CHAPTER 6

Configuring Security Features

The Cisco 800M Series ISR provides the following security features:

Configuring Authentication, Authorization, and Accounting

Authentication, Authorization, and Accounting (AAA) network security services provide the primary framework through which you set up access control on your router. Authentication provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and encryption depending on the security protocol you choose. Authorization provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, Internetwork Packet Exchange (IPX), AppleTalk Remote Access (ARA), and Telnet. Accounting provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

AAA uses protocols such as Remote Authentication Dial-In User Service (RADIUS), Terminal Access Controller Access Control System Plus (TACACS+), or Kerberos to administer its security functions. If your router is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS, TACACS+, or Kerberos security server.

For information about configuring AAA services and supported security protocols, see the following guide:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_aaa/configuration/15-mt/sec-usr-aaa-15-mt-book.html

Configuring Access Lists

Access lists permit or deny network traffic over an interface, based on source IP address, destination IP address, or protocol. Access lists are configured as standard or extended. A standard access list either permits or denies passage of packets from a designated source. An extended access list allows designation of both the destination and the source, and it allows designation of individual protocols to be permitted or denied passage.

An access list is a series of commands with a common tag to bind them together. The tag is either a number or a name. Table 6-1 lists the commands used to configure access lists.

Table 6-1 Access List Configuration Commands

Access Control List (ACL) Type	Configuration Commands
Numbered	
Standard	**access-list** {**1-99**}{**permit** | **deny**} *source-addr* [*source-mask*]
Extended	**access-list** {**100-199**}{**permit** | **deny**} *protocol source-addr* [*source-mask*] *destination-addr* [*destination-mask*]
Named	
Standard	**ip access-list standard** *name* **deny** {*source* | *source-wildcard* | **any**}
Extended	**ip access-list extended** *name* **{permit** | **deny}** *protocol* {*source-addr* [*source-mask*] | **any**}{*destination-addr* [*destination-mask*] | **any**}

For more complete information on creating access lists, see the following web link: http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_acl/configuration/15-mt/sec-data-acl-15-mt-book.html

Access Groups

An access group is a sequence of access list definitions bound together with a common name or number. An access group is enabled for an interface during interface configuration. Use the following guidelines when creating access groups:

- The order of access list definitions is significant. A packet is compared against the first access list in the sequence. If there is no match (that is, if neither a permit nor a deny occurs), the packet is compared with the next access list, and so on.
- All parameters must match the access list before the packet is permitted or denied.
- There is an implicit “deny all” at the end of all sequences.

For information on configuring and managing access groups, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_acl/configuration/15-mt/sec-data-acl-15-mt-book/sec-create-ip-al-filter.html

Configuring Cisco IOS IPS

The Cisco IOS Intrusion Prevention System (IPS) acts as an in-line intrusion detection sensor, watching packets and sessions as they flow through the router and scanning each packet to match any of the Cisco IOS IPS signatures. When Cisco IOS IPS detects suspicious activity, it responds before network security can be compromised and logs the event through Cisco IOS syslog messages or Security Device Event Exchange (SDEE). The network administrator can configure Cisco IOS IPS to choose the appropriate response to various threats. When packets in a session match a signature, Cisco IOS IPS can take any of the following actions, as appropriate:

- Send an alarm to a syslog server or a centralized management interface
- Drop the packet
- Reset the connection
- Deny traffic from the source IP address of the attacker for a specified amount of time
- Deny traffic on the connection for which the signature was seen for a specified amount of time

For more information about configuring Cisco IOS IPS see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios/sec_data_plane/configuration/guide/convert/sec_data_ios_ips_15_1_book/sec_cfg_ips.html

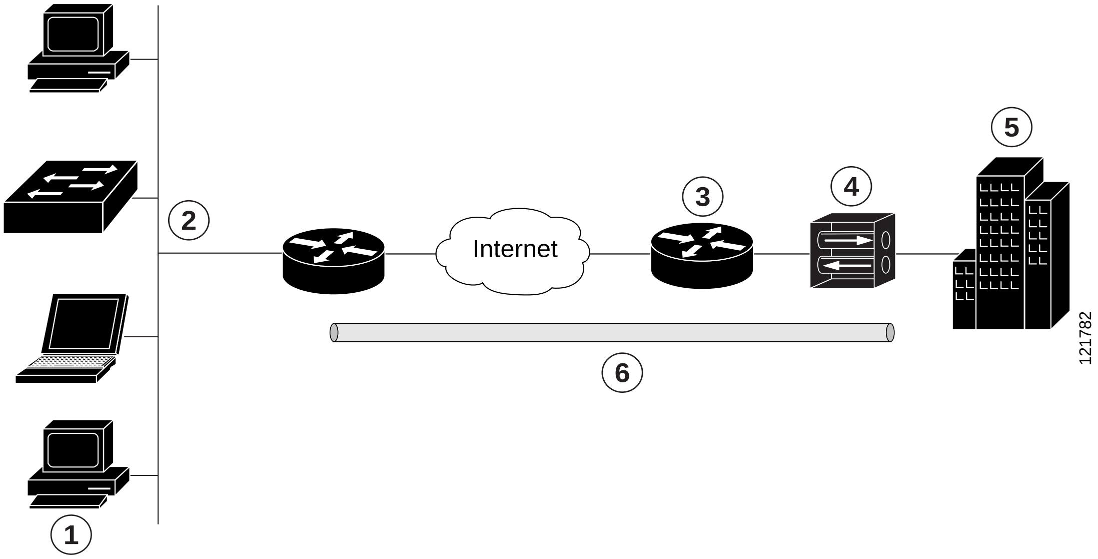
Configuring VPN

A Virtual Private Network (VPN) connection provides a secure connection between two networks over a public network such as the Internet. Cisco 800M Series ISRs support two types of VPNs: site-to-site and remote access. Remote access VPNs are used by remote clients to log in to a corporate network. Site-to-site VPNs connect branch offices to corporate offices. This section gives examples for site-to-site and remote access VPNs.

Remote Access VPN Example

The configuration of a remote access VPN uses Cisco Easy VPN and an IP Security (IPSec) tunnel to configure and secure the connection between the remote client and the corporate network. Figure 6-1 shows a typical deployment scenario.

Figure 6-1 Remote Access VPN Using IPSec Tunnel



1	Remote networked users
2	VPN client—Cisco 800M Series ISR
3	Router—Provides corporate office network access
4	VPN server—Easy VPN server; for example, a Cisco VPN 3000 concentrator with outside interface address 210.110.101.1
5	Corporate office with a network address of 10.1.1.1
6	IPSec tunnel

The Cisco Easy VPN client feature eliminates much of the tedious configuration work by implementing the Cisco Unity Client protocol. This protocol allows most VPN parameters, such as internal IP addresses, internal subnet masks, DHCP server addresses, Windows Internet Naming Service (WINS) server addresses, and split-tunneling flags, to be defined at a VPN server, such as a Cisco VPN 3000 series concentrator that is acting as an IPSec server.

A Cisco Easy VPN server–enabled device can terminate VPN tunnels initiated by mobile and remote workers who are running Cisco Easy VPN Remote software on PCs. Cisco Easy VPN server–enabled devices allow remote routers to act as Cisco Easy VPN Remote nodes.

The Cisco Easy VPN client feature can be configured in one of two modes—client mode or network extension mode. Client mode is the default configuration and allows only devices at the client site to access resources at the central site. Resources at the client site are unavailable to the central site. Network extension mode allows users at the central site (where the Cisco VPN 3000 series concentrator is located) to access network resources on the client site.

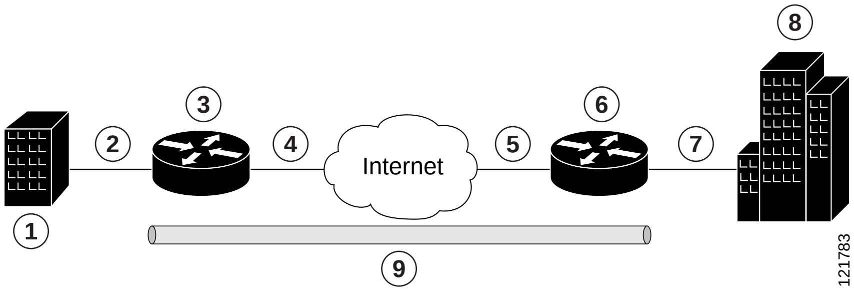
After the IPSec server has been configured, a VPN connection can be created with minimal configuration on an IPSec client. When the IPSec client initiates the VPN tunnel connection, the IPSec server pushes the IPSec policies to the IPSec client and creates the corresponding VPN tunnel connection.

Note The Cisco Easy VPN client feature supports configuration of only one destination peer. If your application requires creation of multiple VPN tunnels, you must manually configure the IPSec VPN and Network Address Translation/Peer Address Translation (NAT/PAT) parameters on both the client and the server.

Site-to-Site VPN Example

The configuration of a site-to-site VPN uses IPSec and the generic routing encapsulation (GRE) protocol to secure the connection between the branch office and the corporate network. Figure 6-2 shows a typical deployment scenario.

Figure 6-2 Site-to-Site VPN Using an IPSec Tunnel and GRE



1	Branch office containing multiple LANs and VLANs
2	Gigabit Ethernet LAN interface—With address 192.165.0.0/16 (also the inside interface for NAT)
3	VPN client—Cisco 800M Series ISR
4	Gigabit Ethernet interface—With address 200.1.1.1 (also the outside interface for NAT)
5	LAN interface—Connects to the Internet; with outside interface address of 210.110.101.1
6	VPN client—Another router, which controls access to the corporate network
7	LAN interface—Connects to the corporate network; with inside interface address of 10.1.1.1
8	Corporate office network
9	IPSec tunnel with GRE

For more information about IPSec and GRE configuration, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_vpnips/configuration/15-mt/sec-sec-for-vpns-w-ipsec-15-mt-book/sec-cfg-vpn-ipsec.html

Configuration Examples

Each example configures a VPN over an IPSec tunnel, using the procedure given in the "Configure a VPN over an IPSec Tunnel" section on page 68. Then, the specific procedure for a remote access configuration is given, followed by the specific procedure for a site-to-site configuration.

The examples shown in this chapter apply only to the endpoint configuration on the Cisco 800M Series ISRs. Any VPN connection requires both endpoints to be properly configured in order to function. See the software configuration documentation as needed to configure VPN for other router models.

VPN configuration information must be configured on both endpoints. You must specify parameters such as internal IP addresses, internal subnet masks, DHCP server addresses, and Network Address Translation (NAT).

- "Configure a VPN over an IPSec Tunnel" section on page 68
- "Create a Cisco Easy VPN Remote Configuration" section on page 77
- "Configure a Site-to-Site GRE Tunnel" section on page 80

Configure a VPN over an IPSec Tunnel

Perform the following tasks to configure a VPN over an IPSec tunnel:

- Configure the IKE Policy, page 69
- Configure Group Policy Information, page 70
- Apply Mode Configuration to the Crypto Map, page 72
- Enable Policy Lookup, page 73
- Configure IPSec Transforms and Protocols, page 74
- Configure the IPSec Crypto Method and Parameters, page 75
- Apply the Crypto Map to the Physical Interface, page 76
- Where to Go Next, page 77

Configure the IKE Policy

To configure the Internet Key Exchange (IKE) policy, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto isakmp policy** *priority*
2. **encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256}**
3. **hash** {**md5** | **sha**}
4. **authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
5. **group** {**1** | **2** | **5**}
6. **lifetime** *seconds*
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 1`	Creates an IKE policy that is used during IKE negotiation. The priority is a number from 1 to 10000, with 1 being the highest. Also enters the ISAKMP[1] policy configuration mode.

Step 2	**encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256}**
	Example:
	`Router(config-isakmp)# encryption 3des`
		Specifies the encryption algorithm used in the IKE policy.
		The example specifies 168-bit DES[2].

Step 3	**hash** {**md5** | **sha**}
	Example:
	`Router(config-isakmp)# hash md5`
		Specifies the hash algorithm used in the IKE policy.
		The example specifies the MD5[3] algorithm. The default is SHA-1[4].

Step 4	**authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
	Example:
	`Router(config-isakmp)# authentication pre-share`
		Specifies the authentication method used in the IKE policy.
		The example specifies a pre-shared key.

Step 5	**group** {**1** | **2** | **5**}
	Example:
	`Router(config-isakmp)# group 2`
		Specifies the Diffie-Hellman group to be used in an IKE policy.

	Command or Action	Purpose
Step 6	**lifetime** *seconds* **Example:** `Router(config-isakmp)# lifetime 480`	Specifies the lifetime, from 60 to 86400 seconds, for an IKE SA[5].
Step 7	**exit** **Example:** `Router(config-isakmp)# exit`	Exits IKE policy configuration mode and enters global configuration mode.

1. ISAKMP = Internet Security Association Key and Management Protocol
2. DES = data encryption standard
3. MD5 = Message Digest 5
4. SHA-1 = Secure Hash standard
5. SA = security association

Configure Group Policy Information

To configure the group policy, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto isakmp client configuration group** {*group-name* | *default*}
2. **key** ***name***
3. **dns** ***primary-server***
4. **domain** ***name***
5. **exit**
6. **ip local pool {default |** ***poolname*****} [*****low-ip-address*** **[*****high-ip-address*****]]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto isakmp client configuration group** {*group-name* | *default*} **Example:** `Router(config)# crypto isakmp client configuration group rtr-remote`	Creates an IKE policy group containing attributes to be downloaded to the remote client. Also enters the ISAKMP group policy configuration mode.
Step 2	**key** *name* **Example:** `Router(config-isakmp-group)# key secret-password`	Specifies the IKE pre-shared key for the group policy.

	Command or Action	Purpose
Step 3	**dns** *primary-server* **Example:** `Router(config-isakmp-group)# dns 10.50.10.1`	Specifies the primary DNS[1] server for the group. You may also want to specify WINS[2] servers for the group by using the **wins** command.
Step 4	**domain** *name* **Example:** `Router(config-isakmp-group)# domain company.com`	Specifies group domain membership.
Step 5	**exit** **Example:** `Router(config-isakmp-group)# exit`	Exits IKE group policy configuration mode and enters global configuration mode.
Step 6	**ip local pool** {**default** | *poolname*} [*low-ip-address* [*high-ip-address*]] **Example:** `Router(config)# ip local pool dynpool 30.30.30.20 30.30.30.30`	Specifies a local address pool for the group. For details about this command and additional parameters that can be set, see *Cisco IOS Dial Technologies Command Reference*.

[1] DNS = Domain Name System

[2] WINS = Windows Internet Naming Service

Apply Mode Configuration to the Crypto Map

To apply mode configuration to the crypto map, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto map** *map-name* **isakmp authorization list** *list-name*
2. **crypto map** *tag* **client configuration address** [**initiate** | **respond**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto map** *map-name* **isakmp authorization list** *list-name* **Example:** `Router(config)# crypto map dynmap isakmp authorization list rtr-remote`	Applies mode configuration to the crypto map and enables key lookup (IKE queries) for the group policy from an AAA server.

Step 2	**crypto map** *tag* **client configuration address** [**initiate** | **respond**]
	Example:
	`Router(config)# crypto map dynmap client configuration address respond`
	`#`
		Configures the router to reply to mode configuration requests from remote clients.

Enable Policy Lookup

To enable policy lookup through AAA, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **aaa new-model**
2. **aaa authentication login** {**default** | *list-name*} *method1* [*method2...*]
3. **aaa authorization {network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]]
4. **username** *name* **{nopassword** | **password** *password* | **password** *encryption-type* *encrypted-password*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**aaa new-model** **Example:** `Router(config)# aaa new-model`	Enables the AAA access control model.
Step 2	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Router(config)# aaa authentication login rtr-remote local`	Specifies AAA authentication of selected users at login, and specifies the method used. This example uses a local authentication database. You could also use a RADIUS server for this. For details, see *Cisco IOS Security Configuration Guide: Securing User Services, Release 15M&T* and *Cisco IOS Security Command Reference*.
Step 3	**aaa authorization {network** | **exec** | **commands** *level* | **reverse-access** | **configuration**} {**default** | *list-name*} [*method1* [*method2...*]] **Example:** `Router(config)# aaa authorization network rtr-remote local`	Specifies AAA authorization of all network-related service requests, including PPP, and specifies the method of authorization.

Step 4	**username** *name* **{nopassword** | **password** *password* | **password** *encryption-type* *encrypted-password*}
	Example:
	`Router(config)# username username1 password 0 password1`
		Establishes a username-based authentication system.

Configure IPSec Transforms and Protocols

A transform set represents a certain combination of security protocols and algorithms. During IKE negotiation, the peers agree to use a particular transform set for protecting data flow.

During IKE negotiations, the peers search multiple transform sets for a transform that is the same at both peers. When a transform set is found that contains such a transform, it is selected and applied to the protected traffic as a part of both peers' configurations.

To specify the IPSec transform set and protocols, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto ipsec profile** *profile-name*
2. **crypto ipsec transform-set** *transform-set-name*
3. **crypto ipsec security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}

DETAILED STEPS

	Command or Action	Purpose

Step 1	**crypto ipsec profile** *profile-name*
	Example:
	`Router(config)# crypto ipsec profile pro1`
	`Router(config)#`
		Configures an IPSec profile to apply protection on the tunnel for encryption.

Step 2	**crypto ipsec transform-set** *transform-set-name* *transform1* [*transform2*] [*transform3*] [*transform4*] **Example:** `Router(config)# crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac`	Defines a transform set—an acceptable combination of IPSec security protocols and algorithms. See *Cisco IOS Security Command Reference* for detail about the valid transforms and combinations.

Step 3	**crypto ipsec security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes*}
	Example:
	`Router(config)# crypto ipsec security-association lifetime seconds 86400`
		Specifies global lifetime values used when IPSec security associations are negotiated.

Configure the IPSec Crypto Method and Parameters

A dynamic crypto map policy processes negotiation requests for new security associations from remote IPSec peers, even if the router does not know all the crypto map parameters (for example, IP address).

To configure the IPSec crypto method, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto dynamic-map** *dynamic-map-name dynamic-seq-num*
2. **set transform-set** ***transform-set-name*** **[*transform-set-name2...transform-set-name6*]**
3. **reverse-route**
4. **exit**
5. **crypto map** ***map-name seq-num*** **[ipsec-isakmp] [dynamic *dynamic-map-name*] [discover] [profile *profile-name*]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **Example:** `Router(config)# crypto dynamic-map dynmap 1`	Creates a dynamic crypto map entry and enters crypto map configuration mode. See *Cisco IOS Security Command Reference* for more detail about this command.
Step 2	**set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*] **Example:** `Router(config-crypto-map)# set transform-set vpn1`	Specifies which transform sets can be used with the crypto map entry.
Step 3	**reverse-route** **Example:** `Router(config-crypto-map)# reverse-route`	Creates source proxy information for the crypto map entry.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(config-crypto-map)# exit`	Returns to global configuration mode.
Step 5	**crypto map** *map-name seq-num* [**ipsec-isakmp**] [**dynamic** *dynamic-map-name*] [**discover**] [**profile** *profile-name*] **Example:** `Router(config)# crypto map static-map 1 ipsec-isakmp dynamic dynmap`	Creates a crypto map profile.

Apply the Crypto Map to the Physical Interface

The crypto maps must be applied to each interface through which IPSec traffic flows. Applying the crypto map to the physical interface instructs the router to evaluate all the traffic against the security associations database. With the default configurations, the router provides secure connectivity by encrypting the traffic sent between remote sites. However, the public interface still allows the rest of the traffic to pass and provides connectivity to the Internet.

To apply a crypto map to an interface, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **interface** *type number*
2. **crypto map** *map-name*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/0`	Enters the interface configuration mode for the interface to which you are applying the crypto map.

	Command or Action	Purpose
Step 2	**crypto map** *map-name* **Example:** `Router(config-if)# crypto map static-map`	Applies the crypto map to the interface.
Step 3	**exit** **Example:** `Router(config-crypto-map)# exit`	Returns to global configuration mode.

Where to Go Next

If you are creating a Cisco Easy VPN remote configuration, go to the "Create a Cisco Easy VPN Remote Configuration" section on page 77.

If you are creating a site-to-site VPN using IPSec tunnels and GRE, go to the "Configure a Site-to-Site GRE Tunnel" section on page 80.

Create a Cisco Easy VPN Remote Configuration

The router that is acting as the Cisco Easy VPN client must create a Cisco Easy VPN remote configuration and assign it to the outgoing interface.

To create the remote configuration, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **crypto ipsec client ezvpn** *name*
2. **group** ***group-name*** **key** ***group-key***
3. **peer {*ipaddress* | *hostname*}**
4. **mode {client | network-extension | network extension plus}**
5. **exit**
6. **crypto isakmp keepalive** ***seconds***
7. **interface** ***type number***
8. **crypto ipsec client ezvpn** ***name*** **[outside | inside]**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto ipsec client ezvpn** *name* **Example:** `Router(config)# crypto ipsec client ezvpn ezvpnclient`	Creates a Cisco Easy VPN remote configuration, and enters Cisco Easy VPN remote configuration mode.
Step 2	**group** *group-name* **key** *group-key* **Example:** `Router(config-crypto-ezvpn)# group ezvpnclient key secret-password`	Specifies the IPSec group and IPSec key value for the VPN connection.

Step 3	**peer** {*ipaddress* | *hostname*}
	Example:
	`Router(config-crypto-ezvpn)# peer 192.168.100.1`
		Specifies the peer IP address or hostname for the VPN connection.
		Note A hostname can be specified only when the router has a DNS server available for hostname resolution.
		Note Use this command to configure multiple peers for use as backup. If one peer goes down, the Easy VPN tunnel is established with the second available peer. When the primary peer comes up again, the tunnel is reestablished with the primary peer.

Step 4	**mode** {**client** | **network-extension** | **network extension plus**}
	Example:
	`Router(config-crypto-ezvpn)# mode client`
		Specifies the VPN mode of operation.

Step 5	**exit** **Example:** `Router(config-crypto-ezvpn)# exit`	Returns to global configuration mode.
Step 6	**crypto isakmp keepalive** *seconds* **Example:** `Router(config-crypto-ezvpn)# crypto isakmp keepalive 10`	Enables dead peer detection messages. Time between messages is given in seconds, with a range of 10 to 3600.

	Command or Action	Purpose
Step 7	**interface** *type number* **Example:** `Router(config)# interface Gigabitethernet 0/2`	Enters the interface configuration mode for the interface to which you are applying the Cisco Easy VPN remote configuration.
Step 8	**crypto ipsec client ezvpn** *name* [**outside** | **inside**] **Example:** `Router(config-if)# crypto ipsec client ezvpn ezvpnclient outside`	Assigns the Cisco Easy VPN remote configuration to the WAN interface which causes the router to automatically create the NAT or PAT[1] and the access list configuration needed for the VPN connection.
Step 9	**exit** **Example:** `Router(config-crypto-ezvpn)# exit`	Returns to global configuration mode.

[1] PAT = port address translation

Configuration Example

The following configuration example shows the EasyVPN client configuration.

```
!
aaa new-model
!
aaa authentication login rtr-remote local
aaa authorization network rtr-remote local
aaa session-id common
!
username username1 password 0 password1
!
crypto isakmp policy 1
    encryption 3des
    authentication pre-share
    group 2
    lifetime 480
!
crypto isakmp client configuration group rtr-remote
    key secret-password
    dns 10.50.10.1 10.60.10.1
    domain company.com
    pool dynpool
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto ipsec security-association lifetime seconds 86400
!
crypto dynamic-map dynmap 1
    set transform-set vpn1
    reverse-route
!
crypto map static-map 1 ipsec-isakmp dynamic dynmap
crypto map dynmap isakmp authorization list rtr-remote
crypto map dynmap client configuration address respond
crypto ipsec client ezvpn ezvpnclient
    connect auto
    group 2 key secret-password
    mode client
    peer 192.168.100.1
!

interface gigabitethernet 0/4
    crypto ipsec client ezvpn ezvpnclient outside
    crypto map static-map

interface vlan 1
    crypto ipsec client ezvpn ezvpnclient inside
!
```

Configure a Site-to-Site GRE Tunnel

To configure a site-to-site GRE tunnel, follow these steps, beginning in global configuration mode.

SUMMARY STEPS

1. **interface** *type number*
2. **ip address** *ip-address mask*
3. **tunnel source** *interface-type number*
4. **tunnel destination** *default-gateway-ip-address*
5. **crypto map** *map-name*
6. **exit**
7. **ip access-list** {**standard** | **extended**} *access-list-name*
8. **permit** *protocol source source-wildcard destination destination-wildcard*
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**interface** *type number* **Example:** `Router(config)# interface tunnel 1`	Creates a tunnel interface and enters interface configuration mode.
Step 2	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.62.1.193 255.255.255.252`	Assigns an address to the tunnel.

	Command or Action	Purpose
Step 3	**tunnel source** *interface-type number* **Example:** `Router(config-if)# tunnel source gigabitethernet 0/0`	Specifies the source endpoint of the router for the GRE tunnel.
Step 4	**tunnel destination** *default-gateway-ip-address* **Example:** `Router(config-if)# tunnel destination 192.168.101.1`	Specifies the destination endpoint of the router for the GRE tunnel.
Step 5	**crypto map** *map-name* **Example:** `Router(config-if)# crypto map static-map`	Assigns a crypto map to the tunnel. **Note** Dynamic routing or static routes to the tunnel interface must be configured to establish connectivity between the sites..
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**ip access-list** {**standard** | **extended**} *access-list-name* **Example:** `Router(config)# ip access-list extended vpnstatic1`	Enters ACL[1] configuration mode for the named ACL that the crypto map uses.
Step 8	**permit** *protocol source source-wildcard destination destination-wildcard* **Example:** `Router(config-acl)# permit gre host 192.168.100.1 host 192.168.101.1`	Specifies that only GRE traffic is permitted on the outbound interface.
Step 9	**exit** **Example:** `Router(config-acl)# exit`	Returns to global configuration mode.

[1] ACL = access control list

Configuration Example

The following configuration example shows a portion of the configuration file for a site-to-site VPN using a GRE tunnel as described in the preceding sections.

```
!
aaa new-model
!
aaa authentication login rtr-remote local
aaa authorization network rtr-remote local
aaa session-id common
!
username username1 password 0 password1
!
interface tunnel 1
    ip address 10.62.1.193 255.255.255.252

tunnel source GigabitEthernet 0/3

tunnel destination interface 192.168.101.1

ip route 20.20.20.0 255.255.255.0 tunnel 1

crypto isakmp policy 1
    encryption 3des
    authentication pre-share
    group 2
!
crypto isakmp client configuration group rtr-remote
    key secret-password
    dns 10.50.10.1 10.60.10.1
    domain company.com
    pool dynpool
!
crypto ipsec transform-set vpn1 esp-3des esp-sha-hmac
!
crypto ipsec security-association lifetime seconds 86400
!
crypto dynamic-map dynmap 1
    set transform-set vpn1
    reverse-route
!
crypto map static-map 1 ipsec-isakmp dynamic dynmap
crypto map dynmap isakmp authorization list rtr-remote
crypto map dynmap client configuration address respond
!
! Defines the key association and authentication for IPsec tunnel.
crypto isakmp policy 1
hash md5
authentication pre-share
crypto isakmp key cisco123 address 200.1.1.1
!
!
! Defines encryption and transform set for the IPsec tunnel.
crypto ipsec transform-set set1 esp-3des esp-md5-hmac
!
! Associates all crypto values and peering address for the IPsec tunnel.
crypto map to_corporate 1 ipsec-isakmp
 set peer 200.1.1.1
 set transform-set set1
 match address 105
!
!
! VLAN 1 is the internal home network.
interface vlan 1
 ip address 10.1.1.1 255.255.255.0
 ip nat inside
 ip inspect firewall in ! Inspection examines outbound traffic.
    crypto map static-map
    no cdp enable
!
! GE4 is the outside or Internet-exposed interface
interface Gigabitethernet 0/4
 ip address 210.110.101.21 255.255.255.0
 ! acl 103 permits IPsec traffic from the corp. router as well as
 ! denies Internet-initiated traffic inbound.
 ip access-group 103 in
 ip nat outside
 no cdp enable
 crypto map to_corporate ! Applies the IPsec tunnel to the outside interface.
!
! Utilize NAT overload in order to make best use of the
! single address provided by the ISP.
ip nat inside source list 102 interface Gigabitethernet 0/1 overload
ip classless
ip route 0.0.0.0 0.0.0.0 210.110.101.1
no ip http server
!
!
! acl 102 associated addresses used for NAT.
access-list 102 permit ip 10.1.1.0 0.0.0.255 any
! acl 103 defines traffic allowed from the peer for the IPsec tunnel.
access-list 103 permit udp host 200.1.1.1 any eq isakmp
access-list 103 permit udp host 200.1.1.1 eq isakmp any
access-list 103 permit esp host 200.1.1.1 any
! Allow ICMP for debugging but should be disabled because of security implications.
access-list 103 permit icmp any any
access-list 103 deny ip any any ! Prevents Internet-initiated traffic inbound.
! acl 105 matches addresses for the IPsec tunnel to or from the corporate network.
access-list 105 permit ip 10.1.1.0 0.0.0.255 192.168.0.0 0.0.255.255
no cdp run
```

Configuring Dynamic Multipoint VPN

The Dynamic Multipoint VPN (DMVPN) feature is a simplified solution to deploy large and small IP Security (IPsec) VPNs by combining GRE tunnels, IPsec encryption, and Next Hop Resolution Protocol (NHRP). DMVPN simplifies the configuration tasks in a large scale VPN deployment and reduces the administrative overhead.

DMVPN is useful in a scenario, when one central router at the head office acts as a hub and other branch routers act as spoke and connected to the hub router to access the company's resources. DMVPN is also useful for spoke to spoke deployment and can be used for branch-to-branch interconnections

See the Example: DMVPN Configuration, page 83 for a typical DMVPN configuration for a hub and spoke deployment. For additional information about configuring DMVPN, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_dmvpn/configuration/15-mt/sec-conn-dmvpn-15-mt-book/sec-conn-dmvpn-dmvpn.html

Example: DMVPN Configuration

The following configuration example shows the configuration for DMVPN hub and spoke deployment model. In this example, Cisco 800M series ISR is configured as spoke and Cisco 2900 Series ISR is configured as hub. For readability some part of the configuration is removed.

This configuration section shows the configuration of 800M Series ISR as a spoke.

800M_spoke# **show running-config**

```
Building configuration...
Current configuration : 2546 bytes
!
! Last configuration change at 09:09:39 UTC Tue Jun 24 2014
!
version 15.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 800M_spoke
!
boot-start-marker
boot-end-marker
!
!
logging buffered 10000000
!
no aaa new-model
!
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
crypto isakmp policy 1
 encr aes
 hash sha256
 authentication pre-share
 group 2
crypto isakmp key ISA_KEY address 0.0.0.0
crypto isakmp keepalive 10 periodic
!

crypto ipsec transform-set DMVPN-TRANS-SET esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile DMVPN-PROFILE
 set security-association lifetime seconds 120
 set transform-set DMVPN-TRANS-SET
!

interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface Tunnel0
 ip address 24.1.1.2 255.255.255.0
 no ip redirects
 ip mtu 1440
 ip nhrp authentication ISA_KEY
 ip nhrp map multicast 172.16.0.1
 ip nhrp map 24.1.1.1 172.16.0.1
 ip nhrp network-id 1
 ip nhrp holdtime 120
 ip nhrp nhs 24.1.1.1
 ip nhrp registration timeout 30
 ip nhrp shortcut
 tunnel source GigabitEthernet0/9
 tunnel mode gre multipoint
 tunnel key 0
 tunnel protection ipsec profile DMVPN-PROFILE
!
```

```
interface GigabitEthernet0/0
 no ip address
!
interface GigabitEthernet0/1
 no ip address
!
interface GigabitEthernet0/2
 no ip address
!
interface GigabitEthernet0/3
 no ip address
!
interface GigabitEthernet0/4
 no ip address
!
interface GigabitEthernet0/5
 no ip address
!
interface GigabitEthernet0/6
 no ip address
!
interface GigabitEthernet0/7
 no ip address
!
interface GigabitEthernet0/8
 ip address 192.168.3.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/9
 ip address 172.15.0.1 255.255.255.0
 duplex auto
 speed auto
!
interface Vlan1
 ip address 190.160.10.111 255.255.255.0
!
!
router eigrp 20
 network 2.2.2.0 0.0.0.255
 network 24.1.1.0 0.0.0.255
!
!
router eigrp 10
 network 172.15.0.0 0.0.0.255
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 192.168.4.0 255.255.255.0 100.100.100.2
ip route 192.168.5.0 255.255.255.0 100.100.100.2
!
access-list 101 permit ip 192.168.3.0 0.0.0.255 192.168.4.0 0.0.0.255
access-list 102 permit ip 100.100.100.0 0.0.0.255 200.200.200.0 0.0.0.255
!
control-plane
!
!
line con 0
 no modem enable
line vty 0 4
 login
 transport input none
!
scheduler allocate 20000 1000
!
end
```

This configuration section shows the configuraton of 2900 Series ISR as hub.

2901_hub# **show running-config**

```
Building configuration...

Current configuration : 3210 bytes
!
! Last configuration change at 07:34:35 UTC Tue Jun 24 2014
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 2901_hub
!
boot-start-marker
boot-end-marker
!
!
logging buffered 10000000
!
no aaa new-model
!
ip cef
!
!
no ipv6 cef
!
multilink bundle-name authenticated
!
license udi pid CISCO2901/K9 sn FGL180322RF
license boot module c2900 technology-package securityk9
!
!
!
redundancy
!

lldp run
!
!
crypto isakmp policy 1
 encr aes
 hash sha256
 authentication pre-share
 group 2
crypto isakmp key ISA_KEY address 0.0.0.0
crypto isakmp keepalive 10 periodic
!
!
crypto ipsec transform-set DMVPN-TRANS-SET esp-aes 256 esp-sha-hmac
 mode tunnel
!
crypto ipsec profile DMVPN-PROFILE
 set security-association lifetime seconds 120
 set transform-set DMVPN-TRANS-SET
!
!

interface Loopback0
 ip address 1.1.1.1 255.255.255.255
 ip ospf message-digest-key 1 md5 cisco
!
```

```
interface Loopback1
 ip address 12.12.12.2 255.255.255.255
!
interface Loopback2
 ip address 12.12.12.3 255.255.255.255
!
interface Loopback3
 ip address 12.12.12.4 255.255.255.255
!
interface Loopback4
 ip address 12.12.12.5 255.255.255.255
!
interface Tunnel0
 ip address 24.1.1.1 255.255.255.0
 no ip redirects
 ip mtu 1440
 no ip split-horizon eigrp 10
 ip nhrp authentication ISA_KEY
 ip nhrp map multicast dynamic
 ip nhrp network-id 1
 ip nhrp shortcut
 ip nhrp redirect
 ip summary-address eigrp 20 192.168.0.0 255.255.0.0
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 0
 tunnel protection ipsec profile DMVPN-PROFILE
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 192.168.5.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 172.16.0.1 255.255.255.0
 ip ospf message-digest-key 1 md5 cisco
 ip ospf priority 10
 duplex auto
 speed auto
!
interface GigabitEthernet0/1/0
 switchport access vlan 2
 no ip address
 shutdown
!
interface GigabitEthernet0/1/1
 switchport access vlan 10
 no ip address
!
interface GigabitEthernet0/1/2
 switchport access vlan 10
 no ip address
!
interface GigabitEthernet0/1/3
 switchport access vlan 20
 no ip address
!
interface GigabitEthernet0/1/4
 no ip address
!
```

```
interface GigabitEthernet0/1/5
 switchport access vlan 10
 no ip address
!
interface GigabitEthernet0/1/6
 no ip address
!
interface GigabitEthernet0/1/7
 no ip address
!
interface Vlan1
 no ip address
!
!
router eigrp 10
 network 172.16.0.0 0.0.0.255
!
!
router eigrp 20
 network 1.1.1.0 0.0.0.255
 network 24.1.1.0 0.0.0.255
 network 192.168.5.0
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 100.100.100.0 255.255.255.0 150.150.150.2
ip route 192.168.3.0 255.255.255.0 150.150.150.2
ip route 192.168.4.0 255.255.255.0 150.150.150.2
ip route 200.200.200.0 255.255.255.0 150.150.150.2
!
!

control-plane
!

line con 0
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Configuring Group Encrypted Transport VPN

Group Encrypted Transport VPN (GETVPN) is a tunnel-less VPN technology that provides end-to-end security for network traffic in a native mode and maintain the mesh topology. GET VPN combines the keying protocol Group Domain of Interpretation (GDOI) with IPsec encryption to provide users with an efficient method of securing IP multicast traffic or unicast traffic. GET VPN enables the router to apply encryption to tunnel-less (native) IP multicast and unicast packets and eliminates the requirement to configure tunnels to protect multicast and unicast traffic.

By removing the need for point-to-point tunnels, meshed networks can scale higher while maintaining network-intelligence features that are critical to voice and video quality, such as QoS, routing, and multicast. GET VPN offers a new standards-based IP security (IPsec) security model that is based on the concept of “trusted” group members. Trusted member routers use a common security methodology that is independent of any point-to-point IPsec tunnel relationship.

A GETVPN deployment has primarily three components, Key Server (KS), Group Member (GM), and Group Domain of Interpretation (GDOI) protocol. GMs encrypt or decrypt the traffic and KS distributes the encryption key to all the group members. The KS decides on one single data encryption key for a given life time. Since all GMs use the same key, any GM can decrypt the traffic encrypted by any other GM. GDOI protocol is used between the GM and KS for group key and group SA management. Minimum one KS is required for a GETVPN deployment.

Unlike traditional IPSec encryption solutions, GET VPN uses the concept of group security association (SA). All members in the GETVPN group can communicate with each other using a common encryption policy and a shared SA and therefore no need to negotiate IPSec between GMs on a peer to peer basis; thereby reducing the resource load on the GM routers.

See the Example: GETVPN Configuration, page 90 for a sample GETVPN deployment configuration.

For additional information about configuring GET VPN, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_getvpn/configuration/15-mt/sec-get-vpn-15-mt-book/sec-get-vpn.html

Example: GETVPN Configuration

The following configuration example shows the configuration for GETVPN deployment. In this example, a Cisco 800M series ISR is configured as GM and the Cisco 1900 Series ISR is configured as KS.

This configuration section shows the configuration of 800M Series ISR as GM.

800M_GM# **show running-config**

```
Building configuration...

Current configuration : 1752 bytes
!
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 800M_GM
!
boot-start-marker
boot-end-marker
!
```

```
no aaa new-model
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2
!
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
cts logging verbose
license udi pid C841M-8X/K9 sn FOC18170PNJ
license accept end user agreement
license boot module c800m level advipservices
!
redundancy
!
crypto isakmp policy 100
 encr aes
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key cisco address 192.168.1.2
!
crypto gdoi group gdoi
 identity number 1234
 server address ipv4 192.168.1.2

!
crypto map crypto 10 gdoi
 set group gdoi
!
interface GigabitEthernet0/0
 no ip address
!
interface GigabitEthernet0/1
 no ip address
!
interface GigabitEthernet0/2
 no ip address
!
interface GigabitEthernet0/3
 no ip address
!
interface GigabitEthernet0/4
 no ip address
!
interface GigabitEthernet0/5
 no ip address
!
interface GigabitEthernet0/6
 no ip address
!
interface GigabitEthernet0/7
 no ip address
!
interface GigabitEthernet0/8
 ip address 10.1.3.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/9
 ip address 192.168.3.2 255.255.255.0
 duplex auto
 speed auto
 crypto map crypto
!
interface Vlan1
 no ip address
!
!
router eigrp 1
 network 10.1.3.0 0.0.0.255
 network 192.168.3.0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
control-plane
!
line con 0
 no modem enable
line vty 0 4
 login
 transport input none
!
scheduler allocate 20000 1000
!
end
```

This configuration section shows the configuration of Cisco 1900 Series ISR as KS.

1921_KS# **show running-config**

```
Building configuration...
Current configuration : 2019 bytes
!
version 15.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 1921_KS
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
!

!
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!

license udi pid CISCO1921/K9 sn FGL155022DY
license boot module c1900 technology-package securityk9
license boot module c1900 technology-package datak9
!
!
!
```

```
redundancy
!
crypto isakmp policy 100
 encr aes
 authentication pre-share
 group 5
 lifetime 3600
crypto isakmp key cisco address 0.0.0.0
!

crypto ipsec transform-set trans esp-aes esp-sha-hmac
 mode tunnel
!
!
crypto ipsec profile ipsec
 set transform-set trans
!
crypto gdoi group gdoi
 identity number 1234
 server local
  rekey algorithm aes 256
  rekey lifetime seconds 3600
  rekey authentication mypubkey rsa vpnkeys
  rekey transport unicast
  sa ipsec 10
   profile ipsec
   match address ipv4 getvpn
   replay counter window-size 64
   no tag
  address ipv4 192.168.1.2
!
!
crypto map crypto 10 gdoi
 set group gdoi
!

interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 192.168.1.2 255.255.255.0
 duplex auto
 speed auto
 crypto map crypto
!
interface Serial0/0/0
 no ip address
 shutdown
!
interface Serial0/0/1
 no ip address
 shutdown
 clock rate 2000000
!

router eigrp 1
 network 192.168.1.0
```

```
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!

ip access-list extended getvpn
 permit ip 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
!

control-plane
!

line con 0
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport output lat pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Configuring SSL VPN

The Secure Socket Layer Virtual Private Network (SSL VPN) feature provides support for remote user access to enterprise networks from anywhere on the Internet. Remote access is provided through a SSL–enabled SSL VPN gateway. The SSL VPN gateway allows remote users to establish a secure VPN tunnel using a web browser. This feature provides a comprehensive solution that allows easy access to a broad range of web resources and web-enabled applications using native HTTP over SSL (HTTPS) browser support. SSL VPN delivers three modes of SSL VPN access: clientless, thin-client, and full-tunnel client support.

See the "Example: SSL VPN Configuration" section for a sample SSL VPN gateway configuration.

For additional information about configuring SSL VPN, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_sslvpn/configuration/15-mt/sec-conn-sslvpn-15-mt-book/sec-conn-sslvpn-ssl-vpn.html

Example: SSL VPN Configuration

This configuration example shows the configuration for SSL VPN gateway using Cisco 800M Series ISR.

800M# **show running-config**

```
Building configuration...

Current configuration : 4053 bytes
!
version 15.5
```

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 800M
!
boot-start-marker
boot-end-marker
!
!
!
aaa new-model
!
!
aaa authentication login default local
aaa authentication login ciscocp_vpn_xauth_ml_1 local
!
!

aaa session-id commont
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2
!
crypto pki trustpoint TP-self-signed-2716339910
 enrollment selfsigned
 subject-name cn=IOS-Self-Signed-Certificate-2716339910
 revocation-check none
 rsakeypair TP-self-signed-2716339910
!
!
crypto pki certificate chain TP-self-signed-2716339910
 certificate self-signed 01
  3082022B 30820194 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 32373136 33333939 3130301E 170D3134 31313132 31313430
  35355A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
  4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 37313633
  33393931 3030819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
  8100A775 D34D41D6 281317C5 427BBC6D 3D97F5B4 F91E924B AB23F5CC F92336E6
  29EBDC57 45A455B7 D7300C0C 07C5DDF8 62E2BDFB CDEB57CC EFAE7006 A72D4C20
  2D9995E7 472D2C4E 079828B3 B63DDB66 A9D3D77F BC844CBD 255D81F0 84564748
  4FAD69E1 94F5AFC9 0450EFDC 9096BD38 3F4FA022 0680E969 174197EA 3F85DD4C
  B1490203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF 301F0603
  551D2304 18301680 145602C5 80924574 A895C527 F177A81B 4EA03C94 EA301D06
  03551D0E 04160414 5602C580 924574A8 95C527F1 77A81B4E A03C94EA 300D0609
  2A864886 F70D0101 05050003 81810090 823846F0 FAA084FB F5C17F04 00E11E54
  D9D9B32A 4EBB96D4 8414C5DD 0DB8728B 84518031 0B22A20A 989C341C 4AB15B7B
  B192E99B E29138E9 56263016 5565DEAA 9CE9E40B D945EF2C 1BFE110C 4622F707
  39E7FA48 DA3B15DD CA66AA8F 61783562 7C09932F BD4E5AB4 A1242A71 90E27B22
  71CD3A0D A0004521 D1DB1E2C D95BEF
        quit
!

ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
cts logging verbose
license udi pid C841M-8X/K9 sn FCW1842005Y
!
!
username cisco privilege 15 password 0 cisco
!
```

```
redundancy
!
crypto vpn anyconnect sdflash:/webvpn/anyconnect-win-3.1.03103-k9.pkg sequence 1


!
interface Loopback10
 ip address 100.100.100.100 255.255.255.255
!
interface GigabitEthernet0/0
 no ip address
!
interface GigabitEthernet0/1
 no ip address
!
interface GigabitEthernet0/2
 no ip address
!
interface GigabitEthernet0/3
 no ip address
!
interface GigabitEthernet0/4
 no ip address
!
interface GigabitEthernet0/5
 no ip address
!
interface GigabitEthernet0/6
 no ip address
!
interface GigabitEthernet0/7
 no ip address
!
interface GigabitEthernet0/8
 ip address 192.168.10.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/9
 ip address 9.43.17.81 255.255.0.0
 duplex auto
 speed auto
!
interface Virtual-Template1
 ip unnumbered GigabitEthernet0/8
 ip virtual-reassembly in
!
interface Vlan1
 no ip address
!
ip local pool IP_Pool 10.10.10.1 10.10.10.10
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
ip route 202.153.144.0 255.255.255.0 9.43.0.1
!

control-plane
!

line con 0
 no modem enable
```

```
line vty 0 4
 transport input none
!
scheduler allocate 20000 1000
!

webvpn gateway gateway_1
 ip address 192.168.10.1 port 443
 ssl trustpoint TP-self-signed-2716339910
 inservice
 !
webvpn context Test
 secondary-color white
 title-color #FF9900
 text-color black
 virtual-template 1
 aaa authentication list ciscocp_vpn_xauth_ml_1
 gateway gateway_1
 !
 ssl authenticate verify all
 inservice
 !
 policy group policy_1
   functions svc-enabled
   svc address-pool "IP_Pool" netmask 255.255.255.255
   svc default-domain "cisco.com"
   svc keep-client-installed
   svc rekey time 240
   svc dns-server primary 10.105.130.1
   svc wins-server primary 10.105.130.1
 default-group-policy policy_1
!
end
```

Configuring FlexVPN

FlexVPN is Cisco's implementation of the IKEv2 standard featuring a unified paradigm and CLI that combines site to site, remote access, hub and spoke topologies and partial meshes (spoke to spoke direct). FlexVPN offers a simple but modular framework that extensively uses the tunnel interface paradigm while remaining compatible with legacy VPN implementations using crypto maps.

See the "Example: FlexVPN Configuration" section for a sample FlexVPN hub and spoke configuration.

For additional information about configuring FlexVPN, see the following link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_ike2vpn/configuration/15-mt/sec-flex-vpn-15-mt-book/sec-intro-ikev2-flex.html

Example: FlexVPN Configuration

The following configuration example shows the configuration for FlexVPN hub and spoke deployment model. In this example, Cisco 800M series ISR is configured as a spoke and Cisco 3900 Series ISR is configured as the hub.

This configuration section shows the configuration of 800M Series ISR as a spoke.

800M# **show running-config**

```
Building configuration...
```

```
Current configuration : 2461 bytes
!
!
version 15.5
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 800M
!
boot-start-marker
boot-end-marker
!

aaa new-model
!
!
aaa authorization network FLEX local
!

aaa session-id common
bsd-client server url https://cloudsso.cisco.com/as/token.oauth2

!
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!
chat-script multimode "" "AT!CALL" TIMEOUT 20 "OK"
cts logging verbose
license udi pid C841M-4X/K9 sn FCW1839001E
!

redundancy
!
crypto ikev2 authorization policy FLEX
 route set interface
!
!
!
crypto ikev2 keyring KEYRING
 peer R1
  address 172.16.0.1
  pre-shared-key CISCO
 !
!
!
crypto ikev2 profile default
 match identity remote address 172.16.0.1 255.255.255.255
 identity local key-id FLEX
 authentication remote pre-share
 authentication local pre-share
 keyring local KEYRING
 aaa authorization group psk list FLEX FLEX
!
!
!
controller Cellular 0/0
 modem link-recovery rssi onset-threshold -110
 modem link-recovery monitor-timer 20
 modem link-recovery wait-timer 10
 modem link-recovery debounce-count 6

!
interface Loopback0
 ip address 2.2.2.2 255.255.255.0
!
interface Tunnel0
 ip address negotiated
 tunnel source GigabitEthernet0/5
 tunnel mode ipsec ipv4
 tunnel destination 172.16.0.1
 tunnel protection ipsec profile default
!
interface GigabitEthernet0/0
 no ip address
!
interface GigabitEthernet0/1
 no ip address
!
interface GigabitEthernet0/2
 no ip address
!
interface GigabitEthernet0/3
 no ip address
!
interface GigabitEthernet0/4
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/5
 ip address 172.16.0.2 255.255.255.0
 duplex auto
 speed auto
!
interface Cellular0/0/0
 no ip address
 encapsulation slip
 dialer in-band
 dialer string multimode
!
interface Serial0/1/0
 no ip address
 shutdown
 clock rate 2000000
!
interface Vlan1
 no ip address
!
!
router eigrp 1
 network 0.0.0.0
 passive-interface default
 no passive-interface Tunnel0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!

control-plane
!
```

```
line con 0
 no modem enable
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 stopbits 1
line 3
 script dialer multimode
 no exec
line vty 0 4
 transport input none
!
scheduler allocate 20000 1000
!
end
```

This configuration section shows the configuration of 800M Series ISR as a spoke.

C3900# **show running-config**

```
Building configuration...

Current configuration : 2690 bytes
!
! Last configuration change at 13:10:19 UTC Fri Oct 31 2014
version 15.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname C3900
!
boot-start-marker
boot-end-marker
!
aqm-register-fnf
!
!
aaa new-model
!
!
aaa authorization network LOCALIKEv2 local

!
!
aaa session-id common

!
!
!
ip cef
no ipv6 cef
!
!
multilink bundle-name authenticated
!

!
voice-card 0
!

!
```

```
license udi pid C3900-SPE200/K9 sn FOC16075NAN
license accept end user agreement
license boot module c3900e technology-package securityk9
license boot module c3900e technology-package datak9
!
!
!
redundancy
!
crypto ikev2 authorization policy AUTHOR-POLICY
 pool POOL
!
!
!
crypto ikev2 keyring KEYRING
 peer R2
  address 172.16.0.2
  pre-shared-key CISCO
 !
!
!
crypto ikev2 profile default
 match identity remote key-id FLEX
 authentication remote pre-share
 authentication local pre-share
 keyring local KEYRING
 aaa authorization group psk list LOCALIKEv2 AUTHOR-POLICY
 virtual-template 1

!
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.0
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 172.16.0.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet0/3
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1/0
 no ip address
!
interface FastEthernet0/1/1
 no ip address
!
interface FastEthernet0/1/2
 no ip address
```

```
!
interface FastEthernet0/1/3
 no ip address
!
interface FastEthernet0/1/4
 no ip address
!
interface FastEthernet0/1/5
 no ip address
!
interface FastEthernet0/1/6
 no ip address
!
interface FastEthernet0/1/7
 no ip address
!
interface FastEthernet0/1/8
 no ip address
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
 tunnel source GigabitEthernet0/1
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile default
!
interface Vlan1
 no ip address
!
!
!
router eigrp 1
 network 1.1.1.1 0.0.0.0
 passive-interface default
 no passive-interface Virtual-Template1
!
ip local pool POOL 192.168.0.1 192.168.0.10
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
!
!
nls resp-timeout 1
cpd cr-id 1
!
!
!
!
control-plane
!
 !

mgcp behavior rsip-range tgcp-only
mgcp behavior comedia-role none
mgcp behavior comedia-check-media-src disable
mgcp behavior comedia-sdp-force disable
!
mgcp profile default
!

gatekeeper
 shutdown
!
```

```
!
!
line con 0
line aux 0
line vty 0 4
 transport input all
!
scheduler allocate 20000 1000
!
end
```

Configuring Zone-Based Policy Firewall

Zone-Based Policy Firewall (also known as Zone-Policy Firewall, or ZFW) changes the firewall configuration from the interface-based model to a more flexible, more easily understood zone-based model. Interfaces are assigned to zones, and inspection policy is applied to traffic moving between the zones. Inter-zone policies offer considerable flexibility and granularity, so different inspection policies can be applied to multiple host groups connected to the same router interface.

For more information about configuring zone-based policy firewall, see the following weblink:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_zbf/configuration/15-mt/sec-data-zbf-15-mt-book/sec-zone-pol-fw.html

Configuring VRF-Aware Cisco Firewall

VRF-Aware Cisco Firewall applies Cisco Firewall functionality to Virtual Routing and Forwarding (VRF) interfaces when the firewall is configured on a service provider (SP) or large enterprise edge device. SPs can provide managed services to small and medium business markets.

For more information about configuring VRF-aware Cisco Firewall, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_zbf/configuration/15-mt/sec-data-zbf-15-mt-book/sec-cbac-vrf-fw.html

Configuring Subscription-Based Cisco IOS Content Filtering

The Subscription-based Cisco IOS Content Filtering feature interacts with the Trend Micro URL filtering service so that HTTP requests can be allowed or blocked, and logged, based on a content filtering policy. The content filtering policy specifies how to handle items such as web categories, reputations (or security ratings), trusted domains, untrusted domains, and keywords. URLs are cached on the router, so that subsequent requests for the same URL do not require a lookup request, thus improving performance.

For more information about configuring subscription-based Cisco IOS content filtering see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_zbf/configuration/15-mt/sec-data-zbf-15-mt-book/subscrip-cont-filter.html

Configuring On-Device Management for Security Features

The On-Device Management for Security Features provides an intuitive and simple management interface, the Cisco Configuration Professional Express, to deploy a variety of security features. You can deploy security features including zone-based firewalls, VPN, Intrusion Detection System (IDS) and URL filtering through the Cisco Configuration Professional Express.

The Cisco Configuration Professional Express uses existing zone-based firewall CLIs in conjunction with Network-Based Application Recognition 2 (NBAR2) CLIs to determine the application category, and position NBAR2 protocols supported by the firewall into the relevant application category.

Fro more information about enabling NBAR2 for zone-based firewalls, see the following web link:

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_data_zbf/configuration/15-mt/sec-data-zbf-15-mt-book/on-device-mgmt.html

Related Documents

Topic	Document Title
DMVPN	Dynamic Multipoint VPN Configuration Guide, Cisco IOS Release 15M&T
GETVPN	Cisco Group Encrypted Transport VPN Configuration Guide, Cisco IOS Release 15M&T
SSL VPN	SSL VPN Configuration Guide, Cisco IOS Release 15M&T
FlexVPN	FlexVPN and Internet Key Exchange Version 2 Configuration Guide, Cisco IOS Release 15M&T
IKE for IPSec VPNs	Internet Key Exchange for IPsec VPNs Configuration Guide, Cisco IOS Release 15M&T