# IOS IKEv1- und IKEv2-Paketaustauschprozesse für Profile mit mehreren Zertifikaten

## Inhalt



## Einführung

In diesem Dokument werden die Paketaustauschprozesse Internet Key Exchange Version 1 (IKEv1) und Internet Key Exchange Version 2 (IKEv2) bei Verwendung der Zertifikatsauthentifizierung und mögliche Probleme beschrieben, die auftreten können.

Im Folgenden finden Sie eine Liste der Themen, die in diesem Dokument beschrieben werden:

- Die Zertifikatauswahlkriterien für den Initiator der Internet Key Exchange (IKE) und den IKE-Responder
- Das IKE-Profil erfüllt Kriterien, wenn mehrere IKE-Profile zugeordnet werden (bei überlappenden und nicht überlappenden Szenarien)
- Die Standardeinstellungen und das Verhalten, wenn unter den IKE-Profilen keine Vertrauenspunkte verwendet werden
- Unterschiede zwischen IKEv1 und IKEv2 hinsichtlich der Profil- und Zertifikatsauswahlkriterien

**Hinweis**: Weitere Informationen zur Behebung eines bestimmten Problems finden Sie im entsprechenden Abschnitt. Eine kurze Zusammenfassung finden Sie am Ende dieses Dokuments.

# Voraussetzungen

## Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- Cisco IOS® VPN-Konfiguration
- IKEv1- und IKEv2-Protokolle (Paketaustausch)

## Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf Cisco IOS Version 15.3T.

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

# Hintergrundinformationen

Die in diesem Dokument beschriebenen Probleme treten auf, wenn mehrere Vertrauenspunkte und mehrere IKE-Profile verwendet werden.

Die ersten Beispiele, die in diesem Dokument verwendet werden, haben einen IKEv1-LAN-zu-LAN-Tunnel mit zwei Vertrauenspunkten auf jedem Router. Zunächst scheint es, als ob die Konfiguration korrekt ist. Der VPN-Tunnel kann jedoch nur von einer Seite der Verbindung initiiert werden, da der Befehl **ca trust-point** für das Profilverhalten der Internet Security Association and Key Management Protocol (ISAKMP) und für die Reihenfolge der im lokalen Speicher registrierten Zertifikate verwendet wird.

Ein anderes Verhalten wird mit dem Befehl **ca trust-point** für das ISAKMP-Profil konfiguriert, wenn der Router der Initiator von ISAKMP ist. Ein Problem kann auftreten, weil der ISAKMP-Initiator das ISAKMP-Profil von Anfang an kennt. Der für das Profil konfigurierte Befehl **ca trust-point** kann daher die Payload für die Zertifikatsanforderung im Main Mode Packet 3 (MM3) beeinflussen. Wenn der Router jedoch der ISAKMP-Responder ist, bindet er den eingehenden Datenverkehr an ein bestimmtes ISAKMP-Profil, nachdem er das Main Mode Packet 5 (MM5) empfängt, das die IKE-ID enthält, die zum Erstellen der Bindung erforderlich ist. Daher ist es nicht möglich, einen

**CA-Trust-Point-**Befehl für das Main Mode Packet 4 (MM4)-Paket anzuwenden, da das Profil nicht vor dem MM5 bestimmt wird.

Die Reihenfolge der Zertifikatsanforderungs-Payload in MM3 und MM4 und die Auswirkungen auf den gesamten Verhandlungsprozess werden in diesem Dokument erläutert. Außerdem wird erläutert, warum die Verbindung nur von einer Seite des VPN-Tunnels aus hergestellt werden kann.

Nachfolgend finden Sie eine Zusammenfassung des Verhaltens von Initiator und Responder für IKEv1:

| | **IKEv1-Initiator** | **IKEv1-Responder** |
|---|---|---|
| **Anfrage senden** | Sendet spezifische Anfragen nur für die Vertrauenspunkte, die im Profil konfiguriert sind | Sendet Anfragen für alle verfügbaren Vertrauenspunkte |
| **Anfrage validieren** | Validiert für bestimmte Vertrauenspunkte, die im Profil konfiguriert werden | Validiert für bestimmte Vertrauenspunkte, die im Profil konfiguriert werden |

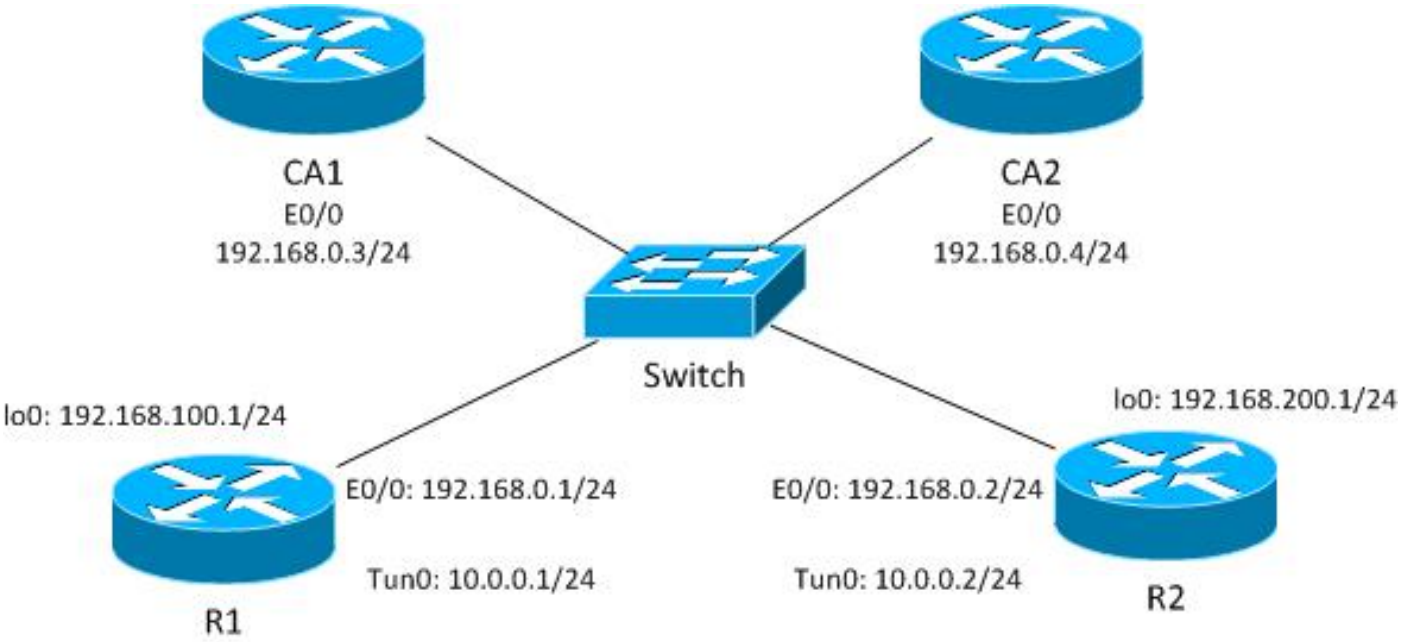
Cisco empfiehlt, den Befehl **ca trust-point** nicht für ISAKMP-Responder zu verwenden, die über mehrere ISAKMP-Profile verfügen und global konfigurierte Trust-Points verwenden. Für ISAKMP-Initiatoren mit mehreren ISAKMP-Profilen empfiehlt Cisco, den Zertifikatauswahlprozess mit dem Befehl **ca trust-point** in jedem Profil einzuschränken.

Das IKEv2-Protokoll hat die gleichen Probleme wie das IKEv1-Protokoll, aber das unterschiedliche Verhalten des **pki-Vertrauenspunktbefehls** verhindert das Auftreten der Probleme. Dies liegt daran, dass der Befehl **pki trustpoint** für den IKEv2-Initiator obligatorisch ist, während der Befehl **ca trust-point** für den IKEv1-Initiator optional ist. Unter bestimmten Umständen (mehrere Treuepunkte unter einem Profil) können die zuvor beschriebenen Probleme auftreten. Aus diesem Grund empfiehlt Cisco, für beide Seiten der Verbindung symmetrische Trust-Point-Konfigurationen zu verwenden (dieselben Vertrauenspunkte, die unter beiden IKEv2-Profilen konfiguriert wurden).

# Topologie

Dies ist eine generische Topologie, die für alle Beispiele in diesem Dokument verwendet wird.

> **Hinweis**: Router 1 (R1) und Router 2 (R2) verwenden Virtual Tunnel Interfaces (VTIs), um auf die Loopbacks zuzugreifen. Diese VTIs sind durch IPSec geschützt.




In diesem IKEv1-Beispiel verfügt jeder Router über zwei Vertrauenspunkte für jede Zertifizierungsstelle (Certificate Authority, CA), und die Zertifikate für jeden der Vertrauenspunkte werden registriert.

Wenn R1 der ISAKMP-Initiator ist, wird der Tunnel korrekt ausgehandelt und der Datenverkehr geschützt. Dieses Verhalten wird erwartet. Wenn R2 der ISAKMP-Initiator ist, schlägt die Phase1-Aushandlung fehl.

> **Hinweis**: Für die IKEv2-Beispiele in diesem Dokument sind Topologie und Adressierung die gleiche wie im IKEv1-Beispiel.

# Paketaustauschprozess

In diesem Abschnitt werden die für den Paketaustauschprozess verwendeten Konfigurationsvarianten IKEv1 und IKEv2 sowie mögliche Probleme beschrieben.

## IKEv1 mit mehreren Zertifikaten

Hier die R1-Netzwerk- und VPN-Konfiguration für IKEv1 mit mehreren Zertifikaten:

```
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
```

Hier die R2-Netzwerk- und VPN-Konfiguration für IKEv1 mit mehreren Zertifikaten:

```
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
```

In diesem Beispiel hat R1 zwei Vertrauenspunkte: einer verwendet **IOSCA1** und der zweite verwendet **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
```

In diesem Beispiel verfügt R2 auch über zwei Vertrauenspunkte: einer verwendet **IOSCA1** und der zweite verwendet **IOSCA2**:

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

Es ist wichtig, dass Sie bei diesen Konfigurationen nur einen Unterschied feststellen: Das R1 ISAKMP-Profil verwendet den Befehl **ca trust-point für den** Vertrauenspunkt **IOSCA1**, der angibt, dass R1 nur den Zertifikaten vertraut, die durch diesen spezifischen Vertrauenspunkt validiert werden. Im Gegensatz dazu vertraut R2 allen Zertifikaten, die von allen global definierten Treuepunkten validiert werden.

## R1 als IKEv1-Initiator

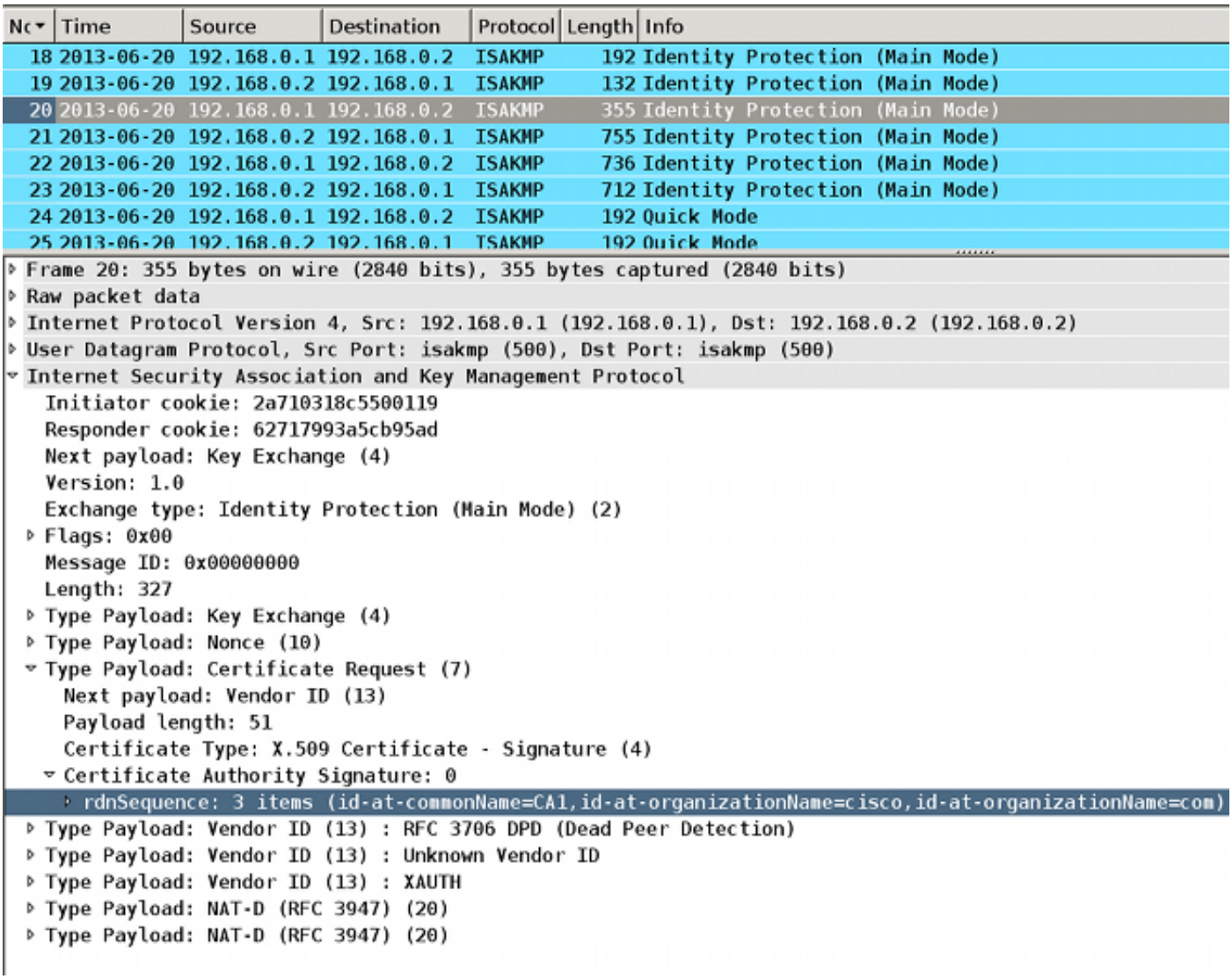
Hier sind die Debug-Befehle für R1 und R2:

- **R1# Debug crypto isakmp**
- **R1#-Debug-Verschlüsselung ipsec**
- **R1#-Debug-Verschlüsselung für die Pki-Validierung**

Hier initiiert R1 den Tunnel und sendet die Zertifikatsanforderung 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
```

Beachten Sie, dass das Paket nur eine Zertifikatsanforderung enthält, die nur für den **IOSCA1**-Vertrauenspunkt gilt. Dieses Verhalten wird bei der aktuellen Konfiguration des ISAKMP-Profils erwartet (**CN=CA1, O=cisco, O=com**). Es werden keine weiteren Zertifikatsanforderungen gesendet, die Sie mit der Funktion Embedded Packet Capture überprüfen können:



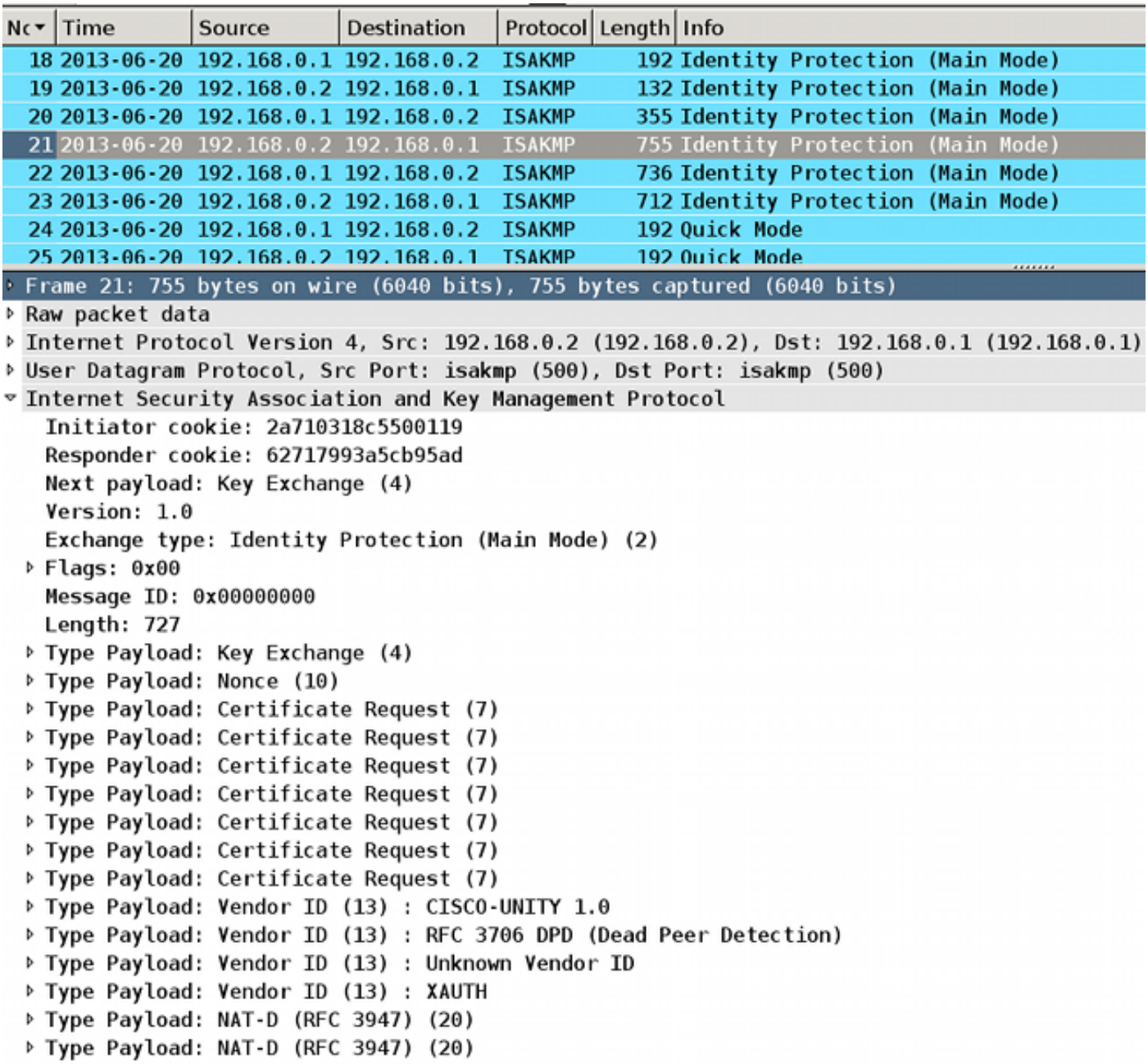

Wenn R2 das Paket empfängt, beginnt es mit der Verarbeitung der Zertifikatsanforderung, die eine Übereinstimmung erstellt, die den Vertrauenspunkt und das zugehörige Zertifikat bestimmt, das für die Authentifizierung im MM5 verwendet wird. Die Prozessreihenfolge entspricht der Payload der Zertifikatsanforderung im ISAKMP-Paket. Dies bedeutet, dass die erste Übereinstimmung verwendet wird. In diesem Szenario gibt es nur eine Übereinstimmung, da R1 mit einem bestimmten Vertrauenspunkt konfiguriert ist und nur eine Zertifikatsanforderung sendet, die dem Vertrauenspunkt zugeordnet ist.

```
*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
```

Anschließend bereitet R2 das MM4 vor. Dies ist das Paket, das die Zertifikatsanforderung für alle vertrauenswürdigen Vertrauenspunkte enthält. Da R2 der ISAKMP-Responder ist, sind alle global definierten Trust-Points vertrauenswürdig (die CA-Trust-Point-Konfiguration wird nicht überprüft). Zwei der Vertrauenspunkte werden manuell definiert (IOSCA1 und IOSCA2), der Rest ist vordefiniert.

```
*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
```

Sie können das Paket mit Wireshark überprüfen. Das MM4-Paket von R2 enthält sieben Einträge für Zertifikatsanfragen:



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

```
▸ Frame 21: 755 bytes on wire (6040 bits), 755 bytes captured (6040 bits)
▸ Raw packet data
▸ Internet Protocol Version 4, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1)
▸ User Datagram Protocol, Src Port: isakmp (500), Dst Port: isakmp (500)
▾ Internet Security Association and Key Management Protocol
    Initiator cookie: 2a710318c5500119
    Responder cookie: 62717993a5cb95ad
    Next payload: Key Exchange (4)
    Version: 1.0
    Exchange type: Identity Protection (Main Mode) (2)
  ▸ 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)
```

Anschließend empfängt R1 das MM4 von R2 mit mehreren Zertifikatsanforderungsfeldern:

```
*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
```

Die First-Match-Regel für R1 stimmt mit der ersten Zertifikatsanforderung mit dem **IOSCA1**-Vertrauenspunkt überein. Hierdurch wird festgelegt, dass R1 das Zertifikat verwendet, das dem Trust-Point **IOSCA1** für die Authentifizierung im MM5 zugeordnet ist. Der vollqualifizierte Domänenname (Fully Qualified Domain Name, FQDN) wird als IKE-ID verwendet. Grund hierfür ist die **FQDN-Selbstidentitätskonfiguration** im ISAKMP-Profil:

```
*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
```

Der MM5 wird von R2 empfangen und verarbeitet. Die erhaltene IKE-ID (**R1.cisco.com**) entspricht dem ISAKMP-Profil **prof1**. Das empfangene Zertifikat wird dann validiert, und die Authentifizierung ist erfolgreich:

```
*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
```

Anschließend bereitet R2 das MM6 mit dem Zertifikat vor, das mit **IOSCA1** verknüpft ist:

```
*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
```

Das Paket wird von R1 empfangen, und R1 überprüft das Zertifikat und die Authentifizierung:

```
*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
```

Damit ist Phase 1 abgeschlossen. Phase 2 wird wie gewohnt verhandelt. Der Tunnel wurde erfolgreich eingerichtet und der Verkehr ist geschützt.

## R2 als IKEv1-Initiator

In diesem Beispiel wird der Prozess beschrieben, wenn R2 denselben IKEv1-Tunnel initiiert, und es wird erläutert, warum dieser nicht eingerichtet wurde.

> **Hinweis**: Teile der Protokolle werden entfernt, um sich nur auf die Unterschiede im Vergleich zum Beispiel im vorherigen Abschnitt zu konzentrieren.

R2 sendet das MM3 mit sieben Zertifikatsanforderungs-Payloads, da R2 nicht über einen Vertrauensbereich verfügt, der dem ISAKMP-Profil zugeordnet ist (alle Vertrauenspunkte sind vertrauenswürdig):

```
*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
```

Wenn R1 das Paket von R2 empfängt, verarbeitet es die Zertifikatsanforderung und stimmt mit dem **IOSCA1**-Vertrauenspunkt überein, der das in MM6 gesendete Zertifikat bestimmt:

```
*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
```

Anschließend bereitet R1 das MM4-Paket mit der Payload für die Zertifikatsanforderung vor. Jetzt gibt es mehrere Payloads für Zertifikatsanfragen:

```
*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
```

Überprüfen Sie die Protokolle mit Embedded Packet Capture (EPC) und Wireshark:

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

```
▷ 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)
```

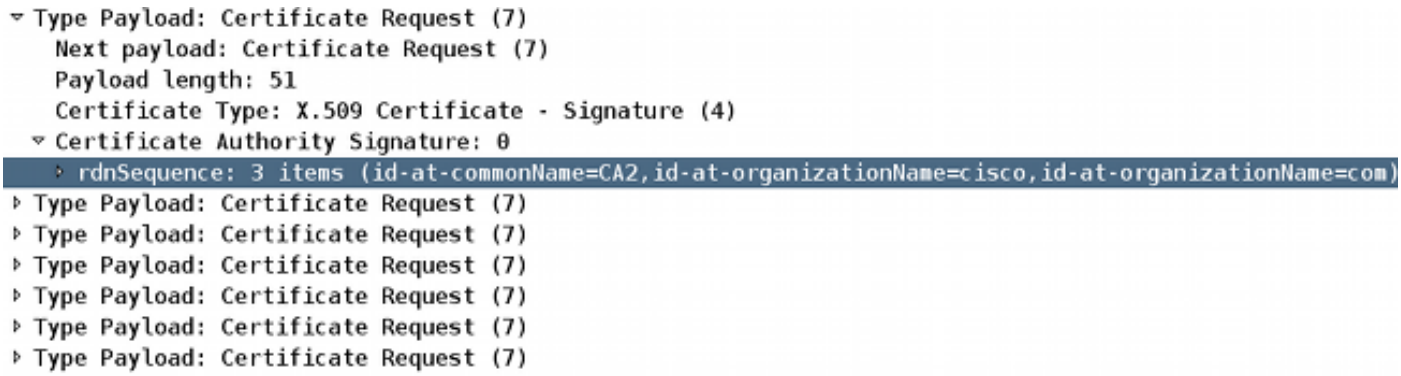
Obwohl R1 für einen einzigen Trust-Point (**IOSCA1**) im ISAKMP-Profil konfiguriert ist, werden mehrere Zertifikatsanforderungen gesendet. Dies liegt daran, dass der Befehl **ca trust-point** im ISAKMP-Profil die Payload der Zertifikatsanforderung bestimmt, jedoch nur, wenn der Router der Initiator der ISAKMP-Sitzung ist. Wenn der Router der Responder ist, gibt es mehrere Payloads für Zertifikatsanfragen für alle global definierten Trust Points, da R1 das für die IKE-Sitzung verwendete ISAKMP-Profil noch nicht kennt.

Die eingehende IKE-Sitzung ist nach dem Empfang des MM5 an ein bestimmtes ISAKMP-Profil gebunden, das die IKE-ID enthält. Anschließend bindet der Befehl **match identity** für das spezifische Profil die IKE-Sitzung an das Profil. Der Router kann dies jedoch bisher nicht feststellen. Es können mehrere ISAKMP-Profile mit unterschiedlichen **CA-Trust-Point-**Befehlen für jedes Profil konfiguriert sein.

Aus diesem Grund muss R1 die Zertifikatsanforderung für alle global konfigurierten Vertrauenspunkte senden.

Weitere Informationen finden Sie in der Befehlsreferenz für den Befehl **ca trust-point**:

**Ein Router, der IKE initiiert, und ein Router, der auf die IKE-Anforderung antwortet, sollten symmetrische Trustpoint-Konfigurationen aufweisen**. Beispielsweise verwendet ein antwortender Router (im IKE-Hauptmodus), der die RSA-Signaturverschlüsselung und -authentifizierung ausführt, möglicherweise Trustpoints, die beim Senden der CERT-REQ-Payloads in der globalen Konfiguration definiert wurden. Der Router kann jedoch eine Liste von vertrauenswürdigen Punkten verwenden, die im ISAKMP-Profil für die Zertifikatsüberprüfung definiert wurden. Wenn der Peer (der IKE-Initiator) so konfiguriert ist, dass er ein Zertifikat verwendet, dessen Vertrauenspunkt in der globalen Liste des antwortenden Routers, jedoch nicht im ISAKMP-Profil des antwortenden Routers enthalten ist, wird das Zertifikat abgelehnt. (Wenn der initiierende Router jedoch keine Informationen zu den Vertrauenspunkten in der globalen Konfiguration des antwortenden Routers erhält, kann das Zertifikat trotzdem authentifiziert werden.)
Überprüfen Sie jetzt die MM4-Paketdetails, um die erste Payload für Zertifikatsanforderung zu ermitteln:




Das MM4-Paket, das von R1 gesendet wird, enthält den **IOSCA2**-Vertrauenspunkt in der ersten Payload für Zertifikatsanforderung, da die Zertifikate in der Reihenfolge installiert sind, in der sie installiert sind. Das erste wird vom **IOSCA2**-Vertrauensbereich signiert:

```
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>
```

Vergleichen Sie das MM3-Paket, das von R2 gesendet wird, wenn der IOSCA1-Vertrauenspunkt in der ersten Zertifikatsanforderungs-Payload enthalten ist:

```
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>
```

Jetzt empfängt R2 das MM4-Paket von R1 und beginnt mit der Verarbeitung der Zertifikatsanforderung. Die erste Zertifikatsanforderungs-Payload entspricht dem IOSCA2-Vertrauenspunkt:

```
*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
```

Wenn R2 das MM5-Paket vorbereitet, wird das Zertifikat verwendet, das dem IOSCA2-Vertrauenspunkt zugeordnet ist:

```
*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.
```

Das MM5-Paket wird von R1 empfangen. Da R1 nur dem IOSCA1-Vertrauenspunkt (für das ISAKMP-Profil prof1) vertraut, schlägt die Zertifikatsvalidierung fehl:

```
*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
```

Diese Konfiguration funktioniert, wenn die Reihenfolge der Zertifikatregistrierung für R1 unterschiedlich ist, da das erste angezeigte Zertifikat vom **IOSCA1**-Vertrauenspunkt signiert wird. Außerdem ist die erste Zertifikatsanforderungs-Payload in MM4 der **IOSCA1**-Vertrauenspunkt, der dann von R2 ausgewählt und auf R1 im MM6 erfolgreich validiert wird.

## IKEv1 ohne *CA-Trust-Point*-Befehl im Profil

Bei Szenarien mit mehreren Profilen und Vertrauenspunkten, aber ohne eine spezifische Trust-Point-Konfiguration in den Profilen gibt es keine Probleme, da keine Validierung bestimmter Vertrauenspunkte erfolgt, die durch eine **CA-Trust-Point-**Befehlskonfiguration bestimmt werden. Der Auswahlprozess ist jedoch möglicherweise nicht offensichtlich. Abhängig vom Router, der der Initiator ist, werden die verschiedenen Zertifikate für den Authentifizierungsprozess in Bezug auf die Reihenfolge der Zertifikatsregistrierung ausgewählt.

Manchmal kann ein Zertifikat nur von einer Seite der Verbindung unterstützt werden, z. B. in x509-Version 1, die keine typische Hash-Funktion ist, die zum Signieren verwendet wird. Der VPN-

Tunnel kann nur von einer Seite der Verbindung aus eingerichtet werden.

## RFC-Referenz für IKEv1

Hier ein Snip aus RFC4945:

3.2.7.1 Festlegen von Zertifizierungsstellen

Beim **Anfordern** des In-Band-Austauschs von Keying-Materialien SOLLTEN Implementierungen CERTREQs für jeden Peer-Trust-Anker generieren, der von der **lokalen Richtlinie** während eines bestimmten Austauschs als vertrauenswürdig angesehen wird.
Die RFC ist nicht klar. Die **lokale Richtlinie** kann sich **explizit** auf den Befehl **ca trust-point** beziehen, der im ISAKMP-Profil für Crypto konfiguriert ist. Das Problem besteht darin, dass Sie in der MM3- und MM4-Phase des Prozesses kein ISAKMP-Profil auswählen können, es sei denn, Sie verwenden eine IP-Adresse für die Identität und die Vertrauenspunkte, da die Authentifizierung in der MM5- und der MM6-Phase des Prozesses zuerst erfolgen muss. Aus diesem Grund bezieht sich die **lokale Richtlinie explizit** auf alle Vertrauenspunkte, die auf dem Gerät konfiguriert sind.

> **Hinweis**: Diese Informationen sind nicht von Cisco spezifisch, sie sind jedoch IKEv1-spezifisch.

## IKEv2-Profilauswahl mit Identitäten, die sich überschneiden

Bevor mehrere Zertifikate für IKEv2 beschrieben werden, ist es wichtig zu wissen, wie die Profile ausgewählt werden, wenn die Übereinstimmungsidentität verwendet wird. Dies ist für alle Profile ausreichend. Dies ist kein empfohlenes Szenario, da die Ergebnisse der IKEv2-Aushandlung von mehreren Faktoren abhängen. Für IKEv1 bestehen dieselben Probleme, wenn Profile verwendet werden, die sich überschneiden.

Nachfolgend finden Sie ein Beispiel für eine IKEv2-Initiatorkonfiguration:

```
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
```

Die Identitätstypadresse wird für beide Seiten der Verbindung verwendet. Authentifizierung über Zertifikate (kann auch Pre-Shared Keys sein) ist in diesem Beispiel nicht wichtig. Der Responder verfügt über mehrere Profile, die alle mit dem eingehenden IKEv2-Datenverkehr übereinstimmen:

```
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
```

Der Initiator sendet das dritte IKEv2-Paket, und der Responder muss das Profil anhand der empfangenen Identität auswählen. Die Identität ist eine IPv4-Adresse (**192.168.0.1**):

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

Alle Profile erfüllen diese Identität aufgrund des konfigurierten Befehls **match identity**. Das IOS wählt die letzte in der Konfiguration aus, die in diesem Beispiel **profile3** ist:

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

Um die Bestellung zu überprüfen, geben Sie den Befehl **show crypto ikev2 profile ein**.

> **Hinweis**: Auch wenn sich im Profil eine generische Adresse (0.0.0.0) befindet, wird sie trotzdem ausgewählt. Das IOS versucht nicht, eine optimale Übereinstimmung zu finden. es versucht, die erste Übereinstimmung zu finden. Dies ist jedoch nur der Fall, weil für alle Profile derselbe **Remote-**Befehl **zur Übereinstimmung der Identität** konfiguriert wurde. Für IKEv1- und IKEv2-Profile mit unterschiedlichen Identitätsregeln wird immer das spezifischste Profil verwendet. Cisco empfiehlt, die Profile nicht mit dem Befehl **für die überlappende Übereinstimmung** konfiguriert zu haben, da es schwierig ist, das ausgewählte Profil vorherzusagen.

In diesem Szenario wird **profile3** vom Responder ausgewählt, **profile1** wird jedoch für die Tunnelschnittstelle verwendet. Dies führt dazu, dass ein Fehler angezeigt wird, wenn die Proxy-ID ausgehandelt wird:

```
*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-Fluss bei Verwendung von Zertifikaten

Wenn Zertifikate für IKEv2 zur Authentifizierung verwendet werden, sendet der Initiator die Payload für die Zertifikatsanforderung im ersten Paket nicht:

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

Der Befragte antwortet mit der Zertifikatsanforderungs-Payload (zweites Paket) und allen CAs, da der Befragte nicht weiß, welches Profil zu diesem Zeitpunkt verwendet werden soll. Das Paket, das die Informationen enthält, wird an den Initiator gesendet:

```
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)
```

Der Initiator verarbeitet das Paket und wählt einen Vertrauenspunkt aus, der der vorgeschlagenen CA entspricht:

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

Der Initiator sendet dann das dritte Paket mit der Zertifikatsanforderung und der ZertifikatsPayload. Dieses Paket ist bereits mit dem Keying-Material der Diffie-Hellman (DH)-Phase verschlüsselt:

```
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)
```

Das vierte Paket wird vom Responder an den Initiator gesendet und enthält nur die Zertifikatsnutzlast:

```
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)
```

Der hier beschriebene Fluss ähnelt dem IKEv1-Fluss. Der Responder muss die Payload für Zertifikatsanforderung vorab senden, ohne dass er Kenntnis von dem zu verwendenden Profil hat. Dies führt zu denselben Problemen, die bereits für IKEv1 beschrieben wurden (aus Protokollsicht). Die IOS-Implementierung ist jedoch für IKEv2 besser als für IKEv1.

## Verbindlicher IKEv2-Vertrauenspunkt für den Initiator

Hier ein Beispiel, wenn ein IKEv2-Initiator versucht, ein Profil mit Zertifikatsauthentifizierung zu verwenden und für dieses Profil kein Vertrauenspunkt konfiguriert wurde:

```
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
```

Das erste Paket wird wie oben beschrieben ohne Payload für Zertifikatsanfragen gesendet. Die Antwort des Responders beinhaltet die Payload der Zertifikatsanforderung für alle im globalen Konfigurationsmodus definierten Vertrauenspunkte. Diese wird vom Initiator empfangen:

```
*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
```

Der Initiator kennt den Vertrauenspunkt nicht, der zum Signieren verwendet werden sollte. Dies ist der Hauptunterschied, wenn die IKEv2-Implementierung mit der IKEv1 verglichen wird. Der IKEv2-Initiator muss über den Vertrauenspunkt verfügen, der im IKEv2-Initiatorprofil konfiguriert wurde.

Dies ist jedoch für den IKEv2-Responder nicht erforderlich.

Es folgt ein Auszug aus der Befehlsreferenz:

Wenn in der Konfiguration des IKEv2-Profils kein Vertrauenspunkt definiert ist, wird standardmäßig **das Zertifikat** unter Verwendung aller Vertrauenspunkte **validiert**, die in der globalen Konfiguration definiert sind.
Es ist möglich, verschiedene Vertrauenspunkte zu definieren. eine, um zu unterzeichnen, und eine andere, um zu validieren. Leider löst der im IKEv2-Profil konfigurierte obligatorische Treuepunkt nicht alle Probleme.

## R2 als IKEv2-Initiator

In diesem Beispiel ist R2 der IKEv2-Initiator:

```
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
```

In diesem Beispiel ist R1 der IKEv2-Responder:

```
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
```

Hier sendet R2 das erste Paket ohne Zertifikatsanforderung. Der Responder antwortet mit einer Zertifikatsanforderung für alle konfigurierten Vertrauenspunkte. Die Reihenfolge der Payloads ähnelt der IKEv1 und hängt von den installierten Zertifikaten ab:

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

Das erste konfigurierte Zertifikat auf R1 ist mit dem **TP2**-Vertrauenspunkt verknüpft, sodass die

erste Zertifikatsanforderungs-Payload für die CA gilt, die dem **TP2**-Vertrauenspunkt zugeordnet ist. R2 wählt sie für die Authentifizierung aus (First Match-Regel):

```
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
```

Anschließend erstellt R2 eine Antwort (Paket 3) mit der Payload der Zertifizierungsanfrage, die dem **TP2** zugeordnet ist. R1 kann dem Zertifikat nicht vertrauen, da es für die Validierung für den **TP1**-Vertrauenspunkt konfiguriert ist:

```
*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)
```

Wie bereits erwähnt, empfiehlt Cisco, unter einem IKEv2-Profil nicht mehrere Vertrauenspunkte zu verwenden. Wenn Sie mehrere Vertrauenspunkte verwenden, muss sichergestellt werden, dass beide Seiten genau denselben Vertrauenspunkten vertrauen. Beispielsweise sind sowohl R1 als auch R2 in ihren Profilen sowohl TP1 als auch TP2 konfiguriert.

# Zusammenfassung

Dieser Abschnitt bietet eine kurze Zusammenfassung der im Dokument beschriebenen Informationen.

Der Inhalt der Zertifikatsanforderung für die Payload hängt von der Konfiguration ab. Wenn ein bestimmter Vertrauenspunkt für das ISAKMP-Profil konfiguriert ist und der Router der Initiator des

ISAKMP ist, enthält die Zertifikatsanforderung in MM3 nur die CA, die dem Vertrauenspunkt zugeordnet ist. Wenn jedoch derselbe Router der ISAKMP-Responder ist, enthält das vom Router gesendete MM4-Paket mehrere Payloads für Zertifikatsanfragen für alle global definierten Vertrauenspunkte (wenn der Befehl **ca trust-point** nicht berücksichtigt wird). Dies liegt daran, dass der ISAKMP-Responder das ISAKMP-Profil bestimmen kann, das nur verwendet werden soll, wenn er das MM5-Protokoll und die Zertifikatsanforderung, die im MM4 enthalten ist, erhält.

Die Payload der Zertifikatsanforderung im MM3 und im MM4 ist aufgrund der Regel für die erste Übereinstimmung wichtig. Die erste Übereinstimmungsregel legt den Vertrauenspunkt fest, der für die Zertifikatauswahl verwendet wird, die für die Authentifizierung in MM5 und MM6 erforderlich ist.

Die Reihenfolge der Payload für Zertifikatsanfragen hängt von der Reihenfolge der installierten Zertifikate ab. Der Aussteller des ersten Zertifikats, das in der Ausgabe des Befehls **show crypto pki certificate** angezeigt wird, wird zuerst gesendet. Dieses erste Zertifikat ist das letzte, das registriert wird.

Es ist möglich, mehrere Vertrauenspunkte für ein ISAKMP-Profil zu konfigurieren. Wenn dies geschieht, gelten weiterhin alle vorherigen Regeln.

Alle in diesem Dokument beschriebenen Probleme und Probleme sind auf das IKEv1-Protokolldesign zurückzuführen. Die Authentifizierungsphase findet im MM5 und im MM6 statt, während die Vorschläge für die Authentifizierung (Zertifikatsanforderungen) zu einem früheren Zeitpunkt (vorab) ohne Kenntnis des ISAKMP-Profils gesendet werden müssen, das verwendet werden soll. Dies ist kein Cisco spezifisches Problem und steht in Zusammenhang mit den Einschränkungen des IKEv1-Protokolldesigns.

Das IKEv2-Protokoll ähnelt dem IKEv1 in Bezug auf den Zertifikatsverhandlungsprozess. Die Implementierung im IOS erfordert jedoch die Verwendung spezifischer Treuepunkte für den Initiator. Dies löst nicht alle Probleme. Wenn mehrere Vertrauenspunkte für ein Profil konfiguriert und ein einzelner Vertrauenspunkt auf der anderen Seite konfiguriert werden, können weiterhin Probleme mit der Authentifizierung auftreten. Cisco empfiehlt, für beide Seiten der Verbindung symmetrische Trust-Point-Konfigurationen zu verwenden (dieselben Vertrauenspunkte, die für beide IKEv2-Profile konfiguriert wurden).

Hier einige wichtige Hinweise zu den in diesem Dokument beschriebenen Informationen:

- Bei asymmetrischen Trust-Point-Konfigurationen für die IKEv1-Profile von Peers kann der Tunnel nur von einer Seite des Tunnels aus initiiert werden. Die Konfiguration des Vertrauenspunkts für das IKEv1-Profil ist optional.

- Bei asymmetrischen Trust-Point-Konfigurationen für die IKEv2-Profile von Peers kann der Tunnel nur von einer Seite des Tunnels aus initiiert werden. Die Konfiguration des Vertrauenspunkts für das IKEv2-Profil ist für den Initiator obligatorisch.

- Die Reihenfolge der Payload-Bestellungen für Zertifikatsanforderungen hängt von der Reihenfolge der Zertifikate ab, die in der Ausgabe des Befehls **show crypto pki certificate** (erste Übereinstimmung) angezeigt werden.

- Die Gehaltsladeleihenfolge der Zertifikatsanforderung bestimmt das Zertifikat, das vom Responder (erste Übereinstimmung) ausgewählt wird.

- Wenn Sie für IKEv1 und IKEv2 mehrere Profile verwenden und dieselben Identitätsregeln konfiguriert haben, ist es schwierig, die Ergebnisse vorherzusagen (zu viele Faktoren sind involviert).

- Cisco empfiehlt die Verwendung symmetrischer Trust-Point-Konfigurationen sowohl für IKEv1 als auch für IKEv2.

## Zugehörige Informationen

- Konfigurationsleitfaden für Internet Key Exchange for IPsec VPNs, Cisco IOS Release 15M&T - Zertifikat für ISAKMP-Profilzuordnung
- Cisco IOS Security Command Reference: Befehle A bis C - können durch eindeutige Eingabe einen Vertrauensvorschuss darstellen
- Technischer Support und Dokumentation - Cisco Systems