Processi di scambio pacchetti IOS IKEv1 e IKEv2 per profili con più certificati

Sommario

Introduzione

In questo documento vengono descritti i processi di scambio dei pacchetti IKEv1 (Internet Key Exchange versione 1) e IKEv2 (Internet Key Exchange versione 2) quando si utilizza l'autenticazione dei certificati e i possibili problemi.

Di seguito è riportato un elenco degli argomenti descritti nel presente documento:

- Criteri di selezione del certificato per l'iniziatore IKE (Internet Key Exchange) e il risponditore IKE
- I criteri di corrispondenza del profilo IKE quando vengono trovati più profili IKE (per scenari di sovrapposizione e non di sovrapposizione)
- Impostazioni e comportamento predefiniti quando non vengono utilizzati trust point nei profili IKE
- Differenze tra IKEv1 e IKEv2 per quanto riguarda i criteri di selezione dei profili e dei certificati

Nota: Per ulteriori informazioni sulla risoluzione di un problema specifico, consultare la sezione corretta. Inoltre, alla fine del presente documento viene fornito un breve riepilogo.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Configurazione VPN Cisco IOS®
- Protocolli IKEv1 e IKEv2 (scambio pacchetti)

Componenti usati

Il riferimento delle informazioni contenute in questo documento è Cisco IOS versione 15.3T.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

I problemi descritti in questo documento si verificano quando si utilizzano più trust point e più profili IKE.

Gli esempi iniziali utilizzati in questo documento hanno un tunnel LAN-LAN IKEv1 con due trust point su ciascun router. A prima vista, potrebbe sembrare che la configurazione sia corretta. Tuttavia, il tunnel VPN può essere avviato solo da un lato della connessione perché il comando **ca trust-point** viene utilizzato per il comportamento del profilo ISAKMP (Internet Security Association and Key Management Protocol) e per l'ordine dei certificati registrati nell'archivio locale.

Quando il router è l'iniziatore ISAKMP, il comando **ca trust-point** per il profilo ISAKMP prevede un comportamento diverso. È possibile che si sia verificato un problema perché l'iniziatore ISAKMP riconosce il profilo ISAKMP dall'inizio, quindi il comando **ca trust-point** configurato per il profilo può influenzare il payload per la richiesta di certificato nel pacchetto 3 in modalità principale (MM3). Tuttavia, quando il router è il risponditore ISAKMP, associa il traffico in entrata a un profilo ISAKMP specifico dopo aver ricevuto il pacchetto 5 in modalità principale (MM5), che include l'ID IKE necessario per creare il binding. Per questo motivo non è possibile applicare alcun comando **ca trust-point** per il pacchetto Modalità principale 4 (MM4) perché il profilo non viene determinato prima di M5.

In questo documento viene spiegato l'ordine del payload della richiesta di certificato in MM3 e MM4 e l'impatto sull'intero processo di negoziazione, nonché il motivo per cui consente di stabilire la connessione solo da un lato del tunnel VPN.

Di seguito è riportato un riepilogo dei comportamenti dell'iniziatore IKEv1 e del risponditore:

	Iniziatore IKEv1	Risponditore IKEv1
Invia richiesta	Invia richieste specifiche solo per i trust point configurati nel profilo	Invia richieste per tutti i trust point disponibili
Convalida richiesta	Esegue la convalida in base a trust point specifici configurati nel profilo	Esegue la convalida in base a trust point specifici configurati nel profilo

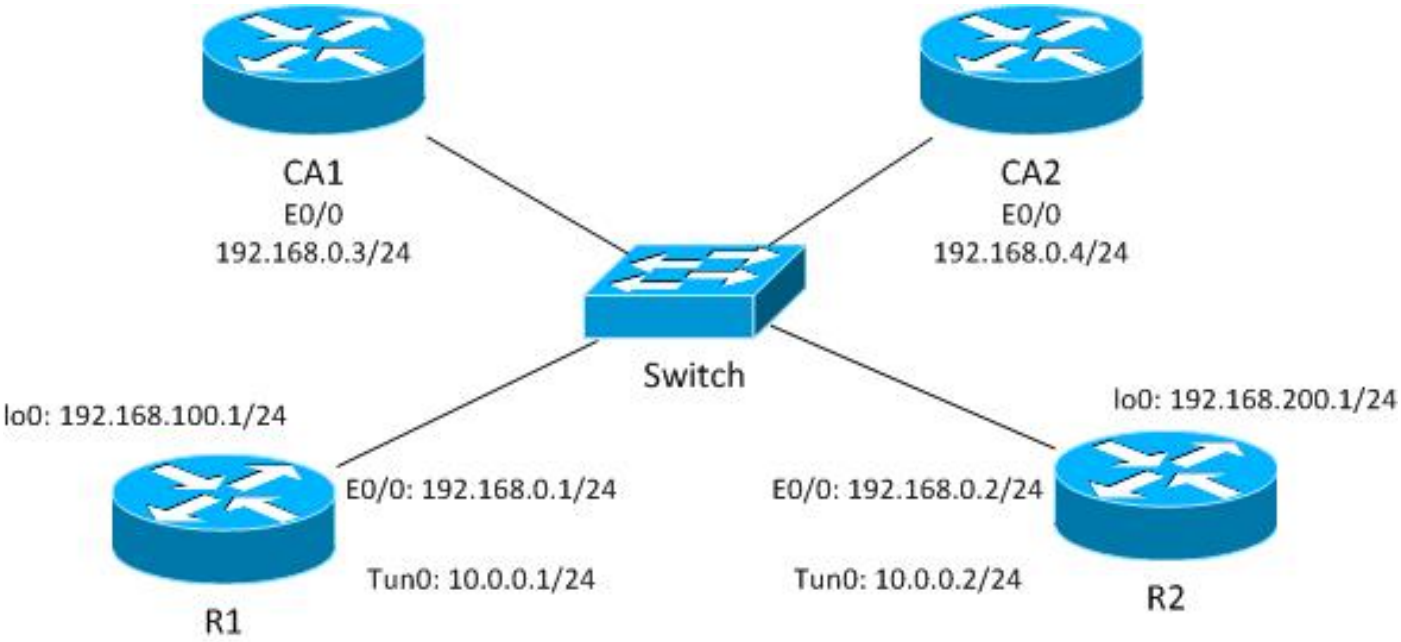
Cisco consiglia di non utilizzare il comando **ca trust-point** per i risponditori ISAKMP con più profili ISAKMP e che utilizzano trust-point configurati globalmente. Per gli iniziatori ISAKMP con più profili ISAKMP, Cisco consiglia di limitare il processo di selezione dei certificati con il comando **ca trust-point** in ciascun profilo.

Il protocollo IKEv2 presenta gli stessi problemi del protocollo IKEv1, ma il diverso comportamento del comando **pki trustpoint** consente di evitare il verificarsi dei problemi. Infatti, il comando **pki trustpoint** è obbligatorio per l'iniziatore IKEv2, mentre il comando **ca trust-point** è facoltativo per l'iniziatore IKEv1. In alcune circostanze (più trust point in un unico profilo) potrebbero verificarsi i problemi descritti in precedenza. Per questo motivo, Cisco consiglia di utilizzare configurazioni di trust point simmetriche per entrambi i lati della connessione (gli stessi trust point configurati in entrambi i profili IKEv2).

Topologia

Questa è una topologia generica utilizzata per tutti gli esempi riportati nel presente documento.

> **Nota:** Il router 1 (R1) e il router 2 (R2) utilizzano le interfacce tunnel virtuali (VTI) per accedere ai loopback. Queste VTI sono protette da IPSec.



Nell'esempio di IKEv1, ogni router dispone di due trust point per ogni Autorità di certificazione (CA) e vengono registrati i certificati per ogni trust point.

Quando R1 è l'iniziatore ISAKMP, il tunnel negozia correttamente e il traffico è protetto. Si tratta di un comportamento normale. Quando R2 è l'iniziatore ISAKMP, la negoziazione Phase1 non riesce.

> **Nota:** Per gli esempi di IKEv2 illustrati in questo documento, la topologia e l'indirizzamento sono gli stessi dell'esempio IKEv1.

Processo Packet Exchange

In questa sezione vengono descritte le variazioni di configurazione IKEv1 e IKEv2 utilizzate per il processo di scambio dei pacchetti e i possibili problemi.

IKEv1 con più certificati

Ecco la configurazione della rete R1 e della VPN per IKEv1 con più certificati:

```
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
```

Ecco la configurazione della rete R2 e della VPN per IKEv1 con più certificati:

```
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 questo esempio, R1 ha due trust-point: uno usa **IOSCA1** e l'altro usa **IOSCA2**:

```
crypto pki trustpoint IOSCA1
 enrollment url http://192.168.0.3:80
 serial-number
 fqdn R1.cisco.com
 ip-address 192.168.0.1
 subject-name CN=R1,OU=IT,O=cisco,O=com
 revocation-check crl
!
crypto pki trustpoint IOSCA2
 enrollment url http://192.168.0.4:80
 serial-number
 fqdn R1.cisco.com
 ip-address 192.168.0.1
 subject-name CN=R1,OU=IT,O=cisco,O=com
 revocation-check crl
```

In questo esempio, R2 include anche due trust point: uno usa **IOSCA1** e l'altro usa **IOSCA2**:

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

Èimportante notare l'unica differenza in queste configurazioni: il profilo ISAKMP R1 utilizza il comando **ca trust-point** per il trust-point **IOSCA1**, che indica che R1 considera attendibili solo i certificati convalidati da quel trust-point specifico. R2, invece, considera attendibili tutti i certificati convalidati da tutti i trust point definiti a livello globale.

R1 come iniziatore IKEv1

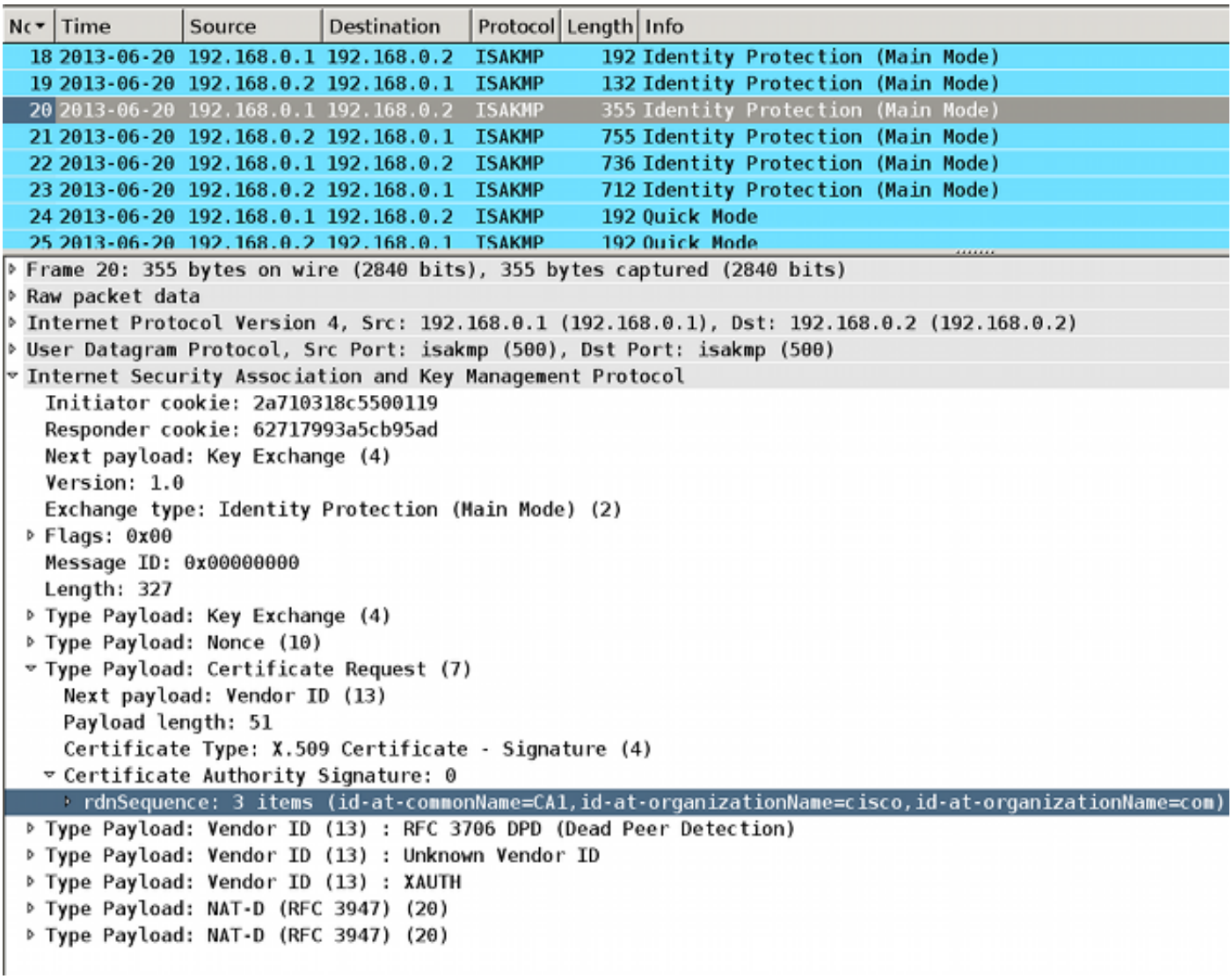
Di seguito sono riportati i comandi di debug per R1 e R2:

- **Disakmp crittografia debug R1#**
- **R1# debug crypto ipsec**
- **Convalida crittografia di debug R1#**

In questo caso, R1 avvia il tunnel e invia il certificato che richiede l'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
```

Èimportante notare che il pacchetto contiene una sola richiesta di certificato, che è solo per il trust point **IOSCA1**. Questo comportamento è previsto con la configurazione corrente del profilo ISAKMP (**CN=CA1, O=cisco, O=com**). Non vengono inviate altre richieste di certificati, che è possibile verificare con la funzionalità Embedded Packet Capture:



Quando R2 riceve il pacchetto, inizia a elaborare la richiesta di certificato, creando una corrispondenza che determina il trust-point e il certificato associato utilizzato per l'autenticazione in MM5. L'ordine di elaborazione è lo stesso del payload della richiesta di certificato nel pacchetto ISAKMP. Ciò significa che viene utilizzata la prima corrispondenza. In questo scenario esiste una sola corrispondenza poiché R1 è configurato con un trust point specifico e invia una sola richiesta di certificato associata al trust point.

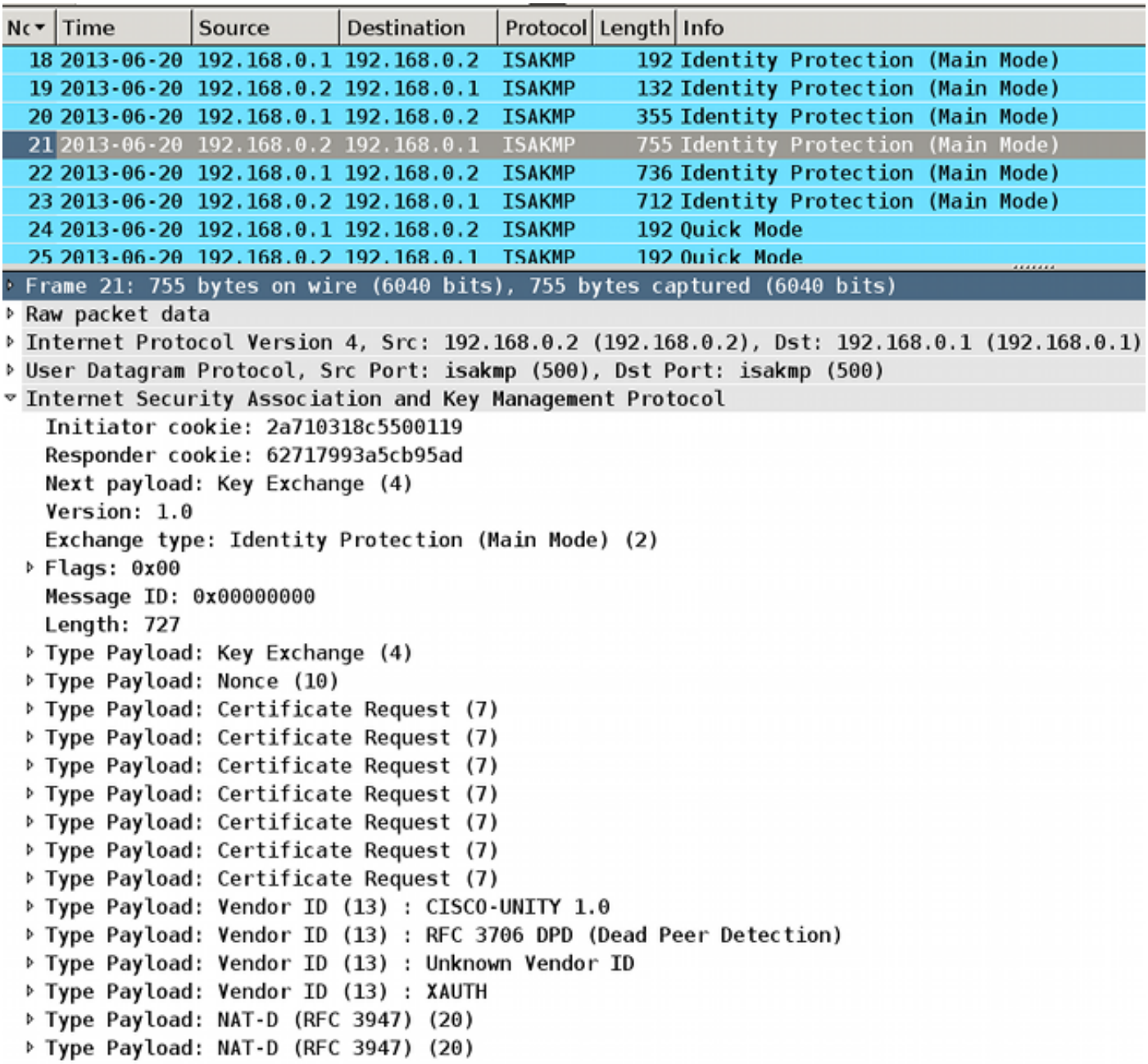
```
*Jun 20 13:00:37.617: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
```

```
*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
```

In seguito R2 prepara M4. Si tratta del pacchetto che contiene la richiesta di certificato per tutti i trust point trusted. Poiché R2 è il risponditore ISAKMP, tutti i trust point definiti a livello globale sono attendibili (la configurazione dei **trust point CA** non viene controllata). Due dei trust point sono definiti manualmente (**IOSCA1** e **IOSCA2**), mentre gli altri sono predefiniti.

```
*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
```

Èpossibile verificare il pacchetto con Wireshark. Il pacchetto M4 di R2 contiene sette voci di richiesta certificato:



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

R1 riceve quindi MM4 da R2 con più campi di richiesta certificato:

```
*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
```

La prima regola di corrispondenza in R1 corrisponde alla prima richiesta di certificato con il trust point **IOSCA1**. Determina che R1 utilizza il certificato associato al trust point **IOSCA1** per l'autenticazione in MM5. Come ID IKE viene utilizzato il nome di dominio completo (FQDN). Ciò è dovuto alla configurazione dell'**FQDN** dell'**identità** nel profilo ISAKMP:

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

L'MM5 viene ricevuto ed elaborato da R2. L'ID IKE ricevuto (**R1.cisco.com**) corrisponde al profilo ISAKMP **prof1**. Il certificato ricevuto viene quindi convalidato e l'autenticazione ha esito positivo:

```
*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
```

R2 prepara quindi il modello M6 con il certificato associato a **IOSCA1**:

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

Il pacchetto viene ricevuto da R1, che verifica il certificato e l'autenticazione:

```
*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
```

La fase 1 è completata. La fase 2 è negoziata come di consueto. Il tunnel è stato stabilito correttamente e il traffico è protetto.

R2 come iniziatore IKEv1

In questo esempio viene descritto il processo in cui R2 avvia lo stesso tunnel IKEv1 e viene spiegato perché non viene stabilito.

> **Nota:** Parti dei registri vengono rimosse per evidenziare solo le differenze rispetto all'esempio illustrato nella sezione precedente.

R2 invia il modulo M3 con sette payload di richieste di certificati perché R2 non ha un trust point associato al profilo ISAKMP (tutti i trust point sono attendibili):

```
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer cn=CA1,o=cisco,o=com
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer cn=CA2,o=cisco,o=com
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer ou=Class 3 Public Primary Certification Authority,
 o=VeriSign, Inc.,c=US
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer cn=Cisco SSCA2,o=Cisco Systems
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer cn=Cisco Manufacturing CA,o=Cisco Systems
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
```

```
 issuer cn=Cisco Root CA 2048,o=Cisco Systems
*Jun 17 18:08:44.321: ISAKMP (0): constructing CERT_REQ for
 issuer cn=Cisco Root CA M1,o=Cisco
*Jun 17 18:08:44.321: ISAKMP (0): sending packet to 192.168.0.1
 my_port 500 peer_port 500 (I) MM_SA_SETUP
```

Quando R1 riceve il pacchetto da R2, elabora la richiesta di certificato e corrisponde al trust point IOSCA1, che determina il certificato inviato in MM6:

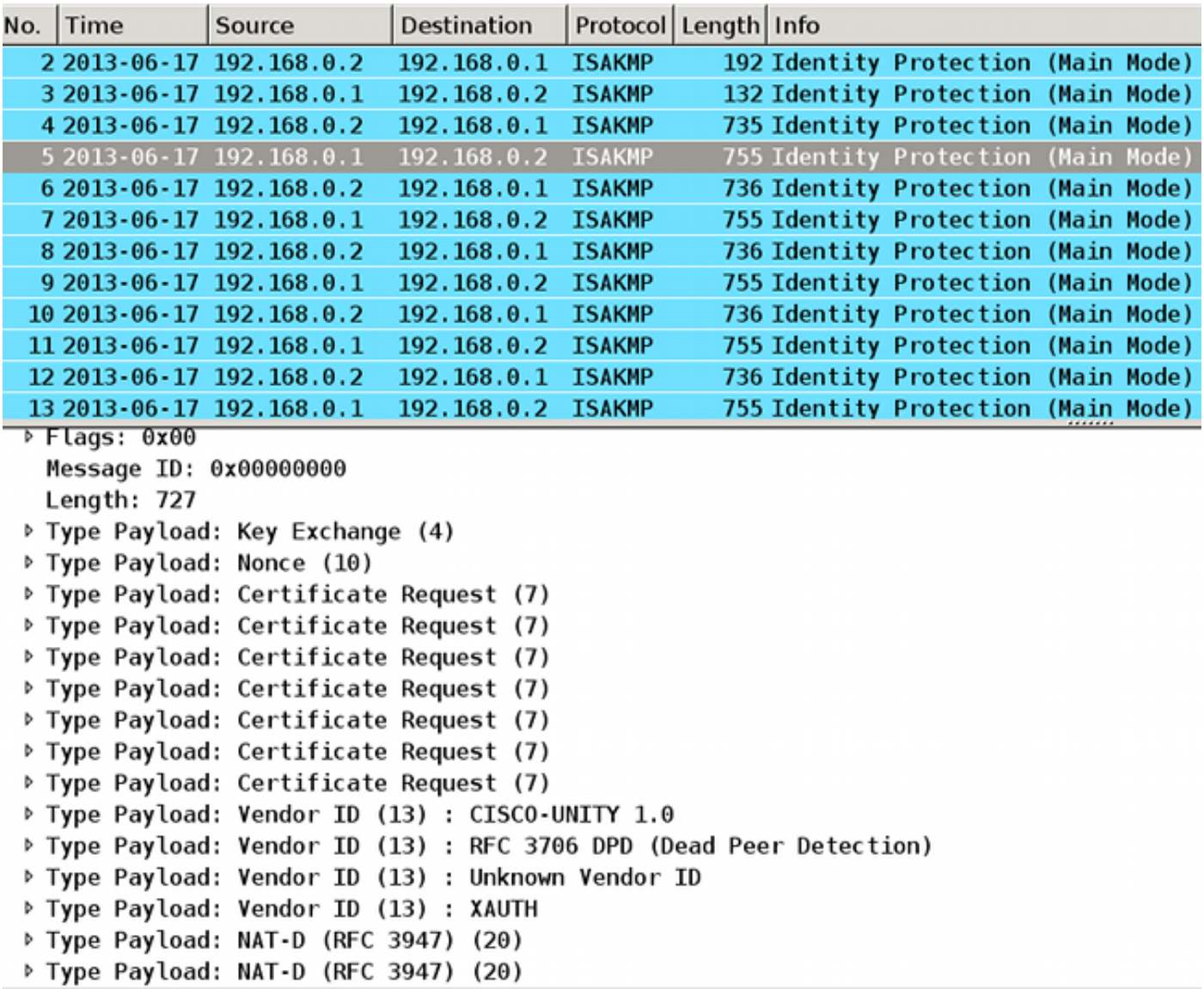
```
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=CA1,o=cisco,o=com
*Jun 17 18:08:14.321:  Choosing trustpoint IOSCA1 as issuer
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=CA2,o=cisco,o=com
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 ou=Class 3 Public Primary Certification Authority,o=VeriSign, Inc.,c=US
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=Cisco SSCA2,o=Cisco Systems
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=Cisco Manufacturing CA,o=Cisco Systems
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=Cisco Root CA 2048,o=Cisco Systems
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=Cisco Root CA M1,o=Cisco
```

Successivamente, R1 prepara il pacchetto M4 con il payload della richiesta di certificato. A questo punto sono disponibili più payload di richiesta di certificato:

```
*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
```

Verificare i registri con Embedded Packet Capture (EPC) e Wireshark:



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

Anche se R1 è configurato per un singolo trust point (**IOSCA1**) nel profilo ISAKMP, vengono inviate più richieste di certificati. Questo si verifica perché il payload della richiesta di certificato è determinato dal comando **ca trust-point** nel profilo ISAKMP, ma solo quando il router è l'iniziatore della sessione ISAKMP. Se il router è il risponditore, vi sono più payload di richieste di certificati per tutti i trust point definiti a livello globale perché R1 non conosce ancora il profilo ISAKMP utilizzato per la sessione IKE.

La sessione IKE in entrata è associata a un profilo ISAKMP specifico dopo la ricezione dell'MM5, che include l'ID IKE. Successivamente, il comando **match identity** per il profilo specifico associa la

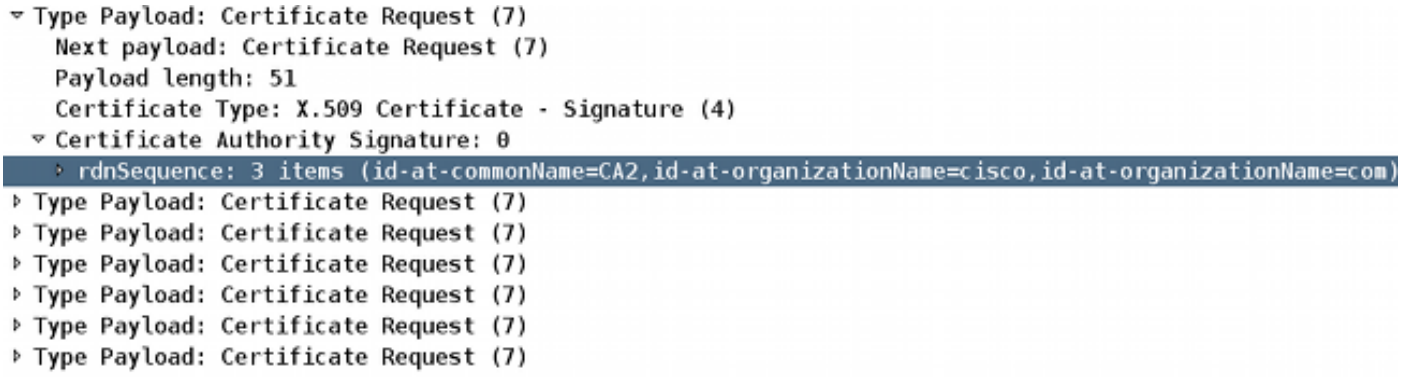
sessione IKE al profilo. Tuttavia, il router non può determinarlo fino a questo momento. Per ogni profilo possono essere configurati più profili ISAKMP con comandi **ca trust-point** diversi.

Per questo motivo, R1 deve inviare la richiesta di certificato per tutti i trust point configurati globalmente.

Fare riferimento alla guida di riferimento del comando **ca trust-point**:

Un router che avvia IKE e un router che risponde alla richiesta IKE devono avere configurazioni di trust point simmetriche. Ad esempio, un router rispondente (in modalità principale IKE) che esegue la crittografia e l'autenticazione della firma RSA potrebbe utilizzare i trust definiti nella configurazione globale durante l'invio dei payload CERT-REQ. Tuttavia, il router potrebbe utilizzare un elenco limitato di trust definiti nel profilo ISAKMP per la verifica del certificato. Se il peer (l'iniziatore IKE) è configurato per utilizzare un certificato il cui punto di attendibilità si trova nell'elenco globale del router rispondente ma non nel profilo ISAKMP del router rispondente, il certificato viene rifiutato. Tuttavia, se il router che avvia la procedura non è a conoscenza dei trust point nella configurazione globale del router che risponde, è comunque possibile autenticare il certificato.

Verificare ora i dettagli del pacchetto MM4 per individuare il primo payload della richiesta di certificato:



Il pacchetto MM4 inviato da R1 include il trust point **IOSCA2** nel primo payload della richiesta di certificato a causa dell'ordine in cui sono installati i certificati; la prima è firmata dal trust point **IOSCA2**:

```
R1#sh crypto pki certificates
Certificate
 Status: Available
 Certificate Serial Number (hex): 03
 Certificate Usage: General Purpose
 Issuer:
    cn=CA2
   o=cisco
   o=com
 Subject:
   Name: R1.cisco.com
   IP Address: 192.168.0.1
   Serial Number: 100
   serialNumber=100+ipaddress=192.168.0.1+hostname=R1.cisco.com
   cn=R1
   ou=IT
```

```
   o=cisco
   o=com
 Validity Date:
   start date: 13:25:01 CET Jun 17 2013
   end   date: 13:25:01 CET Jun 17 2014
 Associated Trustpoints: IOSCA2
...
<output omitted, 1 more R1 cert signed by CA1, 2 more CA certs>
```

Confrontare il pacchetto M3 inviato da R2 quando il trust point **IOSCA1** è incluso nel primo payload della richiesta di certificato:

```
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>
```

Ora R2 riceve il pacchetto MM4 da R1 e inizia a elaborare la richiesta di certificato. Il primo payload della richiesta di certificato corrisponde al trust point **IOSCA2**:

```
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=CA2,o=cisco,o=com
*Jun 17 18:08:44.335:  Choosing trustpoint IOSCA2 as issuer
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=CA1,o=cisco,o=com
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 ou=Class 3 Public Primary Certification Authority,o=VeriSign, Inc.,c=US
```

```
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=Cisco SSCA2,o=Cisco Systems
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=Cisco Manufacturing CA,o=Cisco Systems
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=Cisco Root CA 2048,o=Cisco Systems
*Jun 17 18:08:44.335: ISAKMP:(1100): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:44.335: ISAKMP:(1100): peer wants cert issued by
 cn=Cisco Root CA M1,o=Cisco
```

Quando R2 prepara il pacchetto M5, utilizza il certificato associato al trust point IOSCA2:

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

Il pacchetto MM5 viene ricevuto da R1. Poiché R1 considera attendibile solo il trust point IOSCA1 (per il profilo ISAKMP prof1), la convalida del certificato ha esito negativo:

```
*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
```

Questa configurazione funziona se l'ordine di registrazione del certificato in R1 è diverso in quanto il primo certificato visualizzato è firmato dal trust point **IOSCA1**. Inoltre, il primo payload della richiesta di certificato in MM4 è il trust point **IOSCA1**, che viene quindi scelto da R2 e convalidato correttamente su R1 in MM6.

IKEv1 senza un comando *ca trust-point* nel profilo

Per gli scenari con più profili e trust point ma senza una configurazione di trust point specifica nei profili, non vi sono problemi perché non esiste una convalida di trust point specifici determinata da una configurazione del comando **ca trust-point**. Tuttavia, il processo di selezione potrebbe non essere ovvio. A seconda del router che funge da iniziatore, i diversi certificati vengono selezionati per il processo di autenticazione in relazione all'ordine di registrazione del certificato.

A volte un certificato può essere supportato solo da un lato della connessione, come in x509 versione 1, che non è una tipica funzione hash utilizzata per firmare. Il tunnel VPN potrebbe essere stabilito solo da un lato della connessione.

Riferimento RFC per IKEv1

Di seguito viene riportato un elemento di cattura relativo alla RFC4945:

3.2.7.1. Specifica delle autorità di certificazione

Quando si **richiede** lo scambio in banda del materiale per le chiavi, le implementazioni DEVONO generare CERTREQ per ogni peer trust anchor che il **criterio locale** considera **esplicitamente** attendibile durante un determinato scambio.
RFC non è chiaro. Il **criterio locale** potrebbe essere correlato in **modo esplicito** al comando **ca trust-point** configurato nel profilo ISAKMP di crittografia. Il problema è che nella fase MM3 e MM4 del processo, non è possibile selezionare un profilo ISAKMP a meno che non si utilizzi un indirizzo IP per l'identità e i trust-point, perché l'autenticazione nella fase MM5 e MM6 del processo deve avvenire prima. Per questo motivo, i **criteri locali** fanno riferimento **esplicitamente** a tutti i trust point configurati nel dispositivo.

Nota: Queste informazioni non sono specifiche di Cisco, ma sono specifiche di IKEv1.

Selezione profilo IKEv2 con identità sovrapposte

Prima di descrivere più certificati per IKEv2, è importante conoscere il modo in cui i profili vengono selezionati quando viene utilizzata l'identità di corrispondenza, che è soddisfatta per tutti i profili. Si tratta di uno scenario non consigliato perché i risultati della negoziazione IKEv2 dipendono da più fattori. Gli stessi problemi si verificano per IKEv1 quando si utilizzano profili che si sovrappongono.

Di seguito è riportato un esempio di configurazione dell'iniziatore IKEv2:

```
crypto ikev2 proposal prop-1
 encryption 3des
 integrity md5
 group 2
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 profile profile1
 match identity remote address 192.168.0.2 255.255.255.255
 identity local address 192.168.0.1
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint TP1

crypto ipsec transform-set trans esp-3des esp-sha-hmac
 mode tunnel
!
crypto ipsec profile profile1
 set transform-set trans
 set ikev2-profile profile1
!
interface Loopback0
 ip address 192.168.100.1 255.255.255.255
```

```
!
interface Tunnel1
 ip address 10.0.0.1 255.255.255.0
 tunnel source Ethernet0/0
 tunnel destination 192.168.0.2
 tunnel protection ipsec profile profile1
!
interface Ethernet0/0
 ip address 192.168.0.1 255.255.255.0

ip route 192.168.200.1 255.255.255.255 10.0.0.2
```

L'indirizzo del tipo di identità viene utilizzato per entrambi i lati della connessione. L'autenticazione tramite certificati (che può anche essere una chiave già condivisa) non è importante per questo esempio. Il risponditore dispone di più profili che corrispondono tutti al traffico IKEv2 in entrata:

```
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
```

L'iniziatore invia il terzo pacchetto IKEv2 e il risponditore deve scegliere il profilo in base all'identità ricevuta. L'identità è un indirizzo IPv4 (**192.168.0.1**):

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

Tutti i profili soddisfano questa identità a causa del comando **match identity** configurato. Il sistema operativo IOS sceglie l'ultimo nella configurazione, ovvero **profile3** nell'esempio seguente:

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

Per verificare l'ordine, immettere il comando **show crypto ikev2 profile**.

> **Nota:** Anche se nel profilo è presente un indirizzo generico (0.0.0.0), è ancora selezionato. IOS non cerca di trovare una corrispondenza migliore; cerca di trovare la prima corrispondenza. Tuttavia, questo si verifica solo perché tutti i profili hanno lo stesso comando **remoto match identity** configurato. Per i profili IKEv1 e IKEv2 con regole di identità di corrispondenza diverse, viene sempre utilizzato il profilo più specifico. Cisco consiglia di non configurare i profili con il comando **overlapping match identity** perché è difficile prevedere il profilo selezionato.

In questo scenario, il responder seleziona **profile3**, ma per l'interfaccia del tunnel viene utilizzato **profile1**. In questo modo viene visualizzato un errore quando l'ID proxy viene negoziato:

```
*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
```

Flusso IKEv2 quando vengono utilizzati i certificati

Quando per l'autenticazione vengono utilizzati certificati per IKEv2, l'iniziatore non invia il payload della richiesta di certificato nel primo pacchetto:

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

Il risponditore risponde con il payload della richiesta di certificato (secondo pacchetto) e tutte le CA perché non è a conoscenza del profilo da utilizzare in questa fase. Il pacchetto contenente le informazioni viene inviato all'iniziatore:

```
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)
```

L'iniziatore elabora il pacchetto e sceglie un trust point corrispondente alla CA proposta:

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

L'iniziatore invia quindi il terzo pacchetto con la richiesta di certificato e il payload del certificato. Questo pacchetto è già crittografato con il materiale per le chiavi della fase Diffie-Hellman (DH):

```
IKEv2:(SA ID = 1):Building packet for encryption.
Payload contents:
 VID IDi CERT CERTREQ NOTIFY(HTTP_CERT_LOOKUP_SUPPORTED) AUTH CFG SA TSi
 TSr NOTIFY(INITIAL_CONTACT) NOTIFY(SET_WINDOW_SIZE) NOTIFY(ESP_TFC_NO_SUPPORT)
 NOTIFY(NON_FIRST_FRAGS)
```

Il quarto pacchetto viene inviato dal risponditore all'iniziatore e contiene solo il payload del certificato:

```
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)
```

Il flusso descritto di seguito è simile al flusso IKEv1. Il risponditore deve inviare il payload della richiesta di certificato in anticipo senza conoscere il profilo da utilizzare, creando gli stessi problemi descritti in precedenza per IKEv1 (dal punto di vista del protocollo). Tuttavia, l'implementazione sul sistema operativo IKEv2 è migliore rispetto a IKEv1.

Trust point obbligatorio IKEv2 per l'iniziatore

Di seguito è riportato un esempio di quando un iniziatore IKEv2 tenta di utilizzare un profilo con l'autenticazione del certificato e non dispone di un trust point configurato in tale profilo:

```
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
```

Il primo pacchetto viene inviato senza alcun payload della richiesta di certificato, come descritto in precedenza. La risposta del risponditore include il payload della richiesta di certificato per tutti i trust point definiti nella modalità di configurazione globale. L'iniziatore riceve il messaggio:

```
*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
```

L'iniziatore non conosce il trust point da utilizzare per firmare. Questa è la differenza principale quando si confronta l'implementazione IKEv2 con IKEv1. L'iniziatore IKEv2 deve avere il trust point configurato nel profilo dell'iniziatore IKEv2, ma non è necessario per il risponditore IKEv2.

Di seguito viene riportato un estratto della guida di riferimento del comando:

Se nella configurazione del profilo IKEv2 non è definito alcun trust point, per impostazione predefinita il **certificato viene convalidato** utilizzando tutti i trust point definiti nella configurazione globale
Èpossibile definire diversi trust point; uno per firmare e uno diverso per convalidare. Il trust point obbligatorio configurato nel profilo IKEv2 non risolve tutti i problemi.

R2 come iniziatore IKEv2

Nell'esempio, R2 è l'iniziatore IKEv2:

```
crypto ikev2 profile profile1
 match identity remote address 192.168.0.1 255.255.255.255
 identity local address 192.168.0.2
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint TP1
 pki trustpoint TP2
```

Nell'esempio, R1 è il responder IKEv2:

```
crypto ikev2 profile profile1
 match identity remote address 192.168.0.2 255.255.255.255
 identity local address 192.168.0.1
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint TP1
```

In questo caso, R2 invia il primo pacchetto senza alcuna richiesta di certificato. Il risponditore risponde con una richiesta di certificato per tutti i trust point configurati. L'ordine dei payload è simile a quello di IKEv1 e dipende dai certificati installati:

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

Il primo certificato configurato in R1 è associato al trust point **TP2**, quindi il primo payload della richiesta di certificato è per la CA associata al trust point **TP2**. Pertanto, R2 la seleziona per l'autenticazione (prima regola di corrispondenza):

```
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
```

R2 prepara quindi una risposta (pacchetto 3) con il payload della richiesta di certificazione associato a **TP2**. R1 non può considerare attendibile il certificato perché è configurato per la convalida rispetto al trust point **TP1**:

```
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):[IKEv2 -> PKI] Retrieving trustpoint(s)
 from received certificate hash(es)
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):[PKI -> IKEv2] Retrieved
 trustpoint(s): 'TP1'
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):[IKEv2 -> PKI] Getting cert chain for
 the trustpoint TP1
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):[PKI -> IKEv2] Getting of cert chain
 for the trustpoint PASSED
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):Get peer's authentication method
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):Peer's authentication method is 'RSA'
*Jul 17 18:09:04.550: IKEv2:(SA ID = 1):[IKEv2 -> PKI] Validating
 certificate chain
*Jul 17 18:09:04.554: IKEv2:(SA ID = 1):[PKI -> IKEv2] Validation of certificate
 chain FAILED
*Jul 17 18:09:04.554: IKEv2:(SA ID = 1):Verification of peer's authentication
 data FAILED
*Jul 17 18:09:04.554: IKEv2:(SA ID = 1):Sending authentication failure notify
*Jul 17 18:09:04.554: IKEv2:(SA ID = 1):Building packet for encryption.
Payload contents:
 NOTIFY(AUTHENTICATION_FAILED)
```

Come accennato in precedenza, Cisco consiglia di non utilizzare più trust point in un unico profilo IKEv2. Quando si utilizzano più trust point, è necessario assicurarsi che entrambi i lati considerino attendibili esattamente gli stessi trust point. Ad esempio, sia R1 che R2 hanno entrambi TP1 e TP2 configurati nei loro profili.

Riepilogo

In questa sezione viene fornito un breve riepilogo delle informazioni descritte nel documento.

Il contenuto del payload della richiesta di certificato dipende dalla configurazione. Se per il profilo ISAKMP è configurato un trust point specifico e il router è l'iniziatore ISAKMP, la richiesta di certificato in MM3 conterrà solo la CA associata al trust point. Tuttavia, se lo stesso router è il risponditore ISAKMP, il pacchetto M4 inviato dal router include più payload di richiesta certificato per tutti i trust point definiti a livello globale (quando il comando **ca trust-point** non viene preso in considerazione). Questo si verifica perché il risponditore ISAKMP può determinare il profilo ISAKMP da utilizzare solo dopo aver ricevuto la richiesta MM5 e la richiesta di certificato inclusa in MM4.

Il payload della richiesta di certificato in MM3 e MM4 è importante a causa della prima regola di corrispondenza. La prima regola di corrispondenza determina il trust-point utilizzato per la selezione del certificato, necessario per l'autenticazione in MM5 e MM6.

L'ordine del payload della richiesta di certificato dipende dall'ordine dei certificati installati. L'autorità di certificazione del primo certificato visualizzato nell'output del comando **show crypto pki certificate** viene inviata per prima. Il primo certificato è l'ultimo registrato.

Èpossibile configurare più trust point per un profilo ISAKMP. Se si esegue questa operazione, verranno comunque applicate tutte le regole precedenti.

Tutti i problemi e gli avvertimenti descritti in questo documento sono dovuti alla progettazione del

protocollo IKEv1. La fase di autenticazione avviene nella MM5 e nella MM6, mentre le proposte per l'autenticazione (richieste di certificati) devono essere inviate in una fase precedente (in anticipo) senza conoscere il profilo ISAKMP da utilizzare. Questo non è un problema specifico di Cisco e si riferisce alle limitazioni della progettazione del protocollo IKEv1.

Il protocollo IKEv2 è simile a IKEv1 per quanto riguarda il processo di negoziazione dei certificati. Tuttavia, l'implementazione nel sistema operativo IOS impone l'uso di trust point specifici per l'iniziatore. Questo non risolve tutti i problemi. Quando per un singolo profilo sono configurati più trust point e dall'altro lato è configurato un singolo trust point, è possibile che si verifichino problemi di autenticazione. Cisco consiglia di utilizzare configurazioni di trust point simmetriche per entrambi i lati della connessione (gli stessi trust point configurati per entrambi i profili IKEv2).

Di seguito sono riportate alcune note importanti sulle informazioni descritte nel presente documento:

- Con le configurazioni di trust point asimmetrici per i profili IKEv1 dei peer, il tunnel potrebbe essere avviato da un solo lato del tunnel. La configurazione del trust point per il profilo IKEv1 è facoltativa.
- Con le configurazioni di trust point asimmetrici per i profili IKEv2 dei peer, il tunnel potrebbe essere avviato da un solo lato del tunnel. La configurazione del trust point per il profilo IKEv2 è obbligatoria per l'iniziatore.
- L'ordine di payload della richiesta di certificato dipende dall'ordine dei certificati visualizzati nell'output del comando **show crypto pki certificate** (prima corrispondenza).
- L'ordine del payload della richiesta di certificato determina il certificato selezionato dal risponditore (prima corrispondenza).
- Quando si utilizzano più profili per IKEv1 e IKEv2 e si configurano le stesse regole di identità di corrispondenza, è difficile prevedere i risultati (troppi fattori coinvolti).
- Cisco consiglia di utilizzare configurazioni di trust point simmetriche sia per IKEv1 che per IKEv2.

Informazioni correlate

- Guida alla configurazione delle VPN Internet Key Exchange for IPsec, Cisco IOS release 15M&T - Mappatura certificato a profilo ISAKMP
- Guida di riferimento ai comandi di Cisco IOS Security: Comandi da A a C - ca trust-point tramite clear eou
- Documentazione e supporto tecnico – Cisco Systems