具有多个证书的配置文件的IOS IKEv1和IKEv2数据包交换进程

目录

简介

本文档介绍使用证书身份验证时的互联网密钥交换版本1(IKEv1)和互联网密钥交换版本2(IKEv2)数据包交换过程以及可能发生的问题。

以下是本文档中介绍的主题列表：

- 互联网密钥交换(IKE)发起方和IKE响应方的证书选择条件
- 当匹配多个IKE配置文件时，IKE配置文件匹配条件（对于重叠和非重叠方案）
- IKE配置文件下未使用信任点时的默认设置和行为
- IKEv1和IKEv2在配置文件和证书选择条件方面的差异

> **注意**：有关如何排除特定问题的详细信息，请参阅正确的部分。此外，本文档末尾还提供了一个简短的摘要。

先决条件

要求

Cisco 建议您了解以下主题：

- Cisco IOS® VPN配置
- IKEv1和IKEv2协议（数据包交换）

使用的组件

本文档中的信息基于Cisco IOS版本15.3T。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

背景信息

当使用多个信任点和多个IKE配置文件时，本文档中描述的问题会出现。

本文档中使用的初始示例在每台路由器上有一个带两个信任点的IKEv1 LAN到LAN隧道。起初，配置似乎正确。但是，VPN隧道只能从连接的一端启动，因为**ca trust-point**命令用于互联网安全关联和密钥管理协议(ISAKMP)配置文件行为以及本地存储中注册证书的顺序。

当路由器是ISAKMP发起方时，使用**ca trust-point**命令为ISAKMP配置文件配置不同的行为。可能会发生问题，因为ISAKMP发起方从开始就知道ISAKMP配置文件，因此为配置文件配置的**ca trust-point**命令会影响主模式数据包3(MM3)中证书请求的负载。 但是，当路由器是ISAKMP响应方时，它会在收到主模式数据包5(MM5)后将入站流量绑定到特定ISAKMP配置文件，其中包括创建绑定所需的IKE ID。这就是为什么无法对主模式数据包4(MM4)数据包应用任何**ca trust-point** 命令的原因，因为在MM5之前未确定配置文件。

本文解释了MM3和MM4中证书请求负载的顺序以及对整个协商过程的影响，以及它仅允许从VPN隧道的一端建立连接的原因。

以下是IKEv1发起方和响应方行为的摘要：

	IKEv1启动器	**IKEv1响应器**
发送请求	仅发送配置文件下配置的信任点的特定请求	发送所有可用信任点的请求
验证请求	根据配置文件下配置的特定信任点进行验证	根据配置文件下配置的特定信任点进行验证

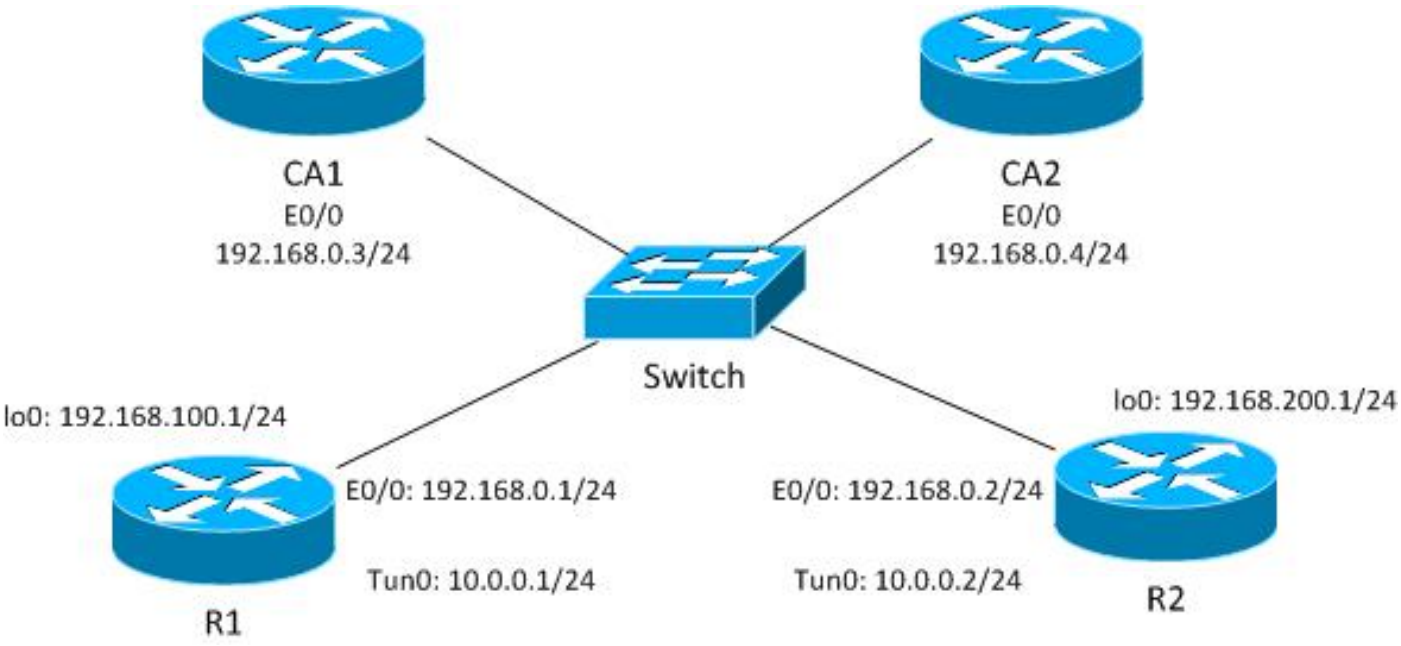
Cisco建议您不要对具有多个ISAKMP配置文件的ISAKMP响应器使用**ca trust-point**命令，并使用全局配置的信任点。对于具有多个ISAKMP配置文件的ISAKMP启动器，思科建议您在每个配置文件中使用**ca trust-point**命令来缩小证**书选择过**程的范围。

IKEv2协议与IKEv1协议存在相同的问题，但pki trustpoint命令的不同行为有助于防止问题的发生。这是因为**pki trustpoint**命令对IKEv2启动器是必需的，而**ca trust-point** 命令对IKEv1启动器是可选的。在某些情况下（一个配置文件下有多个信任点），可能会出现之前描述的问题。因此，思科建议您对连接的两端使用对称信任点配置（在两个IKEv2配置文件下配置的相同信任点）。

拓扑

这是用于本文档中所有示例的通用拓扑。

> **注意**：路由器1(R1)和路由器2(R2)使用虚拟隧道接口(VTI)来访问环回。这些VTI受IPSec保护。



对于此IKEv1示例，每台路由器对每个证书颁发机构(CA)具有两个信任点，并且每个信任点的证书都会注册。

当R1是ISAKMP发起方时，隧道会正确协商并保护流量。这是预料之中的现象。当R2是ISAKMP发起方时，第1阶段协商失败。

> **注意**：对于本文档中的IKEv2示例，拓扑和编址与IKEv1示例所示的拓扑和编址相同。

数据包交换过程

本节介绍用于数据包交换进程的IKEv1和IKEv2配置变体，以及可能出现的问题。

具有多个证书的IKEv1

以下是带有多个证书的IKEv1的R1网络和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
```

以下是带有多个证书的IKEv1的R2网络和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有两个信任点：一个使用IOSCA1，另一个使用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还有两个信任点：一个使用IOSCA1，另一个使用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
```

请注意以下配置中的单一差异：R1 ISAKMP配置文件对IOSCA1 信任点使用ca trust-point 命令，这

表示R1仅信任由该特定信任点验证的证书。相反，R2信任所有全局定义的信任点验证的所有证书。

R1作为IKEv1发起方

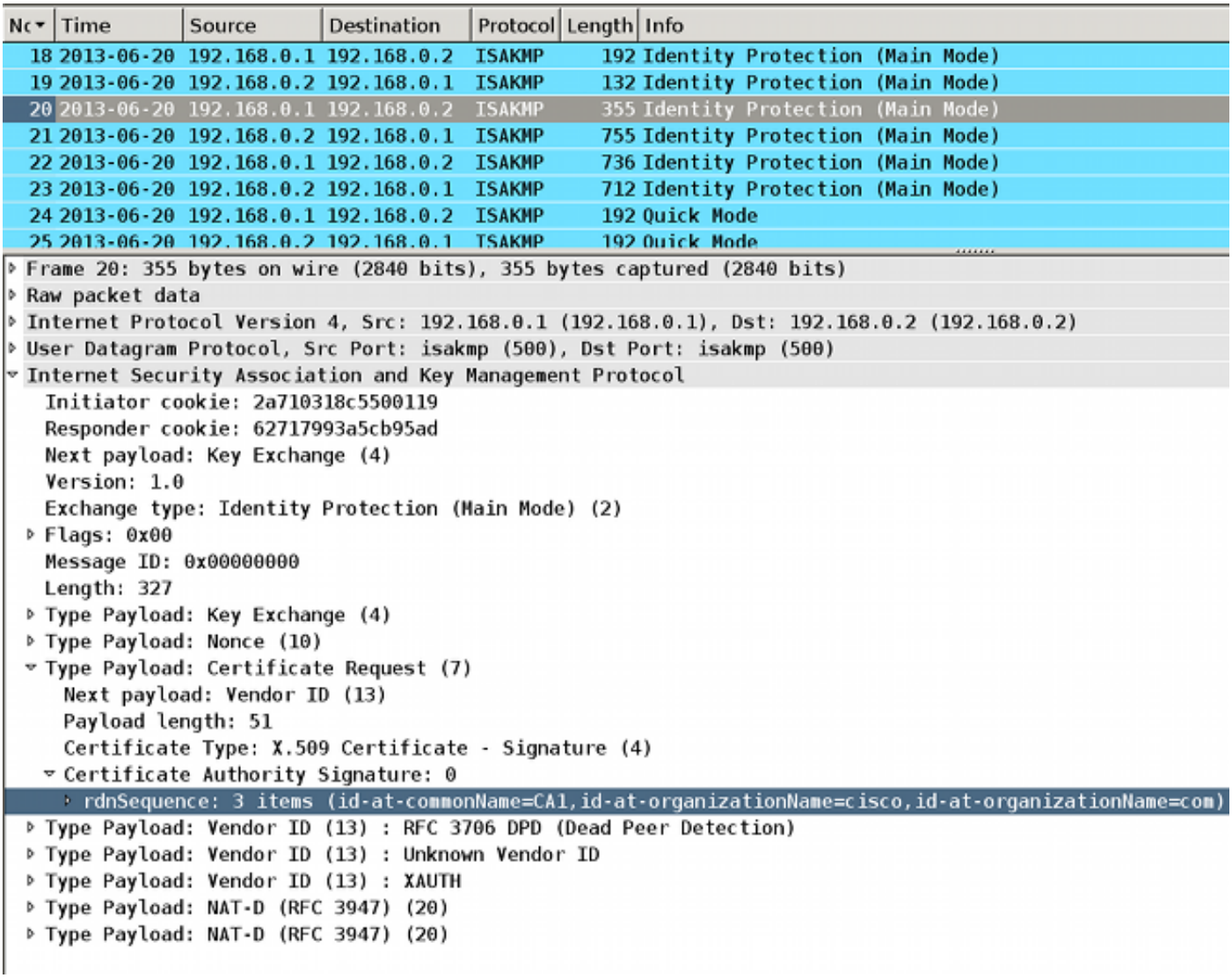
以下是R1和R2的调试命令：

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

在此，R1启动隧道并向MM3发送证书请求：

```
*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
```

请注意，数据包仅包含一个证书请求，该请求仅用于IOSCA1信任点。这是ISAKMP配置文件当前配置(CN=CA1, O=cisco， O=com)的预期行为。 不发送其他证书请求，您可以使用嵌入式数据包捕获功能验证这些请求：



当R2收到数据包时，它开始处理证书请求，该请求会创建一个匹配项，用于确定MM5中用于身份验证的信任点和关联证书。处理顺序与ISAKMP数据包中的证书请求负载相同。这意味着使用第一个匹配。在此场景中，只有一个匹配，因为R1配置了特定信任点，并且只发送一个与信任点关联的证书请求。

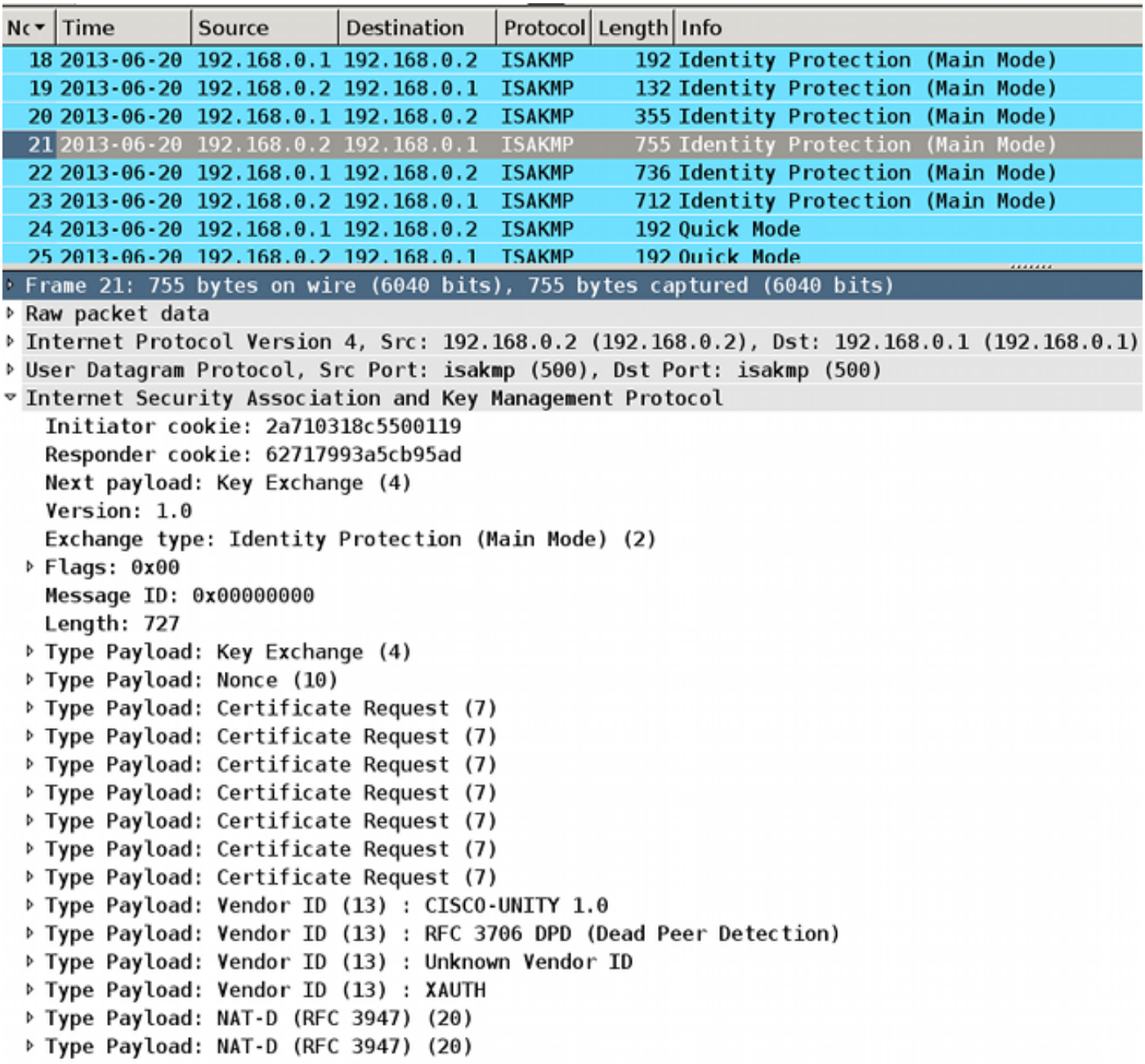
```
*Jun 20 13:00:37.617: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.617: ISAKMP:(1010): peer wants cert issued
 by cn=CA1,o=cisco,o=com
*Jun 20 13:00:37.617:  Choosing trustpoint IOSCA1 as issuer
```

之后，R2准备MM4。这是包含所有受信任信任点的证书请求的数据包。由于R2是ISAKMP响应器，因此所有全局定义的信任点都是受信任的(ca trust-point配置不会被选中)。 手动定义两个信任点(IOSCA1和IOSCA2)，其余预定义。

```
*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验证数据包。来自R2的MM4数据包包含七个证书请求条目：



然后，R1从R2接收MM4，其中包含多个证书请求字段：

```
*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上的第一个匹配规则将第一个证书请求与IOSCA1信任点匹配。这确定R1使用与信任点IOSCA1关联的证书在MM5中进行身份验证。完全限定域名(FQDN)用作IKE ID。这是由于ISAKMP配置文件中的自身身份fqdn配置：

```
*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
```

MM5由R2接收和处理。收到的IKE ID(R1.cisco.com)与ISAKMP配置文件prof1匹配。然后验证接收的证书并成功进行身份验证：

```
*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
```

然后，R2使用与IOSCA1关联的证书准备MM6:

```
*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收到数据包，R1验证证书和身份验证：

```
*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
```

这完成了第1阶段。第2阶段按常规协商。隧道成功建立，流量得到保护。

R2作为IKEv1发起方

本示例描述R2启动同一IKEv1隧道的过程，并说明未建立该隧道的原因。

注意：删除部分日志，以便仅关注与上一节中所示示例相关的差异。

R2发送带有7个证书请求负载的MM3，因为R2没有与ISAKMP配置文件关联的信任点（所有信任点均受信任）：

```
*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
```

当R1收到来自R2的数据包时，它会处理证书请求并匹配IOSCA1 信任点，该信任点确定在MM6中发送的证书：

```
*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
```

之后，R1使用证书请求负载准备MM4数据包。现在有多个证书请求负载：

```
*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
```

使用嵌入式数据包捕获(EPC)和Wireshark验证日志：

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)
```

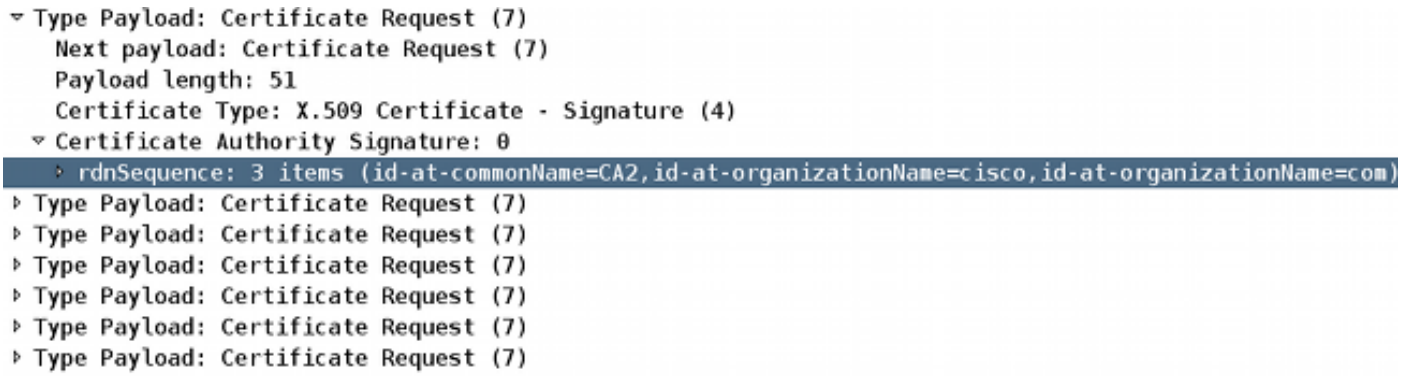
即使R1在ISAKMP配置文件中配置了单个信任点(IOSCA1)，也会发送多个证书请求。这是因为ISAKMP配置文件中的**ca trust-point**命令确定了证书请求负载，但仅当路由器是ISAKMP会话的发起者时才会发生。如果路由器是响应方，则所有全局定义的信任点都有多个证书请求负载，因为R1尚不知道用于IKE会话的ISAKMP配置文件。

在接收MM5（包括IKE ID）后，入站IKE会话将绑定到特定ISAKMP配置文件。之后，特定配置文件的**match identity**命令将IKE会话绑定到配置文件。但是，直到现在，路由器才能确定这一点。可能有多个ISAKMP配置文件，每个配置**文件配置了不**同的ca trust-point命令。

因此，R1必须为所有全局配置的信任点发送证书请求。

请参阅ca trust-point命**令的命令参**考：

发起IKE的路由器和响应IKE请求的路由器应具有对称的信任点配置。例如，执行RSA签名加密和身份验证的响应路由器（在IKE主模式下）可能使用在发送CERT-REQ负载时在全局配置中定义的信任点。但是，路由器可能会使用在ISAKMP配置文件中定义的受限信任点列表进行证书验证。如果对等体（IKE发起方）配置为使用信任点位于响应路由器的全局列表中但不在响应路由器的ISAKMP配置文件中的证书，则拒绝证书。（但是，如果发起路由器不知道响应路由器的全局配置中的信任点，则仍然可以对证书进行身份验证。）
现在检验MM4数据包详细信息以发现第一个证书请求负载：



从R1发送的MM4数据包在第一个证书请求负载中包含**IOSCA2** trust-point，因为证书的安装顺序；第一个由IOSCA2信**任点**签名：

```
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>
```

将IOSCA1信任点包含在第一个证书请求负载中时，与从R2**发送**的MM3数据包进行比较：

```
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>
```

现在，R2从R1接收MM4数据包并开始处理证书请求。第一个证书请求负载与**IOSCA2**信任点匹配：

```
*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
```

当R2准备MM5数据包时，它使用与IOSCA2信任点**关联的**证书：

```
*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.
```

MM5数据包由R1接收。由于R1仅信任**IOSCA1**信任点(对于ISAKMP配置文**件prof1**)，证书验证失败：

```
*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上证书注册的顺序不同，则此配置有效，因为第一个显示的证书由IOSCA1信任点签名。此外，MM4中的第一个证书请求负载是IOSCA1 trust-point，然后由R2选择，并在MM6中的R1上成功验证。

配置文件中不带*ca trust-point*命令的IKEv1

对于具有多个配置文件和信任点但配置文件中没有特定信任点配置的场景，不存在任何问题，因为没有验证由**ca trust-point命令配置确定的特定信任**点。但是，选择过程可能并不明显。根据作为发起方的路由器，会根据证书注册的顺序为身份验证过程选择不同的证书。

有时，只有连接的一端（例如x509版本1）支持证书，该版本不是用于签名的典型哈希函数。VPN隧道只能从连接的一端建立。

IKEv1的RFC参考

以下是RFC4945中的片段:

3.2.7.1.指定认证机构

当请 **求**带内密钥材料交换时，实施应为在给定交换期间本地策略明确认为受信任的每个对等 **体信任**

锚点生成CERTREQ。
RFC不明确。本**地策略**可能与在**加密ISAKMP配置**文件中配置的ca trust-point命令相关。问题是，在流程的MM3和MM4阶段，除非为身份和信任点使用IP地址，否则无法选择ISAKMP配置文件，因为必须首先在流程的MM5和MM6阶段进行身份验证。因此，本**地策略**明确地与设备上配置的所有信任点相关。

注意：此信息不特定于思科，但是特定于IKEv1。

具有重叠标识的IKEv2配置文件选择

在描述IKEv2的多个证书之前，必须了解使用匹配身份时选择配置文件的方式，这对所有配置文件都是满意的。这不是推荐的方案，因为IKEv2协商的结果取决于多个因素。当使用重叠的配置文件时，IKEv1也存在同样的问题。

以下是IKEv2启动器配置示例：

```
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
```

身份类型地址用于连接的两端。通过证书进行身份验证（也可以是预共享密钥）对本示例不重要。响应方有多个配置文件，这些配置文件均与入站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
```

发起方发送第三个IKEv2数据包，响应方必须根据收到的身份选择配置文件。标识是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'
```

所有配置文件都满足此身份，因为配置了**match identity**命令。IOS选择配置中的最后一个，在本例中为**profile3**:

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

要验证订单，请输入**show crypto ikev2 profile**命令。

> **注意**：即使配置文件中有通用地址(0.0.0.0)，也仍会选择它。IOS不尝试查找最佳匹配；它试图找到第一个匹配。但是，这只是因为所有配置文件都配置了相同的**match identity remote**命令。对于具有不同匹配身份规则的IKEv1和IKEv2配置文件，始终使用最具体的规则。Cisco建议您不要使用重叠的match identity命令配**置配置**文件，因为很难预测所选配置文件。

在此场景中，**响应方**选择了profile3，但**profile1**用于隧道接口。这会导致协商代理ID时出现错误：

```
*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
```

使用证书时的IKEv2流

当证书用于IKEv2以进行身份验证时，发起方不会在第一个数据包中发送证书请求负载：

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

响应方使用证书请求负载（第二个数据包）和所有CA进行应答，因为响应方不知道应在此阶段使用的配置文件。包含信息的数据包将发送到发起方：

```
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)
```

发起方处理数据包并选择与建议的CA匹配的信任点：

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

然后，发起方发送包含证书请求和证书负载的第三个数据包。此数据包已使用来自Diffie-Hellman(DH)阶段的密钥材料进行加密：

```
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)
```

第四个数据包从响应方发送到发起方，且仅包含证书负载：

```
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)
```

此处描述的流与IKEv1流类似。响应方必须在不知道应使用的配置文件的情况下提前发送证书请求负载，这会产生与之前为IKEv1描述的问题（从协议角度）相同的问题。 但是，IOS上的实施对IKEv2比对IKEv1更好。

发起方的IKEv2强制信任点

以下是IKEv2发起方尝试使用具有证书身份验证且未在该配置文件下配置信任点的配置文件的示例：

```
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，发送第一个数据包时不会发送任何证书请求负载。响应方的响应包括在全局配置模式下定义的所有信任点的证书请求负载。发起方收到以下消息：

```
*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
```

发起方不知道应该使用哪个信任点来签名。这是将IKEv2实现与IKEv1进行比较时的主要区别。IKEv2发起方必须在IKEv2发起方配置文件下配置信任点，但IKEv2响应方不需要配置信任点。

以下摘自命令参考：

如果IKEv2配置文件配置中未定义信任点，则默认使用全局配置中 **定义的**所有信任点来验证证书
可以定义不同的信任点；一个用于签名，另一个用于验证。遗憾的是，在IKEv2配置文件下配置的强制信任点无法解决所有问题。

R2作为IKEv2发起方

在本例中，R2是IKEv2启动器：

```
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
```

在本例中，R1是IKEv2响应方：

```
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
```

在此，R2发送第一个数据包，而不发送任何证书请求。响应方以针对所有已配置信任点的证书请求

做出响应。负载顺序与IKEv1类似，并且取决于安装的证书：

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

R1上第一个配置的证书与TP2信任点关联，因此第一个证书请求负载是用于与TP2信任点关联的CA。因此，R2选择它进行身份验证（第一个匹配规则）：

```
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
```

然后，R2使用与TP2关联的证书请求负载准备响应（数据包3）。R1无法信任该证书，因为它已配置为针对TP1信任点进行验证：

```
*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，思科建议您不要在一个IKEv2配置文件下使用多个信任点。当您使用多个信任点时，必须确保双方完全信任相同的信任点。例如，R1和R2的配置文件中都配置了TP1和TP2。

摘要

本部分简要概述本文档中介绍的信息。

证书请求负载内容取决于配置。如果为ISAKMP配置文件配置了特定信任点，并且路由器是ISAKMP发起方，则MM3中的证书请求仅包含与信任点关联的CA。但是，如果同一路由器是ISAKMP响应方，则路由器发送的MM4数据包会包含所有全局定义的信任点的多个证书请求负载(当未考虑**ca trust-point** 命令时)。 发生这种情况是因为ISAKMP响应方可以确定仅在收到MM5和MM4中包含的证书请求后才应使用的ISAKMP配置文件。

MM3和MM4中的证书请求负载非常重要，因为第一个匹配规则。第一个匹配规则确定用于证书选择的信任点，在MM5和MM6中进行身份验证需要该信任点。

证书请求负载的顺序取决于安装的证书的顺序。首先发送显示在show crypto pki certificate命令输出中**的第一个证**书的颁发者。第一个证书是最后一个注册的证书。

可以为ISAKMP配置文件配置多个信任点。如果执行此操作，则之前的所有规则仍适用。

本文档中描述的所有问题和警告都源于IKEv1协议设计。身份验证阶段发生在MM5和MM6中，而身份验证建议（证书请求）必须在较早阶段（前面）发送，而不知道应使用的ISAKMP配置文件。这不是思科特有的问题，与IKEv1协议设计的限制有关。

IKEv2协议与IKEv1在证书协商过程方面类似。但是，在IOS上实施会强制对发起方使用特定信任点。这不能解决所有问题。当为单个配置文件配置了多个信任点并且在另一端配置了单个信任点时，仍然可能遇到身份验证问题。思科建议您对连接的两端使用对称信任点配置（为两个IKEv2配置文件配置的相同信任点）。

以下是有关本文档中所述信息的一些重要说明：

- 对于对等体的IKEv1配置文件，使用非对称信任点配置时，隧道可能仅从隧道的一端启动。IKEv1配置文件的信任点配置是可选的。
- 对于对等体的IKEv2配置文件，使用非对称信任点配置，隧道可能仅从隧道的一端启动。IKEv2配置文件的信任点配置对发起方是必需的。
- 证书请求负载顺序取决于在show crypto pki certificate命令（第一个匹配）的输出中**显示的证**书的顺序。
- 证书请求负载顺序确定响应方选择的证书（第一个匹配项）。
- 当您对IKEv1和IKEv2使用多个配置文件并配置了相同的匹配身份规则时，很难预测结果（涉及的因素太多）。
- 思科建议对IKEv1和IKEv2使用对称信任点配置。

相关信息

- IPsec VPN的互联网密钥交换配置指南，Cisco IOS版本15M&T — 证书到ISAKMP配置文件映射
- Cisco IOS安全命令参考：命令A到C - ca trust-point至clear eou
- 技术支持和文档 - Cisco Systems