da ayrıntılarını yazdırmak için aşağıdaki komutları çalıştırarak sertifikanın anahtar deposunda olduğunu doğrulayın:

```
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. Kuyruk ilkesini tanımlama

Bu görev hakkında

İletileri kesmek ve şifreleme anahtarlarına erişmek için kuyruk yöneticisi tarafından oluşturulan ve kesiciler hazırlanarak, `setmqspl` komutunu kullanarak `QM_VERIFY_AMS` üzerinde koruma ilkeleri tanımlamaya başlayabiliriz. Bu komutla ilgili ek bilgi için setmqspl belgesine bakın. Her ilke adı, uygulanacağı kuyruk adıyla aynı olmalıdır.

Örnek

Bu, `TEST.Q` kuyruğu için tanımlanmış bir ilke örneğidir. Örnekte, iletiler Deprecated SHA1 algoritmasıyla imzalanmış ve AES256 algoritmasıyla şifrelenmiştir. `alice` yalnızca geçerli gönderen ve bob bu kuyruktaki iletilerin tek alıcısıdır:

```
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"
```

Not: DN 'ler, anahtar veritabanından ilgili kullanıcının sertifikasında belirtilenlerle tam olarak eşleşiyor.

Sonraki adım

Tanımladığınız ilkeyi doğrulamak için aşağıdaki komutu verin:

```
dspmqspl -m QM_VERIFY_AMS
```

İlke ayrıntılarını `setmqspl` komutları kümesi olarak yazdırmak için `-export` işaretini kullanın. Bu, önceden tanımlanmış ilkelerin saklanmasına izin verir:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Kurulumun sınanması

Bu görev hakkında

Farklı kullanıcılar altında farklı programlar çalıştırarak, uygulamanın doğru yapılandırılıp yapılandırılmadığını doğrulayabilirsiniz.

Yordam

1. Kullanıcıyı kullanıcı olarak çalışacak şekilde değiştir `alice`

   `cmd.exe` öğesini sağ tıklatın ve **Bu şekilde çalıştır ...**seçeneğini belirleyin. İstendiğinde, kullanıcı `alice`olarak oturum açın.
2. Kullanıcı `alice` örnek bir uygulamayı kullanarak bir ileti yazarken:

   ```
   amqsput TEST.Q QM_VERIFY_AMS
   ```
3. İletinin metnini yazın ve Enter tuşuna basın.
4. Kullanıcıyı kullanıcı olarak çalışacak şekilde değiştir bob

   `cmd.exe` öğesini sağ tıklatıp **Çalıştır ...**seçeneğini belirleyerek başka bir pencere açın. İstendiğinde, kullanıcı bobolarak oturum açın.
5. bob kullanıcısı örnek bir uygulamayı kullanarak bir ileti aldıkça:

   ```
   amqsget TEST.Q QM_VERIFY_AMS
   ```

Sonuçlar

Uygulama her iki kullanıcı için de doğru yapılandırıldıysa, bob alma uygulamasını çalıştırdığında `alice` adlı kullanıcının iletisi görüntülenir.

8. Şifrelemenin sınanması

Bu görev hakkında

Şifrelemenin beklendiği gibi gerçekleştiğini doğrulamak için özgün kuyruğa başvuran bir diğer ad kuyruğu oluşturun TEST.Q. Bu diğer ad kuyruğunun güvenlik ilkesi olmayacak ve bu nedenle hiçbir kullanıcı iletinin şifresini çözmek için gereken bilgilere sahip olmayacak ve bu nedenle şifrelenmiş veriler gösterilecek.

Yordam

1. QM_VERIFY_AMS kuyruk yöneticisine karşı **runmqsc** komutunu kullanarak bir diğer ad kuyruğu oluşturun.

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. Diğer ad kuyruğundan göz atmak için bob erişimi ver

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. `alice`kullanıcısı olarak, daha önce olduğu gibi örnek bir uygulamayı kullanarak başka bir ileti girin:

```
amqsput TEST.Q QM_VERIFY_AMS
```

4. bobkullanıcısı olarak, bu kez diğer ad kuyruğu aracılığıyla örnek bir uygulama kullanarak iletiye göz atın:

```
amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. Kullanıcı bobolarak, iletiyi yerel kuyruktan örnek bir uygulama kullanarak alın:

```
amqsget TEST.Q QM_VERIFY_AMS
```

Sonuçlar

`amqsbcg` uygulamasının çıktısı, kuyruktaki şifrelenmiş verileri gösterir ve iletinin şifrelendiğini kanıtlar.

Linux AIX *AIX and Linux üzerinde AMS için Hızlı Başlangıç Kılavuzu*

AIX and Linuxüzerinde ileti güvenliği sağlamak üzere Advanced Message Security ' yi hızlı bir şekilde yapılandırmak için bu kılavuzu kullanın. Bu işlemi tamamladığınızda, kullanıcı kimliklerini doğrulamak için bir anahtar veritabanı ve kuyruk yöneticiniz için tanımlanmış imzalama/şifreleme ilkeleri yaratmış olursunuz.

Başlamadan önce

Sisteminizde en az aşağıdaki bileşenler kurulu olmalıdır:

- Çalıştırma zamanı
- Sunucu
- Örnek programlar
- IBM Global Security Kit (GSKit)
- Advanced Message Security

Her belirli altyapıda bileşen adları için aşağıdaki konulara bakın:

- Linux Linux sistemleri için IBM MQ bileşenleri
- AIX AIX sistemleri için IBM MQ bileşenleri

1. Kuyruk yöneticisi ve kuyruk yaratılması

Bu görev hakkında

Aşağıdaki örneklerde, uygulamalar arasında ileti geçirmek için TEST.Q adlı bir kuyruk kullanılır. Advanced Message Security , iletileri standart IBM MQ arabirimi aracılığıyla IBM MQ altyapısına girdikleri noktada imzalamak ve şifrelemek için kesicileri kullanır. Temel kuruluş IBM MQ içinde yapılır ve aşağıdaki adımlarda yapılandırılır.

IBM MQ Explorer 'ı kullanarak, tüm varsayılan sihirbaz ayarlarını kullanarak QM_VERIFY_AMS kuyruk yöneticisini ve TEST.Q adlı yerel kuyruğunu yaratabilir ya da *MQ_INSTALLATION_PATH*/bin içinde bulunan komutları kullanabilirsiniz. Aşağıdaki yönetim komutlarını çalıştırmak için mqm kullanıcı grubunun bir üyesi olmanız gerektiğini unutmayın.

Yordam

1. Kuyruk yöneticisi yarat

```
crtmqm QM_VERIFY_AMS
```

2. Kuyruk yöneticisini başlat

```
strmqm QM_VERIFY_AMS
```

3. **runmqsc** for queue manager QM_VERIFY_AMS içine aşağıdaki komutu girerek TEST.Q adlı bir kuyruk oluşturun

```
DEFINE QLOCAL(TEST.Q)
```

Sonuçlar

Yordam başarıyla tamamlandıysa, **runmqsc** içine girilen aşağıdaki komut TEST.Qile ilgili ayrıntıları görüntüler:

```
DISPLAY Q(TEST.Q)
```

2. Kullanıcıların oluşturulması ve yetkilendirilmesi

Bu görev hakkında

Bu örnekte görüntülenen iki kullanıcı vardır: alice, gönderen ve bob, alıcı. Uygulama kuyruğunu kullanmak için, bu kullanıcılara bu kuyruğu kullanma yetkisi verilmesi gerekir. Ayrıca, bu kullanıcıları tanımlayacağımız koruma ilkelerinin başarıyla kullanılabilmesi için, bazı sistem kuyruklarına erişim yetkisi verilmesi gerekir. **setmqaut** komutuyla ilgili daha fazla bilgi için bkz. **setmqaut**.

Yordam

1. İki kullanıcıyı oluştur

```
useradd alice
```

```
useradd bob
```

2. Kullanıcılara kuyruk yöneticisine bağlanma ve kuyrukla çalışma yetkisi verme

```
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. Ayrıca, iki kullanıcının sistem ilkesi kuyruğuna göz atmasına ve iletileri hata kuyruğuna koymasına izin vermelisiniz.

```
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
```

Uyarı: IBM MQ , SYSTEM.PROTECTION.POLICY.QUEUE QUEUE.

IBM MQ , kullanılabilir tüm ilkeleri önbelleğe almaz. Çok sayıda ilke varsa, IBM MQ sınırlı sayıda ilkeye önbelleğe alır. Bu nedenle, kuyruk yöneticisinde tanımlanmış az sayıda ilke varsa, SYSTEM.PROTECTION.POLICY.QUEUE.

Ancak, çok sayıda ilke tanımlandıysa ya da eski istemcileri kullanıyorsanız, bu kuyruk için göz atma yetkisi vermeniz gerekir. SYSTEM.PROTECTION.ERROR.QUEUE , AMS kodu tarafından oluşturulan hata iletilerini koymak için kullanılır. Bu kuyruk için koyma yetkisi, yalnızca kuyruğa bir hata iletisi koyma girişiminde bulunduğunuzda denetlenir. AMS korumalı kuyruğuna ileti koyma ya da alma girişiminde bulunduğunuzda, kuyruk için koyma yetkiniz denetlenmez.

Sonuçlar

Kullanıcı grupları şimdi oluşturulur ve kendilerine gerekli yetkiler verilir. Bu şekilde, bu gruplara atanan kullanıcıların kuyruk yöneticisine bağlanma ve kuyruktan alma ve alma izinleri de olur.

Sonraki adım

Adımların doğru gerçekleştirilip gerçekleştirilmediğini doğrulamak için `amqsput` ve `amqsget` örneklerini “8. Şifrelemenin sınanması” sayfa 629bölümünde açıklandığı gibi kullanın.

3. Anahtar veritabanı ve sertifikalarının oluşturulması

Bu görev hakkında

İletiyi şifrelemek için kesici, gönderen kullanıcının özel anahtarını ve alıcının/alıcıların genel anahtarını gerektirir. Bu nedenle, genel ve özel anahtarlarla eşlenen kullanıcı kimliklerinin anahtar veritabanı yaratılmalıdır. Kullanıcıların ve uygulamaların birkaç bilgisayara dağıtıldığı gerçek sistemde, her kullanıcının kendi özel anahtar deposu vardır. Benzer şekilde, bu kılavuzda `alice` ve bob için anahtar veritabanları oluşturuyoruz ve bunlar arasında kullanıcı sertifikalarını paylaşıyoruz.

Not: Bu kılavuzda, yerel bağ tanımlarını kullanarak C dilinde yazılmış örnek uygulamaları kullanıyoruz. İstemci bağ tanımlarını kullanarak Java uygulamalarını kullanmayı planlıyorsanız, JRE ' nin bir parçası olan **`keytool`** komutunu kullanarak bir JKS anahtar deposu ve sertifikaları yaratmanız gerekir (daha fazla ayrıntı için bkz. “Java istemcileriyle AMS için Hızlı Başlangıç Kılavuzu” sayfa 639). Diğer tüm diller ve yerel bağ tanımlarını kullanan Java uygulamaları için bu kılavuzdaki adımlar doğrudur.

Yordam

1. `alice` kullanıcısı için yeni bir anahtar veritabanı oluşturun

```
mkdir /home/alice/.mqs -p
```

```
runmqakm -keydb -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -stash
```

Not:

- Veritabanının güvenliğini sağlamak için güçlü bir parola kullanılması önerilir.
- **stash** değiştirgesi, parolayı `key.sth` kütüğüne saklar; bu kütük, veritabanını açmak için kesicilerin kullanabileceği bir paroladır.

2. Anahtar veritabanının okunabilir olduğundan emin olun

```
chmod +r /home/alice/.mqs/alicekey.kdb
```

3. Şifrelemede kullanılmak üzere `alice` kullanıcısını tanımlayan bir sertifika oluşturun

```
runmqakm -cert -create -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert -dn
"cn=alice,O=IBM,c=GB" -default_cert yes
```

Not:

- Bu kılavuzun amacı doğrultusunda, bir Sertifika Yetkilisi kullanılmadan oluşturulabilen kendinden imzalı sertifika kullanıyoruz. Üretim sistemleri için, kendinden imzalı sertifikaların kullanılmaması, bunun yerine bir Sertifika Yetkilisi tarafından imzalanmış sertifikalara güvenilmesi önerilir.
- **label** parametresi, gerekli bilgileri almak için kesicilerin arayacağı sertifikanın adını belirtir.
- **DN** parametresi, her kullanıcı için benzersiz olması gereken **Ayırt Edici Ad** (DN) ayrıntılarını belirtir.

4. Artık anahtar veritabanını yarattık, sahipliğini ayarlamalı ve diğer tüm kullanıcılar tarafından okunamaz olduğundan emin olmalıydık.

```
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. Kullanıcı için 1-4 arasındaki adımları yineleyin `bob`

Sonuçlar

İki kullanıcının `alice` ve `bob` artık her birinin kendinden onaylı bir sertifikası vardır.

4. keystore.conf oluşturuluyor

Bu görev hakkında

Advanced Message Security kesicilerini anahtar veritabanlarının ve sertifikalarının bulunduğu dizine göstermeniz gerekir. Bu, bu bilgileri düz metin biçiminde tutan `keystore.conf` dosyası aracılığıyla yapılır. Her kullanıcının `.mqs` klasöründe ayrı bir `keystore.conf` dosyası olmalıdır. Bu adım hem `alice` hem de `bob`için yapılmalıdır.

`keystore.conf` içeriği şu biçimde olmalıdır:

```
cms.keystore = dir/keystore_file
```

```
cms.certificate = certificate_label
```

Örnek

Bu senaryoda, `keystore.conf` içeriği aşağıdaki gibidir:

```
cms.keystore = /home/alice/.mqs/alicekey
cms.certificate = Alice_Cert
```

Not:

- Anahtar deposu dosyasının yolu, dosya uzantısı olmadan sağlanmalıdır.
- Şu anahtar deposu biçimleri vardır: `CMS` (Cryptographic Message Sözdizimi), `JKS` (Java Keystore) ve `JCEKS` (Java Cryptographic Extension Keystore). Daha fazla bilgi için bkz. "AMS için anahtar deposu yapılandırma dosyasının (keystore.conf) yapısı" sayfa 653.
- `HOME/.mqs/keystore.conf` , Advanced Message Security ' in `keystore.conf` dosyasını aradığı varsayılan konumdur. `keystore.conf`için varsayılan olmayan bir yerin nasıl kullanılacağına ilişkin bilgi için bkz. "Anahtar depolarının ve sertifikaların AMS ile kullanılması" sayfa 652.

5. Sertifikaları Paylaşma

Bu görev hakkında

Her kullanıcının diğerini başarıyla tanıyabilmesi için sertifikaları iki anahtar veritabanı arasında paylaşın. Bu, her kullanıcının genel sertifikası bir dosyaya açılarak yapılır ve daha sonra diğer kullanıcının anahtar veritabanına eklenir.

Not: *export* (dışa aktarma) seçeneğini değil, *extract* (extract) seçeneğini kullanmaya dikkat edin. *Ayıkla* seçeneği kullanıcının genel anahtarını alırken, *dışa aktarma* seçeneği hem genel hem de özel anahtarı alır. *export* ' un yanlışlıkla kullanılması, özel anahtarını aktararak uygulamanızı tamamen tehlikeye atar.

Yordam

1. `alice` ' i tanıtan sertifikayı bir dış dosyaya çıkarın:

   ```
   runmqakm -cert -extract -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Alice_Cert
   -target alice_public.arm
   ```

2. Sertifikayı bob ' s anahtar deposuna ekleyin:

   ```
   runmqakm -cert -add -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Alice_Cert -file
   alice_public.arm
   ```

3. bobiçin adımı yineleyin:

   ```
   runmqakm -cert -extract -db /home/bob/.mqs/bobkey.kdb -pw passw0rd -label Bob_Cert -target
   bob_public.arm
   ```

4. bob sertifikasını `alice` ' s anahtar deposuna ekleyin:

   ```
   runmqakm -cert -add -db /home/alice/.mqs/alicekey.kdb -pw passw0rd -label Bob_Cert -file
   bob_public.arm
   ```

Sonuçlar

İki kullanıcı `alice` ve bob , kendinden onaylı sertifikalar oluşturup paylaşarak birbirlerini başarıyla tanımlayabilirler.

Sonraki adım

Ayrıntılarını yazdırmak için aşağıdaki komutları çalıştırarak bir sertifikanın anahtar deposunda olduğunu doğrulayın:

```
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. Kuyruk ilkesini tanımlama

Bu görev hakkında

İletileri kesmek ve şifreleme anahtarlarına erişmek için hazırlanan kuyruk yöneticisi ile `setmqspl` komutunu kullanarak `QM_VERIFY_AMS` üzerinde koruma ilkeleri tanımlamaya başlayabiliriz. Bu komutla ilgili ek bilgi için setmqspl belgesine bakın. Her ilke adı, uygulanacağı kuyruk adıyla aynı olmalıdır.

Örnek

Bu, `TEST.Q` kuyruğu için tanımlanan bir ilke örneğidir. Bu örnekte, iletiler Deprecated SHA1 algoritması kullanılarak `alice` kullanıcısı tarafından imzalanır ve 256 bit AES algoritması kullanılarak şifrelenir. `alice` yalnızca geçerli gönderen ve bob bu kuyruktaki iletilerin tek alıcısıdır:

```
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"
```

Not: DN 'ler, anahtar veritabanından ilgili kullanıcının sertifikasında belirtilenlerle tam olarak eşleşiyor.

Sonraki adım

Tanımladığınız ilkeyi doğrulamak için aşağıdaki komutu verin:

```
dspmqspl -m QM_VERIFY_AMS
```

İlke ayrıntılarını `setmqspl` komutları kümesi olarak yazdırmak için `-export` işaretini kullanın. Bu, önceden tanımlanmış ilkelerin saklanmasına izin verir:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Kurulumun sınanması

Bu görev hakkında

Farklı kullanıcılar altında farklı programlar çalıştırarak, uygulamanın doğru yapılandırılıp yapılandırılmadığını doğrulayabilirsiniz.

Yordam

1. Örnekleri içeren dizine geçin. MQ varsayılan olmayan bir yerde kuruluysa, bu farklı bir yerde olabilir.

   ```
   cd /opt/mqm/samp/bin
   ```

2. Kullanıcıyı kullanıcı olarak çalışacak şekilde değiştir `alice`

   ```
   su alice
   ```

3. Kullanıcı `alice`olarak, örnek bir uygulamayı kullanarak bir ileti girin:

   ```
   ./amqsput TEST.Q QM_VERIFY_AMS
   ```

4. İletinin metnini yazın ve Enter tuşuna basın.

5. `alice` kullanıcısı olarak çalıştırmayı durdur

```
exit
```

6. Kullanıcıyı kullanıcı olarak çalışacak şekilde değiştir bob

```
su bob
```

7. bobkullanıcısı olarak örnek bir uygulamayı kullanarak bir ileti alın:

```
./amqsget TEST.Q QM_VERIFY_AMS
```

Sonuçlar

Uygulama her iki kullanıcı için de doğru yapılandırıldıysa, bob alma uygulamasını çalıştırdığında `alice` adlı kullanıcının iletisi görüntülenir.

8. Şifrelemenin sınanması

Bu görev hakkında

Şifrelemenin beklendiği gibi gerçekleştiğini doğrulamak için özgün kuyruğa başvuran bir diğer ad kuyruğu oluşturun TEST.Q. Bu diğer ad kuyruğunun güvenlik ilkesi olmayacak, bu nedenle hiçbir kullanıcı iletinin şifresini çözecek bilgilere sahip olmayacak ve bu nedenle şifrelenmiş veriler gösterilecek.

Yordam

1. QM_VERIFY_AMS kuyruk yöneticisine karşı **runmqsc** komutunu kullanarak bir diğer ad kuyruğu oluşturun.

```
DEFINE QALIAS(TEST.ALIAS) TARGET(TEST.Q)
```

2. Diğer ad kuyruğundan göz atmak için bob erişimi ver

```
setmqaut -m QM_VERIFY_AMS -n TEST.ALIAS -t queue -p bob +browse
```

3. `alice`kullanıcısı olarak, daha önce olduğu gibi örnek bir uygulamayı kullanarak başka bir ileti girin:

```
./amqsput TEST.Q QM_VERIFY_AMS
```

4. bobkullanıcısı olarak, bu kez diğer ad kuyruğu aracılığıyla örnek bir uygulama kullanarak iletiye göz atın:

```
./amqsbcg TEST.ALIAS QM_VERIFY_AMS
```

5. Kullanıcı bobolarak, iletiyi yerel kuyruktan örnek bir uygulama kullanarak alın:

```
./amqsget TEST.Q QM_VERIFY_AMS
```

Sonuçlar

`amqsbcg` uygulamasının çıkışı, kuyruktaki şifrelenmiş verileri gösterecek ve iletinin şifrelendiğini kanıtlayacak.

z/OS *z/OS üzerinde örnek AMS yapılandırmaları*

Bu bölümde, z/OSüzerinde Advanced Message Security kuyruklama senaryoları için örnek ilke ve sertifika yapılandırmaları sağlanmaktadır.

Advanced Message Security'yi nasıl yapılandıracağınıza ilişkin ayrıntılar için Advanced Message Security for z/OS' un yapılandırılması konusuna bakın.

Örnekler, gerekli Advanced Message Security ilkelerini ve kullanıcılarla ve anahtarlarla göreli olarak var olması gereken dijital sertifikaları kapsar. Örnekler, senaryolara dahil olan kullanıcıların Advanced Message Security için kullanıcılara kaynak izinleri veriniçinde sağlanan yönergeler izlenerek kurulduğunu varsayar.

Ayrıca, IBM MQ 9.1.3 ' den başlayarak, bkz. sunucudan sunucuya ileti kanalı önleme örnekleri.

z/OS *z/OS üzerinde AMS için bütünlük korumalı iletilerin yerel olarak kuyruğa alınması*
Bu örnek, Advanced Message Security ilkelerinin ve sertifikalarının, bir kuyruğa/kuyruktan, yerel olarak yerleştirme ve alma uygulamalarına bütünlük korumalı iletiler göndermek ve almak için gerekli olduğunu ayrıntılarıyla belirtir.

Örnek kuyruk yöneticisi ve kuyruğu:

```
BNK6        - Queue manager
FIN.XFER.Q7 - Local queue
```

Bu kullanıcılar kullanılır:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

Kullanıcı sertifikalarını oluştur

Bu örnekte yalnızca bir kullanıcı sertifikası gereklidir. Bu, bütünlük korumalı iletileri imzalamak için gerekli olan gönderen kullanıcının sertifikasıdır. Gönderen kullanıcı: 'TELLER5'.

Sertifika Yetkilisi (CA) sertifikası da gereklidir. CA sertifikası, kullanıcının sertifikasını veren yetkinin sertifikasıdır. Bu bir sertifika zinciri olabilir. Bu durumda, Advanced Message Security görev kullanıcısının anahtar halkasında zincirdeki tüm sertifikalar gereklidir; bu durumda kullanıcı WMQBNK6.

RACF RACDCERT komutu kullanılarak bir CA sertifikası yaratılabilir. Bu sertifika, kullanıcı sertifikalarını vermek için kullanılır. Örneğin:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Bu RACDCERT komutu, 'TELLER5' kullanıcısı için bir kullanıcı sertifikası yayınlamak üzere kullanılabilecek bir CA sertifikası yaratır. Örneğin:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Kuruluşunuzda sertifika seçme ya da yaratma yordamlarının yanı sıra sertifika verme ve bunları ilgili sistemlere dağıtma yordamları da vardır.

Bu sertifikaları dışa ve içe aktarırken Advanced Message Security şunları gerektirir:

- CA sertifikası (zincir).
- Kullanıcı sertifikası ve özel anahtarı.

RACFkullanıyorsanız, sertifikaları bir veri kümesine aktarmak için RACDCERT EXPORT komutu kullanılabilir ve sertifikaları veri kümesinden içe aktarmak için RACDCERT ADD komutu kullanılabilir. Bu ve diğer RACDCERT komutlarıyla ilgili daha fazla bilgi için bkz. *z/OS: Security Server RACF Command Language Reference*.

Bu durumda sertifikalar, BNK6kuyruk yöneticisini çalıştıran z/OS sisteminde gereklidir.

Sertifikalar BNK6çalıştıran z/OS sisteminde içe aktarıldığında, kullanıcı sertifikası TRUST özniteliğini gerektirir. RACDCERT ALTER komutu, sertifikaya TRUST özniteliğini eklemek için kullanılabilir. Örneğin:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

Bu örnekte, alıcı kullanıcı için sertifika gerekli değildir.

Sertifikaları ilgili anahtar halkalarına bağla

Gerekli sertifikalar oluşturulduğunda ya da içe aktarıldığında ve güvenilir olarak ayarlandığında, bunlar BNK6çalıştıran z/OS sisteminde uygun kullanıcı anahtarı halkalarına bağlanmalıdır. Anahtar halkalarını yaratmak için RACDCERT ADDRING komutlarını kullanın:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

Bu, Advanced Message Security görev kullanıcısı WMQBNK6için bir anahtarlık ve gönderen kullanıcı 'TELLER5' için bir anahtarlık oluşturur. drq.ams.keyring anahtarlık adının zorunlu olduğunu ve adın büyük ve küçük harfe duyarlı olduğunu unutmayın.

Anahtar halkaları oluşturulduğunda, ilgili sertifikalar bağlanabilir:

```
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))
```

Gönderen kullanıcı sertifikasının DEFAULT olarak bağlanması gerekir. Gönderen kullanıcının drq.ams.keyringdosyasında birden fazla sertifikası varsa, varsayılan sertifika imzalama amacıyla kullanılır.

Kuyruk yöneticisi durdurulup yeniden başlatılıncaya ya da Advanced Message Security sertifika yapılandırmasını yenilemek için z/OS **MODIFY** komutu kullanılıncaya kadar sertifikaların yaratılması ve değiştirilmesi Advanced Message Security tarafından tanınmaz. Örneğin:

```
F BNK6AMSM,REFRESH KEYRING
```

Advanced Message Security ilkesinin yaratılması

Bu örnekte, bütünlük korumalı iletiler 'TELLER5' kullanıcısı olarak çalışan bir uygulama tarafından FIN.XFER.Q7 kuyruğuna konur ve 'FINADM2' kullanıcısı olarak çalışan bir uygulama tarafından aynı kuyruktan alınır, bu nedenle yalnızca bir Advanced Message Security ilkesi gereklidir.

Advanced Message Security ilkeleri, İleti güvenliği ilkesi yardımcı programı (CSQ0UTIL)adresinde belgelenen CSQ0UTIL yardımcı programı kullanılarak yaratılır.

Aşağıdaki komutu çalıştırmak için CSQ0UTIL yardımcı programını kullanın:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

Bu ilkede, kuyruk yöneticisi BNK6olarak tanımlanır. İlke adı ve ilişkili kuyruk: FIN.XFER.Q7. Gönderenin imzasını oluşturmak için kullanılan algoritma: MD5ve gönderen kullanıcının ayırt edici adı (DN): 'CN=Teller5,O=BCO,C=US'.

İlkeyi tanımladıktan sonra, BNK6 kuyruk yöneticisini yeniden başlatın ya da Advanced Message Security ilke yapılandırmasını yenilemek için z/OS **MODIFY** komutunu kullanın. Örneğin:

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *z/OS üzerinde AMS için gizlilik korumalı iletilerin yerel olarak kuyruğa alınması*

Bu örnek, Advanced Message Security ilkelerinin ve sertifikalarının, bir kuyruğa/kuyruktan, yerel olarak koyma ve alma uygulamaları için gizlilik korumalı iletiler gönderip almak için gerekli olduğunu ayrıntılı olarak gösterir. Gizlilik korumalı mesajlar hem imzalı hem de şifreli.

Örnek kuyruk yöneticisi ve yerel kuyruk aşağıdaki gibidir:

```
BNK6        - Queue manager
FIN.XFER.Q8 - Local queue
```

Bu kullanıcılar kullanılır:

```
WMQBNK6  - AMS task user
TELLER5  - Sending user
FINADM2  - Recipient user
```

Bu senaryoyu yapılandırma adımları şunlardır:

Kullanıcı sertifikalarını oluştur

Bu örnekte, iki kullanıcı sertifikası gereklidir. Bunlar, iletileri imzalamak için gerekli olan gönderen kullanıcı sertifikası ve ileti verilerini şifrelemek ve şifresini çözmek için gerekli olan alıcı kullanıcı sertifikasıdır. Gönderen kullanıcı: 'TELLER5' ve alıcı kullanıcı: 'FINADM2'.

Sertifika Yetkilisi (CA) sertifikası da gereklidir.CA sertifikası, kullanıcının sertifikasını veren yetkinin sertifikasıdır. Bu bir sertifika zinciri olabilir. Bu durumda, Advanced Message Security görev kullanıcısının anahtar halkasında zincirdeki tüm sertifikalar gereklidir; bu durumda kullanıcı WMQBNK6.

RACF RACDCERT komutu kullanılarak bir CA sertifikası yaratılabilir. Bu sertifika, kullanıcı sertifikalarını vermek için kullanılır. Örneğin:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Bu RACDCERT komutu, kullanıcılarınTELLER5've'FINADM2' kullanıcı sertifikalarını yayınlamak için kullanılabilecek bir CA sertifikası yaratır. Örneğin:

```
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)
```

Kuruluşunuzda sertifika seçme ya da yaratma yordamlarının yanı sıra sertifika verme ve bunları ilgili sistemlere dağıtma yordamları da vardır.

Bu sertifikaları dışa ve içe aktarırken Advanced Message Security şunları gerektirir:

- CA sertifikası (zincir).
- Gönderen kullanıcı sertifikası ve özel anahtarı.
- Alıcı kullanıcı sertifikası ve özel anahtarı.

RACFkullanıyorsanız, sertifikaları bir veri kümesine aktarmak için RACDCERT EXPORT komutu kullanılabilir ve sertifikaları veri kümesinden içe aktarmak için RACDCERT ADD komutu kullanılabilir. Bu ve diğer RACDCERT komutlarıyla ilgili daha fazla bilgi için *z/OS: Security Server RACF Command Language Reference*içindeki RACDCERT (Manage RACF digital Sertifika) bölümüne bakın.

Bu durumda sertifikalar, BNK6kuyruk yöneticisini çalıştıran z/OS sisteminde gereklidir.

Sertifikalar BNK6çalıştıran z/OS sisteminde içe aktarıldığında, kullanıcı sertifikaları TRUST özniteliğini gerektirir. RACDCERT ALTER komutu, sertifikaya TRUST özniteliğini eklemek için kullanılabilir. Örneğin:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Sertifikaları ilgili anahtar halkalarına bağla

Gerekli sertifikalar oluşturulduğunda ya da içe aktarıldığında ve güvenilir olarak ayarlandığında, bunlar BNK6çalıştıran z/OS sisteminde uygun kullanıcı anahtarı halkalarına bağlanmalıdır. Anahtar halkalarını yaratmak için RACDCERT ADDRING komutunu kullanın:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

Bu, gönderen ve alıcı kullanıcılar için Advanced Message Security görev kullanıcısı ve anahtar halkaları için bir anahtarlık oluşturur. `drq.ams.keyring` anahtarlık adının zorunlu olduğunu ve adın büyük ve küçük harfe duyarlı olduğunu unutmayın.

Anahtar halkaları oluşturulduğunda, ilgili sertifikalar bağlanabilir.

```
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))
```

Gönderen ve alıcı kullanıcı sertifikalarının DEFAULT olarak bağlanması gerekir. Herhangi bir kullanıcının drq.ams.keyringanahtarlığı içinde birden fazla sertifikası varsa, varsayılan sertifika imzalama ve şifre çözme amacıyla kullanılır.

Alıcı kullanıcının sertifikasının, Advanced Message Security görev kullanıcısının USAGE (SITE) ile anahtarlık halkasına da bağlanması gerekir. Bunun nedeni, Gelişmiş İleti Güvenliği görevinin, ileti verilerini şifrelerken alıcının genel anahtarının gerekli olmasıdır. USAGE (SITE), özel anahtarın anahtar halkasında erişilebilir olmasını önler.

Kuyruk yöneticisi durdurulup yeniden başlatılıncaya ya da Advanced Message Security sertifika yapılandırmasını yenilemek için z/OS **MODIFY** komutu kullanılıncaya kadar sertifikaların yaratılması ve değiştirilmesi Advanced Message Security tarafından tanınmaz. Örneğin:

```
F BNK6AMSM,REFRESH KEYRING
```

Advanced Message Security ilkesinin yaratılması

Bu örnekte, gizlilik korumalı iletiler 'TELLER5' kullanıcısı olarak çalışan bir uygulama tarafından FIN.XFER.Q8 kuyruğuna konur ve 'FINADM2' kullanıcısı olarak çalışan bir uygulama tarafından aynı kuyruktan alınır, bu nedenle yalnızca bir Advanced Message Security ilkesi gereklidir.

Advanced Message Security ilkeleri, İleti güvenliği ilkesi yardımcı programı (CSQ0UTIL)adresinde belgelenen CSQ0UTIL yardımcı programı kullanılarak yaratılır.

Aşağıdaki komutu çalıştırmak için CSQ0UTIL yardımcı programını kullanın:

```
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
```

Bu ilkede, kuyruk yöneticisi BNK6olarak tanımlanır. İlke adı ve ilişkili kuyruk: FIN.XFER.Q8. Gönderenin imzasını oluşturmak için kullanılan algoritma: Deprecated SHA1ve gönderen kullanıcının ayırt edici adı (DN): 'CN=Teller5,O=BCO,C=US' ve alıcı kullanıcı: 'CN=FinAdm2,O=BCO,C=US'. İleti verilerini şifrelemek için kullanılan algoritma Deprecated 3DES' dir.

İlkeyi tanımladıktan sonra, BNK6 kuyruk yöneticisini yeniden başlatın ya da Advanced Message Security ilke yapılandırmasını yenilemek için z/OS **MODIFY** komutunu kullanın. Örneğin:

```
F BNK6AMSM,REFRESH POLICY
```

z/OS *z/OS üzerinde AMS için bütünlük korumalı iletilerin uzaktan kuyruğa alınması*

Bu örnek, iki farklı kuyruk yöneticisi tarafından yönetilen kuyruklara/kuyruklardan bütünlük korumalı iletiler göndermek ve almak için gereken Advanced Message Security ilkelerini ve sertifikalarını ayrıntılı olarak görüntüler. İki kuyruk yöneticisi aynı z/OS sisteminde ya da farklı z/OS sistemlerinde çalışıyor olabilir ya da bir kuyruk yöneticisi Advanced Message Securityçalıştıran dağıtılmış bir sistemde olabilir.

Örnek kuyruk yöneticileri ve kuyrukları şunlardır:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

Not: Bu örnekte, BNK6 ve BNK7 farklı z/OS sistemlerinde çalışan kuyruk yöneticileridir.

Bu kullanıcılar kullanılır:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMStask user on BNK7
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

Bu senaryoyu yapılandırma adımları aşağıdaki gibidir:

Kullanıcı sertifikalarını oluştur

Bu örnekte yalnızca bir kullanıcı sertifikası gereklidir. Bu, bütünlük korumalı iletiyi imzalamak için gerekli olan gönderen kullanıcının sertifikasıdır. Gönderen kullanıcı: 'TELLER5'.

Sertifika Yetkilisi (CA) sertifikası da gereklidir.CA sertifikası, kullanıcının sertifikasını veren yetkinin sertifikasıdır. Bu bir sertifika zinciri olabilir. Bu durumda, Advanced Message Security görev kullanıcısının anahtarlığı için zincirdeki tüm sertifikalar gereklidir; bu durumda WMQBNK7kullanıcısı gerekir.

RACF RACDCERT komutu kullanılarak bir CA sertifikası yaratılabilir. Bu sertifika, kullanıcı sertifikalarını vermek için kullanılır. Örneğin:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Bu RACDCERT komutu, 'TELLER5' kullanıcısı için kullanıcı sertifikası vermek üzere kullanılabilecek bir CA sertifikası yaratır. Örneğin:

```
RACDCERT ID(TELLER5) GENCERT SUBJECTSDN(CN('Teller5') O('BCO') C('US'))
WITHLABEL('Teller5') SIGNWITH(CERTAUTH LABEL('BCOCA'))KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)
```

Kuruluşunuzda sertifika seçme ya da yaratma yordamlarının yanı sıra sertifika verme ve bunları ilgili sistemlere dağıtma yordamları da vardır.

Bu sertifikaları dışa ve içe aktarırken Advanced Message Security şunları gerektirir:

- CA sertifikası (zincir).
- Gönderen kullanıcı sertifikası ve özel anahtarı.

RACFkullanıyorsanız, sertifikaları bir veri kümesine aktarmak için RACDCERT EXPORT komutu kullanılabilir ve sertifikaları veri kümesinden içe aktarmak için RACDCERT ADD komutu kullanılabilir. Bu ve diğer RACDCERT komutlarıyla ilgili daha fazla bilgi için *z/OS: Security Server RACF Command Language Reference*içindeki RACDCERT (Manage RACF digital Sertifika) bölümüne bakın.

Bu durumda sertifikalar, BNK6 ve BNK7kuyruk yöneticisini çalıştıran z/OS sisteminde gereklidir.

Bu örnekte, gönderen sertifikanın BNK6çalıştıran z/OS sisteminde içe aktarılması ve CA sertifikasının BNK7çalıştıran z/OS sisteminde içe aktarılması gerekir. Sertifikalar içe aktarıldığında, kullanıcı sertifikası TRUST özniteliğini gerektirir. RACDCERT ALTER komutu, sertifikaya TRUST özniteliğini eklemek için kullanılabilir. Örneğin, BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

Sertifikaları ilgili anahtar halkalarına bağla

Gerekli sertifikalar oluşturulduğunda ya da içe aktarıldığında ve güvenilir olarak ayarlandığında, bunlar BNK6 ve BNK7çalıştıran z/OS sisteminde uygun kullanıcı anahtarı halkalarına bağlanmalıdır.

Anahtar halkalarını yaratmak için BNK6: üzerinde RACDCERT ADDRING komutunu kullanın:

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

Bu, BNK6üzerinde gönderen kullanıcı için bir anahtarlık oluşturur. drq.ams.keyring anahtarlık adının zorunlu olduğunu ve adın büyük ve küçük harfe duyarlı olduğunu unutmayın.

BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

Bu, BNK7üzerinde Advanced Message Security görev kullanıcısı için bir anahtarlık oluşturur. BNK7üzerinde 'TELLER5' için kullanıcı anahtarlığı gerekmez.

Anahtar halkaları oluşturulduğunda, ilgili sertifikalar bağlanabilir.

BNK6:

```
RACDCERT ID(TELLER5) CONNECT(ID(TELLER5) LABEL('Teller5')
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

BNK7:

```
RACDCERT ID(WMQBNK7) CONNECT(CERTAUTH LABEL('BCOCA')
RING(drq.ams.keyring))
```

Gönderen kullanıcı sertifikasının DEFAULT olarak bağlanması gerekir. Gönderen kullanıcının drq.ams.keyringdosyasında birden fazla sertifikası varsa, varsayılan sertifika imzalama amacıyla kullanılır.

Kuyruk yöneticisi durdurulup yeniden başlatılıncaya ya da Advanced Message Security sertifika yapılandırmasını yenilemek için z/OS **MODIFY** komutu kullanılıncaya kadar sertifikaların yaratılması ve değiştirilmesi Advanced Message Security tarafından tanınmaz. Örneğin:

BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

Advanced Message Security ilkelerinin yaratılması

Bu örnekte, bütünlük korumalı iletiler 'TELLER5' kullanıcısı olarak çalışan bir uygulama tarafından BNK6 üzerinde FIN.XFER.Q7 uzak kuyruğuna yerleştirilir ve 'FINADM2' kullanıcısı olarak çalışan bir uygulama tarafından BNK7 üzerindeki FIN.RCPT.Q7 yerel kuyruğundan alınır, bu nedenle iki Advanced Message Security ilkesi gerekir.

Advanced Message Security ilkeleri, İleti güvenliği ilkesi yardımcı programı (CSQ0UTIL)adresinde belgelenen CSQ0UTIL yardımcı programı kullanılarak yaratılır.

BNK6: üzerinde uzak kuyruk için bir bütünlük ilkesi tanımlamak üzere aşağıdaki komutu çalıştırmak için CSQ0UTIL yardımcı programını kullanın:

```
setmqspl -m BNK6 -p FIN.XFER.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

Bu ilkede, kuyruk yöneticisi BNK6olarak tanımlanır. İlke adı ve ilişkili kuyruk: FIN.XFER.Q7. Gönderenin imzasını oluşturmak için kullanılan algoritma: MD5ve gönderen kullanıcının ayırt edici adı (DN): 'CN=Teller5,O=BCO,C=US'.

Ayrıca, BNK7: üzerinde yerel kuyruk için bir bütünlük ilkesi tanımlamak üzere aşağıdaki komutu çalıştırmak için CSQ0UTIL yardımcı programını kullanın:

```
setmqspl -m BNK7 -p FIN.RCPT.Q7 -s MD5 -a CN=Teller5,O=BCO,C=US
```

Bu ilkede, kuyruk yöneticisi BNK7olarak tanımlanır. İlke adı ve ilişkili kuyruk: FIN.RCPT.Q7. Gönderenin imzası için beklenen algoritma MD5'dir ve gönderen kullanıcının ayırt edici adının (DN)'CN=Teller5,O=BCO,C=US' olması beklenir.

İki ilkeyi tanımladıktan sonra, BNK6 ve BNK7 kuyruk yöneticilerini yeniden başlatın ya da Advanced Message Security ilke yapılandırmalarını yenilemek için z/OS **MODIFY** komutunu kullanın. Örneğin:

BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

z/OS *z/OS üzerinde AMS için gizlilik korumalı iletilerin uzaktan kuyruğa alınması*

Bu örnek, iki farklı kuyruk yöneticisi tarafından yönetilen kuyruklara/kuyruklardan gizlilik korumalı ileti göndermek ve almak için gereken Advanced Message Security ilkelerini ve sertifikalarını ayrıntılı olarak gösterir. İki kuyruk yöneticisi aynı z/OS sisteminde ya da farklı z/OS sistemlerinde çalışıyor olabilir ya da bir kuyruk yöneticisi Advanced Message Securityçalıştıran dağıtılmış bir sistemde olabilir.

Örnek kuyruk yöneticileri ve kuyrukları şunlardır:

```
BNK6        - Sending queue manager
BNK7        - Recipient queue manager
FIN.XFER.Q7 - Remote queue on BNK6
FIN.RCPT.Q7 - Local queue on BNK7
```

Not: Bu örnekte, BNK6 ve BNK7 aynı adı taşıyan farklı z/OS sistemlerinde çalışan kuyruk yöneticileridir.

Bu kullanıcılar kullanılır:

```
WMQBNK6  - AMS task user on BNK6
WMQBNK7  - AMS task user on BNK7
```

```
TELLER5  - Sending user on BNK6
FINADM2  - Recipient user on BNK7
```

Bu senaryoyu yapılandırma adımları aşağıdaki gibidir:

Kullanıcı sertifikalarını oluştur

Bu örnekte, iki kullanıcı sertifikası gereklidir. Bunlar, iletileri imzalamak için gerekli olan gönderen kullanıcı sertifikası ve ileti verilerini şifrelemek ve şifresini çözmek için gerekli olan alıcı kullanıcı sertifikasıdır. Gönderen kullanıcı: 'TELLER5' ve alıcı kullanıcı: 'FINADM2'.

Sertifika Yetkilisi (CA) sertifikası da gereklidir.CA sertifikası, kullanıcının sertifikasını veren yetkinin sertifikasıdır. Bu bir sertifika zinciri olabilir. Bu durumda, Advanced Message Security görev kullanıcısının anahtarlığı için zincirdeki tüm sertifikalar gereklidir; bu durumda WMQBNK7kullanıcısı gerekir.

RACF RACDCERT komutu kullanılarak bir CA sertifikası yaratılabilir. Bu sertifika, kullanıcı sertifikalarını vermek için kullanılır. Örneğin:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('BCOCA') O('BCO') C('US'))
KEYUSAGE(CERTSIGN) WITHLABEL('BCOCA')
```

Bu RACDCERT komutu, kullanıcılarınTELLER5've'FINADM2' kullanıcı sertifikalarını yayınlamak için kullanılabilecek bir CA sertifikası yaratır. Örneğin:

```
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)
```

Kuruluşunuzda sertifika seçme ya da yaratma yordamlarının yanı sıra sertifika verme ve bunları ilgili sistemlere dağıtma yordamları da vardır.

Bu sertifikaları dışa ve içe aktarırken Advanced Message Security şunları gerektirir:

- CA sertifikası (zincir).
- Gönderen kullanıcı sertifikası ve özel anahtarı.
- Alıcı kullanıcı sertifikası ve özel anahtarı.

RACFkullanıyorsanız, sertifikaları bir veri kümesine aktarmak için RACDCERT EXPORT komutu kullanılabilir ve sertifikaları veri kümesinden içe aktarmak için RACDCERT ADD komutu kullanılabilir.

Bu ve diğer RACDCERT komutlarıyla ilgili daha fazla bilgi için *z/OS: Security Server RACF Command Language Reference*içindeki RACDCERT (Manage RACF digital sertifikalar) bölümüne bakın.

Bu durumda sertifikalar, BNK6 ve BNK7kuyruk yöneticisini çalıştıran z/OS sisteminde gereklidir.

Bu örnekte, gönderen ve alıcı sertifikalarının BNK6çalıştıran z/OS sisteminde içe aktarılması ve CA ve alıcı sertifikalarının BNK7çalıştıran z/OS sistemde içe aktarılması gerekir. Sertifikalar içe aktarıldığında, kullanıcı sertifikaları TRUST özniteliğini gerektirir. RACDCERT ALTER komutu, sertifikaya TRUST özniteliğini eklemek için kullanılabilir. Örneğin:

BNK6:

```
RACDCERT ID(TELLER5) ALTER (LABEL('Teller5')) TRUST
```

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

BNK7:

```
RACDCERT ID(FINADM2) ALTER (LABEL('FinAdm2')) TRUST
```

Sertifikaları ilgili anahtar halkalarına bağla

Gerekli sertifikalar oluşturulduğunda ya da içe aktarıldığında ve güvenilir olarak ayarlandığında, bunlar BNK6 ve BNK7çalıştıran z/OS sistemlerinde uygun kullanıcı anahtarı halkalarına bağlanmalıdır.

Anahtar halkalarını yaratmak için RACDCERT ADDRING komutunu kullanın:

BNK6:

```
RACDCERT ID(WMQBNK6) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(TELLER5) ADDRING(drq.ams.keyring)
```

Bu, Advanced Message Security görev kullanıcısı için bir anahtarlık ve BNK6üzerinde gönderen kullanıcı için bir anahtarlık oluşturur. drq.ams.keyring anahtarlık adının zorunlu olduğunu ve adın büyük ve küçük harfe duyarlı olduğunu unutmayın.

BNK7:

```
RACDCERT ID(WMQBNK7) ADDRING(drq.ams.keyring)
```

```
RACDCERT ID(FINADM2) ADDRING(drq.ams.keyring)
```

Bu, Advanced Message Security görev kullanıcısı için bir anahtarlık ve BNK7üzerinde alıcı kullanıcı için bir anahtarlık oluşturur.

Anahtar halkaları oluşturulduğunda, ilgili sertifikalar bağlanabilir.

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))
```

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))
```

Gönderen ve alıcı kullanıcı sertifikalarının DEFAULT olarak bağlanması gerekir. Herhangi bir kullanıcının drq.ams.keyringdosyasında birden fazla sertifikası varsa, varsayılan sertifika imzalama ve şifreleme/şifre çözme amacıyla kullanılır.

BNK6' da, alıcının sertifikasının Advanced Message Security görev kullanıcısının USAGE (SITE) ile anahtarlık halkasına da bağlanması gerekir. Bunun nedeni, Gelişmiş İleti Güvenliği görevinin, ileti verilerini şifrelerken alıcının genel anahtarının gerekli olmasıdır. USAGE (SITE), özel anahtarın anahtar halkasında erişilebilir olmasını önler.

Kuyruk yöneticisi durdurulup yeniden başlatılıncaya ya da Advanced Message Security sertifika yapılandırmasını yenilemek için z/OS **MODIFY** komutu kullanılıncaya kadar sertifikaların yaratılması ve değiştirilmesi Advanced Message Security tarafından tanınmaz. Örneğin:

BNK6:

```
F BNK6AMSM,REFRESH,KEYRING
```

BNK7:

```
F BNK7AMSM,REFRESH,KEYRING
```

Advanced Message Security ilkelerinin yaratılması

Bu örnekte, gizlilik korumalı iletiler 'TELLER5' kullanıcısı olarak çalışan bir uygulama tarafından BNK6 üzerinde FIN.XFER.Q7 uzak kuyruğuna yerleştirilir ve 'FINADM2' kullanıcısı olarak çalışan bir uygulama tarafından BNK7 üzerindeki FIN.RCPT.Q7 yerel kuyruğundan alınır, bu nedenle iki Advanced Message Security ilkesi gerekir.

Advanced Message Security ilkeleri, İleti güvenliği ilkesi yardımcı programı (CSQ0UTIL)adresinde belgelenen CSQ0UTIL yardımcı programı kullanılarak yaratılır.

BNK6: üzerinde uzak kuyruk için bir gizlilik ilkesi tanımlamak üzere aşağıdaki komutu çalıştırmak için CSQ0UTIL yardımcı programını kullanın:

```
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
```

Bu ilkede, kuyruk yöneticisi BNK6olarak tanımlanır. İlke adı ve ilişkili kuyruk: FIN.XFER.Q7. Gönderenin imzasını oluşturmak için kullanılan algoritma: Deprecated SHA1, gönderen kullanıcının ayırt edici adı (DN): 'CN=Teller5,O=BCO,C=US' ve alıcı kullanıcı: 'CN=FinAdm2,O=BCO,C=US'. İleti verilerini şifrelemek için kullanılan algoritma Deprecated 3DES' dir.

Ayrıca, BNK7: üzerinde yerel kuyruk için bir gizlilik ilkesi tanımlamak üzere aşağıdaki komutu çalıştırmak için CSQ0UTIL yardımcı programını kullanın:

```
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
```

Bu ilkede, kuyruk yöneticisi BNK7olarak tanımlanır. İlke adı ve ilişkili kuyruk: FIN.RCPT.Q7. Gönderenin imzası için beklenen algoritma şudur: Deprecated SHA1, gönderen kullanıcının ayırt edici adı (DN) 'CN=Teller5,O=BCO,C=US' ve alıcı kullanıcı 'CN=FinAdm2,O=BCO,C=US'. İleti verilerinin şifresini çözmek için kullanılan algoritma: Deprecated 3DES.

İki ilkeyi tanımladıktan sonra, BNK6 ve BNK7 kuyruk yöneticilerini yeniden başlatın ya da Advanced Message Security ilke yapılandırmasını yenilemek için z/OS **MODIFY** komutunu kullanın. Örneğin:

BNK6:

```
F BNK6AMSM,REFRESH,POLICY
```

BNK7:

```
F BNK7AMSM,REFRESH,POLICY
```

Java istemcileriyle AMS için Hızlı Başlangıç Kılavuzu

İstemci bağ tanımlarını kullanarak bağlanan Java uygulamaları için ileti güvenliğini sağlamak üzere Advanced Message Security ' i hızlı bir şekilde yapılandırmak üzere bu kılavuzu kullanın. Bu işlemi tamamladığınızda, kullanıcı kimliklerini doğrulamak için bir anahtar deposu yaratmış ve kuyruk yöneticiniz için imzalama/şifreleme ilkeleri tanımlamış olursunuz.

Başlamadan önce

Hızlı Başlangıç Kılavuzu (Windows ya da AIX and Linux) içinde açıklandığı gibi uygun bileşenlerin kurulu olduğundan emin olun.

1. Kuyruk yöneticisi ve kuyruk yaratılması

Bu görev hakkında

Aşağıdaki örneklerde, uygulamalar arasında ileti geçirmek için TEST.Q adlı bir kuyruk kullanılır. Advanced Message Security , iletileri standart IBM MQ arabirimi aracılığıyla IBM MQ altyapısına girdikleri noktada

imzalamak ve şifrelemek için engelleyicileri kullanır. Temel kuruluş IBM MQ içinde yapılır ve aşağıdaki adımlarda yapılandırılır.

Yordam

1. Kuyruk yöneticisi yarat

```
crtmqm QM_VERIFY_AMS
```

2. Kuyruk yöneticisini başlat

```
strmqm QM_VERIFY_AMS
```

3. **runmqsc** for queue manager QM_VERIFY_AMS (kuyruk yöneticisi için aşağıdaki komutları girerek bir dinleyici oluşturun ve başlatın)

```
DEFINE LISTENER(AMS.LSTR) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
```

```
START LISTENER(AMS.LSTR)
```

4. **runmqsc** for queue manager QM_VERIFY_AMS (Kuyruk yöneticisi için aşağıdaki komutu girerek uygulamalarımızın bağlanması için bir kanal oluşturun)

```
DEFINE CHANNEL(AMS.SVRCONN) CHLTYPE(SVRCONN)
```

5. **runmqsc** for queue manager QM_VERIFY_AMS içine aşağıdaki komutu girerek TEST.Q adlı bir kuyruk oluşturun

```
DEFINE QLOCAL(TEST.Q)
```

Sonuçlar

Yordam başarıyla tamamlandıysa, **runmqsc** içine girilen aşağıdaki komut TEST.Qile ilgili ayrıntıları görüntüler:

```
DISPLAY Q(TEST.Q)
```

2. Kullanıcıların oluşturulması ve yetkilendirilmesi

Bu görev hakkında

Bu senaryoda görünen iki kullanıcı vardır: alice, gönderen ve bob, alıcı. Uygulama kuyruğunu kullanmak için, bu kullanıcılara bu kuyruğu kullanma yetkisi verilmesi gerekir. Ayrıca, bu senaryoda tanımlanan koruma ilkelerini başarıyla kullanmak için, bu kullanıcılara bazı sistem kuyruklarına erişim yetkisi verilmesi gerekir. **setmqaut** komutuyla ilgili daha fazla bilgi için bkz. **setmqaut**.

Yordam

1. İki kullanıcıyı, altyapınıza ilişkin **Hızlı Başlama Kılavuzu** ' nda (Windows ya da AIX and Linux) açıklandığı şekilde oluşturun.
2. Kullanıcılara kuyruk yöneticisine bağlanma ve kuyrukla çalışma yetkisi verme

```
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. Ayrıca, iki kullanıcının sistem ilkesi kuyruğuna göz atmasına ve iletileri hata kuyruğuna koymasına izin vermelisiniz.

```
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
```

Uyarı: IBM MQ , SYSTEM.PROTECTION.POLICY.QUEUE QUEUE.

IBM MQ , kullanılabilir tüm ilkeleri önbelleğe almaz. Çok sayıda ilke varsa, IBM MQ sınırlı sayıda ilkeye önbelleğe alır. Bu nedenle, kuyruk yöneticisinde tanımlanmış az sayıda ilke varsa, SYSTEM.PROTECTION.POLICY.QUEUE.

Ancak, çok sayıda ilke tanımlanmışsa ya da eski istemcileri kullanıyorsanız, bu kuyruk için göz atma yetkisi vermeniz gerekir. SYSTEM.PROTECTION.ERROR.QUEUE , AMS kodu tarafından oluşturulan hata iletilerini koymak için kullanılır. Bu kuyruk için koyma yetkisi, yalnızca kuyruğa bir hata iletisi koyma girişiminde bulunduğunuzda denetlenir. AMS korumalı kuyruğuna ileti koyma ya da alma girişiminde bulunduğunuzda, kuyruk için koyma yetkiniz denetlenmez.

Sonuçlar

Kullanıcılar şimdi oluşturulur ve kendilerine gerekli yetkiler verilir.

Sonraki adım

Adımların doğru gerçekleştirilip gerçekleştirilmediğini doğrulamak için `JmsProducer` ve `JmsConsumer` örneklerini “7. Kurulumun sınanması” sayfa 644bölümünde açıklandığı gibi kullanın.

3. Anahtar veritabanı ve sertifikalarının oluşturulması

Bu görev hakkında

İletiyi kesiciye şifrelemek için gönderen kullanıcıların genel anahtarı gerekir. Bu nedenle, genel ve özel anahtarlarla eşlenen kullanıcı kimliklerinin anahtar veritabanı yaratılmalıdır. Kullanıcıların ve uygulamaların birkaç bilgisayara dağıtıldığı gerçek sistemde, her kullanıcının kendi özel anahtar deposu vardır. Benzer şekilde, bu kılavuzda `alice` ve bob için anahtar veritabanları oluşturuyoruz ve bunlar arasında kullanıcı sertifikalarını paylaşıyoruz.

Not: Bu kılavuzda, istemci bağ tanımlarını kullanarak Java bağlantısında yazılan örnek uygulamaları kullanıyoruz. Yerel bağ tanımlarını ya da C uygulamalarını kullanarak Java uygulamalarını kullanmayı planlıyorsanız, **runmqakm** komutunu kullanarak bir CMS anahtar deposu ve sertifikaları oluşturmanız gerekir. Bu, **Hızlı Başlangıç Kılavuzu** ' nda (Windows ya da AIX and Linux) gösterilir.

Yordam

1. Anahtar deponuzun yaratılacağı bir dizin yaratın; örneğin, `/home/alice/.mqs`. Bunu, altyapınız için **Hızlı Başlangıç Kılavuzu** ' nun (Windows ya da AIX and Linux) kullandığı dizinle aynı dizinde oluşturmak isteyebilirsiniz.

   Not: Bu dizin, aşağıdaki adımlarda *keystore-dir* olarak adlandırılır.
2. Şifrelemede kullanılacak `alice` kullanıcısını tanıtan yeni bir anahtar deposu ve sertifika yaratır

   Not: keytool komutu JRE ' nin bir parçasıdır.

```
keytool -genkey -alias Alice_Java_Cert -keyalg RSA -keystore keystore-dir/keystore.jks
-storepass passw0rd
-dname "CN=alice, O=IBM, C=GB" -keypass passw0rd
```

Not:

- *keystore-dir* dizininizde boşluk varsa, anahtar deponuzun tam adını tırnak içine almanız gerekir
- Anahtar deposunu korumak için güçlü bir parola kullanılması önerilir.
- Bu kılavuzun amacı doğrultusunda, bir Sertifika Yetkilisi kullanılmadan oluşturulabilen kendinden imzalı sertifika kullanıyoruz. Üretim sistemleri için, kendinden imzalı sertifikaların kullanılmaması, bunun yerine bir Sertifika Yetkilisi tarafından imzalanmış sertifikalara güvenilmesi önerilir.
- **alias** parametresi, gerekli bilgileri almak için kesicilerin arayacağı sertifikanın adını belirtir.
- **dname** parametresi, her kullanıcı için benzersiz olması gereken **Ayırt Edici Ad** (DN) ayrıntılarını belirtir.

3. AIX and Linuxüzerinde, anahtar deposunun okunabilir olduğundan emin olun

```
chmod +r keystore-dir/keystore.jks
```

4. Kullanıcı için step1-4 komutunu yineleyin bob

Sonuçlar

İki kullanıcının `alice` ve bob artık her birinin kendinden onaylı bir sertifikası vardır.

4. keystore.conf dosyasının oluşturulması

Bu görev hakkında

Advanced Message Security kesicilerini anahtar veritabanlarının ve sertifikalarının bulunduğu dizine göstermeniz gerekir. Bu, bu bilgileri düz metin biçiminde tutan `keystore.conf` dosyası aracılığıyla yapılır. Her kullanıcının ayrı bir `keystore.conf` dosyası olmalıdır. Bu adım hem `alice` hem de bobiçin yapılmalıdır.

Örnek

Bu senaryoda, `alice` için `keystore.conf` içeriği aşağıdaki gibidir:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Alice_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

Bu senaryoda, bob için `keystore.conf` içeriği aşağıdaki gibidir:

```
JKS.keystore = keystore-dir/keystore
JKS.certificate = Bob_Java_Cert
JKS.encrypted = no
JKS.keystore_pass = passw0rd
JKS.key_pass = passw0rd
JKS.provider = IBMJCE
```

Not:

- Anahtar deposu dosyasının yolu, dosya uzantısı olmadan sağlanmalıdır.
- Hızlı Başlangıç Kılavuzu 'ndaki (Windows ya da AIX and Linux) yönergeleri izlediğiniz için önceden bir `keystore.conf` dosyeniz varsa, bu satırları eklemek için var olan dosyayı düzenleyebilirsiniz.
- Daha fazla bilgi için bkz “AMS için anahtar deposu yapılandırma dosyasının (keystore.conf) yapısı” sayfa 653.

5. Sertifikaları paylaşma

Bu görev hakkında

Her bir kullanıcının diğerini başarıyla tanıyabilmesi için sertifikaları iki anahtar deposu arasında paylaşın. Bu, her kullanıcının sertifikasının alınması ve diğer kullanıcının anahtar deposuna içe aktarılması yoluyla gerçekleştirilir.

Not: *Çıkar* ve *dışa aktar* terimleri farklı sertifika araçları tarafından farklı şekilde kullanılır. Örneğin, IBM Global Security Kit (GSKit) **strmqikm** komut (ikeyman) aracı, sizin *çıkarma* sertifikalarınızı (ortak anahtarlar) ve sizin *dışa aktarma* özel anahtarlarınızı ayırır. Bu ayrım, her iki seçeneği de sunan araçlar için son derece önemlidir, çünkü *export* ' u yanlışlıkla kullanmak, özel anahtarını aktararak uygulamanızı tamamen tehlikeye atacaktır. Ayrım çok önemli olduğundan, IBM MQ belgeleri bu terimleri tutarlı bir şekilde kullanmayı dener. Ancak Java keytool, yalnızca genel anahtarı ayıklayan *exportcert* adlı bir komut satırı seçeneği sağlar. Bu nedenlerden ötürü, aşağıdaki yordam, *exportcert* seçeneğini kullanarak sertifikalarının alınması anlamına gelir.

Yordam

1. `alice`' i tanıtan sertifikayı çıkarın.

```
keytool -exportcert -keystore alice-keystore-dir/keystore.jks -storepass passw0rd
-alias Alice_Java_Cert -file alice-keystore-dir/Alice_Java_Cert.cer
```

2. `alice` ' i tanıtan sertifikayı, bob tarafından kullanılacak anahtar deposuna aktarın. İstendiğinde, bu sertifikaya güveneceğini belirtin.

```
keytool -importcert -file alice-keystore-dir/Alice_Java_Cert.cer -alias Alice_Java_Cert
-keystore bob-keystore-dir/keystore.jks -storepass passw0rd
```

3. bob için adımları yineleyin

Sonuçlar

İki kullanıcı `alice` ve bob , kendinden onaylı sertifikalar oluşturup paylaşarak birbirlerini başarıyla tanımlayabilirler.

Sonraki adım

Ayrıntılarını yazdırmak için aşağıdaki komutları çalıştırarak, bir sertifikanın anahtar deposunda olduğunu doğrulayın:

```
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. Kuyruk ilkesini tanımlama

Bu görev hakkında

İletileri kesmek ve şifreleme anahtarlarına erişmek için kuyruk yöneticisi tarafından oluşturulan ve kesiciler hazırlanarak, `setmqspl` komutunu kullanarak `QM_VERIFY_AMS` üzerinde koruma ilkeleri tanımlamaya başlayabiliriz. Bu komutla ilgili ek bilgi için setmqspl belgesine bakın. Her ilke adı, uygulanacağı kuyruk adıyla aynı olmalıdır.

Örnek

Bu, TEST.Q kuyruğunda tanımlanan, `alice` kullanıcısı tarafından Deprecated SHA1 algoritması kullanılarak imzalanan ve kullanıcı için 256 bit AES algoritması kullanılarak şifrelenen bir ilke örneğidir 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"
```

Not: DN 'ler, anahtar veritabanından ilgili kullanıcının sertifikasında belirtilenlerle tam olarak eşleşiyor.

Sonraki adım

Tanımladığınız ilkeyi doğrulamak için aşağıdaki komutu verin:

```
dspmqspl -m QM_VERIFY_AMS
```

İlke ayrıntılarını `setmqspl` komutları kümesi olarak yazdırmak için `-export` işareti. Bu, önceden tanımlanmış ilkelerin saklanmasına izin verir:

```
dspmqspl -m QM_VERIFY_AMS -export >restore_my_policies.bat
```

7. Kurulumun sınanması

Başlamadan önce

Kullandığınız Java sürümünde kısıtlamasız JCE ilke dosyalarının kurulu olduğundan emin olun.

Not: IBM MQ kuruluşunda sağlanan Java sürümünde bu ilke dosyaları zaten var. *MQ_INSTALLATION_PATH*`/java/bin`içinde bulunabilir.

Bu görev hakkında

Farklı kullanıcılar altında farklı programlar çalıştırarak, uygulamanın doğru yapılandırılıp yapılandırılmadığını doğrulayabilirsiniz. Farklı kullanıcılar altında program çalıştırmaya ilişkin ayrıntılar için altyapınıza ilişkin **Hızlı Başlangıç Kılavuzu** ' na (Windows ya da AIX) bakın.

Yordam

1. Bu JMS örnek uygulamalarını çalıştırmak için, örnekler dizininin içerildiğinden emin olmak üzere IBM MQ classes for JMS tarafından kullanılan ortam değişkenlerinde gösterildiği gibi altyapınıza ilişkin CLASSPATH ayarını kullanın.
2. `alice`kullanıcısı olarak, istemci olarak bağlanan örnek bir uygulamayı kullanarak bir ileti koyun:

   ```
   java JmsProducer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```
3. bobkullanıcısı olarak, istemci olarak bağlanan örnek bir uygulamayı kullanarak bir ileti alın:

   ```
   java JmsConsumer -m QM_VERIFY_AMS -d TEST.Q -h localhost -p 1414 -l AMS.SVRCONN
   ```

Sonuçlar

Uygulama her iki kullanıcı için de doğru yapılandırıldıysa, bob alma uygulamasını çalıştırdığında `alice` adlı kullanıcının iletisi görüntülenir.

AMS üzerindeki uzak kuyrukların korunması

Uzak kuyrukları tam olarak korumak için, ilkeler uzak kuyrukta ve iletilerin iletileceği yerel kuyrukta belirlenmelidir.

Bir ileti uzak kuyruğa konduğunda, Advanced Message Security işlemi durdurur ve iletiyi uzak kuyruğa ilişkin ilke kümesine göre işler. Örneğin, bir şifreleme ilkesi için, ileti işlemek üzere IBM MQ ' e iletilmeden önce şifrelenir. Advanced Message Security uzak kuyruğa konan iletiyi işledikten sonra, IBM MQ iletiyi ilişkili iletim kuyruğuna koyar ve hedef kuyruk yöneticisine ve hedef kuyruğa iletir.

Yerel kuyrukta GET işlemi gerçekleştirildiğinde, Advanced Message Security iletinin kodunu yerel kuyrukta ayarlanan ilkeye göre çözmeyi dener. İşlemin başarılı olması için, iletinin şifresini çözmek için kullanılan ilkenin, iletiyi şifrelemek için kullanılanla aynı olması gerekir. Herhangi bir uyumsuzluk, iletinin reddedilmesine neden olur.

Herhangi bir nedenle her iki ilke de aynı anda ayarlanamazsa, aşamalı bir devreye alma desteği sağlanır. İlke, tolerans işareti açık yerel bir kuyrukta ayarlanabilir; bu, kuyruktan ileti alma girişimi güvenlik ilkesi kümesi olmayan bir ileti içerdiğinde, kuyrukla ilişkili bir ilkenin yoksayılabileceğini gösterir. Bu durumda GET iletinin şifresini çözmeye çalışır, ancak şifrelenmemiş iletilerin teslim edilmesine izin verir. Bu yolla, uzak kuyruklardaki ilkeler, yerel kuyruklar korunduktan (ve sınandıktan) sonra belirlenebilir.

Unutmayın: Advanced Message Security silme işlemi tamamlandıktan sonra tolerans işaretini kaldırın.

İlgili başvurular

setmqspl (güvenlik ilkesini ayarla)

Korumalı iletileri IBM Integration Bus kullanarak AMS ile yöneltme

Advanced Message Security , IBM Integration Busya da WebSphere Message Broker 8.0.0.1 ' in (ya da sonraki bir düzeyinin) kurulu olduğu bir altyapıdaki iletileri koruyabilir. IBM Integration Bus ortamında güvenliği uygulamadan önce her iki ürünün doğasını anlamanız gerekir.

Bu görev hakkında

Advanced Message Security , ileti bilgi yükünün uçtan uca güvenliğini sağlar. Bu, yalnızca geçerli gönderenler ve bir iletinin alıcıları olarak belirtilen tarafların iletiyi üretebilir ya da alabileceği anlamına gelir. Bu, IBM Integration Busiçinden akan iletileri güvenli bir şekilde işlemek için IBM Integration Bus ' in iletileri içeriğini bilmeden işlemesine izin verebileceğiniz anlamına gelir (Senaryo 1) ya da yetkili bir kullanıcı olarak ileti alabilir ve gönderebilir (Senaryo 2).

1. senaryo- Integration Bus ileti içeriğini göremiyor

Başlamadan önce

IBM Integration Bus ürününüzü var olan bir kuyruk yöneticisine bağlamanız gerekir. *QMgrName* yerine, izleyen komutlarda var olan bu kuyruk yöneticisi adını koyun.

Bu görev hakkında

Bu senaryoda Alice, `QIN`giriş kuyruğuna korumalı bir ileti koyar. `route`Toileti özelliğine dayalı olarak, ileti *bob 'un* (`QBOB`),[1](`QCECIL`) ya da varsayılan (`QDEF`) kuyruğu. Advanced Message Security , korunmayan ve IBM Integration Bustarafından okunabilen üstbilgileri ve özelliklerini değil, yalnızca ileti bilgi yükünü koruduğundan yöneltme mümkündür. Advanced Message Security yalnızca *alice*, *bob* ve *cecil*tarafından kullanılır. Bunu IBM Integration Busiçin kurmanız ya da yapılandırmanız gerekmez.

IBM Integration Bus , iletinin şifresini çözme girişimini önlemek için korunmayan diğer ad kuyruğundan korunan iletiyi alır. Korunan kuyruğu doğrudan kullanacaksa, iletinin şifresini çözmek imkansız olarak DEAD LETTER kuyruğuna konması gerekir. İleti IBM Integration Bus tarafından yönlendirilir ve hedef kuyruğa değişmeden ulaşır. Bu nedenle, özgün yazar tarafından imzalanmaya devam eder (hem *bob* , hem de *cecil* yalnızca *alice* tarafından gönderilen iletileri kabul eder) ve önceki gibi korunur (yalnızca *bob* ve *cecil* okuyabilir). IBM Integration Bus , yöneltilen iletiyi korunmayan bir diğer ada yerleştirir. Alıcılar iletiyi, AMS ' un iletinin şifresini saydam bir şekilde çözeceği korumalı bir çıkış kuyruğundan alır.

[1] Cecil 'in

Yordam

1. *Alice*, *bob* ve *cecil* ' yi **Hızlı Başlangıç Kılavuzu** (Windows ya da AIX) içinde açıklandığı gibi Advanced Message Security ürününü kullanacak şekilde yapılandırın.

   Aşağıdaki adımların tamamlandığından emin olun:

   - Kullanıcıların yaratılması ve yetkilendirilmesi
   - Anahtar Veritabanı ve Sertifikalar Oluşturma
   - keystore.conf oluşturma
2. *alice 'nin* sertifikasını *bob* ve *cecil*için sağlayın, böylece *alice* , iletilerde dijital imzalar denetlenirken bunlar tarafından tanımlanabilir.

   Bunu yapmak için, *alice* ' i tanıtan sertifikayı bir dış dosyaya açın ve çıkarılan sertifikayı *bob 'un* ve *cecil 'in* anahtar depolarına ekleyin. Aşağıda açıklanan yöntemi kullanmanız önemlidir: **Görev 5. Sharing Certificates** in the **Quick Start Guide** (Windows or AIX).
3. *bob* ve *cecil* sertifikalarını *alice*' ye sağlayın, böylece *alice bob* ve *cecil*için şifrelenmiş iletiler gönderebilir.

   Bunu, önceki adımda belirtilen yöntemi kullanarak yapın.
4. Kuyruk yöneticinizde `QIN`, `QBOB`, `QCECIL` ve `QDEF`adlı yerel kuyrukları tanımlayın.

   ```
   DEFINE QLOCAL(QIN)
   ```
5. `QIN` kuyruğu için güvenlik ilkesini uygun bir yapılandırmaya ayarlayın. `QBOB`, `QCECIL` ve `QDEF` kuyrukları için aynı ayarı kullanın.

   ```
   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"
   ```

   Bu senaryoda, *Alice* 'in tek yetkili gönderen olduğu ve *bob* ve *cecil* ' in alıcılar olduğu güvenlik ilkesi varsayılır.
6. Sırasıyla `AIN`, `ABOB` ve `ACECIL` yerel kuyruklara gönderme yapan `QIN`, `QBOB` ve `QCECIL` diğer ad kuyruklarını tanımlayın.

   ```
   DEFINE QALIAS(AIN) TARGET(QIN)
   ```
7. Önceki adımda belirtilen diğer adlara ilişkin güvenlik yapılandırmasının var olmadığını doğrulayın; tersi durumda, ilkesini NONE olarak ayarlayın.

   ```
   dspmqspl -m QMgrName -p AIN
   ```
8. IBM Integration Bus içinde, `AIN` diğer ad kuyruğuna gelen iletileri, iletinin `routeTo` özelliğine bağlı olarak BOB, CECIL ya da DEF düğümüne yöneltmek için bir ileti akışı yaratın. Bunu yapmak için:
   a) `IN` adlı bir `MQInput` düğümü oluşturun ve `AIN` diğer adını kuyruk adı olarak atayın.
   b) `BOB`, `CECIL` ve `DEF`adlı `MQOutput` düğümlerini oluşturun ve diğer ad kuyruklarını `ABOB`, `ACECIL` ve `ADEF` ilgili kuyruk adları olarak atayın.
   c) Bir rota düğümü oluşturun ve buna `TEST`adını verin.
   d) `IN` düğümünü `TEST` düğümünün giriş uçbirimine bağlayın.
   e) `TEST` düğümü için `bob`ve `cecil` çıkış uçbirimleri oluşturun.
   f) `bob` çıkış uçbirimini `BOB` düğümüne bağlayın.
   g) `cecil` çıkış uçbirimini `CECIL` düğümüne bağlayın.
   h) `DEF` düğümünü varsayılan çıkış uçbirimine bağlayın.
   i) Aşağıdaki kuralları uygulayın:

```
$Root/MQRFH2/usr/routeTo/text()="bob"
```

```
$Root/MQRFH2/usr/routeTo/text()="cecil"
```

9. İleti akışını IBM Integration Bus yürütme ortamı bileşenine konuşlandırın.
10. `Alice` kullanıcısı tarafından çalıştırılırken, bob ya da `cecil`değerine sahip `routeTo` adlı bir ileti özelliği de içeren bir ileti konuyor. **amqsstm** örnek uygulamasının çalıştırılması bunu yapmanıza olanak sağlar.

```
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. *bob* kullanıcısı olarak çalışırken, örnek uygulamayı kullanarak `QBOB` kuyruktan iletiyi alın **amqsget**.

Sonuçlar

Alice `QIN` kuyruğuna bir ileti yerleştirdiğinde, ileti korunur. IBM Integration Bus tarafından `AIN` diğer ad kuyruğundan korunan biçimde alınır. IBM Integration Bus , tüm özellikler olarak şifrelenmemiş olan `routeTo` özelliğini okuyan iletinin nereye yönlendirileceğine karar verir. IBM Integration Bus , iletiyi korunmayan uygun diğer ada yerleştirir ve daha fazla korumayı önler. Kuyruktan *bob* ya da *cecil* tarafından alındığında, iletinin şifresi çözülür ve dijital imza doğrulanır.

Senaryo 2- Integration Bus ileti içeriğini görebilir

Bu görev hakkında

Bu senaryoda, bir grup kişinin IBM Integration Bus' e ileti göndermesine izin verilir. Başka bir grup, IBM Integration Bustarafından oluşturulan iletileri alma yetkisine sahiptir. Taraflar ve IBM Integration Bus arasındaki aktarım gizlice dinlenemez.

IBM Integration Bus ' in koruma ilkelerini ve sertifikalarını yalnızca bir kuyruk açıldığında okuduğunu unutmayın; bu nedenle, değişikliklerin yürürlüğe girmesi için koruma ilkelerinde yapılan güncellemeleri yaptıktan sonra yürütme grubunu yeniden yüklemeniz gerekir.

```
mqsireload execution-group-name
```

IBM Integration Bus , ileti bilgi yükünü okumasına ya da imzalamasına izin verilen bir yetkili taraf olarak kabul edilirse, Advanced Message Security ürününü IBM Integration Bus hizmetini başlatan kullanıcı için yapılandırmanız gerekir. İletileri kuyruklara koyan/alan kullanıcının ya da IBM Integration Bus uygulamalarını oluşturan ve dağıtan kullanıcının aynı kullanıcı olması gerekmediğini unutmayın.

Yordam

1. *alice*, *bob*, *cecil* ve *cecil* ve IBM Integration Bus hizmet kullanıcısını **Hızlı Başlangıç Kılavuzu** ' nda (Windows ya da AIX) açıklandığı gibi Advanced Message Security kullanacak şekilde yapılandırın.

   Aşağıdaki adımların tamamlandığından emin olun:

   - Kullanıcıların yaratılması ve yetkilendirilmesi
   - Anahtar Veritabanı ve Sertifikalar Oluşturma
   - keystore.conf oluşturma

2. IBM Integration Bus hizmet kullanıcısına *alice*, *bob*, *cecil* ve *Dave* sertifikalarını sağlayın.

   Bunu yapmak için, dış dosyalara *alice*, *bob*, *cecil* ve *dave* öğelerini tanıtan sertifikaların her birini açın ve çıkarılan sertifikaları IBM Integration Bus anahtar deposuna ekleyin. Aşağıda açıklanan yöntemi kullanmanız önemlidir: **Görev 5. Sharing Certificates** in the **Quick Start Guide** (Windows or AIX).
3. IBM Integration Bus hizmet kullanıcısının *alice*, *bob*, *cecil* ve *dave*için sertifikasını sağlayın.

   Bunu, önceki adımda belirtilen yöntemi kullanarak yapın.

   Not: *Alice* ve *bob* , iletileri doğru şekilde şifrelemek için IBM Integration Bus hizmet kullanıcısının sertifikasına gereksinim duyar. IBM Integration Bus hizmet kullanıcısı, iletilerin yazarlarını doğrulamak için *alice* ve *bob* sertifikalarına gereksinim duyar. IBM Integration Bus hizmet kullanıcısının iletileri şifrelemek için *cecil* ve *Dave* sertifikaları gerekir. *cecil* ve *dave* , iletinin IBM Integration Bus' den geldiğini doğrulamak için IBM Integration Bus hizmet kullanıcısının sertifikasına gereksinim duyar.
4. `IN` adlı bir yerel kuyruk tanımlayın ve yazar olarak belirtilen *alice* ve *bob* ile güvenlik ilkesini ve alıcı olarak belirtilen IBM Integration Bus için hizmet kullanıcısını tanımlayın:

   ```
   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. `OUT`adlı bir yerel kuyruk tanımlayın ve yazar olarak belirtilen IBM Integration Bus için hizmet kullanıcısıyla güvenlik ilkesini ve alıcı olarak belirtilen *cecil* ve *dave* öğelerini tanımlayın:

   ```
   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. IBM Integration Bus içinde bir `MQInput` ve `MQOutput` düğümüyle bir ileti akışı oluşturun. `MQInput` düğümünü `IN` kuyruğunu ve `MQOutput` düğümünü `OUT` kuyruğunu kullanacak şekilde yapılandırın.
7. İleti akışını IBM Integration Bus yürütme ortamı bileşenine konuşlandırın.
8. *alice* ya da *bob* kullanıcısı olarak çalışırken, `IN` örnek uygulamayı kullanarak **amqsput**kuyruğa bir ileti koyun.
9. *cecil* ya da *dave* kullanıcısı olarak çalışırken, `OUT` örnek uygulamayı kullanarak **amqsget**kuyruktan iletiyi alın.

Sonuçlar

alice ya da *bob* tarafından giriş kuyruğuna `IN` gönderilen iletiler şifrelenir ve yalnızca IBM Integration Bus tarafından okunabilir. IBM Integration Bus yalnızca *alice* ve *bob* ' dan gelen iletileri kabul eder ve diğer iletileri reddeder. Kabul edilen iletiler uygun şekilde işlenir, daha sonra *cecil 'in* ve *Dave 'in* çıkış kuyruğuna `OUT`yerleştirilmeden önce anahtarlarıyla imzalanır ve şifrelenir. Yalnızca *cecil* ve *dave* okuma yeteneğine sahiptir, IBM Integration Bus tarafından imzalanmamış iletiler reddedilir.

Advanced Message Security ' yi Managed File Transfer ile kullanma

Bu senaryoda, bir Managed File Transferaracılığıyla gönderilen veriler için ileti gizliliği sağlamak üzere Advanced Message Security ' in nasıl yapılandırılacağı açıklanır.

Başlamadan önce

Korumak istediğiniz Managed File Transfer tarafından kullanılan kuyrukları barındıran IBM MQ kuruluşunda Advanced Message Security bileşeninin kurulu olduğundan emin olun.

Managed File Transfer aracılarınız bağ tanımlama kipinde bağlanıyorsa, yerel kuruluşlarında IBM Global Security Kit (GSKit) bileşeninin de kurulu olduğundan emin olun.

Bu görev hakkında

İki Managed File Transfer aracısı arasında veri aktarımı kesintiye uğradığında, büyük olasılıkla gizli veriler aktarımı yönetmek için kullanılan temel IBM MQ kuyruklarında korumasız kalabilir. Bu senaryoda, Managed File Transfer kuyruklarında bu tür verileri korumak için Advanced Message Security ' in nasıl yapılandırılacağı ve kullanılacağı açıklanmaktadır.

Bu senaryoda, Managed File Transfer senaryosundaaçıklandığı gibi, tek bir kuyruk yöneticisini paylaşan iki Managed File Transfer kuyruğu ve iki aracısı AGENT1 ve AGENT2olan bir makineden oluşan basit bir topoloji ele alınmıştır. Her iki aracı da bağ tanımlama kipinde ya da istemci kipinde aynı şekilde bağlanır.

1. Sertifika yaratılması

Başlamadan önce

Bu senaryo, Managed File Transfer Agent işlemlerini çalıştırmak için `ftagent` in a group FTAGENTS adlı kullanıcının kullanıldığı basit bir modeli kullanır. Kendi kullanıcı ve grup adlarınızı kullanıyorsanız, komutları uygun şekilde değiştirin.

Bu görev hakkında

Advanced Message Security , korunan kuyruklardaki iletileri imzalamak ve/veya şifrelemek için genel anahtar şifrelemesini kullanır.

Not:

- Managed File Transfer aracılarınız bağ tanımlama kipinde çalışıyorsa, CMS (Cryptographic Message Sözdizimi) anahtar deposu oluşturmak için kullandığınız komutlar, altyapınıza ilişkin **Hızlı Başlangıç Kılavuzu** ' nda (Windows ya da AIX) ayrıntılı olarak açıklanmıştır.
- Managed File Transfer aracılarınız istemci kipinde çalışıyorsa, bir JKS (Java Anahtar Deposu) oluşturmanız için gereken komutlar “Java istemcileriyle AMS için Hızlı Başlangıç Kılavuzu” sayfa 639içinde ayrıntılı olarak açıklanmıştır.

Yordam

1. İlgili Hızlı Başlangıç Kılavuzu 'nda ayrıntılı olarak açıklandığı gibi `ftagent` kullanıcısını tanımlamak için kendinden onaylı bir sertifika oluşturun.
   Aşağıdaki gibi bir Ayırt Edici Ad (DN) kullanın:

   ```
   CN=ftagent, OU=MFT, O=<organisation>, L=<location>, ST=<state>, C=<country>
   ```

2. Anahtar deposunun yerini ve içindeki sertifikayı, uygun Hızlı Başlama Kılavuzu 'nda ayrıntılı olarak açıklandığı gibi tanımlamak için bir `keystore.conf` dosyası oluşturun.

2. İleti korumasının yapılandırılması

Bu görev hakkında

`setmqspl` komutunu kullanarak, AGENT2tarafından kullanılan veri kuyruğu için bir güvenlik ilkesi tanımlamanız gerekir. Bu senaryoda, her iki aracıyı da başlatmak için aynı kullanıcı kullanılır, bu nedenle imzalayıcı ve alıcı DN 'si aynı ve oluşturduğumuz sertifikayla eşleşiyor.

Yordam

1. **`fteStopAgent`** komutunu kullanarak koruma hazırlığı için Managed File Transfer aracılarını kapatın.
2. SYSTEM.FTE.DATA.AGENT2 kuyruğunu korumak için bir güvenlik ilkesi oluşturun.

   ```
   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. Managed File Transfer Agent işlemini çalıştıran kullanıcının, sistem ilkesi kuyruğuna göz atma ve iletileri hata kuyruğuna koyma erişimine sahip olduğundan emin olun.

   ```
   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. **fteStartAgent** komutunu kullanarak Managed File Transfer aracılarınızı yeniden başlatın.
5. **fteListAgents** komutunu kullanarak ve aracıların READY durumunda olduğunu doğrulayarak aracılarınızın başarıyla yeniden başlatıldığını doğrulayın.

Sonuçlar

Artık AGENT1 'dan AGENT2' a aktarımları gönderebilirsiniz ve dosya içeriği iki aracı arasında güvenli bir şekilde iletilecektir.

Advanced Message Security Kuruluşa genel bakış

Advanced Message Security bileşenini çeşitli platformlara kurun.

Yordam

- Advanced Message Security ürününü çoklu platformlara kurun.
- IBM MQ Advanced for z/OS ürününü kurma.
- IBM MQ Advanced for z/OS Value Unit Edition ürününü kurma.

İlgili görevler

kaldırmaAdvanced Message Security

z/OS z/OS üzerinde AMS için denetleme

Advanced Message Security (AMS) for z/OS , ilke korumalı kuyruklardaki uygulamalara göre isteğe bağlı işlemlerin denetlenmesi için bir yol sağlar. Etkinleştirildiğinde, ilke korumalı kuyruklarda bu işlemlerin başarılı ve başarısız olması için IBM System Management Facility (SMF) denetim kayıtları oluşturulur. Denetlenen işlemler arasında MQPUT, MQPUT1ve MQGET yer alır.

Denetim varsayılan olarak devre dışı bırakılır, ancak AMS adres alanı için yapılandırılan Dil Ortamı® _CEE_ENVFILE dosyasında _AMS_SMF_TYPE ve _AMS_SMF_AUDIT öğesini yapılandırarak denetimi etkinleştirebilirsiniz. Daha fazla bilgi için bkz. Advanced Message Securityiçin yordamlar oluşturma. _AMS_SMF_TYPE değişkeni, SMF kayıt tipini belirtmek için kullanılır ve 128 ile 255 arasında bir sayıdır. 180 SMF kayıt tipi olağandır, ancak zorunlu değildir. 0 değeri belirtilerek denetim devre dışı bırakılır. _AMS_SMF_AUDIT değişkeni, başarılı olan işlemler, başarısız olan işlemler ya da her ikisi için denetim kayıtlarının oluşturulup oluşturulmayacağını yapılandırır. AMS etkin durumdayken, işletmen komutları kullanılarak denetim seçenekleri de dinamik olarak değiştirilebilir. Daha fazla bilgi için bkz. Çalıştırma Advanced Message Security.

SMF kaydı alt tipler kullanılarak tanımlanır; alt tip 1 genel denetim olayı olur. SMF kaydı, işlenmekte olan istekle ilgili tüm verileri içerir.

SMF kaydı, hedef kitaplık SCSQMACS ' de sağlanan CSQ0KSMF makrosu (makro adındaki sıfır değerini not edin) ile eşlenir. SMF verileri için veri azaltma programları yazıyorsanız, SMF işlem sonrası yordamlarının geliştirilmesine ve özelleştirilmesine yardımcı olmak için bu eşleme makrosunu ekleyebilirsiniz.

Advanced Message Security for z/OStarafından üretilen SMF kayıtlarında veriler bölümler halinde düzenlenir. Kayıt şunlardan oluşur:

- standart bir SMF üstbilgisi
- z/OS için Advanced Message Security tarafından tanımlanan bir üstbilgi uzantısı
- bir ürün bölümü
- bir veri bölümü

SMF kaydının ürün bölümü, Advanced Message Security for z/OStarafından üretilen kayıtlarda her zaman bulunur. Veri bölümü alt tipe göre değişir. Şu anda bir alt tip tanımlanır ve bu nedenle tek bir veri bölümü kullanılır.

SMF, z/OS System Management Facilities (SA22-7630) elkitabında açıklanmaktadır. Geçerli kayıt tipleri, sistem PARMLIB veri kümenizin SMFPRMxx üyesinde açıklanır. Ek bilgi için SMF belgelerine bakın.

Advanced Message Security denetleme raporu üreteci (CSQ0USMF)

Advanced Message Security for z/OS , SCSQAUTH kitaplığında sağlanan CSQ0USMF adlı bir denetleme raporu oluşturma aracı sağlar. CSQ40RSM adlı CSQ0USMF yardımcı programını çalıştırmak için örnek JCL, SCSQPROC kuruluş kitaplığında sağlanır.

CSQ0USMF yardımcı programını çalıştırmadan önce, SMF tipi 180 kayıtlarının sistem SMF veri kümelerinden sıralı bir veri kümesine dökümü alınmalıdır. Örneğin, bu JCL SMF tipi 180 kaydı bir SMF veri kümesinden dökümünü yapar ve bunları bir hedef veri kümesine aktarır:

```
//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))
/*
```

Kuruluşunuz tarafından kullanılan gerçek SMF veri kümesi adlarını doğrulamanız gerekir. Dökümü alınan kayıtlara ilişkin hedef veri kümesinin kayıt biçimi VBS ve kayıt uzunluğu 32760 olmalıdır.

Not: SMF günlük akışları kullanılıyorsa, bir günlük akımının sıralı bir veri kümesine dökümünü almak için IFASMFDL programını kullanmanız gerekir. Kullanılan JCL örneği için İşleme tipi 116 SMF kaydı başlıklı konuya bakın.

Daha sonra hedef veri kümesi, bir AMS denetleme raporu üretmek için CSQ0USMF yardımcı programına giriş olarak kullanılabilir. Örneğin:

```
//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
//
```

CSQ0USMF programı, Çizelge 104 sayfa 651içinde listelenen isteğe bağlı iki parametreyi kabul eder:

Çizelge 104. CSQ0USMF isteğe bağlı değiştirgeler

Değiştirge	**Değer**	**Açıklama**
SMFTYPE	nnn	Denetleme raporu için geçerli SMF kayıt tipi. CSQ0USMF programı, raporu oluştururken yalnızca SMFTYPE değeriyle eşleşen SMF kayıtlarını kullanır. SMFTYPE değerini belirtmezseniz, varsayılan değer olan 180 kullanılır.
M	qmgr	Denetleme raporu için geçerli IBM MQ kuyruk yöneticisi adı. -M parametresini belirtmezseniz, denetleme raporu SMFIN veri kümesinde gösterilen tüm kuyruk yöneticilerine ilişkin tüm denetleme kayıtlarını içerir.

Anahtar depolarının ve sertifikaların AMS ile kullanılması

IBM MQ uygulamalarına şeffaf şifreleme koruması sağlamak için Advanced Message Security , genel anahtar sertifikalarının ve özel anahtarın saklandığı anahtar deposu dosyasını kullanır. z/OSüzerinde, anahtar deposu dosyası yerine SAF anahtar halkası kullanılır.

Advanced Message Securityiçinde kullanıcılar ve uygulamalar, genel anahtar altyapısı (PKI) kimlikleriyle temsil edilir. Bu kimlik tipi, iletileri imzalamak ve şifrelemek için kullanılır. PKI kimliği, imzalı ve şifrelenmiş iletilerle ilişkili bir sertifikada öznenin **ayırt edici adı (DN)** alanıyla gösterilir. Bir kullanıcının ya da uygulamanın iletilerini şifrelemesi için, sertifikaların ve ilişkili özel ve genel anahtarların saklandığı anahtar deposu dosyasına erişilmesi gerekir.

AIX, Linux, and Windowsüzerinde, anahtar deposunun yeri, varsayılan olarak `keystore.conf` olan anahtar deposu yapılandırma dosyasında sağlanır. Her Advanced Message Security kullanıcısının bir anahtar deposu dosyasını gösteren anahtar deposu yapılandırma dosyası olmalıdır. Advanced Message Security , anahtar deposu dosyalarının şu biçimini kabul eder: `.kdb`, `.jceks`, `.jks`.

`keystore.conf` dosyasının varsayılan konumu şöyledir:

- Linux IBM i AIX IBM i, AIX and Linuxsistemlerinde: `$HOME/.mqs/keystore.conf`
- Windows Windowsişletim tarihinde: *%HOMEDRIVE%%HOMEPATH%*`\.mqs\keystore.conf`

Belirtilen bir anahtar deposu dosya adı ve yeri kullanıyorsanız, bunu aşağıdaki örnek komutlarda gösterildiği gibi **MQS_KEYSTORE_CONF** ortam değişkeniyle belirtmeniz gerekir:

- Javaiçin: `java -D`*`MQS_KEYSTORE_CONF=path/filename app_name`*
- C istemcisi ve sunucusu için:
  - AIX and Linuxişletim tarihinde: `export MQS_KEYSTORE_CONF=`*`path/filename`*
  - Windowsişletim tarihinde: `set MQS_KEYSTORE_CONF=`*`path\filename`*

    Not: Windows üzerindeki yol, birden çok sürücü harfi varsa, sürücü harfini belirleyebilir ve belirtmelidir.

keystore.conf dosyasındaki hassas bilgileri koruma

Parolalar gibi anahtar deposu dosyasına duyarlı bilgilere erişmek için, IBM MQ Advanced Message Security ' un (AMS) anahtar deposuna erişebilmesi ve iletileri imzalayabilmesi ve şifreleyebilmesi için belirteçler sağlamanız gerekir.

AMSile verilen **runamscred** komutunu kullanarak anahtar deposu yapılandırma dosyasında bulunan hassas bilgileri korumanız gerekir. Yapılandırma dosyalarının nasıl korunacağına ilişkin ayrıntılar için bkz. “AMS yapılandırma dosyaları için parola korumasının ayarlanması” sayfa 670 .

Parolaları korurken, özel, güçlü bir şifreleme anahtarı kullanmalısınız. Çalıştırma zamanı sırasında parolalara erişmek için bu şifreleme anahtarının AMS' e sağlanması gerekir.

Şifreleme anahtarı dosyasının konumunu sağlamanın iki yöntemi vardır:

- `keystore.conf` dosyasındaki **amscred.keyfile** yapılandırma özelliği
- **MQS_AMSCRED_KEYFILE** ortam değişkeni

Öncelik sırası **MQS_AMSCRED_KEYFILE**, ardından **amscred.keyfile**ve daha sonra varsayılan anahtardır.

İlgili kavramlar

“AMS içinde gönderen ayırt edici adları” sayfa 680
Gönderen ayırt edici adları (DN), bir kuyruğa ileti yerleştirme yetkisi olan kullanıcıları tanımlar. Bir gönderen, iletiyi kuyruğa yerleştirmeden önce, iletiyi imzalamak için sertifikasını kullanır.

“AMS içinde alıcı ayırt edici adları” sayfa 681
Alıcı ayırt edici adları (DN), kuyruktan ileti alma yetkisi olan kullanıcıları tanımlar.

AMS için anahtar deposu yapılandırma dosyasının (keystore.conf) yapısı

Anahtar deposu yapılandırma dosyası (`keystore.conf`), Advanced Message Security ' yi uygun anahtar deposunun konumunu işaret eder.

Aşağıdaki yapılandırma dosyası tiplerinin her birinin bir öneki vardır:

AMSCRED
Parola koruma sistemiyle ilgili parametreler.

CMS
Sertifika Yönetimi Sistemi, yapılandırma girişlerine şu öneki eklenir: `cms.`

PKCS#11
Genel Anahtar Şifreleme Standardı #11, yapılandırma girdilerine şu öneki eklenir: `pkcs11.`

IBM i **PEM**
Gizlilik Gelişmiş Posta biçimi, yapılandırma girdilerinin öneki: `pem.`

JKS
Java KeyStore, yapılandırma girişlerine şu öneki eklenir: `jks.`

JCEKS
Java Şifreleme Şifrelemesi KeyStore, yapılandırma girişlerine şu öneki eklenir: `jceks.`

z/OS MQ Adv. VUE **JCERACFKS**
Java Cryptographic Encryption RACF keyring KeyStore, yapılandırma girişlerine şu öneki eklenir: `jceracfks`.

Önemli: IBM MQ 9.0 ' den `JCEKS.provider` ve `JKS.provider` değerleri yoksayılır. Bouncy Castle sağlayıcısı, kullanılan JRE tarafından sağlanan JCE/JCE ile birlikte kullanılır. Daha fazla bilgi için bkz “AMS ileIBM dışı JRE ' ler için destek” sayfa 657.

Anahtar depolarına ilişkin örnek yapılar:

CMS

```
cms.keystore = /dir/keystore_file
cms.certificate = certificate_label
```

PKCS#11

```
pkcs11.library = dir\cryptoki.dll
pkcs11.certificate = certificatelabel
pkcs11.token = tokenlabel
pkcs11.token_pin = tokenpin
pkcs11.secondary_keystore = dir\signers
```
V 9.3.0 `pkcs11.encrypted = no`

IBM i PEM

```
pem.private = /dir/keystore_file_private_key
pem.public = /dir/keystore_file_public_keys
pem.password = password
```
V 9.3.0 `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
```

Çizelge 105. Her yapılanış kütüğü tipi için gereken değiştirgelerin özeti

Parametreler	Zorunlu	Yapılandırma dosyası tipi				
		Java (PKCS#11, JKS, JCEKS ve JCERACFKS)	**IBM i PEM**	**PKCS#11**	**CMS**	**AMSCRED**
keystore	✓	✓			✓	
IBM i private	✓		IBM i ✓			
IBM i public	✓		IBM i ✓			
IBM i password	✓		IBM i ✓			
library	✓	✓		✓		
certificate	✓	✓		✓	✓	
token	✓	✓		✓		
token_pin	✓	✓		✓		
secondary_keystore	✓	✓		✓		
secondary_keystore_password	✓	✓				

Çizelge 105. Her yapılanış kütüğü tipi için gereken değiştirgelerin özeti (devamı var)						
		Yapılandırma dosyası tipi				
Parametreler	**Zorunlu**	**Java (PKCS#11, JKS, JCEKS ve JCERACFKS)**	**IBM i PEM**	**PKCS#11**	**CMS**	**AMSCRED**
`encrypted`		✓	V 9.3.0 IBM i ✓	V 9.3.0 ✓		
`keystore_pass`	✓	✓				
`key_pass`		✓				
`provider`		✓				
`keyfile`						✓ Siz

simgesini kullanarak yorum ekleyebileceğinizi unutmayın.

Yapılanış kütüğü değiştirgeleri aşağıdaki gibi tanımlanır:

keystore
Yalnızca CMS ve Java yapılandırması.

CMS, JKS ve JCEKS yapılanışına ilişkin anahtar deposu dosyasının yolu.

z/OS MQ Adv. VUE JCERACFKS yapılandırması için RACF anahtarlığına ilişkin URI.

Önemli:

- Anahtar deposu dosyasının yolu dosya uzantısını içermemelidir.
- z/OS MQ Adv. VUE RACF anahtarlığı URI 'si şu biçimde olmalıdır:

```
safkeyring://user/keyring
```

Burada:

– *user* , anahtarlık sahibinin kullanıcı kimliğidir
– *keyring* , anahtarlık adıdır.

IBM i **private**
Yalnızca PEM yapılandırması.

PEM biçiminde özel anahtar ve sertifika içeren bir dosyanın dosya adı.

IBM i **public**
Yalnızca PEM yapılandırması.

PEM biçiminde güvenilen genel sertifikaları içeren bir dosyanın adı.

IBM i **password**
Yalnızca PEM yapılandırması.

Şifrelenmiş bir özel anahtarın şifresini çözmek için kullanılan parola.

V 9.3.0 Bu alanı yerel AMS parola koruma aracını kullanarak korumanız gerekir; bkz. “Parolaları koruma” sayfa 657

library
Yalnızca PKCS#11 .

PKCS#11 kitaplığının yol adı.

certificate
CMS, PKCS#11 ve Java yapılandırması.

Sertifika etiketi.

token
Yalnızca PKCS#11 .

Simge etiketi.

token_pin
Yalnızca PKCS#11 .

Belirtecin kilidini açmak için PIN girin.

Yalnızca Java işlemleri için; bu alanı Java AMS parola koruma aracını kullanarak korumanız gerekir; bkz. “Parolaları koruma” sayfa 657.

V 9.3.0 Yalnızca Yerel işlemler için; bu alanı yerel AMS parola koruma aracını kullanarak korumanız gerekir; bkz. “Parolaları koruma” sayfa 657.

secondary_keystore
Yalnızca PKCS#11 .

PKCS #11 simgesinde saklanan sertifikaların gerektirdiği tutturucu sertifikaları (kök sertifikalar) içeren `.kdb` uzantısı olmadan sağlanan CMS anahtar deposunun yol adı. İkincil anahtar deposu, güven zincirinde ara düzey sertifikaların yanı sıra gizlilik güvenlik ilkesinde tanımlanan alıcı sertifikalarını da içerebilir. Bu CMS anahtar deposuyla birlikte, ikincil anahtar deposuyla aynı dizinde bulunması gereken bir parola saklama dosyası da bulunmalıdır.

Java ortamları için bir JKS anahtar deposu gereklidir ve bir **secondary_keystore_password**sağlamanız gerekir.

secondary_keystore_password
Java PKCS#11 .

`secondary_keystore` özelliği aracılığıyla sağlanan JKS anahtar deposunun parolası. Java AMS parola koruma aracını kullanarak bu alanı korumanız gerekir; bkz. “Parolaları koruma” sayfa 657.

encrypted

Java V 9.3.0 ve, yalnızca IBM MQ 9.3.0içinden PKCS#11 ve IBM i PEM .

Parolanın durumu.

keystore_pass
Yalnızca Java yapılandırması.

Anahtar deposu dosyasının parolası.

Yalnızca Java işlemleri için. Java AMS parola koruma aracını kullanarak bu alanı korumanız gerekir; bkz. “Parolaları koruma” sayfa 657.

key_pass
Yalnızca Java yapılandırması.

Kullanıcının özel anahtarının parolası.

Yalnızca Java işlemleri için; bu alanı Java AMS parola koruma aracını kullanarak korumanız gerekir; bkz. “Parolaları koruma” sayfa 657.

keyfile
Bu yapılandırma dosyasında bulunan parolaları korurken ya da parolaların şifresini çözerken kullanılacak ilk anahtarın konumunu sağlar; bkz. “Parolaları koruma” sayfa 657

provider
Yalnızca Java yapılandırması.

Anahtar deposu sertifikasının gerektirdiği şifreleme algoritmalarını uygulayan Java güvenlik sağlayıcısı.

Önemli: Anahtar deposunda saklanan bilgiler, IBM MQkullanılarak gönderilen güvenli veri akışı için çok önemlidir. Güvenlik yöneticileri bu dosyalara dosya izinleri atarken özellikle dikkat etmelidirler.

Parolaları koruma

`keystore.conf` dosyasında bulunan parolaları ve diğer hassas bilgileri korumanız gerekir. Daha fazla bilgi için bkz. **runamscred**.

`keystore.conf` dosyası örneği:

```
# 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
```

İlgili görevler

“AMS yapılandırma dosyaları için parola korumasının ayarlanması” sayfa 670
Anahtar deposu ve özel anahtar parolalarının düz metin olarak saklanması, Advanced Message Security ' in bir kullanıcının anahtarını kullanarak bu parolaları şifreleyebilecek bir araç sağlaması için bir güvenlik riski oluşturur.

AMS ileIBM dışı JRE ' ler için destek

IBM dışı JRE ' lerle çalışırken IBM MQ classes for Java ve IBM MQ classes for JMS destek Advanced Message Security işlemi.

Advanced Message Security (AMS), Şifreleme İletisi Sözdizimini (CMS)uygular. CMS sözdizimi, isteğe bağlı ileti içeriğini dijital olarak imzalamak, özetlemek, doğrulamak ya da şifrelemek için kullanılır.

IBM MQ 9.0içinden, IBM MQ classes for Java ve IBM MQ classes for JMS içindeki Advanced Message Security desteği, CMS' yi desteklemek için açık kaynak Bouncy Castle paketlerini kullanır. Bu, bu sınıflarınIBM dışı JRE ' lerle çalışırken Advanced Message Security işlemini destekleyebileceği anlamına gelir.

IBM MQ 9.0öncesinde, Advanced Message Security Java istemcilerindeIBM dışı JRE ' lerde desteklenmez. Advanced Message Security support in the IBM MQ classes for Java and IBM MQ classes for JMS depended on CMS support specifically provided by the IBM implementation of the Java Cryptography Extensions (JCE). Bu kısıtlama nedeniyle, işlev yalnızca Java JCE sağlayıcısını içeren bir Java runtime environment (JRE) kullanılırken kullanılabilir.

Bouncy Castle JAR dosyaları için konum ve sürüm numaralandırması

IBM dışı JRE ' ler için destek için gereken Bouncy Castle JAR dosyaları, IBM MQ classes for Java ve IBM MQ classes for JMS kuruluş paketinin bir parçası olarak içerilir.

Kullanılan Bouncy Castle JAR dosyaları aşağıdaki dosyalardır:

Bouncy Castle işlemleri için temel olan sağlayıcı JAR dosyası.

V 9.3.5 Continuous Delivery from IBM MQ 9.3.5için bu JAR dosyası `bcprov-jdk18on.jar`olarak adlandırılır.

LTS IBM MQ 9.3.5öncesinde Long Term Support ve Continuous Delivery için bu JAR dosyası `bcprov-jdk15to18.jar`olarak adlandırılır.

Advanced Message Securitytarafından kullanılan CMS işlemleri desteğini içeren "PKIX" JAR dosyası.

V 9.3.5 Continuous Delivery from IBM MQ 9.3.5için bu JAR dosyası `bcpkix-jdk18on.jar`olarak adlandırılır.

LTS IBM MQ 9.3.5öncesinde Long Term Support ve Continuous Delivery için bu JAR dosyası `bcpkix-jdk15to18.jar`olarak adlandırılır.

Diğer Bouncy Castle JAR dosyaları tarafından kullanılan sınıfları içeren "util" JAR dosyası.

V 9.3.5 Continuous Delivery from IBM MQ 9.3.5için bu JAR dosyası `bcutil-jdk18on.jar`olarak adlandırılır.

LTS IBM MQ 9.3.5öncesinde Long Term Support ve Continuous Delivery için bu JAR dosyası `bcutil-jdk15to18.jar`olarak adlandırılır.

Bağımlılıklar

IBM MQ 9.1 ve sonraki sınıflar IBM JRE 'ler ve Oracle JRE' leri ile sınanmıştır. Bunlar, J2SE-compliant JRE altında da başarılı bir şekilde çalışabilirler. Ancak, aşağıdaki bağımlılıkları not etmelisiniz:

- Advanced Message Security yapılandırmasında değişiklik yok.
- Bouncy Castle sınıfları yalnızca CMS işlemleri için kullanılır. Güvenlikle ilgili diğer tüm işlemler (örneğin, anahtar deposu erişimi, verilerin gerçek şifrelemesi ve imza sağlama toplamlarının hesaplanması) JRE tarafından sağlanan işlevleri kullanır.

  Önemli: Bu nedenle, kullanılan JRE bir JCE sağlayıcısı somutlaması içermelidir.
- Bazı *güçlü* şifreleme algoritmalarını kullanmak için, JRE ' nin JCE uygulamasına ilişkin *sınırsız* ilke dosyalarını kurmanız gerekebilir.

  Daha fazla ayrıntı için JRE belgelerine bakın.
- Java güvenliğini etkinleştirdiyseniz:
  - Bouncy Castle sınıflarının bir güvenlik sağlayıcısı olarak kullanılabilmesi için uygulamaya `java.security.SecurityPermissioninsertProvider.BC` ekleyin.
  - Bouncy Castle JAR dosyalarına `java.security.AllPermission` verin.

    V 9.3.5 IBM MQ 9.3.5' den Continuous Delivery için bu dosyalar şunlardır:

    mq_install_dir`/java/lib/bcutil-jdk18on.jar`
    mq_install_dir`/java/lib/bcpkix-jdk18on.jar`
    mq_install_dir`/java/lib/bcprov-jdk18on.jar`

    LTS Long Term Support ve Continuous Delivery öncesi IBM MQ 9.3.5 için

    mq_install_dir`/java/lib/bcutil-jdk15to18.jar`
    mq_install_dir`/java/lib/bcpkix-jdk15to18.jar`
    mq_install_dir`/java/lib/bcprov-jdk15to18.jar`

İlgili kavramlar

JMS için IBM MQ sınıfları için kurulu olan
Java için IBM MQ sınıfları için kurulu olan

Multi Message Channel Agent (MCA) Müdahalesi ve AMS

MCA kesilmesi, IBM MQ altında çalışan bir kuyruk yöneticisinin, ilkelerin sunucu bağlantısı kanalları için uygulanmasını seçmeli olarak etkinleştirmesini sağlar.

MCA engellemesi, AMS dışında kalan istemcilerin bir kuyruk yöneticisine bağlanmasına ve iletilerinin şifrelenmesine ve şifresinin çözülmesine olanak sağlar.

MCA önleme, istemcide AMS etkinleştirilemediğinde AMS yeteneği sağlamak için tasarlanmıştır. MCA engeli ve AMSkullanabilen bir istemcinin kullanılması, uygulamaların alınması için sorunlu olabilecek iletilerin çift korunmasına neden olur. Daha fazla bilgi için bkz "İstemcide Advanced Message Security ' in devre dışı bırakılması" sayfa 661.

Not: MCA kesicileri AMQP ya da MQTT kanalları için desteklenmez.

Anahtar deposu yapılanış dosyası

Varsayılan olarak, MCA engellemeye ilişkin anahtar deposu yapılanış dosyası `keystore.conf` ' dir ve kuyruk yöneticisini ya da dinleyiciyi başlatan kullanıcının ana dizin yolundaki `.mqs` dizininde bulunur. Anahtar deposu, MQS_KEYSTORE_CONF ortam değişkeni kullanılarak da yapılandırılabilir. AMS anahtar deposunu yapılandırma hakkında daha fazla bilgi için bkz. "Anahtar depolarının ve sertifikaların AMS ile kullanılması" sayfa 652.

MCA engellemesini etkinleştirmek için, anahtar deposu yapılandırma dosyasında kullanmak istediğiniz bir kanalın adını sağlamanız gerekir. MCA Interception için yalnızca bir cms anahtar deposu tipi kullanılabilir.

MCA müdahalesinin ayarlanmasına ilişkin bir örnek için bkz. "AMS için MCA önleme örneği" sayfa 659 .

Uyarı: Yalnızca yetkili istemcilerin bu yeteneğe bağlanabildiğinden ve bu yeteneği kullanabildiğinden emin olmak için, seçilen kanallarda (örneğin, SSL ve SSLPEER ya da CHLAUTH TYPE (SSLPEERMAP) kullanarak) istemci kimlik doğrulamasını ve şifrelemeyi tamamlamanız gerekir.

IBM i

İşletmeniz IBM ikullanıyorsa ve sertifikanızı imzalamak için bir ticari Sertifika Yetkilisi (CA) seçtiyseniz, Digital Certificate Manager PEM (Privacy-Enhanced Mail) biçiminde bir sertifika isteği oluşturur. İsteği seçtiğiniz sertifika kuruluşuna iletmeniz gerekir.

Bunu yapmak için, `channelname`eiçinde belirtilen kanala ilişkin doğru sertifikayı seçmek üzere aşağıdaki komutu kullanmanız gerekir:

```
pem.certificate.channel.channelname
```

AMS için MCA önleme örneği

Bir AMS MCA kesimini nasıl oluşturacağınıza ilişkin örnek görev.

Başlamadan önce

Uyarı: Yalnızca yetkili istemcilerin bu yeteneğe bağlanabildiğinden ve bu yeteneği kullanabildiğinden emin olmak için, seçilen kanallarda (örneğin, SSL ve SSLPEER ya da CHLAUTH TYPE (SSLPEERMAP) kullanarak) istemci kimlik doğrulamasını ve şifrelemeyi tamamlamanız gerekir.

İşletmeniz IBM ikullanıyorsa ve sertifikanızı imzalamak için bir ticari Sertifika Yetkilisi (CA) seçtiyseniz, Digital Certificate Manager PEM (Privacy-Enhanced Mail) biçiminde bir sertifika isteği oluşturur. İsteği seçtiğiniz sertifika kuruluşuna iletmeniz gerekir.

Bu görev hakkında

Bu görev, sisteminizin MCA engelleme özelliğini kullanacak şekilde ayarlanması ve daha sonra, kurulumun doğrulanması işlemleri boyunca size yol gösterir.

Not: IBM MQ, AMS kesicilerini içerir ve bunları MQ istemci ve sunucu çalıştırma zamanı ortamlarında dinamik olarak etkinleştirir.

Uyarı:

- Koddaki `userID` değerini kullanıcı kimliğinizle değiştirin.

- AMS kesmesi istemcide devre dışı bırakılmadıkça, aşağıdaki yordam IBM MQ içinde beklendiği gibi çalışmaz.

Yordam

1. Bir kabuk komut dosyası oluşturmak için aşağıdaki komutları kullanarak anahtar veritabanını ve sertifikaları oluşturun.

   Ayrıca, **INSTLOC** ve **KEYSTORELOC** komutlarını değiştirin ya da gerekli komutları çalıştırın. bobiçin sertifika yaratmanız gerekmeyebileceğini unutmayın.

```
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. Her kullanıcının diğerini başarıyla tanıyabilmesi için sertifikaları iki anahtar veritabanı arasında paylaşın.

   İşletmenizin kullandığı platform için *Hızlı Başlangıç Kılavuzu*' nda sertifikaları paylaşmak için açıklanan yöntemi kullanmanız önemlidir:

   Windows
   Görev 5 Paylaşım sertifikaları

   AIX and Linux
   Görev 5 Paylaşım sertifikaları

   Java müşterileri
   Görev 5 Paylaşım sertifikaları

3. Aşağıdaki yapılandırmayla `keystore.conf` oluşturun: `Keystore.conf location: /home/userID/ssl/ams1/`

```
cms.keystore = /home/userID/ssl/ams1/alicekey
cms.certificate.channel.SYSTEM.DEF.SVRCONN = alice_cert
```

   Uyarı:

   a. Anahtar deposu, kuyruk yöneticisinin bulunduğu sistemde olmalıdır.

   b. MCA müdahalesini etkinleştirmek için `cms.certificate` için belirli bir kanal belirtmeniz gerekir; daha sonra kuyruk yöneticisi, bu kanaldan belirlenen ilkelerle kuyruklara bağlanan uygulamalarda AMS işlemleri gerçekleştirir.

4. Kuyruk yöneticisi yaratılması ve başlatılması `AMSQMGR1`
5. QMGR denetimi altında kullanılabilir bir kapı numarası kullanarak bir TCP dinleyicisi tanımlayın.

   Örneğin:

```
DEFINE LISTENER(MY.LISTENER) TRPTYPE(TCP) PORT(14567) CONTROL(QMGR)
```

6. Dinleyiciyi başlatın ve doğru başlatıldığını doğrulayın.

   Örneğin:

```
START  LISTENER(MY.LISTENER)
DISPLAY LSSTATUS(MY.LISTENER) PORT
```

7. Kuyruk yöneticisini durdurun.

8. Anahtar deposunu ayarla:

```
export MQS_KEYSTORE_CONF=/home/userID/ssl/ams1/keystore.conf
```

9. Kuyruk yöneticisini aynı kabukta başlatın; böylece MQS_KEYSTORE_CONF ortam değişkeni kuyruk yöneticisinin kullanımına sunulur.
10. Güvenlik ilkesini ayarlayın ve doğrulayın:

```
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
```

Ek bilgi için setmqspl ve dspmqspl başlıklı konuya bakın.

11. *MQSERVER* ortam değişkenini ayarlayın:

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/127.0.0.1(14567)'
```

12. Güvenlik ilkesini kaldırın ve sonucu doğrulayın:

```
setmqspl -m AMSQMGR1 -p TESTQ -remove
dspmqspl -m AMSQMGR1
```

13. IBM MQ 9.3 kuruluşunuzda kuyruğa göz atın:

```
/opt/mq93/samp/bin/amqsbcg TESTQ AMSQMGR1
```

Göz atma çıkışı, iletileri şifrelenmiş biçimde gösterir.

14. Güvenlik ilkesini ayarlayın ve sonucu doğrulayın:

```
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. **amqsgetc** komutunu IBM MQ 9.3 kuruluşundan çalıştırın:

```
/opt/mqm/samp/bin/amqsgetc TESTQ TESTQMGR
```

İlgili kavramlar

“AMS için anahtar deposu yapılandırma dosyasının (keystore.conf) yapısı” sayfa 653
Anahtar deposu yapılandırma dosyası (keystore.conf), Advanced Message Security ' yi uygun anahtar deposunun konumunu işaret eder.

İlgili başvurular

“Bilinen AMS sınırlamaları” sayfa 612
Advanced Message Securityiçin desteklenmeyen ya da sınırlamaları olan birçok IBM MQ seçeneği vardır.

İstemcide Advanced Message Security ' in devre dışı bırakılması

Ürünün önceki bir sürümünden bir kuyruk yöneticisine bağlanmak için bir IBM MQ istemcisi kullanıyorsanız ve 2085 (MQRC_UNKNOWN_OBJECT_NAME) hatası bildirildiyse, IBM MQ Advanced Message Security (AMS) öğesini devre dışı bırakmanız gerekir.

Bu görev hakkında

IBM MQ Advanced Message Security (AMS), bir IBM MQ istemcisinde otomatik olarak etkinleştirilir ve varsayılan olarak istemci, kuyruk yöneticisindeki nesnelere ilişkin güvenlik ilkelerini denetlemeye çalışır.

Bu hata bildirilirse, ürünün önceki bir sürümünden bir kuyruk yöneticisine bağlanmaya çalışırken AMS ' i aşağıdaki gibi devre dışı bırakabilirsiniz:

- Java istemcileri için aşağıdaki yollardan herhangi biri:
  - **AMQ_DISABLE_CLIENT_AMS**ortam değişkenini ayarlayarak.
  - Java sistem özelliği com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMSayarlanarak.

  - **DisableClientAMS** özelliğini kullanarak, `mqclient.ini` dosyasındaki Güvenlik kısmı altında.
- C istemcileri için, **MQS_DISABLE_ALL_INTERCEPT**ortam değişkenini ayarlayarak.

Not: C istemcileri için **AMQ_DISABLE_CLIENT_AMS** ortam değişkenini kullanamazsınız. Bunun yerine **MQS_DISABLE_ALL_INTERCEPT** ortam değişkenini kullanmanız gerekir.

Yordam

- İstemcide AMS özelliğini geçersiz kılmak için aşağıdaki seçeneklerden birini kullanın:

  AMQ_DISABLE_CLIENT_AMS ortam değişkeni
  Aşağıdaki durumlarda bu değişkeni ayarlamanız gerekir:

  - IBM Java runtime environment (JRE) dışında bir Java runtime environment (JRE) kullanıyorsanız
  - IBM MQ IBM MQ classes for JMS ya da IBM MQ classes for Java istemcisi kullanıyorsanız.

  AMQ_DISABLE_CLIENT_AMS ortam değişkenini oluşturun ve uygulamanın çalıştığı ortamda TRUE değerine ayarlayın. Örneğin:

  ```
  export AMQ_DISABLE_CLIENT_AMS=TRUE
  ```

  Java sistem özelliği com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS
  IBM MQ classes for JMS ve IBM MQ classes for Java istemcileri için, com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS Java sistem özelliğini Java uygulaması için TRUE değerine ayarlayabilirsiniz.

  Örneğin, Java komutu çağrıldığında Java sistem özelliğini `-D` seçeneği olarak ayarlayabilirsiniz:

  ```
  V 9.3.0 V 9.3.0 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
  ```

  Diğer bir seçenek olarak, uygulama bu dosyayı kullanıyorsa, bir JMS yapılanış dosyasında `jms.config` Java sistem özelliğini belirtebilirsiniz.

  MQS_DISABLE_ALL_INTERCEPT ortam değişkeni
  Yerli istemcilerle IBM MQ kullanıyorsanız ve istemcide AMS geçersiz kılmanız gerekiyorsa, bu ortam değişkenini ayarlamanız gerekir.

  MQS_DISABLE_ALL_INTERCEPT ortam değişkenini oluşturun ve istemcinin çalıştığı ortamda TRUE değerine ayarlayın. Örneğin:

  ```
  export MQS_DISABLE_ALL_INTERCEPT =TRUE
  ```

  MQS_DISABLE_ALL_INTERCEPT ortam değişkenini yalnızca C istemcileri için kullanabilirsiniz. Java istemcileri için bunun yerine **AMQ_DISABLE_CLIENT_AMS** ortam değişkenini kullanmanız gerekir.

  mqclient.ini dosyasında DisableClientAMS özelliği
  Bu seçeneği IBM MQ classes for JMS ve IBM MQ classes for Java istemcileri ve C istemcileri için kullanabilirsiniz.

  Aşağıdaki örnekte gösterildiği gibi, **Security** kısmı `mqclient.ini` dosyasının altına `DisableClientAMS` özellik adını ekleyin:

  ```
  Security:
  DisableClientAMS=Yes
  ```

AMS özelliğini aşağıdaki örnekte gösterildiği gibi etkinleştirebilirsiniz:

```
Security:
DisableClientAMS=No
```

Sonraki adım

AMS korumalı kuyruklarının açılmasına ilişkin daha fazla bilgi için bkz. JMSile AMS kullanılırken korunan kuyruklar açılırken sorunlar.

İlgili kavramlar

"Message Channel Agent (MCA) Müdahalesi ve AMS" sayfa 658
MCA kesilmesi, IBM MQ altında çalışan bir kuyruk yöneticisinin, ilkelerin sunucu bağlantısı kanalları için uygulanmasını seçmeli olarak etkinleştirmesini sağlar.

İlgili görevler

IBM MQ MQI client yapılandırma dosyası, `mqclient.ini`

İlgili başvurular

IBM MQ classes for JMS yapılandırma dosyası

AMS için sertifika gereksinimleri

Sertifikaların Advanced Message Securityile kullanılabilmesi için bir RSA ortak anahtarı olmalıdır.

Farklı genel anahtar tipleri ve bunların nasıl oluşturulacağına ilişkin daha fazla bilgi için bkz. "IBM MQ içinde dijital sertifikalar ve CipherSpec uyumluluğu" sayfa 46.

Anahtar kullanım uzantıları

Anahtar kullanım uzantıları, bir sertifikanın kullanılabilmesi için ek kısıtlamalar uygular.

Advanced Message Securityiçinde, X.509 v3 sertifikalarının anahtar kullanımı RFC 5280 belirtimine uygun olarak ayarlanmalıdır.

Koruma bütünlüğünün kalitesi için, sertifika anahtarı kullanım uzantıları ayarlandıysa, bu küme aşağıdakilerden en az birini içermelidir:

- **`nonRepudiation`**
- **`digitalSignature`**

Koruma gizliliğinin kalitesi için, sertifika anahtarı kullanım uzantıları ayarlandıysa, bu küme aşağıdakileri içermelidir:

- **`keyEncipherment`**

Koruma gizliliğinin kalitesi için, sertifika anahtarı kullanım uzantıları ayarlandıysa, bu küme aşağıdakileri içermelidir:

- **`dataEncipherment`**

Genişletilmiş anahtar kullanımı, anahtar kullanım uzantılarını daha da iyileştirir. Tüm koruma nitelikleri için, sertifika genişletilmiş anahtar kullanımı ayarlanırsa, kümenin aşağıdakileri içermesi gerekir:

- **`emailProtection`**

İlgili kavramlar

"AMS içinde koruma kalitesi" sayfa 683
Advanced Message Security veri koruma ilkeleri, bir koruma kalitesi (QOP) anlamına gelir.

AMS içinde sertifika doğrulama yöntemleri

Kuyruklarınızdaki iletilerin güvenlik standartlarını karşılamayan sertifikalar kullanılarak korunmaması için iptal edilen sertifikaları saptamak ve reddetmek için Advanced Message Security komutunu kullanabilirsiniz.

AMS , Çevrimiçi Sertifika Durumu İletişim Kuralı (OCSP) ya da sertifika iptal listesi (CRL) kullanarak bir sertifika geçerliliğini doğrulamanızı sağlar.

AMS , OCSP ve/veya CRL denetimi için yapılandırılabilir. Her iki yöntem de etkinleştirildiyse, performans nedeniyle AMS önce iptal durumu için OCSP kullanır. OCSP denetiminden sonra bir sertifikanın iptal durumu belirlenmezse, AMS CRL denetimini kullanır.

Hem OCSP hem de CRL denetiminin varsayılan olarak etkinleştirildiğini unutmayın.

İlgili kavramlar

"AMS içinde OCSP (Online Certificate Status Protocol; Çevrimiçi Sertifika Durumu İletişim Kuralı)" sayfa 664

Çevrimiçi Sertifika Durumu İletişim Kuralı (OCSP), bir sertifikanın iptal edilip edilmediğini belirler ve bu nedenle sertifikanın güvenilir olup olmadığını belirlemeye yardımcı olur. Varsayılan olarak OCSP etkindir.

"AMS içindeki sertifika iptal listeleri (CRL ' ler)" sayfa 666

CRL ' ler Sertifika Yetkilisi (CA) tarafından çeşitli nedenlerden ötürü artık güvenilir olarak işaretlenmeyen sertifikaların bir listesini içerir; örneğin, özel anahtar kaybolmuş ya da tehlikeye atılmıştır.

AMS içinde OCSP (Online Certificate Status Protocol; Çevrimiçi Sertifika Durumu İletişim Kuralı)

Çevrimiçi Sertifika Durumu İletişim Kuralı (OCSP), bir sertifikanın iptal edilip edilmediğini belirler ve bu nedenle sertifikanın güvenilir olup olmadığını belirlemeye yardımcı olur. Varsayılan olarak OCSP etkindir.

IBM i sistemleri üzerinde OCSP desteklenmez.

Advanced Message Security yerel kesicileri içinde OCSP denetiminin etkinleştirilmesi

Kullanılmakta olan sertifikalarda yer alan bilgilere dayalı olarak, Advanced Message Security ' da çevrimiçi Sertifika Durumu İletişim Kuralı (OCSP) geri verme varsayılan olarak etkindir.

Yordam

Anahtar deposu yapılanış dosyasına aşağıdaki seçenekleri ekleyin:

Not: Tüm OCSP kısmı isteğe bağlıdır ve bağımsız olarak belirtilebilir.

Seçenek	Açıklama
`ocsp.enable=off`	Denetlenmekte olan sertifikanın PKIX_AD_OCSP erişim yöntemine sahip bir AIA (Authority Info Access; Yetki Bilgisi Erişimi) Uzantısı olup olmadığını denetleyen OCSP ' yi etkinleştirin. Olası değerler: on ya da `off`.
`ocsp.url=`*`responder_URL`*	OCSP yanıtlayıcısının URL adresi. Bu seçenek atlanırsa, AIA dışı OCSP denetimi geçersiz kılınır.
`ocsp.http.proxy.host=`*`OCSP_proxy`*	OCSP yetkili sunucusunun URL adresi. Bu seçenek atlanırsa, AIA dışı çevrimiçi sertifika denetimleri için yetkili sunucu kullanılmaz.
`ocsp.http.proxy.port=`*`port_number`*	OCSP yetkili sunucusunun kapı numarası. Bu seçenek atlanırsa, varsayılan 8080 kapısı kullanılır.
`ocsp.nonce.generation=on/off`	OCSP sorgulanırken nonce oluştur. Varsayılan değer `off`değeridir.
`ocsp.nonce.check=on/off`	OCSP ' den yanıt aldıktan sonra nonce 'yi denetleyin. Varsayılan değer `off`değeridir.

Seçenek	Açıklama
`ocsp.nonce.size=8`	Bayt cinsinden bir kerelik büyüklük.
`ocsp.http.get=on/off`	İstek yönteminiz olarak HTTP GET değerini belirtin. Bu seçenek `off`olarak ayarlanırsa, HTTP POST kullanılır. Varsayılan değer `off`. değeridir.
`ocsp.max_response_size=20480`	Bayt cinsinden sağlanan OCSP yanıtlayıcısından yanıt boyutu üst sınırı.
`ocsp.cache_size=100`	İç OCSP yanıtının önbelleğe alınmasını etkinleştirin ve önbellek girişi sayısı sınırını ayarlayın.
`ocsp.timeout=30`	Advanced Message Security zamanaşımına uğradıktan sonra sunucu yanıtını bekleme süresi (saniye).
`ocsp.unknown=ACCEPT`	Bir OCSP sunucusuna zamanaşımı süresi içinde erişilemediğinde kullanılacak davranışı tanımlar. Olası değerler: • `ACCEPT` Sertifikanın • `WARN` Sertifikaya izin verir ve bir uyarı günlüğe kaydeder • `REJECT` Sertifikanın kullanılmasını önler ve bir hatayı günlüğe kaydeder

AMS içinde Java OCSP geri verilmesinin etkinleştirilmesi
Advanced Message Securityiçinde Java için OCSP denetimini etkinleştirmek üzere `java.security` kütüğünü ya da anahtar deposu yapılanış kütüğünü değiştirin.

Bu görev hakkında

OCSP 'nin Advanced Message Security' de geri verilmesini etkinleştirmenin iki yolu vardır:

java.security kullanılması
Sertifikanız Yetkili Bilgi Erişimi (AIA) sertifika uzantısı içerip içermediğini denetleyin.

Yordam

1. AIA ayarlanmamışsa ya da sertifikanızı geçersiz kılmak istiyorsanız, `$JAVA_HOME/lib/security/java.security` dosyasını aşağıdaki özelliklerle düzenleyin:

   ```
   ocsp.responderURL=http://url.to.responder:port
   ocsp.responderCertSubjectName=CN=Example CA,O=IBM,C=US
   ```

   ve `$JAVA_HOME/lib/security/java.security` kütüğünü aşağıdaki satırla düzenleyerek OCSP denetimini etkinleştirin:

   ```
   ocsp.enable=true
   ```

2. AIA ayarlandıysa, `$JAVA_HOME/lib/security/java.security` dosyasını aşağıdaki satırla düzenleyerek OCSP denetimini etkinleştirin:

   ```
   ocsp.enable=true
   ```

Sonraki adım

Java Security Manager kullanıyorsanız, yapılandırmayı çok fazla tamamlayın, `lib/security/java.policy` için aşağıdaki Java iznini ekleyin

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
```

keystore.conf dosyasının kullanılması

Yordam

Yapılandırma dosyasına şu özniteliği ekleyin:

```
ocsp.enable=true
```

Önemli: Yapılanış dosyasında bu özniteliğin ayarlanması java.security ayarlarını geçersiz kılar.

Sonraki adım

Yapılandırmayı tamamlamak için aşağıdaki Java izinlerini `lib/security/java.policy`dizinine ekleyin:

```
permission java.security.SecurityPermission "getProperty.ocsp.enable";
permission java.security.SecurityPermission "setProperty.ocsp.enable";
```

AMS içindeki sertifika iptal listeleri (CRL ' ler)

CRL ' ler Sertifika Yetkilisi (CA) tarafından çeşitli nedenlerden ötürü artık güvenilir olarak işaretlenmeyen sertifikaların bir listesini içerir; örneğin, özel anahtar kaybolmuş ya da tehlikeye atılmıştır.

Sertifikaları doğrulamak için Advanced Message Security , imzalayanın sertifikasını ve sertifika yetkilisinin (CA ' lar) sertifika zincirinden oluşan bir sertifika zinciri oluşturur. Güvenilirlik çıpası, bir sertifikanın güvenilirliği için kullanılan güvenilir bir sertifika ya da güvenilir bir kök sertifika içeren güvenilir bir anahtar deposu dosyasıdır. AMS , PKIX doğrulama algoritmasını kullanarak sertifika yolunu doğrular. Zincir oluşturulup doğrulandığında AMS , zincirdeki her sertifikanın çıkışının ve süre bitim tarihinin geçerli tarihe göre doğrulanmasını içeren sertifika doğrulamasını tamamlar ve Son Varlık sertifikasında anahtar kullanım uzantısının olup olmadığını denetler. Uzantı sertifikanın sonuna eklenirse, AMS **digitalSignature** ya da **nonRepudiation** ' ın da ayarlanıp ayarlanmadığını doğrular. Değilse, `MQRC_SECURITY_ERROR` raporlanır ve günlüğe kaydedilir. Daha sonra AMS , yapılandırma dosyasında belirtilen değerlere bağlı olarak CRL 'leri dosyalardan ya da LDAP' den yükler. Yalnızca DER biçiminde kodlanmış CRL ' ler AMStarafından desteklenir. Anahtar deposu yapılandırma dosyasında CRL ile ilgili bir yapılandırma bulunamazsa, AMS CRL geçerlilik denetimi gerçekleştirmez. Her CA sertifikası için AMS , CRL 'lerini bulmak üzere bir CA' nın Ayırt Edici Adlarını kullanarak LDAP 'ı CRL' ler için sorgular. LDAP sorgusunda aşağıdaki öznitelikler bulunur:

```
certificateRevocationList,
certificateRevocationList;binary,
authorityRevocationList,
authorityRevocationList;binary
deltaRevocationList
deltaRevocationList;binary,
```

Not: `deltaRevocationList` yalnızca dağıtım noktaları olarak belirtildiğinde desteklenir.

Yerel kesicilerdeki sertifika geçerlilik denetimi ve sertifika iptal listesi desteğinin etkinleştirilmesi
Advanced Message Security 'in LDAP (Lightweight Directory Access Protocol; Temel Dizin Erişimi Protokolü) sunucusundan CLR' leri yükleyebilmesi için anahtar deposu yapılandırma dosyasını değiştirmeniz gerekir.

Bu görev hakkında

IBM i Yerel kesicilerdeki sertifika geçerlilik denetimi ve sertifika iptal listesi desteğinin etkinleştirilmesi, IBM iüzerindeki Advanced Message Security için desteklenmez.

Yordam

Yapılanış kütüğüne aşağıdaki seçenekleri ekleyin:

Not: Tüm CRL kısmı isteğe bağlıdır ve bağımsız olarak belirtilebilir.

Seçenek	Açıklama
`crl.ldap.host=`*`host_name`*	LDAP sunucusu anasistem adı.
`crl.ldap.port=`*`port_number`*	LDAP sunucusu kapı numarası. En çok 11 sunucu belirleyebilirsiniz. LDAP bağlantı hatası durumunda saydam hata durumunda yedek sisteme geçiş sağlamak için birden çok LDAP anasistemi kullanılır. Tüm LDAP sunucularının eşleme olması ve aynı verileri içermesi beklenir. AMS Java kesici bir LDAP sunucusuna başarıyla bağlandığında, sağlanan diğer sunuculardan CRL ' leri yüklemeye çalışmaz.
`crl.cdp=off`	Sertifikalarda CRLDistributionPoints uzantılarını denetlemek ya da kullanmak için bu seçeneği kullanın.
`crl.ldap.version=3`	LDAP iletişim kuralı sürüm numarası. Olası değerler: 2 ya da 3.
`crl.ldap.user=cn=`*`username`*	LDAP sunucusunda oturum açın. Bu değer belirtilmezse, LDAP ' taki CRL öznitelikleri okunabilir olmalıdır
`crl.ldap.pass=`*`password`*	LDAP sunucusunun parolası.
V 9.3.0 `crl.ldap.encrypted=`*`no/yes`*	`clr.ldap.pass` şifrelenmiş olsun ya da olmasın. Daha fazla bilgi için AMS yapılandırma dosyalarında parolaları koruma başlıklı konuya bakın.
`crl.ldap.cache_lifetime=0`	LDAP önbelleği geçerlilik süresi (saniye). Olası değerler: 0-86400.
`crl.ldap.cache_size=50`	LDAP önbellek boyutu. Bu seçenek yalnızca `crl.ldap.cache_lifetime` değeri `0`değerinden büyükse belirtilebilir.
`crl.http.proxy.host=some.host.com`	CDP CRL alımı için http yetkili sunucu kapısı.
`crl.http.proxy.port=8080`	Http yetkili sunucusu kapı numarası.
`crl.http.max_response_size=204800`	IBM Global Security Kit (GSKit)tarafından kabul edilen bir HTTP sunucusundan alınabilecek, bayt cinsinden CRL boyutu üst sınırı.
`crl.http.timeout=30`	AMS zamanaşımına uğradıktan sonra sunucu yanıtını bekleme süresi (saniye).
`crl.http.cache_size=0`	Bayt cinsinden HTTP önbellek boyutu.

Seçenek	Açıklama
`crl.unknown=ACCEPT`	Bir CRL sunucusuna zamanaşımı süresi içinde erişilemediğinde kullanılacak davranışı tanımlar. Olası değerler: • ACCEPT Sertifikanın • WARN Sertifikaya izin verir ve bir uyarı günlüğe kaydeder • REJECT Sertifikanın kullanılmasını önler ve bir hatayı günlüğe kaydeder

AMS içinde Java içinde sertifika iptal listesi desteğini etkinleştirme
Advanced Message Securityiçinde CRL desteğini etkinleştirmek için, anahtar deposu yapılandırma dosyasını AMS 'un LDAP (Lightweight Directory Access Protocol; Temel Dizin Erişimi Protokolü) sunucusundan CRL' leri karşıdan yüklemesine izin verecek şekilde değiştirmeniz ve java.security dosyasını yapılandırmanız gerekir.

Yordam

1. Yapılanış kütüğüne aşağıdaki seçenekleri ekleyin:

Üstbilgi	Açıklama
`crl.ldap.host=`*`host_name`*	LDAP anasistem adı.
`crl.ldap.port=`*`port_number`*	LDAP sunucusu kapı numarası. En çok 11 sunucu belirleyebilirsiniz. LDAP bağlantı hatası durumunda saydam hata durumunda yedek sisteme geçiş sağlamak için birden çok LDAP anasistemi kullanılır. Tüm LDAP sunucularının eşleme olması ve aynı verileri içermesi beklenir. AMS Java kesici bir LDAP sunucusuna başarıyla bağlandığında, sağlanan diğer sunuculardan CRL ' leri yüklemeye çalışmaz. Java , `crl.ldap.user` ve crl.ldaworldp.pass değerlerini kullanmaz. Bir LDAP sunucusuna bağlanırken kullanıcı ve parola kullanmaz. Sonuç olarak, LDAP ' taki CRL öznitelikleri okunabilir olmalıdır.
`crl.cdp=on/off`	Sertifikalarda CRLDistributionPoints uzantılarını denetlemek ya da kullanmak için bu seçeneği kullanın.

2. `JRE/lib/security/java.security` dosyasını aşağıdaki özelliklerle değiştirin:

Özellik Adı	Açıklama
`com.ibm.security.enableCRLDP`	Bu özellik şu değerleri alır: `true`, `false`. `true`olarak ayarlanırsa, sertifika iptal denetimi yapılırken CRL ' ler sertifikanın CRL dağıtım noktaları uzantısından URL kullanılarak bulunur. `false` olarak ayarlanırsa ya da ayarlanmazsa, CRL dağıtım noktaları uzantısını kullanarak CRL ' nin denetlenmesi devre dışı bırakılır.

Özellik Adı	Açıklama
`ibm.security.certpath.ldap.cache.lifetime`	Bu özellik, LDAP CertStore önbelleğindeki girişlerin yaşam süresini saniye cinsinden bir değere ayarlamak için kullanılabilir. 0 değeri önbelleği devre dışı bırakır; -1 değeri, sınırsız kullanım süresi anlamına gelir. Ayarlanmazsa, varsayılan geçerlik süresi 30 saniyedir.
`com.ibm.security.enableAIAEXT`	Bu özellik şu değerleri alır: `true`, `false`. `true`olarak ayarlanırsa, oluşturulmakta olan sertifika yolunun sertifikalarında bulunan Yetki Bilgileri Erişimi uzantıları, bunların LDAP URI ' leri içerip içermediğini belirlemek için incelenir. Bulunan her LDAP URI için bir LDAPCertStore nesnesi oluşturulur ve sertifika yolunu oluşturmak için gerekli olan diğer sertifikaları bulmak için kullanılan CertStores derlemine eklenir. `false` olarak ayarlanırsa ya da ayarlanmazsa, ek LDAPCertStore nesneleri oluşturulmaz.

z/OS *z/OS üzerinde sertifika iptal listelerinin (CRL) etkinleştirilmesi*

Advanced Message Security , veri iletilerini korumak için kullanılan sayısal sertifikaların Sertifika İptal Listesi (CRL) denetimini destekler

Bu görev hakkında

Etkinleştirildiğinde, Advanced Message Security , iletiler gizlilik korumalı bir kuyruğa yerleştirildiğinde alıcı sertifikalarını doğrular ve iletiler korunan bir kuyruktan (bütünlük ya da gizlilik) alındığında gönderen sertifikalarını doğrular. Bu durumda doğrulama, ilgili sertifikaların ilgili CRL ' de kayıtlı olmadığının doğrulanmasını içerir.

Advanced Message Security , gönderen ve alıcı sertifikalarını doğrulamak için IBM Sistem SSL hizmetlerini kullanır. Sistem SSL sertifikası doğrulamasıyla ilgili ayrıntılı belgeleri z/OS Cryptographic Services System Secure Sockets Layer Programming elkitabında bulabilirsiniz.

CRL denetimini etkinleştirmek için, AMS adres alanı için başlatılan JCL görevinde CRLFILE DD aracılığıyla bir CRL yapılandırma dosyasının konumunu belirtirsiniz. Uyarlanabilecek örnek bir CRL yapılanış kütüğü *thlqual*.SCSQPROC (CSQ40CRL) içinde bulunur. Bu dosyada izin verilen ayarlar şunlardır:

Çizelge 106. Advanced Message Security CRL yapılandırma değişkenleri

Değişken	Geçerli değerler	Açıklama
crl.ldap.host[.n]	*hostname -or-hostname: port*	Sertifika veren sertifikalarınızın CRL ' lerini barındıran LDAP sunucunuzun ipaddr/hostname (IP adresi/anasistem adı). LDAP sunucunuz için bir kapı numarası belirtmezseniz, crl.ldap.port ile belirtilen kapı numarası kullanılır.
crl.ldap.port	*kapı*	LDAP sunucunuzun TCP/IP kapı numarası.
crl.ldap.user	*ldap_user*	LDAP sunucusuna bağlanırken kullanılacak LDAP kullanıcı adı.

Çizelge 106. Advanced Message Security CRL yapılandırma değişkenleri (devamı var)		
Değişken	**Geçerli değerler**	**Açıklama**
crl.ldap.pass	*ldap_parolası*	crl.ldap.userile ilişkili LDAP parolası.

Aşağıdaki gibi birden çok LDAP sunucusu anasistem adı ve kapısı belirtebilirsiniz:

```
crl.ldap.host.1 = hostname -or hostname:port
crl.ldap.host.2 = hostname -or hostname:port
crl.ldap.host.3 = hostname -or hostname:port
```

En çok 10 anasistem adı belirleyebilirsiniz. LDAP sunucularınız için bir kapı numarası belirtmezseniz, crl.ldap.port ile belirtilen kapı numarası kullanılır. Her LDAP sunucusu, erişim için aynı crl.ldap.user/password birleşimini kullanmalıdır.

CRL FILE DD belirtildiğinde, yapılandırma Advanced Message Security adres alanının kullanıma hazırlanması sırasında yüklenir ve CRL denetimi etkinleştirilir. CRL FILE DD belirtilmezse ya da CRL yapılandırma dosyası kullanılamıyorsa ya da geçersizse, CRL denetimi devre dışı bırakılır.

AMS , IBM Sistem SSL sertifikası doğrulama hizmetlerini kullanarak bir CRL denetimi gerçekleştirir:

Çizelge 107. Advanced Message Security CRL denetimleri		
İşlem	**Koruma kalitesi**	**Sertifika (lar) denetlendi**
PUT	Gizlilik İlkeleri	Alıcılar
GET	Bütünlük/Gizlilik	Gönderen

Bir ileti işlemi başarısız olursa, CRL denetimi Advanced Message Security aşağıdaki işlemleri gerçekleştirir:

Çizelge 108. Advanced Message Security CRL denetimi hata davranışı	
İşlem	**CRL denetleme hatası**
PUT	İleti hedef kuyruğa konmadı. Uygulamaya bir MQCC_FAILED tamamlanma kodu ve MQRC_SECURITY_ERROR neden kodu döndürüldü.
GET	İleti hedef kuyruktan kaldırılır ve sistem koruma hata kuyruğuna taşınır. Uygulamaya bir MQCC_FAILED tamamlanma kodu ve MQRC_SECURITY_ERROR neden kodu döndürüldü.

AMS for z/OS , CRL ve güven denetimini içeren sertifikaları doğrulamak için IBM Sistem SSL hizmetlerini kullanır.

IBM MQ , sertifika geçerlilik denetiminin LDAP sunucusuyla iletişim kurulmasını gerektirdiği, ancak CRL ' nin tanımlanmasını gerektirmediği bir güvenlik ayarını kullanır.

Not: İlgili LDAP hizmetlerinin kullanılabilir olmasını sağlamak ve ilgili Sertifika Yetkilileri için CRL girdilerini korumak yöneticilerin sorumluluğundadır.

AMS yapılandırma dosyaları için parola korumasının ayarlanması

Anahtar deposu ve özel anahtar parolalarının düz metin olarak saklanması, Advanced Message Security ' in bir kullanıcının anahtarını kullanarak bu parolaları şifreleyebilecek bir araç sağlaması için bir güvenlik riski oluşturur.

Başlamadan önce

`keystore.conf` dosya sahibi, yalnızca dosya sahibinin dosyayı okuma ve dosyaya yazma yetkisine sahip olduğundan emin olmalıdır. Bu konuda açıklanan parola koruması, yalnızca ek bir koruma ölçüsüdür. Ayrıca, bu yordamı güvenli bir sistemde gerçekleştirmeniz gerekir.

V 9.3.0 Yapılandırma dosyasını okuyacak AMS istemcisi tipi için doğru **runamscred** değişkenini kullandığınızdan emin olun. AMS istemcisi bir:

- Java istemcisi, `<IBM MQ installation root>/java/bin` dizininde bulunan Java **runamscred** komutunu kullanmalısınız.
- MQI istemcisi, `<IBM MQ installation root>/bin` içinde bulunan MQI **runmqascred** komutunu kullanmalısınız.

Yordam

1. `keystore.conf` dosyalarını, koruma gerektiren parolalar da içinde olmak üzere tüm gerekli bilgileri içerecek şekilde düzenleyin.

```
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. Parolaları, `keystore.conf` dosyasını koruyan kullanıcının erişebileceği bir dosyanın içine şifrelemek için şifreleme anahtarını yerleştirin.

   V 9.3.0 Bu anahtar, daha sonra AMS istemcisi tarafından kullanılacak anahtarla aynı olmalıdır:

```
ThisIsAnExampleEncryptionKey
```

3. Şifreleme anahtarı dosyasını sağlayan `keystore.conf` dosyasını korumak için **runamscred** komutunu çalıştırın.

```
runamscred -f <location of keystore.conf> -sf <location of encryption keyfile>
```

4. `keystore.conf` dosyasının korunduğunu ve şifrelenmiş parolalar içerdiğini doğrulayın.

Örnek

Aşağıdaki örnek, korumalı `keystore.conf` dosyasının nasıl göründüğünü göstermektedir:

```
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
```

İlgili bilgiler

runamscred: AMS anahtar sözcüklerini koru

z/OS z/OS üzerinde AMS ile sertifikaların kullanılması

Bu görev hakkında

Advanced Message Security , üç koruma düzeyi uygular: bütünlük, gizlilik ve gizlilik.

Bir bütünlük ilkesiyle, iletiler, yaratıcının özel anahtarı (MQPUT ' yi yapan uygulama) kullanılarak imzalanır. Bütünlük, ileti değişikliği algılamasını sağlar, ancak ileti metninin kendisi şifrelenmez.

Bir gizlilik ilkesiyle, ileti kuyruğa yerleştirildiğinde şifrelenir. İleti, ilgili Advanced Message Security ilkesinde belirtilen bir algoritma ve simetrik anahtar kullanılarak şifrelenir. Simetrik anahtarın kendisi, her bir alıcının genel anahtarıyla (MQGET ' i yapan uygulama) şifrelenir. Ortak anahtarlar, anahtar halkalarında saklanan sertifikalarla ilişkilendirilir.

Bir gizlilik politikası ile mesajlar hem imzalanmış hem de şifrelenmiş olur.

Gizlilikle korunan bir ileti MQGET işlemi yapan bir alıcı uygulaması tarafından kuyruğa alındığında, iletinin şifresi çözülmelidir. Alıcının ortak anahtarı kullanılarak şifrelendiği için, bir anahtar halkasında bulunan alıcının özel anahtarı kullanılarak şifresi çözülmelidir.

z/OS *z/OS üzerinde AMS ile SAF tuş halkalarının kullanılması*

Advanced Message Security (AMS), imzalama ve şifreleme için gerekli sertifikaları tanımlamak ve yönetmek için z/OS SAF anahtarlık hizmetlerini kullanır. İşlevsel olarak RACF ile eşdeğer olan güvenlik ürünleri, aynı destek düzeyini sağlamaları durumunda RACF yerine kullanılabilir.

Anahtar halkalarının verimli kullanımı, sertifikaları yönetmek için gereken yönetimi azaltabilir.

Bir sertifika oluşturulduktan (ya da içe aktarıldıktan) sonra, bu sertifikanın erişilebilir olması için bir anahtar halkasına bağlanması gerekir. Aynı sertifika birden çok anahtar halkasına bağlanabilir.

Advanced Message Security , iki anahtar halkası kümesi kullanır. Bir küme, iletileri oluşturan ya da alan tek tek kullanıcı kimliklerine ait anahtar halkalarından oluşur. Her anahtar halkası, sahip olan kullanıcı kimliğinin sertifikasıyla ilişkili özel anahtarı içerir. Her sertifikanın özel anahtarı, bütünlük korumalı ya da gizlilik korumalı kuyruklar için iletileri imzalamak üzere kullanılır. Ayrıca, ileti alırken gizlilik korumalı ya da gizlilik korumalı kuyruklardan gelen iletilerin şifresini çözmek için de kullanılır.

Diğer küme, AMS adres alanı kullanıcısının sahip olduğu tek bir anahtarlık. İleti oluşturanın ve alıcıların sertifikalarını doğrulamak için gereken imza CA sertifikaları zincirini içerir.

Gizlilik ya da gizlilik koruması kullanıldığında, AMS adres alanı kullanıcısının sahip olduğu anahtarlık, ileti alıcılarının sertifikalarını da içerir. Bu sertifikaların ortak anahtarları, ileti korunan kuyruğa yerleştirildiğinde ileti verilerini şifrelemek için kullanılan simetrik anahtarı şifrelemek için kullanılır. Bu iletiler alındığında, ilgili alıcıların özel anahtarı, daha sonra ileti verilerinin şifresini çözmek için kullanılan simetrik anahtarın şifresini çözmek için kullanılır.

Advanced Message Security , sertifikaları ve özel anahtarları ararken **drq.ams.keyring** anahtarlık adını kullanır. Bu, hem kullanıcı hem de AMS adres alanı anahtarı halkaları için geçerli bir durumdur.

Sertifikalara ve anahtarlık ile bunların veri korumasındaki rollerine ilişkin bir şekil ve ek açıklamalar için Sertifikayla ilgili işlemlerin özetibaşlıklı konuya bakın.

İmzalama için kullanılan özel anahtarın herhangi bir etiketi olabilir, ancak varsayılan sertifika olarak bağlanması gerekir. APAR PH44820' den önce, şifre çözme için kullanılan özel anahtarın herhangi bir etiketi olabilir, ancak varsayılan sertifika olarak bağlanması gerekir. APAR PH44820 uygulandığında, şifre çözme için kullanılan özel anahtarın ya da anahtarların herhangi bir etiketi olabilir ve anahtar halkasına bağlanması gerekir, ancak artık varsayılan sertifika olarak bağlanması gerekmez.

Sayısal sertifikalar ve anahtar halkaları, öncelikle RACDCERT komutu kullanılarak RACF içinde yönetilir.

Sertifikalar, etiketler ve RACDCERT komutuyla ilgili daha fazla bilgi için *z/OS: Security Server RACF Command Language Reference* adlı yayına ve *z/OS: Security Server RACF Security Administrator's Guide*adlı yayına bakın.

z/OS *Sertifikaların değiştirilmesi*

Bir sertifika yenilendiğinde ya da değiştirildiğinde (örneğin, var olan sertifika süre bitim tarihine yaklaşırken), gizlilik ya da gizlilik ilkeleri tarafından korunan kuyruklardaki mevcut iletilerden korumayı kaldırmak her zaman mümkün değildir.

Bu durum, sertifika aşağıdaki durumlarda ortaya çıkabilir:

- Aynı özel anahtarla yenilendi ve yeniden verilen sertifika özgün sertifikayı değiştirdi
- Yeni bir özel anahtarla yeniden anahtarlandı ve RACDCERT ROLLOVER komutu özgün özel anahtarı sildi

APAR PH44820' den önce, yeni sertifika varsayılan sertifika olarak kullanıcının anahtar halkasına bağlandığında, eski sertifika kullanılarak şifrelenmiş iletilerin şifresini çözmek artık mümkün değildir. APAR PH44820 uygulandığında, gerekli sertifikanın kullanıcının anahtarına bağlı olması koşuluyla iletilerin şifresi çözülür; artık varsayılan olarak bağlanması gerekmez. Bu, yeni sertifika bağlandığında kuyrukta bulunan iletilerin şifresinin başarıyla çözülmesini sağlar.

Aşağıdaki örnekte, APAR PH44820 uygulandığında var olan sertifikaya dayalı olarak yeni bir sertifikanın nasıl oluşturulabileceği gösterilmektedir:

- Yeni genel/özel anahtar çiftiyle var olan sertifikaya dayalı olarak yeni bir sertifika yaratılır.
- Yeni sertifika veren yetkili tarafından imzalandı.
- Eski sertifikanın genel anahtarı AMS adres alanının anahtarlığından kaldırılır ve yeni sertifikanın genel anahtarı eklenir.
- Yeni sertifika ve özel anahtar, eski sertifikaya ek olarak kullanıcının anahtar halkasına eklenir.

```
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 )
```

Sertifikalar, etiketler ve RACDCERT komutuyla ilgili daha fazla bilgi için *z/OS: Security Server RACF Command Language Reference* adlı yayına ve *z/OS: Security Server RACF Security Administrator's Guide*adlı yayına bakın.

z/OS *z/OS üzerinde AMS için RACDCERT komutuna erişim yetkisi verilmesi*

RACDCERT komutunu kullanma yetkisi, z/OS sistem programcınız tarafından tamamlanması gereken bir kuruluş sonrası görevdir. Bu görev, Advanced Message Security güvenlik yöneticisine ilgili izinlerin verilmesini içerir.

Özet olarak, RACF RACDCERT komutuna erişime izin vermek için aşağıdaki komutlar gereklidir:

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID( admin ) ACCESS(CONTROL)
SETROPTS RACLIST(FACILITY) REFRESH
```

Bu örnekte *admin* , güvenlik denetimcinizin kullanıcı kimliğini ya da RACDCERT komutunu kullanmak istediğiniz herhangi bir kullanıcıyı belirtir.

z/OS *z/OS üzerinde AMS kullanıcıları için sertifika ve anahtar halkaları oluşturma*

Bu bölümde, bir RACF Certificate Authority (CA) kullanılarak Advanced Message Security (AMS) z/OS kullanıcıları için gerekli olan sertifikaları ve anahtar halkalarını oluşturmak için gereken adımlar açıklanmaktadır.

z/OS üzerinde Advanced Message Security kullanırken sertifikalarla ilgili sorunların çözülmesi

Anahtar depolarındaki sertifikalarla ve eksik girdilerle ilgili sorunlarınız varsa, GSKIT izlemesini etkinleştirebilirsiniz.

AMS başlatılan görev yordamında ENVARS DD tarafından başvurulan dosyaya şunu ekleyin:

```
GSK_TRACE_FILE=/u/... /gsktrace
GSK_TRACE=0xff
```

Ek bilgi için Ortam değişkenleri başlıklı konuya bakın.

Anahtar deposuna her erişim için, veriler GSK_TRACE_FILEiçinde belirtilen izleme dosyasına yazılır.

İzleme dosyasını biçimlendirmek için şu komutu kullanın:

```
gsktrace inputtrace file > output_file
```

Senaryo

Gerekli adımları açıklamak için gönderen uygulamanın ve alan uygulamanın senaryosu kullanılır.

Aşağıdaki örneklerde, user1 bir iletinin yaratıcısıdır ve user2 alıcısıdır. Advanced Message Security adres alanının kullanıcı kimliği şudur: WMQAMSD.

Burada gösterilen örneklerdeki tüm komutlar, ISPF seçenek 6 'dan adminyönetimle görevli kullanıcı kimliği tarafından verilir.

z/OS *z/OS üzerinde AMS için yerel Sertifika Yetkilisi sertifikası tanımlama*

Sertifika yetkiliniz olarak RACF kullanıyorsanız, daha önce yapmadıysanız bir sertifika yetkilisi sertifikası oluşturmanız gerekir. Burada gösterilen komut bir sertifika yetkilisi (ya da imzalayıcı) sertifikası yaratır. Bu örnek, Advanced Message Security kullanıcılarının ve uygulamalarının kimliğini yansıtan sonraki sertifikalar oluşturulurken kullanılacak AMSCA adlı bir sertifika yaratır.

Bu komut, özellikle SUBJECTSDN, kuruluşunuzda kullanılan adlandırma yapısını ve kuralları yansıtacak şekilde değiştirilebilir:

```
RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('AMSCA') O('ibm') C('us'))
KEYUSAGE(CERTSIGN) WITHLABEL('AMSCA')
```

Not: Bu yerel sertifika yetkilisi sertifikasıyla imzalanan sertifikalar, RACDCERT LIST komutuyla listelendiğinde CN=AMSCA, O=ibm, C=us sertifika vereni gösterir.

z/OS *z/OS üzerinde AMS için özel anahtarla dijital sertifika oluşturma*

Her Advanced Message Security kullanıcısı için özel anahtarlı bir sayısal sertifika oluşturulmalıdır. Burada gösterilen örnekte, RACDCERT komutları, AMSCA etiketiyle tanıtılan yerel CA sertifikasıyla imzalanmış user1 ve user2için sertifika oluşturmak üzere kullanılır.

```
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
```

Sertifikaya TRUST özniteliğini eklemek için RACDCERT ALTER komutu gereklidir. Bu yordam kullanılarak bir sertifika ilk yaratıldığında, sertifika imzalama sertifikasından farklı bir geçerli tarih aralığına sahiptir. Sonuç olarak, RACF bunu NOTRUST olarak işaretler; bu da sertifikanın kullanılmayacağı anlamına gelir. TRUST özniteliğini ayarlamak için RACDCERT ALTER komutunu kullanın.

KEYUSAGE öznitelikleri HANDSHAKE, DATAENCRYPT ve DOCSIGN, Advanced Message Securitytarafından kullanılan sertifikalar için belirtilmelidir.

Çizelge 109. RACDCERT KEYUSAGE değerleri ve göstergeleri

KEYUSAGE Değeri	**Göstergeler Kümesi**
EL SıKışMA	digitalSignature ve keyEncipherment
DATAENCRYPT	dataEncipherment
İşARETLEYIN	nonRepudiation
CERTSIGN	keyCertİmzala ve cRLSign

z/OS *z/OS üzerinde AMS için RACF anahtar halkalarının oluşturulması*

Burada gösterilen komutlar, RACF-tanımlı kullanıcı kimlikleri user1, user2ve Advanced Message Security adres alanı görevi kullanıcısı WMQAMSD için bir anahtarlık oluşturur. Anahtarlık adı Advanced Message Security ile düzeltilmiştir ve gösterildiği gibi, tırnak işareti olmadan kodlanmalıdır. Ad, büyük ve küçük harfe duyarlıdır.

```
RACDCERT ID(user1) ADDRING(drq.ams.keyring)
RACDCERT ID(user2) ADDRING(drq.ams.keyring)
RACDCERT ID(WMQAMSD) ADDRING(drq.ams.keyring)
```

z/OS *Sertifikaların z/OS üzerinde AMS için anahtar halkalarına bağlanması*

Kullanıcı ve CA sertifikalarını anahtar halkalarına bağlayın:

```
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))
```

APAR PH44820' den önce, şifre çözme için kullanılan özel anahtarı içeren sertifika, varsayılan sertifika olarak kullanıcının anahtar halkasına bağlanmalıdır. APAR PH44820 uygulandığında, şifre çözme için kullanılan özel anahtarı ya da anahtarları içeren sertifikaların kullanıcının anahtarlığına bağlanması gerekir, ancak artık varsayılan sertifika olarak bağlanmaları gerekmez.

RACDCERT USAGE (SITE) özniteliği, özel anahtarın anahtar halkasında erişilebilir olmasını önlerken, RACDCERT USAGE (PERSONAL) özniteliği özel anahtarın (varsa) kullanılmasına izin verir. User2' nin sertifikası, iletileri kuyruğa yerleştirilirken şifrelemek için ortak anahtarı gerektiğinden Advanced Message Security adres alanı anahtarlığı halkasına bağlanmalıdır. KULLANIM (SITE), user2' nin özel anahtarının güvenlik açığını sınırlar.

AMSCA etiketli CERTAUTH sertifikası, iletiyi oluşturan user1sertifikasını imzalamak için kullanıldığından Advanced Message Security adres alanı anahtarlığı halkasına bağlanmalıdır. user1' in imzalama sertifikasını doğrulamak için kullanılır.

z/OS *z/OS üzerinde AMS için anahtarlık doğrulaması*

Tüm komutlar girildikten sonra anahtarlık burada gösterildiği gibi görünmelidir:

```
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
```

Tek tek sertifikaların listelenmesi, halka ilişkisini de gösterir.

```
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<:
```

Başarımı artırmak için, AMS adres alanıyla ilişkili drq.ams.keyring içeriği, adres alanının ömrü boyunca önbelleğe alınır. Bu anahtarlık üzerindeki değişiklikler otomatik olarak etkili olmaz. Yönetici aşağıdaki işlemleri gerçekleştirerek önbelleği yenileyebilir:

- Kuyruk yöneticisi durduruluyor ve yeniden başlatılıyor.
- z/OS MODIFY komutunu kullanarak:

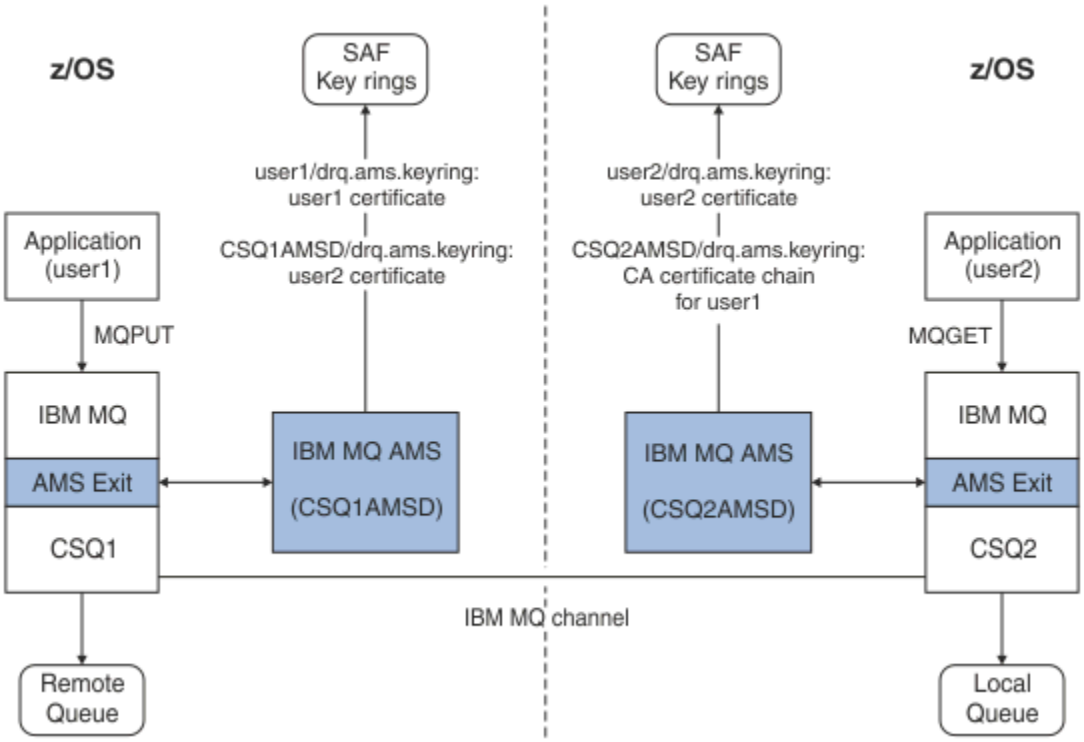
```
F qmgrAMSM,REFRESH KEYRING
```

İlgili görevler

ÇalışırkenAdvanced Message Security

z/OS *z/OS üzerinde AMS için sertifikayla ilgili işlemlerin özeti*

Şekil 35 sayfa 677 içinde, uygulama gönderme ve alma ile ilgili sertifikalar arasındaki ilişkiler gösterilmektedir. Gösterilen senaryo, iki z/OS kuyruk yöneticisi arasında bir veri koruma ilkesi kullanılarak uzaktan kuyruğa alınmayı içerir. Şekil 35 sayfa 677içinde "AMS", " Advanced Message Security".



Şekil 35. Uygulama ve sertifika ilişkileri

Bu şemada, 'user1' olarak çalışan bir uygulama, 'user2' olarak çalışan bir uygulama tarafından CSQ2kuyruk yöneticisi tarafından yönetilen yerel bir kuyruktan alınması amaçlanan CSQ1kuyruk yöneticisi tarafından yönetilen uzak bir kuyruğa bir ileti koyar. Çizge bir Advanced Message Security gizlilik ilkesini varsayar; bu, iletinin hem imzalı hem de şifreli olduğu anlamına gelir.

Advanced Message Security , bir koyma işlemi gerçekleştiğinde iletiyi yakalar ve ileti verilerini şifrelemek için kullanılan simetrik anahtarı şifrelemek için user2' nin sertifikasını (AMS adres alanı kullanıcısının anahtar halkasında saklanır) kullanır.

user2' nin sertifikasının, USAGE (SITE) seçeneğiyle AMS adres alanı kullanıcı anahtarlığı halkasına bağlı olduğunu unutmayın. Bu, AMS adres alanı kullanıcısının sertifikaya ve genel anahtara erişebildiği, ancak özel anahtara erişemediği anlamına gelir.

Alıcı ucunda Advanced Message Security , user2tarafından verilen alma işlemini durdurur ve ileti verilerinin şifresini çözmesi için simetrik anahtarın şifresini çözmek üzere user2' nin sertifikasını kullanır. Daha sonra, AMS adres alanı kullanıcısının anahtar halkasında saklanan user1sertifikasının CA sertifika zincirini kullanarak user1' in imzasını doğrular.

Bu senaryoda, ancak bir veri koruma bütünlüğü ilkesiyle birlikte, user2 için sertifikalar gerekli değildir.

İleti koruma ilkesi gizlilik ya da bütünlük olan IBM MQkorumalı kuyruklardaki iletileri kuyruğa almak üzere Advanced Message Security 'i kullanmak için Advanced Message Security ' in aşağıdaki veri öğelerine erişimi olmalıdır:

- İletiyi kuyruğa alan kullanıcının X.509 V2 ya da V3 sertifikası ve özel anahtarı.
- Tüm ileti imzalayıcıların dijital sertifikalarını imzalamak için kullanılan sertifika zinciri.
- Veri koruma ilkesi gizlilikse, istenen alıcıların X.509 V2 ya da V3 sertifikası. Amaçlanan alıcılar, kuyrukla ilişkili Advanced Message Security ilkesinde listelenir.

z/OSüzerinde çalışan süreçler ve uygulamalar için Advanced Message Security ' in iki yerde sertifikaları olmalıdır:

- Gönderen uygulamanın (korunan iletiyi kuyruğa alan uygulama) RACF kimliğiyle ya da (gizlilik kullanılıyorsa) alıcı uygulamayla ilişkili SAF yönetimli bir anahtar halkasında.

Advanced Message Security ' in bulduğu sertifika varsayılan sertifikadır ve özel anahtarı içermelidir. Advanced Message Security , gönderen uygulamanın z/OS kullanıcı kimliğini varsayar. Yani, kullanıcının özel anahtarına erişebilmesi için vekil görevi görür.

- AMS adres alanı kullanıcısıyla ilişkili SAF tarafından yönetilen bir anahtar halkasında.

  Bu anahtarlık, gizlilikle korunan iletiler gönderirken, ileti alıcılarının genel anahtar sertifikalarını içerir. İleti alınırken, ileti gönderenin imzasını doğrulamak için gereken Sertifika Yetkilisi sertifikaları zincirini içerir.

Önceki örneklerde yerel CA olarak RACF kullanılmıştır. Ancak, kuruluşunuzda başka bir PKI sağlayıcısı (Sertifika Yetkilisi) kullanabilirsiniz. Başka bir PKI ürünü kullanmak istiyorsanız, özel anahtarın ve sertifikanın, Advanced Message Securitytarafından korunan IBM MQ iletilerini oluşturan z/OS RACF kullanıcı kimlikleriyle ilişkili bir anahtarlığa aktarılması gerektiğini unutmayın.

RACF RACDCERT komutunu, dışa aktarılabilen ve verilmek üzere seçtiğiniz PKI sağlayıcısına gönderilebilen sertifika istekleri oluşturmak için düzenek olarak kullanabilirsiniz.

Aşağıda sertifikayla ilgili adımların bir özeti yer almaktadır:

1. RACF öğesinin yerel CA olduğu bir CA sertifikası oluşturulmasını isteyin. Başka bir PKI sağlayıcısı kullanıyorsanız bu adımı atlayın.
2. CA tarafından imzalanmış kullanıcı sertifikaları oluşturun.
3. Kullanıcılar ve Advanced Message Security AMS adres alanı tanıtıcısı için anahtar halkaları oluşturun.
4. Kullanıcı sertifikasını varsayılan öznitelikle kullanıcı anahtarı halkasına bağlayın.
5. Kullanım (site) özniteliğini kullanarak alıcı sertifikalarını Advanced Message Security AMS adres alanı kullanıcı anahtarlığına bağlayın (Bu adım yalnızca, sonuçta gizlilik korumalı iletilerin alıcıları olacak kullanıcı sertifikaları için gereklidir).
6. İleti gönderenler için CA sertifika zincirlerini Advanced Message Security AMS adres alanı kullanıcı anahtarı halkasına bağlayın. (Bu adım yalnızca gönderen imzalarını doğrulayacak AMS görevleri için gereklidir.)

z/OS *z/OS yerleşik olmayan bir PKI ' nin AMS için yapılandırılması*

Advanced Message Security z/OSiçin, IBM MQ kuyruklarına yerleştirilen ya da bu kuyruklardan alınan iletilerin korunmasında X.509 V3 sayısal sertifikalarını kullanır. Advanced Message Security kendisi bu sertifikaların yaşam çevrimini oluşturmaz ya da yönetmez; bu işlev bir genel anahtar altyapısı (PKI) tarafından sağlanır. Bu yayında sertifikaların kullanımını gösteren örnekler, sertifika isteklerini doldurmak için z/OS Güvenlik Sunucusu RACF ' nu kullanır.

z/OS ya daz/OS olmayan yerleşik PKI kullanılıp kullanılmayacağı, z/OS için AMS yalnızca RACF ya da eşdeğeri tarafından yönetilen anahtar halkalarını kullanır. 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.

Bütünlük ya da şifreleme ilkesiyle korunan z/OSkaynaklı iletiler için, kaynak kullanıcı kimliğinin sertifikasının ve özel anahtarının, ileti oluşturucunun z/OS kullanıcı kimliğiyle ilişkili SAF tarafından yönetilen bir anahtar halkasında saklanması gerekir.

RACF , sertifikaları ve özel anahtarları RACFtarafından yönetilen anahtar halkalarına alma yeteneğini içerir. Sertifikaların RACF tarafından yönetilen anahtar halkalarına nasıl yükleneceğine ilişkin ayrıntılar ve örnekler için z/OS Security Server RACF yayınlarına bakın.

Kuruluşunuz desteklenen PKI ürünlerinden birini kullanıyorsa, ürünün nasıl konuşlandırılacağına ilişkin bilgi için ürünle birlikte gönderilen yayınlara bakın.

Advanced Message Security güvenlik ilkelerinin denetlenmesi

Advanced Message Security , kuyruklardan akan iletileri şifrelemek ve doğrulamak için şifreleme ve imza algoritmalarını belirlemek üzere güvenlik ilkelerini kullanır.

AMS için güvenlik ilkelerine genel bakış

Advanced Message Security güvenlik ilkeleri, bir iletinin şifreli olarak şifrelenip imzalanma şeklini açıklayan kavramsal nesnelerdir.

Güvenlik ilkesi özniteliklerine ilişkin ayrıntılar için aşağıdaki alt konulara bakın:

İlgili kavramlar

"AMS içinde koruma kalitesi" sayfa 683
Advanced Message Security veri koruma ilkeleri, bir koruma kalitesi (QOP) anlamına gelir.

"AMS içindeki güvenlik ilkesi öznitelikleri" sayfa 682
Verileri korumak üzere belirli bir algoritma ya da yöntem seçmek için Advanced Message Security kullanabilirsiniz.

AMS içindeki ilke adları

İlke adı, belirli bir Advanced Message Security ilkesini ve uygulandığı kuyruğu tanımlayan benzersiz bir addır.

İlke adı, geçerli olduğu kuyruk adıyla aynı olmalıdır. Advanced Message Security (AMS) arasında bire bir eşleme vardır Kural ve sıra.

Kuyrukla aynı adı taşıyan bir ilke yaratarak, o kuyruk için ilkeyi etkinleştirmiş olur. İlke adları eşleşmeyen kuyruklar AMSile korunmaz.

İlkenin kapsamı, yerel kuyruk yöneticisi ve kuyruklarıyla ilgilidir. Uzak kuyruk yöneticilerinin, yönettikleri kuyruklar için kendi yerel tanımlı ilkeleri olmalıdır.

AMS içinde imza algoritması

İmza algoritması, veri iletilerini imzalarken kullanılması gereken algoritmayı gösterir.

Geçerli değerler şunlardır:

- MD5
- SHA-1
- SHA-2 Yazı Tipi Ailesi:
  - SHA256
  - SHA384 (kabul edilebilir anahtar uzunluğu alt sınırı-768 bit)
  - SHA512 (kabul edilebilir anahtar uzunluğu alt sınırı-768 bit)

Bir imza algoritması belirtmeyen ya da bir NONEalgoritması belirten bir ilke, ilkeyle ilişkili kuyruğa yerleştirilen iletilerin imzalanmadığını belirtir.

Not: İleti koyma ve alma işlevleri için kullanılan koruma kalitesi eşleşmelidir. Kuyruk ile kuyruktaki ileti arasında bir ilke koruma kalitesi uyuşmazlığı varsa, ileti kabul edilmez ve hata işleme kuyruğuna gönderilir. Bu kural hem yerel, hem de uzak kuyruklar için geçerlidir.

AMS içinde şifreleme algoritması

Şifreleme algoritması, ilkeyle ilişkili kuyruğa yerleştirilen veri iletileri şifrelenirken kullanılması gereken algoritmayı gösterir.

Geçerli değerler şunlardır:

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128
- AES256

Şifreleme algoritması belirtmeyen ya da NONE algoritmasını belirten bir ilke, ilkeyle ilişkili kuyruğa yerleştirilen iletilerin şifrelenmediğini belirtir.

NONE dışında bir şifreleme algoritması belirten bir ilkenin, Advanced Message Security şifrelenmiş iletileri de imzalandığı için en az bir Alıcı DN 'si ve bir imza algoritması belirtmesi gerektiğini unutmayın.

Önemli: İleti koyma ve alma işlevleri için kullanılan koruma kalitesi eşleşmelidir. Kuyruk ile kuyruktaki ileti arasında bir ilke koruma kalitesi uyuşmazlığı varsa, ileti kabul edilmez ve hata işleme kuyruğuna gönderilir. Bu kural hem yerel, hem de uzak kuyruklar için geçerlidir.

AMS içinde tolerans

Tolerans özniteliği, Advanced Message Security ' in güvenlik ilkesi belirtilmemiş iletileri kabul edip edemeyeceğini belirtir.

İleti, iletileri şifrelemek için ilke içeren bir kuyruktan alınırken, ileti şifrelenmemişse, çağıran uygulamaya döndürülür. Geçerli değerler şunlardır:

0
Hayır (**varsayılan**).

1
Evet.

Tolerans değeri belirtmeyen ya da 0 değerini belirten bir ilke, ilkeyle ilişkili kuyruğa yerleştirilen iletilerin ilke kurallarıyla eşleşmesi gerektiğini belirtir.

Tolerans isteğe bağlıdır ve ilkelerin kuyruklara uygulandığı, ancak bu kuyrukların zaten belirtilmiş bir güvenlik ilkesi olmayan iletiler içerdiği yapılandırmanın silinmesini kolaylaştırmak için vardır.

AMS içinde gönderen ayırt edici adları

Gönderen ayırt edici adları (DN), bir kuyruğa ileti yerleştirme yetkisi olan kullanıcıları tanımlar. Bir gönderen, iletiyi kuyruğa yerleştirmeden önce, iletiyi imzalamak için sertifikasını kullanır.

Advanced Message Security (AMS) İleti alınıncaya kadar, geçerli bir kullanıcı tarafından veri korumalı bir kuyruğa ileti konup konmadığını denetlemez. Şu anda, ilke bir ya da daha fazla geçerli gönderici öngörüyorsa ve iletiyi kuyruğa yerleştiren kullanıcı geçerli gönderenler listesinde değilse, AMS alan uygulamaya bir hata döndürür ve iletiyi AMS hata kuyruğuna yerleştirir.

Bir ilkenin 0 ya da daha fazla gönderen DN 'si belirtilebilir. İlke için gönderici DN ' leri belirtilmezse, gönderenin sertifikasına güvenildiğini belirten veri korumalı iletileri kuyruğa gönderebilir. Gönderenin sertifikası, genel sertifikayı alan uygulamanın kullanabileceği bir anahtar deposuna eklenerek güvenilir.

Gönderen ayırt edici adları aşağıdaki biçimde bulunur:

```
CN=Common Name,O=Organization,C=Country
```

Önemli:

- Tüm DN Bileşeni adları büyük harfli olmalıdır. DN ' deki tüm bileşen adı tanıtıcıları, aşağıdaki çizelgede gösterilen sırayla belirtilmelidir:

Bileşen adı	Değer
CN	Bir aygıtın tam adı ya da amacı gibi, bu DN ' nin nesnesine ilişkin ortak ad.
Kuruluş Birimi	Ayırt edici ad (DN) nesnesinin bağlı olduğu kuruluş içindeki birim; örneğin, bir kurumsal bölüm ya da bir ürün adı.
O	DN nesnesinin bağlı olduğu kuruluş; örneğin, bir kuruluş.

Bileşen adı	Değer
L	DN nesnesinin bulunduğu yer (şehir ya da belediye).
ST	DN nesnesinin bulunduğu eyaletin ya da bölgenin adı.
C	Ayırt edici ad (DN) nesnesinin bulunduğu ülke.

- İlke için bir ya da daha çok gönderen DN 'si belirtilirse, yalnızca bu kullanıcılar ilkeyle ilişkili kuyruğa ileti yerleştirebilirler.
- Gönderen DN 'leri belirtildiğinde, iletiyi gönderen kullanıcıyla ilişkili sayısal sertifikada bulunan DN' lerle tam olarak eşleşmelidir.
- AMS , yalnızca Latin-1 karakter takımındaki değerleri içeren DN ' leri destekler. Kümenin karakterlerine sahip DN ' ler oluşturmak için öncelikle UTF-8 kodlaması açık ya da **strmqikm** GUI ile AIX and Linux kodlaması kullanılarak UTF-8 kodlamasında oluşturulan bir DN ile bir sertifika oluşturmanız gerekir. Daha sonra, UTF-8 kodlaması açık bir Linux ya da AIX platformundan bir ilke oluşturmanız ya da IBM MQiçin AMS eklentisini kullanmanız gerekir.
- AMStarafından, gönderenin adını x.509 biçiminden DN biçimine dönüştürmek için kullanılan yöntem, İl ya da İl değeri için her zaman ST = kullanır.
- Aşağıdaki özel karakterler için çıkış karakterleri gerekir:

```
, (comma)
+ (plus)
" (double quote)
\ (backslash)
< (less than)
> (greater than)
; (semicolon)
```

- Ayırt edici ad gömülü boşluklar içeriyorsa, DN ' yi çift tırnak içine almanız gerekir.

İlgili kavramlar

“AMS içinde alıcı ayırt edici adları” sayfa 681
Alıcı ayırt edici adları (DN), kuyruktan ileti alma yetkisi olan kullanıcıları tanımlar.

AMS içinde alıcı ayırt edici adları

Alıcı ayırt edici adları (DN), kuyruktan ileti alma yetkisi olan kullanıcıları tanımlar.

Bir ilkenin sıfır ya da daha fazla alıcı DN 'si belirtilebilir. Alıcı ayırt edici adları aşağıdaki biçimdedir:

```
CN=Common Name,O=Organization,C=Country
```

Önemli:

- Tüm DN Bileşeni adları büyük harfli olmalıdır. DN ' deki tüm bileşen adı tanıtıcıları, aşağıdaki çizelgede gösterilen sırayla belirtilmelidir:

Bileşen adı	Değer
CN	Bir aygıtın tam adı ya da amacı gibi, bu DN ' nin nesnesine ilişkin ortak ad.
Kuruluş Birimi	Ayırt edici ad (DN) nesnesinin bağlı olduğu kuruluş içindeki birim; örneğin, bir kurumsal bölüm ya da bir ürün adı.
O	DN nesnesinin bağlı olduğu kuruluş; örneğin, bir kuruluş.

Bileşen adı	Değer
L	DN nesnesinin bulunduğu yer (şehir ya da belediye).
ST	DN nesnesinin bulunduğu eyaletin ya da bölgenin adı.
C	Ayırt edici ad (DN) nesnesinin bulunduğu ülke.

- İlke için alıcı DN ' leri belirtilmezse, herhangi bir kullanıcı ilkeyle ilişkili kuyruktan ileti alabilir.
- İlke için bir ya da daha çok alıcı DN 'si belirtilirse, yalnızca bu kullanıcılar ilkeyle ilişkili kuyruktan ileti alabilir.
- Alıcı DN ' leri belirtildiğinde, iletiyi alan kullanıcıyla ilişkili sayısal sertifikada bulunan DN ile tam olarak eşleşmelidir.
- Advanced Message Security , yalnızca Latin-1 karakter takımındaki değerleri içeren DN ' leri destekler. Kümenin karakterlerine sahip DN ' ler oluşturmak için öncelikle UTF-8 kodlamasında AIX ya da Linux kodlamasında UTF-8 kodlaması açık ya da **strmqikm** GUI ile oluşturulan bir DN ile sertifika oluşturmanız gerekir. Daha sonra, UTF-8 kodlaması açık Linux ya da AIX platformundan bir ilke oluşturmanız ya da IBM MQiçin Advanced Message Security eklentisini kullanmanız gerekir.

İlgili kavramlar

“AMS içinde gönderen ayırt edici adları” sayfa 680
Gönderen ayırt edici adları (DN), bir kuyruğa ileti yerleştirme yetkisi olan kullanıcıları tanımlar. Bir gönderen, iletiyi kuyruğa yerleştirmeden önce, iletiyi imzalamak için sertifikasını kullanır.

AMS içindeki güvenlik ilkesi öznitelikleri

Verileri korumak üzere belirli bir algoritma ya da yöntem seçmek için Advanced Message Security kullanabilirsiniz.

Güvenlik ilkesi, bir iletinin kriptografik olarak şifrelenip imzalanma şeklini açıklayan kavramsal bir nesnedir.

Çizelge 110. AMS içindeki güvenlik ilkesi öznitelikleri

Öznitelikler	Açıklama
İlke adı	Kuyruk yöneticisine ilişkin ilkenin benzersiz adı.
İmza Algoritması	Göndermeden önce iletileri imzalamak için kullanılan şifreleme algoritması.
Şifreleme Algoritması	Göndermeden önce iletileri şifrelemek için kullanılan şifreleme algoritması.
Alıcı listesi	Bir iletinin olası alıcılarının sertifika ayırt edici adlarının (DN) listesi.
İmza DN denetim listesi	İleti alınırken doğrulanacak imza DN ' lerinin listesi.

Advanced Message Securityiçinde, iletiler simetrik bir anahtarla şifrelenir ve simetrik anahtar, alıcıların ortak anahtarlarıyla şifrelenir. Ortak anahtarlar, 2048 bit 'e kadar etkili uzunlukta anahtarlarla RSA algoritmasıyla şifrelenir. Gerçek asimetrik anahtar şifrelemesi, sertifika anahtarı uzunluğuna bağlıdır.

Desteklenen simetrik anahtar algoritmaları şunlardır:

- Deprecated RC2
- Deprecated DES
- Deprecated 3DES
- AES128

- AES256

Advanced Message Security ayrıca aşağıdaki şifreleme hash işlevlerini de destekler:

- Deprecated MD5
- Deprecated SHA-1
- SHA-2 Yazı Tipi Ailesi:
  - SHA256
  - SHA384 (kabul edilebilir anahtar uzunluğu alt sınırı-768 bit)
  - SHA512 (kabul edilebilir anahtar uzunluğu alt sınırı-768 bit)

Not: İleti koyma ve alma işlevleri için kullanılan koruma kalitesi eşleşmelidir. Kuyruk ile kuyruktaki ileti arasında bir ilke koruma kalitesi uyuşmazlığı varsa, ileti kabul edilmez ve hata işleme kuyruğuna gönderilir. Bu kural hem yerel, hem de uzak kuyruklar için geçerlidir.

AMS içinde koruma kalitesi

Advanced Message Security veri koruma ilkeleri, bir koruma kalitesi (QOP) anlamına gelir.

Advanced Message Security içindeki üç koruma düzeyi, IBM MQ 9.0 ve sonraki yayın düzeylerinde dördüncü bir düzeyle tamamlanmıştır ve bunların tümü, iletiyi imzalamak ve şifrelemek için kullanılan şifreleme algoritmalarına bağlıdır:

- Gizlilik-kuyruğa yerleştirilen iletiler imzalanmalı ve şifrelenmelidir.
- Bütünlük-kuyruğa yerleştirilen iletiler gönderen tarafından imzalanmalıdır.
- Gizlilik-kuyruğa yerleştirilen iletiler şifrelenmelidir. Daha fazla bilgi için bkz. “AMS ile sağlanan koruma nitelikleri” sayfa 609
- Yok-veri koruması uygulanamaz.

Bir kuyruğa yerleştirildiğinde iletilerin imzalanması gerektiğini öngören bir ilke, bir QOP ' ye (INTEGRITY) sahiptir. Bir QOP ' nin bütünlüğü, bir politikanın bir imza algoritması öngördüğü, ancak bir şifreleme algoritması gerektirmediği anlamına gelir. Bütünlük korumalı iletilere "İMZALANMış" da denir.

Bir kuyruğa yerleştirildiğinde iletilerin imzalanması ve şifrelenmesi gerektiğini öngören bir ilke, PRIVACY QOP değerine sahiptir. PRIVACY QOP, bir politikanın imza algoritması ve şifreleme algoritması öngördüğü anlamına gelir. Gizlilik korumalı mesajlara "Mühürlü" de denir.

Bir kuyruğa yerleştirildiğinde iletilerin şifrelenmesi gerektiğini öngören bir ilke, bir GIZLILIK QOP 'sine sahiptir. GIZLILIK 'nin QOP' u, bir politikanın bir şifreleme algoritması öngördüğü anlamına gelir.

Bir imza algoritması ya da şifreleme algoritması gerektirmeyen bir ilke, NONE QOP değerine sahiptir. Advanced Message Security , QOP değeri NONE olan bir ilkeye sahip kuyruklar için veri koruması sağlamaz.

AMS içinde güvenlik ilkelerinin yönetilmesi

Güvenlik ilkesi, bir iletinin kriptografik olarak şifrelenip imzalanma şeklini açıklayan kavramsal bir nesnedir.

Güvenlik ilkeleriyle ilgili tüm yönetim görevlerinin çalıştırıldığı konum, kullandığınız platforma bağlıdır.

- ALW AIX, Linux, and Windowsişletim sistemlerinde, güvenlik ilkelerinizi yönetmek için DELETE POLICY, DISPLAY POLICYve SET POLICY (ya da eşdeğer PCF) komutlarını kullanırsınız.
  - Linux AIX AIX and Linuxüzerinde, yönetim görevleri *MQ_INSTALLATION_PATH*`/bin`içinden çalıştırılabilir.
  - Windows Windows platformlarında, PATH ortam değişkeni kuruluşta güncellendiğinden, yönetim görevleri herhangi bir yerden çalıştırılabilir.

- IBM iIBM iişletim sisteminde, IBM MQ kurulduğunda sistemin birincil dili için QSYS sistem kitaplığına DSPMQMSPL, SETMQMSPLve WRKMQMSPL komutları kurulur.

  Ek ulusal dil sürümleri, dil özelliği yüküne göre QSYS29xx kitaplıklarına kurulur. Örneğin, birincil dil olarak ABD İngilizcesi ve ikincil dil olarak Korece olan bir makinede, QSYS 'ye ABD İngilizcesi komutları ve QSYS2962 ' de Korece ikincil dil yükü kurulur.
- z/OSz/OSsistemlerinde, denetim komutları ileti güvenliği ilkesi yardımcı programı (CSQ0UTIL) kullanılarak çalıştırılır. z/OSüzerinde ilkeler yaratıldığında, değiştirildiğinde ya da silindiğinde, kuyruk yöneticisi durdurulup yeniden başlatılıncaya ya da z/OS MODIFY komutu Advanced Message Security ilke yapılandırmasını yenilemek için kullanılıncaya kadar değişiklikler Advanced Message Security tarafından tanınmaz. Örneğin:

  ```
  F <qmgr ssid>AMSM,REFRESH POLICY
  ```

İlgili görevler

“AMS içinde güvenlik ilkeleri oluşturma” sayfa 684
Güvenlik ilkeleri, ileti konduğunda iletinin nasıl korunacağını ya da ileti alındığında iletinin nasıl korunması gerektiğini tanımlar.

“AMS içinde güvenlik ilkelerini değiştirme” sayfa 685
Önceden tanımladığınız güvenlik ilkelerinin ayrıntılarını değiştirmek için Advanced Message Security kullanabilirsiniz.

“AMS içinde güvenlik ilkelerinin görüntülenmesi ve dökümü” sayfa 686
Sağladığınız komut satırı değiştirgelerine bağlı olarak, tüm güvenlik ilkelerinin ya da adlandırılmış bir ilkenin ayrıntılarını görüntülemek için **dspmqspl** komutunu kullanın.

“AMS içinde güvenlik ilkelerini kaldırma” sayfa 687
Advanced Message Securityiçindeki güvenlik ilkelerini kaldırmak için `setmqspl` komutunu kullanmanız gerekir.

ÇalışırkenAdvanced Message Security

İlgili başvurular

İleti güvenlik ilkesi yardımcı programı (CSQ0UTIL)

AMS içinde güvenlik ilkeleri oluşturma

Güvenlik ilkeleri, ileti konduğunda iletinin nasıl korunacağını ya da ileti alındığında iletinin nasıl korunması gerektiğini tanımlar.

Başlamadan önce

Güvenlik ilkeleri yaratılırken karşılanması gereken bazı giriş koşulları vardır:

- Kuyruk yöneticisi çalışıyor olmalıdır.
- Bir güvenlik ilkesinin adı, IBM MQ nesnelerini adlandırma kurallarınauygun olmalıdır.
- Kuyruk yöneticisine bağlanmak ve bir güvenlik ilkesi yaratmak için gereken yetkiye sahip olmanız gerekir:
  - z/OSz/OSüzerinde, İleti güvenliği ilkesi yardımcı programı (CSQ0UTIL)içinde belgelenen yetkilere yetki verin.
  - Multiz/OSdışındaki diğer platformlarda, setmqaut komutunu kullanarak gerekli + connect, + inq ve + chg yetkilerini vermeniz gerekir.

  Güvenliği yapılandırmaya ilişkin daha fazla bilgi için bkz. “Güvenliğin ayarlanması” sayfa 129.
- z/OSz/OSsistemlerinde, gerekli sistem nesnelerinin CSQ4INSMiçindeki tanımlara göre tanımlandığından emin olun.

Örnek

Aşağıda, QMGRkuyruk yöneticisinde ilke yaratma örneği verilmiştir. İlke, iletilerin SHA256 algoritması kullanılarak imzalanacağını ve DN: CN=joe, O=IBM, C=US ve DN: CN=jane, O=IBM, C = US olan sertifikalar için AES256 algoritması kullanılarak şifreleneceğini belirtir. Bu ilke MY.QUEUE' e iliştirilir:

```
setmqspl -m QMGR -p MY.QUEUE -s SHA256 -e AES256 -r CN=joe,O=IBM,C=US -r CN=jane,O=IBM,C=US
```

Aşağıda, QMGRkuyruk yöneticisinde ilke yaratma örneği verilmiştir. İlke, iletilerin DN 'li sertifikalar için 3DES algoritması kullanılarak şifreleneceğini belirtir: CN=can, O=IBM, C=US ve CN=jeff, O=IBM, C=US ve DN' li sertifika için SHA256 algoritmasıyla imzalanmış: 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
```

Not:

- İleti koyma ve alma için kullanılan koruma kalitesi eşleşmelidir. İleti için tanımlanan ilke koruma kalitesi, kuyruk için tanımlanandan daha zayıfsa, ileti hata işleme kuyruğuna gönderilir. Bu ilke hem yerel, hem de uzak kuyruklar için geçerlidir.

İlgili başvurular

setmqspl komutu özniteliklerinin tam listesi

AMS içinde güvenlik ilkelerini değiştirme

Önceden tanımladığınız güvenlik ilkelerinin ayrıntılarını değiştirmek için Advanced Message Security kullanabilirsiniz.

Başlamadan önce

- Üzerinde çalışmak istediğiniz kuyruk yöneticisi çalışıyor olmalıdır.
- Kuyruk yöneticisine bağlanmak ve bir güvenlik ilkesi yaratmak için gereken yetkiye sahip olmanız gerekir.
  - z/OS z/OSüzerinde, İleti güvenliği ilkesi yardımcı programı (CSQ0UTIL)içinde belgelenen yetkilere yetki verin.
  - Multi z/OSdışındaki diğer platformlarda, setmqaut komutunu kullanarak gerekli + connect, + inq ve + chg yetkilerini vermeniz gerekir.

  Güvenliği yapılandırmaya ilişkin daha fazla bilgi için bkz. “Güvenliğin ayarlanması” sayfa 129.

Bu görev hakkında

Güvenlik ilkelerini değiştirmek için, `setmqspl` komutunu yeni öznitelikler sağlayan var olan bir ilkeye uygulayın.

Örnek

Here is an example of creating a policy named MYQUEUE on a queue manager named QMGR, specifying that messages are to be encrypted using the 3DES algorithm for authors (**-a**) having certificates with Distinguished Name (DN) of CN=alice,O=IBM,C=US and signed with the SHA256 algorithm for recipients (**-r**) having certificates with DN of 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
```

Bu ilkeyi değiştirmek için, `setmqspl` komutunu örnekteki tüm özniteliklerle birlikte yürütün ve yalnızca değiştirmek istediğiniz değerleri değiştirin. Bu örnekte, önceden yaratılan ilke yeni bir kuyruğa eklenir ve şifreleme algoritması AES256olarak değiştirilir:

```
setmqspl -m QMGR -p MYQUEUE -e AES256 -s SHA256 -a CN=jeff,O=IBM,C=US -r CN=alice,O=IBM,C=US
```

İlgili başvurular

setmqspl (güvenlik ilkesini ayarla)

AMS içinde güvenlik ilkelerinin görüntülenmesi ve dökümü

Sağladığınız komut satırı değiştirgelerine bağlı olarak, tüm güvenlik ilkelerinin ya da adlandırılmış bir ilkenin ayrıntılarını görüntülemek için **dspmqspl** komutunu kullanın.

Başlamadan önce

- Güvenlik ilkeleri ayrıntılarını görüntülemek için kuyruk yöneticisinin var olması ve çalışıyor olması gerekir.
- Kuyruk yöneticisine bağlanmak ve bir güvenlik ilkesi yaratmak için gereken yetkiye sahip olmanız gerekir.
  - z/OS z/OSüzerinde, İleti güvenliği ilkesi yardımcı programı (CSQ0UTIL)içinde belgelenen yetkilere yetki verin.
  - Multi z/OSdışındaki diğer platformlarda, setmqaut komutunu kullanarak gerekli + connect, + inq ve + chg yetkilerini vermeniz gerekir.

  Güvenliği yapılandırmaya ilişkin daha fazla bilgi için bkz. “Güvenliğin ayarlanması” sayfa 129.

Bu görev hakkında

dspmqspl komut işaretlerinin listesi:

*Çizelge 111. **dspmqspl** komut işaretleri.*

Komut işareti	Açıklama
-m	Kuyruk yöneticisi adı (zorunlu).
-p	İlke adı.
-export	Bu işaretin eklenmesi, farklı bir kuyruk yöneticisine kolayca uygulanabilen çıkış oluşturur.

Örnek

Aşağıdaki örnekte, `venus.queue.manager`için iki güvenlik ilkesinin nasıl yaratılacağı gösterilmektedir:

```
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
```

Bu örnek, `venus.queue.manager` için tanımlanan tüm ilkelerin ayrıntılarını ve ürettiği çıktıyı görüntüleyen bir komutu gösterir:

```
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
```

Bu örnek, `venus.queue.manager` için tanımlanan seçili bir güvenlik ilkesinin ayrıntılarını ve ürettiği çıktıyı görüntüleyen bir komutu gösterir:

```
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
```

Sonraki örnekte, önce bir güvenlik ilkesi oluşturuyoruz, sonra **`-export`** işaretini kullanarak ilkeyi dışa aktarıyoruz:

```
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 z/OSsistemlerinde, dışa aktarılan ilke bilgileri CSQ0UTIL tarafından EXPORT DD ' ye yazılır.

Multi z/OSdışındaki platformlarda, çıkışı bir dosyaya yeniden yönlendirin; örneğin:

```
dspmqspl -m venus.queue.manager -export > policies.[bat|sh]
```

Bir güvenlik ilkesini içe aktarmak için:

- Linux AIX AIX and Linux'ta:
  1. `mqm` IBM MQ yönetim grubuna ait bir kullanıcı olarak oturum açın.
  2. `. policies.sh`sorun.
- Windows Windowsüzerinde `policies.bat`komutunu çalıştırın.
- z/OS z/OS sistemlerinde, dışa aktarılan ilke bilgilerini içeren veri kümesini SYSIN içine belirterek CSQ0UTIL yardımcı programını kullanın.

İlgili başvurular

dspmqspl komutu özniteliklerinin tam listesi

AMS içinde güvenlik ilkelerini kaldırma

Advanced Message Securityiçindeki güvenlik ilkelerini kaldırmak için `setmqspl` komutunu kullanmanız gerekir.

Başlamadan önce

Güvenlik ilkeleri yönetilirken karşılanması gereken bazı giriş koşulları vardır:

- Kuyruk yöneticisi çalışıyor olmalıdır.

- Kuyruk yöneticisine bağlanmak ve bir güvenlik ilkesi yaratmak için gereken yetkiye sahip olmanız gerekir.
  - z/OS z/OSüzerinde, İleti güvenliği ilkesi yardımcı programı (CSQ0UTIL)içinde belgelenen yetkilere yetki verin.
  - Multi z/OSdışındaki diğer platformlarda, setmqaut komutunu kullanarak gerekli + connect, + inq ve + chg yetkilerini vermeniz gerekir.

  Güvenliği yapılandırmaya ilişkin daha fazla bilgi için bkz. “Güvenliğin ayarlanması” sayfa 129.

Bu görev hakkında

setmqspl komutunu **-remove** seçeneğiyle birlikte kullanın.

Örnek

Aşağıda bir ilkeyi kaldırma örneği verilmiştir:

```
setmqspl -m QMGR -remove -p MY.OTHER.QUEUE
```

İlgili başvurular

setmqspl komutu özniteliklerinin tam listesi

AMS içinde sistem kuyruğu koruması

Sistem kuyrukları, IBM MQ ile yan uygulamaları arasında iletişimi sağlar. Bir kuyruk yöneticisi yaratıldığında, IBM MQ iç iletilerini ve verilerini saklamak için bir sistem kuyruğu da yaratılır. Advanced Message Security ile sistem kuyruklarını koruyabilirsiniz; böylece yalnızca yetkili kullanıcılar bunlara erişebilir ya da bunların şifresini çözebilir.

Sistem kuyruk koruması, olağan kuyrukların korunmasıyla aynı kalıbı izler. Bkz. “AMS içinde güvenlik ilkeleri oluşturma” sayfa 684.

Windows Windowssisteminde sistem kuyruğu korumasını kullanmak için `keystore.conf` dosyasını aşağıdaki dizine kopyalayın:

```
c:\Documents and Settings\Default User\.mqs\keystore.conf
```

z/OS z/OSişletim sistemlerinde, `SYSTEM.ADMIN.COMMAND.QUEUE`için koruma sağlamak üzere komut sunucusunun, anahtarlar ve bir yapılandırma içeren `keystore` ve `keystore.conf`ürününe erişimi olmalıdır; böylece, komut sunucusu anahtarlara ve sertifikalara erişebilir. `SYSTEM.ADMIN.COMMAND.QUEUE` güvenlik ilkesinde yapılan tüm değişiklikler, komut sunucusunun yeniden başlatılmasını gerektirir.

Komut kuyruğundan gönderilen ve alınan tüm iletiler, ilke ayarlarına bağlı olarak imzalanır ya da imzalanır ve şifrelenir. Bir yönetici yetkili imzalayıcıları tanımlıyorsa, imzalayıcı Ayırt Edici Ad (DN) denetimini geçmeyen komut iletileri komut sunucusu tarafından yürütülmez ve Advanced Message Security hata işleme kuyruğuna yönlendirilmez. IBM MQ Explorer geçici dinamik kuyruklarına yanıt olarak gönderilen iletiler AMStarafından korunmaz.

Güvenlik ilkelerinin aşağıdaki SYSTEM kuyrukları üzerinde bir etkisi yoktur:

- 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

V 9.3.0 Multi Akış kuyrukları ve AMS

Yinelenen Advanced Message Security (AMS) korumalı iletilerin akışı gerçekleştirilebilir.

Bir kuyrukta, o kuyruğa konan iletilerin imzalanmasına ve/ya da şifrelenmesine neden olan bir AMS ilkesi tanımlanmışsa, kuyruğun **STREAMQ** özniteliğini, her bir korunan iletinin bir kopyasını ikinci bir kuyruğa koyacak şekilde yapılandırabilirsiniz. Yinelenen, akıtılan ileti, özgün kuyruk için yapılandırılan aynı ilke kullanılarak imzalanır ve/veya şifrelenir.

Aşağıdaki örnekte, QUEUE1 ve QUEUE2olmak üzere iki kuyruk yapılandırıyorsunuz. QUEUE1 , akıtılan iletileri QUEUE2: ' ye koyacak şekilde yapılandırılmış **STREAMQ** özniteliğine sahiptir:

```
DEFINE QLOCAL(QUEUE2)

DEFINE QLOCAL(QUEUE1) STREAMQ(QUEUE2)
```

AMS korumalı iletiler, sertifikalı bir kullanıcı tarafından QUEUE1 ' e konuyor `CN=bob,O=IBM,C=GB`.

`CN=alice,O=IBM,C=GB` sertifikalı bir uygulama, QUEUE1' deki iletileri tüketecek.
`CN=fred,O=IBM,C=GB` sertifikalı ayrı bir uygulama, QUEUE2' deki iletileri tüketecek.

QUEUE1 , kendisine uygulanan aşağıdaki AMS gizlilik ilkesine sahiptir:

```
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)
```

QUEUE1ilkesinde bir şifreleme algoritması yapılandırıldıysa, ilkede listelenen alıcılar hem QUEUE1'den gelen özgün iletilerin alıcılarını, hem de QUEUE2' den yinelenen iletileri kullanacak alıcıları içermelidir.

Uygulama QUEUE2 'den gelen iletileri tüketmeye çalıştığında, bütünlük denetimleri gerçekleştirir ve/veya QUEUE2' de ayarlanan ilkeye dayalı olarak iletinin şifresini çözer. Bir uygulama QUEUE2'den akıtılan iletileri kullanmak isterse, QUEUE2 ' de iletilerin bütünlük için denetlenmesine ve şifresinin doğru şekilde çözülmesine izin veren uygun bir ilke ayarlamanız gerekir.

Özellikle, imzalama algoritması, imzalayıcı ve şifreleme algoritması, QUEUE1' e uygulanan ilkeyle aynı olmalıdır. QUEUE2 ilke alıcıları, QUEUE2' deki iletiyi kullanan alıcının kimliğini içermelidir.

Not: QUEUE2 ' ye uygulanan ilkenin QUEUE1ilke kümesinde adı belirtilen tüm alıcıları listemesi gerekli değildir.

Örneğin, `CN=fred,O=IBM,C=GB` sertifika ayırt edici adına sahip bir uygulamanın AMSkorumalı iletilerini okumasına izin vermek için QUEUE2 üzerinde aşağıdaki ilke ayarlanabilir:

```
SET POLICY(QUEUE2) SIGNALG(SHA256) SIGNER('CN=bob,O=IBM,C=GB') ENCALG(AES256)
RECIP('CN=fred,O=IBM,C=GB') ACTION(ADD)
```

İlgili kavramlar

Akış kuyrukları

AMS içinde OAM izinleri verme

Dosya izinleri, tüm kullanıcılara `setmqspl` ve `dspmqspl` komutlarını yürütme yetkisi verir. Ancak Advanced Message Security , Nesne Yetkilisi Yöneticisi 'ne (OAM) dayanır ve bu komutları, IBM MQ yönetim grubu olan mqm grubuna ait olmayan ya da verilen güvenlik ilkesi ayarlarını okuma iznine sahip olmayan bir kullanıcı tarafından yürütülmeye çalışılması bir hatayla sonuçlanır.

Yordam

Bir kullanıcıya gerekli izinleri vermek için şunları çalıştırın:

```
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
```

Not: Bu OAM yetkilerini yalnızca istemcileri Advanced Message Security 7.0.1kullanarak kuyruk yöneticisine bağlamak istiyorsanız ayarlamanız gerekir.

Uyarı: SYSTEM.PROTECTION.POLICY.QUEUE , her durumda zorunlu değildir. IBM MQ , SYSTEM.PROTECTION.POLICY.QUEUE QUEUE.

IBM MQ , kullanılabilir tüm ilkeleri önbelleğe almaz. Çok sayıda ilke varsa, IBM MQ sınırlı sayıda ilkeye önbelleğe alır. Bu nedenle, kuyruk yöneticisinde tanımlanmış az sayıda ilke varsa, SYSTEM.PROTECTION.POLICY.QUEUE.

Ancak, çok sayıda ilke tanımlanmışsa ya da eski istemcileri kullanıyorsanız, bu kuyruk için göz atma yetkisi vermeniz gerekir. SYSTEM.PROTECTION.ERROR.QUEUE , AMS kodu tarafından oluşturulan hata iletilerini koymak için kullanılır. Bu kuyruk için koyma yetkisi, yalnızca kuyruğa bir hata iletisi koyma girişiminde bulunduğunuzda denetlenir. AMS korumalı kuyruğuna ileti koyma ya da alma girişiminde bulunduğunuzda, kuyruk için koyma yetkiniz denetlenmez.

AMS içinde güvenlik izinleri verilmesi

Komut kaynağı güvenliğini kullanırken, Advanced Message Security ' in çalışmasına izin vermek için izinleri ayarlamanız gerekir. Bu konu, örneklerde RACF komutlarını kullanır. İşletmeniz farklı bir dış güvenlik yöneticisi (ESM) kullanıyorsa, bu ESM ' ye ilişkin eşdeğer komutları kullanmanız gerekir.

Güvenlik izinleri verilmesinin üç farklı yönü vardır:

- “AMSM adres alanı” sayfa 691
- “CSQ0UTIL” sayfa 692
- “Advanced Message Security ilkesi tanımlanmış kuyrukların kullanılması” sayfa 692

Notlar: Örnek komutlar aşağıdaki değişkenleri kullanır.

1. *QMgrName* -kuyruk yöneticisinin adı.

   z/OS z/OS' da bu değer, bir kuyruk paylaşım grubunun adı da olabilir.
2. *username* -Bu bir grup adı olabilir.
3. Örnekler MQQUEUE sınıfını gösterir. Bu, MXQUEUE, GMQQUEUE ya da GMXQUEUE de olabilir. Daha fazla bilgi için bkz. “Kuyruk güvenliğine ilişkin profiller” sayfa 197 .

Ayrıca, tanıtım zaten varsa, RDEFINE komutuna gerek yoktur.

AMSM adres alanı

Advanced Message Security adres alanının altında çalıştığı kullanıcı adı için bir IBM MQ güvenliği yayınlamanız gerekir.

- Kuyruk yöneticisine toplu bağlantı için, sorun

```
RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
            PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- SYSTEM.PROTECTION.POLICY.QUEUE, sorun:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
            PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

CSQ0UTIL

Kullanıcıların **setmqspl** ve **dspmqspl** komutlarını çalıştırmasına izin veren yardımcı program, kullanıcı adının iş kullanıcı kimliği olduğu aşağıdaki izinleri gerektirir:

- Kuyruk yöneticisine toplu bağlantı için şu sorunu gerçekleştirin:

```
  RDEFINE MQCONN QMgrName.BATCH UACC(NONE)
             PERMIT QMgrName.BATCH CLASS(MQCONN) ID(username) ACCESS(READ)
```

- SYSTEM.PROTECTION.POLICY.QUEUE, **setmqpol** komutu için gerekli, şu komutu verin:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
              PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(ALTER)
```

- SYSTEM.PROTECTION.POLICY.QUEUE, **dspmqpol** komutu için gerekli, şu komutu verin:

```
  RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
                PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

Advanced Message Security ilkesi tanımlanmış kuyrukların kullanılması

Bir uygulama, üzerinde ilke tanımlanmış olan kuyruklar ile herhangi bir çalışma yaptığında, bu uygulama Advanced Message Security ' in iletileri korumasına izin vermek için ek izinler gerektirir.

Uygulama şunları gerektirir:

- SYSTEM.PROTECTION.POLICY.QUEUE. Şunu yayınlayarak bunu yapın:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)
          PERMIT QMgrName.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

- SYSTEM.PROTECTION.ERROR.QUEUE. Şunu yayınlayarak bunu yapın:

```
RDEFINE MQQUEUE QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)
             PERMIT QMgrName.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE)
ID(username) ACCESS(READ)
```

IBM i IBM i üzerinde AMS için sertifikaların ve anahtar deposu yapılandırma dosyasının ayarlanması

Advanced Message Security korumasını ayarlarken ilk göreviniz bir sertifika oluşturmak ve bunu ortamınızla ilişkilendirmektir. İlişkilendirme, tümleşik dosya sisteminde (IFS) tutulan bir dosya aracılığıyla yapılandırılır.

Yordam

1. IBM iile birlikte gönderilen OpenSSL araçlarını kullanarak kendinden imzalı bir sertifika oluşturmak için QShell 'den aşağıdaki komutu verin:

   ```
   /QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048 -keyout
   $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365
   ```

   Komut, kendinden onaylı yeni bir sertifikaya ilişkin çeşitli ayırt edici ad öznitelikleri için bilgi isteminde bulunur:

   - Ortak Ad (CN =)
   - Kuruluş (O =)
   - Ülke (C =)

Bu, hem PEM (Privacy Enhanced Mail) biçiminde şifrelenmemiş bir özel anahtar hem de eşleşen bir sertifika oluşturur.

Basitlik için, ortak ad, kuruluş ve ülke değerlerini girin. Bu öznitelikler ve değerler, ilke yaratılırken önemlidir.

Ek bilgi istemleri ve öznitelikler, **-config** parametresiyle komut satırında özel bir openssl yapılandırma dosyası belirtilerek özelleştirilebilir. Yapılandırma dosyası sözdizimiyle ilgili daha fazla ayrıntı için OpenSSL belgelerine bakın.

Örneğin, aşağıdaki komut ek X.509 v3 sertifika uzantıları ekler:

```
/QOpenSys/usr/bin/openssl req -x509 -newkey rsa:2048
-keyout $HOME/private.pem -out $HOME/mycert.pem -nodes -days 365 -config myconfig.cnf
```

Burada myconfig.cnf , aşağıdakileri içeren bir ASCII akış dosyasıdır:

```
[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 , hem sertifikanın hem de özel anahtarın aynı dosyada tutulmasını gerektirir. Bunu gerçekleştirmek için aşağıdaki komutu verin:

```
cat $HOME/mycert.pem >> $HOME/private.pem
```

$HOME dosyasındaki `private.pem` dosyası artık eşleşen bir özel anahtar ve sertifika içerirken, `mycert.pem` dosyası iletileri şifreleyebileceğiniz ve imzaları doğrulayabileceğiniz tüm genel sertifikaları içerir.

İki dosyanın, varsayılan konumunuzda bir anahtar deposu yapılandırma dosyası (`keystore.conf`) yaratılarak ortamınızla ilişkilendirilmesi gerekir.

Varsayılan olarak AMS , ana dizininizin `.mqs` alt dizininde anahtar deposu yapılandırmasını arar.

3. QShell 'de `keystore.conf` dosyasını oluşturun:

```
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 *IBM i üzerinde AMS için ilke oluşturma*

Bir ilke yaratmadan önce, korunan iletileri tutmak için bir kuyruk yaratmanız gerekir.

Yordam

1. Bir komut satırı bilgi isteminde şunu girin;

```
CRTMQMQ QNAME(PROTECTED) QTYPE(*LCL) MQMNAME (mqmname)
```

Burada mqmname , kuyruk yöneticinizin adıdır.

Kuyruk yöneticisinin güvenlik ilkelerini kullanabildiğini denetlemek için DSPMQM komutunu kullanın. **Security Policy Capability** içinde *YESdeğerinin gösterildiğinden emin olun.

Tanımlayabileceğiniz en basit ilke, bir sayısal imza algoritması olan ancak şifreleme algoritması olmayan bir ilke yaratarak elde edilen bir bütünlük ilkesidir.

İletiler imzalandı, ancak şifrelenmedi. İletiler şifrelenirse, bir şifreleme algoritması ve bir ya da daha çok hedeflenen ileti alıcısı belirtmeniz gerekir.

Amaçlanan bir ileti alıcısına ilişkin genel anahtar deposundaki bir sertifika, ayırt edici bir adla tanıtılır.

2. QShell 'de aşağıdaki komutu kullanarak, $HOMEiçindeki mycert.pem genel anahtar deposundaki sertifikaların ayırt edici adlarını görüntüleyin:

```
/QOpenSys/usr/bin/openssl x509 -in $HOME/mycert.pem -noout -subject -nameopt RFC2253
```

Ayırt edici adı amaçlanan bir alıcı olarak girmeniz ve ilke adının korunacak kuyruk adıyla eşleşmesi gerekir.

3. CL komut isteminde şunu girin, örneğin:

```
SETMQMSPL POLICY(PROTECTED) MQMNAME (mqmname)SIGNALG(*SHA256) ENCALG(*AES256) RECIP('CN=..,
O=.., C=..')
```

Burada mqmname , kuyruk yöneticinizin adıdır.

İlke oluşturulduktan sonra, bu kuyruk adı aracılığıyla konan, göz atılan ya da yıkıcı bir şekilde kaldırılan iletiler AMS ilkesine tabi olur.

İlgili başvurular

İleti Kuyruğu Yöneticisini Görüntüle (DSPMQM)

MQM Güvenlik İlkesini Ayarla (SETMQMSPL)

IBM i *IBM i üzerinde AMS ilkesinin sınanması*

Güvenlik ilkelerinizi sınamak için ürünle birlikte sağlanan örnek uygulamaları kullanın.

Bu görev hakkında

AMQSPUT4, AMQSGET4, AMQSGBR4gibi IBM MQ ile sağlanan örnek uygulamaları ve PROTECTED kuyruk adını kullanarak iletileri yerleştirmek, göz atmak ve almak için WRKMQMMSG gibi araçları kullanabilirsiniz.

Her şey doğru yapılandırıldıysa, uygulama davranışında, bu kullanıcı için korunmayan bir kuyruktaki davranışında bir fark olmamalıdır.

Advanced Message Securityiçin ayarlanmamış bir kullanıcı ya da iletinin şifresini çözmek için gerekli özel anahtara sahip olmayan bir kullanıcı iletiyi görüntüleyemez. Kullanıcı, MQCC_FAILED (2) ve RC2063 (MQRC_SECURITY_ERROR) neden koduna eşdeğer bir RCFAIL tamamlanma kodu alır.

AMS korumasının etkin olduğunu görmek için, PROTECTED (Korunan) kuyruğuna bazı sınama iletileri koyun; örneğin, AMQSPUT0komutunu kullanın. Daha sonra, atılırken işlenmemiş korunan verilere göz atmak için bir diğer ad kuyruğu yaratabilirsiniz.

Yordam

Bir kullanıcıya gerekli izinleri vermek için şunları çalıştırın:

```
CRTMQMQ QNAME(ALIAS) QTYPE(*ALS) TGTQNAME(PROTECTED) MQMNAME(yourqm)
```

ALIAS kuyruk adı (örneğin, AMQSBCG4 ya da WRKMQMMSG) kullanılarak göz atılması, PROTECTED kuyruğuna göz atıldığında daha büyük `scrambled` iletileri ortaya çıkarmalıdır.

Şifrelenmiş iletiler görünür, ancak AMS ' nin bu adla eşleştirmeyi zorlayabileceği bir ilke olmadığından, ALIAS kuyruğu kullanılarak özgün açık metnin şifresi çözülemez. Bu nedenle, işlenmemiş korunan veriler döndürülür.

İlgili başvurular

MQM Güvenlik İlkesini Ayarla (SETMQMSPL)

MQ İletileriyle Çalışma (WRKMQMMSG)

AMS için komut ve yapılandırma olayları

Advanced Message Securityile, günlüğe kaydedilebilecek ve denetim için ilke değişikliklerinin kaydı olarak hizmet verebilecek komut ve yapılandırma olayı iletileri oluşturabilirsiniz.

IBM MQ tarafından oluşturulan komut ve yapılanış olayları, olayın oluştuğu kuyruk yöneticisindeki adanmış kuyruklara gönderilen PCF biçimindeki iletilerdir.

Yapılandırma olayları iletileri SYSTEM.ADMIN.CONFIG.EVENT kuyruğu.

Komut olayları iletileri SYSTEM.ADMIN.COMMAND.EVENT kuyruğu.

Olaylar, Advanced Message Security güvenlik ilkelerini yönetmek için kullandığınız araçlardan bağımsız olarak oluşturulur.

Advanced Message Securityiçinde, güvenlik ilkelerine ilişkin farklı işlemler tarafından oluşturulan dört olay tipi vardır:

- “AMS içinde güvenlik ilkeleri oluşturma” sayfa 684, iki IBM MQ olay iletisi oluşturur:
  - Bir yapılandırma olayı
  - Bir komut olayı
- “AMS içinde güvenlik ilkelerini değiştirme” sayfa 685, üç IBM MQ olay iletisi oluşturur:
  - Eski güvenlik ilkesi değerlerini içeren bir yapılandırma olayı
  - Yeni güvenlik ilkesi değerlerini içeren bir yapılandırma olayı
  - Bir komut olayı
- “AMS içinde güvenlik ilkelerinin görüntülenmesi ve dökümü” sayfa 686, bir IBM MQ olay iletisi oluşturur:
  - Bir komut olayı
- “AMS içinde güvenlik ilkelerini kaldırma” sayfa 687, iki IBM MQ olay iletisi oluşturur:
  - Bir yapılandırma olayı
  - Bir komut olayı

AMS için olay günlük kaydını etkinleştirme ve devre dışı bırakma

CONFIGEV ve **CMDEV**kuyruk yöneticisi özniteliklerini kullanarak komut ve yapılandırma olaylarını denetleyebilirsiniz. Bu olayları etkinleştirmek için uygun kuyruk yöneticisi özniteliğini ENABLEDolarak ayarlayın. Bu olayları geçersiz kılmak için, uygun kuyruk yöneticisi özniteliğini DISABLEDolarak ayarlayın.

Yordam

Yapılandırma olayları

Yapılandırma olaylarını etkinleştirmek için **CONFIGEV** ayarını ENABLEDolarak ayarlayın. Yapılandırma olaylarını devre dışı bırakmak için **CONFIGEV** ayarını DISABLEDolarak ayarlayın. Örneğin, aşağıdaki MQSC komutunu kullanarak yapılandırma olaylarını etkinleştirebilirsiniz:

```
ALTER QMGR CONFIGEV (ENABLED)
```

Komut olayları

Komut olaylarını etkinleştirmek için **CMDEV** değerini ENABLEDolarak ayarlayın. **DISPLAY MQSC** komutları ve Inquire PCF komutları dışında komutlar için komut olaylarını etkinleştirmek için **CMDEV** değerini NODISPLAYolarak ayarlayın. Komut olaylarını devre dışı bırakmak için **CMDEV** ayarını DISABLEDolarak ayarlayın. Örneğin, aşağıdaki MQSC komutunu kullanarak komut olaylarını etkinleştirebilirsiniz:

```
ALTER QMGR CMDEV (ENABLED)
```

İlgili görevler

IBM MQ içinde yapılandırma, komut ve kaydedici olaylarının denetlenmesi

AMS için komut olayı ileti biçimi

Komut olayı iletisi, bunu izleyen MQCFH yapısı ve PCF değiştirgelerinden oluşur.

Seçilen MQCFH değerleri şunlardır:

```
Type = MQCFT_EVENT;
Command = MQCMD_COMMAND_EVENT;
MsgSeqNumber = 1;
Control = MQCFC_LAST;
ParameterCount = 2;
CompCode = MQCC_WARNING;
Reason = MQRC_COMMAND_PCF;
```

Not: ParameterCount değeri, her zaman MQCFGR tipinde (grup) iki parametre olduğundan ikidir. Her grup uygun parametrelerden oluşur. Olay verileri, CommandContext ve CommandDataolmak üzere iki gruptan oluşur.

CommandContext aşağıdakileri içerir:

EventUserKimliği

Açıklama:	Olayı oluşturan komutu ya da çağrıyı yayınlayan kullanıcı kimliği. (Bu, komutu ya da çağrıyı verme yetkisini denetlemek için kullanılan kullanıcı kimliğiyle aynıdır; bir kuyruktan alınan komutlar için, bu aynı zamanda komut iletisinin MD ' sinden alınan kullanıcı kimliğidir (UserIdentifier)).
Tanıtıcı:	MQCACF_EVENT_USER_ID.
Veri tipi:	MQCFST
Uzunluk üst sınırı:	MQ_USER_ID_LENGTH.
Döndürülen:	Her zaman.

EventOrigin

Açıklama:	Olaya neden olan işlemin kaynağı.
Tanıtıcı:	MQIACF_EVENT_ORIGIN.
Veri tipi:	MQCFIN.
Değerler:	**MQEVO_CONSOLE** Konsol komut satırı. **MQEVO_MSG** IBM MQ Explorer eklentisinden komut iletisi.
Döndürülen:	Her zaman.

EventQMgr

Açıklama:	Komutun ya da çağrım girildiği kuyruk yöneticisi. (Komutun yürütüldüğü ve olayı oluşturan kuyruk yöneticisi olay iletisinin MD ' sinde bulunur).
Tanıtıcı:	MQCACF_EVENT_Q_MGR.
Veri tipi:	MQCFST
Uzunluk üst sınırı:	MQ_Q_MGR_NAME_LENGTH.
Döndürülen:	Her zaman.

EventAccountingSimgesi

Açıklama:	İleti olarak alınan komutlar için (MQEVO_MSG), komut iletisinin MD ' den muhasebe simgesi (AccountingToken).
Tanıtıcı:	MQBACF_EVENT_ACCOUNTING_TOKEN.
Veri tipi:	MQCFBS.
Uzunluk üst sınırı:	MQ_ACCOUNTING_TOKEN_LENGTH.
Döndürülen:	Yalnızca EventOrigin MQEVO_MSG ise.

EventIdentityVerileri

Açıklama:	İleti olarak alınan komutlar (MQEVO_MSG) için, komut iletisinin MD ' den uygulama kimlik verileri (ApplIdentityVerileri).
Tanıtıcı:	MQCACF_EVENT_APPL_IDENTITY.
Veri tipi:	MQCFST
Uzunluk üst sınırı:	MQ_APPL_IDENTITY_DATA_LENGTH.
Döndürülen:	Yalnızca EventOrigin MQEVO_MSG ise.

EventApplTipi

Açıklama:	İleti olarak alınan komutlar için (MQEVO_MSG), komut iletisinin MD ' den alınan uygulama tipi (PutApplTipi).
Tanıtıcı:	MQIACF_EVENT_APPL_TYPE.
Veri tipi:	MQCFIN.
Döndürülen:	Yalnızca EventOrigin MQEVO_MSG ise.

EventApplAdı

Açıklama:	İleti olarak alınan komutlar (MQEVO_MSG) için, komut iletisinin MD ' den uygulamanın adı (PutApplAdı).
Tanıtıcı:	MQCACF_EVENT_APPL_NAME.
Veri tipi:	MQCFST
Uzunluk üst sınırı:	MQ_APPL_NAME_LENGTH.
Döndürülen:	Yalnızca EventOrigin MQEVO_MSG ise.

EventApplKöken

Açıklama:	İleti olarak alınan komutlar (MQEVO_MSG) için, komut iletisinin MD ' den alınan uygulama kaynağı verileri (ApplOriginVerileri).
Tanıtıcı:	MQCACF_EVENT_APPL_ORIGIN.

Veri tipi: MQCFST

Uzunluk üst sınırı: MQ_APPL_ORIGIN_DATA_LENGTH.

Döndürülen: Yalnızca EventOrigin MQEVO_MSG ise.

Komut

Açıklama: Komut kodu.

Tanıtıcı: MQIACF_COMMAND.

Veri tipi: MQCFIN.

Değerler: **MQCMD_INQUIRE_PROT_POLICY sayısal değeri 205**
MQCMD_CREATE_PROT_POLICY sayısal değeri 206
MQCMD_DELETE_PROT_POLICY sayısal değeri 207
MQCMD_CHANGE_PROT_POLICY sayısal değeri 208
Bunlar IBM MQ 8.0 `cmqcfc.h` sürümünde tanımlanmıştır.

Döndürülen: Her zaman.

CommandData , PCF komutunu içeren PCF öğelerini içerir.

AMS için yapılandırma olayı ileti biçimi

Konfigürasyon olayları, standart Advanced Message Security biçimindeki PCF iletileridir.

MQMD ileti tanımlayıcısı için olası değerler Olay iletisi MQMD ' de (ileti tanımlayıcısı)bulunabilir.

Seçilen MQMD değerleri şunlardır:

```
Format = MQFMT_EVENT
Peristence = MQPER_PERSISTENCE_AS_Q_DEF
PutApplType = MQAT_QMGR           //for both CLI and command server
```

İleti arabelleği MQCFH yapısından ve bunu izleyen değiştirge yapısından oluşur. Olası MQCFH değerleri Olay iletisi MQCFH (PCF üstbilgisi)içinde bulunabilir.

Seçilen MQCFH değerleri şunlardır:

```
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}
```

MQCFH ' yi izleyen değiştirgeler şunlardır:

EventUserID

Açıklama: Olayı oluşturan komutu ya da çağrıyı yayınlayan kullanıcı kimliği. (Bu, komutu ya da çağrıyı verme yetkisini denetlemek için kullanılan kullanıcı kimliğiyle aynıdır; bir kuyruktan alınan komutlar için, bu aynı zamanda komut iletisinin MD ' sinden alınan kullanıcı kimliğidir (UserIdentifier)).

Tanıtıcı: **MQCACF_EVENT_USER_ID**

Veri tipi: MQCFST

Uzunluk üst sınırı: MQ_USER_ID_LENGTH.

Döndürülen: Her zaman.

SecurityId

Açıklama: MQMD.AccountingToken ya da yerel komut için Windows SID.

Tanıtıcı: **MQBACF_EVENT_SECURITY_ID**

Veri tipi: MQCBS.

Uzunluk üst sınırı: MQ_SECURITY_ID_LENGTH.

Döndürülen: Her zaman.

EventOrigin

Açıklama: Olaya neden olan işlemin kaynağı.

Tanıtıcı: **MQIACF_EVENT_ORIGIN**

Veri tipi: MQCFIN.

Değerler:

MQEVO_CONSOLE
Konsol komut satırı.

MQEVO_MSG
IBM MQ Explorer eklentisinden komut iletisi.

Döndürülen: Her zaman.

EventQMgr

Açıklama: Komutun ya da çağrım girildiği kuyruk yöneticisi. (Komutun yürütüldüğü ve olayı oluşturan kuyruk yöneticisi olay iletisinin MD ' sinde bulunur).

Tanıtıcı: **MQCACF_EVENT_Q_MGR**

Veri tipi: MQCFST

Uzunluk üst sınırı: MQ_Q_MGR_AD_UZUNLUK

Döndürülen: Her zaman.

ObjectType

Açıklama: Nesne tipi.

Tanıtıcı: **MQIACF_OBJECT_TYPE**

Veri tipi: MQCFIN

Değer:

MQOT_PROT_POLICY
Advanced Message Security koruma ilkesi. **1019** - IBM MQ 8.0 sürümünde ya da `cmqc.h` dosyasında tanımlanan sayısal bir değerdir.

Döndürülen: Her zaman.

PolicyName

Açıklama: Advanced Message Security ilke adı.

Tanıtıcı: **MQCA_POLICY_NAME**.

Veri tipi: MQCFST

Değer: **2112** - IBM MQ 8.0 sürümünde ya da `cmqc.h` dosyasında tanımlanan sayısal bir değerdir.

Uzunluk üst sınırı: MQ_OBJECT_NAME_LENGTH.

Döndürülen: Her zaman.

PolicyVersion

Açıklama: Advanced Message Security ilke sürümü.

Tanıtıcı: **MQIA_POLICY_VERSION**

Veri tipi: MQCFIN

Değer **238** - IBM MQ 8.0 sürümünde ya da cmqc . h dosyasında tanımlanan sayısal bir değerdir.

Döndürülen: Her zaman

TolerateFlag

Açıklama: Advanced Message Security ilke toleransı işareti.

Tanıtıcı: **MQIA_TOLERATE_UNPROTECTED**

Veri tipi: MQCFIN

Değer **235** - IBM MQ 8.0 sürümünde ya da cmqc . h dosyasında tanımlanan sayısal bir değerdir.

Döndürülen: Her zaman.

SignatureAlgorithm

Açıklama: Advanced Message Security ilke imza algoritması.

Tanıtıcı: **MQIA_SIGNATURE_ALGORITHM**

Veri tipi: MQCFIN

Değer: **236** - IBM MQ 8.0 ' da ya da cmqc . h dosyasında tanımlanan sayısal bir değerdir.

Döndürülen: Advanced Message Security ilkesinde tanımlı bir imza algoritması olduğunda

EncryptionAlgorithm

Açıklama: Advanced Message Security ilke şifreleme algoritması.

Tanıtıcı: **MQIA_ENCRYPTION_ALGORITHM**

Veri tipi: MQCFIN

Değer: **237** - IBM MQ 8.0 ya da cmqc . h dosyasında tanımlanan sayısal bir değerdir.

Döndürülen: IBM MQ ilkesinde tanımlı bir şifreleme algoritması olduğunda

SignerDNs

Açıklama: İzin verilen imzalayıcıların konu DistinguishedName .

Tanıtıcı: **MQCA_SIGNER_DN**

Veri tipi: MQCFSL

Değer: **2113** - IBM MQ 8.0 sürümünde ya da cmqc . h dosyasında tanımlanan sayısal bir değerdir.

Uzunluk üst sınırı: İlkedeki en uzun imzalayıcı DN 'si, ancak bundan sonra MQ_AYIRT edici ad_uzunluk

Döndürülen: IBM MQ ilkesinde tanımlandığında.

RecipientDNs

Açıklama:	İzin verilen imzalayıcıların konu DistinguishedName .
Tanıtıcı:	**MQCA_RECIPIENT_DN**
Veri tipi:	MQCFSL
Değer:	**2114** - IBM MQ 8.0 ya da `cmqc.h` dosyasında tanımlanan sayısal bir değerdir.
Uzunluk üst sınırı:	İlkedeki en uzun alıcı ayırt edici adı (DN), ancak artık MQ_AYIRT edici ad_uzun değil.
Döndürülen:	IBM MQ ilkesinde tanımlandığında.

Özel notlar

Bu belge, ABD'de kullanıma sunulan ürünler ve hizmetler için hazırlanmıştır.

IBM, bu belgede sözü edilen ürün, hizmet ya da özellikleri diğer ülkelerde kullanıma sunmayabilir. Bulunduğunuz yerde kullanıma sunulan ürün ve hizmetleri yerel IBM müşteri temsilcisinden ya da çözüm ortağından öğrenebilirsiniz. Bir IBM ürün, program ya da hizmetine gönderme yapılması, açık ya da örtük olarak, yalnızca o IBM ürünü, programı ya da hizmetinin kullanılabileceğini göstermez. Aynı işlevi gören ve IBM'in fikri mülkiyet haklarına zarar vermeyen herhangi bir ürün, program ya da hizmet de kullanılabilir. Ancak, IBM dışı ürün, program ya da hizmetlerle gerçekleştirilen işlemlerin değerlendirilmesi ve doğrulanması kullanıcının sorumluluğundadır.

IBM'in, bu belgedeki konularla ilgili patentleri ya da patent başvuruları olabilir. Bu belgenin size verilmiş olması, patentlerin izinsiz kullanım hakkının da verildiği anlamına gelmez. Lisansla ilgili sorularınızı aşağıdaki adrese yazabilirsiniz:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Çift byte (DBCS) bilgilerle ilgili lisans soruları için, ülkenizdeki IBM'in Fikri Haklar (Intellectual Property) bölümüyle bağlantı kurun ya da sorularınızı aşağıda adrese yazın:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japonya

İzleyen paragraf, bu tür kayıt ve koşulların, yasalarıyla bağdaşmadığı ülkeler ya da bölgeler için geçerli değildir: IBM BU YAYINI, "OLDUĞU GİBİ" , HİÇBİR KONUDA AÇIK YA DA ÖRTÜK GARANTİ VERMEKSİZİN SAĞLAMAKTADIR; TİCARİ KULLANIMA UYGUNLUK AÇISINDAN HER TÜRLÜ GARANTİ VE BELİRLİ BİR AMACA UYGUNLUK İDDİASI AÇIKÇA REDDEDİLİR. Bazı ülkeler bazı işlemlerde garantinin açık ya da örtük olarak reddedilmesine izin vermez; dolayısıyla, bu bildirim sizin için geçerli olmayabilir.

Bu yayın teknik yanlışlar ya da yazım hataları içerebilir. Buradaki bilgiler üzerinde düzenli olarak değişiklik yapılmaktadır; söz konusu değişiklikler sonraki basımlara yansıtılacaktır. IBM, önceden bildirimde bulunmaksızın, bu yayında açıklanan ürünler ve/ya da programlar üzerinde iyileştirmeler ve/ya da değişiklikler yapabilir.

Bu belgede IBM dışı Web sitelerine yapılan göndermeler kullanıcıya kolaylık sağlamak içindir ve bu Web sitelerinin onaylanması anlamına gelmez. Bu Web sitelerinin içerdiği malzeme, bu IBM ürününe ilişkin malzemenin bir parçası değildir ve bu tür Web sitelerinin kullanılmasının sorumluluğu size aittir.

IBM'e bilgi ilettiğinizde, IBM bu bilgileri size karşı hiçbir yükümlülük almaksızın uygun gördüğü yöntemlerle kullanabilir ya da dağıtabilir.

(i) Bağımsız olarak yaratılan programlarla, bu program da içinde olmak üzere diğer programlar arasında bilgi değiş tokuşuna ve (ii) değiş tokuş edilen bilginin karşılıklı kullanımına olanak sağlamak amacıyla bu program hakkında bilgi sahibi olmak isteyen lisans sahipleri şu adrese yazabilirler:

IBM Corporation
Yazılım Birlikte Çalışabilirlik Koordinatörü, Bölüm 49XA
3605 Karayolu 52 N
Rochester, MN 55901
U.S.A.

Bu tür bilgiler, ilgili kayıt ve koşullar altında ve bazı durumlarda bedelli olarak edinilebilir.

Bu belgede açıklanan lisanslı program ve bu programla birlikte kullanılabilecek tüm lisanslı malzeme, IBM tarafından IBM Müşteri Sözleşmesi, IBM Uluslararası Program Lisans Sözleşmesi ya da taraflar arasında yapılan herhangi bir eşdeğer sözleşmenin koşulları kapsamında sağlanır.

Burada belirtilen performans verileri denetimli bir ortamda elde edilmiştir. Bu nedenle, başka işletim ortamlarında çok farklı sonuçlar alınabilir. Bazı ölçümler geliştirilme düzeyindeki sistemlerde yapılmıştır ve bu ölçümlerin genel kullanıma sunulan sistemlerde de aynı olacağı garanti edilemez. Ayrıca, bazı sonuçlar öngörü yöntemiyle elde edilmiş olabilir. Dolayısıyla, gerçek sonuçlar farklı olabilir. Bu belgenin kullanıcıları, kendi ortamları için geçerli verileri kendileri doğrulamalıdırlar.

IBM dışı ürünlerle ilgili bilgiler, bu ürünleri sağlayan firmalardan, bu firmaların yayın ve belgelerinden ve genel kullanıma açık diğer kaynaklardan alınmıştır. IBM bu ürünleri sınamamıştır ve IBM dışı ürünlerle ilgili performans doğruluğu, uyumluluk gibi iddiaları doğrulayamaz. IBM dışı ürünlerin yeteneklerine ilişkin sorular, bu ürünleri sağlayan firmalara yöneltilmelidir.

IBM'in gelecekteki yönelim ve kararlarına ilişkin tüm bildirimler değişebilir ve herhangi bir duyuruda bulunulmadan bunlardan vazgeçilebilir; bu yönelim ve kararlar yalnızca amaç ve hedefleri gösterir.

Bu belge, günlük iş ortamında kullanılan veri ve raporlara ilişkin örnekler içerir. Örneklerin olabildiğince açıklayıcı olması amacıyla kişi, şirket, marka ve ürün adları belirtilmiş olabilir. Bu adların tümü gerçek dışıdır ve gerçek iş ortamında kullanılan ad ve adreslerle olabilecek herhangi bir benzerlik tümüyle rastlantıdır.

YAYIN HAKKI LİSANSI:

Bu belge, çeşitli işletim platformlarında programlama tekniklerini gösteren, kaynak dilde yazılmış örnek uygulama programları içerir. Bu örnek programları, IBM'e herhangi bir ödemede bulunmadan, örnek programların yazıldığı işletim altyapısına ilişkin uygulama programlama arabirimiyle uyumlu uygulama programlarının geliştirilmesi, kullanılması, pazarlanması ya da dağıtılması amacıyla herhangi bir biçimde kopyalayabilir, değiştirebilir ve dağıtabilirsiniz. Bu örnekler her koşul altında tüm ayrıntılarıyla sınanmamıştır. Dolayısıyla, IBM bu programların güvenilirliği, bakım yapılabilirliği ya da işlevleri konusunda açık ya da örtük güvence veremez.

Bu bilgileri elektronik kopya olarak görüntülediyseniz, fotoğraflar ve renkli resimler görünmeyebilir.

Programlama arabirimi bilgileri

Sağlandıysa, programlama arabirimi bilgileri, bu programla birlikte kullanılmak üzere uygulama yazılımı oluşturmanıza yardımcı olmak amacıyla hazırlanmıştır.

Bu kitapta, müşterinin WebSphere MQhizmetlerini elde etmek üzere program yazmasına olanak sağlayan amaçlanan programlama arabirimlerine ilişkin bilgiler yer alır.

Ancak, bu bilgiler tanılama, değiştirme ve ayarlama bilgilerini de içerebilir. Tanılama, değiştirme ve ayarlama bilgileri, uygulama yazılımlarınızda hata ayıklamanıza yardımcı olur.

Önemli: Bu tanılama, değiştirme ve ayarlama bilgilerini bir programlama arabirimi olarak kullanmayın; bu bilgiler değişebilir.

Ticari Markalar

IBM, IBM logosu, ibm.com, IBM Corporation 'ın dünya çapında birçok farklı hukuk düzeninde kayıtlı bulunan ticari markalarıdır. IBM ticari markalarının güncel bir listesine Web üzerinde "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml(Telif hakkı ve ticari marka bilgileri) başlıklı konudan ulaşılabilir. Diğer ürün ve hizmet adları IBM'in veya diğer şirketlerin ticari markaları olabilir.

Microsoft ve Windows, Microsoft Corporation firmasının ABD'de ve/ya da diğer ülkelerdeki markalarıdır.

UNIX, The Open Group şirketinin ABD ve diğer ülkelerdeki tescilli ticari markasıdır.

Linux, Linus Torvalds'ın ABD ve/ya da diğer ülkelerdeki tescilli ticari markasıdır.

Bu ürün, Eclipse Project (https://www.eclipse.org/) tarafından geliştirilen yazılımları içerir.

Java ve Java tabanlı tüm markalar ve logolar, Oracle firmasının ve/ya da iştiraklerinin markaları ya da tescilli markalarıdır.

Parça numarası:

(1P) P/N: