Procesos de intercambio de paquetes IKEv1 e IKEv2 de IOS para perfiles con varios certificados

Contenido

Introducción

Este documento describe los procesos de intercambio de paquetes de Internet Key Exchange versión 1 (IKEv1) e Internet Key Exchange versión 2 (IKEv2) cuando se utiliza la autenticación de certificados y los posibles problemas que podrían ocurrir.

Esta es una lista de los temas que se describen en este documento:

- Los criterios de selección de certificados para el iniciador de Intercambio de claves de Internet (IKE) y el respondedor IKE
- El perfil IKE coincide con los criterios cuando se coinciden varios perfiles IKE (para escenarios de superposición y no superposición)
- La configuración y el comportamiento predeterminados cuando no se utilizan puntos de confianza en los perfiles IKE
- Las diferencias entre el IKEv1 y el IKEv2 en lo que respecta a los criterios de selección de

perfiles y certificados

Nota: Para obtener más información sobre cómo resolver un problema específico, consulte la sección correcta. Además, al final de este documento se proporciona un breve resumen.

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Configuración de VPN Cisco IOS®
- Protocolos IKEv1 e IKEv2 (intercambio de paquetes)

Componentes Utilizados

La información de este documento se basa en la versión 15.3T del IOS de Cisco.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Antecedentes

Los problemas que se describen en este documento surgen cuando se utilizan varios puntos de confianza y varios perfiles IKE.

Los ejemplos iniciales que se utilizan en este documento tienen un túnel de LAN a LAN IKEv1 con dos puntos de confianza en cada router. Al principio, puede parecer que la configuración es correcta. Sin embargo, el túnel VPN sólo se puede iniciar desde un lado de la conexión debido a la forma en que el comando **ca trust-point** se utiliza para el comportamiento del perfil de Asociación de seguridad de Internet y protocolo de administración de claves (ISAKMP) y para el orden de los certificados inscritos en el almacén local.

Se configura un comportamiento diferente con el comando **ca trust-point** para el perfil ISAKMP cuando el router es el iniciador ISAKMP. Puede ocurrir un problema porque el iniciador ISAKMP conoce el perfil ISAKMP desde el principio, de modo que el comando **ca trust-point** configurado para el perfil puede influir en la carga útil de la solicitud de certificado en el Paquete de modo principal 3 (MM3). Sin embargo, cuando el router es el respondedor ISAKMP, enlaza el tráfico entrante a un perfil ISAKMP específico después de recibir el Paquete de modo principal 5 (MM5),

que incluye el ID IKE necesario para crear el enlace. Esta es la razón por la que no es posible aplicar ningún comando **ca trust-point** para el paquete de modo principal 4 (MM4) porque el perfil no se determina antes del MM5.

En este documento se explica el orden de la carga útil de la solicitud del certificado en el MM3 y en el MM4 y el impacto en todo el proceso de negociación, así como la razón por la que sólo permite que la conexión se establezca desde un lado del túnel VPN.

A continuación se muestra un resumen de los comportamientos del iniciador y del respondedor de IKEv1:

	Iniciador IKEv1	Respondedor IKEv1
Enviar solicitud	Envía solicitudes específicas sólo para los puntos de confianza configurados en el perfil	Envía solicitudes para todos los puntos de confianza disponibles
Validar solicitud	Se valida con respecto a los puntos de confianza específicos que se configuran bajo el perfil	Se valida con respecto a los puntos de confianza específicos que se configuran bajo el perfil

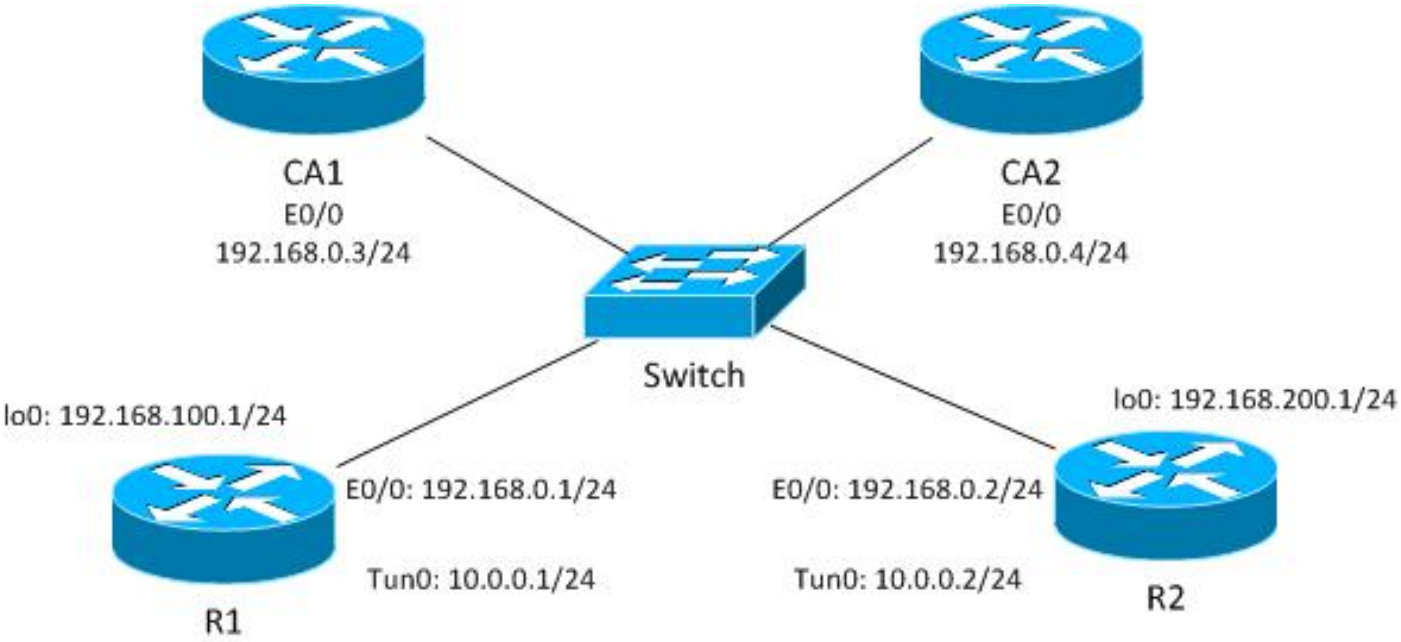
Cisco recomienda que no utilice el comando **ca trust-point** para los respondedores ISAKMP que tienen varios perfiles ISAKMP y utilizan puntos de confianza configurados globalmente. Para los iniciadores ISAKMP con varios perfiles ISAKMP, Cisco recomienda que limite el proceso de selección de certificados con el comando **ca trust-point** en cada perfil.

El protocolo IKEv2 tiene los mismos problemas que el protocolo IKEv1, pero el comportamiento diferente del comando **pki trustpoint** ayuda a prevenir la aparición de los problemas. Esto se debe a que el comando **pki trustpoint** es obligatorio para el iniciador IKEv2, mientras que el comando **ca trust-point** es opcional para el iniciador IKEv1. En algunas circunstancias (varios puntos de confianza en un solo perfil), podrían ocurrir los problemas descritos anteriormente. Por esta razón, Cisco recomienda que utilice configuraciones simétricas de punto de confianza para ambos lados de la conexión (los mismos puntos de confianza configurados en ambos perfiles IKEv2).

Topología

Esta es una topología genérica que se utiliza para todos los ejemplos de este documento.

> **Nota:** El router 1 (R1) y el router 2 (R2) utilizan interfaces de túnel virtual (VTI) para acceder a los loopbacks. Estos VTI están protegidos por IPSec.



Para este ejemplo de IKEv1, cada router tiene dos puntos de confianza para cada autoridad de certificación (CA) y se registran los certificados para cada uno de los puntos de confianza.

Cuando R1 es el iniciador ISAKMP, el túnel negocia correctamente y el tráfico está protegido. Debe ocurrir lo siguiente. Cuando R2 es el iniciador ISAKMP, la negociación Phase1 falla.

> **Nota:** Para los ejemplos de IKEv2 en este documento, la topología y el direccionamiento son los mismos que los que muestran el ejemplo de IKEv1.

Proceso de intercambio de paquetes

Esta sección describe las variaciones de configuración de IKEv1 e IKEv2 que se utilizan para el proceso de intercambio de paquetes, y los posibles problemas que podrían surgir.

IKEv1 con varios certificados

Esta es la configuración de red R1 y VPN para IKEv1 con varios certificados:

```
crypto isakmp policy 10
 encr 3des
 hash md5
 group 2

crypto isakmp profile prof1
   self-identity fqdn
   ca trust-point IOSCA1
   match identity host R2.cisco.com
```

```
!
crypto ipsec transform-set TS esp-aes esp-sha256-hmac
 mode tunnel
!
crypto ipsec profile prof1
 set transform-set TS
 set isakmp-profile prof1
!
interface Loopback0
 description Simulate LAN
 ip address 192.168.100.1 255.255.255.0
!
interface Tunnel1
 ip address 10.0.0.1 255.255.255.0
 tunnel source Ethernet0/0
 tunnel destination 192.168.0.2
 tunnel protection ipsec profile prof1
!
interface Ethernet0/0
 ip address 192.168.0.1 255.255.255.0

ip route 192.168.200.0 255.255.255.0 10.0.0.2
```

Esta es la configuración de red R2 y VPN para IKEv1 con varios certificados:

```
crypto isakmp policy 10
 encr 3des
 hash md5
 group 2

crypto isakmp profile prof1
  self-identity fqdn
  match identity host R1.cisco.com
!
crypto ipsec transform-set TS esp-aes esp-sha256-hmac
 mode tunnel
!
crypto ipsec profile prof1
 set transform-set TS
 set isakmp-profile prof1
!
interface Loopback0
 ip address 192.168.200.1 255.255.255.0
!
interface Tunnel1
 ip address 10.0.0.2 255.255.255.0
 tunnel source Ethernet0/0
 tunnel destination 192.168.0.1
 tunnel protection ipsec profile prof1
!
interface Ethernet0/0
 ip address 192.168.0.2 255.255.255.0

ip route 192.168.100.0 255.255.255.0 10.0.0.1
```

En este ejemplo, R1 tiene dos puntos de confianza: uno utiliza **IOSCA1** y el segundo utiliza **IOSCA2**:

```
crypto pki trustpoint IOSCA1
 enrollment url http://192.168.0.3:80
 serial-number
 fqdn R1.cisco.com
 ip-address 192.168.0.1
 subject-name CN=R1,OU=IT,O=cisco,O=com
 revocation-check crl
!
crypto pki trustpoint IOSCA2
 enrollment url http://192.168.0.4:80
 serial-number
 fqdn R1.cisco.com
 ip-address 192.168.0.1
 subject-name CN=R1,OU=IT,O=cisco,O=com
 revocation-check crl
```

En este ejemplo, R2 también tiene dos puntos de confianza: uno utiliza **IOSCA1** y el segundo utiliza **IOSCA2**:

```
crypto pki trustpoint IOSCA1
 enrollment url http://192.168.0.3:80
 serial-number
 fqdn R2.cisco.com
 ip-address 192.168.0.2
 subject-name CN=R2,OU=IT,O=cisco,O=com
 revocation-check crl
!
crypto pki trustpoint IOSCA2
 enrollment url http://192.168.0.4:80
 serial-number
 fqdn R2.cisco.com
 ip-address 192.168.0.2
 subject-name CN=R2,OU=IT,O=cisco,O=com
 revocation-check crl
```

Es importante tener en cuenta la diferencia única en estas configuraciones: el perfil ISAKMP R1 utiliza el comando **ca trust-point** para el punto de confianza **IOSCA1**, que indica que R1 confía solamente en los certificados validados por ese punto de confianza específico. Por el contrario, R2 confía en todos los certificados validados por todos los puntos de confianza definidos globalmente.

R1 como iniciador IKEv1

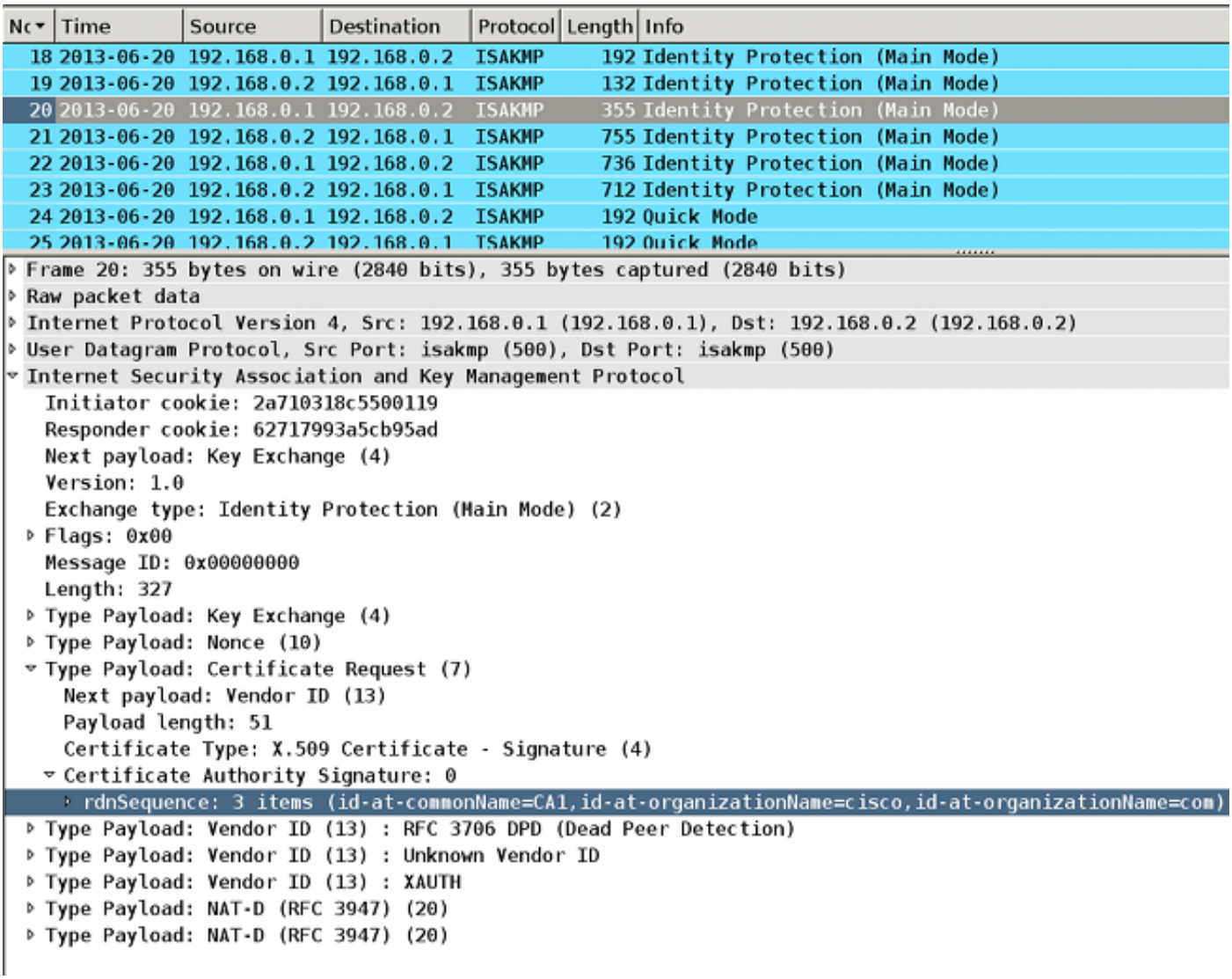
Estos son los comandos de depuración para R1 y R2:

- **R1# debug crypto isakmp**
- **debug crypto ipsec de R1#**
- **Validación de debug crypto pki de R1#**

Aquí, R1 inicia el túnel y envía el certificado que solicita el MM3:

```
*Jun 20 13:00:37.609: ISAKMP:(0): SA request profile is prof1
*Jun 20 13:00:37.610: ISAKMP (0): constructing CERT_REQ for issuer
 cn=CA1,o=cisco,o=com
*Jun 20 13:00:37.610: ISAKMP:(0): sending packet to 192.168.0.2
 my_port 500 peer_port 500 (I) MM_SA_SETUP
*Jun 20 13:00:37.610: ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM3
```

Es importante observar que el paquete contiene solamente una solicitud de certificado, que es sólo para el punto de confianza **IOSCA1**. Esto es un comportamiento esperado con la configuración actual del perfil ISAKMP (**CN=CA1, O=cisco, O=com**). No se envían otras solicitudes de certificado, que puede verificar con la función Captura de paquetes integrada:



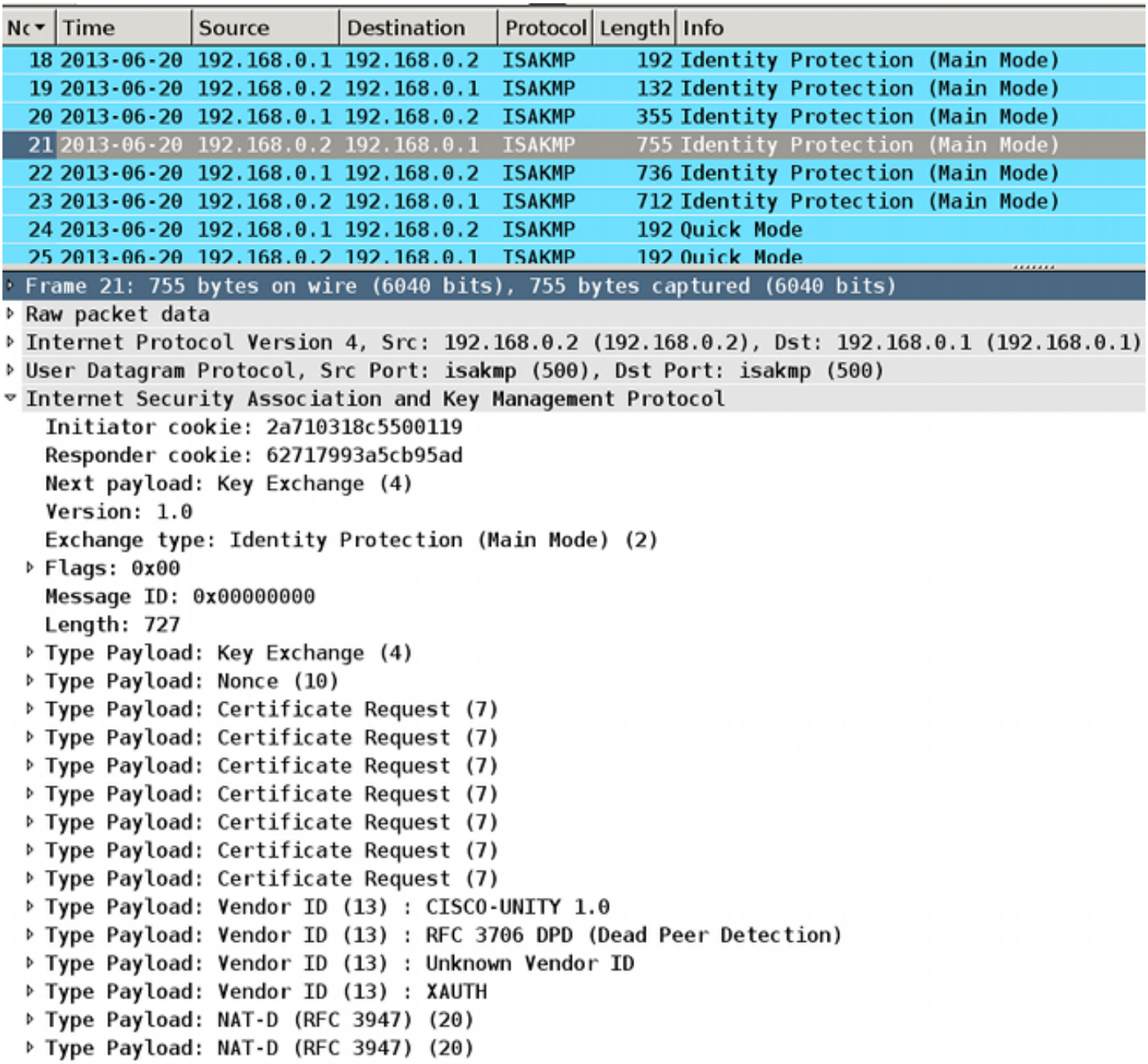
Cuando R2 recibe el paquete, comienza a procesar la solicitud de certificado, lo que crea una coincidencia que determina el punto de confianza y el certificado asociado que se utiliza para la autenticación en el MM5. El orden de proceso es el mismo que la carga útil de solicitud de certificado en el paquete ISAKMP. Esto significa que se utiliza la primera coincidencia. En este escenario, sólo hay una coincidencia ya que R1 se configura con un punto de confianza específico y envía solamente una solicitud de certificado asociada con el punto de confianza.

```
*Jun 20 13:00:37.617: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.617: ISAKMP:(1010): peer wants cert issued
 by cn=CA1,o=cisco,o=com
*Jun 20 13:00:37.617:  Choosing trustpoint IOSCA1 as issuer
```

Después, R2 prepara el MM4. Este es el paquete que contiene la solicitud de certificado para todos los puntos de confianza de confianza. Dado que R2 es el respondedor ISAKMP, todos los puntos de confianza definidos globalmente son de confianza (la configuración **ca trust-point** no está verificada). Dos de los puntos de confianza se definen manualmente (**IOSCA1** e **IOSCA2**), y el resto está predefinido.

```
*Jun 20 13:00:37.617: ISAKMP (1010): constructing CERT_REQ
 for issuer cn=CA1,o=cisco,o=com
*Jun 20 13:00:37.617: ISAKMP (1010): constructing CERT_REQ
 for issuer cn=CA2,o=cisco,o=com
*Jun 20 13:00:37.617: ISAKMP (1010): constructing CERT_REQ
 for issuer ou=Class 3 Public Primary Certification Authority,
 o=VeriSign, Inc.,c=US
*Jun 20 13:00:37.617: ISAKMP (1010): constructing CERT_REQ
 for issuer cn=Cisco SSCA2,o=Cisco Systems
*Jun 20 13:00:37.617: ISAKMP (1010): constructing CERT_REQ
 for issuer cn=Cisco Manufacturing CA,o=Cisco Systems
*Jun 20 13:00:37.617: ISAKMP (1010): constructing CERT_REQ
 for issuer cn=Cisco Root CA 2048,o=Cisco Systems
*Jun 20 13:00:37.617: ISAKMP (1010): constructing CERT_REQ
 for issuer cn=Cisco Root CA M1,o=Cisco
*Jun 20 13:00:37.617: ISAKMP:(1010): sending packet to
 192.168.0.1 my_port 500 peer_port 500 (R) MM_KEY_EXCH
*Jun 20 13:00:37.617: ISAKMP:(1010):Sending an IKE IPv4 Packet.
*Jun 20 13:00:37.617: ISAKMP:(1010):Input = IKE_MESG_INTERNAL,
 IKE_PROCESS_COMPLETE
*Jun 20 13:00:37.617: ISAKMP:(1010):Old State = IKE_R_MM3
 New State = IKE_R_MM4
```

Puede verificar el paquete con Wireshark. El paquete MM4 de R2 contiene siete entradas de solicitud de certificado:



No.	Time	Source	Destination	Protocol	Length	Info
18	2013-06-20	192.168.0.1	192.168.0.2	ISAKMP	192	Identity Protection (Main Mode)
19	2013-06-20	192.168.0.2	192.168.0.1	ISAKMP	132	Identity Protection (Main Mode)
20	2013-06-20	192.168.0.1	192.168.0.2	ISAKMP	355	Identity Protection (Main Mode)
21	2013-06-20	192.168.0.2	192.168.0.1	ISAKMP	755	Identity Protection (Main Mode)
22	2013-06-20	192.168.0.1	192.168.0.2	ISAKMP	736	Identity Protection (Main Mode)
23	2013-06-20	192.168.0.2	192.168.0.1	ISAKMP	712	Identity Protection (Main Mode)
24	2013-06-20	192.168.0.1	192.168.0.2	ISAKMP	192	Quick Mode
25	2013-06-20	192.168.0.2	192.168.0.1	ISAKMP	192	Quick Mode

A continuación, R1 recibe el MM4 de R2 con varios campos de solicitud de certificado:

```
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=CA1,o=cisco,o=com
*Jun 20 13:00:37.623: ISAKMP: Examining profile list for trustpoint IOSCA1
*Jun 20 13:00:37.623: ISAKMP: Found matching profile for IOSCA1
*Jun 20 13:00:37.623:  Choosing trustpoint IOSCA1 as issuer
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=CA2,o=cisco,o=com
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by ou=Class 3
 Public Primary Certification Authority,o=VeriSign, Inc.,c=US
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
```

```
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=Cisco SSCA2,o=Cisco Systems
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=Cisco Manufacturing CA,o=Cisco Systems
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=Cisco Root CA 2048,o=Cisco Systems
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=Cisco Root CA M1,o=Cisco
```

La regla de primera coincidencia en R1 coincide con la primera solicitud de certificado con el punto de confianza **IOSCA1**. Esto determina que R1 utiliza el certificado asociado con **IOSCA1** de punto de confianza para la autenticación en el MM5. El nombre de dominio completo (FQDN) se utiliza como ID IKE. Esto se debe a la configuración **auto-identity fqdn** en el perfil ISAKMP:

```
*Jun 20 13:00:37.624: ISAKMP (1010): constructing CERT payload for serialNumber=
 100+ipaddress=192.168.0.1+hostname=R1.cisco.com,cn=R1,ou=IT,o=cisco,o=com
*Jun 20 13:00:37.624: ISAKMP:(1010): using the IOSCA1 trustpoint's
 keypair to sign
```

R2 recibe y procesa el MM5. La ID IKE recibida (**R1.cisco.com**) coincide con el **prof1** del perfil ISAKMP. El certificado recibido se valida y la autenticación es exitosa:

```
*Jun 20 13:00:37.625: ISAKMP:(1010): processing ID payload. message ID = 0
*Jun 20 13:00:37.625: ISAKMP (1010): ID payload
       next-payload : 6
       type         : 2
       FQDN name    : R1.cisco.com
       protocol     : 17
       port         : 500
       length       : 20
*Jun 20 13:00:37.625: ISAKMP:(0):: peer matches prof1 profile
..........
*Jun 20 13:00:37.626: CRYPTO_PKI: (A0013) Certificate validation succeeded
..........
*Jun 20 13:00:37.626: ISAKMP:(1010):SA authentication status:
       authenticated
```

A continuación, R2 prepara el MM6 con el certificado asociado con **IOSCA1**:

```
*Jun 20 13:00:37.627: ISAKMP (1010): constructing CERT payload for serialNumber=
 101+ipaddress=192.168.0.2+hostname=R2.cisco.com,cn=R2,ou=IT,o=cisco,o=com
*Jun 20 13:00:37.627: ISAKMP:(1010): using the IOSCA1 trustpoint's keypair to sign
*Jun 20 13:00:37.632: ISAKMP:(1010): sending packet to 192.168.0.1
 my_port 500 peer_port 500 (R) MM_KEY_EXCH
```

R1 recibe el paquete y R1 verifica el certificado y la autenticación:

```
*Jun 20 13:00:37.632: ISAKMP (1010): received packet from 192.168.0.2
 dport 500 sport 500 Global (I) MM_KEY_EXCH
*Jun 20 13:00:37.632: ISAKMP:(1010): processing ID payload. message ID = 0
*Jun 20 13:00:37.632: ISAKMP (1010): ID payload
       next-payload : 6
       type         : 2
       FQDN name    : R2.cisco.com
       protocol     : 17
       port         : 500
       length       : 20
....
*Jun 20 13:00:37.632: ISAKMP:(0): Creating CERT validation list: IOSCA1
....
*Jun 20 13:00:37.633: CRYPTO_PKI: (80013) Certificate validation succeeded
....
*Jun 20 13:00:37.637: ISAKMP:(1010):SA authentication status:
       authenticated
*Jun 20 13:00:37.637: ISAKMP:(1010):Old State = IKE_I_MM6
 New State = IKE_P1_COMPLETE
```

Esto completa la Fase 1. La fase 2 se negocia como siempre. El túnel se establece correctamente y el tráfico está protegido.

R2 como iniciador IKEv1

Este ejemplo describe el proceso cuando R2 inicia el mismo túnel IKEv1 y explica por qué no se establece.

> **Nota:** Se eliminan partes de los registros para centrarse solamente en las diferencias en relación con el ejemplo presentado en la sección anterior.

R2 envía el MM3 con siete cargas útiles de solicitud de certificado porque R2 no tiene un punto de confianza asociado con el perfil ISAKMP (todos los puntos de confianza son de confianza):

```
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer cn=CA1,o=cisco,o=com
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer cn=CA2,o=cisco,o=com
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer ou=Class 3 Public Primary Certification Authority,
 o=VeriSign, Inc.,c=US
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer cn=Cisco SSCA2,o=Cisco Systems
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer cn=Cisco Manufacturing CA,o=Cisco Systems
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
```

```
 issuer cn=Cisco Root CA 2048,o=Cisco Systems
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer cn=Cisco Root CA M1,o=Cisco
*Jun 17 18:08:44.321: ISAKMP (0): sending packet to 192.168.0.1
 my_port 500 peer_port 500 (I) MM_SA_SETUP
```

Cuando R1 recibe el paquete de R2, procesa la solicitud de certificado y coincide con el punto de confianza IOSCA1, que determina el certificado que se envía en el MM6:

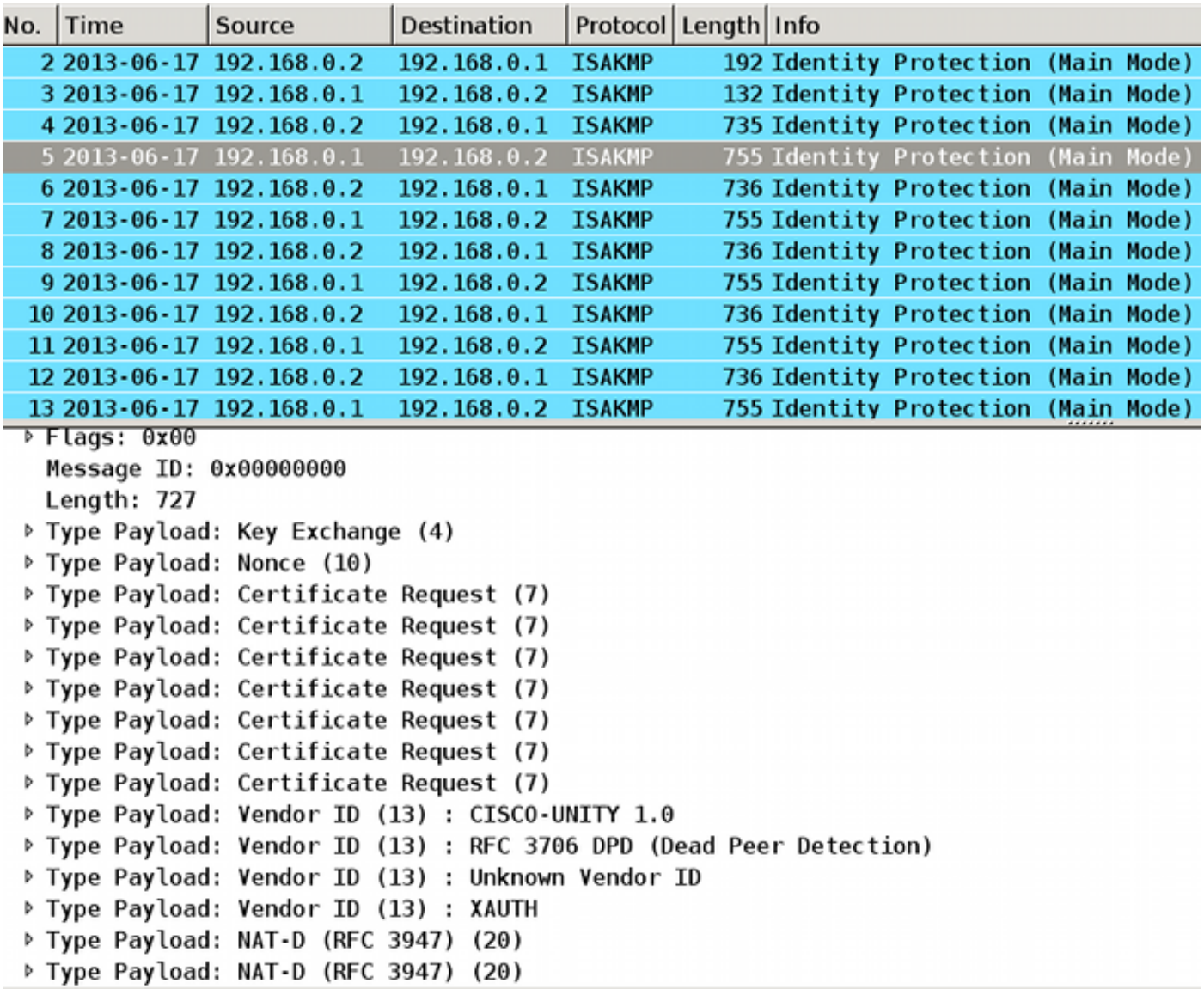
```
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=CA1,o=cisco,o=com
*Jun 17 18:08:14.321:  Choosing trustpoint IOSCA1 as issuer
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=CA2,o=cisco,o=com
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 ou=Class 3 Public Primary Certification Authority,o=VeriSign, Inc.,c=US
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=Cisco SSCA2,o=Cisco Systems
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=Cisco Manufacturing CA,o=Cisco Systems
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=Cisco Root CA 2048,o=Cisco Systems
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=Cisco Root CA M1,o=Cisco
```

Después, R1 prepara el paquete MM4 con la carga útil de solicitud de certificado. Ahora hay varias cargas útiles de solicitudes de certificados:

```
*Jun 17 18:08:14.321: ISAKMP (1099): constructing CERT_REQ for issuer
 cn=CA2,o=cisco,o=com
*Jun 17 18:08:14.321: ISAKMP (1099): constructing CERT_REQ for issuer
 cn=CA1,o=cisco,o=com
*Jun 17 18:08:14.322: ISAKMP (1099): constructing CERT_REQ for issuer
 ou=Class 3 Public Primary Certification Authority,
 o=VeriSign, Inc.,c=US
*Jun 17 18:08:14.322: ISAKMP (1099): constructing CERT_REQ for issuer
```

```
 cn=Cisco SSCA2,o=Cisco Systems
*Jun 17 18:08:14.322: ISAKMP (1099): constructing CERT_REQ for issuer
 cn=Cisco Manufacturing CA,o=Cisco Systems
*Jun 17 18:08:14.322: ISAKMP (1099): constructing CERT_REQ for issuer
 cn=Cisco Root CA 2048,o=Cisco Systems
*Jun 17 18:08:14.322: ISAKMP (1099): constructing CERT_REQ for issuer
 cn=Cisco Root CA M1,o=Cisco
*Jun 17 18:08:14.322: ISAKMP:(1099): sending packet to 192.168.0.2
 my_port 500 peer_port 500 (R) MM_KEY_EXCH
```

Verifique los registros con Embedded Packet Capture (EPC) y Wireshark:



No.	Time	Source	Destination	Protocol	Length	Info
2	2013-06-17	192.168.0.2	192.168.0.1	ISAKMP	192	Identity Protection (Main Mode)
3	2013-06-17	192.168.0.1	192.168.0.2	ISAKMP	132	Identity Protection (Main Mode)
4	2013-06-17	192.168.0.2	192.168.0.1	ISAKMP	735	Identity Protection (Main Mode)
5	2013-06-17	192.168.0.1	192.168.0.2	ISAKMP	755	Identity Protection (Main Mode)
6	2013-06-17	192.168.0.2	192.168.0.1	ISAKMP	736	Identity Protection (Main Mode)
7	2013-06-17	192.168.0.1	192.168.0.2	ISAKMP	755	Identity Protection (Main Mode)
8	2013-06-17	192.168.0.2	192.168.0.1	ISAKMP	736	Identity Protection (Main Mode)
9	2013-06-17	192.168.0.1	192.168.0.2	ISAKMP	755	Identity Protection (Main Mode)
10	2013-06-17	192.168.0.2	192.168.0.1	ISAKMP	736	Identity Protection (Main Mode)
11	2013-06-17	192.168.0.1	192.168.0.2	ISAKMP	755	Identity Protection (Main Mode)
12	2013-06-17	192.168.0.2	192.168.0.1	ISAKMP	736	Identity Protection (Main Mode)
13	2013-06-17	192.168.0.1	192.168.0.2	ISAKMP	755	Identity Protection (Main Mode)

Aunque R1 se configura para un único punto de confianza (**IOSCA1**) en el perfil ISAKMP, se envían varias solicitudes de certificado. Esto ocurre porque el comando **ca trust-point** en el perfil ISAKMP determina la carga útil de solicitud de certificado, pero sólo cuando el router es el iniciador de la sesión ISAKMP. Si el router es el respondedor, hay varias cargas útiles de solicitud de certificado para todos los puntos de confianza definidos globalmente porque R1 todavía no conoce el perfil ISAKMP que se utiliza para la sesión IKE.

La sesión IKE entrante se enlaza a un perfil ISAKMP específico después de la recepción del MM5, que incluye el ID IKE. Después, el comando **match identity** para el perfil específico vincula

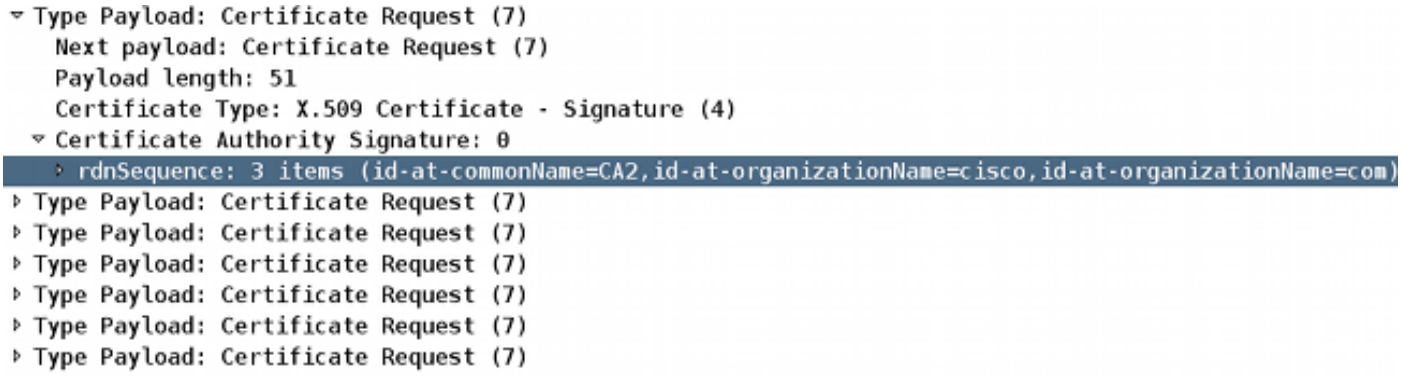
la sesión IKE al perfil. Sin embargo, el router no puede determinar esto hasta ahora. Puede haber varios perfiles ISAKMP con diferentes comandos **ca trust-point** configurados para cada perfil.

Por esta razón, R1 debe enviar la solicitud de certificado para todos los puntos de confianza configurados globalmente.

Consulte la referencia de comandos para el comando **ca trust-point**:

Un router que inicia IKE y un router que responde a la solicitud IKE deben tener configuraciones simétricas de punto de confianza. Por ejemplo, un router de respuesta (en el modo principal IKE) que realiza el cifrado y la autenticación de firma RSA puede utilizar puntos de confianza definidos en la configuración global al enviar las cargas útiles de CERT-REQ. Sin embargo, el router podría utilizar una lista restringida de puntos de confianza que se definieron en el perfil ISAKMP para la verificación del certificado. Si el par (el iniciador IKE) está configurado para utilizar un certificado cuyo punto de confianza está en la lista global del router que responde pero no en el perfil ISAKMP del router que responde, el certificado se rechaza. (Sin embargo, si el router de inicio no conoce los puntos de confianza en la configuración global del router de respuesta, el certificado aún puede ser autenticado.)

Ahora verifique los detalles del paquete MM4 para detectar la primera carga útil de solicitud de certificado:



El paquete MM4 que se envía desde R1 incluye el punto de confianza **IOSCA2** en la primera carga útil de solicitud de certificado debido al orden en que se instalan los certificados; el primero está firmado por el punto de confianza **IOSCA2**:

```
R1#sh crypto pki certificates
Certificate
 Status: Available
 Certificate Serial Number (hex): 03
 Certificate Usage: General Purpose
 Issuer:
    cn=CA2
   o=cisco
   o=com
 Subject:
   Name: R1.cisco.com
   IP Address: 192.168.0.1
   Serial Number: 100
   serialNumber=100+ipaddress=192.168.0.1+hostname=R1.cisco.com
   cn=R1
   ou=IT
```

```
   o=cisco
   o=com
 Validity Date:
   start date: 13:25:01 CET Jun 17 2013
   end   date: 13:25:01 CET Jun 17 2014
 Associated Trustpoints: IOSCA2
...
<output omitted, 1 more R1 cert signed by CA1, 2 more CA certs>
```

Haga una comparación con el paquete MM3 que se envía desde R2 cuando el punto de confianza **IOSCA1** se incluye en la primera carga útil de solicitud de certificado:

```
R2#sh crypto pki certificates
Certificate
 Status: Available
 Certificate Serial Number (hex): 02
 Certificate Usage: General Purpose
 Issuer:
   cn=CA1
   o=cisco
   o=com
 Subject:
   Name: R2.cisco.com
   IP Address: 192.168.0.2
   Serial Number: 101
   serialNumber=101+ipaddress=192.168.0.2+hostname=R2.cisco.com
   cn=R2
   ou=IT
   o=cisco
   o=com
 Validity Date:
   start date: 13:23:49 CET Jun 17 2013
   end   date: 13:23:49 CET Jun 17 2014
 Associated Trustpoints: IOSCA1
 Storage: nvram:CA1#2.cer
...
<output omitted, 1 more R2 cert signed by CA2, 2 more CA certs>
```

Ahora R2 recibe el paquete MM4 de R1 y comienza a procesar la solicitud de certificado. La primera carga útil de solicitud de certificado coincide con el punto de confianza **IOSCA2**:

```
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=CA2,o=cisco,o=com
*Jun 17 18:08:44.335:  Choosing trustpoint IOSCA2 as issuer
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=CA1,o=cisco,o=com
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 ou=Class 3 Public Primary Certification Authority,o=VeriSign, Inc.,c=US
```

```
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=Cisco SSCA2,o=Cisco Systems
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=Cisco Manufacturing CA,o=Cisco Systems
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=Cisco Root CA 2048,o=Cisco Systems
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=Cisco Root CA M1,o=Cisco
```

Cuando R2 prepara el paquete MM5, utiliza el certificado asociado al punto de confianza IOSCA2:

```
*Jun 17 18:08:44.335: ISAKMP:(1100):SA is doing RSA signature authentication
 using id type ID_FQDN
*Jun 17 18:08:44.335: ISAKMP (1100): ID payload
       next-payload : 6
       type         : 2
       FQDN name    : R2.cisco.com
       protocol     : 17
       port         : 500
       length       : 20
*Jun 17 18:08:44.335: ISAKMP:(1100):Total payload length: 20
*Jun 17 18:08:44.335: ISAKMP:(1100): IKE->PKI Get CertificateChain to be sent
 to peer state (I) MM_KEY_EXCH (peer 192.168.0.1)
*Jun 17 18:08:44.335: ISAKMP:(1100): PKI->IKE Got CertificateChain to be sent
 to peer state (I) MM_KEY_EXCH (peer 192.168.0.1)
*Jun 17 18:08:44.336: ISAKMP (1100): constructing CERT payload for
 serialNumber=101+ipaddress=192.168.0.2+hostname=R2.cisco.com,cn=R2,
 ou=IT,o=cisco,o=com
R2#
*Jun 17 18:08:44.336: ISAKMP:(1100): using the IOSCA2 trustpoint's
 keypair to sign
*Jun 17 18:08:44.336: ISAKMP:(1100): sending packet to 192.168.0.1
 my_port 500 peer_port 500 (I) MM_KEY_EXCH
*Jun 17 18:08:44.336: ISAKMP:(1100):Sending an IKE IPv4 Packet.
```

R1 recibe el paquete MM5. Debido a que R1 confía solamente en el punto de confianza IOSCA1 (para el perfil ISAKMP prof1), la validación del certificado falla:

```
*Jun 17 18:08:44.337: ISAKMP (1100): received packet from 192.168.0.2
 dport 500 sport 500 Global (R) MM_KEY_EXCH
*Jun 17 18:08:44.337: ISAKMP:(1100):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
*Jun 17 18:08:44.337: ISAKMP:(1100):Old State =IKE_R_MM4  New State = IKE_R_MM5

*Jun 17 18:08:44.337: ISAKMP:(1100): processing ID payload. message ID = 0
*Jun 17 18:08:44.337: ISAKMP (1100): ID payload
```

```
       next-payload : 6
       type         : 2
       FQDN name    : R2.cisco.com
       protocol     : 17
       port         : 500
       length       : 20
*Jun 17 18:08:44.337: ISAKMP:(0):: peer matches prof1 profile
*Jun 17 18:08:44.337: ISAKMP:(1100): processing CERT payload. message ID = 0
*Jun 17 18:08:44.337: ISAKMP:(1100): processing a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.337: ISAKMP:(1100): IKE->PKI Add peer's certificate state
 (R) MM_KEY_EXCH (peer 192.168.0.2)
*Jun 17 18:08:44.337: CRYPTO_PKI: (900C5) Adding peer certificate
*Jun 17 18:08:44.337: ISAKMP:(1100): PKI->IKE Added peer's certificate state
 (R) MM_KEY_EXCH (peer 192.168.0.2)
*Jun 17 18:08:44.337: ISAKMP:(1100): IKE->PKI Get PeerCertificateChain state
 (R) MM_KEY_EXCH (peer 192.168.0.2)
*Jun 17 18:08:44.337: ISAKMP:(1100): PKI->IKE Got PeerCertificateChain state
 (R) MM_KEY_EXCH (peer 192.168.0.2)
*Jun 17 18:08:44.337: ISAKMP:(1100): peer's pubkey isn't cached
*Jun 17 18:08:44.337: ISAKMP:(1100):Profile has no keyring, aborting key search
*Jun 17 18:08:44.337: ISAKMP:(0): Creating CERT validation list: IOSCA1,
*Jun 17 18:08:44.337: ISAKMP:(1100): IKE->PKI Validate certificate chain state
 (R) MM_KEY_EXCH (peer 192.168.0.2)
*Jun 17 18:08:44.337: CRYPTO_PKI:ip-ext-val:IP extension validation not required
*Jun 17 18:08:44.341: CRYPTO_PKI: (900C5) Check for identical certs
*Jun 17 18:08:44.341: CRYPTO_PKI: (900C5) Create a list of suitable trustpoints
*Jun 17 18:08:44.341: CRYPTO_PKI: (900C5) No suitable trustpoints found
*Jun 17 18:08:44.341: ISAKMP:(1100): PKI->IKE Validate certificate chain state
 (R) MM_KEY_EXCH (peer 192.168.0.2)
*Jun 17 18:08:44.341: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from
 192.168.0.2 is bad: unknown error returned in certificate validation
R1#
*Jun 17 18:08:44.341: ISAKMP:(1100): Unknown error in cert validation, -1
```

Esta configuración funciona si el orden de la inscripción de certificados en R1 es diferente porque el primer certificado mostrado está firmado por el punto de confianza **IOSCA1**. Además, la primera carga útil de solicitud de certificado en el MM4 es el punto de confianza **IOSCA1**, que luego es elegido por R2 y validado con éxito en R1 en el MM6.

IKEv1 sin un comando *ca trust-point* en el perfil

Para escenarios con múltiples perfiles y puntos de confianza pero sin una configuración específica de punto de confianza en los perfiles, no hay problemas porque no hay validación de puntos de confianza específicos determinados por una configuración de comando **ca trust-point**. Sin embargo, es posible que el proceso de selección no sea obvio. Según el router que es el iniciador, se seleccionan los diferentes certificados para el proceso de autenticación en relación con el orden de inscripción de certificados.

A veces, un certificado puede ser soportado solamente por un lado de la conexión, como en x509 Versión 1, que no es una función hash típica que se utiliza para firmar. El túnel VPN puede establecerse sólo desde un lado de la conexión.

Referencia RFC para IKEv1

Aquí hay un fragmento de RFC4945:

3.2.7.1. Especificación de las Autoridades de Certificación

Al **solicitar** el intercambio en banda de materiales de codificación, las implementaciones DEBEN generar CERTREQ para cada anclaje de confianza de pares que la **política local** considere confiables **explícitamente** durante un intercambio determinado.
El RFC no está claro. La **política local** podría relacionarse explícitamente con el **comando ca trust-point** configurado en el perfil ISAKMP criptográfico. El problema es que en la etapa MM3 y MM4 del proceso, no puede seleccionar un perfil ISAKMP a menos que use una dirección IP para la identidad y los puntos de confianza porque la autenticación en la etapa MM5 y MM6 del proceso debe ocurrir primero. Por esta razón, la **política local** se relaciona **explícitamente** con todos los puntos de confianza configurados en el dispositivo.

> **Nota:** Esta información no es específica de Cisco, pero es específica de IKEv1.

Selección de Perfil IKEv2 con Identidades que Superponen

Antes de describir varios certificados para IKEv2, es importante saber cómo se seleccionan los perfiles cuando se utiliza la identidad de coincidencia, que se satisface para todos los perfiles. Este no es un escenario recomendado porque los resultados de la negociación IKEv2 dependen de varios factores. Los mismos problemas existen para IKEv1 cuando se utilizan perfiles que se superponen.

A continuación se muestra un ejemplo de configuración del iniciador de IKEv2:

```
crypto ikev2 proposal prop-1
 encryption 3des
 integrity md5
 group 2
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 profile profile1
 match identity remote address 192.168.0.2 255.255.255.255
 identity local address 192.168.0.1
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint TP1

crypto ipsec transform-set trans esp-3des esp-sha-hmac
 mode tunnel
!
crypto ipsec profile profile1
 set transform-set trans
 set ikev2-profile profile1
!
interface Loopback0
```

```
 ip address 192.168.100.1 255.255.255.255
!
interface Tunnel1
 ip address 10.0.0.1 255.255.255.0
 tunnel source Ethernet0/0
 tunnel destination 192.168.0.2
 tunnel protection ipsec profile profile1
!
interface Ethernet0/0
 ip address 192.168.0.1 255.255.255.0

ip route 192.168.200.1 255.255.255.255 10.0.0.2
```

La dirección del tipo de identidad se utiliza para ambos lados de la conexión. La autenticación mediante certificados (también puede ser claves previamente compartidas) no es importante para este ejemplo. El respondedor tiene varios perfiles que coinciden con el tráfico IKEv2 entrante:

```
crypto ikev2 proposal prop-1
 encryption 3des
 integrity md5
 group 2
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 profile profile1
 match identity remote address 192.168.0.1 255.255.255.255
 identity local address 192.168.0.2
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint TP1
!
crypto ikev2 profile profile2
 match identity remote address 192.168.0.1 255.255.255.255
 identity local address 192.168.0.2
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint TP1
!
crypto ikev2 profile profile3
 match identity remote address 192.168.0.1 255.255.255.255
 identity local address 192.168.0.2
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint TP1

crypto ipsec transform-set trans esp-3des esp-sha-hmac
 mode tunnel
!
crypto ipsec profile profile1
 set transform-set trans
 set ikev2-profile profile1
!
interface Loopback0
 ip address 192.168.200.1 255.255.255.255
!
interface Tunnel1
 ip address 10.0.0.2 255.255.255.0
 tunnel source Ethernet0/0
```

```
 tunnel destination 192.168.0.1
 tunnel protection ipsec profile profile1
!
interface Ethernet0/0
 ip address 192.168.0.2 255.255.255.0

ip route 192.168.100.1 255.255.255.255 10.0.0.1
```

El iniciador envía el tercer paquete IKEv2 y el respondedor debe elegir el perfil en función de la identidad recibida. La identidad es una dirección IPv4 (**192.168.0.1**):

```
IKEv2:(SA ID = 1):Searching policy based on peer's identity '192.168.0.1' of
 type 'IPv4 address'
```

Todos los perfiles satisfacen esta identidad debido al comando **match identity** configurado. El IOS elige el último en la configuración, que es **profile3** en este ejemplo:

```
IKEv2:found matching IKEv2 profile 'profile3'
```

Para verificar el orden, ingrese el comando **show crypto ikev2 profile**.

> **Nota:** Incluso cuando hay una dirección genérica (0.0.0.0) en el perfil, todavía está seleccionada. El IOS no intenta encontrar una mejor coincidencia; intenta encontrar la primera coincidencia. Sin embargo, esto sólo ocurre porque todos los perfiles tienen el mismo comando **match identity remote** configurado. Para los perfiles IKEv1 e IKEv2 que tienen diferentes reglas de identidad de coincidencia, siempre se utiliza la más específica. Cisco recomienda que no tenga los perfiles configurados con el comando **superposición de coincidencia de identidad** porque es difícil predecir el perfil seleccionado.

En este escenario, el respondedor selecciona **profile3**, pero **profile1** se utiliza para la interfaz de túnel. Esto hace que aparezca un error cuando se negocia el ID de proxy:

```
*Jul 17 09:23:48.187: map_db_check_isakmp_profile profile did not match
*Jul 17 09:23:48.187: map_db_find_best did not find matching map
*Jul 17 09:23:48.187: IPSEC(ipsec_process_proposal):
 proxy identities not supported
*Jul 17 09:23:48.187: IKEv2:(SA ID = 1):There was no
 IPSEC policy found for received TS
*Jul 17 09:23:48.187: IKEv2:(SA ID = 1):
*Jul 17 09:23:48.187: IKEv2:(SA ID = 1):Sending TS unacceptable notify
```

Flujo de IKEv2 cuando se utilizan certificados

Cuando los certificados se utilizan para IKEv2 para la autenticación, el iniciador no envía la carga

útil de solicitud de certificado en el primer paquete:

```
IKEv2 IKE_SA_INIT Exchange REQUEST
Payload contents:
 SA KE N VID VID NOTIFY(NAT_DETECTION_SOURCE_IP)
 NOTIFY(NAT_DETECTION_DESTINATION_IP)
```

El respondedor responde con la carga útil de solicitud de certificado (segundo paquete) y todas las CA porque el respondedor no conoce el perfil que se debe utilizar en esta etapa. El paquete que contiene la información se envía al iniciador:

```
IKEv2 IKE_SA_INIT Exchange RESPONSE
Payload contents:
 SA KE N VID VID NOTIFY(NAT_DETECTION_SOURCE_IP) NOTIFY
 (NAT_DETECTION_DESTINATION_IP) CERTREQ NOTIFY(HTTP_CERT_LOOKUP_SUPPORTED)
```

El iniciador procesa el paquete y elige un punto de confianza que coincida con la CA propuesta:

```
IKEv2:(SA ID = 1):[IKEv2 -> PKI] Retrieving trustpoint(s) from
 received certificate hash(es)
IKEv2:(SA ID = 1):[PKI -> IKEv2] Retrieved trustpoint(s): 'TP1'
```

El iniciador entonces envía el tercer paquete con la solicitud de certificado y la carga útil del certificado. Este paquete ya está cifrado con material de codificación de la fase Diffie-Hellman (DH):

```
IKEv2:(SA ID = 1):Building packet for encryption.
Payload contents:
 VID IDi CERT CERTREQ NOTIFY(HTTP_CERT_LOOKUP_SUPPORTED) AUTH CFG SA TSi
 TSr NOTIFY(INITIAL_CONTACT) NOTIFY(SET_WINDOW_SIZE) NOTIFY(ESP_TFC_NO_SUPPORT)
 NOTIFY(NON_FIRST_FRAGS)
```

El cuarto paquete se envía del respondedor al iniciador y contiene solamente la carga útil del certificado:

```
IKEv2 IKE_AUTH Exchange RESPONSE
Payload contents:
 VID IDr CERT AUTH SA TSi TSr NOTIFY(SET_WINDOW_SIZE) NOTIFY(ESP_TFC_NO_SUPPORT)
NOTIFY(NON_FIRST_FRAGS)
```

El flujo descrito aquí es similar al flujo IKEv1. El respondedor debe enviar la carga útil de solicitud de certificado por adelantado sin conocer el perfil que debe utilizarse, lo que crea los mismos problemas que se describieron previamente para IKEv1 (desde una perspectiva de protocolo). Sin embargo, la implementación en el IOS es mejor para el IKEv2 que para el IKEv1.

Punto de confianza obligatorio IKEv2 para el iniciador

Este es un ejemplo de cuándo un iniciador IKEv2 intenta utilizar un perfil con autenticación de certificado y no tiene ningún punto de confianza configurado bajo ese perfil:

```
crypto ikev2 profile profile1
 match identity remote address 192.168.0.2 255.255.255.255
 identity local address 192.168.0.1
 authentication remote rsa-sig
 authentication local rsa-sig
```

El primer paquete se envía sin ninguna carga útil de solicitud de certificado, como se ha descrito anteriormente. La respuesta del respondedor incluye la carga útil de solicitud de certificado para todos los puntos de confianza definidos en el modo de configuración global. Esto es recibido por el iniciador:

```
*Jul 17 17:40:43.183: IKEv2:(SA ID = 1):[IKEv2 -> PKI] Retrieving trustpoint(s)
 from received certificate hash(es)
*Jul 17 17:40:43.183: IKEv2:(SA ID = 1):[PKI -> IKEv2] Retrieved
 trustpoint(s): 'TP1'
*Jul 17 17:40:43.183: CRYPTO_PKI: crypto_pki_get_cert_record_by_subject()
*Jul 17 17:40:43.183: CRYPTO_PKI: Found a subject match
*Jul 17 17:40:43.183: CRYPTO_PKI: crypto_pki_get_cert_record_by_subject()
*Jul 17 17:40:43.183: CRYPTO_PKI: Found a subject match
*Jul 17 17:40:43.183: CRYPTO_PKI: Trust-Point TP1 picked up
*Jul 17 17:40:43.183: CRYPTO_PKI: 1 matching trustpoints found
*Jul 17 17:40:43.183: IKEv2:(SA ID = 1):[IKEv2 -> PKI] Retrieving trustpoint(s)
 from received certificate hash(es)
*Jul 17 17:40:43.183: IKEv2:(SA ID = 1):[PKI -> IKEv2] Retrieved
 trustpoint(s): 'TP2'
*Jul 17 17:40:43.183: CRYPTO_PKI: Trust-Point TP2 picked up
*Jul 17 17:40:43.183: CRYPTO_PKI: crypto_pki_get_cert_record_by_subject()
*Jul 17 17:40:43.183: CRYPTO_PKI: Found a subject match
*Jul 17 17:40:43.183: CRYPTO_PKI: crypto_pki_get_cert_record_by_subject()
*Jul 17 17:40:43.183: CRYPTO_PKI: Found a subject match
*Jul 17 17:40:43.183: CRYPTO_PKI: 1 matching trustpoints found
*Jul 17 17:40:43.183: IKEv2:(SA ID = 1):Failed to build certificate payload
```

El iniciador no conoce el punto de confianza que se debe utilizar para firmar. Esta es la diferencia principal cuando se compara la implementación de IKEv2 con IKEv1. El iniciador IKEv2 debe tener el punto de confianza configurado bajo el perfil del iniciador IKEv2, pero no es necesario para el respondedor IKEv2.

Aquí hay un extracto de la referencia de comandos:

Si no hay ningún punto de confianza definido en la configuración del perfil IKEv2, el valor predeterminado es **validar el certificado** usando todos los puntos de confianza definidos en la configuración global

Es posible definir diferentes puntos de confianza; uno para firmar y otro diferente para validar. Desafortunadamente, el punto de confianza obligatorio configurado bajo el perfil IKEv2 no

resuelve todos los problemas.

R2 como iniciador IKEv2

En este ejemplo, R2 es el iniciador IKEv2:

```
crypto ikev2 profile profile1
 match identity remote address 192.168.0.1 255.255.255.255
 identity local address 192.168.0.2
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint TP1
 pki trustpoint TP2
```

En este ejemplo, R1 es el respondedor IKEv2:

```
crypto ikev2 profile profile1
 match identity remote address 192.168.0.2 255.255.255.255
 identity local address 192.168.0.1
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint TP1
```

Aquí, R2 envía el primer paquete sin ninguna solicitud de certificado. El respondedor responde con una solicitud de certificado para todos los puntos de confianza configurados. El orden de las cargas útiles es similar al IKEv1 y depende de los certificados instalados:

```
R1#show crypto pki certificates
Certificate
 Status: Available
 Certificate Serial Number (hex): 04
 Certificate Usage: General Purpose
 Issuer:
   cn=CA2
....
 Associated Trustpoints: TP2
```

El primer certificado configurado en R1 está asociado con el punto de confianza **TP2**, por lo que la primera carga útil de solicitud de certificado es para la CA que está asociada con el punto de confianza **TP2**. Por lo tanto, R2 lo selecciona para la autenticación (regla de primera coincidencia):

```
R2#
*Jul 17 18:09:04.542: IKEv2:(SA ID = 1):Processing IKE_SA_INIT message
*Jul 17 18:09:04.542: IKEv2:(SA ID = 1):[IKEv2 -> PKI] Retrieving trustpoint(s)
 from received certificate hash(es)
```

```
*Jul 17 18:09:04.542: IKEv2:(SA ID = 1):[PKI -> IKEv2] Retrieved
 trustpoint(s): 'TP2'
*Jul 17 18:09:04.542: IKEv2:(SA ID = 1):[IKEv2 -> PKI] Getting cert chain for
 the trustpoint TP2
*Jul 17 18:09:04.542: IKEv2:(SA ID = 1):[PKI -> IKEv2] Getting of cert chain
 for the trustpoint PASSED
```

Luego, R2 prepara una respuesta (paquete 3) con la carga útil de solicitud de certificación asociada con **TP2**. R1 no puede confiar en el certificado ya que está configurado para validación con el punto de confianza **TP1**:

```
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):[IKEv2 -> PKI] Retrieving trustpoint(s)
 from received certificate hash(es)
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):[PKI -> IKEv2] Retrieved
 trustpoint(s): 'TP1'
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):[IKEv2 -> PKI] Getting cert chain for
 the trustpoint TP1
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):[PKI -> IKEv2] Getting of cert chain
 for the trustpoint PASSED
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):Get peer's authentication method
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):Peer's authentication method is 'RSA'
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):[IKEv2 -> PKI] Validating
 certificate chain
*Jul 17 18:09:04.554: IKEv2:(SA ID = 1):[PKI -> IKEv2] Validation of certificate
 chain FAILED
*Jul 17 18:09:04.554: IKEv2:(SA ID = 1):Verification of peer's authentication
 data FAILED
*Jul 17 18:09:04.554: IKEv2:(SA ID = 1):Sending authentication failure notify
*Jul 17 18:09:04.554: IKEv2:(SA ID = 1):Building packet for encryption.
Payload contents:
 NOTIFY(AUTHENTICATION_FAILED)
```

Como se mencionó anteriormente, Cisco recomienda que no utilice varios puntos de confianza bajo un perfil IKEv2. Cuando utiliza varios puntos de confianza, es necesario asegurarse de que ambos lados confían exactamente en los mismos puntos de confianza. Por ejemplo, tanto R1 como R2 tienen tanto TP1 como TP2 configurados en sus perfiles.

Summary

Esta sección proporciona un breve resumen de la información que se describe en el documento.

El contenido de la carga útil de la solicitud de certificado depende de la configuración. Si se configura un punto de confianza específico para el perfil ISAKMP y el router es el iniciador ISAKMP, entonces la solicitud de certificado en el MM3 contiene solamente la CA asociada con el punto de confianza. Sin embargo, si el mismo router es el respondedor ISAKMP, el paquete MM4 que envía el router incluye varias cargas útiles de solicitud de certificado para todos los puntos de confianza definidos globalmente (cuando el comando **ca trust-point** no se toma en consideración). Esto ocurre porque el respondedor ISAKMP puede determinar el perfil ISAKMP que se debe utilizar sólo después de recibir el MM5 y la solicitud de certificado que se incluye en el MM4.

La carga útil de solicitud de certificado en el MM3 y el MM4 es importante debido a la primera regla de coincidencia. La primera regla de coincidencia determina el punto de confianza que se

utiliza para la selección del certificado, que es necesario para la autenticación en el MM5 y el MM6.

El orden de carga útil de la solicitud de certificado depende del orden de los certificados instalados. El emisor del primer certificado que aparece en la salida del comando **show crypto pki certificate** se envía primero. Este primer certificado es el último que está inscrito.

Es posible configurar varios puntos de confianza para un perfil ISAKMP. Si esto se realiza, se seguirán aplicando todas las reglas anteriores.

Todos los problemas y advertencias que se describen en este documento se deben al diseño del protocolo IKEv1. La fase de autenticación se produce en el MM5 y el MM6, mientras que las propuestas de autenticación (solicitudes de certificado) deben enviarse en una fase anterior (inicial) sin conocer el perfil ISAKMP que debe utilizarse. Este no es un problema específico de Cisco y está relacionado con las limitaciones del diseño del protocolo IKEv1.

El protocolo IKEv2 es similar al IKEv1 con respecto al proceso de negociación de certificados. Sin embargo, la implementación en el IOS fuerza el uso de puntos de confianza específicos para el iniciador. Esto no resuelve todos los problemas. Cuando se configuran varios puntos de confianza para un solo perfil y se configura un único punto de confianza en el otro lado, todavía es posible encontrar problemas con la autenticación. Cisco recomienda utilizar configuraciones simétricas de punto de confianza para ambos lados de la conexión (los mismos puntos de confianza configurados para ambos perfiles IKEv2).

A continuación se muestran algunas notas importantes sobre la información que se describe en este documento:

- Con configuraciones de punto de confianza asimétricas para los perfiles IKEv1 de peers, el túnel podría iniciarse desde sólo un lado del túnel. La configuración del punto de confianza para el perfil IKEv1 es opcional.

- Con configuraciones de punto de confianza asimétricas para los perfiles IKEv2 de peers, el túnel podría iniciarse desde sólo un lado del túnel. La configuración del punto de confianza para el perfil IKEv2 es obligatoria para el iniciador.

- El orden de carga útil de solicitud de certificado depende del orden de los certificados que aparecen en el resultado del comando **show crypto pki certificate** (primera coincidencia).

- El orden de carga útil de solicitud de certificado determina el certificado seleccionado por el respondedor (primera coincidencia).

- Cuando utiliza varios perfiles para IKEv1 y IKEv2 y tiene configuradas las mismas reglas de identidad de coincidencia, es difícil predecir los resultados (hay demasiados factores involucrados).

- Cisco recomienda utilizar configuraciones de punto de confianza simétricas tanto para IKEv1 como para IKEv2.

Información Relacionada

- Guía de Configuración de Intercambio de Claves de Internet para VPNs IPSec, Cisco IOS Release 15M&T - Certificate to ISAKMP Profile Mapping
- Referencia de Comandos de Seguridad de Cisco IOS: Comandos A a C - ca trust-point a través de clear eou
- Soporte Técnico y Documentación - Cisco Systems