# プロファイルに複数の証明書がある場合の IOS IKEv1 および IKEv2 パケット交換プロセス

## 内容



## 概要

このマニュアルは、証明書認証を使用する場合のインターネット キー エクスチェンジ バージョン 1（IKEv1）とインターネット キー エクスチェンジ バージョン 2（IKEv2）パケット交換プロセスについておよび発生する可能性のある問題について説明します。

このマニュアルで説明する内容のリストを次に示します。

- インターネット キー エクスチェンジ（IKE）イニシエータおよび IKE レスポンダの証明書選択基準
- 複数の IKE プロファイルが一致したときの IKE プロファイルの一致基準（オーバーラップ シナリオおよび非オーバーラップ シナリオ）
- トラストポイントが IKE プロファイルで使用されていない場合のデフォルト設定と動作
- プロファイルと証明書選択基準に関する IKEv1 と IKEv2 の違い

**注**：特定の問題をトラブルシューティングする方法の詳細については、修正セクションを参

照してください。この文書の末尾に、簡単な要約も示されています。

# 前提条件

## 要件

次の項目に関する知識があることが推奨されます。

- Cisco IOS® VPN の設定
- IKEv1 および IKEv2 プロトコル（パケット交換）

## 使用するコンポーネント

このドキュメントの情報は、Cisco IOS バージョン 15.3T に基づくものです。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期（デフォルト）設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

# 背景説明

このマニュアルで説明している問題は、複数のトラストポイントと複数の IKE プロファイルが使用されている場合に発生します。

このマニュアルで使用している最初の例では、2 個のトラストポイントを持つ IKEv1 LAN-to-LAN トンネルが各ルータにあります。最初は、この設定は正しいように思えます。ただし、この VPN トンネルは接続の片方の側からのみ開始できます。これは、**ca trust-point コマンドが Internet Security Association and Key Management Protocol（ISAKMP）プロファイルの動作のために使用されている方法とローカル ストア内に登録されている証明書の順序が理由です。**

ルータが ISAKMP イニシエータである場合は、ISAKMP プロファイル用に **ca trust-point コマンドに異なる動作が設定されます。**ISAKMP イニシエータは開始側からの ISAKMP プロファイルを認識しているため、プロファイルに設定された **ca trust-point コマンドはメイン モード パケット 3（MM3）内の証明書要求ペイロードに影響を与えることがあるため、問題が発生することがあります。** 一方、ルータが ISAKMP レスポンダである場合は、バインドを作成するために必要な IKE ID を含んでいるメイン モード パケット 5（MM5）の受信後に特定の ISAKMP プロファイルに着信トラフィックをバインドします。したがって、プロファイルは MM5 以前には判別されないため、メイン モード パケット 4（MM4）パケットには **ca trust-point** コマンドを適用できません。

MM3 および MM4 での証明書要求ペイロードの順序およびネゴシエーション プロセス全体に対するインパクトについてと、VPN トンネルの一方からのみ接続を確立できる理由は、このマニュアルで説明してあります。

IKEv1 イニシエータとレスポンダの動作の要約を次に示します。

| | **IKEv1 イニシエータ** | **IKEv1 レスポンダ** |
|---|---|---|
| **要求を送信** | プロファイルで設定されたトラストポイントのみに特定の要求を送信する | 使用可能なすべてのトラストポイントの送信を要求する |
| **要求を検証** | プロファイルで設定された特定のトラストポイントに対して検証する | プロファイルで設定された特定のトラストポイントに対して検証する |

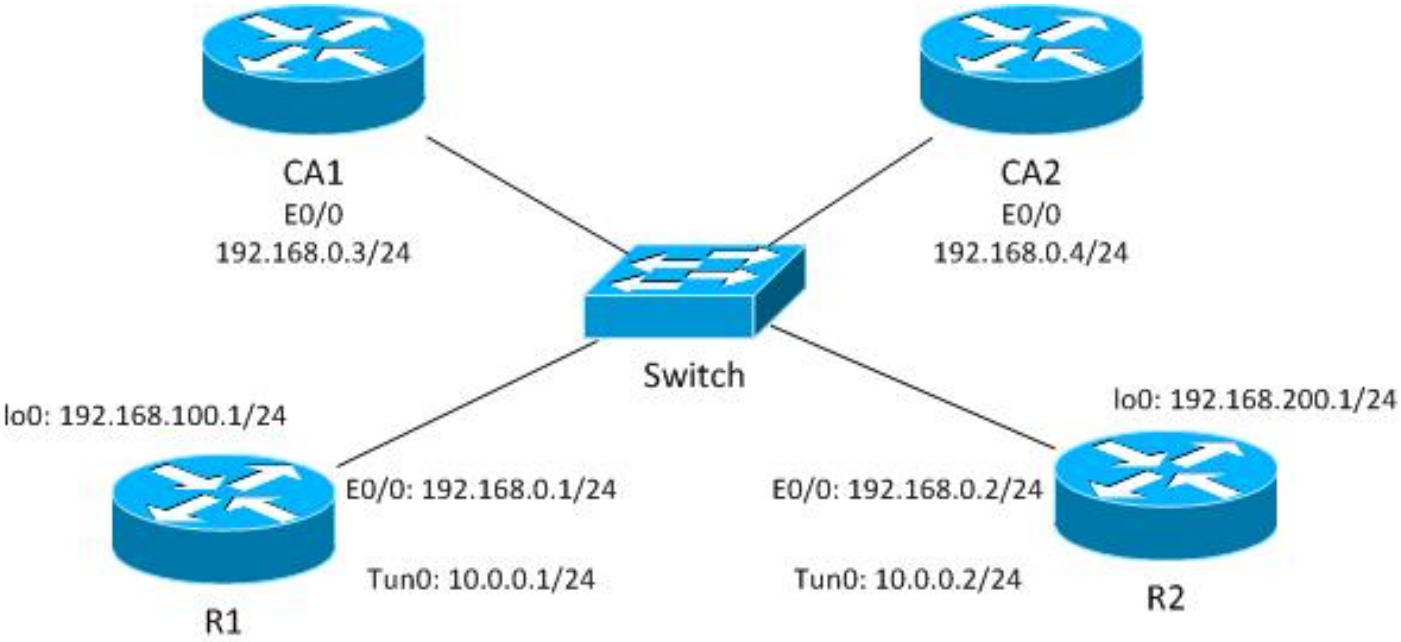
Cisco では、複数の ISAKMP プロファイルがあり、グローバル設定されたトラストポイントを使用する ISAKMP レスポンダに **ca trust-point コマンドを使用しないことを推奨します。**Cisco では、複数の ISAKMP プロファイルを持つ ISAKMP イニシエータに対しては、各プロファイルで **ca trust-point コマンドを使用して証明書選択プロセスを絞り込むことを推奨します。**

IKEv2 プロトコルには IKEv1 プロトコルと同じ問題がありますが、**pki trustpoint コマンドの動作が異なることが、問題の発生を防止するために役だっています。**これは、**pki trustpoint コマンドは IKEv2 イニシエータで必須であるのに対し、ca trust-point コマンドは IKEv1 イニシエータでオプションであるためです。**特定の状況下（単一プロファイルに複数のトラストポイント）では、前述の問題が発生するおそれがあります。そのため、Cisco では接続の両側に対称トラストポイント設定を使用することを推奨します（両方の IKEv2 プロファイルで同じトラストポイントを設定）。

# トポロジ

これは、このマニュアルですべての例に使用される汎用トポロジです。

> **注：**ルータ 1（R1）とルータ 2（R2）ではループバックにアクセスするために仮想トンネル インターフェイス（VTI）を使用します。これらの VTI は、IPSec によって保護されています。




この IKEv1 の例では、各ルータの認証局（CA）ごとに 2 個のトラストポイントがあり、各トラストポイントの証明書が登録されています。

R1 が ISAKMP イニシエータである場合、トンネルは適切にネゴシエートされ、トラフィックは保護されます。これは正常な動作です。R2 が ISAKMP イニシエータである場合、フェーズ 1 ネゴシエーションは失敗します。

> **注**：このマニュアルの IKEv2 の例では、トポロジとアドレッシングは IKEv1 の例で示したものと同じです。

# パケット交換プロセス

ここでは、パケット交換プロセスに使用される、IKEv1 と IKEv2 の設定のバリエーションと、発生する可能性のある問題について説明します。

## 複数の証明書を使用する IKEv1

複数の証明書を使用する IKEv1 の R1 ネットワークと VPN 設定を次に示します。

```
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
```

複数の証明書を使用する IKEv1 の R2 ネットワークと VPN 設定を次に示します。

```
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
```

この例では、R1 に 2 個のトラストポイントがあります。1 個目では IOSCA1 を使用し、2 個目では IOSCA2 を使用します。

```
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
```

この例では、R2 にも 2 個のトラストポイントがあります。1 個目では **IOSCA1 を使用し、2 個目では IOSCA2** を使用します。

```
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
```

これらの設定にある 1 つの違いに注意することが重要です。R1 ISAKMP プロファイルでは **IOSCA1 トラストポイント用に ca trust-point コマンドを使用します。これは、R1 では、この特定のトラストポイントによって検証される証明書のみを信頼することを示します。**これに対し、R2 はグローバルに定義されたすべてのトラストポイントによって検証されるすべての証明書を信頼します。

## IKEv1 イニシエータとしての R1

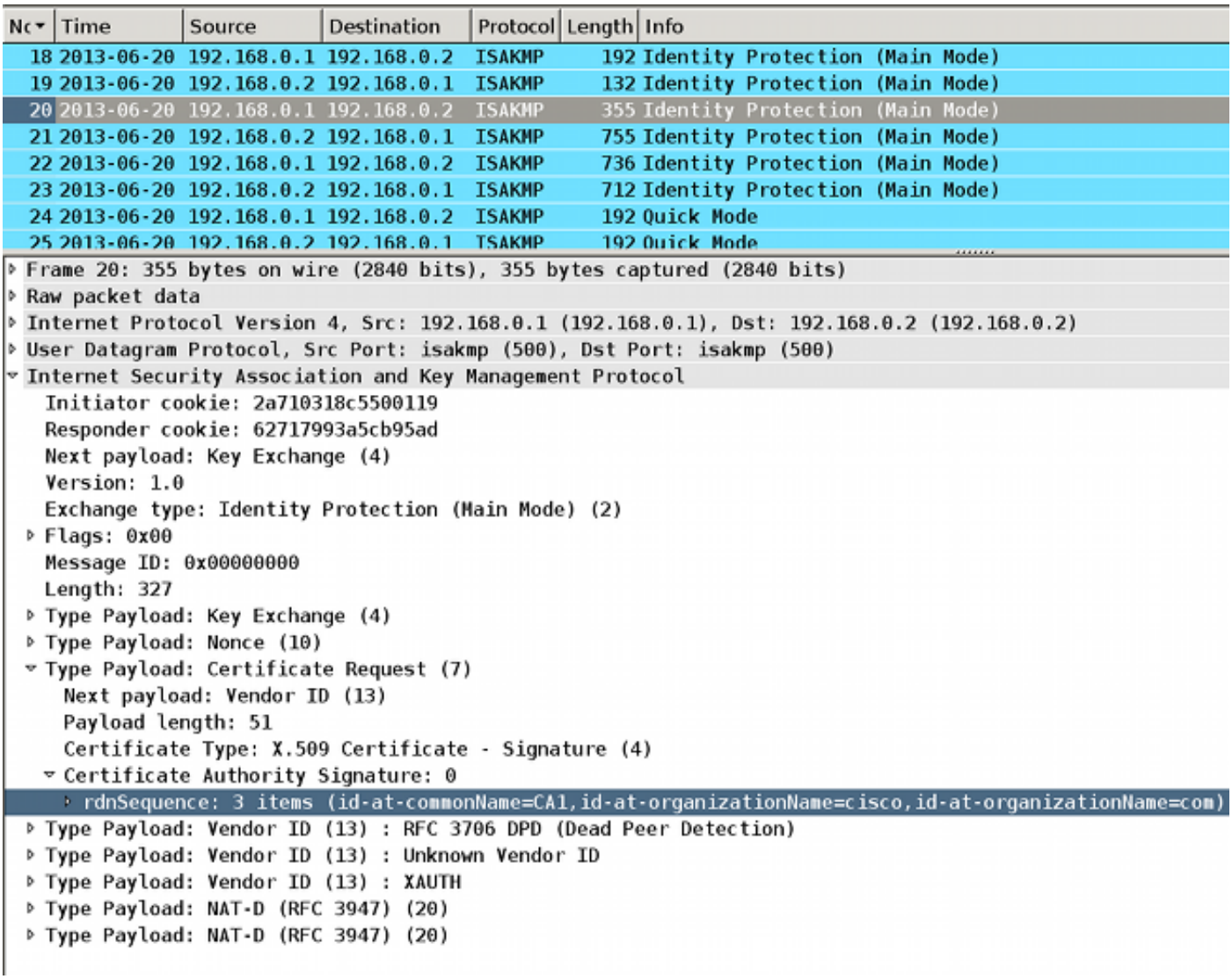
次に、R1 と R2 の両方のデバッグ コマンドを示します。

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

ここで、R1 は、トンネルを開始し、MM3 に入れて証明書要求を送信します。

```
*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
```

このパケットには単一の証明書要求のみが含まれていることに注目してください。これは**IOSCA1 トラストポイント専用です。**これは、ISAKMP プロファイル（**CN=CA1, O=cisco, O=com**）の現在の設定で予期される動作です。 他に送信される証明書要求はありません。このことは、Embedded Packet Capture 機能で検証できます。



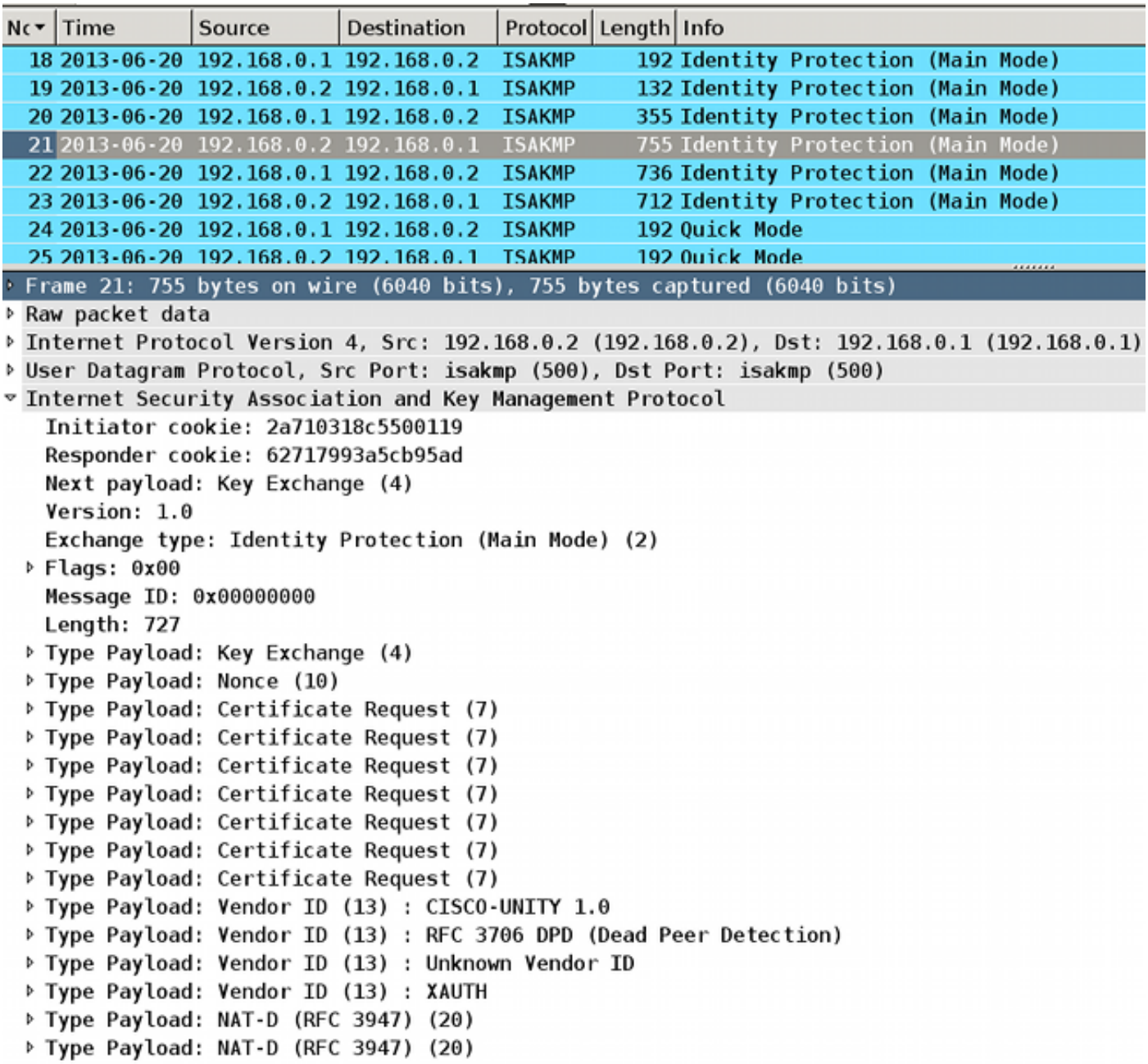

R2はパケットを受信すると、証明書要求の処理を開始します。これにより、MM5の認証に使用されるトラストポイントと関連付けられた証明書を決定する一致が作成されます。プロセスの順序は、ISAKMPパケットの証明書要求ペイロードと同じです。これは、最初の一致が使用されることを意味します。このシナリオでは、R1 に特定の単一トラストポイントが設定されており、このトラストポイントに関連付けられる単一の証明書要求のみが送信されるため単一の一致のみがあります。

```
*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
```

その後、R2はMM4を準備します。これは、すべての信頼されたトラストポイントの証明書要求を含むパケットです。R2 は ISAKMP レスポンダであるため、グローバル定義されたすべてのトラストポイントが信頼されます（ca trust-point 設定は検査されない）。 トラストポイントのうち 2 個（IOSCA1 および IOSCA2）は手動で定義されており、残りは事前に定義されています。

```
*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
```

Wireshark でパケットを検証できます。R2 からの MM4 パケットは 7 個の証明書要求エントリを含んでいます。



| 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 |



次に、R1 は複数の証明書要求フィールドを持つ MM4 を R2 から受信します。

```
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=CA1,o=cisco,o=com
*Jun 20 13:00:37.623: ISAKMP: Examining profile list for trustpoint IOSCA1
*Jun 20 13:00:37.623: ISAKMP: Found matching profile for IOSCA1
*Jun 20 13:00:37.623:  Choosing trustpoint IOSCA1 as issuer
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=CA2,o=cisco,o=com
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by ou=Class 3
 Public Primary Certification Authority,o=VeriSign, Inc.,c=US
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
```

```
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=Cisco SSCA2,o=Cisco Systems
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=Cisco Manufacturing CA,o=Cisco Systems
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=Cisco Root CA 2048,o=Cisco Systems
*Jun 20 13:00:37.623: ISAKMP:(1010): processing CERT_REQ payload. message ID = 0
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants a CT_X509_SIGNATURE cert
*Jun 20 13:00:37.623: ISAKMP:(1010): peer wants cert issued by
 cn=Cisco Root CA M1,o=Cisco
```

R1 の最初の一致ルールにより、**IOSCA1 トラストポイントの最初の証明書要求と一致します。**これにより、R1がトラストポイント**IOSCA1に関連付けられた証明書を**MM5の認証に使用することが決定されます。完全修飾ドメイン名(FQDN)がIKE IDとして使用されます。これは、ISAKMP プロファイル内の **self-identity fqdn 設定によります。**

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

MM5がR2によって受信され、処理されます。受信したIKE ID(**R1.cisco.com**)はISAKMPプロファイル**prof1**と一致します。受信した証明書が検証され、認証が成功します。

```
*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
```

次に、R2 は **IOSCA1** と関連付けられている証明書によって MM6 を準備します。

```
*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
```

パケットは R1 で受信され、R1 は証明書と認証を検証します。

```
*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
```

これで、フェーズ 1 が完了します。 フェーズ 2 は通常どおりにネゴシエートされます。トンネルは正常に確立され、トラフィックが保護されます。

## IKEv1 イニシエータとしての R2

この例では、R2 が同じ IKEv1 トンネルを開始したときのプロセスと、確立されない理由について説明します。

> **注**：前の項で示した例との差異に関連する部分のみに焦点を当てるように、ログの一部分は削除されています。

R2 には ISAKMP プロファイルと関連付けられたトラストポイントがないため（すべてのトラストポイントを信頼）、R2 では 7 つの証明書要求ペイロードを含む MM3 を送信します。

```
*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
```

R1 では R2 からパケットを受信すると、証明書要求を処理し、**IOSCA1 トラストポイントが一致します。これにより、MM6 で送信される証明書が決定します。**

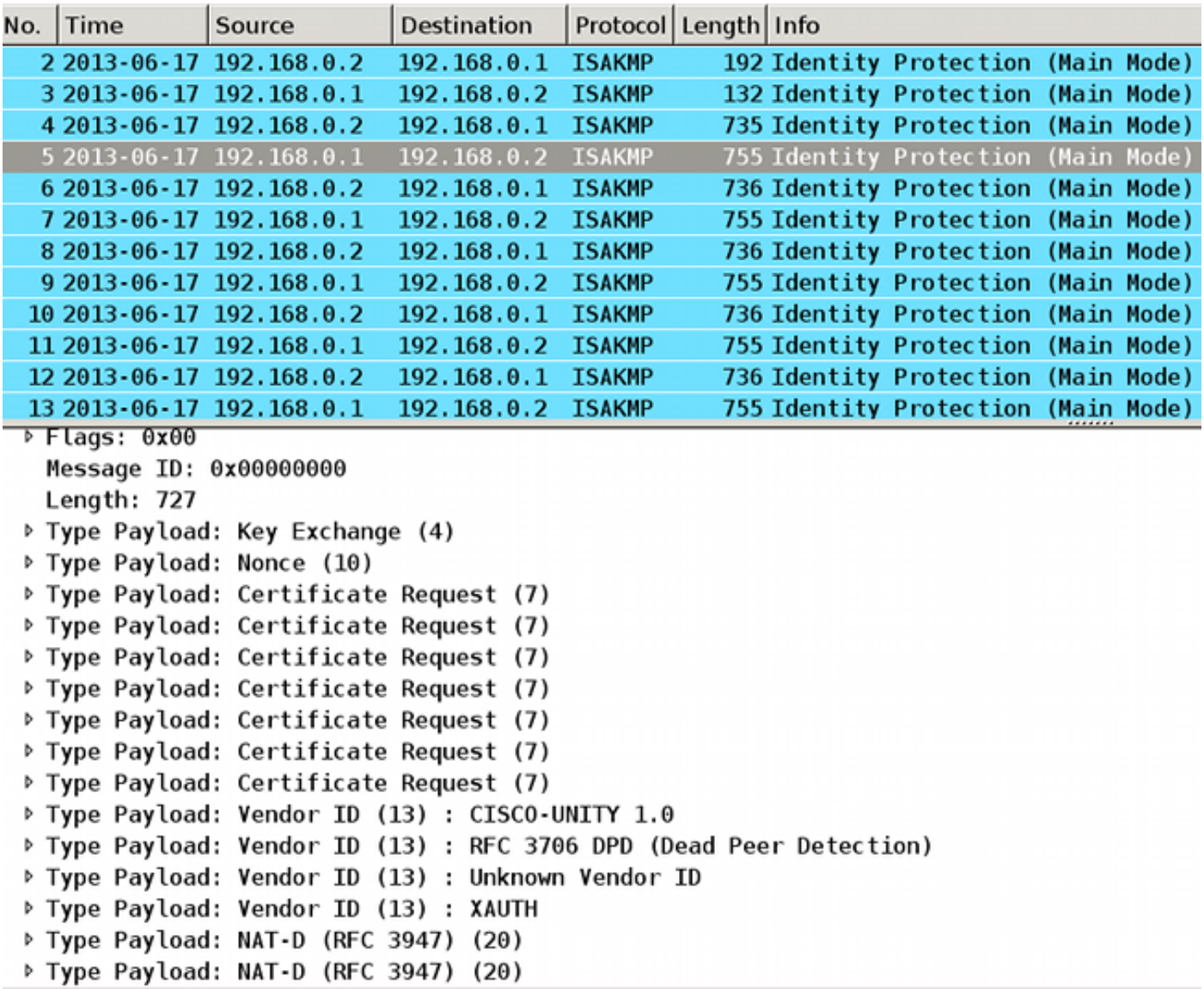
```
*Jun 17 18:08:14.321: ISAKMP:(1099): processing CERT_REQ payload.
 message ID = 0
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants a CT_X509_SIGNATURE cert
*Jun 17 18:08:14.321: ISAKMP:(1099): peer wants cert issued by
 cn=CA1,o=cisco,o=com
*Jun 17 18:08:14.321:  Choosing trustpoint IOSCA1 as issuer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

その後、R1 はこの証明書要求ペイロードで MM4 パケットを準備します。これで証明書要求ペイロードが複数存在するようになりました。

```
*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
```

Embedded Packet Capture（EPC）および Wireshark でログを検証します。



| 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) |



R1 の ISAKMP プロファイルは単一のトラストポイント（**IOSCA1**）用に設定されていますが、送信される複数の証明書要求があります。これは、ISAKMP プロファイルの **ca trust-point コマンドが証明書要求ペイロードを決定する一方で、ルータがこの ISAKMP セッションのイニシエータである場合に限られるためです。**ルータがレスポンダである場合、R1 ではこの IKE セッションに使用する ISAKMP プロファイルをまだ把握できないため、グローバル定義されたすべてのトラストポイントに対する複数の証明書要求ペイロードがあります。

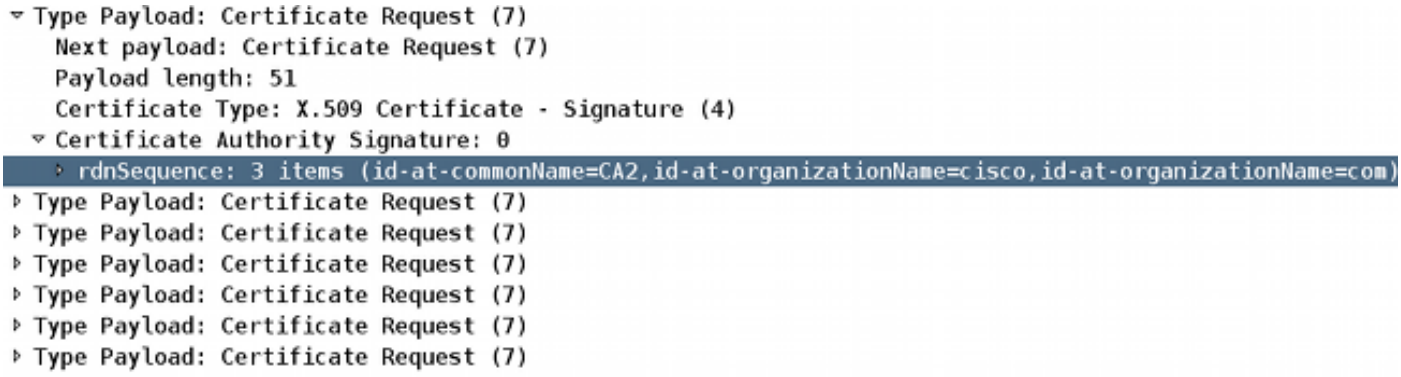
着信IKEセッションは、IKE IDを含むMM5の受信後、特定のISAKMPプロファイルにバインドされます。その後、特定のプロファ**イルに対する**match identityコマンドによって、IKEセッションが

プロファイルにバインドされます。ただし、ルータではこの時点までにこれを判別できません。各プロファイル用に設定されている異なる **ca trust-point コマンドによる複数の ISAKMP プロファイルがある可能性があります。**

したがって、R1 ではグローバル設定されたすべてのトラストポイントの証明書要求を送信する必要があります。

**ca trust-point コマンドの**コマンド リファレンスを参照してください。

**IKE を開始するルータと IKE 要求に応答するルータのトラストポイント設定は互いに対称的である必要があります。**たとえば、RSA シグネチャ暗号化および認証を実行中の応答ルータ（IKE メイン モード）では、CERT-REQ ペイロードの送信時に、グローバル コンフィギュレーション内で定義されたトラストポイントが使用されている場合があります。しかし、そのルータでは、証明書の検証のために ISAKMP プロファイル内で定義されたトラストポイントの制限リストが使用されている場合があります。ピア（IKE のイニシエータ）が、トラストポイントが応答ルータのグローバル リスト内に存在するが、応答ルータの ISAKMP プロファイル内には存在しない証明書を使用するように設定されている場合、その証明書は拒否されます。（ただし、開始ルータによって、応答ルータのグローバル コンフィギュレーション内のトラストポイントが認識されていない場合は、その証明書は認証されます）。
ここで最初の証明書要求ペイロードを検出するために MM4 パケットの詳細を検証します。




R1 から送信される MM4 パケットには、証明書がインストールされている順序が理由で、最初の証明書要求ペイロードに **IOSCA2 トラストポイントが含められています。**最初のペイロードは **IOSCA2 トラストポイントによって署名されています。**

```
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>
```

最初の証明書要求ペイロードに IOSCA1 トラストポイントが含まれるときの R2 から送信される MM3 パケットと比較してください。

```
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>
```

ここで、R2 は、R1 から MM4 パケットを受信し、証明書要求の処理を開始します。最初の証明書要求ペイロードは **IOSCA2 トラストポイントに一致します。**

```
*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
```

MM5 パケットを準備するとき、R2 では IOSCA2 トラストポイントと関連付けられている証明書を使用します。

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

MM5パケットがR1で受信されます。R1はIOSCA1トラストポイント(ISAKMPプロファイル prof1用)のみを信頼するために、証明書の検証は失敗します。

```
*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
```

この設定は、R1 での証明書の登録順序が異なっていれば、最初に表示される証明書が IOSCA1 トラストポイントによって署名されているため機能します。また、MM4 の最初の証明書要求ペイロードは IOSCA1 トラストポイントであり、これは次に R2 で選択されて、MM6 に入れて R1 で正常に検証されます。

## プロファイルに *ca trust-point* コマンドのない *IKEv1*

複数のプロファイルとトラストポイントを使用している一方で、プロファイルに特定のトラストポイントを設定していないシナリオでは、**ca trust-point コマンド設定によって判別される特定のトラストポイントの検証がないため、問題はありません。**ただし、選択プロセスは明らかではないことがあります。イニシエータであるルータに応じて、証明書の登録の順序に関して認証プロセス用に異なる証明書が選択されます。

証明書は、署名用に使用される一般的なハッシュ関数ではない x509 バージョン 1 など、接続の一方の側だけでサポートできることがあります。VPN トンネルは接続の一方の側からのみ確立できることがあります。

## IKEv1 の RFC リファレンス

次に RFC4945 の一部分を示します。

3.2.7.1.証明機関の指定

キー関連情報のインバンド交換を **要求する場合、実装では、特定の交換時にローカル ポリシーで明示的に**信頼できると見なすピア トラスト アンカーごとに CERTREQ を生成する必要があります。
RFC は明確ではありません。クリプト ISAKMP プロファイルで設定された **ca trust-point コマンドとローカル ポリシーが明示的に関連している場合があります。**問題は、プロセスの MM3 および MM4 ステージでは、ID の IP アドレスおよびトラストポイントを使用していない限り、プロセスの MM5 ステージと MM6 ステージでの認証がまず発生する必要があるため、ISAKMP プロファイルを選択できないことにあります。したがって、デバイスに設定されているすべてのトラストポイントとローカル ポリシーは明示的に関連します。

**注：**この情報は、Cisco 固有ではなく、IKEv1 固有です。

## オーバーラップする ID のある IKEv2 プロファイル選択

IKEv2 の複数の証明書を説明する前に、すべてのプロファイルを満たす一致する ID が使用される場合のプロファイル選択方法を理解することが重要です。このシナリオでは、IKEv2 ネゴシエーションの結果が複数の要因に依存するため、推奨されるシナリオではありません。IKEv1 でオーバーラップするプロファイルを使用する場合にも同じ問題が存在します。

次に IKEv2 イニシエータ設定の例を示します。

```
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
```

ID タイプのアドレスは、接続の両側に使用されます。この例では、証明書を介した認証 ( 事前共有キーにすることも可能 ) は重要ではありません。着信 IKEv2 トラフィックとすべて一致する複数のプロファイルがレスポンダにあります。

```
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
```

イニシエータは 3 番目の IKEv2 パケットを送信し、レスポンダは受信した ID に基づいてプロファイルを選択する必要があります。ID は IPv4 アドレス ( **192.168.0.1 ) です。**

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

すべてのプロファイルは設定されている **match identity コマンドによってこの ID を満たします。** IOS は設定の最後のプロファイル、この例では **profile3** を選択します。

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

順序を検証するには、**show crypto ikev2 profile コマンドを入力します。**

> **注：**プロファイルにジェネリック アドレス ( 0.0.0.0 ) がある場合でも、選択は同じです。IOS は最適な一致の検索を試行しません。最初の一致の検索を試行します。ただし、これが発生するのはすべてのプロファイルに同じ **match identity remote コマンドが設定されている場合だけです。**異なる一致 ID ルールがある IKEv1 および IKEv2 プロファイルの場合は、最も具体的なプロファイルが常に使用されます。Cisco では、選択されるプロファイルを予測することが難しいため **overlapping match identity コマンドを設定したプロファイルを保持しないことを推奨します。**

このシナリオでは、**profile3 をレスポンダで選択していますが、profile1 をトンネル インターフェイスに使用しています。**これにより、プロキシ ID がネゴシエートされるときにエラーが表示されます。

```
*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
```

### 証明書を使用する場合の IKEv2 のフロー

証明書が IKEv2 で認証に使用される場合、イニシエータは最初のパケットで証明書要求ペイロー

ドを送信しません。

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

レスポンダはこのステージでは使用するプロファイルに関する知識を持たないため、証明書要求ペイロード（第 2 パケット）とすべての CA で応答します。情報を含むパケットがイニシエータに送信されます。

```
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)
```

イニシエータはパケットを処理し、提案された CA と一致するトラストポイントを選択します。

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

イニシエータは、次に、証明書要求と証明書ペイロードの両方を含めて 3 番目のパケットを送信します。このパケットは Diffie-Hellman（DH）フェーズのキー関連情報によってすでに暗号化されています。

```
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)
```

4 番目のパケットはレスポンダからイニシエータに送信され、証明書ペイロードのみを含んでいます。

```
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)
```

ここで説明するフローは、IKEv1 と類似しています。レスポンダは、使用するプロファイルの知識なしで証明書要求ペイロードを事前に送信する必要があります。この結果、前述の IKEv1 における問題と同じ問題が発生します（プロトコルの観点から）。 ただし、IOS の実装は、IKEv1 よりも IKEv2 に適しています。

### イニシエータ用の IKEv2 必須トラストポイント

次に、IKEv2 イニシエータが証明書認証ありでプロファイルの使用を試行し、そのプロファイルにトラストポイントが設定されていない例を示します。

```
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
```

最初のパケットは、前述したように証明書要求ペイロードなしで送信されます。レスポンダからの応答には、グローバル コンフィギュレーション モードで定義されたすべてのトラストポイントの証明書要求ペイロードが含まれています。これがイニシエータによって受信されます。

```
*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
```

イニシエータでは署名に使用する必要のあるトラストポイントがわかりません。これは、IKEv2の実装がIKEv1と比較される主な違いです。IKEv2イニシエータには、IKEv2イニシエータプロファイルでトラストポイントが設定されている必要がありますが、IKEv2レスポンダには必要ありません。

コマンド リファレンスからの抜粋を次に示します。

IKEv2プロファイル設定にトラストポイントが定義されていない場合、デフォルトでは、グローバル設定で定義されているすべてのトラストポイントを使用して証明書を検証します
異なるトラストポイントを定義することができます。署名用に 1 個と検証用に別に 1 個です。残念なことに、IKEv2 プロファイルに必須トラストポイントを設定することでは解決されない問題があります。

## IKEv2 イニシエータとしての R2

この例では、R2 が IKEv2 イニシエータです。

```
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
```

この例では、R1 が IKEv2 レスポンダです。

```
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
```

ここで、R2 は証明書要求なしで最初のパケットを送信します。レスポンダは、設定されているすべてのトラストポイントに対する証明書要求で応答します。ペイロードの順序は IKEv1 と同様であり、インストールされている証明書に依存します。

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

R1 に設定されている最初の証明書は **TP2 トラストポイントと関連付けられているため、最初の証明書要求ペイロードは TP2 トラストポイントと関連付けられている CA 用です。**したがって、R2 では、これを認証用に選択します（最初の一致ルール）。

```
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
```

次に、R2はTP2に関連付けられた証明書要求ペイロードを含む応答 ( パケット3 ) を準備します。R1はTP1トラストポイントに対する検証用に設定されているため、証明書を信頼できません。

```
*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)
```

前述したように、Cisco では単一の IKEv2 プロファイルに複数のトラストポイントを使用しないことを推奨します。複数のトラストポイントを使用する場合は、両側で完全に同じトラストポイントを信頼することを確認する必要があります。たとえば、R1 と R2 では、プロファイルに TP1 および TP2 が設定されています。

# 要約

ここでは、このマニュアルで説明した内容の要約を示します。

証明書要求ペイロードの内容は設定によって異なります。特定のトラストポイントが ISAKMP プロファイル用に設定されており、ルータが ISAKMP イニシエータである場合、MM3 の証明書要求はこのトラストポイントと関連付けられている CA だけを含みます。ただし、同じルータが ISAKMP レスポンダの場合、このルータによって送信される MM4 パケットにはグローバル定義されたすべてのトラストポイント用の複数証明書要求ペイロードが含まれています ( ca trust-point コマンドを考慮しない場合 ) 。 これは、ISAKMP レスポンダでは、MM5 を受信して MM4 に含まれている証明書要求を受信した後で初めて、使用する必要のある ISAKMP プロファイルを特定できるためです。

MM3 および MM4 の証明書要求ペイロードは、最初の一致ルールがあるために重要です。最初の一致ルールは、MM5 と MM6 での認証に必要な証明書の選択に使用されるトラストポイントを決定します。

証明書要求ペイロードの順序は、インストールされている証明書の順序によって決まります。show crypto pki certificate コマンドの出力に表示される最初の証明書の発行元が最初に送信され

**ます。**この最初の証明書は最後に登録された証明書です。

単一の ISAKMP プロファイル用に複数のトラストポイントを設定することができます。これを実行した場合、前述のすべてのルールが引き続き適用されます。

このマニュアルに記載されている問題および警告はすべて、IKEv1 プロトコルの設計によるものです。認証ステージは MM5 と MM6 で発生しますが、認証用の提示（証明書要求）は使用する必要のある ISAKMP プロファイルに関する知識なしで、先行ステージで（事前に）送信するが必要があります。これは Cisco 固有の問題ではなく、IKEv1 プロトコルの設計にある制限に関連しています。

IKEv2 プロトコルは証明書のネゴシエーション プロセスについて IKEv1 と似ています。ただし、IOS での実装では、イニシエータ用に特定のトラストポイントの使用が強制されます。これによって、問題がすべて解決されるわけではありません。単一のプロファイル用に複数のトラストポイントが設定され、反対側で単一のトラストポイントが設定されている場合は、まだ認証に関する問題が発生するおそれがあります。Cisco では接続の両側に対称トラストポイント設定を使用することを推奨します（両方の IKEv2 プロファイルで同じトラストポイントを設定）。

このマニュアルで説明してある内容に関する重要な注意事項を次に示します。

- ピアの IKEv1 プロファイルに対して非対称トラストポイント設定を使用している場合、トンネルは、トンネルの一方の側からだけ開始できることがあります。IKEv1 プロファイルのトラストポイント設定はオプションです。

- ピアの IKEv2 プロファイルに対して非対称トラストポイント設定を使用している場合、トンネルは、トンネルの一方の側からだけ開始できることがあります。IKEv2 プロファイルのトラストポイント設定は、イニシエータでは必須です。

- 証明書要求ペイロードの順序は、**show crypto pki certificate コマンドの出力に表示される証明書の順序によって決まります（最初の一致）。**

- 証明書要求ペイロードの順序によってレスポンダで選択される証明書が決まります（最初の一致）。

- IKEv1 および IKEv2 に複数のプロファイルを使用し、同じ一致 ID ルールが設定されている場合、結果を予測することは困難です（関係する要因が多すぎる）。

- Cisco では IKEv1 と IKEv2 の両方に対称トラストポイント設定を使用することを推奨します。

# 関連情報

- Internet Key Exchange for IPsec VPNs Configuration Guide, Cisco IOS Release 15M&T - Certificate to ISAKMP Profile Mapping
- Cisco IOS Security Command Reference:Commands A to C - ca trust-point through clear eou
- テクニカル サポートとドキュメント – Cisco Systems