Processos de troca de pacotes IOS IKEv1 e IKEv2 para perfis com vários certificados

Contents

Introduction

Este documento descreve os processos de troca de pacotes IKEv1 (Internet Key Exchange Version 1) e IKEv2 (Internet Key Exchange Version 2) quando a autenticação de certificado é usada e os possíveis problemas que podem ocorrer.

Aqui está uma lista de assuntos descritos neste documento:

- Os critérios de seleção de certificado para o iniciador do Internet Key Exchange (IKE) e o respondedor IKE
- Os critérios de correspondência do perfil IKE quando vários perfis IKE são correspondidos (para cenários de sobreposição e não sobreposição)
- As configurações e o comportamento padrão quando nenhum ponto de confiança é usado nos perfis IKE
- As diferenças entre o IKEv1 e o IKEv2 no que diz respeito aos critérios de seleção de perfil e certificado

Note: Para obter detalhes sobre como solucionar um problema específico, consulte a seção correta. Além disso, um breve resumo é fornecido no final deste documento.

Prerequisites

Requirements

A Cisco recomenda que você tenha conhecimento destes tópicos:

- Configuração de VPN do Cisco IOS®
- Protocolos IKEv1 e IKEv2 (troca de pacotes)

Componentes Utilizados

As informações neste documento são baseadas na versão 15.3T do Cisco IOS.

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.

Informações de Apoio

Os problemas descritos neste documento surgem quando vários pontos de confiança e vários perfis IKE são usados.

Os exemplos iniciais usados neste documento têm um túnel de LAN para LAN de IKEv1 com dois pontos de confiança em cada roteador. No início, pode parecer que a configuração está correta. No entanto, o túnel VPN pode ser iniciado somente de um lado da conexão devido à forma como o comando **ca trust-point** é usado para o comportamento do perfil ISAKMP (Internet Security Association and Key Management Protocol) e para a ordem dos certificados inscritos no armazenamento local.

Um comportamento diferente é configurado com o comando **ca trust-point** para o perfil ISAKMP quando o roteador é o iniciador ISAKMP. Um problema pode ocorrer porque o iniciador de ISAKMP está ciente do perfil de ISAKMP desde o início, de modo que o comando **ca trust-point** configurado para o perfil pode influenciar o payload para a solicitação de certificado no Main Mode Packet 3 (MM3). No entanto, quando o roteador é o respondente ISAKMP, ele vincula o tráfego de entrada a um perfil ISAKMP específico depois de receber o Main Mode Packet 5 (MM5), que inclui o ID IKE necessário para criar a associação. É por isso que não é possível aplicar qualquer comando **ca trust-point** para o pacote do Main Mode 4 (MM4) porque o perfil não

é determinado antes do MM5.

A ordem da carga do pedido de certificado no MM3 e no MM4 e o impacto em todo o processo de negociação são explicados neste documento, bem como o motivo pelo qual ele permite apenas que a conexão seja estabelecida de um lado do túnel VPN.

Aqui está um resumo dos comportamentos do iniciador e do respondedor IKEv1:

	Iniciador IKEv1	Respondedor IKEv1
Enviar solicitação	Envia solicitações específicas apenas para os pontos de confiança configurados no perfil	Envia solicitações para todos os pontos confiáveis disponíveis
Validar solicitação	Valida contra pontos de confiança específicos configurados no perfil	Valida contra pontos de confiança específicos configurados no perfil

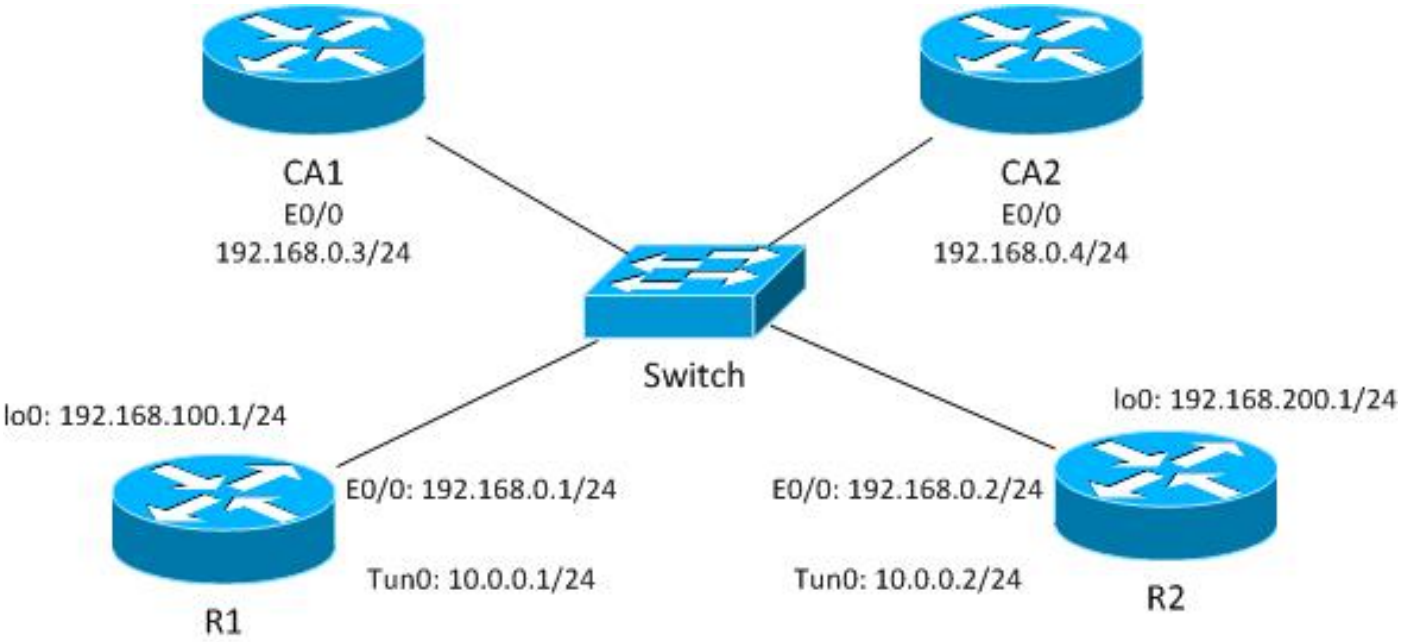
A Cisco recomenda que você não use o comando **ca trust-point** para os respondentes ISAKMP que têm vários perfis ISAKMP e usam pontos de confiança configurados globalmente. Para iniciadores ISAKMP com vários perfis ISAKMP, a Cisco recomenda que você restrinja o processo de seleção de certificado com o comando **ca trust-point** em cada perfil.

O protocolo IKEv2 tem os mesmos problemas que o protocolo IKEv1, mas o comportamento diferente do comando **pki trustpoint** ajuda a evitar a ocorrência dos problemas. Isso ocorre porque o comando **pki trustpoint** é obrigatório para o iniciador IKEv2, enquanto o comando **ca trust-point** é opcional para o iniciador IKEv1. Em algumas circunstâncias (vários pontos confiáveis sob um perfil), os problemas descritos anteriormente podem ocorrer. Por esse motivo, a Cisco recomenda que você use configurações simétricas de ponto de confiança para ambos os lados da conexão (os mesmos pontos de confiança configurados nos dois perfis IKEv2).

Topologia

Esta é uma topologia genérica usada para todos os exemplos neste documento.

> **Note**: O roteador 1 (R1) e o roteador 2 (R2) usam VTIs (Virtual Tunnel Interfaces) para acessar os loopbacks. Esses VTIs são protegidos por IPSec.



Para este exemplo de IKEv1, cada roteador tem dois pontos de confiança para cada autoridade de certificação (CA), e os certificados para cada um dos pontos de confiança são inscritos.

Quando R1 é o iniciador ISAKMP, o túnel negocia corretamente e o tráfego é protegido. Este é um comportamento esperado. Quando R2 é o iniciador ISAKMP, a negociação Phase1 falha.

> **Note**: Para os exemplos de IKEv2 neste documento, a topologia e o endereçamento são os mesmos mostrados no exemplo de IKEv1.

Processo de troca de pacotes

Esta seção descreve as variações de configuração de IKEv1 e IKEv2 que são usadas para o processo de troca de pacotes e os possíveis problemas que podem surgir.

IKEv1 com vários certificados

Aqui estão a rede R1 e a configuração de VPN para IKEv1 com vários 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
```

Aqui estão a rede R2 e a configuração de VPN para IKEv1 com vários 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
```

Neste exemplo, R1 tem dois pontos de confiança: um usa **IOSCA1** e o segundo usa **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
```

Neste exemplo, R2 também tem dois pontos de confiança: um usa **IOSCA1** e o segundo usa **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
```

Éimportante observar a única diferença nessas configurações: o perfil ISAKMP de R1 usa o comando **ca trust-point** para o ponto de confiança **IOSCA1**, que indica que R1 confia somente nos certificados validados por esse ponto de confiança específico. Em contrapartida, o R2 confia em todos os certificados validados por todos os pontos de confiança definidos globalmente.

R1 como o iniciador IKEv1

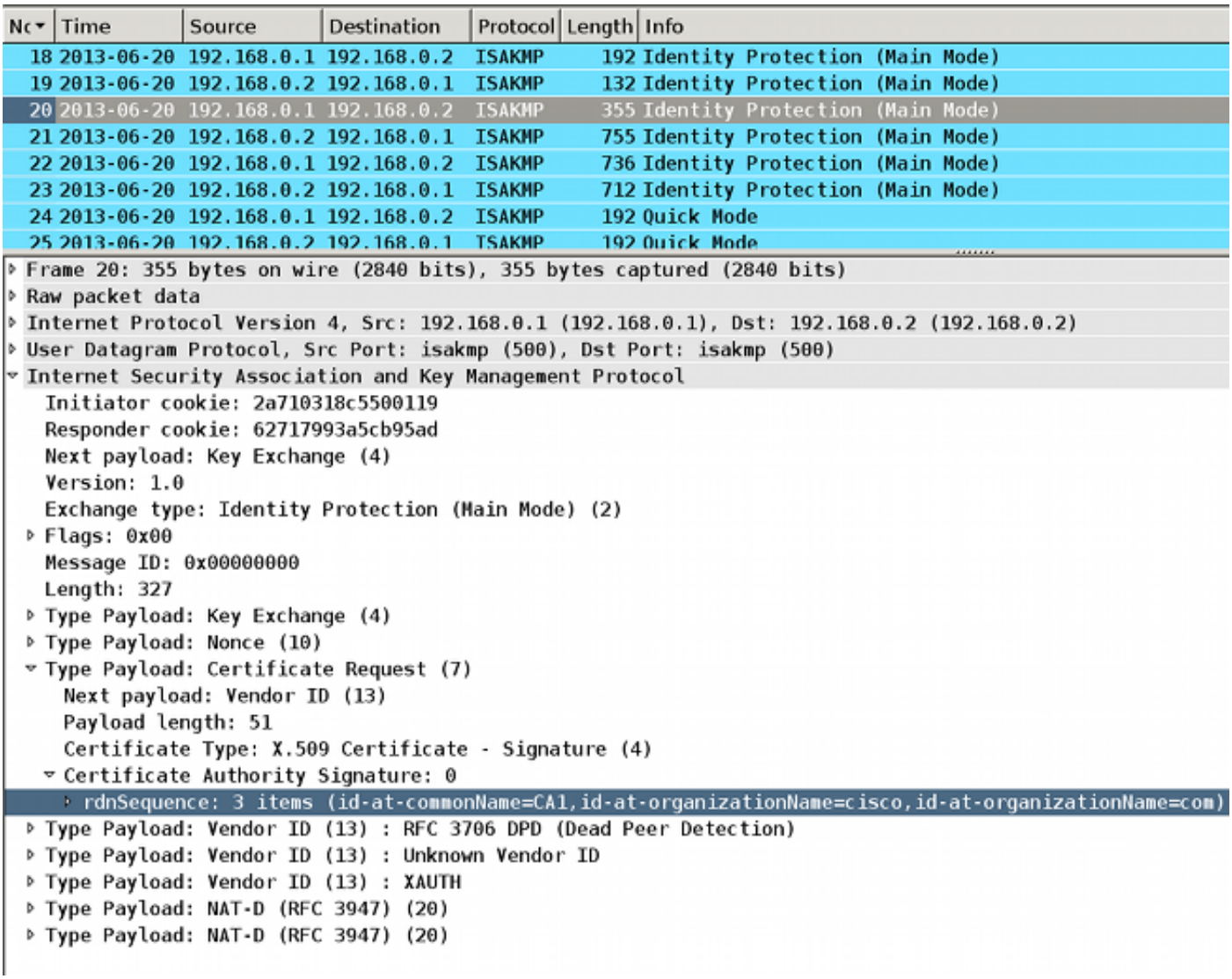
Aqui estão os comandos debugs para R1 e R2:

- **R1#debug crypto isakmp**
- **R1#debug crypto ipsec**
- **R1#debug crypto pki validation**

Aqui, R1 inicia o túnel e envia a solicitação de certificado no 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
```

Éimportante observar que o pacote contém apenas uma solicitação de certificado, que é somente para o ponto de confiança **IOSCA1**. Este é um comportamento esperado com a configuração atual do perfil ISAKMP (**CN=CA1, O=cisco, O=com**). Nenhuma outra solicitação de certificado é enviada, que você pode verificar com o recurso Captura de pacote incorporado:



Quando o R2 recebe o pacote, ele começa a processar a solicitação de certificado, o que cria uma correspondência que determina o ponto de confiança e o certificado associado que é usado para autenticação no MM5. A ordem do processo é a mesma do payload de solicitação de certificado no pacote ISAKMP. Isso significa que a primeira correspondência é usada. Neste cenário, há apenas uma correspondência, pois R1 está configurado com um ponto de confiança específico e envia apenas uma solicitação de certificado associada ao ponto de confiança.

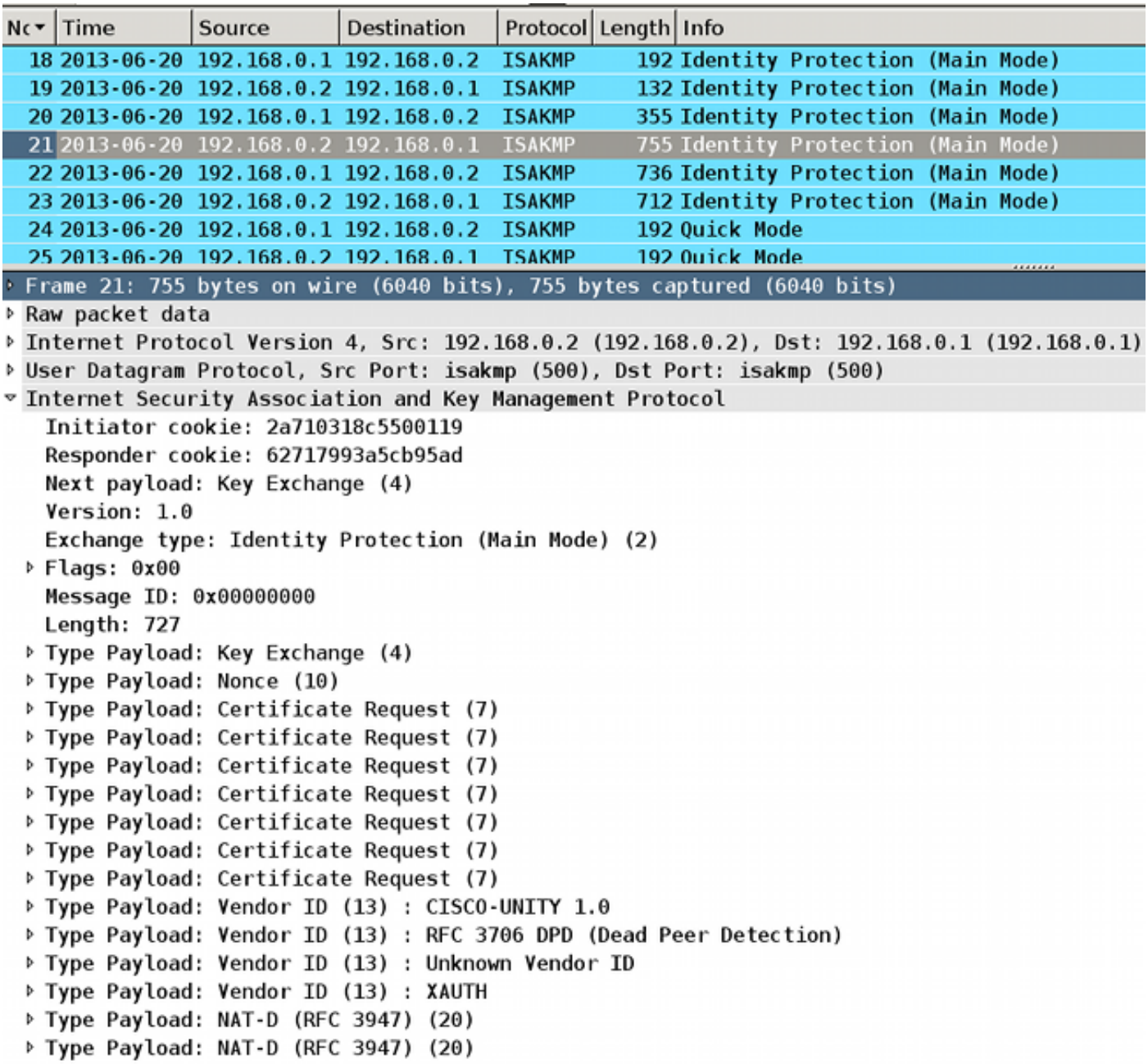
```
*Jun 20 13:00:37.617: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
```

```
*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
```

Depois, R2 prepara o MM4. Este é o pacote que contém a solicitação de certificado para todos os pontos confiáveis. Como R2 é o respondedor ISAKMP, todos os pontos de confiança definidos globalmente são confiáveis (a configuração **ca trust-point** não está marcada). Dois dos pontos de confiança são definidos manualmente (**IOSCA1** e **IOSCA2**) e os restantes são predefinidos.

```
*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
```

Você pode verificar o pacote com o Wireshark. O pacote MM4 de R2 contém sete entradas de solicitação 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

Em seguida, R1 recebe o MM4 de R2 com vários campos de solicitação 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
```

A regra de primeira correspondência em R1 corresponde à primeira solicitação de certificado com o ponto de confiança **IOSCA1**. Isso determina que R1 usa o certificado associado ao ponto de confiança **IOSCA1** para autenticação no MM5. O nome de domínio totalmente qualificado (FQDN) é usado como ID de IKE. Isso se deve à configuração **fqdn de autoidentidade** no 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
```

O MM5 é recebido e processado por R2. A ID IKE recebida (**R1.cisco.com**) corresponde ao perfil ISAKMP **prof1**. O certificado recebido é validado e a autenticação é bem-sucedida:

```
*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
```

Em seguida, R2 prepara o MM6 com o certificado associado ao **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
```

O pacote é recebido por R1 e R1 verifica o certificado e a autenticação:

```
*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
```

Isso conclui a Fase 1. A fase 2 é negociada como de costume. O túnel é estabelecido com êxito e o tráfego é protegido.

R2 como o iniciador IKEv1

Este exemplo descreve o processo quando R2 inicia o mesmo túnel IKEv1 e explica por que ele não está estabelecido.

> **Note**: Partes dos registros são removidas para focalizar somente as diferenças em relação ao exemplo apresentado na seção anterior.

R2 envia o MM3 com sete payloads de solicitação de certificado porque R2 não tem um ponto de confiança associado ao perfil ISAKMP (todos os pontos de confiança são confiáveis):

```
*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
```

Quando R1 recebe o pacote de R2, processa a solicitação de certificado e corresponde ao ponto confiável IOSCA1, que determina o certificado enviado no MM6:

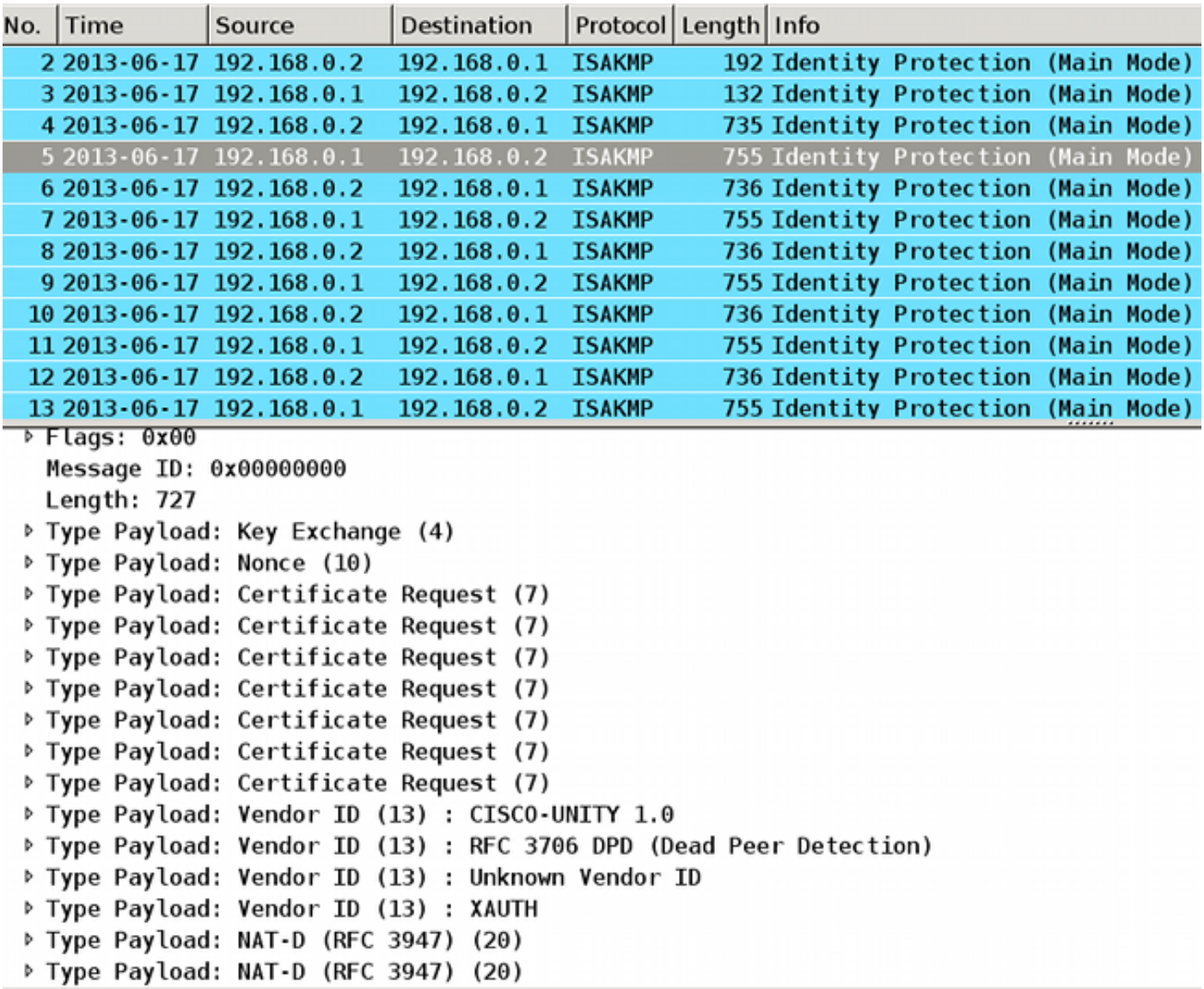
```
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=CA1,o=cisco,o=com
*Jun 17 18:08:14.321:  Choosing trustpoint IOSCA1 as issuer
*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
```

Depois, R1 prepara o pacote MM4 com o payload de solicitação de certificado. Agora há várias cargas úteis de solicitação de certificado:

```
*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 os registros com Embedded Packet Capture (EPC) e 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)

Embora R1 esteja configurado para um único ponto de confiança (**IOSCA1**) no perfil ISAKMP, há várias solicitações de certificado enviadas. Isso ocorre porque o comando **ca trust-point** no perfil ISAKMP determina o payload da solicitação de certificado, mas somente quando o roteador é o iniciador da sessão ISAKMP. Se o roteador for o respondente, haverá várias cargas úteis de solicitação de certificado para todos os pontos de confiança definidos globalmente porque R1 ainda não conhece o perfil ISAKMP usado para a sessão IKE.

A sessão IKE de entrada está vinculada a um perfil ISAKMP específico após a recepção do MM5, que inclui o ID IKE. Depois, o comando **match identity** para o perfil específico vincula a sessão

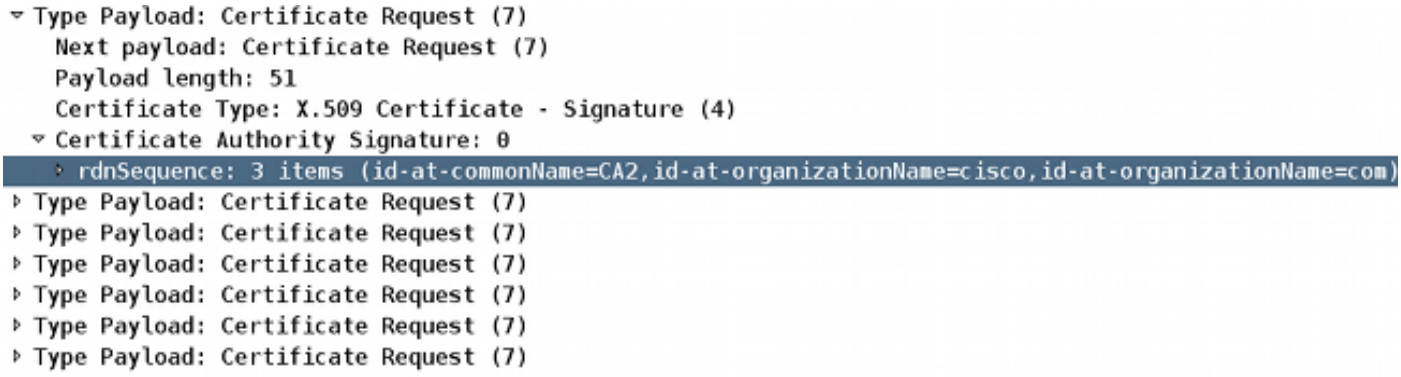
IKE ao perfil. No entanto, o roteador não pode determinar isso até agora. Pode haver vários perfis ISAKMP com diferentes comandos **ca trust-point** configurados para cada perfil.

Por esse motivo, R1 deve enviar a solicitação de certificado para todos os pontos de confiança configurados globalmente.

Consulte a referência de comando para o comando **ca trust-point**:

Um roteador iniciando o IKE e um roteador que responda à solicitação IKE devem ter configurações de ponto de confiança simétricas. Por exemplo, um roteador respondente (no modo principal IKE) que executa a criptografia de assinatura RSA e a autenticação pode usar pontos de confiança definidos na configuração global ao enviar as cargas úteis CERT-REQ. No entanto, o roteador pode usar uma lista restrita de pontos confiáveis que foram definidos no perfil ISAKMP para a verificação do certificado. Se o peer (o iniciador IKE) estiver configurado para usar um certificado cujo ponto de confiança está na lista global do roteador que responde, mas não no perfil ISAKMP do roteador que responde, o certificado será rejeitado. (No entanto, se o roteador iniciador não souber sobre os pontos de confiança na configuração global do roteador respondente, o certificado ainda poderá ser autenticado.)

Agora, verifique os detalhes do pacote MM4 para descobrir o primeiro payload de solicitação de certificado:



O pacote MM4 enviado de R1 inclui o ponto de confiança **IOSCA2** no primeiro payload de solicitação de certificado devido à ordem em que os certificados são instalados; o primeiro é assinado pelo ponto de confiança **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>
```

Faça uma comparação com o pacote MM3 que é enviado de R2 quando o ponto confiável IOSCA1 é incluído no primeiro payload de solicitação 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>
```

Agora R2 recebe o pacote MM4 de R1 e começa a processar a solicitação de certificado. O primeiro payload de solicitação de certificado corresponde ao ponto de confiança 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
```

Quando R2 prepara o pacote MM5, usa o certificado associado ao ponto de confiança 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.
```

O pacote MM5 é recebido por R1. Como R1 confia somente no ponto confiável IOSCA1 (para o perfil ISAKMP prof1), a validação do certificado falha:

```
*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
```

Essa configuração funciona se a ordem da inscrição de certificado em R1 for diferente porque o primeiro certificado exibido é assinado pelo ponto de confiança **IOSCA1**. Além disso, o primeiro payload de solicitação de certificado no MM4 é o ponto confiável **IOSCA1**, que é então escolhido por R2 e validado com êxito em R1 no MM6.

IKEv1 sem um comando *ca trust-point* no perfil

Para cenários com vários perfis e pontos de confiança, mas sem uma configuração de ponto de confiança específica nos perfis, não há problemas porque não há validação de pontos de confiança específicos determinados por uma configuração de comando **ca trust-point**. No entanto, o processo de seleção pode não ser óbvio. Dependendo do roteador que é o iniciador, os diferentes certificados são selecionados para o processo de autenticação em relação à ordem de inscrição de certificado.

Às vezes, um certificado pode ser suportado por apenas um lado da conexão, como na versão 1 do x509, que não é uma função hash típica usada para assinar. O túnel VPN pode ser estabelecido somente de um lado da conexão.

Referência de RFC para IKEv1

Aqui está um trecho do RFC4945:

3.2.7.1. Especificando autoridades de certificação

Ao **solicitar** troca in-band de materiais essenciais, as implementações DEVEM gerar CERTREQs para cada âncora de confiança de peer que a **política local explicitamente** considere confiável durante uma determinada troca.
O RFC não está claro. A **política local explicitamente** pode se relacionar ao comando **ca trust-point** configurado no perfil ISAKMP de criptografia. O problema é que, nos estágios MM3 e MM4 do processo, você não pode selecionar um perfil ISAKMP a menos que use um endereço IP para a identidade e os pontos de confiança porque a autenticação nos estágios MM5 e MM6 do processo deve ocorrer primeiro. Por esse motivo, a **política local** se relaciona **explicitamente** a todos os pontos de confiança configurados no dispositivo.

Note: Essas informações não são específicas da Cisco, mas são específicas do IKEv1.

Seleção de perfil IKEv2 com identidades que se sobrepõem

Antes que vários certificados para IKEv2 sejam descritos, é importante saber como os perfis são selecionados quando a identidade de correspondência é usada, o que é satisfeito para todos os perfis. Este não é um cenário recomendado porque os resultados da negociação de IKEv2 dependem de vários fatores. Os mesmos problemas existem para IKEv1 quando os perfis que se sobrepõem são usados.

Aqui está um exemplo de configuração do iniciador 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
```

O endereço do tipo de identidade é usado para ambos os lados da conexão. A autenticação via certificados (também pode ser chaves pré-compartilhadas) não é importante para este exemplo. O respondente tem vários perfis que correspondem ao tráfego IKEv2 de entrada:

```
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
```

O iniciador envia o terceiro pacote IKEv2 e o respondente deve escolher o perfil com base na identidade recebida. A identidade é um endereço 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 os perfis satisfazem esta identidade devido ao comando **match identity** configurado. O IOS escolhe o último na configuração, que é **profile3** neste exemplo:

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

Para verificar o pedido, insira o comando **show crypto ikev2 profile**.

> **Note**: Mesmo quando há um endereço genérico (0.0.0.0) no perfil, ele ainda é selecionado. O IOS não tenta encontrar a melhor correspondência; ele tenta encontrar a primeira correspondência. No entanto, isso só ocorre porque todos os perfis têm o mesmo comando **match identity remote** configurado. Para os perfis IKEv1 e IKEv2 que têm regras de identidade de correspondência diferentes, o mais específico é sempre usado. A Cisco recomenda que você não tenha os perfis configurados com o comando **overlapping match identity** porque é difícil prever o perfil selecionado.

Neste cenário, **profile3** é selecionado pelo respondente, mas **profile1** é usado para a interface do túnel. Isso faz com que um erro apareça quando a ID do proxy é negociada:

```
*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
```

Fluxo IKEv2 quando os certificados são usados

Quando os certificados são usados para IKEv2 a fim de autenticar, o iniciador não envia o

payload de solicitação de certificado no primeiro pacote:

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

O respondente responde com o payload de solicitação de certificado (segundo pacote) e todas as CAs porque o respondente não tem conhecimento do perfil que deve ser usado neste estágio. O pacote que contém as informações é enviado ao 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)
```

O iniciador processa o pacote e escolhe um ponto de confiança que corresponda à CA proposta:

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

Em seguida, o iniciador envia o terceiro pacote com a solicitação de certificado e o payload de certificado. Este pacote já está criptografado com material de chave da 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)
```

O quarto pacote é enviado do respondente para o iniciador e contém apenas o payload do 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)
```

O fluxo descrito aqui é semelhante ao fluxo IKEv1. O respondente deve enviar o payload de solicitação de certificado antecipadamente sem saber o perfil que deve ser usado, o que cria os mesmos problemas que foram descritos anteriormente para IKEv1 (de uma perspectiva de protocolo). No entanto, a implementação no IOS é melhor para o IKEv2 do que para o IKEv1.

Ponto de confiança obrigatório IKEv2 para o iniciador

Aqui está um exemplo de quando um iniciador IKEv2 tenta usar um perfil com autenticação de certificado e não tem nenhum ponto de confiança configurado nesse 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
```

O primeiro pacote é enviado sem nenhum payload de solicitação de certificado, conforme descrito anteriormente. A resposta do respondente inclui o payload de solicitação de certificado para todos os pontos de confiança definidos no modo de Configuração Global. Isso é recebido pelo 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
```

O iniciador não sabe o ponto de confiança que deve ser usado para assinar. Essa é a principal diferença quando a implementação IKEv2 é comparada ao IKEv1. O iniciador IKEv2 deve ter o ponto de confiança configurado no perfil do iniciador IKEv2, mas não é necessário para o respondente IKEv2.

Aqui está um trecho da referência do comando:

Se não houver um ponto de confiança definido na configuração do perfil IKEv2, o padrão é **validar o certificado** usando todos os pontos de confiança definidos na configuração global

Épossível definir diferentes pontos de confiança; um para assinar e outro para validar. Infelizmente, o ponto de confiança obrigatório configurado no perfil IKEv2 não resolve todos os problemas.

R2 como o iniciador IKEv2

Neste exemplo, R2 é o iniciador de 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
```

Neste exemplo, R1 é o 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
```

Aqui, R2 envia o primeiro pacote sem qualquer solicitação de certificado. O respondente responde com uma solicitação de certificado para todos os pontos de confiança configurados. A ordem dos payloads é semelhante ao IKEv1 e depende dos 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
```

O primeiro certificado configurado em R1 está associado ao ponto de confiança **TP2**, portanto, o primeiro payload de solicitação de certificado é para a CA associada ao ponto de confiança **TP2**. Assim, R2 o seleciona para autenticação (regra de primeira correspondência):

```
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
```

Em seguida, R2 prepara uma resposta (pacote 3) com o payload de solicitação de certificação associado ao **TP2**. R1 não pode confiar no certificado porque ele está configurado para validação em relação ao ponto de confiança **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 mencionado anteriormente, a Cisco recomenda que você não use vários pontos confiáveis em um perfil IKEv2. Ao usar vários pontos de confiança, é necessário garantir que ambos os lados confiem exatamente nos mesmos pontos de confiança. Por exemplo, R1 e R2 têm TP1 e TP2 configurados em seus perfis.

Summary

Esta seção fornece um breve resumo das informações descritas no documento.

O conteúdo do payload da solicitação de certificado depende da configuração. Se um ponto de confiança específico for configurado para o perfil ISAKMP e o roteador for o iniciador ISAKMP, a solicitação de certificado no MM3 conterá apenas a CA associada ao ponto de confiança. No entanto, se o mesmo roteador for o respondente ISAKMP, o pacote MM4 que é enviado pelo roteador inclui várias cargas úteis de solicitação de certificado para todos os pontos confiáveis definidos globalmente (quando o comando **ca trust-point** não é considerado). Isso ocorre porque o respondente ISAKMP pode determinar o perfil ISAKMP que deve ser usado somente depois de receber o MM5 e a solicitação de certificado incluída no MM4.

O payload de solicitação de certificado no MM3 e no MM4 é importante devido à primeira regra de correspondência. A regra de primeira correspondência determina o ponto de confiança usado para a seleção do certificado, que é necessário para autenticação no MM5 e no MM6.

A ordem do payload de solicitação de certificado depende da ordem dos certificados instalados. O emissor do primeiro certificado que aparece na saída do comando **show crypto pki certificate** é enviado primeiro. Este primeiro certificado é o último que está inscrito.

Épossível configurar vários pontos de confiança para um perfil ISAKMP. Se isso for executado, todas as regras anteriores ainda serão aplicadas.

Todos os problemas e advertências descritos neste documento se devem ao projeto do protocolo IKEv1. A etapa de autenticação ocorre em MM5 e MM6, enquanto as propostas de autenticação (pedidos de certificado) devem ser enviadas em uma fase anterior (frente) sem conhecimento do perfil ISAKMP que deve ser usado. Esse não é um problema específico da Cisco e está relacionado às limitações do projeto do protocolo IKEv1.

O protocolo IKEv2 é semelhante ao IKEv1 em relação ao processo de negociação de certificado. No entanto, a implementação no IOS força o uso de pontos de confiança específicos para o iniciador. Isso não resolve todos os problemas. Quando vários pontos de confiança são configurados para um único perfil e um único ponto de confiança é configurado do outro lado, ainda é possível encontrar problemas com a autenticação. A Cisco recomenda que você use configurações de ponto de confiança simétrico para ambos os lados da conexão (os mesmos pontos de confiança configurados para ambos os perfis IKEv2).

Aqui estão algumas notas importantes sobre as informações descritas neste documento:

- Com configurações de ponto confiável assimétrico para os perfis IKEv1 de peers, o túnel pode iniciar de apenas um lado do túnel. A configuração do ponto de confiança para o perfil IKEv1 é opcional.

- Com configurações de ponto confiável assimétrico para os perfis IKEv2 de peers, o túnel pode iniciar de apenas um lado do túnel. A configuração do ponto de confiança para o perfil IKEv2 é obrigatória para o iniciador.

- A ordem de payload da solicitação de certificado depende da ordem dos certificados que aparecem na saída do comando **show crypto pki certificate** (primeira correspondência).

- A ordem de payload da solicitação de certificado determina o certificado selecionado pelo respondente (primeira correspondência).

- Quando você usa vários perfis para o IKEv1 e o IKEv2 e tem as mesmas regras de identidade de correspondência configuradas, é difícil prever os resultados (muitos fatores envolvidos).

- A Cisco recomenda que você use configurações de ponto de confiança simétrico para IKEv1 e IKEv2.

Informações Relacionadas

- Guia de Configuração do Internet Key Exchange para VPNs IPsec, Cisco IOS versão 15M&T - Mapeamento de Perfil de Certificado para ISAKMP
- Referência do comando de segurança do Cisco IOS: Comandos A a C - ca trust-point através de clear eou
- Suporte Técnico e Documentação - Cisco Systems