



VRF 認識 IPsec

VRF-Aware IPsec 機能には、マルチプロトコル ラベル スイッチング (MPLS) バーチャル プライベート ネットワーク (VPN) 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。



(注) セキュリティに対する脅威は、脅威からの保護に役立つ暗号化技術と同様に絶え間なく変化しています。シスコの暗号化に関する最新の推奨事項の詳細は、『[Next Generation Encryption](#)』(NGE) ホワイトペーパーを参照してください。

- [VRF-Aware IPsec に関する制約事項 \(1 ページ\)](#)
- [VRF-Aware IPsec に関する情報 \(2 ページ\)](#)
- [VRF-Aware IPsec の設定方法 \(4 ページ\)](#)
- [VRF-Aware IPsec の設定例 \(22 ページ\)](#)
- [その他の参考資料 \(34 ページ\)](#)
- [VRF-Aware IPsec の機能情報 \(35 ページ\)](#)
- [用語集 \(36 ページ\)](#)

VRF-Aware IPsec に関する制約事項

- クリプト 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、ユニキャスト RPF (uRPF) がクリプト 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、クリプト 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 機能をクリプト マップと使用した場合、このクリプト 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 またはダイナミック VTI (DVTI) 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。

- ISAKMP プロファイルとキーリング内で VRF と一緒にローカルアドレスを使用している場合は、**local-address** コマンドに VRF を含める必要があります。

VRF-Aware IPsec に関する情報

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、プロバイダーエッジ (PE) 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、派生シスコ エクスプレス フォワーディング (CEF) テーブル、転送テーブルを使用するインターフェイスのセット、ルーティングテーブルに含まれる情報を制御するルールおよびルーティングプロトコルパラメータのセット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

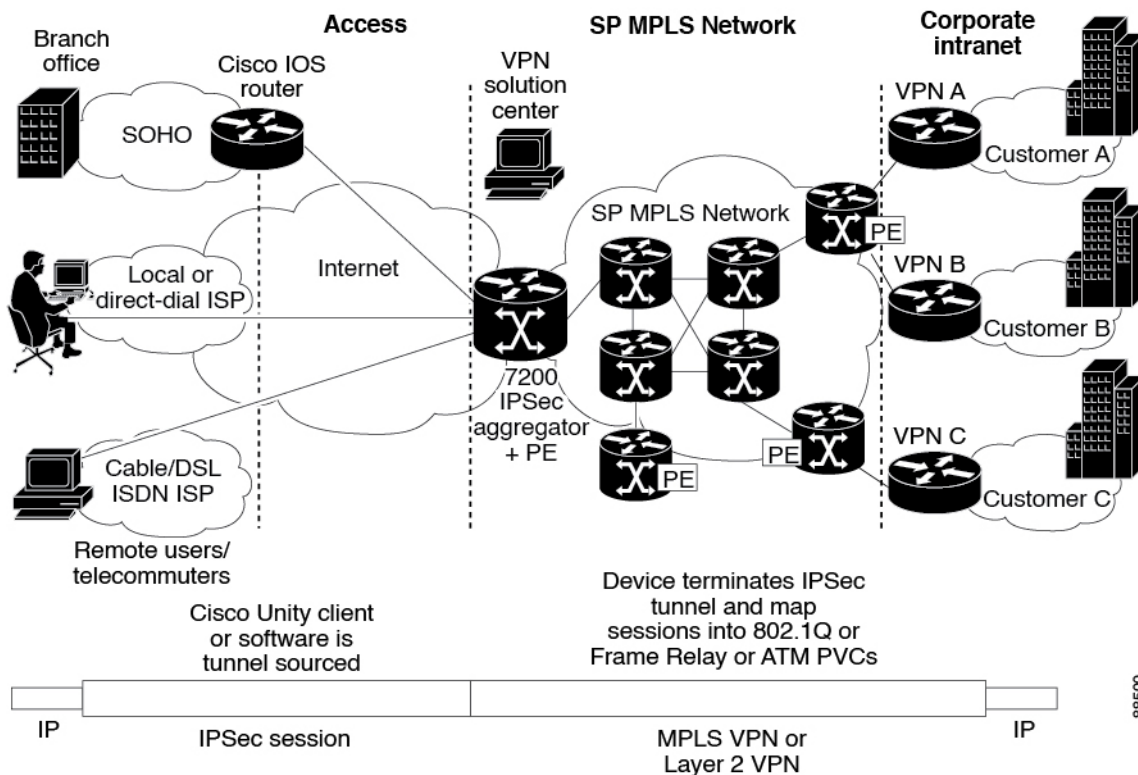
Front Door VRF (FVRF) と Inside VRF (IVRF) が、この機能を理解するうえで重要な概念となります。

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、クリプト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。

次の図は、MPLS およびレイヤ 2 VPN に対する IPsec のシナリオを示しています。

図 1: MPLS およびレイヤ 2 VPN に対する IPsec



88500

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 とアクセスコントロールリスト (ACL) 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、セキュリティパラメータインデックス (SPI)、宛先、およびプロトコルのセキュリティアソシエーション (SA) 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

暗号化キーリングの設定

暗号化キーリングは、事前共有キーおよび Rivest, Shamir, and 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 例： Router> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： 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 <i>keyring-name</i> [vrf <i>fvrf-name</i>] 例： Router (config)# crypto keyring VPN1	キーリングの名前として <i>keyring-name</i> 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。 • (任意) vrf キーワードおよび <i>fvrf-name</i> 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

	コマンドまたはアクション	目的
		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 <i>string</i> 例 : 例 : 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 <i>address</i> [<i>mask</i>] hostname <i>hostname</i> } key <i>key</i> 例 : 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 <i>address</i> name <i>fqdn</i> } [encryption signature] 例 : 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。 <ul style="list-style-type: none"> • 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。 • オプションの signature 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 <i>ip-address</i> 例 : 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 <i>serial-number</i> 例 : 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。
ステップ 9	key-string 例 : 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。

	コマンドまたはアクション	目的
ステップ 10	text 例： <pre>Router (config-pubkey)# 00302017 4A7D385B 1234EF29 335FC973</pre>	公開キーを指定します。 (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 例： <pre>Router (config-pubkey)# quit</pre>	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 例： <pre>Router (config-pubkey)# exit</pre>	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 例： <pre>Router (config-keyring)# exit#</pre>	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、一連のピアのインターネット キー交換 (IKE) 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、キーペアライブ、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) ルータから認証局 (CA) 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) 取得用)、または Lightweight Directory Access Protocol (LDAP) サーバーへのトラフィック (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。



(注) 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*] [*fvrif*] | **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 例： Router> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： 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 <i>profile-name</i> 例 : <pre>Router (config)# crypto isakmp profile vpnprofile</pre>	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 <i>string</i> 例 : <pre>Router (conf-isa-prof)# description configuration for VPN profile</pre>	(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 <i>ivrf-name</i> 例 : <pre>Router (conf-isa-prof)# vrf VPN1</pre>	(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。 (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トタにもなります。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 <i>seconds</i> retry <i>retry-seconds</i> 例 : <pre>Router (conf-isa-prof)# keepalive 60 retry 5</pre>	(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。 <ul style="list-style-type: none"> 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。 <i>seconds</i> : DPD メッセージ間の秒数。指定できる範囲は 10 ~ 3600 秒です。 retry <i>retry-seconds</i> : 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 { <i>address</i> <i>fqdn</i> user-fqdn <i>user-fqdn</i> } 例 : <pre>Router (conf-isa-prof)# self-identity address</pre>	(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。 <ul style="list-style-type: none"> 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。 address : 出力インターフェイスの IP アドレスを使用します。 fqdn-- : ルータの完全修飾ドメイン名 (FQDN) を使用します。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • user-fqdn : 指定された値を使用します。
ステップ 8	keyring <i>keyring-name</i> 例 : <pre>Router (conf-isa-prof)# keyring VPN1</pre>	(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。 <ul style="list-style-type: none"> • 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 { <i>trustpoint-name</i> } 例 : <pre>Router (conf-isa-prof)# ca trustpoint VPN1-trustpoint</pre>	(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。 <ul style="list-style-type: none"> • 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。
ステップ 10	match identity { group <i>group-name</i> address <i>address</i> [<i>mask</i>] [<i>fvr</i>] host <i>host-name</i> host domain <i>domain-name</i> user <i>user-fqdn</i> user domain <i>domain-name</i> } 例 : <pre>Router (conf-isa-prof)# match identity address 10.1.1.1</pre>	照合されるクライアント IKE の ID を指定します。 <ul style="list-style-type: none"> • group <i>group-name</i> : <i>group-name</i> と ID タイプ ID_KEY_ID を照合します。また、<i>group-name</i> と認定者名 (DN) の組織ユニット (OU) フィールドも照合します。 • address <i>address</i> [<i>mask</i>]<i>fvr</i> : <i>address</i> と ID タイプ ID_IPV4_ADDR を照合します。<i>Mask</i> 引数を使用して、アドレスの範囲を指定できます。<i>fvr</i> 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。 • host <i>hostname</i> : <i>hostname</i> と ID タイプ ID_FQDN を照合します。 • host domain <i>domain-name</i> : <i>domain-name</i> を、ドメイン名が <i>domain-name</i> 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。 • user <i>username</i> : <i>username</i> と ID タイプ ID_USER_FQDN を照合します。 • user domain <i>domainname</i> : ドメイン名が <i>domainname</i> と一致する ID タイプ ID_USER_FQDN を照合します。

	コマンドまたはアクション	目的
ステップ 11	client configuration address {initiate respond} 例： <pre>Router (conf-isa-prof)# client configuration address initiate</pre>	(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 例： <pre>Router (conf-isa-prof)# client authentication list xauthlist</pre>	(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 AAA (認証、許可、アカウントिंग)。
ステップ 13	isakmp authorization list list-name 例： <pre>Router (conf-isa-prof)# isakmp authorization list ikessaaalist</pre>	(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。
ステップ 14	initiate mode aggressive 例： <pre>Router (conf-isa-prof)# initiate mode aggressive</pre>	(任意) アグレッシブ モード交換を開始します。 <ul style="list-style-type: none"> 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 例： <pre>Router (conf-isa-prof)# exit</pre>	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 \(10 ページ\)](#) に進みます。

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、クリプト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。クリプトマップ上の ISAKMP プロファイルを設定するには、次の作業を実行します。

始める前に

クリプトマップ上で 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 例： Router> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： 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 map <i>map-name</i> isakmp-profile <i>isakmp-profile-name</i> 例： Router (config)# crypto map vpnmap isakmp-profile vpnprofile	(任意) クリプト 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、クリプトマップコンフィギュレーションモードを開始します。 • 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 <i>profile-name</i> 例： Router (config-crypto-map)# set isakmp-profile vpnprofile	(任意) トラフィックがクリプト 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 例： 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 xauth** 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 例： Router> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： 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 例： 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、設定情報およびセキュリティ アソシエーション (SA) 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 例： Router> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	show crypto ipsec sa [map <i>map-name</i>] address identity interface <i>interface</i> / peer [vrf <i>fvrfr-name</i>] address vrf <i>ivrfr-name</i>] [detail] 例 : 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 例 : 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。 <ul style="list-style-type: none"> 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 例 : 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 例 : 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。 <ul style="list-style-type: none"> 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** *fvrfr-name*] *address* | **spi** *address* {**ah** | **esp**} *spi* | **vrf** *ivrfr-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 例 : Router> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> パスワードを入力します（要求された場合）。
ステップ 2	clear crypto sa [counters map <i>map-name</i> peer [vrf <i>fvrfr-name</i>] <i>address</i> spi <i>address</i> { ah esp } <i>spi</i> vrf <i>ivrfr-name</i>]	IPsec SA をクリアします。

	コマンドまたはアクション	目的
	例： Router# clear crypto sa vrf VPN1	

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 例： Router> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	debug crypto ipsec 例： Router# debug crypto ipsec	IP セキュリティ（IPsec）イベントを表示します。
ステップ 3	debug crypto isakmp 例： 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 AES-CBC
04:32:55: ISAKMP: hash SHA
04:32:55: ISAKMP: default group 14
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_MSG_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_MSG_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_MSG_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_MSG_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_MSG_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_MSG_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_MSG_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_MSG_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_MSG_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_MSG_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_MSG_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 :.lI.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_AES
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-aes 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-aes 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_AES
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-aes 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: 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_MSG_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_MSG_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_MSG_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 (type=4),
      remote_proxy= 10.4.1.4/0.0.0.0/0 (type=1),

```

```

protocol= ESP, transform= esp-aes esp-sha-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-aes esp-sha-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-aes esp-sha-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-aes esp-sha-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

例：静的 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 MPLS VPN、「VPN1」および「VPN2」にマッピングされます。IPsec トンネルは両方とも、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 aes
 authentication pre-share
 group 14
!

```

```

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-aes esp-sha-hmac
crypto ipsec transform-set vpn2 esp-aes 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
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 aes
  authentication pre-share
  group 14
crypto isakmp key vpn1 address 172.18.1.1
!
!
crypto ipsec transform-set vpn1 esp-aes 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 aes
  authentication pre-share
  group 14
!
crypto isakmp key vpn2 address 172.18.1.1
!
!
crypto ipsec transform-set vpn2 esp-aes esp-sha-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-aes 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-aes 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

次のように、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-aes 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-aes 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-aes esp-sha-hmac
    crypto ipsec transform-set vpn2 esp-aes esp-sha-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

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-aes esp-sha-hmac
crypto ipsec transform-set VPN2 esp-aes 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-aes esp-sha-hmac
crypto ipsec transform-set VPN2 esp-aes 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-aes esp-sha-hmac
crypto ipsec transform-set VPN2-RA esp-aes esp-sha-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-aes esp-sha-hmac
crypto ipsec transform-set VPN2-RA esp-aes esp-sha-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 アクセスの設定の組み合わ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-aes esp-sha-hmac
crypto ipsec transform-set VPN2 esp-aes esp-sha-hmac
!
crypto ipsec transform-set VPN1-RA esp-aes esp-sha-hmac
crypto ipsec transform-set VPN2-RA esp-aes esp-sha-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-aes esp-sha-hmac
crypto ipsec transform-set VPN2 esp-aes esp-sha-hmac
!
crypto ipsec transform-set VPN1-RA esp-aes esp-sha-hmac
crypto ipsec transform-set VPN2-RA esp-aes esp-sha-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

関連項目	マニュアルタイトル
IPsec の設定作業	「Configuring Security for VPNs with IPsec」
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」
推奨される暗号化アルゴリズム	『Next Generation Encryption』

標準

標準	タイトル
なし	--

MIB

MIB	MIB のリンク
なし	<p>選択したプラットフォーム、Cisco ソフトウェアリリース、およびフィーチャセットの 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。</p> <p>http://www.cisco.com/go/mibs</p>

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。これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。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

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェアリリーストレインで各機能のサポートが導入されたときのソフトウェアリリースだけを示しています。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。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 1: VRF-Aware IPsec の機能情報

機能名	リリース	機能情報
VRF 認識 IPsec	12.2(15)T	<p>VRF-Aware IPsec 機能には、マルチプロトコル ラベル スイッチング (MPLS) バーチャルプライベート ネットワーク (VPN) 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、VPN ルーティング/転送 (VRF) に対して IPsec トンネルをマッピングできます。</p> <p>この機能は、Cisco IOS Release 12.2(15)T で導入されました。</p> <p>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。</p> <p>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)。</p>
	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 (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 (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。UNIX オペレーティングシステムと DOS では、CLI が使用できます。

client : マルチプロトコルラベルスイッチング (MPLS) ネットワーク内の UUT の対応する IPsec IOS ピア。

dead peer : 到達できなくなった IKE ピア。

DN : Distinguished Name (識別名)。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 (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 (フレームリレー)。FRは、接続されたデバイス間におけるハイレベルデータリンク (HDLC) 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 : 前面扉 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 (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 (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 keepalive : 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 (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 (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 (仮想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は「トンネリング」を使用して、IP レベルですべての情報を暗号化します。

VRF : Virtual Route Forwarding (仮想ルーティングおよびフォワーディング)。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、取得されたルーティングテーブル、そのルーティングテーブルを使用する一連のインターフェイス、ルーティング 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 (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。

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。