# Processos do intercâmbio de pacotes IKEv1 e IKEv2 IO para perfis com certificados múltiplos

## Índice



## Introdução

Este documento descreve a versão 1 do intercâmbio de chave de Internet (IKEv1) e processos do intercâmbio de pacotes da versão 2 do intercâmbio de chave de Internet (IKEv2) quando o certificado de autenticação é usado e os problemas possíveis que puderam ocorrer.

Está aqui uma lista de assuntos que são descritos neste documento:

- Os critérios de seleção do certificado para o iniciador do Internet Key Exchange (IKE) e o que responde IKE
- Os critérios de verificação de repetição de dados do perfil IKE quando os perfis múltiplos IKE forem combinados (para encenações da sobreposição e da NON-sobreposição)
- As configurações padrão e o comportamento quando nenhum confiança-ponto for usado sob os perfis IKE
- As diferenças entre o IKEv1 e o IKEv2 com respeito aos critérios de seleção do perfil e do certificado

Nota: Para detalhes sobre como pesquisar defeitos um problema específico, refira a seção correta. Também, um sumário sucinto é fornecido na extremidade deste documento.

# Pré-requisitos

## Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

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

## Componentes Utilizados

A informação neste documento é baseada no Cisco IOS Version15.3T.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

# Informações de Apoio

Os problemas que são descritos neste documento elevaram quando os confiança-pontos múltiplos e os perfis múltiplos IKE são usados.

Os exemplos iniciais que são usados neste documento têm um túnel de LAN para LAN IKEv1 com dois confiança-pontos em cada roteador. No início, pôde-se parecer que a configuração está correta. Contudo, o túnel VPN pode ser iniciado somente de um lado da conexão devido à maneira que o comando do confiança-**ponto Ca** é usado para o comportamento do perfil do Internet Security Association and Key Management Protocol (ISAKMP) e para a ordem dos Certificados registrados na loja local.

Um comportamento diferente está configurado com o comando do confiança-**ponto Ca** para o perfil ISAKMP quando o roteador é o iniciador ISAKMP. Um problema pôde ocorrer porque o iniciador ISAKMP está ciente do perfil ISAKMP desde o início, assim o comando do confiança-**ponto Ca** que é configurado para o perfil pode influenciar o payload para o pedido do certificado no pacote 3 do modo principal (MM3). Contudo, quando o roteador é o que responde ISAKMP, liga o tráfego de entrada a um perfil específico ISAKMP depois que recebe o pacote 5 do modo principal (MM5), que inclui o IKE ID que é necessário a fim criar o ligamento. Eis porque não é

possível aplicar nenhum comando do confiança-**ponto Ca** para o pacote do pacote 4 do modo principal (MM4) porque o perfil não é determinado antes do MM5.

A ordem do requestpayload do certificado no MM3 e no MM4 e do processo de negociação do impacto é explicada em geral neste documento, assim como na razão que permite somente que a conexão esteja estabelecida de um lado do túnel VPN.

Está aqui um sumário dos comportamentos do iniciador IKEv1 e do que responde:

| | **Iniciador IKEv1** | **Que responde IKEv1** |
|---|---|---|
| **Envie o pedido** | Envia pedidos específicos somente para os confiança-pontos que são configurados sob o perfil | Envia pedidos para todos os confiança-pontos disponíveis |
| **Valide o pedido** | Valida contra os confiança-pontos específicos que são configurados sob o perfil | Valida contra os confiança-pontos específicos que são configurados sob o perfil |

Cisco recomenda que você não usa o comando do confiança-**ponto Ca** para os que respondes ISAKMP que têm perfis múltiplos ISAKMP e usam confiança-pontos globalmente-configurados. Para iniciadores ISAKMP com perfis múltiplos ISAKMP, Cisco recomenda que você reduz o processo de seleção do certificado com o comando do confiança-**ponto Ca em** cada perfil.

O protocolo IKEv2 tem as mesmas edições que o protocolo IKEv1, mas o comportamento diferente das ajudas do comando do **ponto confiável do pki** impede a ocorrência dos problemas. Isto é porque o comando do **ponto confiável do pki** é imperativo para o iniciador IKEv2, quando o comando do confiança-**ponto Ca** for opcional para o iniciador IKEv1. Sob algumas circunstâncias (confiança-pontos múltiplos sob um perfil), os problemas previamente descritos puderam ocorrer. Por este motivo, Cisco recomenda que você usa configurações simétricas do confiança-ponto para ambos os lados da conexão (os mesmos confiança-pontos configurados sob ambos os perfis IKEv2).

# Topologia

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

> Nota: Interfaces de túnel virtuais do uso do roteador1 (r1) e do roteador2 (R2) (VTIs) a fim alcançar os laços de retorno. Este VTIs é protegido pelo IPsec.

Para este exemplo IKEv1, cada roteador tem dois confiança-pontos para cada Certificate Authority (CA), e os Certificados para cada um dos confiança-pontos são registrados.

Quando o r1 é o iniciador ISAKMP, o túnel negocia corretamente e o tráfego é protegido. Este é um comportamento esperado. Quando o R2 é o iniciador ISAKMP, a negociação Phase1 falha.

Nota: Para os exemplos IKEv2 neste documento, a topologia e o endereçamento são a mesma que que mostrado o exemplo IKEv1.

# Processo do intercâmbio de pacotes

Esta seção descreve os IKEv1 e as variações de configuração IKEv2 que são usados para o processo do intercâmbio de pacotes, e os problemas possíveis que puderam elevarar.

## IKEv1 com certificados múltiplos

Estão aqui a rede e a configuração de VPN do r1 para IKEv1 com certificados múltiplos:

```
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
```

Estão aqui a rede R2 e a configuração de VPN para IKEv1 com certificados múltiplos:

```
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
```

Neste exemplo, o r1 tem dois confiança-pontos: um usa IOSCA1 e os segundos usos 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, o R2 igualmente tem dois confiança-pontos: um usa IOSCA1 e os segundos usos 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 notar a única diferença nestas configurações: o perfil do r1 ISAKMP usa o comando do confiança-**ponto Ca** para o confiança-ponto **IOSCA1**, que indica que o r1 confia somente os Certificados que são validados por esse confiança-ponto específico. Ao contrário, o R2 confia todos os Certificados que são validados por todos os confiança-pontos globalmente-definidos.

## R1 como o iniciador IKEv1

Estão aqui os comandos debugs para o r1 e o R2:

- **Isakmp do debug crypto R1#**
- **IPsec do debug crypto R1#**
- **Validação do pki do debug crypto R1#**

Aqui, o r1 inicia o túnel e envia ao requestin do certificado o 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 somente um pedido do certificado, que é somente para o confiança-ponto **IOSCA1**. Este é comportamento esperado com a configuração atual do perfil ISAKMP (**CN=CA1, O=cisco, O=com**). Nenhum outro pedido do certificado é enviado, que você pode verificar com a característica encaixada da captura de pacote de informação:

Quando o R2 recebe o pacote, começa a processar o pedido do certificado, que cria um fósforo que determine o confiança-ponto e o certificado associado que é usado para a autenticação no MM5. A ordem do processo é a mesma que o payload do pedido do certificado no pacote ISAKMP. Isto significa que o primeiro fósforo está usado. Nesta encenação, há somente um fósforo desde que o r1 é configurado com um confiança-ponto específico e envia somente um pedido do certificado que é associado com o confiança-ponto.

```
*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
```

Mais tarde, o R2 prepara o MM4. Este é o pacote que contém o pedido do certificado para todos os confiança-pontos confiados. Desde que o R2 é o que responde ISAKMP, todos os confiança-pontos globalmente-definidos são confiados (a configuração do confiança-**ponto Ca** não é verificada). Dois dos confiança-pontos são definidos manualmente (**IOSCA1** e **IOSCA2**), e o resto é 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
```

Você pode verificar o pacote com Wireshark. O pacote MM4 do R2 contém sete entradas do pedido do certificado:

Então, o r1 recebe o MM4 do R2 com campos do pedido dos certificados múltiplos:

```
*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 do primeiro-fósforo no r1 combina o primeiro pedido do certificado com o confiança-ponto

IOSCA1. Isto determina que o r1 usa o certificado que é associado com o confiança-ponto IOSCA1 para a autenticação no MM5. O nome de domínio totalmente qualificado (FQDN) é usado como o IKE ID. Isto é devido à configuração FQDN da auto-identidade 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 pelo R2. O IKE recebido ID (R1.cisco.com) combina o perfil prof1 ISAKMP. O certificado recebido é validado então 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
```

Então, o R2 prepara o MM6 com o certificado que é associado com o 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 pelo r1, e o 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
```

Isto termina a fase onde 1. fases 2 são negociadas como de costume. O túnel é estabelecido com

sucesso e o tráfego é protegido.

## R2 como o iniciador IKEv1

Este exemplo descreve o processo quando o R2 inicia o mesmo túnel IKEv1 e explica porque não se estabelece.

> Nota: As parcelas dos logs são removidas a fim centrar-se somente sobre as diferenças com relação ao exemplo apresentado na seção anterior.

O R2 envia o MM3 com as sete cargas úteis do pedido do certificado porque o R2 não tem um confiança-ponto associado com o perfil ISAKMP (todos os confiança-pontos são confiados):

```
*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 o r1 recebe o pacote do R2, processa o pedido do certificado e combina o confiança-ponto **IOSCA1**, que determina o certificado que é 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
```

Mais tarde, o r1 prepara o pacote MM4 com o payload do pedido do certificado. Agora há cargas úteis do pedido dos certificados múltiplos:

```
*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 logs com captura de pacote de informação encaixada (EPC) e Wireshark:

Mesmo que o r1 seja configurado para um único confiança-ponto (IOSCA1) no perfil ISAKMP, há uns pedidos dos certificados múltiplos enviados. Isto ocorre porque o comando do confiança-ponto Ca no perfil ISAKMP determina o payload do pedido do certificado, mas somente quando o roteador é o iniciador da sessão ISAKMP. Se o roteador é o que responde, há cargas úteis do pedido dos certificados múltiplos para todos os confiança-pontos globalmente-definidos porque o r1 não conhece ainda o perfil ISAKMP que é usado para a sessão de IKE.

A sessão de IKE de entrada é limitada a um perfil específico ISAKMP após a recepção do MM5, que inclui a identificação IKE mais tarde, o comando da identidade do fósforo para o perfil específico liga a sessão de IKE ao perfil. Contudo, o roteador não pode determinar este até aqui. Pôde haver uns perfis múltiplos ISAKMP com os comandos diferentes do confiança-ponto Ca configurados para cada perfil.

Por este motivo, o r1 deve enviar o pedido do certificado para todos os confiança-pontos globalmente-configurados.

Refira a referência de comandos para o comando do confiança-**ponto Ca**:

**Um roteador que iniciam o IKE e um roteador que responde ao pedido IKE devem ter configurações simétricas do ponto confiável**. Por exemplo, um roteador de resposta (no modo principal IKE) que executa a criptografia e a autenticação da assinatura de RSA pôde usar os pontos confiáveis que foram definidos na configuração global ao enviar as cargas úteis CERT REQ. Contudo, o roteador pôde usar uma lista restrita de pontos confiáveis que foram definidos no perfil ISAKMP para a verificação de certificado. Se o par (iniciador IKE) é configurado para usar um certificado cujo o ponto confiável esteja na lista global do roteador de resposta mas não no perfil ISAKMP do roteador de resposta, o certificado é rejeitado. (Contudo, se o roteador de início não sabe sobre os pontos confiáveis na configuração global do roteador de resposta, o certificado pode ainda ser autenticado.)

Verifique agora os detalhes de pacote MM4 a fim descobrir o primeiro payload do pedido do certificado:

O pacote MM4 que é enviado do r1 inclui o confiança-ponto **IOSCA2 no** primeiro payload do pedido do certificado devido à ordem em que os Certificados são instalados; primeiro é assinado pelo confiança-ponto **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 está enviado do R2 quando o confiança-ponto **IOSCA1** é incluído no primeiro payload do pedido do 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 o R2 recebe o pacote MM4 do r1 e começa a processar o pedido do certificado. O primeiro payload do pedido do certificado combina o confiança-ponto 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 o R2 prepara o pacote MM5, usa o certificado que é associado com o confiança-ponto 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 pelo r1. Porque o r1 confia somente o confiança-ponto IOSCA1 (para o perfil prof1 ISAKMP), a validação certificada 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
```

Esta configuração trabalha se a ordem do certificado de registro no r1 é diferente porque o primeiro certificado indicado está assinado pelo confiança-ponto **IOSCA1**. Também, o primeiro payload do pedido do certificado no MM4 é o confiança-ponto **IOSCA1**, que então é escolhido pelo R2 e validado com sucesso no r1 no MM6.

## IKEv1 sem um comando do confiança-*ponto Ca no* perfil

Para encenações com perfis e confiança-pontos múltiplos mas sem uma configuração específica do confiança-ponto nos perfis, não há nenhuma edição porque não há nenhuma validação dos confiança-pontos específicos determinados por um comando configuration do confiança-**ponto Ca**. Contudo, o processo de seleção não pôde ser óbvio. O dependente em cima do roteador que é o iniciador, os Certificados diferentes é selecionado para o processo de autenticação com relação à ordem de certificado de registro.

Às vezes um certificado pode ser apoiado por somente um lado da conexão, como na versão 1 x509, que não é uma função de mistura típica que seja usada a fim assinar. O túnel VPN pôde ser estabelecido somente de um lado da conexão.

## Referência RFC para IKEv1

Está aqui um pique do RFC4945:

3.2.7.1. Especificando autoridades de certificação

Ao **pedir a** troca da em-faixa dos materiais de ajuste, as aplicações DEVEM gerar CERTREQs para cada âncora da confiança do par que a **política local** julga **explicitamente** confiada durante uma troca dada.
O RFC não é claro. **A política local explicitamente** pôde relacionar-se ao comando do confiança-**ponto Ca** que é configurado no perfil cripto ISAKMP. O problema é aquele na fase MM3 e MM4 do processo, você não pode selecionar um perfil ISAKMP a menos que você usar um endereço IP de Um ou Mais Servidores Cisco ICM NT para a identidade e os confiança-pontos porque a autenticação no MM5 e a fase MM6 do processo devem ocorrer primeiramente. Por este motivo, a **política local** relaciona-se **explicitamente a** todos os confiança-pontos que são configurados no dispositivo.

> Nota: Esta informação não é específico da Cisco, mas é IKEv1-specific.

## Seleção do perfil IKEv2 com identidades que sobrepõem

Antes que os certificados múltiplos para IKEv2 estejam descritos, é importante conhecer a

maneira que os perfis são selecionados quando a identidade do fósforo é usada, que está satisfeita para todos os perfis. Esta não é uma encenação recomendada porque os resultados da negociação IKEv2 dependem dos fatores múltiplos. Os mesmos problemas existem para IKEv1 quando os perfis que sobrepõem são usados.

Está aqui uma configuração do iniciador do exemplo IKEv2:

```
*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
```

O tipo endereço da identidade é usado para ambos os lados da conexão. A autenticação através dos Certificados (podem igualmente ser as chaves pré-compartilhada) não é importante para este exemplo. O que responde tem perfis múltiplos esse todo o fósforo o 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 que responde deve escolher o perfil baseado na identidade que é recebida. A identidade é um endereço do 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 da **identidade do fósforo** que é configurado. Os IO escolhem último na configuração, que é **profile3 n**este exemplo:

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

A fim verificar a ordem, inscreva o **comando profile ikev2 cripto da mostra**.

> Nota: Mesmo quando há um endereço genérico (0.0.0.0) no perfil, está selecionado ainda. Os IO não tentam encontrar um melhor fósforo; tenta encontrar o primeiro fósforo. Contudo, isto ocorre somente porque todos os perfis têm o mesmo **comando remote da identidade do**

fósforo configurado. Para os IKEv1 e os perfis IKEv2 que têm regras diferentes da identidade do fósforo, a mais específica é usado sempre. Cisco recomenda que você para não ter os perfis configurados com a identidade de sobreposição do fósforo comanda porque é difícil prever o perfil que é selecionado.

Nesta encenação, profile3 é selecionado pelo que responde, mas profile1 é usado para a interface de túnel. Isto faz com que um erro apareça quando o ID de proxy é negociado:

```
*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
```

### IKEv2 fluem quando os Certificados são usados

Quando os Certificados são usados para IKEv2 a fim autenticar, o iniciador não envia o payload do pedido do certificado no primeiro pacote:

```
*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
```

As respostas do que responde com o payload do pedido do certificado (segundo pacote) e todos os CA porque o que responde não tem nenhum conhecimento do perfil que deve ser usado nesta fase. O pacote que contém a informação 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 confiança-ponto que combine CA proposto:

```
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 envia então o terceiro pacote com o pedido do certificado e o payload do certificado. Este pacote é cifrado já com o material de ajuste da fase do 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 que responde ao iniciador e contém somente 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 similar ao IKEv1 flui. O que responde deve enviar o payload do pedido do certificado honesto sem conhecimento do perfil que deve ser usado, que cria os mesmos problemas que são descritos previamente para IKEv1 (de uma perspectiva do protocolo). Contudo, a aplicação nos IO é melhor para o IKEv2 do que para o IKEv1.

## Confiança-ponto IKEv2 imperativo para o iniciador

Está aqui um exemplo de quando um iniciador IKEv2 tenta usar um perfil com certificado de autenticação e não tem nenhum confiança-ponto configurado sob esse perfil:

```
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 primeiro pacote é enviado sem nenhum payload do pedido do certificado, como descrito anteriormente. A resposta do que responde inclui o payload do pedido do certificado para todos os confiança-pontos que são definidos no modo de configuração global. Isto 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 conhece o confiança-ponto que deve ser usado a fim assinar. Este é o principal diferença quando a aplicação IKEv2 é comparada ao IKEv1. O iniciador IKEv2 deve ter o confiança-ponto configurado sob o perfil do iniciador IKEv2, mas não é necessário para o que

responde IKEv2.

Está aqui um trecho da referência de comandos:

Se não há nenhum ponto confiável definido na configuração de perfil IKEv2, o padrão é **validar o certificado** usando todos os pontos confiáveis que são definidos na configuração global
Épossível definir confiança-pontos diferentes; um a fim assinar e diferente a fim validar. Infelizmente, o confiança-ponto imperativo que é configurado sob o perfil IKEv2 não resolve todos os problemas.

## R2 como o iniciador IKEv2

Neste exemplo, o R2 é o 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
```

Neste exemplo, o r1 é o que responde 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, o R2 envia o primeiro pacote sem nenhum pedido do certificado. O que responde responde com um pedido do certificado para todos os confiança-pontos configurados. A ordem das cargas úteis é similar ao IKEv1 e é dependente dos Certificados que são 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 no r1 é associado com o confiança-ponto **TP2**, assim que o primeiro payload do pedido do certificado é para CA que é associado com o confiança-ponto **TP2**. Assim, o R2 seleciona-o para a autenticação (primeira regra do fósforo):

```
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
```

Então, o R2 prepara uma resposta (pacote 3) com o payload do pedido da certificação que seja associada com o **TP2**. O r1 não pode confiar o certificado desde que é configurado para a validação contra o confiança-ponto **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 previamente, Cisco recomenda que você não usa confiança-pontos múltiplos sob um perfil IKEv2. Quando você usa confiança-pontos múltiplos, é necessário assegurar-se de que os ambos os lados confiem exatamente os mesmos confiança-pontos. Por exemplo, o r1 e o R2 têm TP1 e TP2 configurados em seus perfis.

# Resumo

Esta seção fornece um sumário breve da informação que é descrita no documento.

O índice do payload do pedido do certificado depende da configuração. Se um confiança-ponto específico está configurado para o perfil ISAKMP e o roteador é o iniciador ISAKMP, a seguir o pedido do certificado no MM3 contém somente CA que é associado com o confiança-ponto. Contudo, se o mesmo roteador é o que responde ISAKMP, a seguir o pacote MM4 que é enviado pelo roteador inclui cargas úteis do pedido dos certificados múltiplos para todos os confiança-pontos globalmente-definidos (quando o comando do confiança-**ponto Ca** não é tomado na consideração). Isto ocorre porque o que responde ISAKMP pode determinar o perfil ISAKMP que deve ser usado somente depois que recebe o MM5 e o pedido do certificado que está incluído no MM4.

O payload do pedido do certificado no MM3 e no MM4 é importante devido à primeira regra do fósforo. A primeira regra do fósforo determina o confiança-ponto que é usado para a seleção do certificado, que é precisada para a autenticação no MM5 e no MM6.

A ordem de payload do pedido do certificado depende na ordem dos Certificados que são instalados. O expedidor do primeiro certificado que aparece na saída do **comando certificate cripto do pki da mostra** é enviado primeiramente. Este primeiro certificado é último que é registrado.

Épossível configurar confiança-pontos múltiplos para um perfil ISAKMP. Se isto é executado, a seguir todas as regras precedentes ainda aplicam-se.

Todos os problemas e advertências que são descritos neste documento são devido ao projeto do protocolo IKEv1. O estágio da autentificação ocorre no MM5 e no MM6, quando as propostas para a autenticação (pedidos do certificado) deverem ser enviadas em uma fase mais adiantada (honesto) sem conhecimento do perfil ISAKMP que deve ser usado. Este não é um problema específico da Cisco e é relacionado às limitações do projeto do protocolo IKEv1.

O protocolo IKEv2 é similar ao IKEv1 com respeito ao processo de negociação do certificado. Contudo, a aplicação nos IO força o uso de confiança-pontos específicos para o iniciador. Isto não resolve todas as edições. Quando os confiança-pontos múltiplos estão configurados para um único perfil e um único confiança-ponto está configurado no outro lado, é ainda possível encontrar problemas com autenticação. Cisco recomenda que você usa configurações simétricas do confiança-ponto para ambos os lados da conexão (os mesmos confiança-pontos configurados para ambos os perfis IKEv2).

Estão aqui algumas observações importantes sobre a informação que é descrita neste documento:

- Com configurações assimétricas do confiança-ponto para os perfis IKEv1 dos pares, o túnel pôde iniciar de somente um lado do túnel. A configuração do confiança-ponto para o perfil IKEv1 é opcional.
- Com configurações assimétricas do confiança-ponto para os perfis IKEv2 dos pares, o túnel pôde iniciar de somente um lado do túnel. A configuração do confiança-ponto para o perfil IKEv2 é imperativa para o iniciador.
- A ordem do payload do pedido do certificado depende na ordem dos Certificados que aparecem na saída do **comando certificate cripto do pki da mostra** (primeiro fósforo).
- A ordem do payload do pedido do certificado determina o certificado que é selecionado pelo que responde (primeiro fósforo).
- Quando você usa perfis múltiplos para o IKEv1 e o IKEv2 e tem as mesmas regras da identidade do fósforo configuradas, é difícil prever os resultados (fatores demais envolvidos).
- Cisco recomenda que você usa configurações simétricas do confiança-ponto para o IKEv1 e o IKEv2.

# Informações Relacionadas

- Intercâmbio de chave de Internet para o manual de configuração do IPsec VPN, Cisco IOS

Release 15M&T - certificado ao mapeamento do perfil ISAKMP

- Referência de comandos do Cisco IOS Security: Comandos A ao C - confiança-ponto Ca com o eou claro
- Suporte Técnico e Documentação - Cisco Systems