適用於具有多個證書的配置檔案的IOS IKEv1和IKEv2資料包交換過程

目錄

簡介

本檔案介紹使用憑證驗證時Internet金鑰交換版本1(IKEv1)和Internet金鑰交換版本2(IKEv2)封包交換程式以及可能會發生的問題。

以下是本文所述主題的清單：

- 網際網路金鑰交換(IKE)發起方和IKE響應方的證書選擇標準
- 當多個IKE配置檔案匹配時，IKE配置檔案匹配條件（對於重疊和非重疊情況）
- 在IKE配置檔案下不使用信任點時的預設設定和行為
- IKEv1和IKEv2在配置檔案和證書選擇標準方面的差異

> **附註**：有關如何對特定問題進行故障排除的詳細資訊，請參閱正確的部分。此外，本文檔末尾提供了簡短摘要。

必要條件

需求

思科建議您瞭解以下主題：

- Cisco IOS® VPN配置
- IKEv1和IKEv2通訊協定（封包交換）

採用元件

本檔案中的資訊是根據Cisco IOS版本15.3T。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路正在作用，請確保您已瞭解任何指令可能造成的影響。

背景資訊

當使用多個信任點和多個IKE配置檔案時，會出現本文檔中描述的問題。

本文檔中使用的初始示例具有每台路由器上帶有兩個信任點的IKEv1 LAN到LAN隧道。開始時，配置可能看起來是正確的。但是，由於**ca trust-point**命令用於Internet安全關聯和金鑰管理協定(ISAKMP)配置檔案行為以及本地儲存中註冊證書的順序，因此只能從連線的一端啟動VPN隧道。

當路由器是ISAKMP啟動器時，使用**ca trust-point**命令為ISAKMP配置檔案配置不同的行為。發生問題的原因可能是ISAKMP發起程式從一開始就知道ISAKMP配置檔案，因此為配置檔案配置的**ca trust-point**命令可能會影響主模式資料包3(MM3)中證書請求的負載。 但是，當路由器是ISAKMP響應方時，它在收到主模式資料包5(MM5)（包括建立繫結所需的IKE ID）後，將入站流量繫結到特定ISAKMP配置檔案。這就是不能對主模式封包4(MM4)套用任何**ca trust-point** 指令的原因，因為設定檔在MM5之前未確定。

本檔案將說明MM3和MM4中憑證請求負載的順序，以及對整個交涉過程的影響，以及它僅允許從VPN通道的一端建立連線的原因。

以下是IKEv1啟動器和響應器行為的摘要：

	IKEv1啟動器	**IKEv1回應端**
傳送請求	僅對配置檔案中配置的信任點傳送特定請求	傳送對所有可用信任點的請求
驗證請求	根據配置檔案中配置的特定信任點進行驗證	根據配置檔案中配置的特定信任點進行驗證

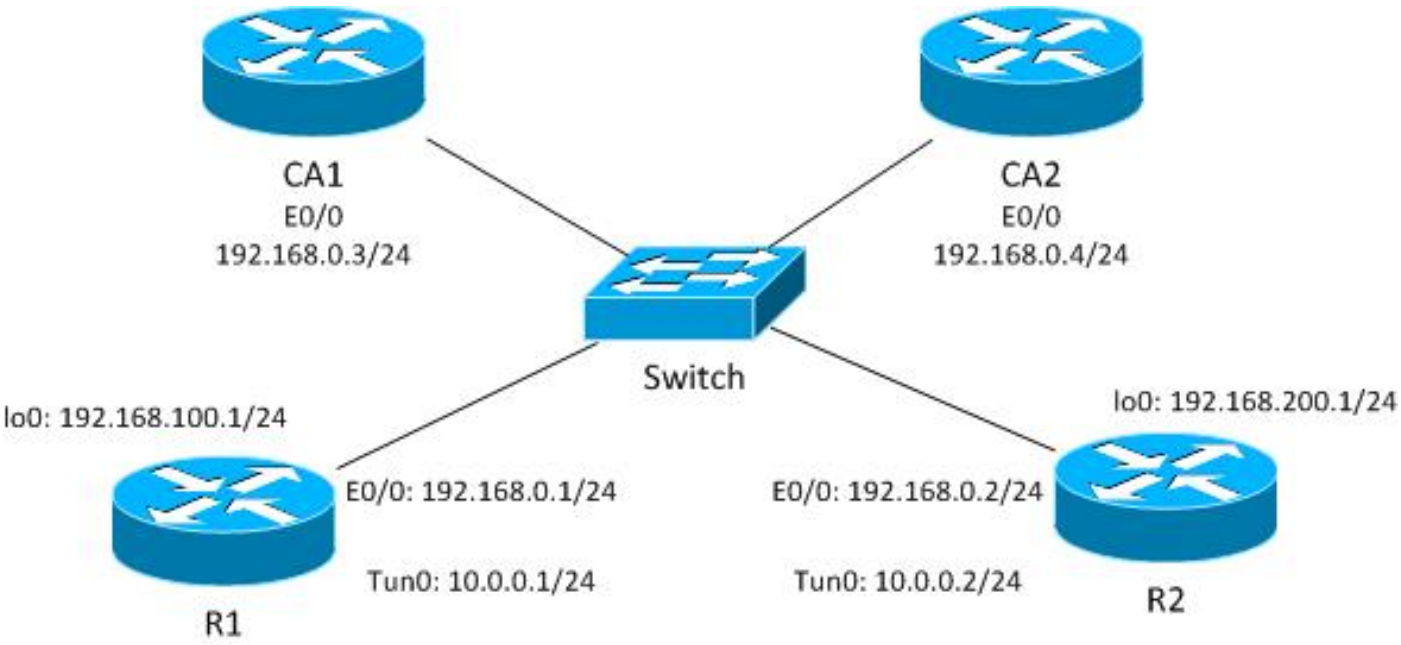
對於具有多個ISAKMP配置檔案並使用全域性配置的信任點的ISAKMP響應者，思科建議您不要使用**ca trust-point**命令。對於具有多個ISAKMP配置檔案的ISAKMP啟動器，思科建議您在每個配置檔案中使用**ca trust-point**命令來縮小證書選擇過程的範圍。

IKEv2協定與IKEv1協定存在相同的問題，但**pki trustpoint**命令的不同行為有助於防止問題的發生。這是因為**pki trustpoint**命令對於IKEv2啟動器是強制性的，而**ca trust-point**命令對於IKEv1啟動器是可選的。在某些情況下（同一配置檔案下存在多個信任點），可能會出現上述問題。因此，思科建議對連線的兩端使用對稱信任點配置（在兩個IKEv2配置檔案下配置的相同信任點）。

拓撲

這是一個用於本文檔中所有示例的通用拓撲。

附註：路由器1(R1)和路由器2(R2)使用虛擬通道介面(VTI)訪問環回。這些VTI受IPSec保護。



對於此IKEv1示例，每台路由器為每個證書頒發機構(CA)具有兩個信任點，並且每個信任點的證書都已註冊。

當R1是ISAKMP發起方時，隧道會正確協商，流量會受到保護。這是預期行為。當R2是ISAKMP發起方時，第1階段協商失敗。

附註：對於本文檔中的IKEv2示例，拓撲和編址與所示的IKEv1示例相同。

封包交換程式

本節介紹用於資料包交換過程的IKEv1和IKEv2配置變體，以及可能出現的問題。

具有多個證書的IKEv1

以下是具有多個證書的IKEv1的R1網路和VPN配置：

```
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
```

以下是具有多個證書的IKEv1的R2網路和VPN配置：

```
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
```

在本例中，R1有兩個信任點：一個使用IOSCA1，另一個使用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
```

在本例中，R2也有兩個信任點：一個使用IOSCA1，另一個使用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
```

必須注意這些配置中的單一差異：r1 ISAKMP配置檔案對IOSCA1 trust-point使用ca trust-point命令

，這表示R1僅信任由該特定信任點驗證的證書。相反，R2信任所有全域性定義的信任點驗證的所有證書。

R1作為IKEv1啟動器

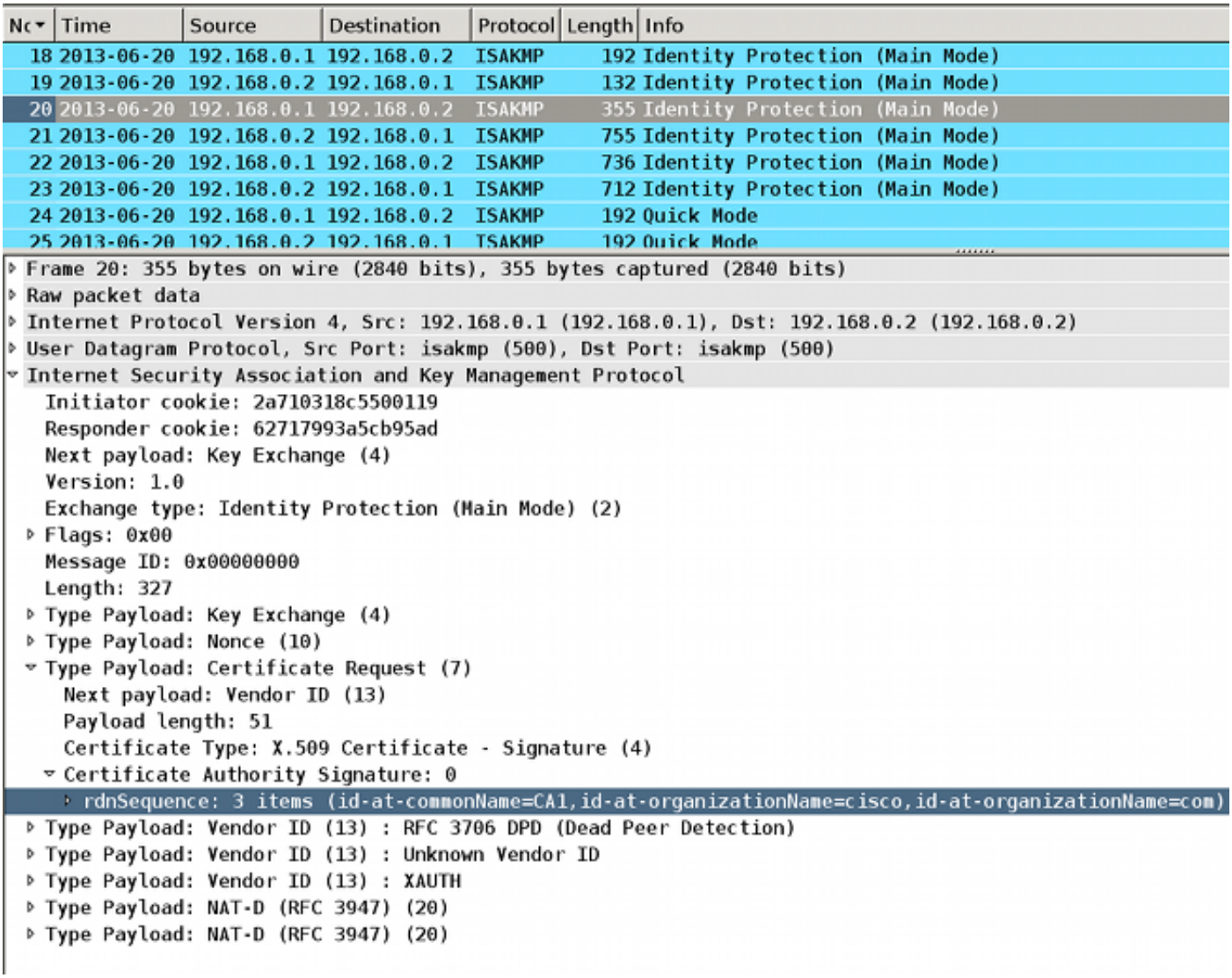
以下是R1和R2的debug命令：

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

此處，R1啟動隧道並將證書請求傳送到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
```

必須注意的是，資料包僅包含一個證書請求，該請求僅用於IOSCA1信任點。這是ISAKMP配置檔案的當前配置(CN=CA1, O=cisco， O=com)的預期行為。 不會傳送其他證書請求，可以使用嵌入式資料包捕獲功能進行驗證：



當R2收到資料包時，它開始處理證書請求，該請求會建立一個匹配，確定信任點以及用於在MM5中進行身份驗證的關聯證書。處理順序與ISAKMP資料包中的證書請求負載相同。這表示使用第一個相符專案。在此案例中，只有一個匹配，因為R1配置了特定信任點，並且只傳送了一個與該信任點關聯的證書請求。

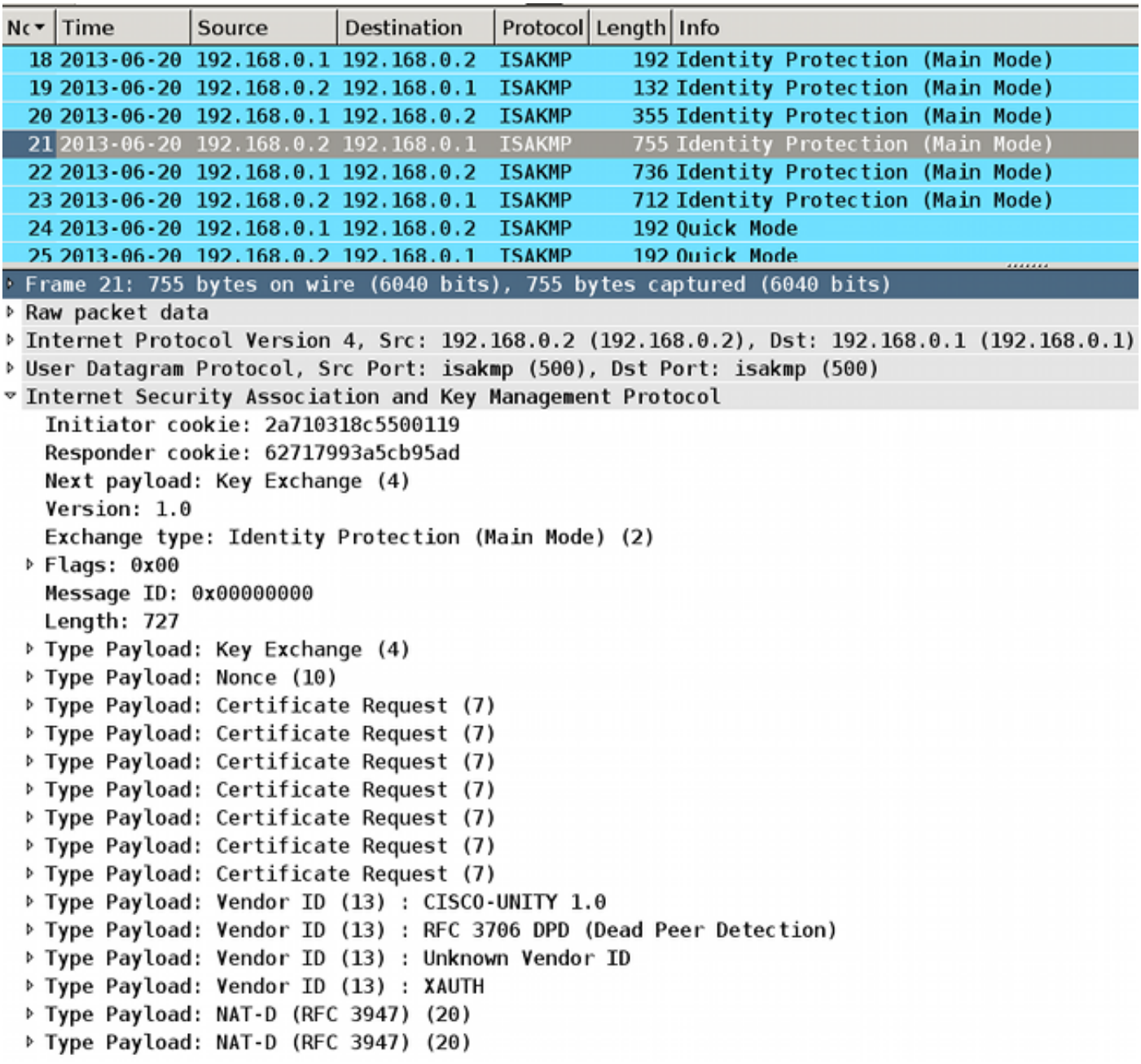
```
*Jun 20 13:00:37.617: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.617: ISAKMP:(1010): peer wants cert issued
 by cn=CA1,o=cisco,o=com
*Jun 20 13:00:37.617:  Choosing trustpoint IOSCA1 as issuer
```

之後，R2準備MM4。這是包含所有受信任信任點的證書請求的資料包。由於R2是ISAKMP響應方，因此所有全域性定義的信任點都是受信任的(未檢查ca trust-point 配置)。 其中兩個信任點是手動定義的(IOSCA1和IOSCA2)，其餘信任點是預定義的。

```
*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
```

您可以使用Wireshark驗證封包。來自R2的MM4資料包包含七個證書請求條目：



然後，R1收到來自R2的MM4，其中包含多個證書請求欄位：

```
*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
```

R1上的第一個匹配規則將第一個證書請求與**IOSCA1**信任點匹配。這確定R1使用與信任點**IOSCA1關聯的證書在**MM5中進行身份驗證。完全限定域名(FQDN)用作IKE ID。這是由於ISAKMP配置檔案中的**自身**身份fqdn配置所致：

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

R2接收並處理MM5。接收的IKE ID(**R1.cisco.com**)與ISAKMP配置檔案**prof1**匹配。然後驗證接收的證書並成功進行身份驗證：

```
*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
```

然後，R2使用與IOSCA1關聯的證書為MM6**準備**:

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

R1收到資料包，R1驗證證書和身份驗證：

```
*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
```

第一階段到此結束。第二階段按常規協商。已成功建立通道並保護流量。

R2作為IKEv1啟動器

以下示例說明了R2啟動同一IKEv1隧道的過程，並說明為什麼沒有建立該隧道。

附註：刪除部分日誌，以便只關注與上一節所示示例有關的差異。

R2向MM3傳送七個證書請求負載，因為R2沒有與ISAKMP配置檔案關聯的信任點（所有信任點都是受信任的）：

```
*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
```

當R1收到來自R2的資料包時，它處理證書請求並匹配IOSCA1信任點，後者確定MM6中傳送的證書：

```
*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
```

然後，R1準備具有證書請求負載的MM4資料包。現在有多個證書請求負載：

```
*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
```

使用嵌入式資料包捕獲(EPC)和Wireshark驗證日誌：

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)

```
▸ Flags: 0x00
  Message ID: 0x00000000
  Length: 727
▸ Type Payload: Key Exchange (4)
▸ Type Payload: Nonce (10)
▸ Type Payload: Certificate Request (7)
▸ Type Payload: Certificate Request (7)
▸ Type Payload: Certificate Request (7)
▸ Type Payload: Certificate Request (7)
▸ Type Payload: Certificate Request (7)
▸ Type Payload: Certificate Request (7)
▸ Type Payload: Certificate Request (7)
▸ Type Payload: Vendor ID (13) : CISCO-UNITY 1.0
▸ Type Payload: Vendor ID (13) : RFC 3706 DPD (Dead Peer Detection)
▸ Type Payload: Vendor ID (13) : Unknown Vendor ID
▸ Type Payload: Vendor ID (13) : XAUTH
▸ Type Payload: NAT-D (RFC 3947) (20)
▸ Type Payload: NAT-D (RFC 3947) (20)
```

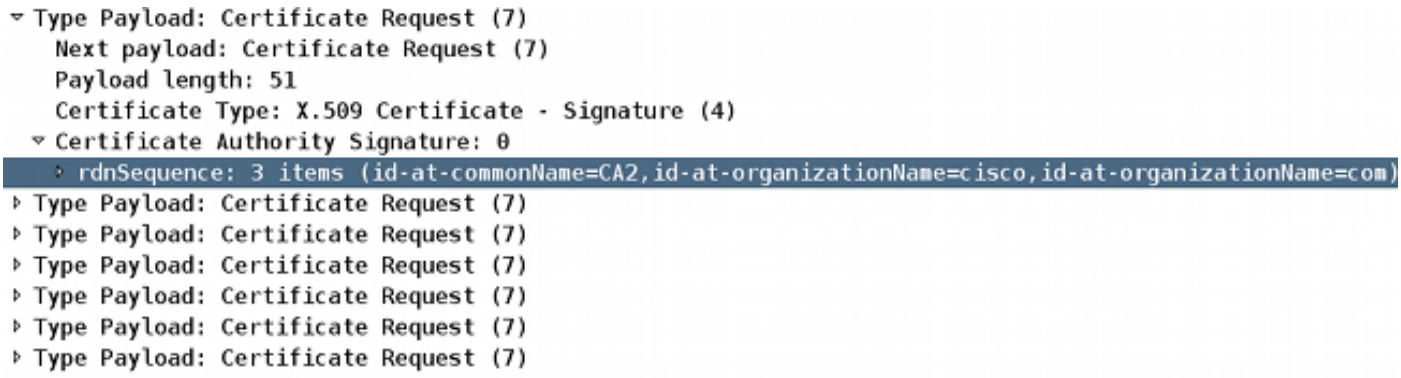
即使R1在ISAKMP配置檔案中配置為單個信任點(IOSCA1)，仍然會傳送多個證書請求。之所以會出現這種情況，是因為ISAKMP配置檔案中的**ca trust-point**命令確定了證書請求負載，但僅當路由器是ISAKMP會話的發起方時。如果路由器是響應方，則所有全域性定義的信任點都有多個證書請求負載，因為R1尚不知道用於IKE會話的ISAKMP配置檔案。

接收MM5後，入站IKE會話繫結到特定的ISAKMP配置檔案，該配置檔案包括IKE ID。然後，特定配置檔案的**match identity**命令將IKE會話繫結到配置檔案。但是，路由器現在才能確定這一點。可能有多個ISAKMP配置檔案，每個配置檔案的**ca trust-point**命令不同。

因此，R1必須傳送所有全域性配置的信任點的證書請求。

請參閱**ca trust-point** 命令的命令參考:

啟動IKE的路由器和響應IKE請求的路由器應具有對稱信任點配置。例如，在傳送CERT-REQ負載時，執行RSA簽名加密和身份驗證的響應路由器（在IKE主模式下）可能會使用全域性配置中定義的信任點。但是，路由器可能會使用在ISAKMP配置檔案中定義的信任點受限清單進行證書驗證。如果將對等體（IKE啟動器）配置為使用其信任點在響應路由器的全域性清單中，但不在響應路由器的ISAKMP配置檔案中使用的證書，則會拒絕該證書。（但是，如果發起方路由器不知道響應方路由器的全域性配置中的信任點，則證書仍可以進行身份驗證。）
現在驗證MM4封包詳細資訊，以探索第一個憑證要求負載：



由於證書的安裝順序，從R1傳送的MM4資料包在第一證書請求負載中包括IOSCA2信任點；第一個由IOSCA2信任點簽署：

```
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>
```

將IOSCA1信任點包含在第一個證書請求負載中時，與從R2傳送的MM3資料包進行比較：

```
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>
```

現在R2收到來自R1的MM4資料包，並開始處理證書請求。第一個證書請求負載與IOSCA2 trust-point匹配：

```
*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
```

當R2準備MM5資料包時，它使用與IOSCA2信任點關聯的證書：

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

R1收到MM5資料包。由於R1僅信任IOSCA1 trust-point(對於ISAKMP配置檔案prof1)，因此證書驗證失敗：

```
*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
```

如果R1上的證書註冊順序不同(因為第一個顯示的證書由**IOSCA1信任點簽名)，則此配**置會起作用。此外，MM4中的第一個證書請求負載是**IOSCA1** trust-point，R2會選擇此點，並在MM6中的R1上成功驗證。

在配置檔案中沒有*ca trust-point*命令的IKEv1

對於具有多個配置檔案和信任點但在配置檔案中沒有特定信任點配置的方案，不存在任何問題，因為沒有驗證由**ca trust-point命令配置確定的特定信任**點。然而，選擇過程可能並不明顯。根據發起者的路由器，會根據證書註冊的順序為身份驗證過程選擇不同的證書。

有時，僅連線的一方可以支援證書，如x509版本1，它不是用於簽名的典型雜湊函式。VPN隧道只能從連線的一端建立。

IKEv1的RFC參考

以下是RFC4945中的片段：

3.2.7.1.指定證書頒發機構

當 **請求**金鑰材料的帶內交換時，實現應該為本地策略在給定交換期間明確視為可信的每個對等信任錨點生成CERTREQ。

RFC不清楚。**本地策略**可能會與**ca trust-point**命令顯式相關，該命令在加密ISAKMP配置檔案中配置。問題在於，在進程的MM3和MM4階段，您不能選擇ISAKMP配置檔案，除非您使用IP地址作為身份和信任點，因為進程的MM5和MM6階段中的身份驗證必須首先進行。因此，本**地策略**明確與裝置上配置的所有信任點相關。

附註：此資訊不是思科專用資訊，但是IKEv1專用資訊。

具有重疊標識的IKEv2配置檔案選擇

在描述IKEv2的多個證書之前，重要的是知道在使用匹配標識時選擇配置檔案的方式，這在所有配置檔案中是令人滿意的。這不是推薦的方案，因為IKEv2協商的結果取決於多個因素。當使用重疊的配置檔案時，IKEv1存在相同的問題。

以下是IKEv2啟動器配置的示例：

```
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
```

標識型別地址用於連線的兩端。通過證書進行身份驗證（也可以是預共用金鑰）對於本示例並不重要。響應方有多個配置檔案，這些配置檔案都與入站IKEv2流量匹配：

```
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
```

發起方傳送第三個IKEv2資料包，響應方必須根據收到的標識選擇配置檔案。標識是IPv4地址(192.168.0.1):

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

由於配置了match identity命令，因此所有配置檔案都滿足此身份。IOS會選擇組態中的最後一個專案，在本範例中為profile3:

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

若要驗證順序，請輸入show crypto ikev2 profile命令。

> **附註**：即使配置檔案中存在通用地址(0.0.0.0)，它仍然處於選中狀態。IOS不會嘗試尋找最佳相符專案；它試圖找到第一個匹配項。但是，之所以會出現這種情況，是因為所有配置檔案都配置了相同的match identity remote命令。對於具有不同匹配身份規則的IKEv1和IKEv2配置檔案，始終使用最具體的匹配身份規則。思科建議您不要使用overlapping match identity命令配置配置檔案，因為很難預測所選的配置檔案。

在此案例中，profile3由回應者選擇，但profile1用於通道介面。這會導致在交涉代理ID時出現錯誤：

```
*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
```

使用證書時的IKEv2流

當證書用於IKEv2進行身份驗證時，發起方不會在第一個資料包中傳送證書請求負載：

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

回應者會使用憑證請求負載（第二個封包）和所有CA回應，因為回應者不知道應該在這個階段使用的設定檔。包含此資訊的資料包將傳送到啟動器：

```
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)
```

發起方處理資料包並選擇與建議的CA匹配的信任點：

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

然後，發起方傳送包含證書請求和證書負載的第三個資料包。此封包已使用Diffie-Hellman(DH)階段的金鑰資料加密：

```
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)
```

第四個資料包從響應方傳送到發起方，並且僅包含證書負載：

```
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)
```

此處描述的流與IKEv1流類似。響應方必須提前傳送證書請求負載，而不知道應使用的配置檔案，這與（從協定的角度）之前描述的IKEv1問題相同。 但是，在IOS上實施對於IKEv2比IKEv1更好。

發起方的IKEv2強制信任點

以下是IKEv2發起方嘗試使用帶有證書身份驗證的配置檔案並且在該配置檔案下未配置信任點的示例：

```
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
```

如前所述，第一個資料包是在沒有任何證書請求負載的情況下傳送的。來自響應方的響應包括在全域性配置模式下定義的所有信任點的證書請求負載。啟動器接收此資訊：

```
*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
```

發起方不知道應該用於簽名的信任點。這是將IKEv2實現與IKEv1進行比較時的主要區別。IKEv2發起方必須在IKEv2發起方配置下配置信任點，但對於IKEv2響應方則沒有必要。

以下為指令參考摘要:

如果IKEv2配置檔案配置中沒有定義信任點，則預設使用全域性配置中定義的所有信任點來驗證證書
可以定義不同的信任點；一個用於簽名，另一個用於驗證。遺憾的是，在IKEv2配置檔案下配置的強制信任點不能解決所有問題。

R2作為IKEv2啟動器

在本示例中，R2是IKEv2啟動器：

```
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
```

在本示例中，R1是IKEv2響應方：

```
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
```

這裡，R2傳送第一個資料包，沒有任何證書請求。響應方對所有已配置的信任點發出證書請求。負載的順序與IKEv1類似，並取決於安裝的證書：

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

R1上第一個配置的證書與**TP2信任點關聯，因此第一個證書請求負載適用於與TP2信任點關聯的**CA。因此，R2選擇它進行身份驗證（第一個匹配規則）：

```
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
```

然後，R2使用與**TP2關聯的證書請求負載準備響應（資料包3）。** R1無法信任證書，因為它配置為根據**TP1信任點進行驗**證：

```
*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)
```

如前所述，思科建議您不要在一個IKEv2配置檔案下使用多個信任點。當使用多個信任點時，必須確保兩端信任完全相同的信任點。例如，R1和R2在其配置檔案中都配置了TP1和TP2。

摘要

本節簡短總結了本文所述資訊。

證書請求負載內容取決於配置。如果為ISAKMP配置檔案配置了特定信任點，並且路由器是ISAKMP啟動器，則MM3中的證書請求僅包含與信任點關聯的CA。但是，如果同一路由器是ISAKMP響應方，則路由器傳送的MM4資料包包含所有全域性定義的信任點(未考慮**ca trust-point**命令時)的多個證書請求負載。 出現這種情況是因為ISAKMP響應方可以確定僅在收到MM5和MM4中包含的證書請求之後才應使用的ISAKMP配置檔案。

由於第一個匹配規則，MM3和MM4中的證書請求負載非常重要。第一個匹配規則確定用於證書選擇的信任點，在MM5和MM6中進行身份驗證需要該信任點。

證書請求負載的順序取決於安裝的證書的順序。首先傳送**show crypto pki certificate**命令輸出中顯示的第一個證書的頒發者。第一個證書是最後一個註冊的證書。

可以為ISAKMP配置檔案配置多個信任點。如果執行此步驟，則以前的所有規則仍然適用。

本文檔中描述的所有問題和警告都是由IKEv1協定設計引起的。身份驗證階段在MM5和MM6中進行，而身份驗證建議（證書請求）必須在更早的階段（前）傳送，而不知道應使用的ISAKMP配置檔案。這不是思科特有的問題，與IKEv1協定設計的侷限性有關。

IKEv2通訊協定在憑證交涉流程方面與IKEv1類似。但是，在IOS上的實施強制使用特定信任點作為啟動器。這並不能解決所有問題。如果為單個配置檔案配置了多個信任點，而在另一端配置了單個信任點，仍有可能遇到身份驗證問題。思科建議對連線的兩端使用對稱信任點配置（為兩個IKEv2配置檔案配置的信任點相同）。

以下是有關本檔案所述資訊的一些重要說明：

- 如果對等體的IKEv1配置檔案具有非對稱信任點配置，則隧道可能僅從隧道的一側啟動。IKEv1配置檔案的信任點配置是可選的。
- 如果對等體的IKEv2配置檔案具有非對稱信任點配置，則隧道可能僅從隧道的一側啟動。IKEv2配置檔案的信任點配置對於啟動器是必需的。
- 憑證請求負載順序取決於**show crypto pki certificate**命令輸出中顯示的憑證的順序（第一個匹配）。
- 證書請求負載順序確定響應方選擇的證書（第一個匹配）。
- 當您為IKEv1和IKEv2使用多個配置檔案並且配置了相同的匹配身份規則時，很難預測結果（涉及的因素太多）。
- 思科建議您對IKEv1和IKEv2使用對稱信任點配置。

相關資訊

- Internet Key Exchange for IPsec VPN配置指南，Cisco IOS版本15M&T — 證書到ISAKMP配置檔案對映
- Cisco IOS安全命令參考：命令A到C - ca trust-point到clear eou
- 技術支援與文件 - Cisco Systems