# IPsec Virtual Tunnel Interface

**First Published: October 18, 2004**
**Last Updated: January 5, 2011**

IP security (IPsec) virtual tunnel interfaces (VTIs) provide a routable interface type for terminating IPsec tunnels and an easy way to define protection between sites to form an overlay network. IPsec VTIs simplify configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing.

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for IPsec Virtual Tunnel Interface" section on page 29.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

## Contents



**Americas Headquarters:**
**Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA**

# Restrictions for IPsec Virtual Tunnel Interface

#### IPsec Transform Set

The IPsec transform set must be configured in tunnel mode only.

#### IKE Security Association

The Internet Key Exchange (IKE) security association (SA) is bound to the VTI. Because IKE SA is bound to the VTI, the same IKE SA cannot be used for a crypto map.

#### IPsec SA Traffic Selectors

Static VTIs support only a single IPsec SA that is attached to the VTI interface. The traffic selector for the IPsec SA is always "IP any any."

A dynamic VTI also is a point-point interface that supports only a single IPsec SA, but the dynamic VTI is flexible in that it can accept the IPsec selectors that are proposed by the initiator.

#### Proxy

Static VTIs support only the "IP any any" proxy.

Dynamic VTIs support only one proxy, which can be "IP any any" or any subset of it.

#### QoS Traffic Shaping

The shaped traffic is process switched.

#### Stateful Failover

IPsec stateful failover is not supported with IPsec VTIs.

#### Tunnel Protection

The **shared** keyword is not required and must not be configured when using the **tunnel mode ipsec ipv4** command for IPsec IPv4 mode.

#### Static VTIs Versus GRE Tunnels

The IPsec VTI is limited to IP unicast and multicast traffic only, as opposed to GRE tunnels, which have a wider application for IPsec implementation.

#### VRF-Aware IPsec Configuration

In VRF-aware IPsec configurations with either static or dynamic VTIs (DVTIs), the VRF must *not* be configured in the Internet Security Association and Key Management Protocol (ISAKMP) profile. Instead, the VRF must be configured on the tunnel interface for static VTIs. For DVTIs, you must apply VRF to the virtual template using the **ip vrf forwarding** command.

# Information About IPsec Virtual Tunnel Interface

The use of IPsec VTIs both greatly simplifies the configuration process when you need to provide protection for remote access and provides a simpler alternative to using generic routing encapsulation (GRE) or Layer 2 Tunneling Protocol (L2TP) tunnels for encapsulation and crypto maps with IPsec. A major benefit associated with IPsec VTIs is that the configuration does not require a static mapping of

IPsec sessions to a physical interface. The IPsec tunnel endpoint is associated with an actual (virtual) interface. Because there is a routable interface at the tunnel endpoint, many common interface capabilities can be applied to the IPsec tunnel.

The IPsec VTI allows for the flexibility of sending and receiving both IP unicast and multicast encrypted traffic on any physical interface, such as in the case of multiple paths. Traffic is encrypted or decrypted when it is forwarded from or to the tunnel interface and is managed by the IP routing table. Using IP routing to forward the traffic to the tunnel interface simplifies the IPsec VPN configuration compared to the more complex process of using access control lists (ACLs) with the crypto map in native IPsec configurations. DVTIs function like any other real interface so that you can apply quality of service (QoS), firewall, and other security services as soon as the tunnel is active.

Without Virtual Private Network (VPN) Acceleration Module2+ (VAM2+) accelerating virtual interfaces, the packet traversing an IPsec virtual interface is directed to the router processor (RP) for encapsulation. This method tends to be slow and has limited scalability. In hardware crypto mode, all the IPsec VTIs are accelerated by the VAM2+ crypto engine, and all traffic going through the tunnel is encrypted and decrypted by the VAM2+.

The following sections provide details about the IPsec VTI:



# Benefits of Using IPsec Virtual Tunnel Interfaces

IPsec VTIs allow you to configure a virtual interface to which you can apply features. Features for clear-text packets are configured on the VTI. Features for encrypted packets are applied on the physical outside interface. When IPsec VTIs are used, you can separate the application of features such as NAT, ACLs, and QoS and apply them to clear-text or encrypted text, or both. When crypto maps are used, there is no simple way to apply encryption features to the IPsec tunnel.

There are two types of VTI interfaces: static VTIs (SVTIs) and dynamic VTIs (DVTIs).

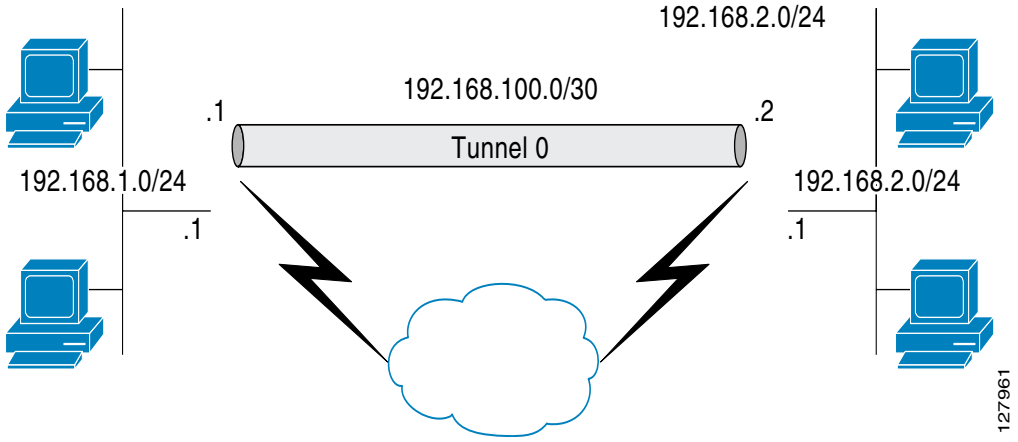
# Static Virtual Tunnel Interfaces

SVTI configurations can be used for site-to-site connectivity in which a tunnel provides always-on access between two sites. The advantage of using SVTIs as opposed to crypto map configurations is that users can enable dynamic routing protocols on the tunnel interface without the extra 4 bytes required for GRE headers, thus reducing the bandwidth for sending encrypted data.

Additionally, multiple Cisco IOS software features can be configured directly on the tunnel interface and on the physical egress interface of the tunnel interface. This direct configuration allows users to have solid control on the application of the features in the pre- or post-encryption path.

Figure 1 illustrates how a static VTI is used.

*Figure 1 IPsec Static VTI*




The IPsec VTI supports native IPsec tunneling and exhibits most of the properties of a physical interface.

# Dynamic Virtual Tunnel Interfaces

DVTIs can provide highly secure and scalable connectivity for remote-access VPNs. The DVTI technology replaces dynamic crypto maps and the dynamic hub-and-spoke method for establishing tunnels.

Dynamic VTIs can be used for both the server and remote configuration. The tunnels provide an on-demand separate virtual access interface for each VPN session. The configuration of the virtual access interfaces is cloned from a virtual template configuration, which includes the IPsec configuration and any Cisco IOS software feature configured on the virtual template interface, such as QoS, NetFlow, or ACLs.

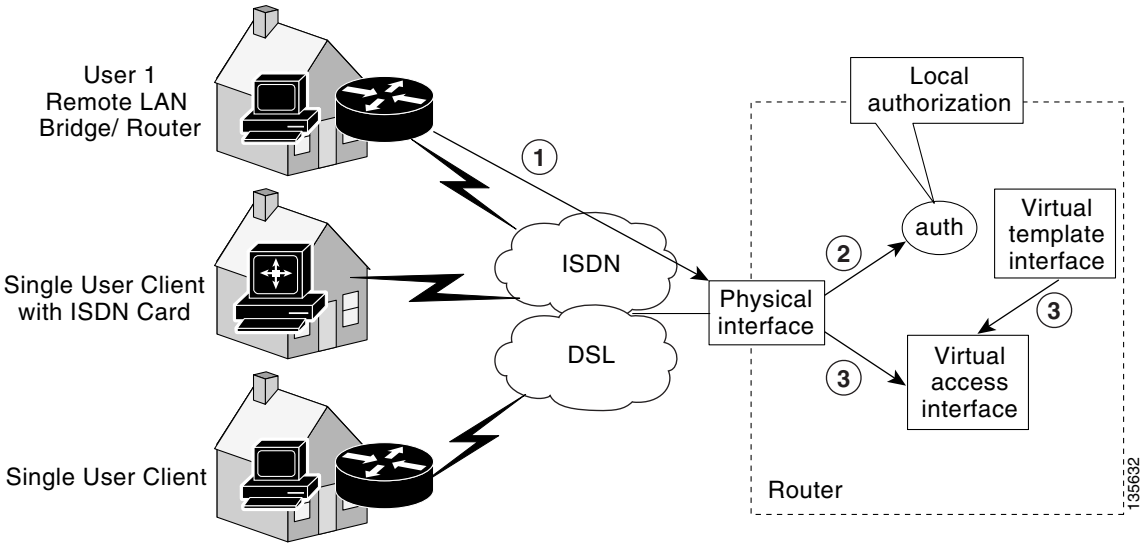
Dynamic VTIs function like any other real interface so that you can apply QoS, firewall, other security services as soon as the tunnel is active. QoS features can be used to improve the performance of various applications across the network. Any combination of QoS features offered in Cisco IOS software can be used to support voice, video, or data applications.

Dynamic VTIs provide efficiency in the use of IP addresses and provide secure connectivity. Dynamic VTIs allow dynamically downloadable per-group and per-user policies to be configured on a RADIUS server. The per-group or per-user definition can be created using extended authentication (Xauth) User or Unity group, or it can be derived from a certificate. Dynamic VTIs are standards based, so interoperability in a multiple-vendor environment is supported. IPsec DVTIs allow you to create highly secure connectivity for remote access VPNs and can be combined with Cisco Architecture for Voice, Video, and Integrated Data (AVVID) to deliver converged voice, video, and data over IP networks. The DVTI simplifies Virtual Private Network (VRF) routing and forwarding- (VRF-) aware IPsec deployment. The VRF is configured on the interface.

A DVTI requires minimal configuration on the router. A single virtual template can be configured and cloned.

The DVTI creates an interface for IPsec sessions and uses the virtual template infrastructure for dynamic instantiation and management of dynamic IPsec VTIs. The virtual template infrastructure is extended to create dynamic virtual-access tunnel interfaces. Dynamic VTIs are used in hub-and-spoke configurations. A single DVTI can support several static VTIs. Figure 2 illustrates the DVTI authentication path.

*Figure 2 Dynamic IPsec VTI*




The authentication shown in Figure 2 follows this path:

1. User 1 calls the router.
2. Router 1 authenticates User 1.
3. IPsec clones virtual access interface from virtual template interface.

# Dynamic Virtual Tunnel Interface Life Cycle

IPsec profiles define policy for dynamic VTIs. The dynamic interface is created at the end of IKE Phase 1 and IKE Phase 1.5. The interface is deleted when the IPsec session to the peer is closed. The IPsec session is closed when both IKE and IPsec SAs to the peer are deleted.

# Routing with IPsec Virtual Tunnel Interfaces

Because VTIs are routable interfaces, routing plays an important role in the encryption process. Traffic is encrypted only if it is forwarded out of the VTI, and traffic arriving on the VTI is decrypted and routed accordingly. VTIs allow you to establish an encryption tunnel using a real interface as the tunnel endpoint. You can route to the interface or apply services such as QoS, firewalls, network address translation, and Netflow statistics as you would to any other interface. You can monitor the interface, route to it, and it has an advantage over crypto maps because it is a real interface and provides the benefits of any other regular Cisco IOS interface.

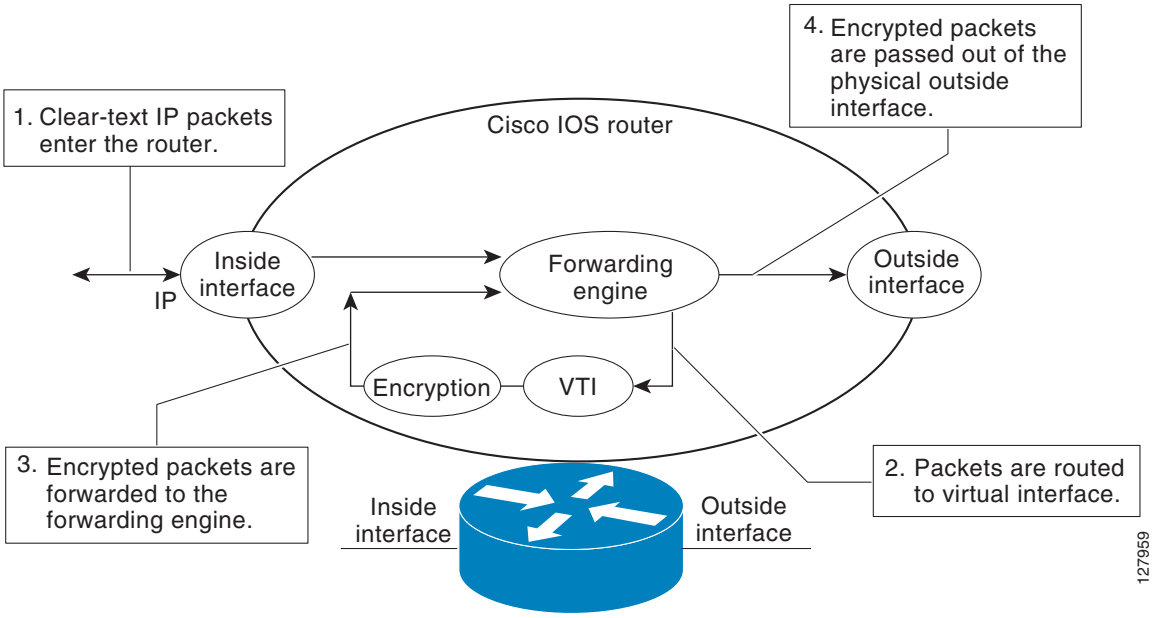
# Traffic Encryption with the IPsec Virtual Tunnel Interface

When an IPsec VTI is configured, encryption occurs in the tunnel. Traffic is encrypted when it is forwarded to the tunnel interface. Traffic forwarding is handled by the IP routing table, and dynamic or static routing can be used to route traffic to the SVTI. DVTI uses reverse route injection to further

simplify the routing configurations. Using IP routing to forward the traffic to encryption simplifies the IPsec VPN configuration because the use of ACLs with a crypto map in native IPsec configurations is not required. The IPsec virtual tunnel also allows you to encrypt multicast traffic with IPsec.

IPsec packet flow into the IPsec tunnel is illustrated in Figure 3.

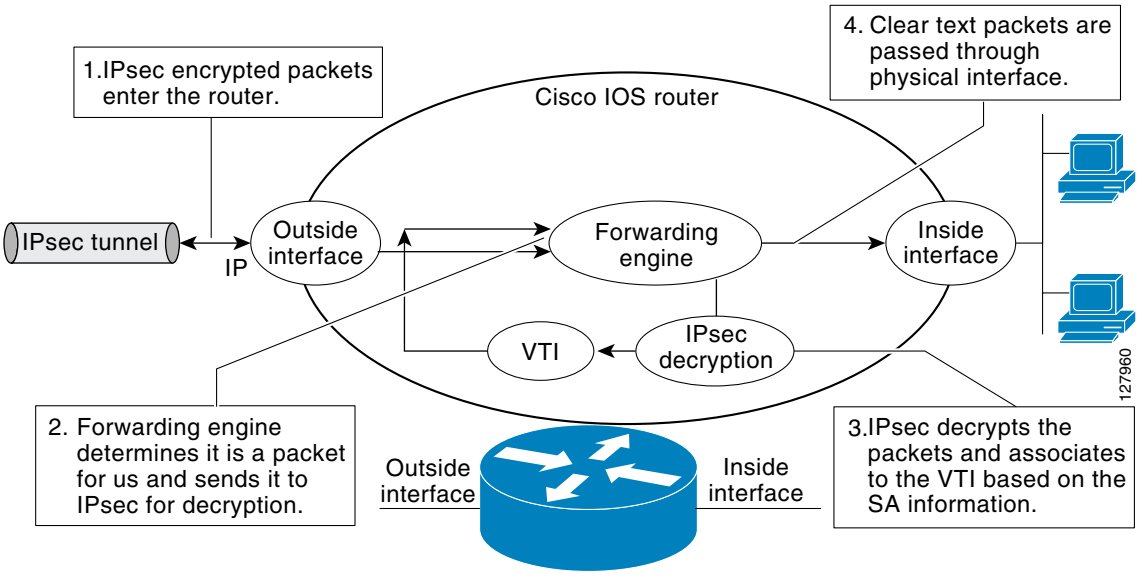
***Figure 3 Packet Flow into the IPsec Tunnel***




After packets arrive on the inside interface, the forwarding engine switches the packets to the VTI, where they are encrypted. The encrypted packets are handed back to the forwarding engine, where they are switched through the outside interface.

Figure 4 shows the packet flow out of the IPsec tunnel.

***Figure 4 Packet Flow out of the IPsec Tunnel***




# Per-User Attribute Support for Easy VPN Servers

The Per-User Attribute Support for Easy VPN Servers feature provides users with the ability to support per-user attributes on Easy VPN servers. These attributes are applied on the virtual access interface.

## Local Easy VPN AAA Server

For a local Easy VPN AAA server, the per-user attributes can be applied at the group level or at the user level using the command-line interface (CLI).

To configure per-user attributes for a local Easy VPN server, see "Configuring Per-User Attributes on a Local Easy VPN AAA Server."

## Remote Easy VPN AAA Server

Attribute value (AV) pairs can be defined on a remote Easy VPN AAA server as shown in this example:

```
cisco-avpair = "ip:outacl#101=permit tcp any any established
```

## Per-User Attributes

The following per-user attributes are currently defined in the AAA server and are applicable to IPsec:

- inacl
- interface-config
- outacl
- route

- rte-fltr-in
- rte-fltr-out
- sub-policy-In
- sub-policy-Out
- policy-route
- prefix

# How to Configure IPsec Virtual Tunnel Interface



## Configuring Static IPsec Virtual Tunnel Interfaces

This configuration shows how to configure a static IPsec VTI.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto IPsec profile** *profile-name*
4. **set transform-set** *transform-set-name*
5. **interface** *type number*
6. **ip address** *address mask*
7. **tunnel mode ipsec ipv4**
8. **tunnel source** *interface*
9. **tunnel destination** *ip-address*
10. **tunnel protection IPsec profile** *profile-name* [**shared**]

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **crypto IPsec profile** *profile-name*<br><br>**Example:**<br>`Router(config)# crypto IPsec profile PROF` | Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers. |
| **Step 4** | **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]<br><br>**Example:**<br>`Router(config)# set transform-set tset` | Specifies which transform sets can be used with the crypto map entry. |
| **Step 5** | **interface** *type number*<br><br>**Example:**<br>`Router(config)# interface tunnel0` | Specifies the interface on which the tunnel will be configured and enters interface configuration mode. |
| **Step 6** | **ip address** *address mask*<br><br>**Example:**<br>`Router(config-if)# ip address 10.1.1.1 255.255.255.0` | Specifies the IP address and mask. |
| **Step 7** | **tunnel mode ipsec ipv4**<br><br>**Example:**<br>`Router(config-if)# tunnel mode ipsec ipv4` | Defines the mode for the tunnel. |
| **Step 8** | **tunnel source** *interface*<br><br>**Example:**<br>`Router(config-if)# tunnel source loopback0` | Specifies the tunnel source as a loopback interface. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 9** | **tunnel destination** *ip-address*<br><br>**Example:**<br>`Router(config-if)# tunnel destination 172.16.1.1` | Identifies the IP address of the tunnel destination. |
| **Step 10** | **tunnel protection IPsec profile** *profile-name* [**shared**]<br><br>**Example:**<br>`Router(config-if)# tunnel protection IPsec profile PROF` | Associates a tunnel interface with an IPsec profile. |

# Configuring Dynamic IPsec Virtual Tunnel Interfaces

This task shows how to configure a dynamic IPsec VTI.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto IPsec profile** *profile-name*
4. **set transform-set** *transform-set-name*
5. **interface virtual-template** *number*
6. **tunnel mode** *mode*
7. **tunnel protection IPsec profile** *profile-name* [**shared**]
8. **exit**
9. **crypto isakamp profile** *profile-name*
10. **virtual-template** *template-number*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **crypto IPsec profile** *profile-name*<br><br>**Example:**<br>`Router(config)# crypto IPsec profile PROF` | Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers. |
| **Step 4** | **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]<br><br>**Example:**<br>`Router(config)# set transform-set tset` | Specifies which transform sets can be used with the crypto map entry. |
| **Step 5** | **interface virtual-template** *number*<br><br>**Example:**<br>`Router(config)# interface virtual-template 2` | Defines a virtual-template tunnel interface and enters interface configuration mode. |
| **Step 6** | **tunnel mode ipsec ipv4**<br><br>**Example:**<br>`Router(config-if)# tunnel mode ipsec ipv4` | Defines the mode for the tunnel. |
| **Step 7** | **tunnel protection IPsec profile** *profile-name* [**shared**]<br><br>**Example:**<br>`Router(config-if)# tunnel protection IPsec profile PROF` | Associates a tunnel interface with an IPsec profile. |
| **Step 8** | **exit**<br><br>**Example:**<br>`Router(config-if)# exit` | Exits interface configuration mode. |
| **Step 9** | **crypto isakamp profile** *profile-name*<br><br>**Example:**<br>`Router(config)# crypto isakamp profile red` | Defines the ISAKAMP profile to be used for the virtual template. |
| **Step 10** | **virtual-template** *template-number*<br><br>**Example:**<br>`Router(config)# virtual-template 1` | Specifies the virtual template attached to the ISAKAMP profile. |

# Configuring Per-User Attributes on a Local Easy VPN AAA Server

To configure per-user attributes on a local Easy VPN AAA server, perform the following steps.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa attribute list** *list-name*
4. **attribute type** *name value* [**service** *service*] [**protocol** *protocol*]
5. **exit**
6. **crypto isakmp client configuration group** *group-name*
7. **crypto aaa attribute list** *list-name*

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **aaa attribute list** *list-name*<br><br>**Example:**<br>`Router(config)# aaa attribute list list1` | Defines a AAA attribute list locally on a router and enters attribute list configuration mode. |
| **Step 4** | **attribute type** *name value* [**service** *service*] [**protocol** *protocol*]<br><br>**Example:**<br>`Router(config-attr-list)# attribute type attribute xxxx service ike protocol ip` | Defines an attribute type that is to be added to an attribute list locally on a router. |
| **Step 5** | **exit**<br><br>**Example:**<br>`Router(config-attr-list)# exit` | Exits attribute list configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| Step 6 | **crypto isakmp client configuration group** *group-name*<br><br>**Example:**<br>`Router (config)# crypto isakmp client configuration group group1` | Specifies to which group a policy profile will be defined and enters ISAKMP group configuration mode. |
| Step 7 | **crypto aaa attribute list** *list-name*<br><br>**Example:**<br>`Router (config-isakmp-group)# crypto aaa attribute list listname1` | Defines a AAA attribute list locally on a router. |

# Configuration Examples for IPsec Virtual Tunnel Interface

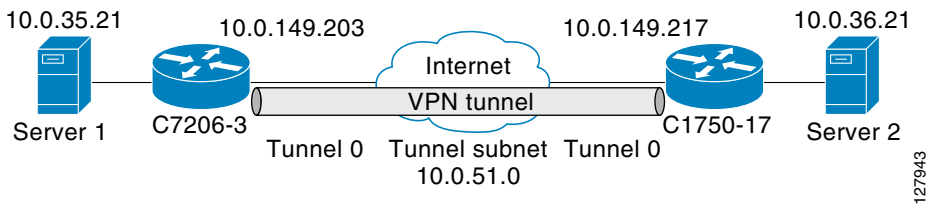
The following examples are provided to illustrate configuration scenarios for IPsec VTIs:



## Static Virtual Tunnel Interface with IPsec: Example

The following example configuration uses a preshared key for authentication between peers. VPN traffic is forwarded to the IPsec VTI for encryption and then sent out the physical interface. The tunnel on subnet 10 checks packets for IPsec policy and passes them to the Crypto Engine (CE) for IPsec encapsulation. Figure 5 illustrates the IPsec VTI configuration.

***Figure 5 VTI with IPsec***



### C7206 Router Configuration

```
version 12.3

service timestamps debug datetime
service timestamps log datetime
hostname 7200-3
no aaa new-model
ip subnet-zero
ip cef
controller ISA 6/1
!
crypto isakmp policy 1
encr 3des
authentication pre-share
group 2
crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto IPsec transform-set T1 esp-3des esp-sha-hmac
crypto IPsec profile P1
set transform-set T1
!


interface Tunnel0
 ip address 10.0.51.203 255.255.255.0
 ip ospf mtu-ignore
 load-interval 30
 tunnel source 10.0.149.203
 tunnel destination 10.0.149.217
 tunnel mode IPsec ipv4
 tunnel protection IPsec profile P1
!
interface Ethernet3/0
 ip address 10.0.149.203 255.255.255.0
 duplex full
!
interface Ethernet3/3
 ip address 10.0.35.203 255.255.255.0
 duplex full
!
ip classless
ip route 10.0.36.0 255.255.255.0 Tunnel0
line con 0
line aux 0
line vty 0 4
end
```

### C1750 Router Configuration

```
version 12.3

hostname c1750-17
no aaa new-model
ip subnet-zero
ip cef
crypto isakmp policy 1
encr 3des
authentication pre-share
group 2

crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto IPsec transform-set T1 esp-3des esp-sha-hmac
crypto IPsec profile P1
set transform-set T1
```

```
!
interface Tunnel0
 ip address 10.0.51.217 255.255.255.0
 ip ospf mtu-ignore
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
!
interface FastEthernet0/0
 ip address 10.0.149.217 255.255.255.0
 speed 100
 full-duplex
!
interface Ethernet1/0
 ip address 10.0.36.217 255.255.255.0
 load-interval 30
 full-duplex
!


ip classless
ip route 10.0.35.0 255.255.255.0 Tunnel0
line con 0
line aux 0
line vty 0 4
end
```

## Verifying the Results for the IPsec Static Virtual Tunnel Interface: Example

This section provides information that you can use to confirm that your configuration is working properly. In this display, Tunnel 0 is "up," and the line protocol is "up." If the line protocol is "down," the session is not active.

### Verifying the C7206 Status

```
Router# show interface tunnel 0

Tunnel0 is up, line protocol is up
Hardware is Tunnel
Internet address is 10.0.51.203/24
MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
reliability 255/255, txload 103/255, rxload 110/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.0.149.203, destination 10.0.149.217
Tunnel protocol/transport IPsec/IP, key disabled, sequencing disabled
Tunnel TTL 255

Checksumming of packets disabled, fast tunneling enabled
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Tunnel protection via IPsec (profile "P1")
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 1/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
30 second input rate 13000 bits/sec, 34 packets/sec
30 second output rate 36000 bits/sec, 34 packets/sec
191320 packets input, 30129126 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
```

```
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
59968 packets output, 15369696 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out

Router# show crypto session

Crypto session current status
Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.0.149.217 port 500
IKE SA: local 10.0.149.203/500 remote 10.0.149.217/500 Active
IPsec FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
Active SAs: 4, origin: crypto map

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C 10.0.35.0/24 is directly connected, Ethernet3/3
S 10.0.36.0/24 is directly connected, Tunnel0
C 10.0.51.0/24 is directly connected, Tunnel0
C 10.0.149.0/24 is directly connected, Ethernet3/0
```

# VRF-Aware Static Virtual Tunnel Interface: Example

To add VRF to the static VTI example, include the **ipvrf** and **ip vrf forwarding** commands to the configuration as shown in the following example.

### C7206 Router Configuration

```
hostname c7206
.
.
ip vrf sample-vti1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
.
.
interface Tunnel0
 ip vrf forwarding sample-vti1
 ip address 10.0.51.217 255.255.255.0
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
.
.
!
end
```

# Static Virtual Tunnel Interface with QoS: Example

You can apply any QoS policy to the tunnel endpoint by including the **service-policy** statement under the tunnel interface. The following example is policing traffic out the tunnel interface.

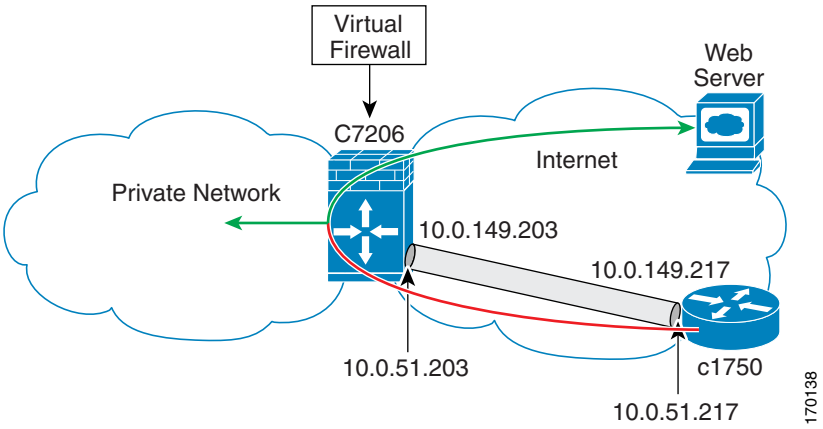
### C7206 Router Configuration

```
hostname c7206
.
.
class-map match-all VTI
 match any
!
policy-map VTI
  class VTI
  police cir 2000000
    conform-action transmit
    exceed-action drop
!
.
.
interface Tunnel0
 ip address 10.0.51.217 255.255.255.0
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
 service-policy output VTI
!
.
.
!
end
```

# Static Virtual Tunnel Interface with Virtual Firewall: Example

Applying the virtual firewall to the static VTI tunnel allows traffic from the spoke to pass through the hub to reach the internet. Figure 6 illustrates a static VTI with the spoke protected inherently by the corporate firewall.

***Figure 6 Static VTI with Virtual Firewall***




The basic static VTI configuration has been modified to include the virtual firewall definition.

### C7206 Router Configuration

```
hostname c7206
.
.
ip inspect max-incomplete high 1000000
ip inspect max-incomplete low 800000
ip inspect one-minute high 1000000
ip inspect one-minute low 800000
ip inspect tcp synwait-time 60
ip inspect tcp max-incomplete host 100000 block-time 2
ip inspect name IOSFW1 tcp timeout 300
ip inspect name IOSFW1 udp
!
.
.
interface GigabitEthernet0/1
 description Internet Connection
 ip address 172.18.143.246 255.255.255.0
 ip access-group 100 in
 ip nat outside
!
interface Tunnel0
 ip address 10.0.51.217 255.255.255.0
 ip nat inside
 ip inspect IOSFW1 in
 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
```

```
ip nat translation timeout 120
ip nat translation finrst-timeout 2
ip nat translation max-entries 300000
ip nat pool test1 10.2.100.1 10.2.100.50 netmask 255.255.255.0
ip nat inside source list 110 pool test1 vrf test-vti1 overload
!
access-list 100 permit esp any any
access-list 100 permit udp any eq isakmp any
access-list 100 permit udp any eq non500-isakmp any
access-list 100 permit icmp any any
access-list 110 deny   esp any any
access-list 110 deny   udp any eq isakmp any
access-list 110 permit ip any any
access-list 110 deny   udp any eq non500-isakmp any
!
end
```

# Dynamic Virtual Tunnel Interface Easy VPN Server: Example

The following example illustrates the use of the DVTI Easy VPN server, which serves as an IPsec remote access aggregator. The client can be a home user running a Cisco VPN client or it can be a Cisco IOS router configured as an Easy VPN client.

### C7206 Router Configuration

```
hostname c7206
!
aaa new-model
aaa authentication login local_list local
aaa authorization network local_list local
aaa session-id common
!
ip subnet-zero
ip cef
!
username cisco password 0 cisco123
!
controller ISA 1/1
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp client configuration group group1
 key cisco123
 pool group1pool
 save-password
!
crypto isakmp profile vpn1-ra
   match identity group group1
   client authentication list local_list
   isakmp authorization list local_list
   client configuration address respond
   virtual-template 1
!
crypto ipsec transform-set VTI-TS esp-3des esp-sha-hmac
!
crypto ipsec profile test-vti1
 set transform-set VTI-TS
!
```

```
interface GigabitEthernet0/1
 description Internet Connection
 ip address 172.18.143.246 255.255.255.0
!
interface GigabitEthernet0/2
 description Internal Network
 ip address 10.2.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered GigabitEthernet0/1
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
!
ip local pool group1pool 192.168.1.1 192.168.1.4
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
end
```

## Verifying the Results for the Dynamic Virtual Tunnel Interface Easy VPN Server: Example

The following examples show that a dynamic VTI has been configured for an Easy VPN server.

```
Router# show running-config interface Virtual-Access2

Building configuration...

Current configuration : 250 bytes
!
interface Virtual-Access2
 ip unnumbered GigabitEthernet0/1
 ip virtual-reassembly
 tunnel source 172.18.143.246
 tunnel destination 172.18.143.208
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
 no tunnel protection ipsec initiate
end

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.2.1.10 to network 0.0.0.0

     172.18.0.0/24 is subnetted, 1 subnets
C       172.18.143.0 is directly connected, GigabitEthernet0/1
     192.168.1.0/32 is subnetted, 1 subnets
S       192.168.1.1 [1/0] via 0.0.0.0, Virtual-Access2
     10.0.0.0/24 is subnetted, 1 subnets
C       10.2.1.0 is directly connected, GigabitEthernet0/2
S*   0.0.0.0/0 [1/0] via 172.18.143.1
```

# Dynamic Virtual Tunnel Interface Easy VPN Client: Example

The following example shows how you can set up a router as the Easy VPN client. This example uses basically the same idea as the Easy VPN client that you can run from a PC to connect. In fact, the configuration of the Easy VPN server will work for the software client or the Cisco IOS client.

```
hostname c1841
!
no aaa new-model
!
ip cef
!
username cisco password 0 cisco123
!
crypto ipsec client ezvpn CLIENT
 connect manual
 group group1 key cisco123
 mode client
 peer 172.18.143.246
 virtual-interface 1
 username cisco password cisco123
 xauth userid mode local
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
interface FastEthernet0/0
 description Internet Connection
 ip address 172.18.143.208 255.255.255.0
 crypto ipsec client ezvpn CLIENT
!
interface FastEthernet0/1
 ip address 10.1.1.252 255.255.255.0
 crypto ipsec client ezvpn CLIENT inside
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
!
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
end
```

The client definition can be set up in many different ways. The mode specified with the **connect** command can be automatic or manual. If the connect mode is set to manual, the IPsec tunnel has to be initiated manually by a user.

Also note use of the **mode** command. The mode can be client, network-extension, or network-extension-plus. This example indicates client mode, which means that the client is given a private address from the server. Network-extension mode is different from client mode in that the client specifies for the server its attached private subnet. Depending on the mode, the routing table on either end will be slightly different. The basic operation of the IPSec tunnel remains the same, regardless of the specified mode.

## Verifying the Results for the Dynamic Virtual Tunnel Interface Easy VPN Client: Example

The following examples illustrate different ways to display the status of the DVTI.

```
Router# show running-config interface Virtual-Access2

Building configuration...
```

```
Current configuration : 148 bytes
!
interface Virtual-Access2
 ip unnumbered Loopback1
 tunnel source FastEthernet0/0
 tunnel destination 172.18.143.246
 tunnel mode ipsec ipv4
end

Router# show running-config interface Loopback1

Building configuration...

Current configuration : 65 bytes
!
interface Loopback1
 ip address 192.168.1.1 255.255.255.255
end

Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 172.18.143.1 to network 0.0.0.0

     10.0.0.0/32 is subnetted, 1 subnets
C       10.1.1.1 is directly connected, Loopback0
     172.18.0.0/24 is subnetted, 1 subnets
C       172.18.143.0 is directly connected, FastEthernet0/0
     192.168.1.0/32 is subnetted, 1 subnets
C       192.168.1.1 is directly connected, Loopback1
S*   0.0.0.0/0 [1/0] via 172.18.143.1
               [1/0] via 0.0.0.0, Virtual-Access2

Router# show crypto ipsec client ezvpn

Easy VPN Remote Phase: 6

Tunnel name : CLIENT
Inside interface list: FastEthernet0/1
Outside interface: Virtual-Access2 (bound to FastEthernet0/0)
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
Address: 192.168.1.1
Mask: 255.255.255.255
Save Password: Allowed
Current EzVPN Peer: 172.18.143.246
```

# VRF-Aware IPsec with Dynamic VTI: Example

This example shows how to configure VRF-Aware IPsec to take advantage of the dynamic VTI:

```
hostname c7206
.
.
```

```
ip vrf test-vti1
 rd 1:1
 route-target export 1:1
 route-target import 1:1
!
.
.
interface Virtual-Template1 type tunnel
 ip vrf forwarding test-vti1
 ip unnumbered Loopback0
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
!
.
.
end
```

# Dynamic Virtual Tunnel Interface with Virtual Firewall: Example

The DVTI Easy VPN server can be configured behind a virtual firewall. Behind-the-firewall configuration allows users to enter the network, while the network firewall is protected from unauthorized access. The virtual firewall uses Context-Based Access Control (CBAC) and NAT applied to the Internet interface as well as to the virtual template.

```
hostname c7206
.
.
ip inspect max-incomplete high 1000000
ip inspect max-incomplete low 800000
ip inspect one-minute high 1000000
ip inspect one-minute low 800000
ip inspect tcp synwait-time 60
ip inspect tcp max-incomplete host 100000 block-time 2
ip inspect name IOSFW1 tcp timeout 300
ip inspect name IOSFW1 udp
!
.
.
interface GigabitEthernet0/1
 description Internet Connection
 ip address 172.18.143.246 255.255.255.0
 ip access-group 100 in
 ip nat outside
!
interface GigabitEthernet0/2
 description Internal Network
 ip address 10.2.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
 ip nat inside
 ip inspect IOSFW1 in
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
ip nat translation timeout 120
```

```
ip nat translation finrst-timeout 2
ip nat translation max-entries 300000
ip nat pool test1 10.2.100.1 10.2.100.50 netmask 255.255.255.0
ip nat inside source list 110 pool test1 vrf test-vti1 overload
!
access-list 100 permit esp any any
access-list 100 permit udp any eq isakmp any
access-list 100 permit udp any eq non500-isakmp any
access-list 100 permit icmp any any
access-list 110 deny   esp any any
access-list 110 deny   udp any eq isakmp any
access-list 110 permit ip any any
access-list 110 deny   udp any eq non500-isakmp any
!
end
```

# Dynamic Virtual Tunnel Interface with QoS: Example

You can add QoS to the DVTI tunnel by applying the service policy to the virtual template. When the template is cloned to make the virtual-access interface, the service policy will be applied there. The following example shows the basic DVTI configuration with QoS added.

```
hostname c7206
.
.
class-map match-all VTI
 match any
!
policy-map VTI
  class VTI
  police cir 2000000
    conform-action transmit
    exceed-action drop
!
.
.
interface Virtual-Template1 type tunnel
 ip vrf forwarding test-vti1
 ip unnumbered Loopback0
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
 service-policy output VTI
!
.
.
!
end
```

# Per-User Attributes on an Easy VPN Server: Example

The following example shows that per-user attributes have been configured on an Easy VPN server.

```
!

aaa new-model
!
!
aaa authentication login default local
```

```
aaa authentication login noAAA none
aaa authorization network default local
!
aaa attribute list per-group
 attribute type inacl "per-group-acl" service ike protocol ip mandatory
!
aaa session-id common
!
resource policy
!
ip subnet-zero
!
!
ip cef
!
!
username example password 0 example
!
!
crypto isakmp policy 3
 authentication pre-share
 group 2
crypto isakmp xauth timeout 90
!
crypto isakmp client configuration group PerUserAAA
 key cisco
 pool dpool
 crypto aaa attribute list per-group
!
crypto isakmp profile vi
 match identity group PerUserAAA
 isakmp authorization list default
 client configuration address respond
 client configuration group PerUserAAA
 virtual-template 1
!
!
crypto ipsec transform-set set esp-3des esp-sha-hmac
!
crypto ipsec profile vi
 set transform-set set
 set isakmp-profile vi
!
!
interface GigabitEthernet0/0
 description 'EzVPN Peer'
 ip address 192.168.1.1 255.255.255.128
 duplex full
 speed 100
 media-type rj45
 no negotiation auto
!
interface GigabitEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
 no negotiation auto

interface Virtual-Template1 type tunnel
 ip unnumbered GigabitEthernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile vi
```

```
!
ip local pool dpool 10.5.0.1 10.5.0.10
ip classless
!
no ip http server
no ip http secure-server
!
!
ip access-list extended per-group-acl
 permit tcp any any
 deny   icmp any any
logging alarm informational
logging trap debugging
!
control-plane
!
gatekeeper
 shutdown
!
line con 0
line aux 0
 stopbits 1
line vty 0 4
!
!
end
```

# Additional References

The following sections provide references related to the IPsec virtual tunnel interface feature.

## Related Documents

| Related Topic | Document Title |
|---|---|
| Security commands | *Cisco IOS Security Configuration Guide* |
| QoS | *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 15.0 |
| VPN | *Cisco IOS Security Configuration Guide: Secure Connectivity*, Release 15.0. |

## Standards

| Standard | Title |
|---|---|
| None. | — |

## MIBs

| MIB | MIBs Link |
|---|---|
| None. | To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

## RFCs

| RFC | Title |
|---|---|
| RFC 2401 | *Security Architecture for the Internet Protocol* |
| RFC 2408 | *Internet Security Association and Key Management Protocol* |
| RFC 2409 | *The Internet Key Exchange (IKE)* |

# Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.<br><br>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.<br><br>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

# Feature Information for IPsec Virtual Tunnel Interface

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

**Note** Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

*Table 1 Feature Information for IPsec Virtual Tunnel Interface*

| Feature Name | Releases | Feature Configuration Information |
|---|---|---|
| Static IPsec VTIs | 12.3(7)T<br>12.3(14)T<br>12.2(33)SRA<br>12.2(33)SXH | IPsec VTIs (VTIs) provide a routable interface type for terminating IPsec tunnels and an easy way to define protection between sites to form an overlay network. IPsec VTIs simplify configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing.<br>Static tunnel interfaces can be configured to encapsulate IPv6 or IPv4 packets in IPv6. |
| Dynamic IPsec VTIs | 12.3(7)T<br>12.3(14)T | Dynamic VTIs provide efficiency in the use of IP addresses and provide secure connectivity. Dynamic VTIs allow dynamically downloadable per-group and per-user policies to be configured on a RADIUS server. The per-group or per-user definition can be created using Xauth User or Unity group, or it can be derived from a certificate. Dynamic VTIs are standards based, so interoperability in a multiple-vendor environment is supported. IPsec dynamic VTIs allow you to create highly secure connectivity for remote access VPNs and can be combined with Cisco Architecture for Voice, Video, and Integrated Data (AVVID) to deliver converged voice, video, and data over IP networks. The dynamic VTI simplifies VRF-aware IPsec deployment. The VRF is configured on the interface. |
| Per-User Attribute Support for Easy VPN Servers | 12.4(9)T | This feature provides per-user attribute support on an Easy VPN server.<br>The following sections provide information about this feature:<br>• "Per-User Attribute Support for Easy VPN Servers" section on page 7<br>The following commands were added or modified by this feature: **crypto aaa attribute list** and **crypto isakmp client configuration group**. |