# Virtual Tunnel Interface

This chapter describes how to configure a VTI tunnel.



## About Virtual Tunnel Interfaces

ASA supports a logical interface called the Virtual Tunnel Interface (VTI). As an alternative to policy-based VPN, you can create a VPN tunnel between peers using VTIs. VTIs support route-based VPN with IPsec profiles attached to the end of each tunnel. You can use dynamic or static routes. Egressing traffic from the VTI is encrypted and sent to the peer, and the associated SA decrypts the ingress traffic to the VTI.

Using VTI does away with the requirement of configuring static crypto map access lists and mapping them to interfaces. You no longer have to track all remote subnets and include them in the crypto map access list. Deployments become easier, and having static VTI which supports route-based VPN with dynamic routing protocol also satisfies many requirements of a virtual private cloud.

### Static VTI

You can use static VTI configurations for site-to-site connectivity in which a tunnel is always-on between two sites. For a static VTI interface, you must define a physical interface as a tunnel source. You can associate a maximum of 1024 VTIs per device. To create a static VTI interface, see Add a VTI Interface, on page 6.

## Guidelines for Virtual Tunnel Interfaces

### Context Mode and Clustering

- Supported in single mode only.
- No support for clustering.

### Firewall Mode

Supported in routed mode only.

### IPv6 Support

IPv6 is not supported.

### General Configuration Guidelines

- VTIs are only configurable in IPsec mode. To terminate GRE tunnels on an ASA is unsupported.
- You can use BGP or static routes for traffic using the tunnel interface.
- The MTU for VTIs is automatically set, according to the underlying physical interface. However, if you change the physical interface MTU after the VTI is enabled, you must disable and reenable the VTI to use the new MTU setting.
- VTI supports IKE versions v1, v2, and uses IPsec for sending and receiving data between the tunnel's source and destination.
- If NAT has to be applied, the IKE and ESP packets are encapsulated in the UDP header.
- IKE and IPsec security associations will be re-keyed continuously regardless of data traffic in the tunnel. This ensures that VTI tunnels are always up.
- The tunnel group name must match what the peer sends as its IKEv1 or IKEv2 identity.
- For IKEv1 in site-to-site tunnel groups, you can use names which are not IP addresses, if the tunnel authentication method is digital certificates and/or the peer is configured to use aggressive mode.
- VTI and crypto map configurations can co-exist on the same physical interface, provided the peer address configured in the crypto map and the tunnel destination for the VTI are different.
- Access rules can be applied on a VTI interface to control traffic through VTI.
- ICMP ping is supported between VTI interfaces.
- If the peer device for an IKEv2 site-to-site VPN tunnel sends IKEv2 configuration request payloads, the ASA cannot establish an IKEv2 tunnel with the device. You must disable the config-exchange request on the peer device for the ASA to establish a VPN tunnel with the peer device.

### Default Settings

- By default, all traffic through VTI is encrypted.
- By default, the security level for VTI interfaces is 0. You cannot configure the security level.

### Limitations for VTI

ASA drops Security Group Tag (SGT) frames and packets after VTI decryption.

# Create a VTI Tunnel

To configure a VTI tunnel, create an IPsec proposal (transform set). You will need to create an IPsec profile that references the IPsec proposal, followed by a VTI interface with the IPsec profile. Configure the remote peer with identical IPsec proposal and IPsec profile parameters. SA negotiation will start when all tunnel parameters are configured.

**Note** For the ASA which is a part of both the VPN VTI domains, and has BGP adjacency on the physical interface:

When a state change is triggered due to the interface health check, the routes in the physical interface will be deleted until BGP adjacency is re-established with the new active peer. This behavior does not apply to logical VTI interfaces.

Access control lists can be applied on a VTI interface to control traffic through VTI. To permit any packets that come from an IPsec tunnel without checking ACLs for the source and destination interfaces, enter the sysopt connection permit-vpn command in global configuration mode.

You can use the following command to enable IPsec traffic through the ASA without checking ACLs:

**hostname(config)# sysopt connection permit-vpn**

When an outside interface and VTI interface have the security level of 0, if you have ACL applied on VTI interface, it will not be hit if you do not have same-security-traffic configured.

To configure this feature, use the **same-security-traffic** command in global configuration mode with its **intra-interface** argument.

For more information, see Permitting Intra-Interface Traffic (Hairpinning).

### Procedure

**Step 1** Add an IPsec Proposal (Transform Sets).

**Step 2** Add an IPsec Profile.

**Step 3** Add a VTI Tunnel.

# Add an IPsec Proposal (Transform Sets)

A transform set is required to secure traffic in a VTI tunnel. Used as a part of the IPsec profile, it is a set of security protocols and algorithms that protects the traffic in the VPN.

### Before you begin

- You can use either pre-shared key or certificates for authenticating the IKE session associated with a VTI. IKEv2 allows asymmetric authentication methods and keys. For both IKEv1 and IKEv2, you must configure the pre-shared key under the tunnel group used for the VTI.
- For certificate based authentication using IKEv1, you must specify the trustpoint to be used at the initiator. For the responder, you must configure the trustpoint in the tunnel-group command. For IKEv2, you must configure the trustpoint to be used for authentication under the tunnel group command for both initiator and responder.

## Procedure

Add an IKEv1 transform set, or an IKEv2 IPsec proposal to establish the security association.

Add an IKEv1 transform set:

**crypto ipsec ikev1 transform-set** {*transform-set-name* | *encryption* | *authentication*}

**Example:**

```
ciscoasa(config)#crypto ipsec ikev1 transform-set SET1 esp-aes esp-sha-hmac
```

*Encryption* specifies which encryption method protects IPsec data flows:

- esp-aes—Uses AES with a 128-bit key.
- esp-aes-192—Uses AES with a 192-bit key.
- esp-aes-256—Uses AES with a 256-bit key.
- esp-null—No encryption.

*Authentication* specifies which encryption method to protect IPsec data flows:

- esp-md5-hmac—Uses the MD5/HMAC-128 as the hash algorithm.
- esp-sha-hmac—Uses the SHA/HMAC-160 as the hash algorithm.
- esp-none—No HMAC authentication.

Add an IKEv2 IPsec proposal.

**Note**
For the IOS platform, use the **no config-exchange request** command in the IKEv2 profile configuration mode to disable configuration exchange options. See http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/security/a1/sec-a1-cr-book/sec-cr-c2.html#wp3456426280 for more information.

- Specify a name for the IPsec proposal:

  **crypto ipsec ikev2 ipsec-proposal** *IPsec proposal name*

**Example:**

```
ciscoasa(config)#crypto ipsec ikev2 ipsec-proposal SET1
```

- Specify the security parameters in the crypto IPsec ikev2 ipsec-proposal configuration mode:

  **protocol esp {encryption {aes | aes-192 | aes-256 | aes-gcm | aes-gcm-192 | aes-gcm-256 | null} | integrity {sha-1 | sha-256 | sha-384 | sha-512 | null}**

**Example:**

```
ciscoasa(config-ipsec-proposal)#protocol esp encryption aes aes-192
```

# Add an IPsec Profile

An IPsec profile contains the required security protocols and algorithms in the IPsec proposal or transform set that it references. This ensures a secure, logical communication path between two site-to-site VTI VPN peers.

## Procedure

**Step 1** Set a name for the profile:

**crypto ipsec profile** *name*

**Example:**

```
ciscoasa(config)#crypto ipsec profile PROFILE1
```

**Step 2** Set the IKEv1 or IKEv2 proposal. You can choose either an IKEv1 transform set or an IKEv2 IPsec proposal.

a) Set the IKEv1 transform set.

- To set the IKEv1 proposal, enter the following command in the crypto ipsec profile command sub-mode:

  **set ikev1 transform set** *set_name*

  In this example, SET1 is the IKEv1 proposal set created previously.

  ```
  ciscoasa(config-ipsec-profile)#set ikev1 transform-set SET1
  ```

b) Set the IKEv2 proposal.

- To set the IKEv2 proposal, enter the following command in the crypto ipsec profile command sub-mode:

  **set ikev2 ipsec-proposal** *IPsec_proposal_name*

  In this example, SET1 is the IKEv2 IPsec proposal created previously.

  ```
  ciscoasa(config-ipsec-profile)#set ikev2 ipsec-proposal SET1
  ```

**Step 3** (Optional) Specify the duration of the security association:

**set security-association lifetime** {**seconds** *number* | **kilobytes** {*number* | **unlimited**}}

**Example:**

```
ciscoasa(config-ipsec-profile)#set security-association lifetime
seconds 120 kilobytes 10000
```

**Step 4** (Optional) Configure the end of the VTI tunnel to act only as a responder:

**responder-only**

- You can configure one end of the VTI tunnel to perform only as a responder. The responder-only end will not initiate the tunnel or rekeying.
- If you are using IKEv2, set the duration of the security association lifetime, greater than the lifetime value in the IPsec profile in the initiator end. This is to facilitate successful rekeying by the initiator end and ensure that the tunnels remain up.

- If you are using IKEv1, IOS should always be in responder-only mode since IOS doesn't support continuous channel mode. The ASA becomes the initiator and session and rekeys.
- If the rekey configuration in the initiator end is unknown, remove the responder-only mode to make the SA establishment bi-directional, or configure an infinite IPsec lifetime value in the responder-only end to prevent expiry.

**Step 5** (Optional) Specify the PFS group. Perfect Forward Secrecy (PFS) generates a unique session key for each encrypted exchange. This unique session key protects the exchange from subsequent decryption. To configure PFS, you have to select the Diffie-Hellman key derivation algorithm to use when generating the PFS session key. The key derivation algorithms generate IPsec security association (SA) keys. Each group has a different size modulus. A larger modulus provides higher security, but requires more processing time. You must have matching Diffie-Hellman groups on both peers.

**set pfs** { **group14** }

**Example:**

```
ciscoasa(config-ipsec-profile)# set pfs group14
```

**Step 6** (Optional) Specify a trustpoint that defines the certificate to be used while initiating a VTI tunnel connection.

**set trustpoint** *name*

**Example:**

```
ciscoasa(config-ipsec-profile)#set trustpoint TPVTI
```

# Add a VTI Interface

To create a new VTI interface and establish a VTI tunnel, perform the following steps:

**Note** Implement IP SLA to ensure that the tunnel remains up when a router in the active tunnel is unavailable. See Configure Static Route Tracking in the ASA General Operations Configuration Guide in http://www.cisco.com/go/asa-config.

## Procedure

**Step 1** Create a new tunnel interface:

**interface tunnel** *tunnel_interface_number*

Specify a tunnel ID, from a range of 0 to 10413. Up to 10413 VTI interfaces are supported.

**Note**
If you will be migrating configurations from other devices to ASA 5506 devices, use the tunnel ID range of 1 - 10413. This is to ensure compatibility of the tunnel range of 1 - 10413 available in ASA 5506 devices.

**Example:**

```
ciscoasa(config)#interface tunnel 100
```

**Step 2** Enter the name of the VTI interface.

Enter the following command in the **interface tunnel** command submode:

**nameif** *interface name*

**Example:**

```
ciscoasa(config-if)#nameif vti
```

**Step 3** Enter the IP address of the VTI interface.

**ip address** *IP addressmask*

**Example:**

```
ciscoasa(config-if)#ip address 192.168.1.10 255.255.255.254
```

**Step 4** Specify the tunnel source interface.

**tunnel source interface** *interface_name*

The source interface can be a physical interface.

**Example:**

```
ciscoasa(config-if)#tunnel source interface outside
```

**Step 5** Specify the tunnel destination IP address.

**tunnel destination** *ip_address*

**Example:**

```
ciscoasa(config-if)#tunnel destination 10.1.1.1
```

**Step 6** Configure the tunnel with tunnel mode IPsec IPv4.

**tunnel mode ipsec** *ipv4*

**Example:**

```
ciscoasa(config-if)#tunnel mode ipsec ipv4
```

**Step 7** Assign the IPsec profile to tunnel.

**tunnel protection ipsec** *IPsec profile*

**Example:**

```
ciscoasa(config-if)#tunnel protection ipsec Profile1
```

---

### Example

Example configuration of a VTI tunnel (with IKEv2) between ASA and an IOS device:

```
ASA:

crypto ikev2 policy 1
 encryption aes-gcm-256
 integrity null
 group 21
```

```
 prf sha512
 lifetime seconds 86400
!
crypto ipsec ikev2 ipsec-proposal gcm256
 protocol esp encryption aes-gcm-256
 protocol esp integrity null
!
crypto ipsec profile asa-vti
 set ikev2 ipsec-proposal gcm256
!
interface Tunnel 100
 nameif vti
 ip address 10.10.10.1 255.255.255.254
 tunnel source interface [asa-source-nameif]
 tunnel destination [router-ip-address]
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile asa-vti
!
tunnel-group [router-ip-address] ipsec-attributes
 ikev2 remote-authentication pre-shared-key cisco
 ikev2 local-authentication pre-shared-key cisco
!
crypto ikev2 enable [asa-interface-name]


IOS:

!
crypto ikev2 proposal asa-vti
 encryption aes-gcm-256
 prf sha512
 group 21
!
crypto ikev2 policy asa-vti
 match address local [router-ip-address]
 proposal asa-vti
!
crypto ikev2 profile asa-vti
 match identity remote address [asa-ip-address] 255.255.255.255
 authentication local pre-share key cisco
 authentication remote pre-share key cisco
 no config-exchange request
!
crypto ipsec transform-set gcm256 esp-gcm 256
!
crypto ipsec profile asa-vti
 set ikev2-profile asa-vti
 set transform-set gcm256
!
interface tunnel 100
 ip address 10.10.10.0 255.255.255.254
 tunnel mode ipsec ipv4
 tunnel source [router-interface]
 tunnel destination [asa-ip-address]
 tunnel protection ipsec profile asa-vti
!
```

# Feature History for Virtual Tunnel Interface

| Feature Name | Releases | Feature Information |
|---|---|---|
| DHCP Relay Server Support on VTI | 9.14(1) | ASA allows VTI interfaces to be configured as DHCP relay server connecting interfaces.<br>We modified the following commands: **dhcprelay server** *ip_address vti_ifc_name*. |
| Support for IKEv2, certificate based authentication, and ACL in VTI | 9.8.(1) | Virtual Tunnel Interface (VTI) now supports BGP (static VTI). You can now use IKEv2 in standalone and high availability modes. You can use certificate based authentication by setting up a trustpoint in the IPsec profile. You can also apply access lists on VTI using access-group commands to filter ingress traffic.<br>We introduced the following command in the IPsec profile configuration mode: **set trustpoint**. |
| Virtual Tunnel Interface (VTI) support | 9.7.(1) | The ASA is enhanced with a new logical interface called Virtual Tunnel Interface (VTI), used to represent a VPN tunnel to a peer. This supports route based VPN with IPsec profiles attached to each end of the tunnel. Using VTI does away with the need to configure static crypto map access lists and map them to interfaces.<br>We introduced the following commands: **crypto ipsec profile, interface tunnel, responder-only, set ikev1 transform-set, set pfs, set security-association lifetime, tunnel destination, tunnel mode ipsec, tunnel protection ipsec profile, tunnel source interface**. |