Processus d'échange de paquets IOS IKEv1 et IKEv2 pour les profils avec plusieurs certificats

Contenu

Introduction

Ce document décrit les processus d'échange de paquets IKEv1 (Internet Key Exchange Version 1) et IKEv2 (Internet Key Exchange Version 2) lors de l'utilisation de l'authentification de certificat et les problèmes éventuels.

Voici une liste des sujets décrits dans ce document :

- Critères de sélection de certificat pour l'initiateur IKE (Internet Key Exchange) et le répondeur IKE
- Critères de correspondance du profil IKE lorsque plusieurs profils IKE sont mis en correspondance (pour les scénarios de chevauchement et de non-chevauchement)
- Les paramètres et le comportement par défaut lorsqu'aucun point de confiance n'est utilisé sous les profils IKE
- Différences entre IKEv1 et IKEv2 en ce qui concerne les critères de sélection des profils et des certificats

Note: Pour plus d'informations sur le dépannage d'un problème spécifique, reportez-vous à la section appropriée. En outre, un bref résumé est fourni à la fin du présent document.

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Configuration VPN Cisco IOS®
- Protocoles IKEv1 et IKEv2 (échange de paquets)

Components Used

Les informations de ce document sont basées sur la version 15.3T de Cisco IOS.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Informations générales

Les problèmes décrits dans ce document surviennent lorsque plusieurs points de confiance et plusieurs profils IKE sont utilisés.

Les exemples initiaux utilisés dans ce document ont un tunnel IKEv1 LAN à LAN avec deux points de confiance sur chaque routeur. Au début, il peut sembler que la configuration est correcte. Cependant, le tunnel VPN ne peut être initié qu'à partir d'un côté de la connexion en raison de la façon dont la commande **ca trust-point** est utilisée pour le comportement de profil ISAKMP (Internet Security Association and Key Management Protocol) et pour l'ordre des certificats inscrits dans le magasin local.

Un comportement différent est configuré avec la commande **ca trust-point** pour le profil ISAKMP lorsque le routeur est l'initiateur ISAKMP. Un problème peut se produire car l'initiateur ISAKMP est conscient du profil ISAKMP dès le début, de sorte que la commande **ca trust-point** configurée pour le profil peut influencer la charge utile de la demande de certificat dans le paquet Main Mode 3 (MM3). Cependant, lorsque le routeur est le répondeur ISAKMP, il lie le trafic entrant à un profil ISAKMP spécifique après réception du paquet Main Mode 5 (MM5), qui inclut l'ID IKE nécessaire pour créer la liaison. C'est pourquoi il n'est pas possible d'appliquer une commande **ca trust-point** pour le paquet du mode principal 4 (MM4), car le profil n'est pas déterminé avant le MM5.

L'ordre de la charge utile de demande de certificat dans les MM3 et MM4 et l'impact sur l'ensemble du processus de négociation sont expliqués dans ce document, ainsi que la raison pour laquelle il autorise uniquement l'établissement de la connexion à partir d'un côté du tunnel VPN.

Voici un résumé des comportements de l'initiateur et du répondeur IKEv1 :

	Initiateur IKEv1	Répondeur IKEv1
Envoyer la demande	Envoie des requêtes spécifiques uniquement pour les points d'approbation configurés sous le profil	Envoie des demandes pour tous les points d'approbation disponibles
Valider la demande	Valide par rapport à des points d'approbation spécifiques configurés sous le profil	Valide par rapport à des points d'approbation spécifiques configurés sous le profil

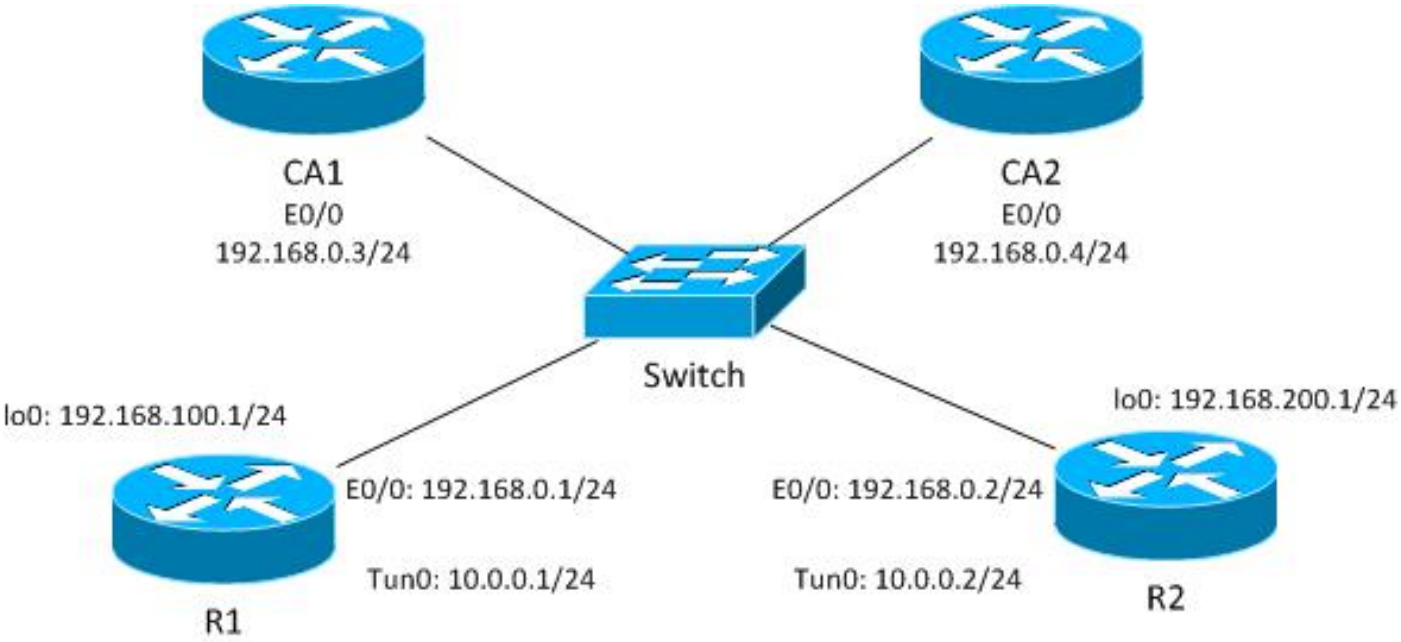
Cisco recommande de ne pas utiliser la commande **ca trust-point** pour les répondeurs ISAKMP qui ont plusieurs profils ISAKMP et qui utilisent des points de confiance configurés globalement. Pour les initiateurs ISAKMP avec plusieurs profils ISAKMP, Cisco recommande de restreindre le processus de sélection des certificats à l'aide de la commande **ca trust-point** dans chaque profil.

Le protocole IKEv2 présente les mêmes problèmes que le protocole IKEv1, mais le comportement différent de la commande **pki trustpoint** permet d'éviter les problèmes. Ceci est dû au fait que la commande **pki trustpoint** est obligatoire pour l'initiateur IKEv2, tandis que la commande **ca trust-point** est facultative pour l'initiateur IKEv1. Dans certaines circonstances (plusieurs points de confiance sous un même profil), les problèmes décrits précédemment peuvent survenir. Pour cette raison, Cisco vous recommande d'utiliser des configurations de points d'approbation symétriques pour les deux côtés de la connexion (les mêmes points d'approbation configurés sous les deux profils IKEv2).

Topologie

Il s'agit d'une topologie générique utilisée pour tous les exemples de ce document.

> **Note**: Les routeurs 1 (R1) et 2 (R2) utilisent des VTI (Virtual Tunnel Interfaces) afin d'accéder aux bouclages. Ces VTI sont protégés par IPSec.



Pour cet exemple IKEv1, chaque routeur a deux points de confiance pour chaque autorité de certification (CA) et les certificats pour chacun des points de confiance sont inscrits.

Lorsque R1 est l'initiateur ISAKMP, le tunnel négocie correctement et le trafic est protégé. C'est un comportement attendu. Lorsque R2 est l'initiateur ISAKMP, la négociation de phase1 échoue.

> **Note**: Pour les exemples IKEv2 de ce document, la topologie et l'adressage sont identiques à ceux illustrés dans l'exemple IKEv1.

Processus d'échange de paquets

Cette section décrit les variations de configuration IKEv1 et IKEv2 utilisées pour le processus d'échange de paquets, ainsi que les problèmes éventuels qui peuvent survenir.

IKEv1 avec plusieurs certificats

Voici la configuration réseau et VPN de R1 pour IKEv1 avec plusieurs certificats :

```
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
```

Voici la configuration réseau et VPN de R2 pour IKEv1 avec plusieurs certificats :

```
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
```

Dans cet exemple, R1 a deux points d'approbation : l'une utilise IOSCA1 et l'autre utilise 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
```

Dans cet exemple, R2 a également deux points de confiance : l'une utilise **IOSCA1** et l'autre utilise **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
```

Il est important de noter la différence unique dans ces configurations : le profil ISAKMP de R1 utilise la commande **ca trust-point** pour le point d'approbation **IOSCA1**, qui indique que R1 ne fait confiance qu'aux certificats validés par ce point d'approbation spécifique. En revanche, R2 fait confiance à tous les certificats validés par tous les points d'approbation définis globalement.

R1 en tant qu'initiateur IKEv1

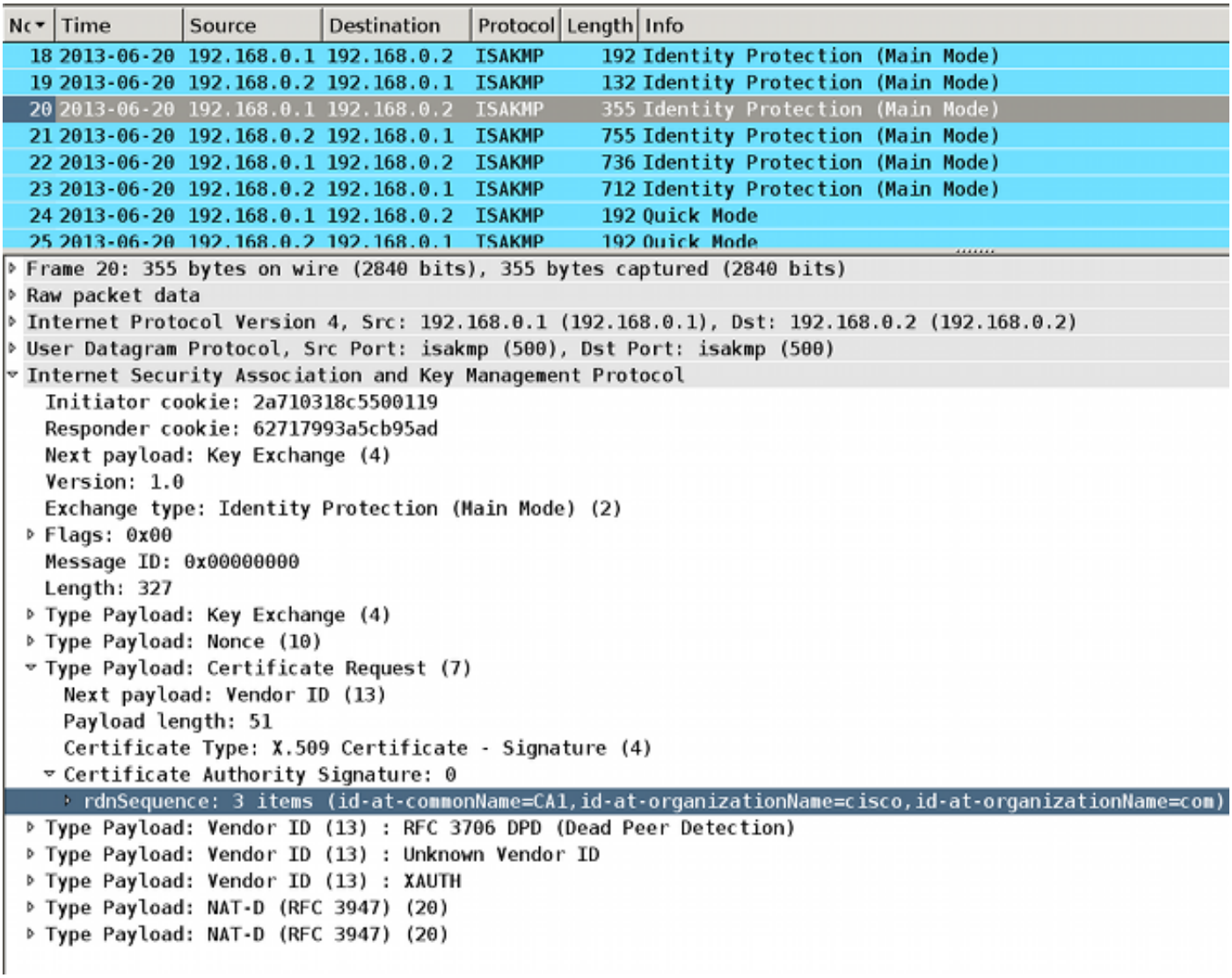
Voici les commandes de débogage pour R1 et R2 :

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

Ici, R1 lance le tunnel et envoie le certificat demandant le 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
```

Il est important de noter que le paquet ne contient qu'une seule demande de certificat, qui est uniquement destinée au point d'approbation **IOSCA1**. Ce comportement est attendu avec la configuration actuelle du profil ISAKMP (**CN=CA1, O=cisco, O=com**). Aucune autre demande de certificat n'est envoyée, que vous pouvez vérifier avec la fonctionnalité Embedded Packet Capture :



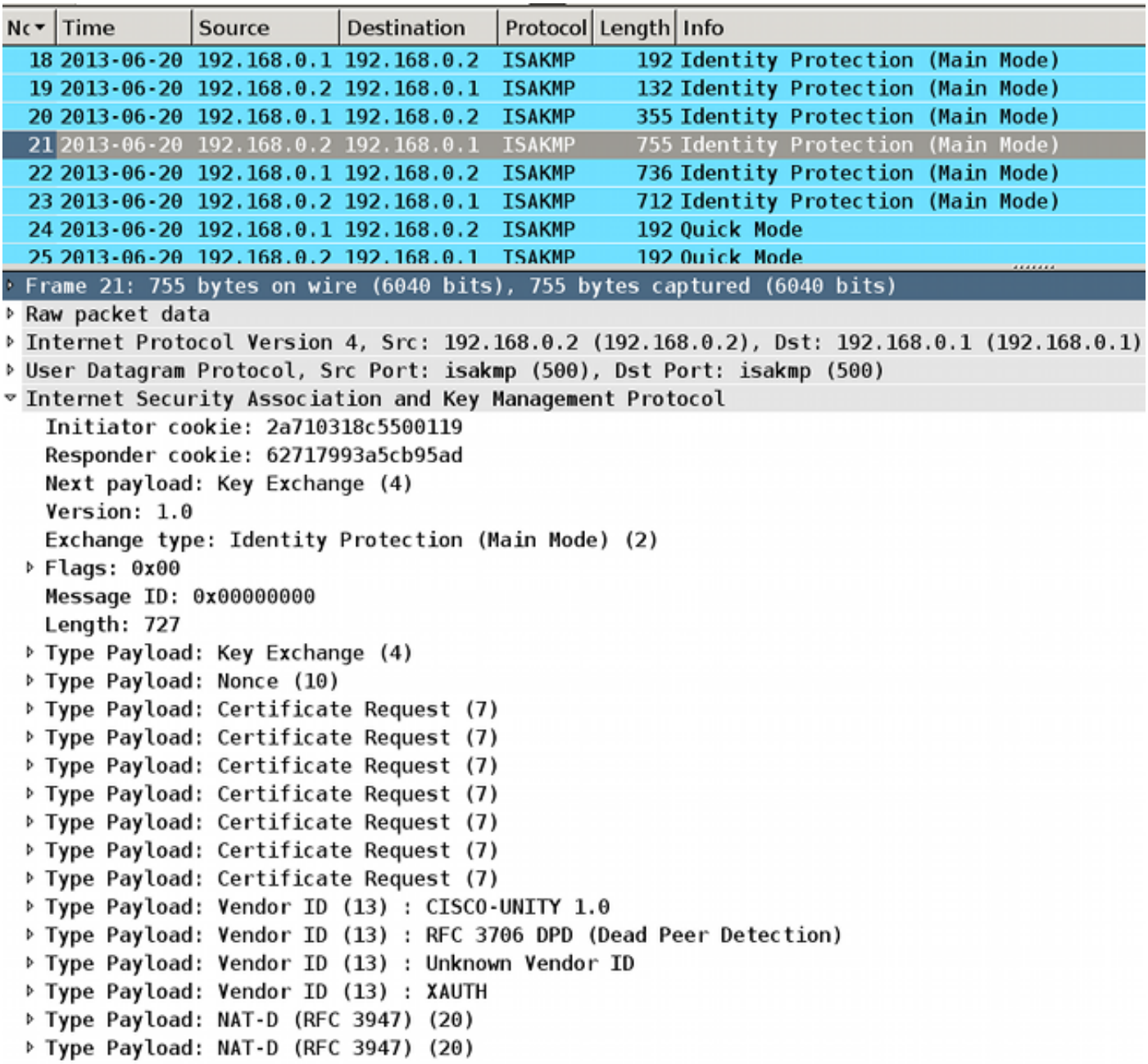
Lorsque R2 reçoit le paquet, il commence à traiter la demande de certificat, ce qui crée une correspondance qui détermine le point d'approbation et le certificat associé qui est utilisé pour l'authentification dans le MM5. L'ordre de traitement est identique à la charge utile de la demande de certificat dans le paquet ISAKMP. Cela signifie que la première correspondance est utilisée. Dans ce scénario, il n'y a qu'une seule correspondance puisque R1 est configuré avec un point d'approbation spécifique et envoie une seule demande de certificat associée au point d'approbation.

```
*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
```

Ensuite, R2 prépare le MM4. Il s'agit du paquet qui contient la demande de certificat pour tous les points de confiance approuvés. R2 étant le répondeur ISAKMP, tous les points d'approbation définis globalement sont approuvés (la configuration **ca** du **point d'approbation** n'est pas vérifiée). Deux des points de confiance sont définis manuellement (**IOSCA1** et **IOSCA2**) et les autres sont prédéfinis.

```
*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
```

Vous pouvez vérifier le paquet avec Wireshark. Le paquet MM4 de R2 contient sept entrées de demande de certificat :



Nc	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

Ensuite, R1 reçoit le MM4 de R2 avec plusieurs champs de demande de certificat :

```
*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 règle de première correspondance sur R1 correspond à la première demande de certificat avec le point d'approbation **IOSCA1**. Cela détermine que R1 utilise le certificat associé à **IOSCA1** de point d'approbation pour l'authentification dans le MM5. Le nom de domaine complet (FQDN) est utilisé comme ID IKE. Ceci est dû à la configuration **self-identity fqdn** dans le profil 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
```

Le MM5 est reçu et traité par R2. L'ID IKE reçu (**R1.cisco.com**) correspond au profil ISAKMP **prof1**. Le certificat reçu est ensuite validé et l'authentification réussit :

```
*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
```

Ensuite, R2 prépare le MM6 avec le certificat associé à **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
```

Le paquet est reçu par R1 et R1 vérifie le certificat et l'authentification :

```
*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 phase 1 est terminée. La phase 2 est négociée comme d'habitude. Le tunnel est correctement établi et le trafic est protégé.

R2 en tant qu'initiateur IKEv1

Cet exemple décrit le processus lorsque R2 lance le même tunnel IKEv1 et explique pourquoi il n'est pas établi.

> **Note**: Certaines parties des journaux sont supprimées afin de se concentrer uniquement sur les différences par rapport à l'exemple présenté dans la section précédente.

R2 envoie le MM3 avec sept charges utiles de demande de certificat, car R2 n'a pas de point d'approbation associé au profil ISAKMP (tous les points d'approbation sont approuvés) :

```
*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
```

Lorsque R1 reçoit le paquet de R2, il traite la demande de certificat et correspond au point d'approbation IOSCA1, qui détermine le certificat envoyé dans le MM6 :

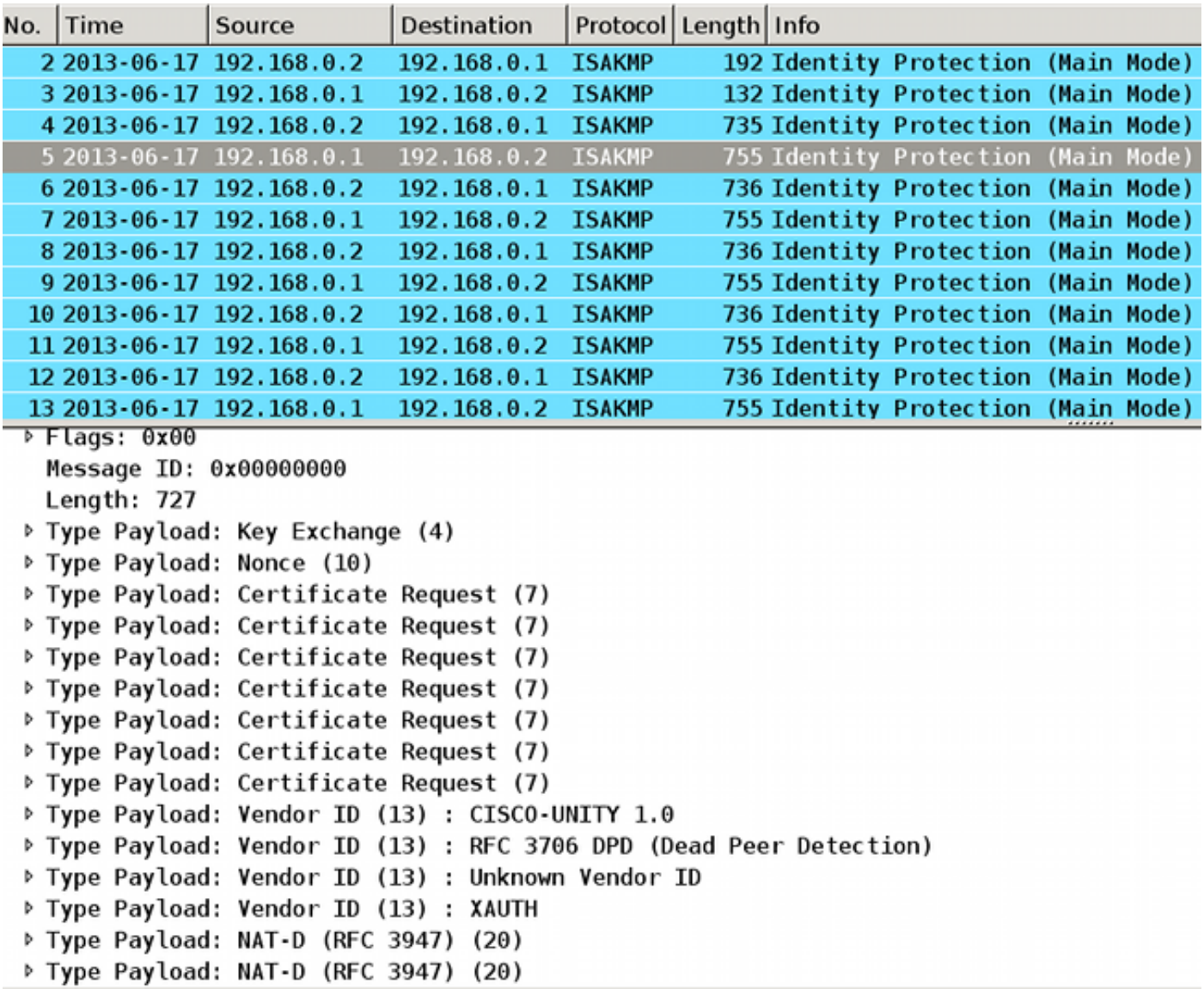
```
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=CA1,o=cisco,o=com
*Jun 17 18:08:14.321:  Choosing trustpoint IOSCA1 as issuer
*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
```

Ensuite, R1 prépare le paquet MM4 avec la charge utile de demande de certificat. Il existe maintenant plusieurs charges utiles de demande de certificat :

```
*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
```

Vérifiez les journaux avec Embedded Packet Capture (EPC) et Wireshark :



Même si R1 est configuré pour un seul point d'approbation (**IOSCA1**) dans le profil ISAKMP, plusieurs demandes de certificat sont envoyées. Cela se produit parce que la commande **ca trust-point** dans le profil ISAKMP détermine la charge utile de la demande de certificat, mais uniquement lorsque le routeur est l'initiateur de la session ISAKMP. Si le routeur est le répondeur, il existe plusieurs charges utiles de demande de certificat pour tous les points d'approbation définis globalement, car R1 ne connaît pas encore le profil ISAKMP utilisé pour la session IKE.

La session IKE entrante est liée à un profil ISAKMP spécifique après la réception du MM5, qui inclut l'ID IKE. Ensuite, la commande **match identity** pour le profil spécifique lie la session IKE au

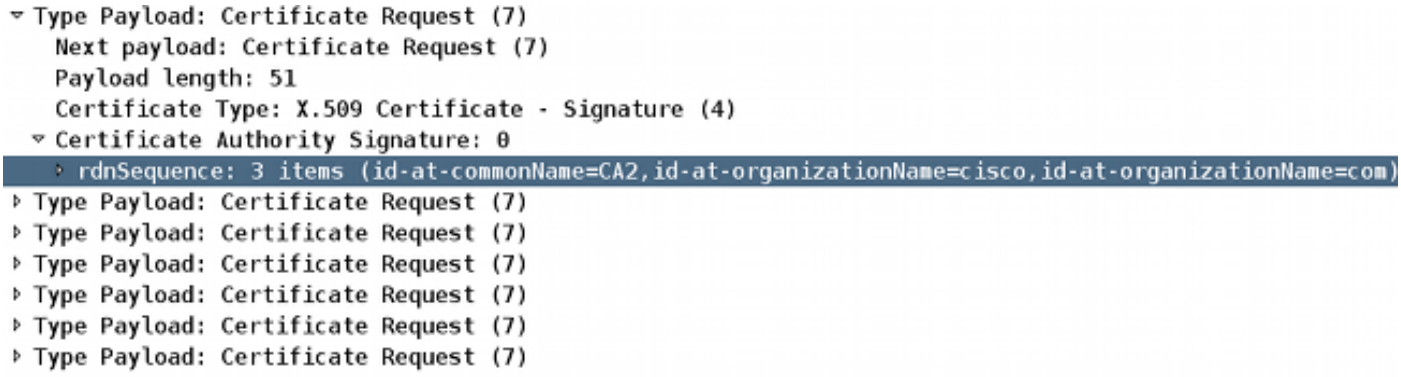
profil. Cependant, le routeur ne peut pas le déterminer jusqu'à présent. Il peut y avoir plusieurs profils ISAKMP avec différentes commandes **ca trust-point** configurées pour chaque profil.

Pour cette raison, R1 doit envoyer la demande de certificat pour tous les points d'approbation configurés globalement.

Reportez-vous à la référence de commande pour la commande **ca trust-point** :

Un routeur lançant IKE et un routeur répondant à la requête IKE doivent avoir des configurations de point de confiance symétriques. Par exemple, un routeur répondant (en mode principal IKE) qui effectue le chiffrement et l'authentification des signatures RSA peut utiliser des points de confiance définis dans la configuration globale lors de l'envoi des charges utiles CERT-REQ. Cependant, le routeur peut utiliser une liste restreinte de points de confiance définis dans le profil ISAKMP pour la vérification du certificat. Si l'homologue (l'initiateur IKE) est configuré pour utiliser un certificat dont le point de confiance figure dans la liste globale du routeur répondant mais pas dans le profil ISAKMP du routeur répondant, le certificat est rejeté. (Toutefois, si le routeur qui lance le certificat ne connaît pas les points de confiance dans la configuration globale du routeur qui répond, le certificat peut toujours être authentifié.)

Maintenant, vérifiez les détails du paquet MM4 afin de découvrir la première charge utile de demande de certificat :



Le paquet MM4 envoyé à partir de R1 inclut le point d'approbation **IOSCA2** dans la charge utile de la première demande de certificat en raison de l'ordre dans lequel les certificats sont installés ; le premier est signé par le point d'approbation **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>
```

Comparez avec le paquet MM3 envoyé depuis R2 lorsque le point de confiance IOSCA1 est inclus dans la première charge utile de demande de certificat :

```
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>
```

Maintenant, R2 reçoit le paquet MM4 de R1 et commence à traiter la demande de certificat. La première charge utile de demande de certificat correspond au point d'approbation 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
```

Lorsque R2 prépare le paquet MM5, il utilise le certificat associé au point d'approbation 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.
```

Le paquet MM5 est reçu par R1. Étant donné que R1 ne fait confiance qu'au point d'approbation IOSCA1 (pour le profil ISAKMP prof1), la validation du certificat échoue :

```
*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
```

Cette configuration fonctionne si l'ordre d'inscription du certificat sur R1 est différent, car le premier certificat affiché est signé par le point d'approbation **IOSCA1**. En outre, la première charge utile de demande de certificat dans le MM4 est le point d'approbation **IOSCA1**, qui est ensuite choisi par R2 et validé avec succès sur R1 dans le MM6.

IKEv1 sans commande *ca trust-point* dans le profil

Pour les scénarios avec plusieurs profils et points de confiance mais sans configuration de point de confiance spécifique dans les profils, il n'y a aucun problème car il n'y a pas de validation de points de confiance spécifiques déterminés par une configuration de commande **ca point de confiance**. Cependant, le processus de sélection n'est peut-être pas évident. Selon le routeur qui est l'initiateur, les différents certificats sont sélectionnés pour le processus d'authentification en fonction de l'ordre d'inscription des certificats.

Parfois, un certificat ne peut être pris en charge que par un seul côté de la connexion, par exemple dans x509 Version 1, qui n'est pas une fonction de hachage classique utilisée pour signer. Le tunnel VPN peut être établi uniquement d'un côté de la connexion.

Référence RFC pour IKEv1

Voici un extrait de RFC4945 :

3.2.7.1 . Spécification des autorités de certification

Lors de la **demande** d'échange intrabande de matériel de clé, les mises en oeuvre DEVRAIENT générer des CERTREQ pour chaque point d'ancrage de confiance homologue que la **politique locale** juge fiable **explicitement** lors d'un échange donné.
La RFC n'est pas claire. La **stratégie locale** peut **explicitement** se rapporter à la commande **ca trust-point** configurée dans le profil ISAKMP de chiffrement. Le problème est qu'à l'étape MM3 et MM4 du processus, vous ne pouvez pas sélectionner un profil ISAKMP à moins d'utiliser une adresse IP pour l'identité et les points d'approbation, car l'authentification à l'étape MM5 et MM6 du processus doit avoir lieu en premier. Pour cette raison, la **stratégie locale** se rapporte **explicitement** à tous les points d'approbation configurés sur le périphérique.

> **Note**: Ces informations ne sont pas spécifiques à Cisco, mais sont spécifiques à IKEv1.

Sélection de profil IKEv2 avec identités qui chevauchent

Avant de décrire plusieurs certificats pour IKEv2, il est important de savoir comment les profils sont sélectionnés lorsque l'identité de correspondance est utilisée, ce qui est satisfait pour tous les profils. Ce scénario n'est pas recommandé car les résultats de la négociation IKEv2 dépendent de plusieurs facteurs. Les mêmes problèmes existent pour IKEv1 lorsque des profils qui se chevauchent sont utilisés.

Voici un exemple de configuration d'initiateur 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'adresse de type d'identité est utilisée des deux côtés de la connexion. L'authentification via des certificats (peut également être des clés pré-partagées) n'est pas importante dans cet exemple. Le répondeur a plusieurs profils qui correspondent tous au trafic IKEv2 entrant :

```
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'initiateur envoie le troisième paquet IKEv2 et le répondeur doit choisir le profil en fonction de l'identité reçue. L'identité est une adresse IPv4 (**192.168.0.1**) :

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

Tous les profils satisfont cette identité en raison de la commande **match identity** configurée. L'IOS choisit le dernier dans la configuration, qui est **profile3** dans cet exemple :

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

Afin de vérifier l'ordre, entrez la commande **show crypto ikev2 profile**.

> **Note**: Même lorsqu'il y a une adresse générique (0.0.0.0) dans le profil, elle est toujours sélectionnée. L'IOS ne tente pas de trouver la meilleure correspondance ; il essaie de trouver la première correspondance. Cependant, cela se produit uniquement parce que tous les profils ont la même commande **match identity remote** configurée. Pour les profils IKEv1 et IKEv2 qui ont des règles d'identité de correspondance différentes, la plus spécifique est toujours utilisée. Cisco vous recommande de ne pas configurer les profils avec la commande **match identity qui se** superpose, car il est difficile de prédire le profil sélectionné.

Dans ce scénario, **profile3** est sélectionné par le répondeur, mais **profile1** est utilisé pour l'interface de tunnel. Une erreur s'affiche lors de la négociation de l'ID de proxy :

```
*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
```

Flux IKEv2 lorsque des certificats sont utilisés

Lorsque des certificats sont utilisés pour IKEv2 afin de s'authentifier, l'initiateur n'envoie pas la

charge utile de demande de certificat dans le premier paquet :

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

Le répondeur répond avec la charge utile de la demande de certificat (deuxième paquet) et toutes les autorités de certification car le répondeur ne connaît pas le profil qui doit être utilisé à cette étape. Le paquet qui contient les informations est envoyé à l'initiateur :

```
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’initiateur traite le paquet et choisit un point d’approbation correspondant à l’autorité de certification proposée :

```
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'initiateur envoie ensuite le troisième paquet avec la demande de certificat et la charge utile de certificat. Ce paquet est déjà chiffré avec du matériel de frappe de la phase 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)
```

Le quatrième paquet est envoyé du répondeur à l'initiateur et contient uniquement la charge utile du certificat :

```
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)
```

Le flux décrit ici est similaire au flux IKEv1. Le répondeur doit envoyer la charge utile de la demande de certificat au début sans connaître le profil à utiliser, ce qui crée les mêmes problèmes que ceux précédemment décrits pour IKEv1 (du point de vue du protocole). Cependant, la mise en oeuvre sur IOS est meilleure pour IKEv2 que pour IKEv1.

Point de confiance obligatoire IKEv2 pour l'initiateur

Voici un exemple de la tentative d'un initiateur IKEv2 d'utiliser un profil avec authentification de certificat et sans point de confiance configuré sous ce profil :

```
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
```

Le premier paquet est envoyé sans charge utile de demande de certificat, comme décrit précédemment. La réponse du répondeur inclut la charge utile de la demande de certificat pour tous les points d'approbation définis en mode de configuration globale. Ceci est reçu par l'initiateur :

```
*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'initiateur ne connaît pas le point d'approbation qui doit être utilisé pour signer. C'est la principale différence lorsque l'implémentation IKEv2 est comparée à IKEv1. Le point d'approbation de l'initiateur IKEv2 doit être configuré sous le profil d'initiateur IKEv2, mais il n'est pas nécessaire pour le répondeur IKEv2.

Voici un extrait de la référence de commande :

Si aucun point de confiance n'est défini dans la configuration du profil IKEv2, la valeur par défaut est de **valider le certificat** en utilisant tous les points de confiance définis dans la configuration globale

Il est possible de définir différents points de confiance ; une pour signer et une autre pour valider. Malheureusement, le point de confiance obligatoire configuré sous le profil IKEv2 ne résout pas

tous les problèmes.

R2 en tant qu'initiateur IKEv2

Dans cet exemple, R2 est l'initiateur 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
```

Dans cet exemple, R1 est le répondeur 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
```

Ici, R2 envoie le premier paquet sans aucune demande de certificat. Le répondeur répond par une demande de certificat pour tous les points d'approbation configurés. L'ordre des charges utiles est similaire à l'IKEv1 et dépend des certificats installés :

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

Le premier certificat configuré sur R1 est associé au point d'approbation **TP2**, de sorte que la première charge utile de demande de certificat est pour l'autorité de certification associée au point d'approbation **TP2**. Ainsi, R2 le sélectionne pour l'authentification (première règle de correspondance) :

```
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
```

Ensuite, R2 prépare une réponse (paquet 3) avec la charge utile de la demande de certification associée à **TP2**. R1 ne peut pas faire confiance au certificat car il est configuré pour la validation par rapport au point d'approbation **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)
```

Comme indiqué précédemment, Cisco recommande de ne pas utiliser plusieurs points de confiance sous un profil IKEv2. Lorsque vous utilisez plusieurs points de confiance, il est nécessaire de s'assurer que les deux parties font exactement confiance aux mêmes points de confiance. Par exemple, R1 et R2 ont tous deux TP1 et TP2 configurés dans leurs profils.

Résumé

Cette section fournit un bref résumé des informations décrites dans le document.

Le contenu de la charge utile de la demande de certificat dépend de la configuration. Si un point d'approbation spécifique est configuré pour le profil ISAKMP et que le routeur est l'initiateur ISAKMP, la demande de certificat dans le MM3 contient uniquement l'autorité de certification associée au point d'approbation. Cependant, si le même routeur est le répondeur ISAKMP, alors le paquet MM4 envoyé par le routeur inclut plusieurs charges utiles de demande de certificat pour tous les points de confiance définis globalement (lorsque la commande **ca trust-point** n'est pas prise en compte). Cela se produit parce que le répondeur ISAKMP peut déterminer le profil ISAKMP qui doit être utilisé uniquement après avoir reçu le MM5 et la demande de certificat qui est inclus dans le MM4.

La charge utile de la demande de certificat dans le MM3 et le MM4 est importante en raison de la

première règle de correspondance. La première règle de correspondance détermine le point d'approbation utilisé pour la sélection du certificat, qui est nécessaire pour l'authentification dans MM5 et MM6.

L'ordre de la charge utile de demande de certificat dépend de l'ordre des certificats installés. L'émetteur du premier certificat qui apparaît dans la sortie de la commande **show crypto pki certificate** est envoyé en premier. Ce premier certificat est le dernier qui est inscrit.

Il est possible de configurer plusieurs points de confiance pour un profil ISAKMP. Si cette opération est effectuée, toutes les règles précédentes s'appliquent toujours.

Tous les problèmes et mises en garde décrits dans ce document sont dus à la conception du protocole IKEv1. L'étape d'authentification se produit dans le MM5 et le MM6, tandis que les propositions pour l'authentification (demandes de certificat) doivent être envoyées à une étape antérieure (avant) sans connaissance du profil ISAKMP qui doit être utilisé. Il ne s'agit pas d'un problème propre à Cisco et il est lié aux limites de la conception du protocole IKEv1.

Le protocole IKEv2 est similaire à IKEv1 en ce qui concerne le processus de négociation de certificat. Cependant, la mise en oeuvre de l'IOS force l'utilisation de points de confiance spécifiques pour l'initiateur. Cela ne résout pas tous les problèmes. Lorsque plusieurs points de confiance sont configurés pour un seul profil et qu'un seul point de confiance est configuré de l'autre côté, il est toujours possible de rencontrer des problèmes d'authentification. Cisco recommande d'utiliser des configurations de points d'approbation symétriques pour les deux côtés de la connexion (les mêmes points d'approbation configurés pour les deux profils IKEv2).

Voici quelques notes importantes sur les informations décrites dans ce document :

- Avec des configurations de points d'approbation asymétriques pour les profils IKEv1 des homologues, le tunnel peut démarrer à partir d'un seul côté du tunnel. La configuration du point d'approbation pour le profil IKEv1 est facultative.

- Avec des configurations de points d'approbation asymétriques pour les profils IKEv2 des homologues, le tunnel peut démarrer à partir d'un seul côté du tunnel. La configuration du point d'approbation pour le profil IKEv2 est obligatoire pour l'initiateur.

- L'ordre de charge utile de la demande de certificat dépend de l'ordre des certificats qui apparaissent dans la sortie de la commande **show crypto pki certificate** (première correspondance).

- L'ordre de charge utile de la demande de certificat détermine le certificat sélectionné par le répondeur (première correspondance).

- Lorsque vous utilisez plusieurs profils pour IKEv1 et IKEv2 et que les mêmes règles d'identité de correspondance sont configurées, il est difficile de prédire les résultats (trop de facteurs impliqués).

- Cisco vous recommande d'utiliser des configurations de points d'approbation symétriques pour IKEv1 et IKEv2.

Informations connexes

- Guide de configuration d'Internet Key Exchange for IPsec VPN, Cisco IOS version 15M&T - Correspondance de certificat avec profil ISAKMP
- Référence des commandes de sécurité Cisco IOS : Commandes A à C - ca trust point via clear eou
- Support et documentation techniques - Cisco Systems