



IPsec Virtual Tunnel Interfaces

Last Updated: October 15, 2012

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 the configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing.



Note

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption](#) (NGE) white paper.

- [Finding Feature Information](#), page 1
- [Restrictions for IPsec Virtual Tunnel Interfaces](#), page 1
- [Information About IPsec Virtual Tunnel Interfaces](#), page 3
- [How to Configure IPsec Virtual Tunnel Interfaces](#), page 7
- [Configuration Examples for IPsec Virtual Tunnel Interfaces](#), page 16
- [Additional References](#), page 26
- [Feature Information for IPsec Virtual Tunnel Interfaces](#), page 27

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and 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 table at the end of this module.

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

Restrictions for IPsec Virtual Tunnel Interfaces



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

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. Therefore, the same IKE SA cannot be used for a crypto map.

IPsec SA Traffic Selectors

Static VTIs (SVTIs) 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 (DVTI) is also a point-to-point interface that can support multiple IPsec SAs. The DVTI can accept the multiple IPsec selectors that are proposed by the initiator.

IPv4 and IPv6 Packets

This feature supports SVTIs that are configured to encapsulate IPv4 packets or IPv6 packets, but IPv4 packets cannot carry IPv6 packets and IPv6 packets cannot carry IPv4 packets.

Proxy

SVTIs support only the “IP any any” proxy.

DVTIs support multiple proxies, but DVTIs do not allow mixing “any any” proxies with non-“any any” proxies. DVTIs permit only one type of proxy at a time, either a single “any any” proxy or multiple “no any any” proxies.

Quality of Service (QoS) Traffic Shaping

The shaped traffic is process switched.

Stateful Failover

IPsec stateful failover is not supported with IPsec VTIs.

Static VTIs Versus GRE Tunnels

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

Single Template Model

In the single template model, the VPN routing and forwarding (VRF) is configured in the ISAKMP profile. In this model, each virtual access that is created belongs to the internal VRF (IVRF) specified in the ISAKMP profile. But because the IP address of the virtual access is derived from the interface to which the virtual access is unnumbered to, the IP address of the interface will not be available in the virtual access routing table. This happens because the unnumbered interface does not belong to the IVRF routing table of the virtual access. In such cases, a ping to the virtual access IP address fails.

Tunnel Protection

Do not configure the **shared** keyword when using the **tunnel mode ipsec ipv4** command for IPsec IPv4 mode.

Virtual Template Lock

Effective with CSCtt26236, the virtual template lock allows you to modify or delete a virtual template of type tunnel only when the virtual template is not associated with any cloned virtual access interfaces. The virtual template lock prevents dynamic command updates from virtual templates to the cloned virtual access interfaces, which can cause instability in some scenarios.

If you try to modify or delete an active virtual template of type tunnel, the following error message appears:

```
Device(config)# interface virtual-template 1
% Virtual-template config is locked, active vaccess present
```

Although the virtual template cannot be modified when the virtual template is associated with a virtual access interface, perform the following steps to modify an existing virtual template configuration:

- 1 Configure a new virtual template interface. For more information, see “[Configuring Dynamic IPsec Virtual Tunnel Interfaces, page 14.](#)”
- 2 Associate the new virtual template to the IKEv2 profile. For more information, see the *Configuring IKEv2 Profile (Basic)* module.
- 3 Clear the active sessions using the **clear crypto session** command or wait for session termination.

The new session will use the new virtual template.

VRF-Aware IPsec Configuration

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

Information About IPsec Virtual Tunnel Interfaces

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. Because DVTIs function like any other real interface you can apply quality of service (QoS), firewall, and other security services as soon as the tunnel is active.

Without 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, page 4](#)

- [Static Virtual Tunnel Interfaces, page 4](#)
- [Dynamic Virtual Tunnel Interfaces, page 5](#)
- [Dynamic Virtual Tunnel Interface Life Cycle, page 6](#)
- [Routing with IPsec Virtual Tunnel Interfaces, page 6](#)
- [Traffic Encryption with the IPsec Virtual Tunnel Interface, page 6](#)

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 Network Address Translation (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 extra features to the IPsec tunnel.

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

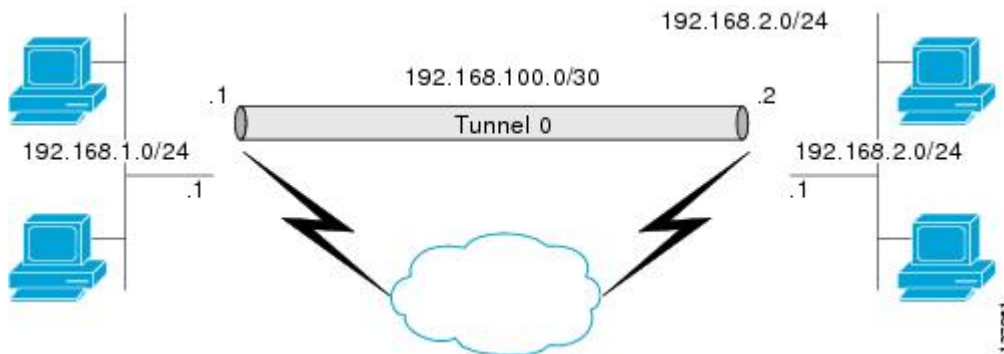
Static Virtual Tunnel Interfaces

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 24 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.

The figure below illustrates how a SVTI is used.

Figure 1 IPsec SVTI



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



Note

When configuring IPsec SVTI with high availability (HA), the standby router reload does not affect the existing security associations.

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.

DVTIs can be used for both the server and the 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.

DVTIs function like any other real interface, so you can apply QoS, firewall, or 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.

DVTIs provide efficiency in the use of IP addresses and provide secure connectivity. DVTIs 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 an extended authentication (Xauth) User or Unity group, or can be derived from a certificate. DVTIs 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 VPN 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. DVTIs are used in hub-and-spoke configurations. A single DVTI can support several static VTIs.

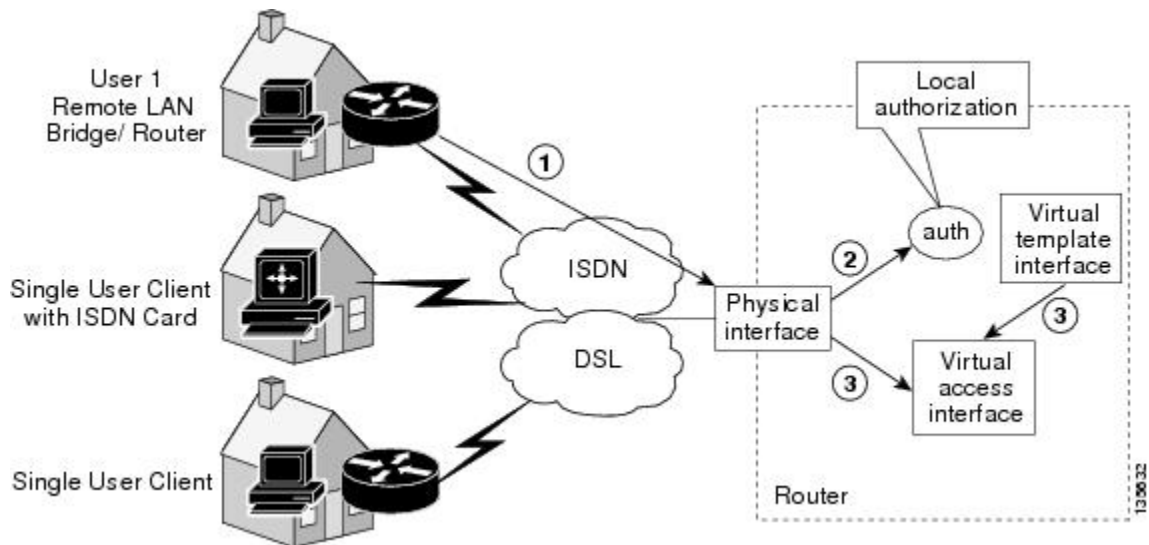


Note

DVTI is supported only in Easy VPNs. That is, the DVTI end must be configured as an Easy VPN server.

The figure below illustrates the DVTI authentication path.

Figure 2 Dynamic IPsec VTI



The authentication shown in the figure above follows this path:

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

Dynamic Virtual Tunnel Interface Life Cycle

IPsec profiles define the policy for DVTIs. 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

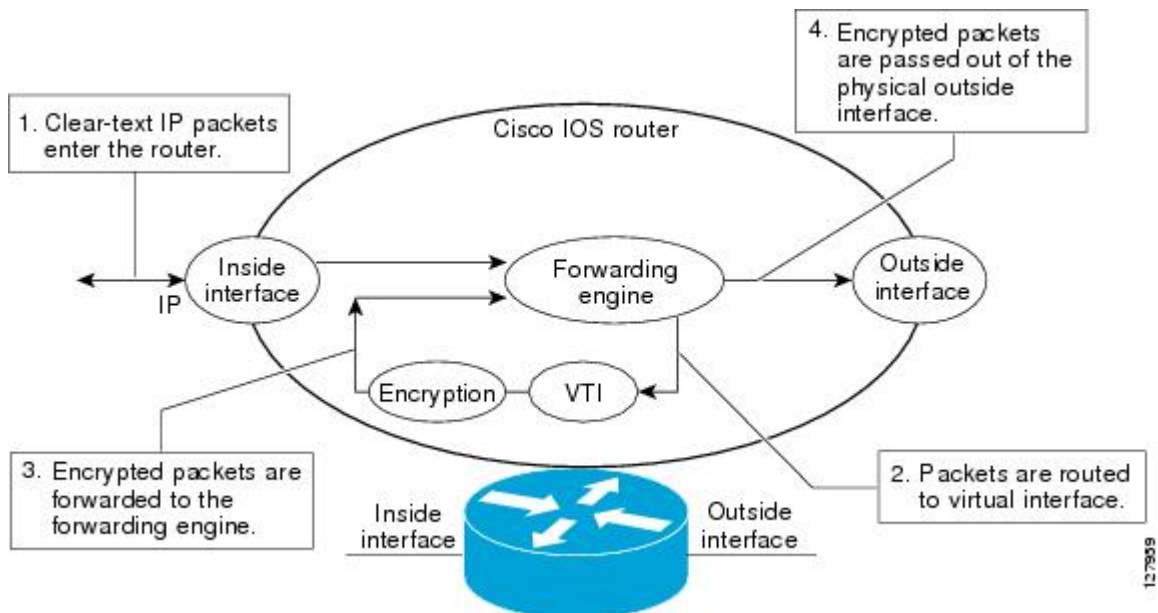
Because VTIs are routable interfaces, routing plays an important role in the encryption process. Traffic is encrypted only if it is forwarded out of the VTI, and traffic arriving on the VTI is decrypted and routed accordingly. VTIs allow you to establish an encryption tunnel using a real interface as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

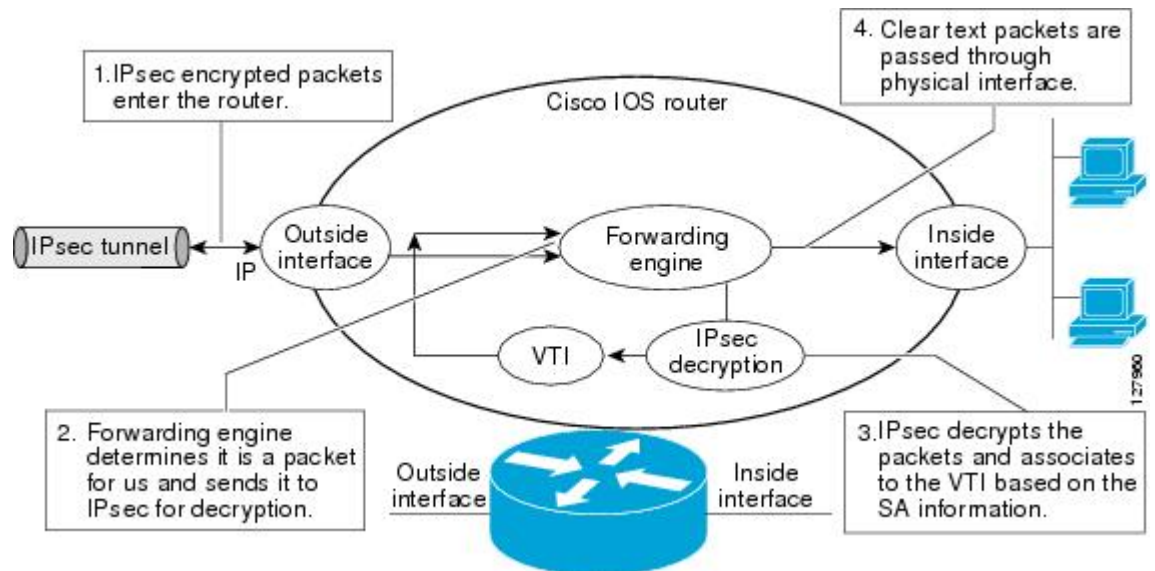
Figure 3 Packet Flow into the IPsec Tunnel



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.

The figure below shows the packet flow out of the IPsec tunnel.

Figure 4 Packet Flow out of the IPsec Tunnel



How to Configure IPsec Virtual Tunnel Interfaces

- [Configuring Static IPsec Virtual Tunnel Interfaces](#), page 8
- [Configuring User-Defined Static IPsec Virtual Tunnel Interfaces](#), page 10
- [Configuring Dynamic IPsec Virtual Tunnel Interfaces](#), page 14

Configuring Static IPsec Virtual Tunnel Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto IPsec profile** *profile-name*
4. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
5. **exit**
6. **interface** *type number*
7. **ip address** *address mask*
8. **tunnel mode ipsec ipv4**
9. **tunnel source** *interface-type interface-type*
10. **tunnel destination** *ip-address*
11. **tunnel protection IPsec profile** *profile-name* [**shared**]
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	crypto IPsec profile <i>profile-name</i> Example: Device(config)# crypto IPsec profile PROF	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices, and enters IPsec profile configuration mode.
Step 4	set transform-set <i>transform-set-name</i> [<i>transform-set-name2...transform-set-name6</i>] Example: Device(ipsec-profile)# set transform-set tset	Specifies which transform sets can be used with the crypto map entry.

	Command or Action	Purpose
Step 5	exit Example: Device(ipsec-profile)# exit	Exits IPsec profile configuration mode, and enters global configuration mode.
Step 6	interface <i>type number</i> Example: Device(config)# interface tunnel 0	Specifies the interface on which the tunnel will be configured and enters interface configuration mode.
Step 7	ip address <i>address mask</i> Example: Device(config-if)# ip address 10.1.1.1 255.255.255.0	Specifies the IP address and mask.
Step 8	tunnel mode ipsec ipv4 Example: Device(config-if)# tunnel mode ipsec ipv4	Defines the mode for the tunnel.
Step 9	tunnel source <i>interface-type interface-type</i> Example: Device(config-if)# tunnel source loopback 0	Specifies the tunnel source as a loopback interface.
Step 10	tunnel destination <i>ip-address</i> Example: Device(config-if)# tunnel destination 172.16.1.1	Identifies the IP address of the tunnel destination.
Step 11	tunnel protection IPsec profile <i>profile-name</i> [shared] Example: Device(config-if)# tunnel protection IPsec profile PROF	Associates a tunnel interface with an IPsec profile.

Command or Action	Purpose
Step 12 end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring User-Defined Static IPsec Virtual Tunnel Interfaces

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto keyring *keyring name*
4. crypto keyring vrf *vrf name*
5. pre-shared-key address *address*
6. pre-shared-key address key *key*
7. exit
8. interface *type number*
9. ip vrf forwarding *vrf name*
10. ip address *address mask*
11. tunnel source *address*
12. tunnel destination *address*
13. tunnel mode ipsec ipv4
14. tunnel vrf *vrf name*
15. tunnel protection IPsec profile *profile name*
16. exit
17. interface *interface-type*
18. ip vrf forwarding *vrf name*
19. ip address *ip-address*
20. no shutdown
21. end

DETAILED STEPS

Command or Action	Purpose
Step 1 enable Example: Device> enable	Enables privileged Exec mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enables global configuration mode.
Step 3	crypto keyring <i>keyring name</i> Example: Device(config)# crypto keyring KR-FVRF	Defines the crypto keyring name to which the keyring will be referenced.
Step 4	crypto keyring vrf <i>vrf name</i> Example: Device(config)# crypto keyring vrf FVRF	Defines the crypto keyring name for a virtual routing and forwarding (VRF) instance to which the keyring will be referenced.
Step 5	pre-shared-key address <i>address</i> Example: Device(config-keyring)# pre-shared-key address 10.0.0.2	Defines the preshared key IP address of the remote host.
Step 6	pre-shared-key address key <i>key</i> Example: Device(config-keyring)# pre-shared-key address 10.0.0.2 key cisco	Defines the secret key for the preshared key IP address of the remote host.
Step 7	exit Example: Device(config-keyring)# exit	Exits keyring configuration mode, and enters global configuration mode.
Step 8	interface <i>type number</i> Example: Device(config)# interface tunnel 0	Specifies the interface on which the tunnel will be configured and enters interface configuration mode.

Command or Action	Purpose
<p>Step 9 ip vrf forwarding <i>vrf name</i></p> <p>Example: Device(config-if)# ip vrf forwarding IVRF</p>	Specifies the virtual routing and forwarding (VRF) name for the interface.
<p>Step 10 ip address <i>address mask</i></p> <p>Example: Device(config-if)# ip address 10.0.0.1 255.255.255.0</p>	Specifies the IP address and address mask.
<p>Step 11 tunnel source <i>address</i></p> <p>Example: Device(config-if)# tunnel source 10.0.0.1</p>	Specifies the IP address of the tunnel source.
<p>Step 12 tunnel destination <i>address</i></p> <p>Example: Device(config-if)# tunnel destination 10.0.0.2</p>	Specifies the IP address of the tunnel destination.
<p>Step 13 tunnel mode ipsec ipv4</p> <p>Example: Device(config-if)# tunnel mode ipsec ipv4</p>	Defines the mode for the tunnel.
<p>Step 14 tunnel vrf <i>vrf name</i></p> <p>Example: Device(config-if)# tunnel vrf FVRF</p>	Defines the virtual routing and forwarding (VRF) instance for the tunnel.
<p>Step 15 tunnel protection IPsec profile <i>profile name</i></p> <p>Example: Device(config-if)# tunnel protection IPsec profile IPSec-Profile</p>	Associates a tunnel interface with an IPsec profile.

Command or Action	Purpose
Step 16 exit Example: Device(config-if)# exit	Exits interface configuration mode, and enters global configuration mode.
Step 17 interface <i>interface-type</i> Example: Device(config)# interface FastEthernet0/0	Specifies the interface type.
Step 18 ip vrf forwarding <i>vrf name</i> Example: Device(config-if)# ip vrf forwarding FVRF	Specifies the VRF forwarding table for the interface.
Step 19 ip address <i>ip-address</i> Example: Device(config-if)# ip address 10.0.0.1 255.255.255.0	Specifies the IP address and address mask.
Step 20 no shutdown Example: Device(config-if)# no shutdown	Enables the interface (brings it up).
Step 21 end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Dynamic IPsec Virtual Tunnel Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *profile-name*
4. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
5. **exit**
6. **interface virtual-template** *number*
7. **tunnel mode ipsec ipv4**
8. **tunnel protection IPsec profile** *profile-name* [**shared**]
9. **exit**
10. **crypto isakamp profile** *profile-name*
11. **match identity address** *ip-addressmask*
12. **virtual template** *template-number*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	crypto ipsec profile <i>profile-name</i> Example: Device(config)# crypto ipsec profile PROF	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters IPsec profile configuration mode.

	Command or Action	Purpose
Step 4	<p>set transform-set <i>transform-set-name</i> [<i>transform-set-name2...transform-set-name6</i>]</p> <p>Example:</p> <pre>Device(ipsec-profile)# set transform-set tset</pre>	Specifies which transform sets can be used with the crypto map entry.
Step 5	<p>exit</p> <p>Example:</p> <pre>Device(ipsec-profile)# exit</pre>	Exits ipsec profile configuration mode and enters global configuration mode.
Step 6	<p>interface virtual-template <i>number</i></p> <p>Example:</p> <pre>Device(config)# interface virtual-template 2</pre>	Defines a virtual-template tunnel interface and enters interface configuration mode.
Step 7	<p>tunnel mode ipsec ipv4</p> <p>Example:</p> <pre>Device(config-if)# tunnel mode ipsec ipv4</pre>	Defines the mode for the tunnel.
Step 8	<p>tunnel protection IPsec profile <i>profile-name</i> [shared]</p> <p>Example:</p> <pre>Device(config-if)# tunnel protection ipsec profile PROF</pre>	Associates a tunnel interface with an IPsec profile.
Step 9	<p>exit</p> <p>Example:</p> <pre>Device(config-if)# exit</pre>	Exits interface configuration mode.
Step 10	<p>crypto isakamp profile <i>profile-name</i></p> <p>Example:</p> <pre>Device(config)# crypto isakamp profile profile1</pre>	Defines the ISAKAMP profile to be used for the virtual template.

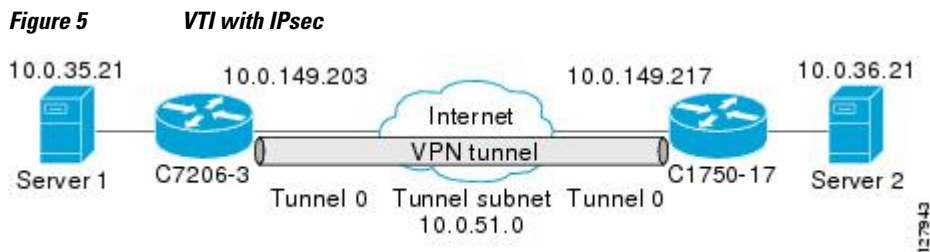
Command or Action	Purpose
Step 11 <code>match identity address ip-addressmask</code> Example: <pre>Device(conf-isa-prof)# match identity address 10.1.1.0 255.255.255.0</pre>	Matches an identity from the ISAKMP profile and enters isakmp-profile configuration mode.
Step 12 <code>virtual template template-number</code> Example: <pre>Device(config)# virtual-template 1</pre>	Specifies the virtual template attached to the ISAKAMP profile.
Step 13 <code>end</code> Example: <pre>Device(config)# end</pre>	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for IPsec Virtual Tunnel Interfaces

- [Example: Static Virtual Tunnel Interface with IPsec, page 16](#)
- [Example: VRF-Aware Static Virtual Tunnel Interface, page 19](#)
- [Example: Static Virtual Tunnel Interface with QoS, page 19](#)
- [Example: Static Virtual Tunnel Interface with Virtual Firewall, page 20](#)
- [Example: Dynamic Virtual Tunnel Interface Easy VPN Server, page 21](#)
- [Example: Dynamic Virtual Tunnel Interface Easy VPN Client, page 22](#)
- [Example: VRF-Aware IPsec with Dynamic VTI, page 24](#)
- [Example: Dynamic Virtual Tunnel Interface with Virtual Firewall, page 24](#)
- [Example: Dynamic Virtual Tunnel Interface with QoS, page 25](#)

Example: Static Virtual Tunnel Interface with IPsec

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 the IPsec policy and passes them to the Crypto Engine (CE) for IPsec encapsulation. The figure below illustrates the IPsec VTI configuration.



Cisco 7206 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 aes
authentication pre-share
group 14
crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set T1 esp-aes 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
```

Cisco 1750 Router Configuration

```
version 12.3
hostname c1750-17
no aaa new-model
ip subnet-zero
ip cef
crypto isakmp policy 1
encr aes
authentication pre-share
group 14
crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set T1 esp-aes 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

```

- [Example: Verifying the Results for the IPsec Static Virtual Tunnel Interface, page 18](#)

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

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 Cisco 7206 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

```

Example: VRF-Aware Static Virtual Tunnel Interface

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

Cisco 7206 Router Configuration

```

hostname cisco 7206
.
.
ip vrf sample-vtil
rd 1:1
route-target export 1:1
route-target import 1:1
!
.
.
interface Tunnel0
ip vrf forwarding sample-vtil
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

```

Example: Static Virtual Tunnel Interface with QoS

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

Cisco 7206 Router Configuration

```

hostname cisco 7206
.
.
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

```

```

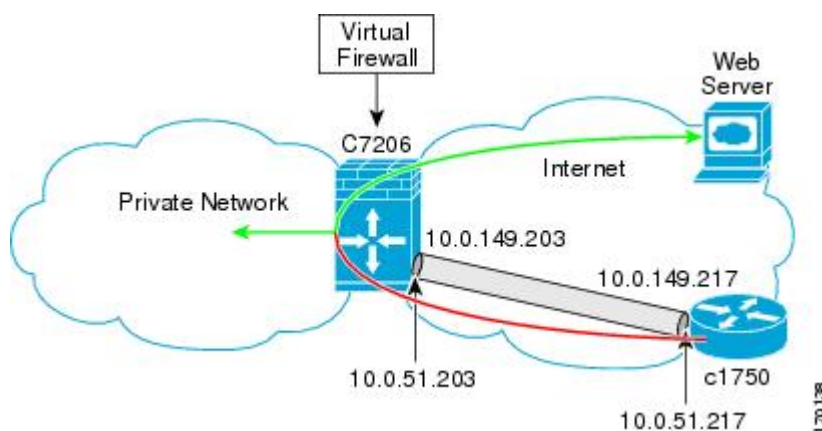
tunnel mode ipsec ipv4
tunnel protection ipsec profile P1
service-policy output VTI
!
.
!
end

```

Example: Static Virtual Tunnel Interface with Virtual Firewall

Applying the virtual firewall to the SVTI tunnel allows traffic from the spoke to pass through the hub to reach the Internet. The figure below illustrates an SVTI with the spoke protected inherently by the corporate firewall.

Figure 6 Static VTI with Virtual Firewall



The basic SVTI configuration has been modified to include the virtual firewall definition:

Cisco 7206 Router Configuration

```

hostname cisco 7206
.
.
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-vt11 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

```

Example: Dynamic Virtual Tunnel Interface Easy VPN Server

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 a Cisco IOS router configured as an Easy VPN client.

Cisco 7206 Router Configuration

```

hostname cisco 7206
!
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 aes
  authentication pre-share
  group 14
!
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-aes esp-sha-hmac
!
crypto ipsec profile test-vt11
  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-vtil
!
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

```

- [Example: Verifying the Results for the Dynamic Virtual Tunnel Interface Easy VPN Server, page 22](#)

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

The following examples show that a DVTI 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-vtil
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

```

Example: Dynamic Virtual Tunnel Interface Easy VPN Client

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

```

hostname cisco 1841
!
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-Templat1 type tunnel
 ip unnumbered Loopback0
!
ip route 0.0.0.0 0.0.0.0 172.18.143.1 254
!
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.

Note the use of the **mode** command. The mode can be a client, network-extension, or network-extension-plus. This example indicates the client mode, which means that the client is given a private address from the server. The network-extension mode is different from the 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.

- [Example: Verifying the Results for the Dynamic Virtual Tunnel Interface Easy VPN Client, page 23](#)

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

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 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

```

Example: VRF-Aware IPsec with Dynamic VTI

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

```

hostname c7206
.
.
ip vrf test-vtil
 rd 1:1
  route-target export 1:1
  route-target import 1:1
!
.
.
interface Virtual-Templatel type tunnel
 ip vrf forwarding test-vtil
 ip unnumbered Loopback0
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vtil
!
.
.
end

```

Example: Dynamic Virtual Tunnel Interface with Virtual Firewall

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 cisco 7206
.
.
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-Templat1 type tunnel
ip unnumbered Loopback0
ip nat inside
ip inspect IOSFW1 in
tunnel mode ipsec ipv4
tunnel protection ipsec profile test-vt1
!
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-vt1 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

```

Example: Dynamic Virtual Tunnel Interface with QoS

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 also be applied to the virtual access interface. The following example shows the basic DVTI configuration with QoS added.

```

hostname cisco 7206
.
.
class-map match-all VTI
match any
!
policy-map VTI
class VTI
police cir 2000000
conform-action transmit
exceed-action drop
!
.
.
interface Virtual-Templat1 type tunnel
ip vrf forwarding test-vt1
ip unnumbered Loopback0

```

```

ip virtual-reassembly
tunnel mode ipsec ipv4
tunnel protection ipsec profile test-vtil
service-policy output VTI
!
.
.
!
end

```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	<ul style="list-style-type: none"> • Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IPsec configuration	Configuring Security for VPNs with IPsec
QoS configuration	Cisco IOS Quality of Service Solutions Configuration Guide
VPN configuration	<ul style="list-style-type: none"> • Cisco Easy VPN Remote • Easy VPN Server
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/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 and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPsec Virtual Tunnel Interfaces

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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

Table 1 Feature Information for IPsec Virtual Tunnel Interfaces

Feature Name	Releases	Feature Configuration Information
Dynamic IPsec VTIs	12.3(7)T 12.3(14)T	Dynamic VTIs enable efficient 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. IPsec dynamic VTIs allow you to create highly secure connectivity for remote access VPNs. The dynamic VTI simplifies VRF-aware IPsec deployment. The following commands were introduced or modified: crypto isakmp profile, interface virtual-template, show vtemplate, tunnel mode, virtual-template.

Feature Name	Releases	Feature Configuration Information
Multi-SA for Dynamic VTIs	15.2(1)T	<p>The DVTI can accept multiple IPsec selectors that are proposed by the initiator.</p> <p>The following commands were introduced or modified: set security-policy limit, set reverse-route.</p>
Static IPsec VTIs	12.2(33)SRA 12.2(33)SXH 12.3(7)T 12.3(14)T	<p>IPsec 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.</p>

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2012 Cisco Systems, Inc. All rights reserved.