# VRF-Aware IPSec

VRF-Aware IPsec 機能には、Multiprotocol Label Switching（MPLS; マルチプロトコル ラベル スイッチング）Virtual Private Network（VPN; バーチャル プライベート ネットワーク）に対する IP Security（IPsec）トンネル マッピングが導入されています。VRF-Aware IPsec 機能を使用すれば、シングル パブリック方向アドレスによって、Virtual Routing and Forwarding（VPN Routing and Forwarding; VPN ルーティング / 転送）に対して IPsec トンネルをマッピングできます。

## 機能情報の入手

ご使用のソフトウェア リリースでは、この章で説明されるすべての機能がサポートされているとは限りません。最新の機能情報と注意事項については、ご使用のプラットフォームとソフトウェア リリースに対応したリリース ノートを参照してください。この章に記載されている機能の詳細、および各機能がサポートされているリリースのリストについては、「VRF-Aware IPsec の機能情報」（P.36）を参照してください。

Cisco Feature Navigator を使用すると、プラットフォーム、および Cisco ソフトウェア イメージの各サポート情報を検索できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

## この章の構成

# VRF-Aware IPsec に関する制約事項

- 暗号マップ設定を使用して VRF-Aware IPsec 機能を設定し、Inside VRF（IVRF）が Front Door VRF（FVRF）とは異なる場合、Unicast Reverse Path Forwarding（uRPF; ユニキャスト RPF）が暗号マップ インターフェイス上でイネーブルになっていると、この機能と uRPF の相互運用はできなくなります。ネットワークに URPF が必要な場合、暗号マップではなく、IPsec の Virtual Tunnel Interface（VTI）を使用することを推奨します。
- VRF-Aware IPsec 機能では、VRF 間における IPsec トンネル マッピングはできません。たとえば、VRF vpn1 から VRF vpn2 への IPsec トンネル マッピングはできません。
- VRF-Aware IPsec 機能を暗号マップと使用した場合、この暗号マップではグローバル VRF を IVRF として使用し、非グローバル VRF を FVRF として使用することはできません。しかし、仮想トンネル インターフェイスに基づく設定にその制限はありません。VTI または Dynamic VTI（DVTI; ダイナミック VTI）を使用した場合、グローバル VRF を IVRF と使用すると同時に、非グローバル VRF を FVRF として使用できます。

# VRF-Aware IPsec に関する情報

VRF-Aware IPsec 機能を使用すれば、IPsec トンネルを MPLS VPN にマッピングできます。この機能を設定および使用するには、次の考えを理解する必要があります。

- 「VRF インスタンス」（P.2）
- 「MPLS 配信プロトコル」（P.2）
- 「VRF-Aware IPsec 機能の概要」（P.2）

## VRF インスタンス

VRF は、VPN ごとのルーティング情報リポジトリであり、Provider Edge（PE; プロバイダー エッジ）ルータに接続されたカスタマー サイトの VPN メンバーシップが定義されています。VRF は、IP ルーティング テーブル、取得された Cisco Express Forwarding（CEF; シスコ エクスプレス フォワーディング）テーブル、転送テーブルを使用する一連のインターフェイス、およびルーティング テーブルに格納されている情報を制御するための一連のルールおよびルーティング プロトコル パラメータで構成されています。各 VPN カスタマーに対して、別個の一連のルーティング テーブルおよび Cisco Express Forwarding（CEF）テーブルが維持されます。

## MPLS 配信プロトコル

MPLS 配信プロトコルは、高性能のパケット転送テクノロジーであり、データ リンク層スイッチングのパフォーマンスおよびトラフィック管理機能と、ネットワーク層ルーティングのスケーラビリティ、柔軟性、およびパフォーマンスが統合されています。

## VRF-Aware IPsec 機能の概要

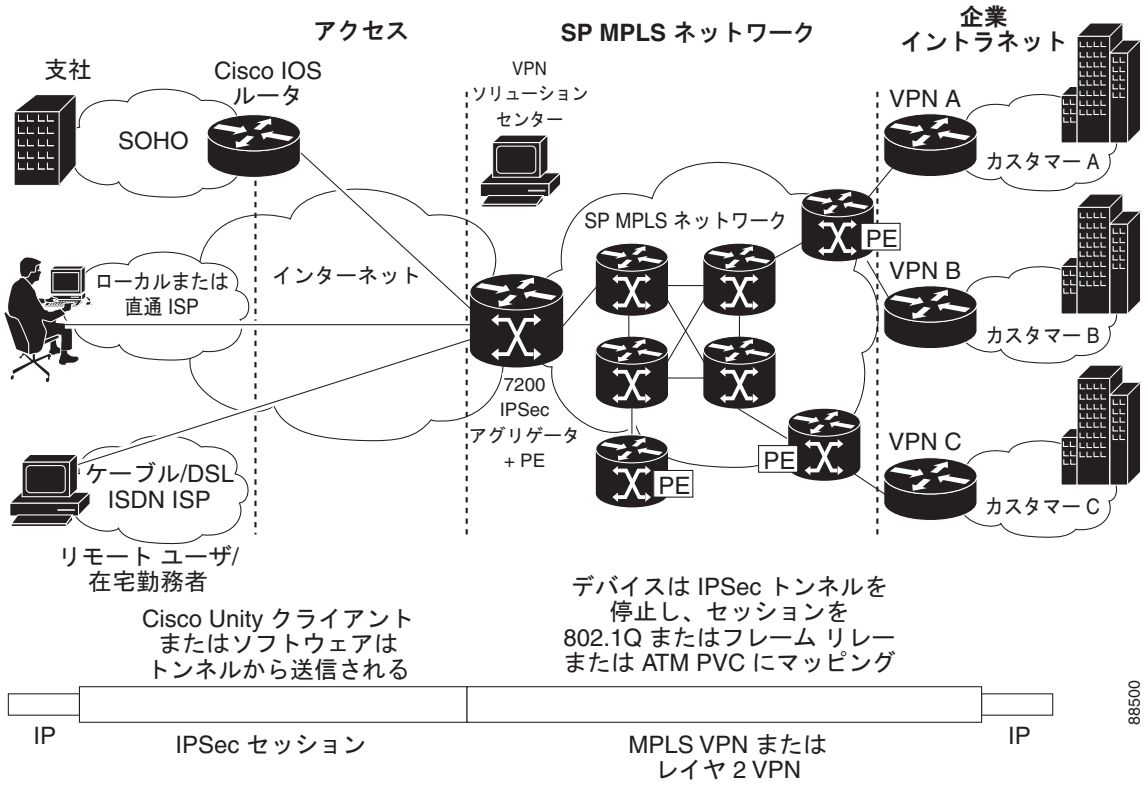
Front Door VRF（FVRF）と Inside VRF（IVRF）が、この機能を理解するうえで重要な概念となります。

各 IPsec トンネルは、2 つの VRF ドメインに関連付けられます。外部のカプセル化されたパケットは 1 つの VRF ドメイン（本マニュアルでは FVRF と呼びます）に所属し、内部の保護された IP パケットは IVRF と呼ばれる別のドメインに所属します。言い換えると、IPsec トンネルのローカル エンドポイントは FVRF に所属し、内部パケットの発信元および宛先アドレスは IVRF に所属します。

1 つ以上の IPsec トンネルを、単一のインターフェイス上で終了できます。これらのトンネルのすべての FVRF は同じものであり、そのインターフェイス上で設定されている VRF に設定されます。これらのトンネルの IVRF は異なる可能性があり、暗号マップエントリに付加された Security Association and Key Management Protocol（ISAKMP）プロファイル内で定義されている VRF に依存します。

図 1 は、MPLS およびレイヤ 2 VPN への IPsec を示すシナリオの図です。

**図 1　　MPLS およびレイヤ 2 VPN への IPsec**




## IPsec トンネルへのパケット フロー

- VPN パケットが、サービス プロバイダー MPLS のバックボーン ネットワークから PE へ到着し、インターネット方向のインターフェイスを介してルーティングされます。
- パケットが Security Policy Database（SPD）と照合され、IPsec カプセル化されます。SPD には、IVRF と Access Control List（ACL; アクセス コントロール リスト）が格納されています。
- 次に、IPsec カプセル化パケットが、FVRF ルーティング テーブルによって転送されます。

## IPsec トンネルからのパケット フロー

- IPsec カプセル化パケットが、リモート IPsec エンドポイントから PE ルータに到着します。

- IPsec によって、Security Parameter Index（SPI; セキュリティ パラメータ インデックス）、宛先、およびプロトコルの Security Association（SA; セキュリティ アソシエーション）検索が実行されます。
- パケットが、SA によってカプセル開放され、IVRF に関連付けられます。
- パケットが、IVRF ルーティング テーブルによって、さらに転送されます。

# VRF-Aware IPsec の設定方法



## クリプト キーリングの設定

クリプト キーリングは、事前共有キー、および、Rivest、Shamir、Adelman（RSA）公開キーのリポジトリです。Cisco IOS ルータ上には、0 以上のキーリングを設定できます。

### 手順の概要

1. **enable**
2. **configure terminal**
3. **crypto keyring** *keyring-name* [**vrf** *fvrf-name*]
4. **description** *string*
5. **pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname*} **key** *key*
6. **rsa-pubkey** {**address** *address* | **name** *fqdn*} [**encryption** | **signature**]
7. **address** *ip-address*
8. **serial-number** *serial-number*
9. **key-string**
10. **text**
11. **quit**
12. **exit**
13. **exit**

## 手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **enable**<br><br>**例：**<br>`Router> enable` | 特権 EXEC モードをイネーブルにします。<br>• プロンプトが表示されたら、パスワードを入力します。 |
| ステップ 2 | **configure terminal**<br><br>**例：**<br>`Router# configure terminal` | グローバル コンフィギュレーション モードを開始します。 |
| ステップ 3 | **crypto keyring** *keyring-name* [**vrf** *fvrf-name*]<br><br>**例：**<br>`Router (config)# crypto keyring VPN1` | キーリングの名前として *keyring-name* を指定してキーリングを定義し、キーリング コンフィギュレーション モードを開始します。<br>• （任意）**vrf** キーワードおよび *fvrf-name* 引数は、キーリングが Front Door Virtual Routing and Forwarding（FVRF）にバインドされることを意味します。ローカル エンドポイントが FVRF 内にある場合、キーリング内のキーが検索されます。**vrf** を指定しない場合、キーリングはグローバルにバインドされます。 |
| ステップ 4 | **description** *string*<br><br>`Router (config-keyring)# description The keys for VPN1` | （任意）キーリングに関する 1 行の説明です。 |
| ステップ 5 | **pre-shared-key** {**address** *address* [*mask*] \| **hostname** *hostname*} **key** *key*<br><br>**例：**<br>`Router (config-keyring)# pre-shared-key address 10.72.23.11 key VPN1` | （任意）アドレスまたはホスト名によって、事前共有キーを定義します。 |
| ステップ 6 | **rsa-pubkey** {**address** *address* \| **name** *fqdn*} [**encryption** \| **signature**]<br><br>**例：**<br>`Router(config-keyring)# rsa-pubkey name host.vpn.com` | （任意）アドレスまたはホスト名によって RSA 公開キーを定義し、rsa-pubkey コンフィギュレーション モードを開始します。<br>• オプションの **encryption** キーワードでは、キーが暗号化のために使用されることが指定されます。<br>• オプションの **encryption** キーワードでは、キーがシグニチャ用に使用されることが指定されます。デフォルトでは、キーはシグニチャ用に使用されます。 |
| ステップ 7 | **address** *ip-address*<br><br>**例：**<br>`Router(config-pubkey-key)# address 10.5.5.1` | （任意）RSA 公開キーの IP アドレスを定義します。 |
| ステップ 8 | **serial-number** *serial-number*<br><br>**例：**<br>`Router(config-pubkey-key)# serial-number 1000000` | （任意）公開キーのシリアル番号を指定します。値は 0 から始まり、無制限です。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 9 | **key-string**<br><br>**例：**<br>`Router (config-pubkey-key)# key-string` | 公開キーを定義するためのテキスト モードを開始します。 |
| ステップ 10 | **text**<br><br>**例：**<br>`Router (config-pubkey)# 00302017 4A7D385B 1234EF29 335FC973` | 公開キーを指定します。<br>**（注）** この手順で追加できる公開キーは 1 つだけです。 |
| ステップ 11 | **quit**<br><br>**例：**<br>`Router (config-pubkey)# quit` | 公開キー コンフィギュレーション モードを終了します。 |
| ステップ 12 | **exit**<br><br>**例：**<br>`Router (config-pubkey)# exit` | キーリング コンフィギュレーション モードに戻ります。 |
| ステップ 13 | **exit**<br><br>**例：**<br>`Router(config-keyring)# exit#` | グローバル コンフィギュレーション モードに戻ります。 |

# ISAKMP プロファイルの設定

ISAKMP プロファイルは、一連のピアの Internet Key Exchange（IKE; インターネット キー エクスチェンジ）フェーズ 1 および IKE フェーズ 1.5 設定のリポジトリです。ISAKMP プロファイルでは、IKE フェーズ 1 およびフェーズ 1.5 交換中に、キープアライブ、トラストポイント、ピアの ID、および XAUTH AAA リストなどのアイテムが定義されます。Cisco IOS ルータ上には、0 以上の ISAKMP プロファイルを設定できます。

**（注）**

- ルータから Certification Authority（CA; 認証局）へのトラフィック（認証および登録用、または、証明書失効リスト（CRL）所得用）、または Lightweight Directory Access Protocol（LDAP）サーバへのトラフィック（CRL 取得用）を VRF を介してルーティングする必要がある場合、トラストポイントに **vrf** コマンドを追加する必要があります。追加しない場合、トラフィックはデフォルトのルーティング テーブルを使用します。
- プロファイルに 1 つ以上のトラストポイントが指定されていない場合、ルータ内のすべてのトラストポイントが使用されて、ピアの証明書の確認が試行されます（IKE メイン モードまたはシグニチャ認証）。1 つ以上のトラストポイントが指定されている場合、それらのトラストポイントだけが使用されます。

## 制約事項

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 シグニ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、証明書の確認のために 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 の発信側）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）。

### 手順の概要

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name*
4. **description** *string*
5. **vrf** *ivrf-name*
6. **keepalive** *seconds* **retry** *retry-seconds*
7. **self-identity** {**address** | **fqdn** | **user-fqdn** *user-fqdn*}
8. **keyring** *keyring-name*
9. **ca trust-point** *trustpoint-name*
10. **match identity** {**group** *group-name* | **address** *address* [*mask*] [*fvrf*] | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*}
11. **client configuration address** {**initiate** | **respond**}
12. **client authentication list** *list-name*
13. **isakmp authorization list** *list-name*
14. **initiate mode aggressive**
15. **exit**

### 手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **enable**<br><br>**例：**<br>`Router> enable` | 特権 EXEC モードをイネーブルにします。<br>• プロンプトが表示されたら、パスワードを入力します。 |
| ステップ 2 | **configure terminal**<br><br>**例：**<br>`Router# configure terminal` | グローバル コンフィギュレーション モードを開始します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 3 | **crypto isakmp profile** *profile-name*<br><br>**例：**<br>`Router (config)# crypto isakmp profile vpnprofile` | Internet Security Association and Key Management Protocol（ISAKMP）プロファイルを定義し、isakmp プロファイル コンフィギュレーション モードを開始します。 |
| ステップ 4 | **description** *string*<br><br>**例：**<br>`Router (conf-isa-prof)# description configuration for VPN profile` | （任意）ISAKMP プロファイルの 1 行の説明を指定します。 |
| ステップ 5 | **vrf** *ivrf-name*<br><br>**例：**<br>`Router (conf-isa-prof)# vrf VPN1` | （任意）IPsec トンネルを Virtual Routing and Forwarding（VRF）インスタンスにマッピングします。<br>**（注）** VRF は、Security Policy Database（SPD）の照合のためのマッチングのためのセレクタにもなります。VRF が ISAKMP プロファイル内で指定されていない場合、IPsec トンネルの IVRF は、その FVRF と同じになります。 |
| ステップ 6 | **keepalive** *seconds* **retry** *retry-seconds*<br><br>**例：**<br>`Router (conf-isa-prof)# keepalive 60 retry 5` | （任意）ゲートウェイに対して、Dead Peer Detection（DPD）メッセージのピアへの送信を許可します。<br>• 定義しない場合、ゲートウェイではグローバル コンフィギュレーション値が使用されます。<br>• *seconds*：DPD メッセージ間の秒数。指定できる範囲は 10 ～ 3600 秒です。<br>• **retry** *retry-seconds*：DPD メッセージがエラーになった場合の、リトライ間の秒数 指定できる範囲は 2 ～ 60 秒です。 |
| ステップ 7 | **self-identity** {**address** \| **fqdn** \| **user-fqdn** *user-fqdn*}<br><br>**例：**<br>`Router (conf-isa-prof)# self-identity address` | （任意）ローカル IKE によって、リモート ピアに対して IKE 自身を識別させるために使用される、ID を指定します。<br>• 定義しない場合、IKE ではグローバル コンフィギュレーション値が使用されます。<br>• **address**：出力インターフェイスの IP アドレスを使用します。<br>• **fqdn**：ルータの Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）を使用します。<br>• **user-fqdn**：指定した値を使用します。 |
| ステップ 8 | **keyring** *keyring-name*<br><br>**例：**<br>`Router (conf-isa-prof)# keyring VPN1` | （任意）フェーズ 1 認証用に使用するキーリングを指定します。<br>• キーリングを指定しない場合、グローバル キー定義が使用されます。 |
| ステップ 9 | **ca trust-point** {*trustpoint-name*}<br><br>**例：**<br>`Router (conf-isa-prof)# ca trustpoint VPN1-trustpoint` | （任意）Rivest、Shamir、Adelman（RSA）証明書を確認するためのトラストポイントを指定します。<br>• ISAKMP プロファイル内でトラストポイントが指定されていない場合、Cisco IOS ルータ上で設定されているすべてのトラストポイントが証明書の確認に使用されます。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 10 | **match identity** {**group** *group-name* \| **address** *address* [*mask*] [*fvrf*] \| **host** *host-name* \| **host domain** *domain-name* \| **user** *user-fqdn* \| **user domain** *domain-name*}<br><br>**例：**<br>`Router (conf-isa-prof)# match identity address 10.1.1.1` | 照合されるクライアント IKE の ID を指定します。<br>• **group** *group-name* : *group-name* と ID タイプ ID_KEY_ID を照合します。また、*group-name* と Distinguished Name（DN; 認定者名）の Organizational Unit（OU; 組織ユニット）フィールドも照合します。<br>• **address** *address* [*mask*] *fvrf* : *address* と ID タイプ ID_IPV4_ADDR を照合します。*mask* 引数を使用して、アドレスの範囲を指定できます。*fvrf* 引数では、アドレスが Front Door Virtual Routing and Forwarding（FVRF）にあることを指定します。<br>• **host** *hostname* : *hostname* と ID タイプ ID_FQDN を照合します。<br>• **host domain** *domainname* : *domainname* を、ドメイン名が *domainname* と同じ IP タイプ ID_FQDN と照合します。このコマンドを使用して、ドメイン内のすべてのホストを照合します。<br>• **user** *username* : *username* と ID タイプ ID_USER_FQDN を照合します。<br>• **user domain** *domainname* : ドメイン名が *domainname* と一致する ID タイプ ID_USER_FQDN を照合します。 |
| ステップ 11 | **client configuration address** {**initiate** \| **respond**}<br><br>**例：**<br>`Router (conf-isa-prof)# client configuration address initiate` | （任意）モード設定交換を開始するか、モード設定要求に応答するかを指定します。 |
| ステップ 12 | **client authentication list** *list-name*<br><br>**例：**<br>`Router (conf-isa-prof)# client authentication list xauthlist` | （任意）Extended Authentication（XAUTH）交換中にリモート クライアントを認証するために使用する Authentication, Authorization, and Accounting（AAA; 認証、認可、アカウンティング）。 |
| ステップ 13 | **isakmp authorization list** *list-name*<br><br>**例：**<br>`Router (conf-isa-prof)# isakmp authorization list ikessaaalist` | （任意）フェーズ 1 キーおよびその他の AV のペアを受信するためのネットワーク認証サーバ。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 14 | **initiate mode aggressive**<br><br>**例：**<br>`Router (conf-isa-prof)# initiate mode aggressive` | （任意）アグレッシブ モード交換を開始します。<br>• 指定しない場合、IKE によって、メイン モード交換が常に開始されます。 |
| ステップ 15 | **exit**<br><br>**例：**<br>`Router (conf-isa-prof)# exit` | グローバル コンフィギュレーション モードに戻ります。 |

## 次の作業

「暗号マップ上における ISAKMP プロファイルの設定」（P.10）の項を参照してください。

# 暗号マップ上における ISAKMP プロファイルの設定

ISAKMP プロファイルを、暗号マップに適用する必要があります。ISAKMP プロファイル上の IVRF は、VPN トラフィックの照合時にセレクタとして使用されます。ISAKMP プロファイル上に IVRF が存在しない場合、IVRF は FVRF と同じになります。暗号マップ上の ISAKMP プロファイルを設定するには、次の作業を実行します。

## 前提条件

暗号マップ上で ISAKMP プロファイルを設定する前に、ルータに対して基本 IPsec の設定を行っておく必要があります。

### 手順の概要

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name* **isakmp-profile** *isakmp-profile-name*
4. **set isakmp-profile** *profile-name*
5. **exit**

### 手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **enable**<br><br>**例：**<br>`Router> enable` | 特権 EXEC モードをイネーブルにします。<br>• プロンプトが表示されたら、パスワードを入力します。 |
| ステップ 2 | **configure terminal**<br><br>**例：**<br>`Router# configure terminal` | グローバル コンフィギュレーション モードを開始します。 |

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 3 | **crypto map** *map-name* **isakmp-profile** *isakmp-profile-name*<br><br>**例：**<br>`Router (config)# crypto map vpnmap isakmp-profile vpnprofile` | （任意）暗号マップ セット用に Internet Key Exchange and Key Management Protocol（ISAKMP）プロファイルを指定し、暗号マップ コンフィギュレーション モードを開始します。<br>• ISAKMP プロファイルは、IKE 交換中に使用されます。 |
| ステップ 4 | **set isakmp-profile** *profile-name*<br><br>**例：**<br>`Router (config-crypto-map)# set isakmp-profile vpnprofile` | （任意）トラフィックが暗号マップ エントリと一致した際に使用する ISAKMP プロファイルを指定します。 |
| ステップ 5 | **exit**<br><br>**例：**<br>`Router (config-crypto-map)# exit` | グローバル コンフィギュレーション モードに戻ります。 |

# IKE フェーズ 1 ネゴシエーション中に拡張認証を無視するように設定

IKE フェーズ 1 ネゴシエーション中に XAUTH を無視するには、**no crypto xaut** コマンドを使用します。Unity クライアントの拡張認証が不要な場合、**no crypto xauth** コマンドを使用します。

## 手順の概要

1. **enable**
2. **configure terminal**
3. **no crypto xauth** *interface*

## 手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **enable**<br><br>**例：**<br>`Router> enable` | 特権 EXEC モードをイネーブルにします。<br>• プロンプトが表示されたら、パスワードを入力します。 |
| ステップ 2 | **configure terminal**<br><br>**例：**<br>`Router# configure terminal` | グローバル コンフィギュレーション モードを開始します。 |
| ステップ 3 | **no crypto xauth** *interface*<br><br>**例：**<br>`Router(config)# no crypto xauth ethernet0` | インターフェイスの IP アドレスを宛先とする要求の XAUTH 提案を無視します。デフォルトでは、IKE によって、XAUTH 提案が処理されます。 |

# VRF-Aware IPsec の確認

VRF-Aware IPsec の設定を確認するには、次の **show** コマンドを使用します。これらの **show** コマンドによって、設定情報および Security Association（SA; セキュリティ アソシエーション）を表示できます。

### 手順の概要

1. **enable**
2. **show crypto ipsec sa** [**map** *map-name* | **address** | **identity | interface** *interface* | **peer** [**vrf** *fvrf-name*] **address | vrf** *ivrf-name*] [**detail**]
3. **show crypto isakmp key**
4. **show crypto isakmp profile**
5. **show crypto key pubkey-chain rsa**

### 手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **enable**<br><br>**例：**<br>`Router> enable` | 特権 EXEC モードをイネーブルにします。<br>• プロンプトが表示されたら、パスワードを入力します。 |
| ステップ 2 | **show crypto ipsec sa** [**map** *map-name* \| **address** \| **identity** \| **interface** *interface* \| **peer** [**vrf** *fvrf-name*] **address** \| **vrf** *ivrf-name*] [**detail**]<br><br>**例：**<br>`Router# show crypto ipsec sa vrf vpn1` | 現在の SA によって使用される設定の表示を許可します。 |
| ステップ 3 | **show crypto isakmp key**<br><br>**例：**<br>`Router# show crypto isakmp key` | すべてのキーリングおよびその事前共有キーを一覧表示します。<br>• このコマンドを使用して、クリプト キーリング設定を確認します。 |
| ステップ 4 | **show crypto isakmp profile**<br><br>**例：**<br>`Router# show crypto isakmp profile` | すべての ISAKMP プロファイルおよびその設定を一覧表示します。 |
| ステップ 5 | **show crypto key pubkey-chain rsa**<br><br>**例：**<br>`Router# show crypto key pubkey-chain rsa` | ルータに保存されている、ピアの RSA 公開キーを表示します。<br>• 出力が、公開キーが所属するキーリングを表示するように拡張されます。 |

# セキュリティ アソシエーションのクリア

次の **clear** コマンドによって、SA をクリアできます。

### 手順の概要

1. **enable**
2. **clear crypto sa** [**counters** | **map** *map-name* | **peer** [**vrf** *fvrf-name*] *address* | **spi** *address* {**ah** | **esp**} *spi* | **vrf** *ivrf-name*]

### 手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **enable**<br><br>**例：**<br>`Router> enable` | 特権 EXEC モードをイネーブルにします。<br>• プロンプトが表示されたら、パスワードを入力します。 |
| ステップ 2 | **clear crypto sa** [**counters** \| **map** *map-name* \| **peer** [**vrf** *fvrf-name*] *address* \| **spi** *address* {**ah** \| **esp**} *spi* \| **vrf** *ivrf-name*]<br><br>**例：**<br>`Router# clear crypto sa vrf VPN1` | IPsec SA をクリアします。 |

# VRF-Aware IPsec のトラブルシューティング

VRF-Aware IPsec のトラブルシューティングを行うには、次の **debug** コマンドを使用します。

### 手順の概要

1. **enable**
2. **debug crypto ipsec**
3. **debug crypto isakmp**

### 手順の詳細

| | コマンドまたはアクション | 目的 |
|---|---|---|
| ステップ 1 | **enable**<br><br>**例：**<br>`Router> enable` | 特権 EXEC モードをイネーブルにします。<br>• プロンプトが表示されたら、パスワードを入力します。 |
| ステップ 2 | **debug crypto ipsec**<br><br>**例：**<br>`Router# debug crypto ipsec` | IP security（IPsec; IP セキュリティ）イベントを表示します。 |
| ステップ 3 | **debug crypto isakmp**<br><br>**例：**<br>`Router(config)# debug crypto isakmp` | IKE に関するメッセージを表示します。 |

# VRF-Aware IPsec のデバッグ例

次に、VRF-aware IPsec 設定のサンプル デバッグ出力を示します。

## IPsec PE

```
Router# debug crypto ipsec

Crypto IPSEC debugging is on
IPSEC-PE#debug crypto isakmp
Crypto ISAKMP debugging is on
IPSEC-PE#debug crypto isakmp d
04:31:28: ISAKMP (0:12): purging SA., sa=6482B354, delme=6482B354
04:31:28: ISAKMP: Unlocking IKE struct 0x63C142F8 for declare_sa_dead(), count 0
IPSEC-PE#debug crypto isakmp detail
Crypto ISAKMP internals debugging is on
IPSEC-PE#
IPSEC-PE#
IPSEC-PE#
04:32:07: ISAKMP: Deleting peer node by peer_reap for 10.1.1.1: 63C142F8
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B DC887D4E
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.68.1.1
04:32:55: ISAKMP cookie AA8F7B41 49A60E88
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B DBC8E125
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 B4BDB5B7
04:32:55: ISAKMP (0:0): received packet from 10.1.1.1 dport 500 sport 500 Global (N) NEW
SA
04:32:55: ISAKMP: local port 500, remote port 500
04:32:55: ISAKMP: hash from 729FA94 for 619 bytes
04:32:55: ISAKMP: Packet hash:
64218CC0:                   B91E2C70 095A1346          9.,p.Z.F
64218CD0: 0EDB4CA6 8A46784F B314FD3B 00        .[L&.FxO.};.
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:32:55: ISAKMP cookie AA8F7B41 F7ACF384
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:32:55: ISAKMP cookie AA8F7B41 0C07C670
04:32:55: ISAKMP: insert sa successfully sa = 6482B354
04:32:55: ISAKMP (0:13): processing SA payload. message ID = 0
04:32:55: ISAKMP (0:13): processing ID payload. message ID = 0
04:32:55: ISAKMP (0:13): peer matches vpn2-ra profile
04:32:55: ISAKMP: Looking for a matching key for 10.1.1.1 in default
04:32:55: ISAKMP: Created a peer struct for 10.1.1.1, peer port 500
04:32:55: ISAKMP: Locking peer struct 0x640BBB18, IKE refcount 1 for
crypto_ikmp_config_initialize_sa
04:32:55: ISAKMP (0:13): Setting client config settings 648252B0
04:32:55: ISAKMP (0:13): (Re)Setting client xauth list and state
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 157 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 123 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v2
04:32:55: ISAKMP (0:13) Authentication by xauth preshared
04:32:55: ISAKMP (0:13): Checking ISAKMP transform 1 against priority 1 policy
04:32:55: ISAKMP:      encryption 3DES-CBC
04:32:55: ISAKMP:      hash SHA
04:32:55: ISAKMP:      default group 2
04:32:55: ISAKMP:      auth XAUTHInitPreShared
04:32:55: ISAKMP:      life type in seconds
04:32:55: ISAKMP:      life duration (VPI) of 0x0 0x20 0xC4 0x9B
```

```
04:32:55: ISAKMP (0:13): atts are acceptable. Next payload is 3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 157 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v3
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 123 mismatch
04:32:55: ISAKMP (0:13): vendor ID is NAT-T v2
04:32:55: ISAKMP (0:13): processing KE payload. message ID = 0
04:32:55: ISAKMP (0:13): processing NONCE payload. message ID = 0
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID is DPD
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID seems Unity/DPD but major 175 mismatch
04:32:55: ISAKMP (0:13): vendor ID is XAUTH
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): claimed IOS but failed authentication
04:32:55: ISAKMP (0:13): processing vendor id payload
04:32:55: ISAKMP (0:13): vendor ID is Unity
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_AM_EXCH
04:32:55: ISAKMP (0:13): Old State = IKE_READY  New State = IKE_R_AM_AAA_AWAIT

04:32:55: ISAKMP cookie gen for src 11.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 7AE6E1DF
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 4 AA 31 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP: got callback 1
04:32:55: ISAKMP (0:13): SKEYID state generated
04:32:55: ISAKMP: Unity/DPD ID: vendor_id_payload:
        next: 0xD, reserved: 0x0, len 0x14
04:32:55: ISAKMP: Unity/DPD ID payload dump:
63E66D70:                           0D000014             ....
63E66D80: 12F5F28C 457168A9 702D9FE2 74CC0100  .ur.Eqh)p-.btL..
63E66D90: 00                                   .
04:32:55: ISAKMP: Unity/DPD ID: vendor_id_payload:
        next: 0xD, reserved: 0x0, len 0x14
04:32:55: ISAKMP: Unity/DPD ID payload dump:
63E66D90: 0D000014 AFCAD713 68A1F1C9 6B8696FC  ..../JW.h!qIk..|
63E66DA0: 77570100 00                          wW...
04:32:55: ISAKMP (0:13): constructed NAT-T vendor-03 ID
04:32:55: ISAKMP (0:13): SA is doing pre-shared key authentication plus XAUTH using id
type ID_IPV4_ADDR
04:32:55: ISAKMP (13): ID payload
        next-payload : 10
        type         : 1
        addr         : 172.16.1.1
        protocol     : 17
        port         : 0
        length       : 8
04:32:55: ISAKMP (13): Total payload length: 12
04:32:55: ISAKMP (0:13): constructed HIS NAT-D
04:32:55: ISAKMP (0:13): constructed MINE NAT-D
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
AG_INIT_EXCH
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, PRESHARED_KEY_REPLY
04:32:55: ISAKMP (0:13): Old State = IKE_R_AM_AAA_AWAIT  New State = IKE_R_AM2

04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B D99DA70D
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 9C69F917
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 5 21FF 1 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
```

```
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B 00583224
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 C1B006EE
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 5 21FF 1 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
AG_INIT_EXCH
04:32:55: ISAKMP: hash from 7003A34 for 132 bytes
04:32:55: ISAKMP: Packet hash:
64218CC0:                   D1202D99 2BB49D38          Q -.+4.8
64218CD0: B8FBB1BE 7CDC67D7 4E26126C 63        8{1>|\gWN&.lc
04:32:55: ISAKMP (0:13): processing HASH payload. message ID = 0
04:32:55: ISAKMP:received payload type 17
04:32:55: ISAKMP (0:13): Detected NAT-D payload
04:32:55: ISAKMP (0:13): recalc my hash for NAT-D
04:32:55: ISAKMP (0:13): NAT match MINE hash
04:32:55: ISAKMP:received payload type 17
04:32:55: ISAKMP (0:13): Detected NAT-D payload
04:32:55: ISAKMP (0:13): recalc his hash for NAT-D
04:32:55: ISAKMP (0:13): NAT match HIS hash
04:32:55: ISAKMP (0:13): processing NOTIFY INITIAL_CONTACT protocol 1
        spi 0, message ID = 0, sa = 6482B354
04:32:55: ISAKMP (0:13): Process initial contact,
bring down existing phase 1 and 2 SA's with local 172.16.1.1 remote 10.1.1.1 remote port
500
04:32:55: ISAKMP (0:13): returning IP addr to the address pool
04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 05D315C5
04:32:55: ISAKMP cookie gen for src 172.16.1.1 dst 10.1.1.1
04:32:55: ISAKMP cookie 3123100B 041A85A6
04:32:55: ISAKMP (0:13): SA has been authenticated with 10.1.1.1
04:32:55: ISAKMP: Trying to insert a peer 172.16.1.1/10.1.1.1/500/,  and inserted
successfully.
04:32:55: ISAKMP: set new node -803402627 to CONF_XAUTH
04:32:55: IPSEC(key_engine): got a queue event...
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) QM_IDLE
04:32:55: ISAKMP (0:13): purging node -803402627
04:32:55: ISAKMP: Sending phase 1 responder lifetime 86400

04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_AM_EXCH
04:32:55: ISAKMP (0:13): Old State = IKE_R_AM2  New State = IKE_P1_COMPLETE

04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.168.1.1
04:32:55: ISAKMP cookie AA8F7B41 25EEF256
04:32:55: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP (0:13): Need XAUTH
04:32:55: ISAKMP (0:13): Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
04:32:55: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State =
IKE_XAUTH_AAA_START_LOGIN_AWAIT

04:32:55: ISAKMP cookie gen for src 10.1.1.1 dst 172.16.1.1
04:32:55: ISAKMP cookie AA8F7B41 2CCFA491
04:32:55: ISAKMP:        isadb_post_process_list: crawler: B 27FF 12 (6482B354)
04:32:55:       crawler my_cookie AA8F7B41 F7ACF384
04:32:55:        crawler his_cookie E46E088D F227FE4D
04:32:55: ISAKMP: got callback 1
04:32:55: ISAKMP: set new node -1447732198 to CONF_XAUTH
04:32:55: ISAKMP/xauth: request attribute XAUTH_USER_NAME_V2
04:32:55: ISAKMP/xauth: request attribute XAUTH_USER_PASSWORD_V2
04:32:55: ISAKMP (0:13): initiating peer config to 10.1.1.1. ID = -1447732198
```

```
04:32:55: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_XAUTH
04:32:55: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_START_LOGIN
04:32:55: ISAKMP (0:13): Old State = IKE_XAUTH_AAA_START_LOGIN_AWAIT  New State =
IKE_XAUTH_REQ_SENT

04:33:00: ISAKMP (0:13): retransmitting phase 2 CONF_XAUTH    -1447732198 ...
04:33:00: ISAKMP (0:13): incrementing error counter on sa: retransmit phase 2
04:33:00: ISAKMP (0:13): incrementing error counter on sa: retransmit phase 2
04:33:00: ISAKMP (0:13): retransmitting phase 2 -1447732198 CONF_XAUTH
04:33:00: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_XAUTH
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 124D4618
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 B0C91917
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 0E294692
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 091A7695
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
CONF_XAUTH
04:33:03: ISAKMP: hash from 7292D74 for 92 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                    84A1AF24 5D92B116          .!/$].1.
64218CD0: FC2C6252 A472C5F8 152AC860 63        |,bR$rEx.*H`c
04:33:03: ISAKMP (0:13): processing transaction payload from 11.1.1.1. message ID =
-1447732198
04:33:03: ISAKMP: Config payload REPLY
04:33:03: ISAKMP/xauth: reply attribute XAUTH_USER_NAME_V2
04:33:03: ISAKMP/xauth: reply attribute XAUTH_USER_PASSWORD_V2
04:33:03: ISAKMP (0:13): deleting node -1447732198 error FALSE reason "done with xauth
request/reply exchange"
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_REPLY
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_REQ_SENT  New State =
IKE_XAUTH_AAA_CONT_LOGIN_AWAIT

04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 A1B3E684
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 12 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP: got callback 1
04:33:03: ISAKMP: set new node 524716665 to CONF_XAUTH
04:33:03: ISAKMP (0:13): initiating peer config to 10.1.1.1. ID = 524716665
04:33:03: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_XAUTH
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_CONT_LOGIN
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_AAA_CONT_LOGIN_AWAIT  New State =
IKE_XAUTH_SET_SENT
004:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 5C83A09D
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 2BEBEFD4
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
```

```
04:33:03: ISAKMP cookie 3123100B DA00A46B
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 FDD27773
04:33:03: ISAKMP:        isadb_post_process_list: crawler: B 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
CONF_XAUTH
04:33:03: ISAKMP: hash from 7292A34 for 68 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   5034B99E B8BA531F          P49.8:S.
64218CD0: 6267B8BD F3006989 DC118796 63        bg8=s.i.\...c
04:33:03: ISAKMP (0:13): processing transaction payload from 11.1.1.1. message ID =
524716665
04:33:03: ISAKMP: Config payload ACK
04:33:03: ISAKMP (0:13):        XAUTH ACK Processed
04:33:03: ISAKMP (0:13): deleting node 524716665 error FALSE reason "done with
transaction"
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_ACK
04:33:03: ISAKMP (0:13): Old State = IKE_XAUTH_SET_SENT  New State = IKE_P1_COMPLETE

04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 E0BB50E9
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP (0:13): Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
04:33:03: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State = IKE_P1_COMPLETE

04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 7794EF6E
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 C035AAE5
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B F1FCC25A
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 31744F44
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F207FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
QM_IDLE
04:33:03: ISAKMP: set new node -1639992295 to QM_IDLE
04:33:03: ISAKMP: hash from 7293A74 for 100 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                   9D7DF4DF FE3A6403          .}t_~:d.
64218CD0: 3F1D1C59 C5D138CE 50289B79 07        ?..YEQ8NP(.y.
04:33:03: ISAKMP (0:13): processing transaction payload from 10.1.1.1. message ID =
-1639992295
04:33:03: ISAKMP: Config payload REQUEST
04:33:03: ISAKMP (0:13): checking request:
04:33:03: ISAKMP:    IP4_ADDRESS
04:33:03: ISAKMP:    IP4_NETMASK
04:33:03: ISAKMP:    IP4_DNS
04:33:03: ISAKMP:    IP4_DNS
04:33:03: ISAKMP:    IP4_NBNS
04:33:03: ISAKMP:    IP4_NBNS
04:33:03: ISAKMP:    SPLIT_INCLUDE
04:33:03: ISAKMP:    DEFAULT_DOMAIN
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_PEER, IKE_CFG_REQUEST
```

```
04:33:03: ISAKMP (0:13): Old State = IKE_P1_COMPLETE  New State =
IKE_CONFIG_AUTHOR_AAA_AWAIT

04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 B02E0D67
04:33:03: ISAKMP:        isadb_post_process_list: crawler: C 27FF 12 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP: got callback 1
04:33:03: ISAKMP (0:13): attributes sent in message:
04:33:03:         Address: 10.2.0.0
04:33:03: ISAKMP (0:13): allocating address 10.4.1.4
04:33:03: ISAKMP: Sending private address: 10.4.1.4
04:33:03: ISAKMP: Sending DEFAULT_DOMAIN default domain name: vpn2.com
04:33:03: ISAKMP (0:13): responding to peer config from 10.1.1.1. ID = -1639992295
04:33:03: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R)
CONF_ADDR
04:33:03: ISAKMP (0:13): deleting node -1639992295 error FALSE reason ""
04:33:03: ISAKMP (0:13): Input = IKE_MESG_FROM_AAA, IKE_AAA_GROUP_ATTR
04:33:03: ISAKMP (0:13): Old State = IKE_CONFIG_AUTHOR_AAA_AWAIT  New State =
IKE_P1_COMPLETE

04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 881D5411
04:33:03: ISAKMP cookie gen for src 11.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 6FD82541
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F227FE4D
04:33:03: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:03: ISAKMP cookie 3123100B 8A94C1BE
04:33:03: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:03: ISAKMP cookie AA8F7B41 F3BA766D
04:33:03: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:03:       crawler my_cookie AA8F7B41 F7ACF384
04:33:03:        crawler his_cookie E46E088D F207FE4D
04:33:03: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
QM_IDLE
04:33:03: ISAKMP: set new node 17011691 to QM_IDLE
04:33:03: ISAKMP: hash from 70029F4 for 540 bytes
04:33:03: ISAKMP: Packet hash:
64218CC0:                    AFBA30B2 55F5BC2D          /:02Uu<-
64218CD0: 3A86B1C9 00D2F5BA 77BF5589 07        :.1I.Ru:w?U..
04:33:03: ISAKMP (0:13): processing HASH payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing SA payload. message ID = 17011691
04:33:03: ISAKMP (0:13): Checking IPSec proposal 1
04:33:03: ISAKMP: transform 1, ESP_3DES
04:33:03: ISAKMP:   attributes in transform:
04:33:03: ISAKMP:      encaps is 1
04:33:03: ISAKMP:      SA life type in seconds
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x20 0xC4 0x9B
04:33:03: ISAKMP:      SA life type in kilobytes
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
04:33:03: ISAKMP:      authenticator is HMAC-SHA
04:33:03: ISAKMP (0:13): atts are acceptable.
04:33:03: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-sha-hmac,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
```

```
04:33:03: IPSEC(validate_transform_proposal): transform proposal not supported for
identity:
    {esp-3des esp-sha-hmac}
04:33:03: ISAKMP (0:13): IPSec policy invalidated proposal
04:33:03: ISAKMP (0:13): Checking IPSec proposal 2
04:33:03: ISAKMP: transform 1, ESP_3DES
04:33:03: ISAKMP:   attributes in transform:
04:33:03: ISAKMP:      encaps is 1
04:33:03: ISAKMP:      SA life type in seconds
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x20 0xC4 0x9B
04:33:03: ISAKMP:      SA life type in kilobytes
04:33:03: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
04:33:03: ISAKMP:      authenticator is HMAC-MD5
04:33:03: ISAKMP (0:13): atts are acceptable.
04:33:03: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-md5-hmac,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:03: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
04:33:03: ISAKMP (0:13): processing NONCE payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing ID payload. message ID = 17011691
04:33:03: ISAKMP (0:13): processing ID payload. message ID = 17011691
04:33:03: ISAKMP (0:13): asking for 1 spis from ipsec
04:33:03: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
04:33:03: ISAKMP (0:13): Old State = IKE_QM_READY  New State = IKE_QM_SPI_STARVE
04:33:03: IPSEC(key_engine): got a queue event...
04:33:03: IPSEC(spi_response): getting spi 2749516541 for SA
        from 172.18.1.1     to 10.1.1.1        for prot 3
04:33:03: ISAKMP: received ke message (2/1)
04:33:04: ISAKMP (13): ID payload
        next-payload : 5
        type         : 1
        addr         : 10.4.1.4
        protocol     : 0
        port         : 0
04:33:04: ISAKMP (13): ID payload
        next-payload : 11
        type         : 4
        addr         : 0.0.0.0
        protocol     : 0
        port         : 0
04:33:04: ISAKMP (0:13): sending packet to 10.1.1.1 my_port 500 peer_port 500 (R) QM_IDLE
04:33:04: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_IPSEC, IKE_SPI_REPLY
04:33:04: ISAKMP (0:13): Old State = IKE_QM_SPI_STARVE  New State = IKE_QM_R_QM2
04:33:04: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:04: ISAKMP cookie 3123100B 93DE46D2
04:33:04: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:04: ISAKMP cookie AA8F7B41 088A0A16
04:33:04: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:04:       crawler my_cookie AA8F7B41 F7ACF384
04:33:04:        crawler his_cookie E46E088D F227FE4D
04:33:04: ISAKMP cookie gen for src 172.18.1.1 dst 10.1.1.1
04:33:04: ISAKMP cookie 3123100B A8F23F73
04:33:04: ISAKMP cookie gen for src 10.1.1.1 dst 172.18.1.1
04:33:04: ISAKMP cookie AA8F7B41 93D8D879
04:33:04: ISAKMP:        isadb_post_process_list: crawler: 9 27FF 2 (6482B354)
04:33:04:       crawler my_cookie AA8F7B41 F7ACF384
04:33:04:        crawler his_cookie E46E088D F227FE4D
04:33:04: ISAKMP (0:13): received packet from 10.1.1.1 dport 500 sport 500 Global (R)
QM_IDLE
```

```
04:33:04: ISAKMP: hash from 7290DB4 for 60 bytes
04:33:04: ISAKMP: Packet hash:
64218CC0:                   4BB45A92 7181A2F8          K4Z.q."x
64218CD0: 73CC12F8 091875C0 054F77CD 63        sL.x..u@.OwMc
04:33:04: ISAKMP: Locking peer struct 0x640BBB18, IPSEC refcount 1 for stuff_ke
04:33:04: ISAKMP (0:13): Creating IPSec SAs
04:33:04:         inbound SA from 10.1.1.1 to 172.18.1.1 (f/i)  0/ 2
        (proxy 10.4.1.4 to 0.0.0.0)
04:33:04:         has spi 0xA3E24AFD and conn_id 5127 and flags 2
04:33:04:         lifetime of 2147483 seconds
04:33:04:         lifetime of 4608000 kilobytes
04:33:04:         has client flags 0x0
04:33:04:         outbound SA from 172.18.1.1     to 10.1.1.1        (f/i)  0/ 2 (proxy
0.0.0.0         to 10.4.1.4       )
04:33:04:         has spi 1343294712 and conn_id 5128 and flags A
04:33:04:         lifetime of 2147483 seconds
04:33:04:         lifetime of 4608000 kilobytes
04:33:04:         has client flags 0x0
04:33:04: ISAKMP (0:13): deleting node 17011691 error FALSE reason "quick mode done
(await)"
04:33:04: ISAKMP (0:13): Node 17011691, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
04:33:04: ISAKMP (0:13): Old State = IKE_QM_R_QM2  New State = IKE_QM_PHASE2_COMPLETE
04:33:04: IPSEC(key_engine): got a queue event...
04:33:04: IPSEC(initialize_sas): ,
  (key eng. msg.) INBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-md5-hmac ,
    lifedur= 2147483s and 4608000kb,
    spi= 0xA3E24AFD(2749516541), conn_id= 5127, keysize= 0, flags= 0x2
04:33:04: IPSEC(initialize_sas): ,
  (key eng. msg.) OUTBOUND local= 172.18.1.1, remote= 10.1.1.1,
    local_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
    remote_proxy= 10.4.1.4/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp-3des esp-md5-hmac,
    lifedur= 2147483s and 4608000kb,
    spi= 0x50110CF8(1343294712), conn_id= 5128, keysize= 0, flags= 0xA
04:33:04: IPSEC(kei_proxy): head = ra, map->ivrf = vpn1, kei->ivrf = vpn2
04:33:04: IPSEC(kei_proxy): head = ra, map->ivrf = vpn2, kei->ivrf = vpn2
04:33:04: IPSEC(rte_mgr): VPN Route Added 10.4.1.4 255.255.255.255 via 10.1.1.1 in vpn2
04:33:04: IPSEC(add mtree): src 0.0.0.0, dest 10.4.1.4, dest_port 0

04:33:04: IPSEC(create_sa): sa created,
  (sa) sa_dest= 172.18.1.1, sa_prot= 50,
    sa_spi= 0xA3E24AFD(2749516541),
    sa_trans= esp-3des esp-md5-hmac, sa_conn_id= 5127
04:33:04: IPSEC(create_sa): sa created,
  (sa) sa_dest= 10.1.1.1, sa_prot= 50,
    sa_spi= 0x50110CF8(1343294712),
    sa_trans= esp-3des esp-md5-hmac, sa_conn_id= 5128
04:33:53: ISAKMP (0:13): purging node -1639992295
04:33:54: ISAKMP (0:13): purging node 17011691
```

# VRF-Aware IPsec の設定例

次の例では、VRF-Aware IPsec の設定方法を示しています。



## 例：スタティック IPsec-to-MPLS VPN

次のサンプルでは、IPsec トンネルを MPLS VPN にマッピングするスタティック設定を示しています。この設定により、IPsec トンネルが「VPN1」および「VPN2」にマッピングされます。両方の IPsec トンネルが、シングル パブリック方向インターフェイス上で終了します。

### IPsec PE の設定

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf vpn2
 rd 101:1
 route-target export 101:1
 route-target import 101:1
!
crypto keyring vpn1
  pre-shared-key address 172.16.1.1 key vpn1
crypto keyring vpn2
  pre-shared-key address 10.1.1.1 key vpn2
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp profile vpn1
 vrf vpn1
 keyring vpn1
 match identity address 172.16.1.1 255.255.255.255
!
crypto isakmp profile vpn2
 vrf vpn2
 keyring vpn2
 match identity address 10.1.1.1 255.255.255.255
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
crypto ipsec transform-set vpn2 esp-3des esp-md5-hmac
!
crypto map crypmap 1 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
crypto map crypmap 3 ipsec-isakmp
 set peer 10.1.1.1
 set transform-set vpn2
```

```
 set isakmp-profile vpn2
 match address 102
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
ip route 172.16.1.1 255.255.255.255 172.18.1.2
ip route 10.1.1.1 255.255.255.255 172.18.1.2
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
ip route vrf vpn2 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
access-list 102 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
```

### VPN1 用 IPsec Customer Provided Edge（CPE）設定

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
crypto isakmp key vpn1 address 172.18.1.1
!
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

### VPN2 用 IPsec CPE 設定

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp key vpn2 address 172.18.1.1
!
!
crypto ipsec transform-set vpn2 esp-3des esp-md5-hmac
!
crypto map vpn2 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn2
 match address 101
!
interface FastEthernet0
 ip address 10.1.1.1 255.255.255.0
```

```
 crypto map vpn2
!
interface FastEthernet1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
```

# 例：RSA 暗号化を使用した IPsec-to-MPLS VPN

次の例では、RSA 暗号化を使用した IPsec-to-MPLS VPN 設定を示します。

### PE ルータ設定

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
crypto isakmp policy 10
 authentication rsa-encr
!
crypto keyring vpn1
 rsa-pubkey address 172.16.1.1 encryption
  key-string
   305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00DBF381 00DDECC8
   DC4AA490 40320C52 9912D876 EB36717C 63DCA95C 7E5EC02A 84F276CE 292B42D7
   D664F324 3726F4E0 39D33093 ECB81B95 482511A5 F064C4B3 D5020301 0001
   quit
!
crypto isakmp profile vpn1
 vrf vpn1
 keyring vpn1
 match identity address 172.16.1.1 255.255.255.255
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map crypmap 1 ipsec-isakmp
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
ip route 172.16.1.1 255.255.255.255 172.18.1.2
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
```

### VPN1 用 IPsec CPE 設定

```
crypto isakmp policy 10
 authentication rsa-encr
!
crypto key pubkey-chain rsa
 addressed-key 172.18.1.1 encryption
```

```
 key-string
  3082011B 300D0609 2A864886 F70D0101 01050003 82010800 30820103 0281FB00
  C90CC78A 6002BDBA 24683396 B7D7877C 16D08C47 E00C3C10 63CF13BC 4E09EA23
  92EB8A48 4113F5A4 8796C8BE AD7E2DC1 3B0742B6 7118CE7C 1B0E21D1 AA9724A4
  4D74FCEA 562FF225 A2B11F18 E53C4415 61C3B741 3A06E75D B4F9102D 6163EE40
  16C68FD7 6532F660 97B59118 9C8DE3E5 4E2F2925 BBB87FCB 95223D4E A5E362DB
  215CB35C 260080805 17BBE1EF C3050E13 031F3D5B 5C22D16C FC8B1EC5 074F07A5
  D050EC80 7890D9C5 EC20D6F0 173FE2BA 89F5B5F9 2EADC9A6 D461921E 3D5B60016
  ABB8B6B9 E2124A21 93F0E4AE B487461B E7F1F1C4 032A0B0E 80DC3E15 CB268EC9
  5D76B9BD 3C78CB75 CE9F68C6 484D6573 CBC3EB59 4B5F3999 8F9D0203 010001
  quit
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

# 例：RSA シグニチャを使用した IPsec-to-MPLS VPN

次のに、RSA シグニチャを使用した IPsec-to-MPLS VPN 設定を示します。

### PE ルータ設定

```
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
crypto ca trustpoint bombo
 enrollment url http://172.31.68.59:80
 crl optional
!
crypto ca certificate chain bombo
 certificate 03C0
 308203BF 308202A7 A0030201 02020203 C0300D06 092A8648 86F70D01 01050500
 . . .
 quit
 certificate ca 01
 30820379 30820261 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
 . . .
 quit
!
crypto isakmp profile vpn1
 vrf vpn1
 ca trust-point bombo
 match identity address 172.16.1.1 255.255.255.255
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map crypmap 1 ipsec-isakmp
```

```
 set peer 172.16.1.1
 set transform-set vpn1
 set isakmp-profile vpn1
 match address 101
!
interface Ethernet1/1
 ip address 172.31.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map crypmap
!
ip route 172.16.1.1 255.255.255.255 172.18.1.2
ip route vrf vpn1 10.2.0.0 255.255.0.0 172.18.1.2 global
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
!
```

### VPN1 用 IPsec CPE 設定

```
crypto ca trustpoint bombo
 enrollment url http://172.31.68.59:80
 crl optional
 !
crypto ca certificate chain bombo
 certificate 03BF
  308203BD 308202A5 A0030201 02020203 BF300D06 092A8648 86F70D01 01050500
  . . .
  quit
 certificate ca 01
  30820379 30820261 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  . . .
  quit
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto map vpn1 1 ipsec-isakmp
 set peer 172.18.1.1
 set transform-set vpn1
 match address 101
!
interface FastEthernet1/0
 ip address 172.16.1.1 255.255.255.0
 crypto map vpn1
!
interface FastEthernet1/1
 ip address 10.2.1.1 255.255.0.0
!
access-list 101 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
```

# 例：IPsec Remote Access-to-MPLS VPN

次に、IPsec Remote Access-to-MPLS VPN 設定を示します。この設定により、IPsec トンネルが MPLS VPN にマッピングされます。IPsec トンネルが、シングル パブリック方向インターフェイス上で終了します。

### PE ルータ設定

```
aaa new-model
```

```
!
aaa group server radius vpn1
 server-private 10.1.1.1 auth-port 1645 acct-port 1646 timeout 5 retransmit 3 key vpn1
!
aaa group server radius vpn2
 server-private 10.1.1.1 auth-port 1645 acct-port 1646 timeout 5 retransmit 3 key vpn2
!
aaa authorization network aaa-list group radius
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
!
ip vrf vpn2
 rd 101:1
 route-target export 101:1
 route-target import 101:1
!
crypto isakmp profile vpn1-ra
   vrf vpn1
   match identity group vpn1-ra
   client authentication list vpn1
   isakmp authorization list aaa-list
   client configuration address initiate
   client configuration address respond
crypto isakmp profile vpn2-ra
   vrf vpn2
   match identity group vpn2-ra
   client authentication list vpn2
   isakmp authorization list aaa-list
   client configuration address initiate
   client configuration address respond
!
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
crypto ipsec transform-set vpn2 esp-3des esp-md5-hmac
!
crypto dynamic-map vpn1 1
 set transform-set vpn1
 set isakmp-profile vpn1-ra
 reverse-route
!
crypto dynamic-map vpn2 1
 set transform-set vpn2
 set isakmp-profile vpn2-ra
 reverse-route
!
!
crypto map ra 1 ipsec-isakmp dynamic vpn1
crypto map ra 2 ipsec-isakmp dynamic vpn2
!
interface Ethernet1/1
 ip address 172.17.1.1 255.255.0.0
 tag-switching ip
!
interface Ethernet1/2
 ip address 172.18.1.1 255.255.255.0
 crypto map ra
!
ip local pool vpn1-ra 10.4.1.1 10.4.1.254 group vpn1-ra
ip local pool vpn2-ra 10.4.1.1 10.4.1.254 group vpn2-ra
!
```

# Cisco Network-Based IPsec VPN Solution の旧バージョンからのアップデート

Cisco Network-Based IPsec VPN Solution リリース 1.5 における VRF-Aware IPsec 機能では、既存の設定を変更する必要があります。次のサンプル設定では、既存の設定に対して行う必要がある変更を示します。


- 「Site-to-Site 設定のアップグレード」（P.28）
- 「リモート アクセス設定のアップグレード」（P.29）
- 「Site-to-Site とリモート アクセスの組み合せのアップグレード」（P.31）


## Site-to-Site 設定のアップグレード

次の設定では、旧バージョンの Network-Based IPsec VPN Solution から Cisco Network-Based IPsec VPN Solution リリース 1.5 への Site-to-Site 設定のアップグレードに必要な変更を示します。

### 旧バージョンの Site-to-Site 設定

```
crypto isakmp key VPN1 address 172.21.25.74
 crypto isakmp key VPN2 address 172.21.21.74
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 match address 101
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 match address 102
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
  encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

### 新バージョンの Site-to-Site 設定

次に、同じ Site-to-Site 設定の、Cisco Network-Based IPsec VPN Solution リリース 1.5 ソリューションへアップグレードされたバージョンを示します。

（注） 2 つのキーリングを変更する必要があります。VRF-Aware Upset 機能では、IKE ローカル エンドポイントが VRF 内に存在している場合、キーを VRF に関連付ける必要があります。

```
crypto keyring VPN1-KEYS vrf VPN1
```

```
 pre-shared-key address 172.21.25.74 key VPN1
!
 crypto keyring VPN2-KEYS vrf VPN2
  pre-shared-key address 172.21.21.74 key VPN2
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
  set transform-set VPN1
 match address 101
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
  set transform-set VPN2
 match address 102
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

## リモート アクセス設定のアップグレード

次の設定では、旧バージョンの Network-Based IPsec VPN Solution から Cisco Network-Based IPsec VPN Solution リリース 1.5 へのリモート アクセス設定のアップグレードに必要な変更を示します。

### 旧バージョンのリモート アクセス設定

```
crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
  set transform-set VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
  reverse-route
!
!
 crypto map VPN1 client authentication list VPN1-RA-LIST
 crypto map VPN1 isakmp authorization list VPN1-RA-LIST
 crypto map VPN1 client configuration address initiate
 crypto map VPN1 client configuration address respond
```

```
 crypto map VPN1 10 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 client authentication list VPN2-RA-LIST
 crypto map VPN2 isakmp authorization list VPN2-RA-LIST
 crypto map VPN2 client configuration address initiate
 crypto map VPN2 client configuration address respond
 crypto map VPN2 10 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
  encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

## 新バージョンのリモート アクセス設定

次のインスタンスでは、アップグレードはありません。次の設定を変更することを推奨します。

```
crypto isakmp client configuration group VPN1-RA-GROUP
  key VPN1-RA
  pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto isakmp profile VPN1-RA
 match identity group VPN1-RA-GROUP
  client authentication list VPN1-RA-LIST
 isakmp authorization list VPN1-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto isakmp profile VPN2-RA
 match identity group VPN2-RA-GROUP
 client authentication list VPN2-RA-LIST
 isakmp authorization list VPN2-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 set isakmp-profile VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 set isakmp-profile VPN2-RA
 reverse-route
!
 crypto map VPN1 10 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 10 ipsec-isakmp dynamic VPN2-RA
!
```

```
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

## Site-to-Site とリモート アクセスの組み合せのアップグレード

次の設定では、旧バージョンの Network-Based IPsec VPN Solution から Cisco Network-Based IPsec VPN Solution リリース 1.5 への Site-to-Site およびリモート アクセス設定のアップグレードに必要な変更を示します。

### 旧バージョンの Site-to-Site およびリモート アクセスの設定

```
crypto isakmp key VPN1 address 172.21.25.74 no-xauth
crypto isakmp key VPN2 address 172.21.21.74 no-xauth
!
 crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 reverse-route
!
 crypto map VPN1 client authentication list VPN1-RA-LIST
 crypto map VPN1 isakmp authorization list VPN1-RA-LIST
 crypto map VPN1 client configuration address initiate
 crypto map VPN1 client configuration address respond
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 match address 101
 crypto map VPN1 20 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 client authentication list VPN2-RA-LIST
 crypto map VPN2 isakmp authorization list VPN2-RA-LIST
 crypto map VPN2 client configuration address initiate
 crypto map VPN2 client configuration address respond
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
```

```
 set transform-set VPN2
 match address 102
 crypto map VPN2 20 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

### 新バージョンの Site-to-Site およびリモート アクセスの設定

この設定をアップグレードする必要があります。

**（注）** Site-to-Site 設定に XAUTH が不要な場合、XAUTH 設定なしで ISAKMP プロファイルを設定します。リモート アクセス設定に XAUTH が必要な場合、XAUTH ありで ISAKMP プロファイルを設定します。

```
crypto keyring VPN1-KEYS vrf VPN1
 pre-shared-key address 172.21.25.74 key VPN1
!
 crypto keyring VPN2-KEYS vrf VPN2
 pre-shared-key address 172.21.21.74 key VPN2
!
 crypto isakmp client configuration group VPN1-RA-GROUP
 key VPN1-RA
 pool VPN1-RA
!
 crypto isakmp client configuration group VPN2-RA-GROUP
 key VPN2-RA
 pool VPN2-RA
!
 crypto isakmp profile VPN1
 keyring VPN1-KEYS
 match identity address 172.21.25.74 VPN1
!
 crypto isakmp profile VPN2
 keyring VPN2-KEYS
 match identity address 172.21.21.74 VPN2
!
 crypto isakmp profile VPN1-RA
 match identity group VPN1-RA-GROUP
 client authentication list VPN1-RA-LIST
 isakmp authorization list VPN1-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto isakmp profile VPN2-RA
 match identity group VPN2-RA-GROUP
 client authentication list VPN2-RA-LIST
 isakmp authorization list VPN2-RA-LIST
 client configuration address initiate
 client configuration address respond
!
 crypto ipsec transform-set VPN1 esp-des esp-sha-hmac
 crypto ipsec transform-set VPN2 esp-3des esp-sha-hmac
```

```
!
 crypto ipsec transform-set VPN1-RA esp-3des esp-sha-hmac
 crypto ipsec transform-set VPN2-RA esp-3des esp-md5-hmac
!
 crypto dynamic-map VPN1-RA 1
 set transform-set VPN1-RA
 set isakmp-profile VPN1-RA
 reverse-route
!
 crypto dynamic-map VPN2-RA 1
 set transform-set VPN2-RA
 set isakmp-profile VPN2-RA
 reverse-route
!
 crypto map VPN1 10 ipsec-isakmp
 set peer 172.21.25.74
 set transform-set VPN1
 set isakmp-profile VPN1
 match address 101
 crypto map VPN1 20 ipsec-isakmp dynamic VPN1-RA
!
 crypto map VPN2 10 ipsec-isakmp
 set peer 172.21.21.74
 set transform-set VPN2
 set isakmp-profile VPN2
 match address 102
 crypto map VPN2 20 ipsec-isakmp dynamic VPN2-RA
!
 interface FastEthernet0/0.1
 encapsulation dot1Q 1 native
 ip vrf forwarding VPN1
 ip address 172.21.25.73 255.255.255.0
 crypto map VPN1
!
 interface FastEthernet0/0.2
 encapsulation dot1Q 2 native
 ip vrf forwarding VPN2
 ip address 172.21.21.74 255.255.255.0
 crypto map VPN2
```

# その他の参考資料

## 関連資料

| 内容 | 参照先 |
|---|---|
| IPsec の設定作業 | 「Configuring Security for VPNs with IPsec」の章 |
| IPsec コマンド | 『*Cisco IOS Security Command Reference*』 |
| IKE フェーズ 1 とフェーズ 2、アグレッシブ モード、およびメイン モード | 「Configuring Internet Key Exchange for IPsec VPNs」 |
| IKE DPD | 「*Easy VPN Server*」 |

## 規格

| 規格 | タイトル |
|---|---|
| なし | — |

## MIB

| MIB | MIB リンク |
|---|---|
| なし | 選択したプラットフォーム、Cisco ソフトウェア リリース、および機能セットの MIB を検索してダウンロードする場合は、次の URL にある Cisco MIB Locator を使用します。<br>http://www.cisco.com/go/mibs |

## RFC

| RFC | タイトル |
|---|---|
| なし | — |

# シスコのテクニカル サポート

| 説明 | リンク |
|---|---|
| 右の URL にアクセスして、シスコのテクニカル サポートを最大限に活用してください。<br><br>以下を含むさまざまな作業にこの Web サイトが役立ちます。<br>・テクニカル サポートを受ける<br>・ソフトウェアをダウンロードする<br>・セキュリティの脆弱性を報告する、またはシスコ製品のセキュリティ問題に対する支援を受ける<br>・ツールおよびリソースへアクセスする<br>- Product Alert の受信登録<br>- Field Notice の受信登録<br>- Bug Toolkit を使用した既知の問題の検索<br>・Networking Professionals（NetPro）コミュニティで、技術関連のディスカッションに参加する<br>・トレーニング リソースへアクセスする<br>・TAC Case Collection ツールを使用して、ハードウェアや設定、パフォーマンスに関する一般的な問題をインタラクティブに特定および解決する<br><br>この Web サイト上のツールにアクセスする際は、Cisco.com のログイン ID およびパスワードが必要です。 | http://www.cisco.com/cisco/web/support/index.html |

# VRF-Aware IPsec の機能情報

表 1 に、この章に記載されている機能および具体的な設定情報へのリンクを示します。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェア イメージのサポート情報を検索できます。Cisco Feature Navigator を使用すると、ソフトウェア イメージがサポートする特定のソフトウェア リリース、機能セット、またはプラットフォームを確認できます。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

**（注）** 表 1 には、一連のソフトウェア リリースのうち、特定の機能が初めて導入されたソフトウェア リリースだけが記載されています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

**表 1　VRF-Aware IPsec の機能情報**

| 機能名 | リリース | 機能情報 |
|---|---|---|
| VRF-Aware IPSec | 12.2(15)T | VRF-Aware IPsec 機能には、Multiprotocol Label Switching（MPLS; マルチプロトコル ラベル スイッチング）Virtual Private Network（VPN; バーチャル プライベート ネットワーク）に対する IP Security（IPsec）トンネル マッピングが導入されています。VRF-Aware IPsec 機能を使用すれば、シングル パブリック方向アドレスによって、Virtual Routing and Forwarding（VPN Routing and Forwarding; VPN ルーティング/転送）に対して IPsec トンネルをマッピングできます。<br>この機能は、Cisco IOS Release 12.2(15)T で導入されました。<br>この機能に関する詳細については、次の各項を参照してください。<br>• 「VRF-Aware IPsec に関する情報」（P.2）<br>• 「VRF-Aware IPsec の設定方法」（P.4）<br>次のコマンドが導入または変更されました。**address**、**ca trust-point**、**client authentication list**、**client configuration address**、**crypto isakmp profile**、**crypto keyring**、**crypto map isakmp-profile**、**initiate-mode**、**isakmp authorization list**、**keepalive (isakmp profile)**、**keyring**、**key-string**、**match identity**、**no crypto xauth**、**pre-shared-key**、**quit**、**rsa-pubkey**、**self-identity**、**serial-number**、**set isakmp-profile**、**show crypto isakmp key**、**show crypto isakmp profile**、**vrf**、**clear crypto sa**、**crypto isakmp peer**、**crypto map isakmp-profile**、**show crypto dynamic-map**、**show crypto ipsec sa**、**show crypto isakmp sa**、**show crypto map (IPsec)** |
| | 15.1(1)S | この機能は、Cisco IOS Release 15.1(1)S に統合されました。 |

# 用語集

**CA**：Certification Authority（CA; 認証局）。CA はデジタル証明書を発行するエンティティ（特に X.509 証明書）で、証明書のデータ項目間のバインディングを保証します。

**CLI**：Command-Line Interface（CLI; コマンドラインインターフェイス）。CLI は、ユーザが、コマンドおよびオプションの引数を入力することによって、オペレーティング システムとやり取りをすることを可能にするインターフェイスです。UNIS オペレーティング システムと DOS では、CLI が使用できます。

**DN**：Distinguished Name（DN; 認定者名）。オープン システム インターコネクション（OSI ディレクトリ（X.500））内のエントリの、グローバルな権威ある名前です。

**FQDN**：Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）。FQDN は、単なるホスト名ではなく、システムにおける正式な名前です。たとえば、aldebaran はホスト名で、aldebaran.interop.com は FQDN です。

**FR**：Frame Relay（FR; フレームリレー）。FR は、接続されたデバイス間における High-Level Data Link（HDLC; ハイレベル データ リンク）カプセル化を使用して、複数の仮想回線を処理するための、業界標準の、スイッチデータリンク層プロトコルです。フレーム リレーは、一般的に置き代替可能と考えられているプロトコルである X.25 より効率的です。

**FVRF**：Front Door Virtual Routing and Forwarding（VRF）のリポジトリ。FVRF は、暗号化されたパケットをピアにルーティングするために使用される VRF です。

**IDB**：Interface Descriptor Block（IDB; インターフェイス記述子ブロック）。IDB サブブロックは、アプリケーションに対してプライベートとなっているメモリ領域です。この領域には、アプリケーションにとって IDB またはインターフェイスに関連付ける必要があるプライベート情報およびステートが格納されます。アプリケーションによって IDB が使用されてポインタがそのサブブロックに登録されますが、サブブロック自体の内容には登録されません。

**IKE**：Internet Key Exchange（IKE; インターネット キー エクスチェンジ）。IKE によって、キーが必要なサービス（IPsec など）のための共有セキュリティ ポリシーおよび認証キーが確立されます。IPsec トラフィックを通過させる前に、ルータ、ファイアウォール、ホストそれぞれでピアの ID を検証する必要があります。それには、事前共有キーを両ホストに手動で入力するか、CA サービスを使用します。

**IKE キープアライブ**：IKE ピアの活性を判断するための双方向メカニズム。

**IPsec**：IP 用セキュリティ プロトコル。

**IVRF**：Inside Virtual Routing and Forwarding。IVRF は、暗号化されていないテキスト パケットの VRF です。

**MPLS**：Multiprotocol Label Switching（MPLS; マルチプロトコル ラベル スイッチング）。MPLS は、ラベルを使用して IP トラフィックを転送するスイッチング方式です。このラベルによって、ネットワーク内のルータおよびスイッチが、事前に確立された IP ルーティング情報に基づくパケットの転送先を指示されます。

**RSA**：Rivest、Shamir、Adelman は、RSA 技術の発明者です。RSA 技術は、暗号化および認証に使用可能な公開キー暗号化システムです。

**SA**：Security Association（SA; セキュリティ アソシエーション）。SA は、データ フローに適用されるセキュリティ ポリシーおよびキー関連情報のインスタンスです。

**VPN**：Virtual Private Network（VPN; バーチャル プライベート ネットワーク）。VPN を使用すると、ネットワーク間のトラフィックをすべて暗号化することにより、パブリック TCP/IP ネットワーク経由でも IP トラフィックをセキュアに転送できます。VPN では、「トンネリング」が使用され、すべての情報が IP レベルで暗号化されます。

**VRF**：Virtual Route Forwarding。VRF は、VPN ルーティングおよび転送インスタンスです。VRF は、IP ルーティング テーブル、取得された転送テーブル、その転送テーブルを使用する一連のインターフェイス、転送テーブルに登録されるものを決定する一連のルールおよびルーティング プロトコルで構成されています。一般に、VRF には、PE ルータに付加されるカスタマー VPN サイトが定義されたルーティング情報が格納されています。

**XAUTH**：Extended Authentication（XAUTH; 拡張認証）。XAUTH は、IKE フェーズ 1 と IKE フェーズ 2 の間における任意の交換です。XAUTH では、ルータが、（ピアの認証ではなく）実際のユーザの認証試行において、追加の認証情報を要求します。

**クライアント**：Multi Protocol Label Switching（MPLS）ネットワーク内の UUT の対応する IPsec IOS ピア。

**デッド ピア**：到達できなくなった IKE ピア。

---