CCSP SNRS
Quick Reference Sheets

Brandon James Carroll

ciscopress.com

shortcut

Your Short Cut to Knowledge

About the Author

Brandon James Carroll is one of the country's leading instructors for Cisco security technologies, teaching classes that include the CCNA, CCNP, CCSP courses, a number of the CCVP courses, as well as custom developed courseware. In his six years with Ascolta, Brandon has developed and taught many private Cisco courses for companies such as Boeing, Intel, and Cisco themselves. He is a CCNA, CCNP, CCSP, and a Certified Cisco Systems Instructor (CCSI). Brandon is the author of *Cisco Access Control Security*.

Prior to becoming a technical instructor for Ascolta, Mr. Carroll was a technician and an ADSL specialist for GTE Network Services and Verizon Communications. His duties involved ISP router support and network design. As a lead engineer, he tested and maintained Frame Relay connections between Lucent B-STDX and Cisco routers. His team was in charge of troubleshooting ISP Frame Relay to ATM cut-overs for ADSL customers. Brandon trained new employees at Verizon to the EPG in ADSL testing and troubleshooting procedures, and managed a "Tekwizard" database for technical information and troubleshooting techniques. Mr. Carroll majored in Information Technology at St. Leo University.

About the Technical Reviewer

Ronald Trunk, CCIE, CISSP, is a highly experienced consultant and network architect with a special interest in secure network design and implementation. He has designed complex multimedia networks for both government and commercial clients. He is the author of several articles on network security and troubleshooting. Ron lives in suburban Washington DC.

CHAPTER 1

Layer 2 Security

Examining Layer 2 Attacks

Security is a topic on every network administrator's mind, regardless of whether it's even part of his or her job. And to protect networks, people deploy a variety of devices, including firewalls and intrusion prevention systems. Although these types of devices need to be present, they don't protect a certain area of the network that is often left vulnerable to attack: Layer 2. That's right; the access layer is often forgotten. This leaves your network open to myriad simple-to-run attacks that can wreak havoc on a network.

Those preparing for the CCSP-SNRS certification exam must understand Layer 2 attacks and their mitigation techniques. An understanding of these concepts and mitigation techniques will not only help you pass the test, it will also assist you in securing your production networks.

Types of Layer 2 Attacks

Switches are susceptible to many of the same Layer 3 attacks as routers, but switches are vulnerable to Layer 2 attacks, too, including the following:

- Content-addressable memory (CAM) table overflow
- VLAN hopping
- Spanning-tree manipulation

- MAC spoofing
- Private VLAN (PVLAN) attacks
- DHCP attacks

CAM Table Overflow Attack

This attack involves an attacker who floods the switch with bogus MAC addresses. The MAC table learns the bogus addresses, and thus those bogus addresses fill up the MAC table, leaving no room to learn real MAC addresses. Because the switch cannot now learn real MAC addresses, when a host sends traffic to another device, the switch must flood the traffic to all ports except the one it was heard on. This, in effect, enables the attacker to get a copy of the frame. This type of attack can be done by anyone running Knoppix STD (Security Tools Distribution), using an application called macof. To mitigate this type of attack, implement port security.

Port Security

With the port security feature, you can restrict input to an interface by identifying and *limiting the number of MAC addresses* that are allowed to be learned (and for that matter, even gain network access on a particular port). Port security enables you to specify MAC addresses for each port or to permit a limited number of MAC addresses that are not statically defined. When a secure port receives a packet, the source MAC address of the packet is compared to the list of secure source addresses that were manually configured or autoconfigured (learned) on the port. If a MAC address of a device attached to the port differs from the list of secure addresses, the port either shuts down permanently (default mode) or drops incoming packets from the insecure host.

NOTE

Cisco recommends that you configure the port security feature to issue a shutdown instead of dropping packets from insecure hosts through the restrict option. The restrict option may fail under the load of an attack, and the port will be disabled anyway.

Default Port Security Configuration

The default port security interface configuration settings are as follows:

- Ports security is disabled.
- Maximum MAC addresses setting is 1.
- Violation mode is shutdown.
- Sticky address learning is disabled.
- Port security aging is disabled. Aging time is 0, and the default type is absolute.

Port Security Configuration Guidelines

The following guidelines are only a few of the port security guidelines that you should be aware of. Some implications with port security and VoIP configurations are not covered here.

- Port security can be configured only on static access ports.
- A secure port cannot be a dynamic access port or a trunk port. This means that you must indicate to the switch whether the port is in switchport mode access or switchport mode trunk.
- A secure port cannot be a destination port for Switched Port Analyzer (SPAN).
- A secure port cannot belong to a Fast EtherChannel or Gigabit EtherChannel port group.
- You cannot configure port security on a per-VLAN basis.

NOTE
You can find a more detailed discussion of port security at the following site:

http://www.cisco.com/en/US/docs/switches/lan/catalyst2960/software/release/12.2_25_see/configuration/guide/swtrafc.html#wp1038501

Enabling and Configuring Port Security

To configure port security, issue the following interface commands on the port that you want port security enabled on:

```
switchport mode access

switchport port-security

switchport port-security maximum value

switchport port-security violation {protect ¦ restrict ¦ shutdown}

switchport port-security mac-address mac-address

switchport port-security mac-address sticky
```

The following configuration enables port security on Fast Ethernet 0/2, allowing a maximum of two devices on the interface. Both MAC addresses will be dynamically learned and statically added using the **sticky** command:

```
Switch#config t
Switch(config)#interface f0/2
```

The port *must* be an access port to enable port security. The following configuration command accomplishes this:

```
Switch(config-if)#switchport mode access
```

The next command enables port security:

```
Switch(config-if)#switchport port-security
```

The next command sets the maximum number of MAC addresses to be learned at two. This would work in a *non-VoIP* implementation. For VoIP, you need this value to be set to three:

```
Switch(config-if)#switchport port-security maximum 2
```

The next command enables the sticky learning of the first two MAC addresses, based on the **switchport port-security maximum** command. Sticky learning means the MAC address can either be statically or dynamically learned, but when they are and the configuration is saved, if the switch reboots it will not need to learn the MAC addresses again:

```
Switch(config-if)#switchport port-security mac-address sticky
Switch(config-if)#
```

Verifying Port Security

To verify port security, use the **show port-security**, **show port-security interface**, and **show port-security address** commands. The following command, **show port-security**, tells us that on Fast Ethernet 0/1 we have the maximum number of addresses that can be learned set to two, and currently we see two addresses on that interface. We can also see that six violations have occurred in the past, and that when there is a violation, the action is to restrict that port. Restricting on that port does not shut down the port, however; it just prevents traffic from the restricted address:

```
SNRS_SWITCH#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
                (Count)       (Count)          (Count)
---------------------------------------------------------------------
      Fa0/1       2             2                  6         Restrict
---------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 1
Max Addresses limit in System (excluding one mac per port) : 1024

SNRS_SWITCH#
```

In the following output of the **show port-security interface fa0/1** command, we can see detailed information about the port security configuration on this interface:

```
SNRS_SWITCH#show port-security interface f0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Restrict
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 2
Total MAC Addresses        : 2
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 2
Last Source Address        : 001c.b01d.d383
Security Violation Count   : 6

SNRS_SWITCH#
```

The following command, **show port-security address**, enables us to see information about our secure MAC address table. In this secure MAC address table, we can see that there are two MAC addresses that have been learned via the **sticky** command, and both have been learned on interface Fast Ethernet 0/1:

```
SNRS_SWITCH#show port-security address
          Secure Mac Address Table
-------------------------------------------------------------------
Vlan    Mac Address       Type                Ports   Remaining Age
                                                         (mins)
----    -----------       ----                -----   -------------
   1    0006.d7a4.4081    SecureSticky        Fa0/1        -
   1    001c.b01d.d3c1    SecureSticky        Fa0/1        -
-------------------------------------------------------------------
```

```
Total Addresses in System (excluding one mac per port)     : 1
Max Addresses limit in System (excluding one mac per port) : 1024

SNRS_SWITCH#
```

VLAN-Hopping Attacks

This attack involves an attacker who gains access to a VLAN other than the one he or she is assigned to. The attacker accomplishes this attack by connecting to a switch port that is enabled and mimicking the dynamic trunking protocol to establish a trunk link between itself, the attacker, and the switch. By establishing a trunk link, an attacker has access to all VLANs that can be carried on that trunk. The attacker can then send traffic to any VLAN that he wants, essentially hopping from VLAN to VLAN.

Another method of VLAN hopping involves double tagging, where a second 802.1q. tag is inserted in front of another 802.1q tag. Some switches will strip off only the first tag and then send the frame across a trunk link. With the second tag still intact, the attacker has successfully hopped VLANs. This type of attack is usually only successful as a one-way attack, but it can still be used for denial-of-service (DoS) attacks.

To mitigate VLAN hopping, set unused ports to access mode using the **switchport mode access** command, and assign it to a VLAN that is not in use. By assigning this port as an access port, you disable the ability for attackers to pretend that they are a trunk and to thus a establish trunk relationship on the port. By assigning it to a VLAN that is not in use, we black-hole this user who is trying to attack the network.

STP Vulnerabilities

This attack involves an attacker who wants to manipulate the Spanning Tree Protocol (STP) in an attempt to change the root bridge of the network or subnet. Because of the way STP works, all that has to happen is a bridge protocol data unit

(BPDU) needs to be heard on any port; in this case, spanning tree will have to reconverge. You can implement BPDU filtering, BPDU guard, and root guard to help protect your network from this type of attack. You can find more information about these mitigation techniques at the following site:

http://www.cisco.com/en/US/docs/switches/lan/catalyst2960/software/release/12.2_25_see/configuration/guide/swstpopt.html

MAC Spoofing: Man-in-the-Middle Attacks

This attack involves an attacker who falsifies his MAC address to execute a man-in-the-middle attack. One way that this can happen is by sending a gratuitous Address Resolution Protocol (ARP) and spoofing the MAC address of the device, such as the default gateway. When this happens and users send traffic to the default gateway, it will go through the attacker (thus creating a man-in-the-middle attack) and often you won't even know this is happening.

PVLAN Vulnerabilities

In a PVLAN attack, an attacker tries to gain access to data on a PVLAN. Using a Layer 3 device such as a router, an attacker sends traffic to the IP address of the device he is trying to attack. But, the attacker uses the MAC address of the router, hoping that the router will forward packets to the device being attacked using the IP address.

Configuring DHCP Snooping

DHCP snooping is a switch feature that determines which switch ports can respond to DHCP requests. You need this because two other attacks can be performed at Layer 2: DHCP starvation attacks and DHCP spoofing attacks. This section covers how these attacks work and how to configure DHCP snooping to help prevent them from happening.

DHCP Starvation and Spoofing Attacks

A DHCP starvation attack is a DoS attack in which an attacker floods the DHCP server with DHCP IP address requests in an attempt to use up all the DHCP addresses and starve the rest of the clients of valid IP addresses.

In a DHCP spoofing attack, the attacker sets up a DHCP server on a network to hand out erroneous DHCP addresses. This is an easy attack to perform because you don't need much to be a DHCP server. In fact, you can use Knoppix STD to do it. One example of how attackers benefit by becoming a DHCP server on the network is that they can then make themselves the default gateway for any clients they allocate DHCP addressing to. This creates a man-in-the-middle attack, and your data is then compromised. Any traffic you send can be decoded by the attacker using software such as WireShark.

Understanding DHCP Snooping and Mitigating DHCP Attacks

DHCP snooping is a switch feature that determines which switch ports can respond to DHCP requests. To accomplish this configuration, you must configure a port as either trusted or untrusted. Untrusted ports can source requests only, whereas trusted ports can source DHCP replies. This will help you prevent the attack by controlling where the DHCP server is and the path that you expect DHCP replies to come from.

Enabling and Configuring DHCP Snooping

To enable DHCP snooping, follow these steps:

1. Globally enable DHCP snooping. The following command globally enables DHCP snooping.

   ```
   switch(config)#ip dhcp snooping
   ```

2. Enable DHCP snooping on a VLAN or range of VLANs. The following command enables DHCP snooping for a range of VLANs. DHCP snooping is enabled on a VLAN only if both the global snooping and the VLAN snooping are enabled:

   ```
   switch(config)#ip dhcp snooping vlan vlan-range
   ```

3. Enter interface configuration mode. This will be the interface that is trusted (that is, where we expect to see a DHCP reply coming from):

   ```
   switch(config)#interface interface-id
   ```

4. Configure the interface as trusted where a DHCP server is connected to the switch. Use this command to enable trust on the interface:

   ```
   switch(config-if)#ip dhcp snooping trust
   ```

Optionally, configure the number of DHCP packets per second that an interface can receive. You configure this rate-limit command on untrusted interfaces, and you might not want to configure it to a hundred packets per second. Keep in mind that you can rate limit on trusted interfaces, but a trusted interface aggregates all DHCP traffic in the switch and so you must adjust that rate limit to a higher number:

```
switch(config-if)#ip dhcp snooping limit rate rate
```

Verifying DHCP Snooping

After configuring DHCP snooping, you can display the DHCP snooping configuration for a switch by using the **show ip dhcp snooping** command. In the following example, DHCP snooping is configured on interface Fast Ethernet 0/1. This port is configured as a trusted port:

```
SNRS_SWITCH#sh ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
none
Insertion of option 82 is enabled
Interface                    Trusted     Rate limit (pps)
— — — — — — — — — — — —      — — — -     — — — — — — — —
FastEthernet0/1              yes         unlimited
SNRS_SWITCH#
4d00h: %SYS-5-CONFIG_I: Configured from console by console
SNRS_SWITCH#
```

You can also display the dynamically configured bindings in the DHCP snooping binding database using the **show ip dhcp snooping binding** command.

CHAPTER 2
Trust and Identity

Implementing Identity Management

An important aspect of trust and identity being established in a network involves the ability to authenticate users and devices to a central, trusted repository. Cisco devices will use the TACACS+ plus or RADIUS protocol to authenticate users back to an authentication, authorization, and accounting (AAA) server. A number of AAA servers are on the market, including the Cisco Secure Access Control Server (ACS). The Cisco Secure ACS can be installed on a Microsoft Windows server and provides a central location for network devices to request authentication and authorization and to perform accounting.

AAA is the process of performing authentication, authorization, and accounting for users who require network resources. AAA is a framework in which additional protocols are needed for communication between AAA servers and AAA clients. Those additional protocols include TACACS+ and RADIUS. A brief discussion of each follows.

Cisco Secure ACS for Windows Overview

Cisco Secure ACS for Windows is a centralized identity networking solution that simplifies the management of users across all Cisco devices and security management applications. Cisco Secure ACS provides enforcement of policy for administrators and users who access a network. With reporting capabilities, ACS provides records for use in billing and network audits.

Cisco Secure ACS enables you to manage administrators of devices such as Cisco IOS routers, virtual private networks (VPNs), firewalls, dialup and digital subscriber line (DSL) connections, cable access solutions, storage, content, VoIP, Cisco wireless solutions, and Cisco Catalyst switches using IEEE 802.1x access control. Cisco Secure ACS is also an important component of Cisco Admission Control (NAC).

Authentication, Authorization, and Accounting

Authentication is the process of confirming the identity of a person or device that requests access to the network or for network resources. *Authorization* is the process of ensuring that authenticated users are allowed to perform the request based on policy. *Accounting* is the process of recording the activity of users or devices that have accessed the network.

TACACS+ and RADIUS

TACACS itself is an Internet Engineering Task Force (IETF) standard. TACACS+ is a Cisco proprietary extension to that standard and is TCP based and uses port 49. TACACS+ encrypts the entire body of the message that is sent between the network access server (NAS), which is the server that performs the authentication (in our case, Cisco Secure ACS), and the TACACS+ daemon that runs on the client device (IOS router, VPN concentrator, Adaptive Security Appliance [ASA], and so on). TACACS+ supports the use of Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), and MS-CHAP, and also provides command authorization capabilities.

RADIUS is a protocol that was developed by Livingston Enterprises. RADIUS is now an IETF standard that can be found in RFC 2865. RADIUS is User Datagram Protocol (UDP) based and uses ports 1645 and 1646 in most implementations, although those ports are not assigned to the RADIUS protocol. RADIUS is assigned ports 1812 and 1813, and newer implementations will use these ports. Two ports are used because authentication and authorization are done together on port 1812 or 1645 depending on implementation, and accounting is done separately using port 1813 or 1645 depending on implementation.

Either TACACS+ or RADIUS is required for a Cisco IOS device to communicate AAA information between the Cisco Secure ACS server and itself. Your decision to use one over the other may include the type of device that you will be using for authentication; for example, non-Cisco equipment would not use TACACS+. Another reason for choosing one over the other might be the type of feature that you are implementing; for example, if you're going to do command authorization, you need to use TACACS+; if you want to do downloadable IP access control lists (ACL), UDP is RADIUS.

Configuring TACACS+ and RADIUS

To enable the Cisco IOS device to communicate with the Cisco Secure ACS using TACACS+, follow these steps:

1. Globally enable AAA.
2. Specify AAA lists and methods.
3. Specify AAA server hosts' addresses.
4. Specify encryption keys used to encrypt data between the NAS and the AAA server.

The following configuration example first shows AAA being enabled on the SNRS router. It then shows an authentication method list for logins to the router using TACACS+. When users log in to the router, they will be authenticated with a username and password that is stored on the TACACS+ server. The TACACS+ server in this case is the Cisco Secure ACS server. Then in the configuration, authorization is configured using the **aaa authorization** and **exec** command. With this command, it instructs the router to check with the TACACS+ server and verify whether the user is allowed exact privileges. With the **aaa accounting** and **exec** command, accounting messages will be sent to the TACACS+ server, both when the session starts and when the session stops. The last two configuration lines define the protocol being used to communicate with the Cisco Secure ACS server as TACACS+. They also define the secret key that is used to encrypt the messages between the router and the AAA server:

```
SNRS_ROUTER(config)#aaa new-model
SNRS_ROUTER (config)#aaa authentication login default group tacacs+
SNRS_ROUTER (config)#aaa authorization exec default group tacacs+
SNRS_ROUTER (config)#aaa accounting  exec default start-stop group tacacs+
SNRS_ROUTER (config)#tacacs-server key secretkey
SNRS_ROUTER (config)#tacacs-server host 172.26.10.1 ref
```

This is just a simple configuration example, but there is much more to be understood with AAA configurations. For a detailed discussion about AAA and the Cisco Secure ACS, refer to *Cisco Secure Access Control Security AAA Administrative Services*, by Brandon Carroll (Cisco Press).

To enable the Cisco IOS device to communicate with the Cisco Secure ACS using RADIUS, follow these steps:

1. Globally enable AAA.
2. Specify AAA lists and methods.
3. Specify AAA server hosts' addresses.
4. Specify encryption keys used to encrypt data between the NAS and the AAA server.

The following configuration example is similar to the TACACS example shown previously. The difference with this example is that rather than using TACACS, we are using the RADIUS protocol for communication between the router and the AAA server:

```
SNRS_ROUTER(config)#aaa new-model
SNRS_ROUTER (config)#aaa authentication login default group tacacs+
SNRS_ROUTER (config)#aaa authorization exec default group tacacs+
SNRS_ROUTER (config)#aaa accounting exec default start-stop group tacacs+
SNRS_ROUTER (config)#radius-server key secretkey
SNRS_ROUTER (config)#radius-server host 172.26.10.1 ref
```

You can find a number of configuration examples at the following site:

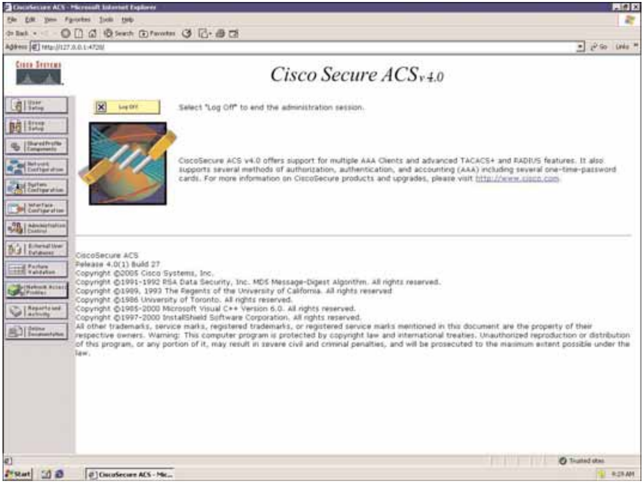
http://www.cisco.com/en/US/tech/tk59/tech_configuration_examples_list.html

Working in Cisco Secure ACS

Cisco Secure ACS is an AAA server. In the preceding section, you enabled the IOS devices to communicate with the AAA server. In this section, you will enable the AAA server (in this case, Cisco Secure ACS) to communicate to the IOS device.

Just about any administration tasks can be performed in the Cisco Secure ACS web interface. You access the web interface by browsing to http://<*server address*>:2002. From the web interface, you can easily modify and view the Cisco Secure ACS configuration. Figure 2-1 shows the layout of the HTML interface.

FIGURE 2-1
Cisco Secure ACS Interface Layout



Select "Log Off" to end the administration session.

CiscoSecure ACS v4.0 offers support for multiple AAA Clients and advanced TACACS+ and RADIUS features. It also supports several methods of authorization, authentication, and accounting (AAA) including several one-time-password cards. For more information on CiscoSecure products and upgrades, please visit http://www.cisco.com.

CiscoSecure ACS
Release 4.0(1) Build 27
Copyright ©2005 Cisco Systems, Inc.
Copyright ©1991-1992 RSA Data Security, Inc. MD5 Message-Digest Algorithm. All rights reserved.
Copyright ©1989, 1993 The Regents of the University of California. All rights reserved
Copyright ©1986 University of Toronto. All rights reserved.
Copyright ©1985-2000 Microsoft Visual C++ Version 6.0. All rights reserved.
Copyright ©1997-2000 InstallShield Software Corporation. All rights reserved.
All other trademarks, service marks, registered trademarks, or registered service marks mentioned in this document are the property of their respective owners. Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

If you plan to access and administer the Cisco Secure ACS from the network, you have to create and enable an administrator first. An administrative account is not created by default. To create one, follow these steps:

1. Click Administration Control.
2. Click Add Administrator.
3. Complete the text entry fields in the Administrator Details table to create the administrator name and password.
4. Click Grant All to choose all privileges, including user group editing privileges for all user groups.

User Setup

This is where you add a new user, search for an existing user, find users alphabetically or numerically, or list all users at once.

Group Setup

This is where you apply configurations from shared profile components and specific TACACS+ and RADIUS attributes. Group Setup is also where you can configure any parameters common to a group of users. Group settings can include enable passwords, time-of-day restrictions, downloadable IP access control lists (ACL), and any other setting that pertains to the entire group.

Shared Profile Components

This button enables an administrator to specify shell command authorization sets. These let you do two things: The first feature is command authorization, meaning that you can control the commands that can be entered on the IOS devices. The second is protocol authorization, meaning that you can control which protocols average users can pass through firewalls. You don’t need to know the latter feature for the certification exam, but it is something that you can do. Command authorization is accomplished by applying the command authorization set to the user profile in the TACACS+ settings or at the group level. It also requires some configuration on the IOS device.

Network Configuration

This button is where an administrator can add, delete, or modify settings for AAA clients (network access devices [NAD]). This is important because your IOS device is an AAA client.

Other areas of configuration include the following:

- System configuration
- Interface configuration
- Administration control
- External user databases
- Posture validation
- Network access profiles
- Reports and activity
- Online documentation

Of these additional configuration areas, the only one we cover is the network access profiles.

Network Access Profiles

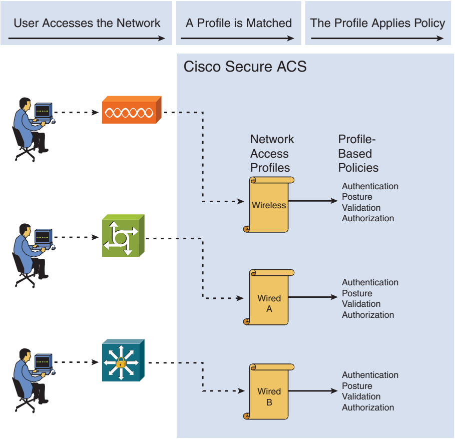
Cisco Secure ACS introduces the concept of network access profiles (NAP). Because organizations have many different users who access the network in many different ways, it's important to apply a security policy that fits the scenario in which they're accessing the network. NAPs are an ordered list of rules that, when a RADIUS transaction occurs, ACS uses to map the transaction to a policy. This is useful when doing network admission control (NAC).

Profile-Based Policies

Policies are applied by ACS going down the list of active NAPs. ACS processes down the list until a match is made similar to the way a router processes an access list. Actions are defined in the policies. When ACS matches the profile, it takes the action found in the policy.

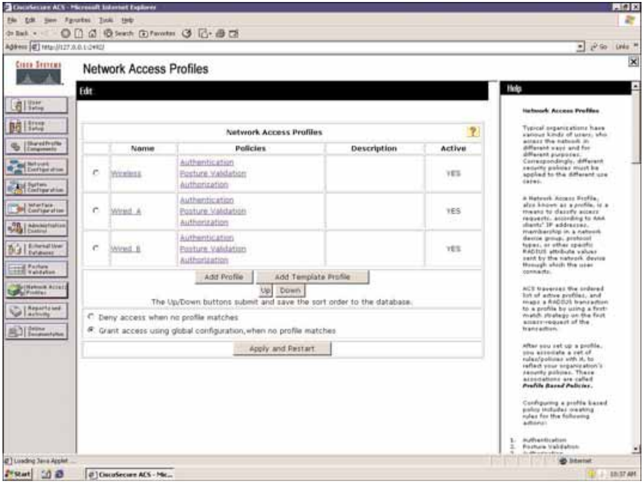
Figure 2-2 shows a sample network where NAPs might be used. When a user accesses the network and authenticates and the NAP called wireless is matched, authentication, posture validation, and authorization policies are applied. When a user accesses the network and authenticates via the "wired A" NAP, a separate set of authentication, posture validation, and authorization policies is applied (likewise when a user authenticates in to the NAP called wired B).

FIGURE 2-2
Network Access Profiles Example


User Accesses the Network
A Profile is Matched
The Profile Applies Policy
Cisco Secure ACS
Network Access Profiles
Profile-Based Policies
Wireless
Authentication Posture Validation Authorization
Wired A
Authentication Posture Validation Authorization
Wired B
Authentication Posture Validation Authorization

You can see this configuration in Figure 2-3. This figure shows a wireless profile. A Wired A profile and a Wired B profile. Each profile has authentication policies, posture validation policies, and authorization policies. We can also see that each of these profiles is active. By selecting the name wireless in the Network Access Profiles page, we gain access to the Profile Setup page, as shown in Figure 2-4. From this output, you can see that you can assign a description to a profile, you can select whether it's active, and you can apply a network access filter. In this example, no network access filter is applied; it just has the word *any*.

FIGURE 2-3
Network Access Profiles Configuration Page in ACS



Network Access Profiles

Network Access Profiles

	Name	Policies	Description	Active
○	Wireless	Authentication Posture Validation Authorization		YES
○	Wired A	Authentication Posture Validation Authorization		YES
○	Wired B	Authentication Posture Validation Authorization		YES

Add Profile Add Template Profile

Up Down

The Up/Down buttons submit and save the sort order to the database.

○ Deny access when no profile matches

● Grant access using global configuration,when no profile matches

Apply and Restart

Network Access Profiles

Typical organizations have various kinds of users, who access the network in different ways and for different purposes. Correspondingly, different security policies must be applied to the different use cases.

A Network Access Profile, also known as a *profile*, is a means to classify access requests, according to AAA clients' IP addresses, membership in a network device group, protocol types, or other specific RADIUS attribute values sent by the network device through which the user connects.

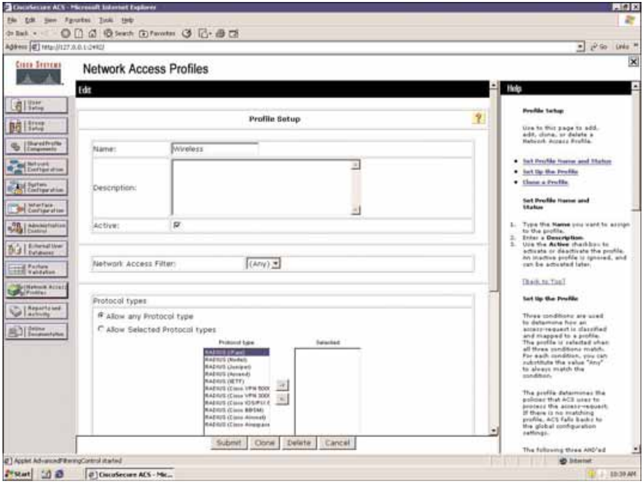
ACS traverses the ordered list of active profiles, and maps a RADIUS transaction to a profile by using a first-match strategy on the first access-request of the transaction.

After you set up a profile, you associate a set of rules/policies with it, to reflect your organization's security policies. These associations are called ***Profile Based Policies***.

Configuring a profile based policy includes creating rules for the following actions:

1. Authentication
2. Posture Validation

FIGURE 2-4 Profile Setup Page in ACS



Network Access Profiles

Profile Setup

Name: Wireless

Description:

Active: ☑

Network Access Filter: (Any)

Protocol types

● Allow any Protocol type
○ Allow Selected Protocol types

Profile Setup

Use to this page to add, edit, clone, or delete a Network Access Profile.

- Set Profile Name and Status
- Set Up the Profile
- Clone a Profile

Set Profile Name and Status

1. Type the **Name** you want to assign to the profile.
2. Enter a **Description**.
3. Use the **Active** checkbox to activate or deactivate the profile. An inactive profile is ignored, and can be activated later.

[Back to Top]

Set Up the Profile

Three conditions are used to determine how an access-request is classified and mapped to a profile. The profile is selected when all three conditions match. For each condition, you can substitute the value "Any" to always match the condition.

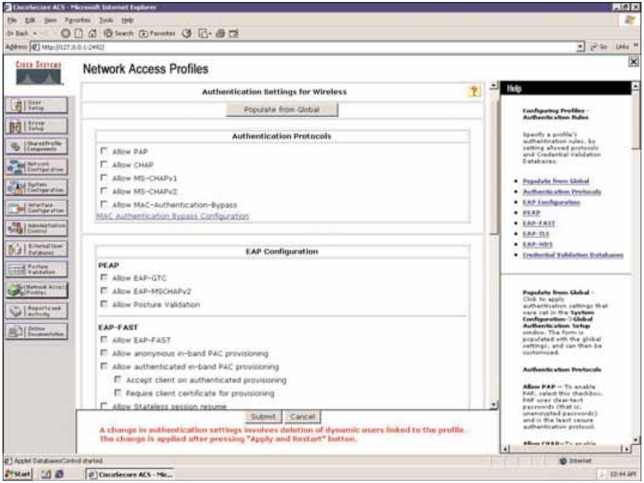
The profile determines the policies that ACS uses to protect the access-request. If there is no matching profile, ACS falls back to the global configuration settings.

The following three AND'ed

A network access filter is a way that you can apply this profile only when the request comes through specific network access devices. A network access device is a AAA client.

Returning to the Network Access Profiles configuration page shown in Figure 2-3, we can now explore the policies by clicking Authentication, Posture Validation, or Authorization. Figure 2-5 shows some of the options available in the Authentication Settings for Wireless configuration page. Notice here that you can set up authentication protocols such as allowing PAP or CHAP, and you can also set Extensible Authentication Protocol (EAP) configuration options.

FIGURE 2-5
Authentication Settings for Wireless



Network Access Profiles

Authentication Settings for Wireless

Authentication Protocols

- □ Allow PAP
- □ Allow CHAP
- □ Allow MS-CHAPv1
- □ Allow MS-CHAPv2
- □ Allow MAC-Authentication-Bypass

MAC Authentication Bypass Configuration

EAP Configuration

PEAP

- □ Allow EAP-GTC
- □ Allow EAP-MSCHAPv2
- □ Allow Posture Validation

EAP-FAST

- □ Allow EAP-FAST
- □ Allow anonymous in-band PAC provisioning
- □ Allow authenticated in-band PAC provisioning
  - □ Accept client on authenticated provisioning
  - □ Require client certificate for provisioning
- □ Allow Stateless session resume

A change in authentication settings involves deletion of dynamic users linked to the profile.
The change is applied after pressing "Apply and Restart" button.

Help

Configuring Profiles - Authentication Rules

Specify a profile's authentication rules, by setting allowed protocols and Credential Validation Databases.

- Populate from Global
- Authentication Protocols
- EAP Configuration
- PEAP
- EAP-FAST
- EAP-TLS
- EAP-MD5
- Credential Validation Databases

Populate from Global - Click to apply authentication settings that were set in the **System Configuration->Global Authentication Setup** window. The form is populated with the global settings, and can then be customized.

Authentication Protocols

Allow PAP — To enable PAP, select this checkbox. PAP uses clear-text passwords (that is, unencrypted passwords) and is the least secure authentication protocol.

Allow CHAP—To enable

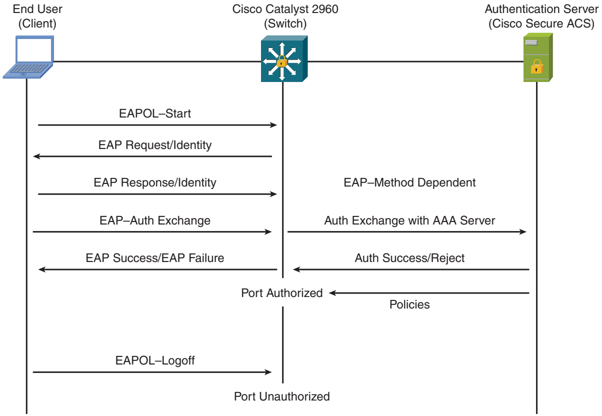
Implementing Cisco IBNS

The Cisco Identity-Based Networking Services (IBNS) model is another important topic related to the CCSP certification, in addition to being a key concept in the security of a network.

Cisco IBNS, 802.1x, and Port-Based Authentication

IBNS involves multiple protocols, concepts, and devices that include the IEEE 802.1x security. In a nutshell, IBNS provides services to network users depending on their identity. This involves the Extensible Authentication Protocol (EAP) for the user to communicate with the access devices. It also includes the RADIUS protocol for the access device to communicate with the AAA server. Figure 2-6 demonstrates the process of 802.1x in an IBNS environment.

FIGURE 2-6
802.1x Process in IBNS



Consider an example of this. When a user connects to the network, one of the first things needed is an IP address. To get an address, a PC sends out a request for one using DHCP. To provide IBNS, a user will use 802.1x before getting an IP address. For PCs that are enabled for 802.1x, the first request is an Extensible Authentication Protocol over LAN (EAPOL) request.

This request is received by the access device, such as a switch or a router. When the access device sees this request, it challenges the PC, which responds with the appropriate credentials. These credentials could be a user ID and password. The switch then forwards the request to a AAA server (Cisco Secure ACS) to authenticate the user's credentials via RADIUS.

If the user logs in successfully, the PC is provided an IP address and other information via DHCP on a subnet that allows access to the enterprise via the switch.

To perform this process, a number of EAP protocols can be used. EAP-MD5 is shown in Figure 2-7. EAP-TLS is shown in Figure 2-8. PEAP with MS-CHAPv2 is shown in Figure 2-9, and EAP-FAST is shown in Figure 2-10.

FIGURE 2-7
EAP-MD5

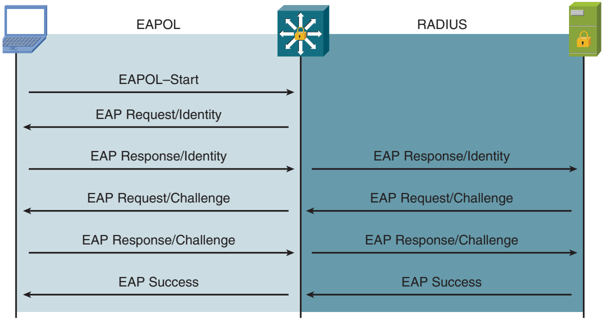


FIGURE 2-8
EAP-TLS

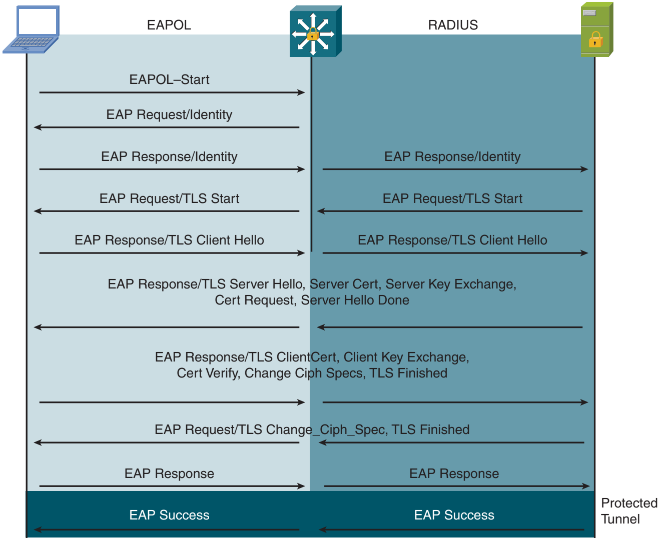

EAPOL
RADIUS
EAPOL–Start
EAP Request/Identity
EAP Response/Identity
EAP Response/Identity
EAP Request/TLS Start
EAP Request/TLS Start
EAP Response/TLS Client Hello
EAP Response/TLS Client Hello
EAP Response/TLS Server Hello, Server Cert, Server Key Exchange, Cert Request, Server Hello Done
EAP Response/TLS ClientCert, Client Key Exchange, Cert Verify, Change Ciph Specs, TLS Finished
EAP Request/TLS Change_Ciph_Spec, TLS Finished
EAP Response
EAP Response
EAP Success
EAP Success
Protected Tunnel

FIGURE 2-9
PEAP with MS-CHAPv2

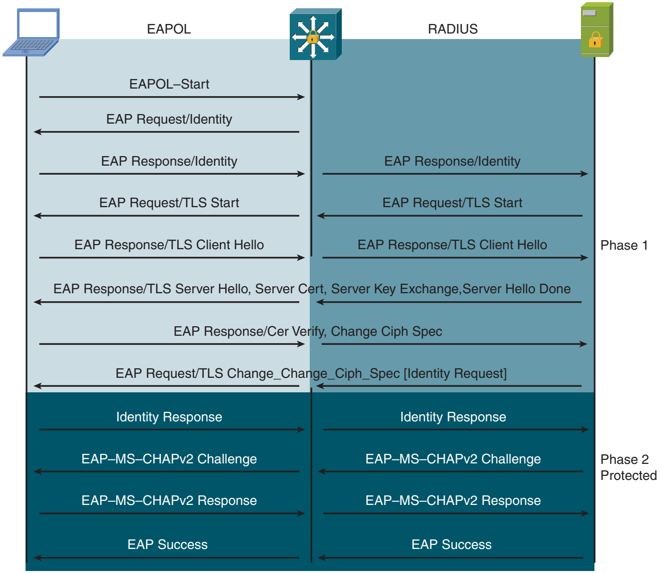
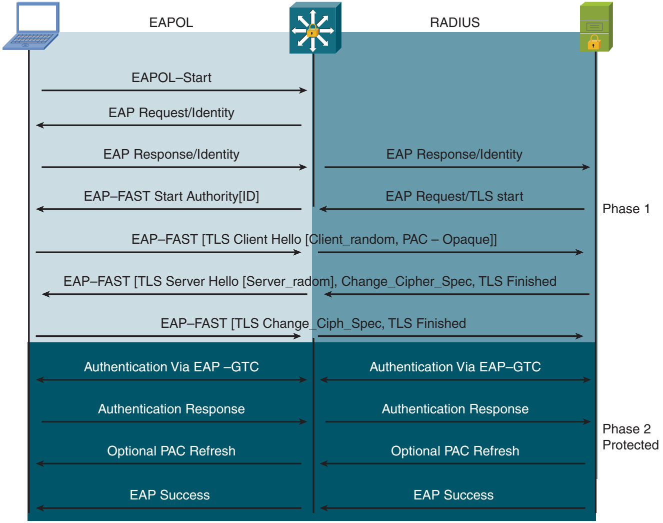


FIGURE 2-10
EAP-FAST


EAPOL
RADIUS
EAPOL–Start
EAP Request/Identity
EAP Response/Identity
EAP Response/Identity
EAP–FAST Start Authority[ID]
EAP Request/TLS start
Phase 1
EAP–FAST [TLS Client Hello [Client_random, PAC – Opaque]]
EAP–FAST [TLS Server Hello [Server_radom], Change_Cipher_Spec, TLS Finished
EAP–FAST [TLS Change_Ciph_Spec, TLS Finished
Authentication Via EAP –GTC
Authentication Via EAP–GTC
Authentication Response
Authentication Response
Optional PAC Refresh
Optional PAC Refresh
Phase 2
Protected
EAP Success
EAP Success

If a PC is not 802.1x capable, or the user does not log in successfully, the PC can be provided with limited access to the network, or be given no network access at all. The following site provides a more detailed explanation of the 802.1x protocol exchanges:

http://www.cisco.com/en/US/docs/switches/lan/catalyst4000/7.6/configuration/guide/8021x.html

802.1x

You'll want to understand the following characteristics of the IEEE 802.1x standard:

- 802.1x is a standard set by the IEEE 802.1 working group.
- It is designed to provide port-based control using authentication.
- EAP over LAN is the primary protocol used by 802.1x.
- The switch to PC Layer 2 protocol used is EAP.
- The actual enforcement is via MAC-based filtering and port-state monitoring.

802.1x defines the following components:

- **Supplicant**: Equivalent to a client
- **Authenticator**: Equivalent to an access device such as a switch or wireless access point (AP)
- **Authentication server**: Equivalent to a RADIUS server such as the Cisco Secure ACS

NOTE
For more information, refer to the *CCSP SNRS Exam Certification Guide* (Cisco Press).

802.1x and VLAN Assignment

IBNS enables you to control which VLANS your users are assigned to. This provides a convenient way of enforcing security policies. For example, a common security policy limits network access for certain users by using VLAN assignment. Back in Figure 2-6, we saw the process of 802.1x when a supplicant accesses the network. It's after the authentication success that policies are sent from the ACS to the authenticator (or in this case, the switch). Along with those policies comes the VLAN assignment for this user.

You will accomplish this using the **aaa authorization network {default} group radius** command. To configure 802.1x to provide VLAN assignment, follow these steps:

1. Enable AAA authorization on the switch.
2. Enable IEEE 802.1x on the switch.
3. Assign vendor-specific tunnel attributes in the RADIUS (Cisco Secure ACS) server. The RADIUS server must return these attributes to the switch:

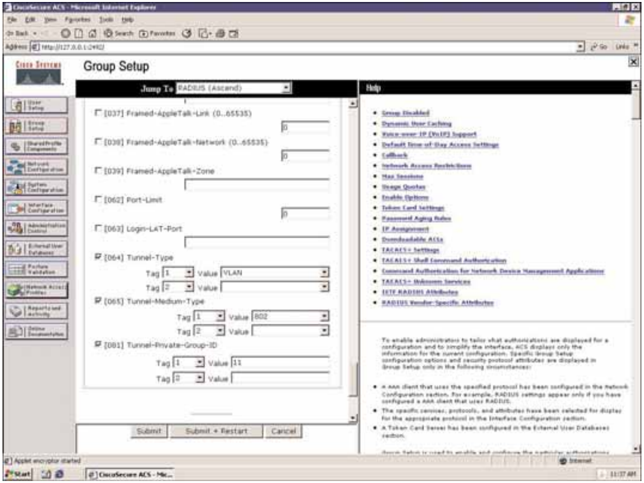
   [64] Tunnel-Type = VLAN

   [65] Tunnel-Medium-Type = IEEE 802

   [81] Tunnel-Private-Group-ID = VLAN name or VLAN ID

Attribute [64] must contain the value **VLAN**. Attribute [65] must contain the value **802**. Attribute [81] specifies the VLAN name or VLAN ID assigned to the IEEE 802.1x-authenticated user. Figure 2-11 shows this configuration on the ACS server.

FIGURE 2-11
Configuring the ACS Attributes



Group Setup

Jump To RADIUS (Ascend)

☐ [037] Framed-AppleTalk-Link (0..65535) 0

☐ [038] Framed-AppleTalk-Network (0..65535) 0

☐ [039] Framed-AppleTalk-Zone

☐ [062] Port-Limit 0

☐ [063] Login-LAT-Port

☑ [064] Tunnel-Type
Tag 1 Value VLAN
Tag 2 Value

☑ [065] Tunnel-Medium-Type
Tag 1 Value 802
Tag 2 Value

☑ [081] Tunnel-Private-Group-ID
Tag 1 Value 11
Tag 0 Value

Submit | Submit + Restart | Cancel

Help

- Group Disabled
- Dynamic User Caching
- Voice-over-IP (VoIP) Support
- Default Time-of-Day Access Settings
- Callback
- Network Access Restrictions
- Max Sessions
- Usage Quotas
- Enable Options
- Token Card Settings
- Password Aging Rules
- IP Assignment
- Downloadable ACLs
- TACACS+ Settings
- TACACS+ Shell Command Authorization
- Command Authorization for Network Device Management Applications
- TACACS+ Unknown Services
- IETF RADIUS Attributes
- RADIUS Vendor-Specific Attributes

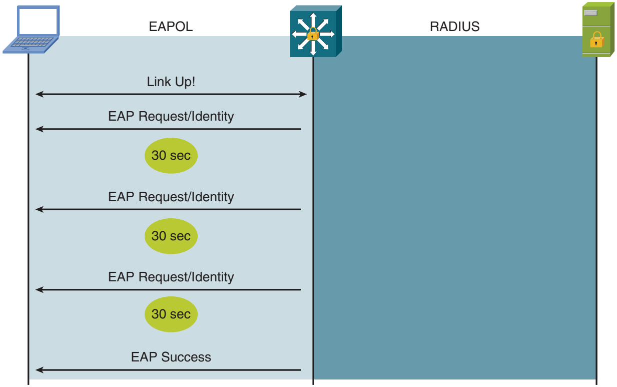
To enable administrators to tailor what authorizations are displayed for a configuration and to simplify the interface, ACS displays only the information for the current configuration. Specific Group Setup configuration options and security protocol attributes are displayed in Group Setup only in the following circumstances:

- A AAA client that uses the specified protocol has been configured in the Network Configuration section. For example, RADIUS settings appear only if you have configured a AAA client that uses RADIUS.
- The specific services, protocols, and attributes have been selected for display for the appropriate protocol in the Interface Configuration section.
- A Token Card Server has been configured in the External User Databases section.

802.1x and Guest VLANs

It is possible to configure a guest VLAN for each IEEE 802.1x port on the switch to provide limited services to clients, such as Internet access or downloading the IEEE 802.1x client. This type of configuration is important if you're going through migration or if you're just supporting guest devices on the network that do not support the 802.1x supplicant capability. When a device connects to a port, an EAPOL request is sent from the switch when the link comes up. If no response occurs for a period of time, another request is sent. Finally, after another timeout value, if the response is not seen, another request is sent. If after the third request no response is seen, and guest VLAN is configured for the port, 802.1x is disabled for the port, and it is placed in the guest VLAN. Figure 2-12 shows this process.

FIGURE 2-12
802.1x and Guest VLAN Process



To configure a guest VLAN, follow these steps:

1. Enable AAA.
2. Enable 802.1x guest VLAN behavior globally.
3. Configure the switch port as an access port.
4. Configure dot1x port control as **auto**.
5. Specify an active VLAN as a guest VLAN.

The following example configures interface Fa0/1 for guest VLAN behavior beginning with Step 3:

```
SNRS_SWITCH(config)#interface f0/1
SNRS_SWITCH(config-if)#switchport mode access
SNRS_SWITCH(config-if)#dot1x port-control auto
SNRS_SWITCH(config-if)#dot1x guest-vlan 10
```

802.1x and Restricted VLANs

The restricted VLAN feature lets you set up a VLAN that can be assigned to a client that does have 802.1x capabilities but is unable to authenticate. A good example of this is a user who uses 802.1x on his network and then travels with his laptop to a partner network that also uses 802.1x. The user is not going to have credentials in the partner authentication database and will fail authentication. Because his device is EAP0L capable, the user cannot be placed into the guest VLAN. A restricted VLAN will get them connected. To configure restricted VLANS, use the same configuration as the guest VLAN, but add the **dot1x auth-fail** *vlan_id* command.

Configuring 802.1x for a Wireless AP

The configuration of the Cisco switch or Cisco wireless AP is the same for any IEEE 802.1x deployment regardless of the EAP method chosen for authentication. One of the following EAP methods is negotiated between the client and the AAA server:

- EAP-MD5
- EAP-TLS
- PEAP with EAP-MS-CHAPv2
- EAP-FAST

To configure 802.1x, follow these steps:

1. Enable AAA.
2. Configure 802.1x authentication.
3. (Optional) Configure 802.1x authorization.
4. Configure RADIUS communications.
5. Enable 802.1x globally on the switch.
6. Verify 802.1x operation

The following example configuration enables 802.1x on a Fast Ethernet port.

Enable the AAA process:

```
Switch#configure terminal
Switch(config)#aaa new-model
```

Enable 802.1x authentication via the RADIUS server:

```
Switch(config)#aaa authentication dot1x default group radius
```

Enter the interface:

```
Switch(config)#interface fastethernet0/1
```

Enable 802.1x authentication control for the port:

```
Switch(config-if)#dot1x port-control auto
Switch(config-if)#exit
```

Define the RADIUS server, authentication port, and secret key:

```
Switch(config)#radius-server host 172.120.39.46 auth-port 1612 key secretkey
Switch(config)#end
```

To verify your configuration, you can use the **show dot1x statistics interface** command. To verify the AAA servers that are configured, use the **show aaa servers** command. The following output is from the **show dot1x statistics interface** command:

```
Switch#show dot1x statistics interface gigabitethernet0/1

GigabitEthernet0/1

    Rx: EAPOL     EAPOL     EAPOL     EAPOL     EAP       EAP       EAP
        Start     Logoff    Invalid   Total     Resp/Id   Resp/Oth  LenError
        0         0         0         21        0         0         0

        Last      Last
        EAPOLVer  EAPOLSrc
        1         0002.4b29.2a03

    Tx: EAPOL     EAP       EAP
        Total     Req/Id    Req/Oth
        622       445       0
```

CHAPTER 3
Cisco Network Foundation Protection

Introducing Cisco Network Foundation Protection

Cisco Network Foundation Protection (NFP) is a concept designed to protect the network infrastructure. Today our networks must connect to the Internet, and because we're connected to the Internet, we are open to numerous risks. NFP protects your network by providing security for your network infrastructure devices themselves. Your network devices are typically broken down into three pieces. The control plane routes your traffic. The data plane forwards your packets. And the management plane provides you management access. If any of these planes is inaccessible, that becomes a problem. NFP provides protection for each one of these planes. NFP uses the following IOS tools and features:

- Cisco AutoSecure, which provides you an easy way to secure your devices
- Control Plane Policing (CoPP)
- Control Plane Protection (CPPr)
- Flexible Packet Matching (FPM)
- Management Plane Protection (MPP)
- Quality of service (QoS) tools
- Unicast Reverse Path Forwarding (uRPF)

Although each of these features is important to the network, they are not all covered on the SNRS exam. The following site provides more information about NFS:

http://www.cisc.com/en/US/products/ps6642/products_ios_protocol_group_home.html

Securing the Control Plane

The following configuration creates a policy that polices the control plane. This policy defines a trusted host with the address 172.30.101.1. This host can forward traffic to the control plane without constraint. Other traffic that is sent to the control plane will be policed at 50,000 packets per second.

Create an ACL that denies the trusted host from being matched, and matches on all other untrusted addresses:

```
cisco_router(config)#ip access-list extended CP-acl
cisco_router(config-ext-nacl)#deny tcp host 172.30.101.1 any eq telnet
cisco_router(config-ext-nacl)#deny tcp host 172.30.101.1 any eq www
cisco_router(config-ext-nacl)#permit tcp any any eq telnet
cisco_router(config-ext-nacl)#permit tcp any any eq www
cisco_router(config-ext-nacl)#exit
```

Create a class map that matches the traffic from the ACL:

```
cisco_router(config)#class-map match-any CP-class
cisco_router(config-cmap)#match access-group name CP-acl
cisco_router(config-cmap)#exit
```

Create a policy map that calls the traffic from the ACL and polices it:

```
cisco_router(config)#policy-map CP-policy
cisco_router(config-pmap)#class CP-class
cisco_router(config-pmap-c)#police rate 50000 pps conform-action transmit exceed-action drop
cisco_router(config-pmap-c-police)#exit
cisco_router(config-pmap-c)#exit
cisco_router(config-pmap)#exit
```

Access the control plane and apply the policy using the **service-policy** command:

```
cisco_router(config)#control-plane host
cisco_router(config-cp-host))#service-policy input CP-policy
cisco_router(config-cp-host)#end
```

You can find a more detailed discussion about control plane protection at the following site:

http://www.cisc.com/en/US/products/ps6642/products_white_paper0900aecd805ffde8.shtml

Management Plane Protection

The management plane handles communication with the router itself, using protocols such as Telnet, Secure Shell (SSH), Simple Network Management Protocol (SNMP), Hypertext Transfer Protocol (HTTP), and HTTP over Secure Sockets Layer (HTTPS). If you lose management plane access, you cannot configure the device (and thus you essentially lose control of the device). You can use the following tools to protect the management plane:

- Cisco Management Plane Protection (MPP)
- SSH (allow SSH only)
- ACLs to filter the vty ports
- Cisco IOS Software login enhancement
- Role-based command-line interface (CLI) views

The Cisco MPP feature enables you to specify one or more interfaces as the management interface. What this configuration does is allow SSH and SNMP traffic to only access the device on interface Fast Ethernet 0/0. To configure MPP, enter the following commands:

```
control-plane host
management-interface FastEthernet 0/0 allow ssh snmp
```

You can find a more detailed discussion about MPP at the following site:

http://www.cisco.com/en/US/products/ps6441/products_feature_guide09186a0080617022.html

Securing the Data Plane

The data plane, also called the forwarding plane, is what moves most of your traffic that passes through the router. You can prevent certain attacks by denying them from passing through the router. To secure the data plane on Cisco routers, use Flexible Packet Matching (FPM). FPM provides deeper inspection than standard IOS tools to protect against data plane attacks such as Code Red, Nimda, the SQL Slammer, and Blaster. FPM uses Protocol Header Definition File (PHDF), which is nothing more than an Extensible Markup Language (XML) file that is ready-packaged by Cisco and used to match patterns in traffic. When deploying FPM, follow these steps (taken from the FPM deployment guide):

1. Determine the characteristics of the attack. Some questions that may help in understanding the nature of the attack include these: Does the attack use a specific protocol? Are unique patterns present at specific places within the packets? Does the attack always target a specific port? Are the packets always a specific length?
2. If the results of Step 1 conclude that FPM is useful for mitigating the attack, determine whether existing PHDFs, a custom PHDF, or no PHDFs are required to define the FPM policy. If existing PHDFs are acceptable, skip Step 3 and proceed to Step 4. If a custom PHDF is required, proceed to Step 3. If no PHDFs are required (in which case class maps must only use the two permanently defined starting points from the Layer 2 header or the Layer 3 header), skip Steps 3 and 4 and proceed directly to Step 5.
3. Write a custom PHDF for any protocol involved in the attack that is not already covered by an existing PHDF.
4. Load all PHDFs needed to describe the packet contents so that match statements can be written based on convenient PHDF-defined offsets.
5. Configure class maps, policy maps, and services policies to identify the traffic and take an action.
6. Apply the service policies to appropriate interfaces.

The preceding six steps detail how to design and implement FPM. To configure FPM, follow these steps:

1. Load a PHDF from flash memory.

   It is loaded into the router to define additional protocols that the router can filter.

   ```
   router(config)#load protocol flash:ip.phdf
   router(config)#load protocol flash:udp.phdf
   ```

   After the appropriate PHDFs have been loaded, you must define a **class-map** command with type **stack** so that FPM knows which headers are present and in which order.

   After the stack of protocols has been defined, a class map of type **access-control** is defined for classifying packets.

2. Create a traffic class by defining class maps:

   ```
   router(config)#class-map type stack match-all ip-udp
   router(config-cmap)#description match UDP over IP packets
   router(config-cmap)#match field ip protocol eq 0x11 next udp
   router(config-cmap)#exit
   router(config)#class-map type access-control match-all slammer
   router(config-cmap)#description "match on slammer packets"
   router(config-cmap)#match field udp dest-port eq 0x59A
   router(config-cmap)#match field ip length eq 0x194
   router(config-cmap)#match start l3-start offset 224 size 4 eq 0x4011010
   ```

   A policy map is an ordered set of classes that are associated to actions. The policy binds the class and action. Actions can be drop, Internet Control Message Protocol (ICMP) response, and log, or service policy to nest another policy.

3. Create a traffic policy by defining a service policy:

```
router(config)#policy-map type access-control fpm-udp-policy
router(config-pmap)#description "policy for UDP based attacks"
router(config-pmap)#class slammer
router(config-pmap-c)#drop
router(config-pmap-c)#exit
router(config-pmap)#exit
router(config)#policy-map type access-control fpm-policy
router(config-pmap)#description "drop worms and malicious attacks"
router(config-pmap)#class ip-udp
router(config-pmap-c)# service-policy fpm-udp-policy
router(config-pmap-c)#exit
router(config-pmap)#exit
```

4. Apply the service policy to an interface:

```
router(config)#interface FastEthernet 0/1
router(config-if)#service-policy type access-control input fpm-policy
```

To verify FPM, use the commands **show protocols phdf ip**, **show flash**, **show class-map type stack**, **show class-map type access-control**, **show policy-map type access-control**, and **show policy-map type access-control interface** *interface*.

You can find the FPM deployment guide at the following site:

http://www.cisco.com/en/US/products/ps6723/products_white_paper0900aecd803936f6.shtml

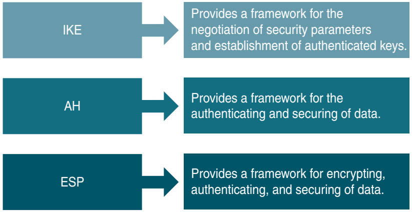
CHAPTER 4

Secured Connectivity

IPsec Overview

IPsec is a suite of protocols designed to provide security services for IP traffic. Major IPsec protocols include Authentication Header (AH), Encapsulating Security Payload (ESP), and Internet Key Exchange (IKE), as shown in Figure 4-1.

FIGURE 4-1
Major IPsec Protocols


IKE
Provides a framework for the negotiation of security parameters and establishment of authenticated keys.
AH
Provides a framework for the authenticating and securing of data.
ESP
Provides a framework for encrypting, authenticating, and securing of data.

It will be helpful to understand the major terms and protocols that are involved in IPsec negotiations. Those protocols and terminologies include the following:

- **Authentication Header**: AH, defined in RFC 2402, uses IP number 51 and is a mechanism for providing integrity and authentication of IP packets. AH does not provide encryption.

- **Encapsulating Security Payload**: ESP is defined in RFC 2406 and uses IP number 50. ESP provides authentication and encryption services for IP packets.
- **Internet Key Exchange**: IKE is defined in RFC 2409 and is actually a hybrid protocol that consists of SKEME, Oakley, and Internet Security Association and Key Management Protocol (ISAKMP). IKE uses the Diffie-Hellman (DH) key exchange to create a shared secret key between devices by which encryption keys are derived. Without IKE, this would be a daunting, manual task that would not scale. IKE works in two phases. In Phase 1, peers establish a secure channel using an ISAKMP policy. This is negotiated using either main mode or aggressive mode exchange. During this process, each peer is authenticated and DH exchanges take place. When the secure channel has been established, IKE moves to Phase 2. In Phase 2, security associations (SA) are negotiated between devices that need key material or parameter negotiation. This is done in what is known as quick mode. After an SA has been established, encrypted data can be sent.
- **Internet Security Association and Key Management Protocol**: ISAKMP is defined in RFC 2408. Cisco uses this term and the term IKE synonymously, but they are different. ISAKMP uses UDP port 500 and defines the method for authenticating a peer, creating and managing an SA, and techniques for generating keys.
- **Rivest, Shamir, and Adleman (RSA)**: This is an asymmetric public key cryptographic algorithm (named after its inventors, Rivest, Shamir, and Adleman) with a variable key length. This method of encryption is not used to secure user traffic in a VPN; instead, it is used to authenticate peers during DH negotiation.
- **Secure Hash Algorithm (SHA)**: SHA is a one-way, 160-bit hash algorithm used to provide data authentication.
- **Message Digest Algorithm 5 (MD5)**: MD5 is a one-way, 128-bit hash algorithm used to provide data authentication.
- **Data Encryption Standard (DES)**: DES is a 56-bit encryption algorithm used in classic cryptography but considered weak by today's standards. With DES, one 64-bit key is used.
- **Triple DES (3DES)**: 3DES is a mode of DES that encrypts three times. With 3DES, three 64-bit keys are used.
- **Advanced Encryption Standard (AES)**: AES is based on the Rijndael algorithm, which specifies how to use keys with a length of 128, 192, or 256 bits to encrypt blocks with a length of 128, 192, or 256 bits (all nine combinations

of key length and block length are possible).

- **Diffie-Hellman**: DH is a component of Oakley that negotiates a shared key over an insecure medium such as the public Internet.
- **Perfect Forward Secrecy (PFS)**: PFS ensures that an SA is not derived from another SA.
- **Security association**: Both AH and ESP make use of SAs, and a major function of IKE is the establishment and maintenance of SAs. An SA is a simplex connection that provides security services, by the use of AH or ESP, to the traffic carried by it. For ESP, you will have an inbound SA and an outbound SA. The SA is found in the SA database (SADB). This SADB is referenced when looking for the agreed upon rules for encrypting or decrypting traffic to or from an IPsec peer.

Cisco IOS VPNs

Because IPsec VPNs are used as an alternative to WAN technologies, various deployment options are available. Those include variations of site-to-site VPNs and remote-access VPNs. This section highlights those deployment options.

The IPsec VPN deployment options that we address include the following:

- **Fully meshed IPsec VPNs**: This is a site-to-site VPN. In a fully meshed site-to-site VPN, traffic from the LAN on one side of the VPN tunnel is encrypted when communicating with the LAN on the other end of the VPN tunnel. This is an alternative to traditional WANs such as Frame Relay and leased lines. Common characteristics of site-to-site VPNs include the following:

  Public IP addresses between pairs

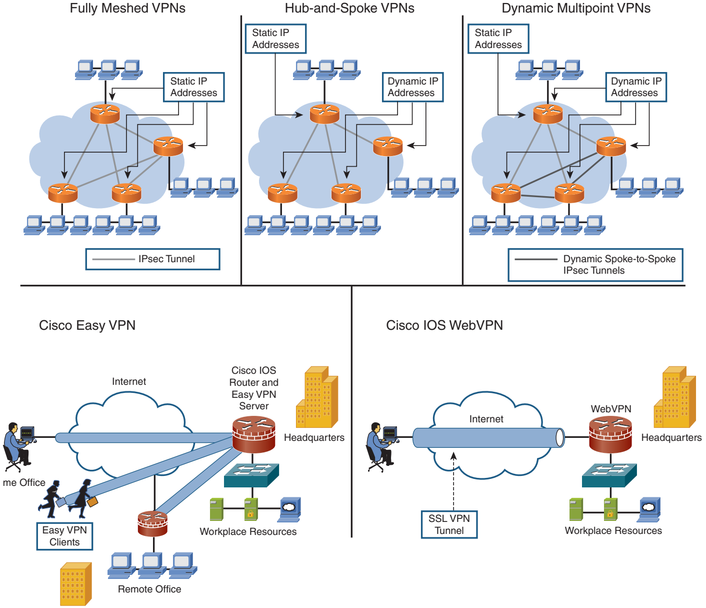
  Local LAN addressing that is either private or public IP addressing

  The drawback or restriction is that each site needs to have a public, static IP address that is reachable from the other site. When you add a site in a fully meshed site-to-site scenario, all sites need to be configured to talk to the new site. A full mesh differs from hub and spoke in that with a full mesh all sites are configured to establish a tunnel to all sites. The benefit here is that all sites can encrypt directly to the other. The drawback is that each site has a higher number of IPsec connections (which consumes valuable router resources).

- **Hub-and-Spoke IPsec VPNs**: This is a site-to-site VPN. This type of VPN connects sites, so it's still a site-to-site VPN, but the spoke side doesn't have to have a static IP address. This is a common VPN structure in small-branch and home-office scenarios. One problem with this VPN topology is that getting one spoke to talk to another spoke can prove difficult.
- **Dynamic multipoint VPNs (DMVPN)**: This is a site-to-site VPN. This scenario overcomes the problem seen in hub-and-spoke VPNs in getting one spoke to talk to another spoke. DMVPN uses additional technologies that need to be combined together. These technologies include multipoint generic routing encapsulation (mGRE) and Next Hop Resolution Protocol (NHRP). In a nutshell, you have a static tunnel to the hub site, and using NHRP you can resolve spoke sites when you need direct VPN connections to them.
- **Cisco Easy VPN**: This is a remote-access VPN. This technology makes it simple for a client to connect remotely to a central site. In Cisco Easy VPN, a client software application needs to be installed on computers that want to remotely access the network. A simple configuration is all that is needed to connect to the central site.
- **WebVPN**: This is a remote-access VPN. With WebVPN, the client side is even more simplified because you don't have to install a VPN client. Instead, end users access the network through a secure SSL tunnel established using a web browser. In this case, you no longer have to support VPN client issues.

Figure 4-2 shows these VPN deployments.

FIGURE 4-2
VPN Deployment Options



Implementing IPsec VPNs Using Pre-Shared Keys

To configure IPsec using pre-shared keys, follow the steps outlined in this section.

Step 1. Preparing for IPsec on the Network

In this task, ensure that the IPsec protocol is allowed through the network. Check the ACLs both on the device that will terminate the IPsec VPN and on all devices in the transit path. You may need to allow UDP-500 (ISAKMP), UDP-4500 (NAT-T), ESP, and AH.

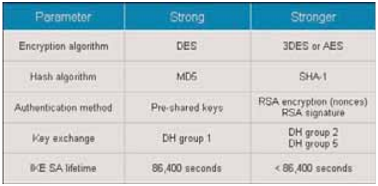
Step 2. Planning the IKE Policy

You need to plan the IKE Phase 1 parameters. These can include DES, MD5, SHA, DH, a keying method, and lifetimes. An example of an IKE policy follows:

- AES-128
- SHA
- DH-2
- Pre-shared keys
- Lifetime 86,400 seconds

Figure 4-3 displays a comparison of IKE policy options.

FIGURE 4-3
IKE Policy Comparison



Parameter	Strong	Stronger
Encryption algorithm	DES	3DES or AES
Hash algorithm	MD5	SHA-1
Authentication method	Pre-shared keys	RSA encryption (nonces) RSA signature
Key exchange	DH group 1	DH group 2 DH group 5
IKE SA lifetime	86,400 seconds	< 86,400 seconds

Step 3. Planning the IPsec Policy

Plan an IPsec policy that will secure you user traffic adequately. Your policy can include an encryption, a hash method, and an access list that defines which traffic to encrypt. And example of an IPsec policy follows:

- AES
- SHA
- PFS
- Access list (100, for example):

```
access-list 100 permit ip 10.0.0.0 0.0.0.255 192.168.1.0 0.0.0.255
```

- Lifetime 86,400 seconds or 4.6 GB

Step 4. Configuring ISAKMP

Start by using the **crypto isakmp enable** command to ensure that ISAKMP is turned on. Then use the **isakmp policy** commands to configure an ISAKMP policy. The following example enables an ISAKMP policy 150 using AES encryption, SHA hashing, pre-shared key authentication, and DH group 2:

```
SNRS_ROUTER(config)#crypto isakmp enable
SNRS_ROUTER(config)#crypto isakmp policy 150
SNRS_ROUTER(config-isakmp)#encryption aes-128
SNRS_ROUTER(config-isakmp)#hash sha
SNRS_ROUTER(config-isakmp)#authentication pre-share
SNRS_ROUTER(config-isakmp)#group 2
```

Step 5. Configuring Pre-Shared Keys

The next step is to configure the pre-shared keys. Use the **crypto isakmp key** command to configure the pre-shared key to be used to authenticate the remote peer. The following example configures a pre-shared key of cisco_snrs to be used with the remote peer at address 151.32.67.1:

```
SNRS_ROUTER(config)#crypto isakmp key cisco_snrs address 151.32.67.1
```

Step 6. Configuring IPsec Policies

After configuring the Phase 1 parameters, configure the Phase 2 parameters. These parameters include a transform set and global IPsec SA lifetimes. In addition to the transform set, you can change the mode that the tunnel operates in. There are two modes: tunnel mode and transport mode. Tunnel mode is the default mode. In tunnel mode, the original IP addresses are tunneled inside the encrypted header. In transport mode, the original IP addresses in the header are not encrypted and are used in the routing of the packet. In the following example, a transform set is configured with the name of MYTRANSFORM**.** This transform uses AES encryption and SHA for hashing. The mode is then set to tunnel. Finally, the default lifetime of 86,400 is changed to 43,200 seconds.

```
SNRS_ROUTER(config)#crypto ipsec transform-set MYTRANSFORM esp-aes-128 esp-sha-hmac
SNRS_ROUTER (cfg-crypto-trans)#mode tunnel
SNRS_ROUTER (config)#crypto ipsec security-association lifetime seconds 43200
```

Step 7. Defining the Crypto ACLs

The next step in the IPsec configuration is to define the traffic that is going to be encrypted. This is done by defining a crypto ACL. A crypto ACL is a regular, extended IP ACL, but the implicit deny that is usually found at the end of an ACL is not a deny that will drop the packet; instead, it is a deny of the packet being encrypted. To put it simply, if your packet matches the ACL, it gets encrypted; and if it doesn't match, it is sent in clear text. The following example will cause traffic from the 192.168.1.0/24 network to be encrypted when destined for the 192.168.2.0/24 network:

```
SNRS_ROUTER(config)#access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
```

Step 8. Creating the Crypto Map

The next step is to tie the configuration together using a crypto map. A crypto map defines the following:

- Which traffic should be protected by IPsec (crypto ACL)
- Where IPsec-protected traffic should be sent (the remote IPsec peer)
- Which IPsec security type should be applied (transform sets)
- Whether SAs are established manually or are established via IKE
- Other parameters that might be necessary to define an IPsec SA

The following example creates the crypto map MYMAP that matches access list 101, terminates to the peer 151.32.67.1 using transform set MYTRANSFORM to protect user data, and sets the lifetime to 28,800 seconds. In addition, this VPN will be negotiated dynamically using ISAKMP:

```
SNRS_ROUTER(config)#crypto map MYMAP 10 ipsec-isakmp
SNRS_ROUTER (config-crypto-map)#match address 101
SNRS_ROUTER (config-crypto-map)#set peer 151.32.67.1
SNRS_ROUTER (config-crypto-map)#set transform-set MYTRANSFORM
SNRS_ROUTER (config-crypto-map)#set security-association lifetime seconds 28800
```

Step 9. Applying Crypto Maps to Interfaces

The last step in establishing the VPN is to apply this policy to the interface. This is done using the **crypto map** command on the interface. The following example enables the policy on interface g0/1:

```
router(config)#interface g0/1
router(config-if)#crypto map MYMAP
```

To verify the connection, use the **show crypto ipsec sa** command. This command will give detailed information about the SA, including packets encrypted and decrypted:

```
SNRS_ROUTER#show crypto ipsec sa

interface: FastEthernet0
    Crypto map tag: CISCO, local addr. 14.15.16.17

   local  ident (addr/mask/prot/port): (10.5.6.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.23.9.0/255.255.255.0/0/0)
   current_peer: 22.23.24.25
     PERMIT, flags={origin_is_acl,}
```

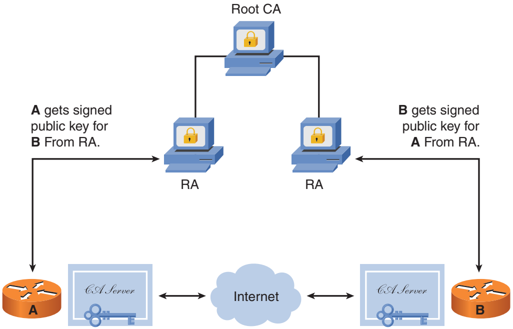
```
# pkts encaps: 12, # pkts encrypt: 12, # pkts digest 12
# pkts decaps: 23, # pkts decrypt: 23, # pkts verify 23
# pkts compressed: 0, # pkts decompressed: 0
# pkts not compressed: 0, # pkts compr. failed: 0, # pkts decompress failed: 0
# send errors 0, # recv errors 0

 local crypto endpt.: 14.15.16.17, remote crypto endpt.: 22.23.24.25
 path mtu 1500, media mtu 1500
 current outbound spi: 3C39A800

 inbound esp sas:
  spi: 0xD7228E4B(3609366091)
    transform: esp-3des esp-sha-hmac ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 2000, flow_id: 1, crypto map: CISCO
    sa timing: remaining key lifetime (k/sec): (4607999/3574)
    IV size: 8 bytes
    replay detection support: Y

 inbound ah sas:

 inbound pcp sas:

 outbound esp sas:
  spi: 0x3C39A800(1010411520)
    transform: esp-3des esp-sha-hmac ,
    in use settings ={Tunnel, }
```

```
       slot: 0, conn id: 2001, flow_id: 2, crypto map: CISCO
       sa timing: remaining key lifetime (k/sec): (4607999/3574)
       IV size: 8 bytes
       replay detection support: Y

    outbound ah sas:

    outbound pcp sas:
```

Implementing IPsec VPNs Using PKI

Public Key Infrastructure with IPsec makes your deployment more scalable by using a trusted third party to authenticate each user, as opposed to pre-shared keys on each device. Instead of having pre-shared keys to configure on each router that must match on the other end, you just configure the router to enroll with a trusted third party that will authenticate the routers identity. To perform a VPN connection using this authentication method, you need an X.509v3 certificate. The certificate is digitally signed by a certificate authority (CA). The signature is encrypted using RSA key pairs. Sometimes you get a certificate directly from a CA server, and at other times you get it from a registration authority (RA). An RA can issue certificates on behalf of another CA server. Figure 4-4 shows a typical CA/RA setup and how each endpoint enrolls.

FIGURE 4-4
CA/RA Scenario


Root CA
A gets signed public key for B From RA.
B gets signed public key for A From RA.
RA
RA
A
CA Server
Internet
CA Server
B

The Enrollment Process and Simple Certificate Enrollment Protocol

When enrolling with a CA server, you need to obtain two certificates. The first certificate is the CA server's certificate. The second certificate is your identity certificate that has been digitally signed by the CA server verifying who you are. This can be a manual process, in which case you would first obtain the CA server certificate and then send an enrollment request to the CA server. The enrollment request contains information about your identity. The CA server makes an identity certificate available to you after verifying the information about you. This process can be done in a more automated fashion using a Cisco-developed protocol called the Simple Certificate Enrollment Protocol (SCEP).

The process of enrolling with a CA using SCEP is as follows:

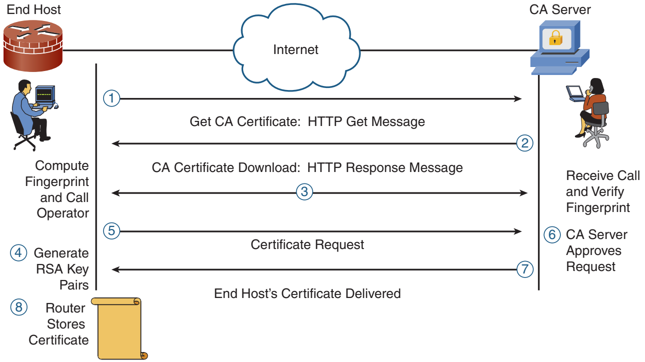
1. The router generates a Get CA certificate: HTTP Get message to retrieve the CA server certificate.
2. The server generates a CA certificate download: HTTP Response message.

NOTE
The RSA key generation can be done manually, but that process is not covered in this document

3. The server certificate is downloaded, and the fingerprint is verified via a phone call.
4. The router generates RSA key pairs.
5. The router generates a certificate request and sends it to the CA server.
6. The CA server receives the request and either requires manual intervention to approve it or approves the request depending on the configuration of the CA server.
7. Once approved, the CA server signs the certificate using its private key and returns the certificate to the host.
8. The router stores the certificate.

To perform this process and handle the communication with the CA server, the router uses SCEP. SCEP uses HTTP to communicate with the CA server. SCEP is Cisco proprietary. Figure 4-5 shows the SCEP process.

FIGURE 4-5
The SCEP Process


End Host
Internet
CA Server
1
Get CA Certificate: HTTP Get Message
2
CA Certificate Download: HTTP Response Message
3
Compute Fingerprint and Call Operator
Receive Call and Verify Fingerprint
5
Certificate Request
6 CA Server Approves Request
4 Generate RSA Key Pairs
7
End Host's Certificate Delivered
8 Router Stores Certificate

Configuring IPsec VPN Using Digital Certificates

There is only a slight difference in the configuration of IKE Phase 1 for use of digital certificates. To do so, just change the command **authentication pre-share** to **authenticate rsa-sig**. The following example changes the authentication method and configured CA support:

```
router(config)#clock timezone cst -6
```

Set the time so that the certificates will be valid (*time is important*):

```
router#clock set 23:21:00 08 September 2007
```

Set the hostname that's used in generating RSA keys:

```
router(config)#hostname SNRS_ROUTER
```

Set the domain that's used in generating RSA keys:

```
SNRS_ROUTER(config)#ip domain-name ciscopress.com
```

Create a host entry for the CA server that will allow you to use the name rather than the IP address later on:

```
SNRS_ROUTER(config)#ip host vpnca 172.26.26.51
```

Generate the RSA key pairs:

```
SNRS_ROUTER(config)#crypto key generate rsa
The name for the keys will be: SNRS_ROUTER.ciscopress.com
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
*Sep 08 23:46:09.839: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

Define the CA server:

```
SNRS_ROUTER(config)#crypto pki trustpoint vpnca
```

Define the enrollment URL that SCEP will use in contacting the CA server:

```
SNRS_ROUTER(ca-trustpoint)#enrollment url http://vpnca:80
```

Authenticate the CA server by calling the administrator of the CA and verifying the fingerprint:

```
SNRS_ROUTER(config)#crypto pki authenticate VPNCA
Certificate has the following attributes:
       Fingerprint MD5: 02DA1AB0 4FC8EFDE 3FB2ED92 5C96B72E
      Fingerprint SHA1: FFDE44F8 FA712C7B FA66F08C 08D548B7 5F05933D

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
```

Tell the router to enroll with the CA server:

```
SNRS_ROUTER(config)#crypto pki enroll VPNCA%
% Start certificate enrollment.
% Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to
 revoke your certificate. For security reasons your password will not be saved in the configuration. Please make a
 note of it.
Password: cisco123
Re-enter password: cisco123
% The subject name in the certificate will include: SNRS_ROUTER.ciscopress.com
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Request certificate from CA? [yes/no]: yes
```

```
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate vpnca verbose' command will show the fingerprint.
*Sep 08 23:49:17.404: CRYPTO_PKI:  Certificate Request Fingerprint MD5: D35C6688
 E6EBADEF 504EE6F2 BEC8FA13
*Sep 08 23:49:18.412: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 1A45EA0
A 6725B055 E84018FB 9DE5DD88 4E1C2CF5
*Sep 08 23:49:20.933: %PKI-6-CERTRET: Certificate received from Certificate Authority
```

Save the certificates that have been received:

```
SNRS_ROUTER#copy system:running-config nvram:startup-config
SNRS_ROUTER#config t
SNRS_ROUTER(config)#crypto isakmp policy 150
```

Use the certificates in authenticating the peer by changing this to **rsa-sig**. Remember that the authentication was “pre-shared keys” in the previous example:

```
SNRS_ROUTER(config-isakmp)#authentication rsa-sig
SNRS_ROUTER(config-isakmp)#end
SNRS_ROUTER#
```

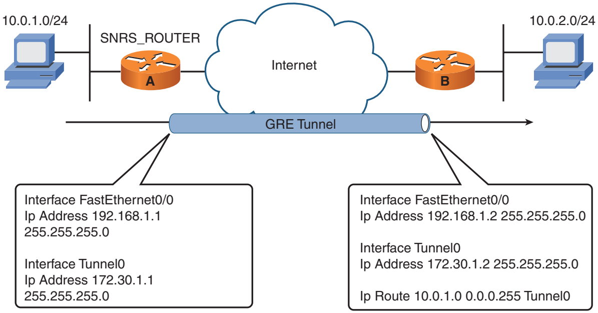
Configuring GRE Tunnels (and Why That’s Important)

Sometimes you will need to (or want to) encrypt and tunnel your routing protocol traffic. Some popular routing protocols cannot be encrypted and tunneled by IPsec. GRE is a tunneling protocol that is designed to encapsulate network layer packets inside of a new IP header. It is defined in RFCs 1701, 1702, and 2784. GRE uses IP number 47. This will help us accomplish the task of sending encrypted routing protocols over the Internet.

To begin, start by tunneling traffic without encryption using the **interface tunnel** command. The following example configures the router for GRE, as shown in Figure 4-6:

```
SNRS_ROUTER(config)#interface tunnel 0
SNRS_ROUTER(config-if)#ip address 172.16.1.1 255.255.255.0
SNRS_ROUTER(config-if)#tunnel source 192.168.1.1 255.255.255.0
SNRS_ROUTER(config-if)#tunnel destination 192.168.1.2 255.255.255.0
SNRS_ROUTER(config-if)#no shutdown
SNRS_ROUTER(config-if)#exit
SNRS_ROUTER(config)#ip route 10.0.2.0 255.255.255.0 tunnel 0
```

FIGURE 4-6
GRE Tunnels



Interface FastEthernet0/0
Ip Address 192.168.1.1
255.255.255.0

Interface Tunnel0
Ip Address 172.30.1.1
255.255.255.0

Interface FastEthernet0/0
Ip Address 192.168.1.2 255.255.255.0

Interface Tunnel0
Ip Address 172.30.1.2 255.255.255.0

Ip Route 10.0.1.0 0.0.0.255 Tunnel0

Ip Route 10.0.2.0 0.0.0.255 Tunnel0

Configuring GRE Tunnels and Encryption

GRE, although a tunneling protocol, is not secure because it does not perform encryption. To add to the security of GRE, we can encrypt it with IPsec. This is a common configuration when trying to run dynamic routing protocols such as Enhanced Interior Gateway Routing Protocol (EIGRP) or Open Shortest Path First (OSPF) Protocol between sites across the Internet. Routing protocols such as EIGRP and OSPF send packets to multicast destination addresses. Multicast packets cannot be encrypted by IPsec. We first tunnel the multicast packet in a GRE header, and then encrypt the GRE packet (because GRE uses unicast addressing). So, the process is simple: Tunnel EIGRP or OSPF in GRE and encrypt the GRE tunnel. The following example shows the configuration.

Create the GRE tunnel interface:

```
SNRS_ROUTER(config)#interface tunnel 0
```

Assign an IP address to the tunnel interface:

```
SNRS_ROUTER(config-if)#ip address 172.16.1.1
```

Define where the packets of this interface will be sourced:

```
255.255.255.0SNRS_ROUTER(config-if)#tunnel source 172.30.1.2
```

Define where this tunnel goes:

```
255.255.255.0SNRS_ROUTER(config-if)#tunnel destination 172.30.2.2 255.255.255.0
```

Attach the crypto map that will encrypt GRE:

```
SNRS_ROUTER(config-if)#crypto map SNRS-MAP
SNRS_ROUTER(config-if)#no shutdown
SNRS_ROUTER(config-if)#exit
```

This access list defines the GRE traffic that travels between this router and the other side. Match this access list in crypto map SNRS-MAP (not shown here):

```
SNRS_ROUTER(config)#ip access-list 101 permit gre host 172.30.1.2 host 172.30.2.2
```

Notice in the example that the access list only specifies GRE. By specifying GRE as the protocol to be encrypted, you encrypt anything that is traveling inside of the GRE tunnel.

Configuring a DMVPN

DMVPN takes advantage of mGRE, NHRP, and IPsec profiles. The advantages you gain with DMVPN over traditional hub-and-spoke VPN is that the hub configuration is greatly simplified, the encryption initiation is automated, and you can have dynamically addressed spokes. An added benefit is the ability for a spoke to create a tunnel directly to another spoke, which is not a capability of traditional hub-and-spoke VPNs.

How NHRP Is Used

As mentioned previously, DMVPN uses NHRP and mGRE. So what is NHRP? What is mGRE? Let's begin with NHRP. NHRP is a client and server protocol where the hub is the server and the spokes are the clients. The hub keeps an NHRP database of the public interface addresses of the each spoke, even if those addresses are dynamic. Each spoke registers its real IP address when it boots, and it queries the NHRP database for the real addresses of the destination spokes when it needs to build direct tunnels to them. NHRP is used in DMVPN to enable a source DMVPN gateway (a router at the spoke site) to determine the physical interface IP address of the DMVPN gateway (router) closest to a destination where encrypted traffic needs to be sent. When the source spoke site learns the destination address of the router at the destination, it then dynamically builds a GRE/IPsec tunnel directly to the DMVPN gateway, or router, closest to the destination. After the tunnel is built, the source DMVPN gateway forwards user traffic directly to the destination site.

How mGRE Is Used

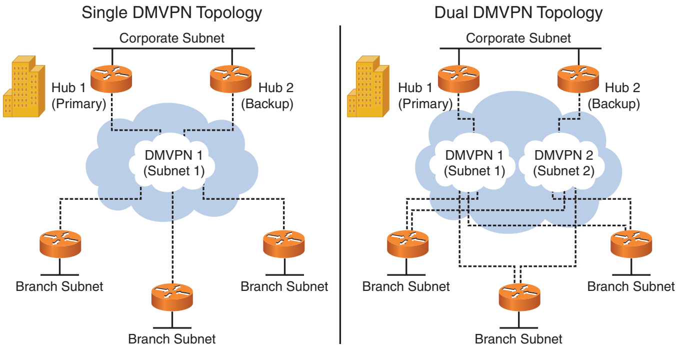
mGRE allows a tunnel interface to support multiple GRE/IPsec tunnels. mGRE is required regardless of how many endpoints you have. Without mGRE, you can establish only a single point-to-point GRE/IPsec tunnel. The hub, for example, will aggregate multiple GRE/IPsec tunnels from the spokes.

Two topologies are recommended:

- Dual hub - dual DMVPN cloud
- Dual hub - single DMVPN cloud

Figure 4-7 shows these two topologies.

FIGURE 4-7
DMVPN Topologies



To configure DMVPN on the hub, follow these steps:

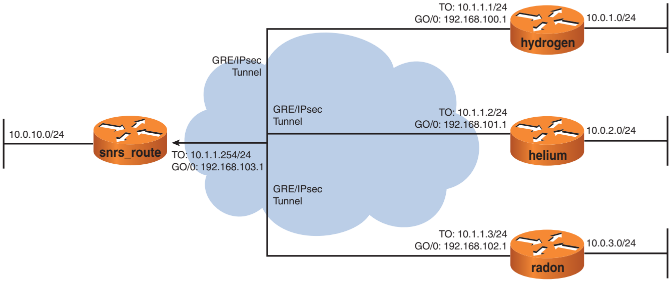
1. Configure the IKE policy (including authentication).
2. Configure the IPsec transform set.
3. Configure the IPsec profile.
4. Configure common parameters on the mGRE tunnel interface.
5. Configure the routing protocol for site-to-site reachability.

To configure DMVPN on the spoke routers, follow these steps:

1. Configure the IKE policy (including authentication).
2. Configure the IPsec transform set.
3. Configure the crypto map or IPsec profile.
4. Configure common parameters on the GRE tunnel interface.
5. Configure the routing protocol for site-to-site reachability.

The following sections detail the configuration and the process of a DMVPN connecting in the network, as shown in Figure 4-8.

FIGURE 4-8
DMVPN Example Topology



In this network, there are three spoke routers: hydrogen, helium, and radon. Platinum is the hub router. When one of the spokes comes up, or is configured for DMVPN, it will initiate an IPsec-protected GRE tunnel to the hub router (platinum). The hub router is configured, and an NHRP next-hop server and the spoke routers are configured as NHRP next-hop clients. Each client sends an NHRP registration message informing the hub of the physical IP address on interface g0/0 and the GRE interface address on T0. The hub caches this information and sends a reply.

It's not a requirement, but the hub can be configured to add broadcast/multicast mappings for each spoke. This will allow routing protocols to be sent over the tunnel. It is important that routing protocols be sent over the tunnel is because without this aspect one spoke site would not be able to find another spoke site. Here is the process of actually routing traffic from one spoke to another (refer back to Figure 4-8):

1. A user on the 10.0.1.0/24 subnet of the hydrogen router wants to send traffic to a user on the 10.0.3.0/24 subnet connected to the radon router.
2. The hydrogen router finds a route to the 10.0.3.0 network with the next hop of 10.1.1.3, which is the GRE interface of radon.

The issue with finding the next hop of 10.1.1.3 is that its not a globally routable IP address. Because it's not globally routable, hydrogen needs to find a globally routable next-hop address; in this case, it's the physical interface address that's assigned to interface g0/0 (192.168.102.1) on the radon router.

3. To find the next-hop of the ***radon*** router, the ***hydrogen*** router checks its local NHRP cache but there is no mapping for the next-hop address 10.1.1.3.
4. The hydrogen router then sends an NHRP resolution request to the SNRS_ROUTER. The SNRS_ROUTER is the NHS, so hydrogen is requesting the physical interface address at which 10.1.1.3 can be reached.
5. The SNRS_ROUTER looks in its NHRP cache and finds that 192.168.102.1 is mapped to the next-hop address of 10.1.1.3.
6. The SNRS_ROUTER sends the mapping to the hydrogen router in an NHRP resolution reply.
7. The hydrogen router enters the mapping into its local NHRP cache.
8. Now that the hydrogen router has all the information it needs, it builds an IPsec-protected GRE tunnel directly to the radon router.
9. After this tunnel has been built, traffic can be sent directly between the hydrogen router and the radon router.

In this example, the SNRS_ROUTER acted as only a hub and was used to provide next-hop resolution to the hydrogen router.

Configuring the IKE Policy and IPsec Transform Set

The configuration shown here in the example is similar to the configuration of a site-to-site VPN tunnel. The big difference here is the pre-shared key. Notice that it does not specify the actual address of the peer; instead, it uses the wildcard 0.0.0.0 0.0.0.0. This is necessary because we don't know the IP address of the multiple sites that will be connecting. A more secure alternative is to use digital certificates:

```
SNRS_ROUTER(config)#crypto isakmp policy 10
SNRS_ROUTER(config-isakmp)#hash md5
SNRS_ROUTER(config-isakmp)#encryption 3des
SNRS_ROUTER(config-isakmp)#authentication pre-share
SNRS_ROUTER(config)#crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0
SNRS_ROUTER(config)#crypto ipsec transform-set MYTRANS esp-3des
```

Configuring the IPsec Profile

Using IPsec profiles allows certain portions of the IPsec configurations to be placed under a label that can be called in different portions of the configuration (in which case, you don't have to specify the peer in the crypto map or the access list defining what gets encrypted). In the following example, an IPsec profile is created, and the transform set is defined:

```
SNRS_ROUTER(config)#crypto ipsec profile DMVPN
SNRS_ROUTER(ipsec-profile)#set transform-set MYTRANS
```

Configuring the Tunnel Interface on the Hub in a Spoke-to-Spoke DMVPN

In the hub configuration, the tunnel interface is defined as tunnel 0, and the IP address, 10.1.1.254, is applied to the interface:

```
SNRS_ROUTER(config)#interface Tunnel 0
SNRS_ROUTER(config-if)#ip address 10.1.1.254 255.255.255.0
```

The **ip nhrp authentication** string command is used to configure an authentication string for DMVPN gateways. This string must be identical for all gateways in a particular DMVPN:

```
SNRS_ROUTER(config-if)#ip nhrp authentication cisco123
```

The **ip nhrp map multicast dynamic** command allows the hub gateway to automatically add spoke gateways to its NHRP multicast mapping table. This is necessary to ensure that dynamic routing protocols (that send multicast traffic) work correctly over the DMVPN:

```
SNRS_ROUTER(config-if)#ip nhrp map multicast dynamic
```

The **ip nhrp network-id** *number* command is used to configure the NHRP network ID. The NHRP network ID uniquely identifies the NHRP network. This must be configured identically on all gateways in the same DMVPN:

```
SNRS_ROUTER(config-if)#ip nhrp network-id 99
```

The **tunnel source** command is used to configure the source IP address of the tunnel. Unlike normal GRE tunnels, there is no **tunnel-destination** command needed for DMVPNs:

```
SNRS_ROUTER(config-if)#tunnel source FastEthernet 0/1
```

The **tunnel key** *key-number* command is used to configure an ID key for the tunnel interface. This ID key, the authentication string, and network ID are used to map NHRP and tunnel packets to this particular mGRE interface. The tunnel key must be identical on all gateways for a particular DMVPN:

```
SNRS_ROUTER(config-if)#tunnel key 999
```

The tunnel mode is then configured as mGRE using the **tunnel mode gre multipoint command**:

```
SNRS_ROUTER(config-if)#tunnel mode gre multipoint
```

The **tunnel protection** command associates the IPsec profile with the tunnel interface:

```
SNRS_ROUTER(config-if)#tunnel protection ipsec profile DMVPN
```

Configuring the Routing Protocols

For DMVPN, you can use either EIGRP or OSPF to route traffic. The Cisco preferred method is EIGRP, so we will not discuss OSPF.

Enable the EIGRP routing protocol:

```
SNRS_ROUTER(config-if)#exit
SNRS_ROUTER(config)#router eigrp 1
SNRS_ROUTER(config-router)#network 10.0.0.0
SNRS_ROUTER(config-router)#no auto-summary
```

For spoke-to-spoke DMVPN networks, a unique challenge exists because the spokes cannot directly exchange information with one another, even though they are on the same logical subnet. This means that the hub router needs to advertise subnets from the other spokes on the same subnet. The IP routing rule known as split horizon prevents the hub from doing this:

```
SNRS_ROUTER(config—router)#interface tunnel 0
SNRS_ROUTER(config-if)#no ip split-horizon eigrp 1
```

In addition, the advertised route must contain the original next hop as learned by the hub router. The **ip next-hop-self** command was added to allow the hub to forward the IP address of the next hop when advertising the route to another spoke:

```
SNRS_ROUTER(config-if)#no ip next-hop-self eigrp 1
```

Configuring the Spoke in a Spoke-to-Spoke DMVPN

The following configuration enables the spoke for DMVPN. For the sake of brevity, only the tunnel interface configuration is explained. It is assumed that the IKE, IPsec transform set, and IPsec profile are already configured (identical to the hub).

Configure common parameters on the GRE tunnel interface:

```
hydrogen(config)#interface Tunnel 0
hydrogen(config-if)#ip address 10.1.1.1 255.255.255.0
```

The **ip nhrp authentication** string command is used to configure an authentication string for DMVPN gateways. This string must be identical for all gateways in a particular DMVPN:

```
hydrogen(config-if)#ip nhrp authentication cisco123
```

This command is used to build an NHRP mapping of the hub gateway's outside physical interface IP address to its GRE tunnel interface IP address. In this example, the IP address configured on the hub gateway's physical interface is 192.168.103.1, and the IP address configured on its mGRE tunnel interface is 10.1.1.254:

```
hydrogen(config-if)#ip nhrp map 10.1.1.254 192.168.103.1
```

The **ip nhrp nhs** *nhs-gre-interface-ip-address* command is used to specify the tunnel interface IP address of the NHS. The NHS is the SNRS_ROUTER:

```
hydrogen (config-if)#ip nhrp nhs 172.16.16.1
hydrogen (config-if)#ip nhrp map multicast 172.30.1.2
```

The **ip nhrp** *network-id number* command is used to configure the NHRP network ID. The NHRP network ID uniquely identifies the NHRP network. This must be configured identically on all gateways in the same DMVPN:

```
hydrogen(config-if)#ip nhrp network-id 99
```

Define the tunnel source:

```
hydrogen(config-if)#tunnel source FastEthernet 0/1
```

Define the tunnel key:

```
hydrogen(config-if)#tunnel key 999
```

Define the tunnel mode:

```
hydrogen(config-if)#tunnel mode gre multipoint
```

Apply the tunnel protection:

```
hydrogen(config-if)#tunnel protection ipsec profile DMVPN
```

Configure EIGRP parameters:

```
hydrogen(config)#router eigrp 1
hydrogen(config-router)#network 10.0.0.0
hydrogen(config-router)#no auto-summary
hydrogen(config-router)#eigrp stub connected
hydrogen(config-router)#interface tunnel 0
hydrogen(config-if)#no ip next-hop-self eigrp
hydrogen(config-if)#no ip split-horizon eigrp 1
```

Verifying DMVPN

You can use the following commands to verify DMVPN:

- **show crypto map**
- **show ip nhrp**
- **show crypto isakmp sa**
- **show crypto ipsec sa**

For a more detailed discussion about DMVPN, refer to *The Complete Cisco VPN Configuration Guide*, by Richard Deal (Cisco Press).

Configuring Cisco IOS SSL VPN (WebVPN)

The Cisco IOS supports the configuration of SSL, or WebVPNs. WebVPNs are useful for users who need access to internal websites, file shares, and some legacy applications, but who are not appropriate for the power user group. With WebVPN, all traffic is tunneled through an SSL connection that is established with a web browser. Because SSL is native to the browser, no additional VPN software has to be managed. This creates a secure web portal that can be used to check email, access company intranet pages, and access files on servers.

Overview of Cisco IOS SSL VPN (WebVPN)

WebVPN offers the following modes of SSL VPN:

- **Clientless**: This mode is useful for accessing most of the content that you would access in a web browser, such as websites, databases, and online tools that use web interfaces. Access into the network is achieved via a web browser.

- **Thin client (port-forwarding Java applet)**: Thin-client mode uses a Java applet to enable port forwarding, which lets you access TCP-based applications such as Post Office Protocol Version 3 (POP3), Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol (IMAP), Telnet, and Secure Shell (SSH).
- **Tunnel mode**: Full tunnel client mode uses a dynamically downloaded Cisco switched virtual circuit (SVC) for WebVPN. This client is a lightweight, centrally configured, and easy-to-support SSL VPN tunneling client that provides network layer access to almost any application.

WebVPN Configuration Tasks

The SSL VPN feature (also known as WebVPN) provides support, in Cisco IOS Software, for remote user access to enterprise networks from anywhere on the Internet. Remote access is provided through an 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 HTTPS browser support. SSL VPN delivers three modes of SSL VPN access: clientless, thin client, and full-tunnel client support.

The configuration of WebVPN consists of the following:

- **Configuring the WebVPN gateway**: This part of the configuration is required. In this step, the address, hostname, and trustpoint for the WebVPN connection are configured.
- **Configuring the WebVPN context**: This part of the configuration is also required. This step includes the customization of the user interface such options as login messages and URLs that will appear on the WebVPN portal.
- **Configuring an SSL VPN policy group**: This part is also required.

The following configuration enables WebVPN on the SNRS_ROUTER. The configuration begins with the WebVPN gateway part.

Enter the WebVPN gateway configuration mode:

```
SNRS_ROUTER(config)#webvpn gateway SNRS-GW
```

Configure the hostname for the WebVPN gateway:

```
SNRS_ROUTER(config-webvpn-gateway)#hostname GW-1
```

The following command enables the WebVPN gateway to listen to port 80 and redirect it port 443:

```
SNRS_ROUTER(config-webvpn-gateway)#http-redirect
```

The following command defines the proxy IP address for the WebVPN gateway:

```
SNRS_ROUTER(config-webvpn-gateway)#ip address 10.0.1.3 port 443
```

The next command defines the encryption algorithm for the WebVPN connection:

```
SNRS_ROUTER(config-webvpn-gateway)#ssl encryption aes-sha
```

The next command configures the SSL trustpoint if self-signed certificates are going to be used:

```
SNRS_ROUTER(config-webvpn-gateway)#ssl trustpoint SNRS-CA
```

The following command puts the WebVPN gateway “in service.” You cannot enable the WenVPN gateway until the proxy IP address has been defined:

```
SNRS_ROUTER(config-webvpn-gateway)#inservice
```

Now the WebVPN context will be configured. In this part, the user interface customization takes place. This includes the look and feel of the WebVPN interface and the login message that users see when they connect and any URLs that will appear on the WebVPN page.

The next command enters the WebVPN context configuration mode. Cisco recommends that you name the context the domain name or virtual hostname:

```
SNRS_ROUTER(config)#webvpn context SSLVPN
```

The next command tells the WebVPN how to authenticate users. This points to a method list that is configured elsewhere in global configuration mode. If this command is not configured, the WebVPN will use global AAA parameters. The VPN-ACS method list should point to a TACACS+ server and from there could tie into Active Directory:

```
SNRS_ROUTER(config-webvpn-context)#aaa authentication list VPN-ACS
```

The next command creates a policy group called SSL-Policy. This command places you into the policy configuration mode, where you can set up special policies for the WebVPN users:

```
SNRS_ROUTER(config-webvpn-context)#policy-group SSL-Policy
```

The next command ties the policy group SSL-Policy to the WebVPN:

```
SNRS_ROUTER(webvpn-group-policy)#default-group-policy SSL-Policy
```

The following command associates the WebVPN gateway with the WebVPN context:

```
SNRS_ROUTER(webvpn-group-policy)#gateway SNRS-GW
```

Place the context in service. This will not work until the context is associates with a gateway:

```
SNRS_ROUTER(config-webvpn-gateway)#inservice
```

Enter the context configuration mode:

```
SNRS_ROUTER(config)#webvpn context SSLVPN
```

Set a login message for the context:

```
SNRS_ROUTER(config-webvpn-context)#login-message "Please enter your username and password"
```

Set the title for the WebVPN page:

```
SNRS_ROUTER(config-webvpn-context)#title "CiscoPress SNRS WebVPN Page"
```

Set the title color:

```
SNRS_ROUTER(config-webvpn-context)#title-color darkseagreen
```

Set the logo image:

```
SNRS_ROUTER(config-webvpn-context)#logo file flash:/cisco.gif
```

Define the maximum number of users:

```
SNRS_ROUTER(config-webvpn-context)#max-users 300
```

Configure the color of the secondary title bars on the login and portal pages of an SSL VPN:

```
SNRS_ROUTER(config-webvpn-context)#secondary-color darkgreen
SNRS_ROUTER(config-webvpn-context)#secondary-text-color white
```

The next configuration section defines the URLs that will be seen in the WebVPN user interface and how those will appear:

```
SNRS_ROUTER(config)#webvpn context SSLVPN
SNRS_ROUTER(config-webvpn-context)#url-list "Internal"
SNRS_ROUTER(config-webvpn-url)#heading "FastLinks"
SNRS_ROUTER(config-webvpn-url)#url-text "My HomePage" url-value home.ciscopress.com
SNRS_ROUTER(config-webvpn-url)#url-text "EMAIL" url-value email.ciscopress.com
```

The policy group is a container that defines the presentation of the portal and the permissions for resources that are configured for a group of remote users. Entering the **policy group** command places the router in WebVPN group policy configuration mode. Remember that previously you tied this policy to the WebVPN context configuration by configuring the **default-group-policy** command:

```
SNRS_ROUTER(config)#webvpn context SSLVPN
```

Enter the policy group configuration mode:

```
SNRS_ROUTER(config-webvpn-context)#policy group SSL-policy
```

Define the banner:

```
SNRS_ROUTER(config-webvpn-group)#banner "Login Successful"
```

Define NBNS name servers to be found in the list NBNS-SERVERS:

```
SNRS_ROUTER(config-webvpn-group)#nbns-list NBNS-SERVERS
```

Define the idle timeout:

```
SNRS_ROUTER(config-webvpn-group)#timeout idle 1800
```

Define the session timeout:

```
SNRS_ROUTER(config-webvpn-group)#timeout session 36000
```

Define the URL list:

```
SNRS_ROUTER(config-webvpn-group)#url-list Internal
```

Define the port-forwarding list:

```
SNRS_ROUTER(config-webvpn-group)#port-forward Portlist
```

Enter the WebVPN context:

```
SNRS_ROUTER(config)#webvpn context SSLVPN
```

Create the port-forward list:

```
SNRS_ROUTER(config-webvpn-context)#port-forward Portlist
SNRS_ROUTER(config-webvpn-port-fwd)#local-port 30020 remote-server mail.ciscopress.com remote-port 25 description "SMTP"
SNRS_ROUTER(config-webvpn-port-fwd)#local-port 30021 remote-server mail.ciscopress.com remote-port 110 description "POP3"
SNRS_ROUTER(config-webvpn-port-fwd)#local-port 30022 remote-server mail.ciscopress.com remote-port 143 description "IMAP"
SNRS_ROUTER(config-webvpn-port-fwd)#exit
SNRS_ROUTER(config-webvpn-context)#policy group SSL-policy
SNRS_ROUTER(config-webvpn-group)#port-forward Portlist
SNRS_ROUTER(config)#webvpn context SSLVPN
```

Create the NBNS_SERVERS list:

```
SNRS_ROUTER(config-webvpn-context)#nbns-list NBNS-SERVERS
SNRS_ROUTER(config-webvpn-nbnslist)#nbns-server 172.16.1.1 master
SNRS_ROUTER(config-webvpn-nbnslist)#nbns-server 172.16.2.2 timeout 10 retries
SNRS_ROUTER(config)#webvpn context SSLVPN
SNRS_ROUTER(config-webvpn-context)#policy group SSL-policy
SNRS_ROUTER(config-webvpn-group)#nbns-list NBNS-SERVERS
```

Define the functions allowed for the WebVPN connection:

```
SNRS_ROUTER(config-webvpn-group)#functions file-access
SNRS_ROUTER(config-webvpn-group)#functions file-browse
SNRS_ROUTER(config-webvpn-group)#functions file-entry
```

Configuring Cisco Easy VPN Remote Access

Cisco Easy VPN configuration is primarily performed on the Easy VPN server. The following devices can act as an Easy VPN server:

- Cisco IOS router
- Cisco ASA 5500 series appliance
- PIX 500 series appliance

The following devices can act as an EASY VPN client:

- Cisco 806 broadband router, Cisco 826 ADSL router, Cisco 827 ADSL router, Cisco 828 ADSL router, Cisco 831 Ethernet broadband router, Cisco 836 ADSL over ISDN broadband router, and Cisco 837 ADSL broadband router
- Cisco 1700 series modular access routers
- Cisco 1800 series fixed-configuration routers
- Cisco 1812 integrated services router
- Cisco 2600 series multiservice platforms
- Cisco 3620, 3640, 3660 multiservice platform

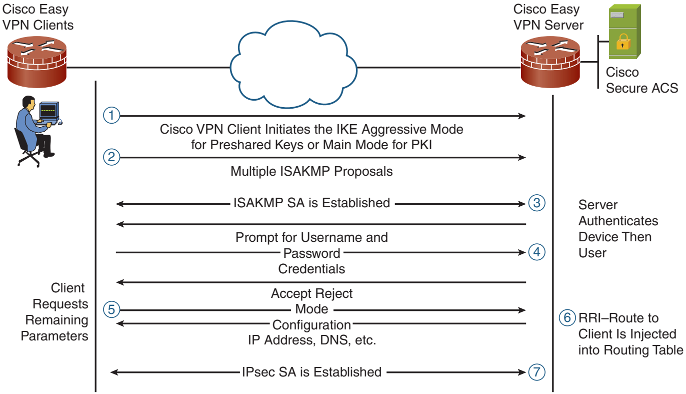
- Cisco 7100 series VPN routers
- Cisco 7200 series routers
- Cisco 7500 series routers
- Cisco PIX 500 series SAs
- Cisco VPN 3000 series concentrators

For the SNRS requirements, you only need to be familiar with the IOS routers as an Easy VPN server.

Easy VPN Operation

When using the Cisco VPN client, the tunnel is established when the user clicks connect. When this happens, IKE Phase 1 is initiated. The VPN client sends numerous IKE proposals in order of the security. When a proposal is agreed upon, the devices perform hardware authentication using pre-shared keys or digital certificates. The next step is often called IKE Phase 1.5. In this phase, the user is authenticated. The user sees a popup window with fields to enter the username and password. Once authenticated, the client then requests mode config. Mode config is when the Easy VPN server pushes down parameters to the user. These parameters include an IP address, split-tunneling list, Domain Name System (DNS), Windows Internet Naming Service (WINS), and any other parameters that are needed by the user. After mode config completes, IKE Phase 2 is negotiated. After the VPN SAs have been established, the connection is up and traffic can pass. In addition, with the reverse route injection (RRI), a host route is injected into the routing process, and devices on the internal network can now find their way back to the VPN client. Figure 4-9 shows this process.

FIGURE 4-9
The Easy VPN Process


Cisco Easy VPN Clients
Cisco Easy VPN Server
Cisco Secure ACS
1
Cisco VPN Client Initiates the IKE Aggressive Mode for Preshared Keys or Main Mode for PKI
2
Multiple ISAKMP Proposals
ISAKMP SA is Established
3
Server Authenticates Device Then User
Prompt for Username and Password
4
Credentials
Accept Reject
Client Requests Remaining Parameters
5
Mode Configuration
IP Address, DNS, etc.
6
RRI–Route to Client Is Injected into Routing Table
IPsec SA is Established
7

The Easy VPN remote can operate in three modes:

- **Client mode**: Where Port Address Translation (PAT) is used. In this mode, the client is assigned an IP address from a pool that is configured on the Easy VPN server. The client is seen as one device to the Easy VPN server, even if the client is a Cisco router with multiple hosts connected behind it. All traffic is port address translated to the address assigned from the pool.
- **Network extension mode**: Where PAT is not used. In this mode, the network and each host on the network connected behind the client can be reached directly. This is similar to a site-to-site VPN, but in this case the VPN tunnel must be established by the Easy VPN client.

- **Network extension plus mode**: Where the client can request an IP address via mode config. This can be used if the client wants to appear as a specific address. This can help to facilitate policy configurations on the server side.

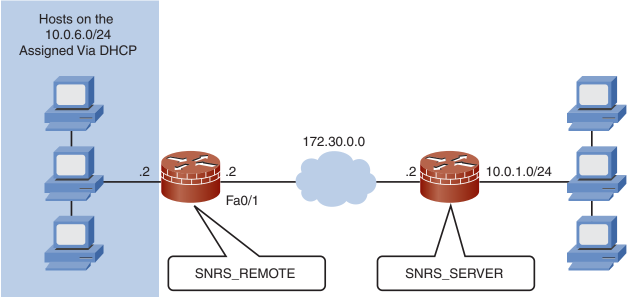
Configuration Tasks for Cisco Easy VPN Remote for Access Routers

To configure Easy VPN Remote, follow these steps:

1. Configure the DHCP server pool.
2. Configure the Cisco Easy VPN Remote client profile.
3. Configure the group and key.
4. Define the peer.
5. Configure the mode of operation (client, network extension).
6. Configure manual or automatic tunnel control.
7. Assign the Cisco Easy VPN Remote client profile to the interfaces.
8. Verify the Cisco Easy VPN configuration.

The following configuration enables the SNRS_REMOTE shown in Figure 4-10 as an Easy VPN client.

FIGURE 4-10
Easy VPN Example



Define a DHCP pool that can assign addresses to devices connected to the inside interface of the remote router:

```
SNRS_REMOTE(config)#ip dhcp pool Local-Pool
SNRS_REMOTE(dhcp-config)#network 10.0.6.0 255.255.255.0
SNRS_REMOTE(dhcp-config)#default-router 10.0.6.2
SNRS_REMOTE(dhcp-config)#exit
SNRS_REMOTE(config)#ip dhcp excluded-address 10.0.6.2
```

NOTE

Remember that static IP addresses work, too. The DHCP pool is completely optional.

Configure the Easy VPN client profile:

```
SNRS_REMOTE(config)#crypto ipsec client ezvpn SNRS_REMOTE-Client
SNRS_REMOTE(config-crypto-ezvpn)#group SNRS_REMOTE key VPNKEY
SNRS_REMOTE(config-crypto-ezvpn)#peer 172.30.1.2
SNRS_REMOTE(config-crypto-ezvpn)#mode client
SNRS_REMOTE(config-crypto-ezvpn)#connect auto
SNRS_REMOTE(config-crypto-ezvpn)#end
```

Assign Easy VPN Remote to an interface:

```
SNRS_REMOTE(config)#interface FastEthernet 0/1
SNRS_REMOTE(config-if)#crypto ipsec client ezvpn SNRS_REMOTE-Client
SNRS_REMOTE(config-if)#exit
SNRS_REMOTE(config)#interface FastEthernet 0/0
SNRS_REMOTE(config-if)#crypto ipsec client ezvpn SNRS_REMOTE-Client inside
SNRS_REMOTE(config-if)#end
```

This optional configuration saves the username and password locally. It is not the best security measure, but does automate the connection a little more:

```
SNRS_REMOTE(config)#crypto ipsec client ezvpn SNRS_REMOTE-Client
SNRS_REMOTE(config-crypto-ezvpn)#username cisco password 0 cisco
SNRS_REMOTE(config-crypto-ezvpn)#end
```

Use the following command to initiate the Easy VPN connection to the Easy VPN server:

```
SNRS_REMOTE#crypto ipsec client ezvpn xauth
Enter Username and Password: vpnusers
Password: ********
```

Use the following command to verify the Easy VPN connection:

```
SNRS_REMOTE#show crypto ipsec client ezvpn
Easy VPN Remote Phase: 6

Tunnel name : SNRS_REMOTE-Client
Inside interface list: FastEthernet0/0
Outside interface: FastEthernet0/1
Current State: IPSEC_ACTIVE
Last Event: SOCKET_UP
```

```
Address: 10.0.1.100
Mask: 255.255.255.255
Default Domain: cisco.com
Save Password: Allowed
Current EzVPN Peer: 172.30.1.2
```

Configuring Cisco Easy VPN Server

To enable Easy VPN Server functionality, follow these steps:

1. (Optional) Create an IP address pool for connecting clients.
2. Enable group policy lookup via AAA.
3. Create an ISAKMP policy for remote VPN client access.
4. Define a group policy for mode configuration push.
5. Apply mode configuration and XAUTH.
6. Enable RRI for the client.
7. Enable IKE Dead Peer Detection (DPD).
8. Configure XAUTH.
9. (Optional) Enable the XAUTH Save Password feature.

The following configuration enables the SNRS_SERVER shown in Figure 4-10 as an Easy VPN server.

Define the local pool of addresses that will be assigned to users when they VPN in. If you are in client mode, you have to be able to assign an address; otherwise, the tunnel will not come up:

```
SNRS_SERVER(config)#ip local pool Remote-Pool 10.0.1.100 10.0.1.150
```

Enable group policy lookup:

```
SNRS_SERVER(config)#aaa new-model
SNRS_SERVER(config)#aaa authentication login vpn-users local
SNRS_SERVER(config)#aaa authorization network vpn-group local
SNRS_SERVER(config)#username cisco password 0 cisco
```

Create the group policy for the SNRS_REMOTE that enables the mode configure policy push:

```
SNRS_SERVER(config)#crypto isakmp client configuration group SNRS_REMOTE-Client
SNRS_SERVER(config-isakmp-group)#key VPNKEY
SNRS_SERVER(config-isakmp-group)#dns 10.0.1.13 10.0.1.14
SNRS_SERVER(config-isakmp-group)#wins 10.0.1.13 10.0.1.14
SNRS_SERVER(config-isakmp-group)#domain cisco.com
SNRS_SERVER(config-isakmp-group)#pool Remote-Pool
SNRS_SERVER(config-isakmp-group)#save-password
```

Create the ISAKMP policy:

```
SNRS_SERVER(config)#crypto isakmp enable
SNRS_SERVER(config)#crypto isakmp policy 10
SNRS_SERVER(config-isakmp)#authentication pre-share
SNRS_SERVER(config-isakmp)#encryption aes
SNRS_SERVER(config-isakmp)#group 2
SNRS_SERVER(config-isakmp)#end
```

Define the transform set:

```
SNRS_SERVER(config)#crypto ipsec transform-set VPNTRANSFORM esp-aes esp-sha-hmac
SNRS_SERVER(cfg-crypto-trans)#end
```

Because you don’t know the IP address of the remote peer, you can’t use the **set peer** command in the crypto map. Instead, use a **dynamic crypto map** to attach a transform set to a static crypto map. You can only assign a static crypto map to an interface. This configuration also enables RRI, which places a host rout to the VPN client into the IP routing protocol. In addition, RRI is configured using the **reverse route** command. This enables the injection of a host route into the routing process when users are connected to the VPN. If you don’t have this, devices on the network might not be able to find their way back the VPN client:

```
SNRS_SERVER(config)#crypto dynamic-map Dynamic-Map 10
SNRS_SERVER(config-crypto-map)#set transform-set VPNTRANSFORM
SNRS_SERVER(config-crypto-map)#reverse-route
SNRS_SERVER(config-crypto-map)#end
```

Next, create a static crypto map so that you have something to attach to the interface. Tie the dynamic crypto map to it and the group lookup authorization and XAUTH authentication commands. The **address respond** suboption on the crypto map allows the server to respond to mode configuration requests:

```
SNRS_SERVER(config)#crypto map ClientMap client configuration address respond
SNRS_SERVER(config)#crypto map ClientMap isakmp authorization list vpn-group
SNRS_SERVER(config)#crypto map ClientMap client authentication list vpn-users
SNRS_SERVER(config)#crypto map ClientMap 65535 ipsec-isakmp dynamic Dynamic-Map
```

Attach the crypto map to the interface:

```
SNRS_SERVER(config)#interface ethernet0/1
SNRS_SERVER(config-if)#crypto map ClientMap
SNRS_SERVER(config-if)#end
```

Enable ISAKMP DPD for the remote tunnel. This is used to ensure that the peer is still there, or active. If the keepalive times out, the SA is removed.

The following command instructs the router to send keepalives every 20 seconds, and every 10 seconds between retries:

```
SNRS_SERVER(config)#crypto isakmp keepalive 20 10
```

Enable AAA login authentication, an XAUTH timeout, and enable XAUTH for the group:

```
SNRS_SERVER(config)#aaa authentication login VPNUSERS local
SNRS_SERVER(config)#crypto isakmp xauth timeout 20
SNRS_SERVER(config)#crypto map CLIENTMAP client authentication list VPNUSERS
```

In the following option command, you enable the saving of passwords on the remote:

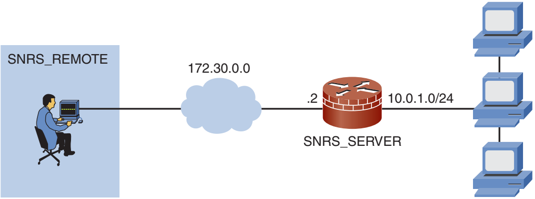
```
SNRS_SERVER(config)#crypto isakmp client configuration group SNRS_REMOTE-Client
SNRS_SERVER(config-isakmp-group)#save-password
```

Configuring Cisco VPN Client 4.x

Figure 4-11 shows an alternative topology for Easy VPN. In this scenario, the SNRS_REMOTE is a computer running the Cisco VPN client.

The examples you see here are all done on Windows XP.

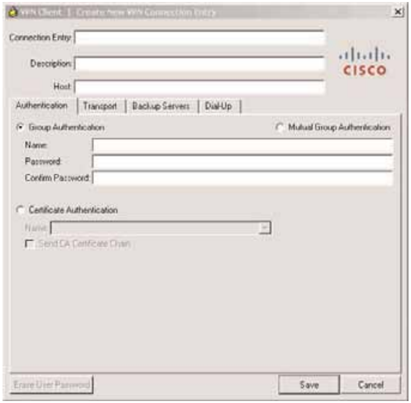
FIGURE 4-11
Easy VPN Example Using the Cisco VPN Client



After completing the install, the minimum needed to connect into the SNRS_SERVER is to add a connection entry. To do so, follow these steps:

1. Launch Cisco VPN Client. It should be under Program Files > Cisco Systems > Cisco VPN Client.
2. Click the New button in the VPN Client application. This opens the window shown in Figure 4-12.

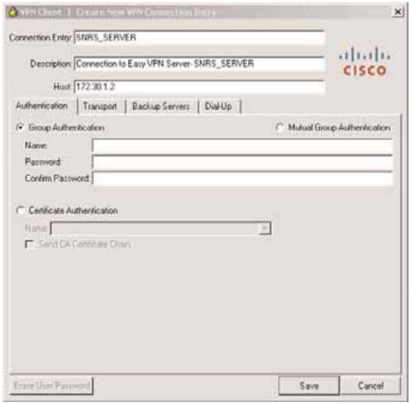
FIGURE 4-12
Cisco VPN Client:
New Entry Window



● Group Authentication
○ Mutual Group Authentication
○ Certificate Authentication
□ Send CA Certificate Chain

3. Enter the name of the connection, an optional description, and the Easy VPN server that you want to connect to, as shown in Figure 4-13.

FIGURE 4-13
Cisco VPN Client: Defining the Entry



VPN Client | Create New VPN Connection Entry

Connection Entry: SNRS_SERVER

Description: Connection to Easy VPN Server- SNRS_SERVER

Host: 172.30.1.2

● Group Authentication ○ Mutual Group Authentication

Name:

Password:

Confirm Password:

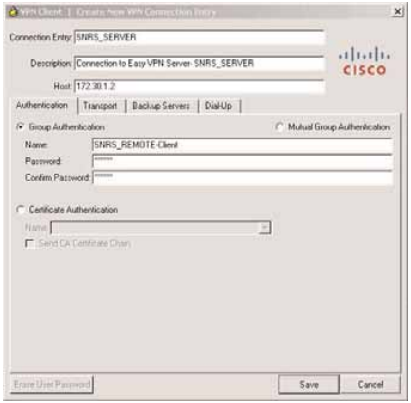
○ Certificate Authentication

Name:

□ Send CA Certificate Chain

4. Enter the VPN group that you will be authenticating into and the password for the group. This is the pre-shared key for the connection. Figure 4-14 shows the final entry.

FIGURE 4-14
Cisco VPN Client:
Final Connection Entry



Connection Entry: SNRS_SERVER
Description: Connection to Easy VPN Server- SNRS_SERVER
Host: 172.30.1.2

● Group Authentication ○ Mutual Group Authentication
Name: SNRS_REMOTE-Client
Password: *******
Confirm Password: *******

○ Certificate Authentication
Name:
□ Send CA Certificate Chain

You should certainly be familiar with other configurations, too, such as adding backup servers and changing transport options. You can also explore the directory of the VPN client, specifically the \Profiles\ directory. In this directory, you will find a PCF file for each connection entry that you have created. You can view this file in any text editor. In this file, you will find many of the parameters that you set up in the GUI. You can change many of these options in the text editor, and they will take effect in the GUI the next time it is opened. One other item of interest is the customization guide for the VPN client found at the following site:

http://cisco.com/en/US/docs/security/vpn_client/cisco_vpn_client/vpn_client46/administration/guide/vcAch7.html

CHAPTER 5

Adaptive Threat Defense

The Cisco Adaptive Threat Defense (ATD) initiative, announced in 2005, was designed to increase a network's ability to identify, prevent, and adapt to security threats. The ATD consists of the following three major components:

- Anti-X defenses
- Application security
- Network control and containment

The first element, Anti-X, includes traffic-inspection services that identify attacks from viruses, spyware, and URL filtering.

The core technologies of ATD include firewall, IPS (intrusion prevention system), anomaly detection, and DDoS (distributed DoS) mitigation. The second element, application security, provides protection through the use of application-level access controls, inspection, and enforcement of appropriate policies.

The third element is network control and containment, which allows sophisticated auditing and correlation capabilities to control and help protect any networked element or service such as VoIP with active management and mitigation capabilities.

This chapter discusses the core technologies of ATD, including the following:

- The classic IOS firewall
- The zone-based firewall
- The authentication proxy
- The IPS

Configuring Cisco IOS Firewall

If you plan to use your Cisco router as an IOS Firewall, you enable a number of features that make your router a robust firewall solution. The following components are found as part of the Cisco IOS Firewall feature set:

- **Classic Firewall**

  The IOS Classic Firewall enables firewall capabilities that filter traffic not only at Layer 3 and 4, but also at the application layer. This lets the firewall filter traffic based on session information of well-known TCP and UDP applications such as FTP and HTTP. The IOS Firewall keeps a state table (which is a mapping of session state), protects against DoS attacks, and dynamically opens and closes ports. Without this functionality, all traffic will be blocked on all ports except that which has been explicitly permitted with an access list.

- **Authentication Proxy**

  The Authentication Proxy feature enables you to authenticate protocol traffic that is passing through the firewall to an AAA server, such as Cisco Secure ACS. By authenticating users, you can tie their network authorizations to a proxy ACL that is downloaded to the firewall and applied to the user while authenticated. In my experience working with a major telecommunications provider, I have often found in workgroups that have more than 100 administrators (all in the same group, but all having very different privileges) that the use of Authentication Proxy makes it easy to allow Internet access to senior administrators and block access for junior administrators.

- **Cisco IOS IPS**

  The Cisco IOS Intrusion Prevention System (IPS) feature enables the firewall to act as an inline IPS sensor. Using a signature database, the IPS determines whether the traffic matches known attack signatures and then gives the option to alarm, drop, reset the connection, deny traffic from the attacker for a period of time, or deny the connection for a period of time.

- **Standard and extended ACLs**

  Standard and extended ACLs enable the most basic packet filtering for the router. This allows filtering of network layer addresses and Layer 4 ports and protocols. ACLs give both inbound and outbound filtering capabilities.

- **Dynamic lock-and-key ACLs**

  Lock-and-key provides traffic filtering with the ability to allow temporary access through the firewall for certain users. The user must authenticate to the firewall first, and then the firewall temporarily opens ports and protocols for the user. After a period of time, the entries are timed out and removed. This is what gives this method its dynamic nature. It's common to see an SSH or Telnet connection first to authenticate to the firewall, then perhaps access to an internal web server can be allowed by the firewall.

- **Reflexive ACLs**

  Reflexive ACLs enable an access list to allow traffic into the network only if it was seen leaving the network first. That means that the allowed traffic into the network is usually valid replies to outbound requests. This feature would not be configured if you are using Context Based Access Control (CBAC).

- **TCP Intercept**

  TCP Intercept is a feature that allows protection against SYN flood attacks. This is another feature that you wouldn't use if CBAC is enabled.

- **PAM**

  PAM is port-to-application mapping. The Cisco IOS already has a number of ports mapped to specific applications. For example, port 80 is mapped to the HTTP application, and port 21 is mapped to the FTP application. PAM enables you to customize ports that the IOS recognizes certain applications on so that it can then use firewall functionality to apply protection to those applications. For example, you could use the Classic Firewall to filter web traffic for one of your servers, and it will watch session information for port 80. If for some reason you use a nonstandard port such as port 8080 for the web server, you must indicate to the firewall that port 8080 traffic is web traffic.

- **NAT**

  Network Address Translation is another capability of the Cisco IOS Firewall. It enables the use of RFC 1918 addressing on internal networks and the translation capability for that address space when it "speaks" to devices out on the Internet where RFC 1918 addressing can't be used. NAT also conserves address space and hides the internal topology.

- **Security server support such as RADIUS, TACACS+, and Kerberos and user authentication and authorization**

  This security server support is needed for the AAA process to communicate with an external AAA server such as Cisco Secure ACS. This is also the ability to use AAA to authenticate users to the router for management or through the router for protocol traffic.

Configuring Cisco IOS Classic Firewall

The Cisco IOS Classis Firewall, once called Context Based Access Control, or CBAC, enables stateful control of traffic through the router. The actual terminology for the firewall process is Cisco IOS SPI, or stateful packet inspection. This provides traffic filtering and the ability to inspect the traffic and send alerts and audit trails to remote syslog servers. When we say that the traffic is inspected, what is really happening is that the traffic is being identified and monitored as it makes a connection through the router. Common attributes of TCP sessions, for example, are cached and used to verify return packets for legitimacy. Understand that TCP is not the only protocol that can be identified and inspected. The supported protocols are as follows:

- TCP
- UDP
- RPC
- FTP

- TFTP
- UNIX **r** commands (such as **rlogin**, **rexec**, and **rsh**)
- SMTP
- HTTP (Java blocking)
- SQL*Net
- RTSP (such as Real Networks)
- H.323 (such as NetMeeting, ProShare)
- CUseeMe
- Other multimedia

Although each of these protocols functionally differs from the others, the firewall function is for the most part the same. The Classic Firewall will inspect these protocols for malicious activity, watch the traffic that leaves, and allow valid replies and send alerts and audit trails on an individual protocol basis.

Cisco IOS Classic Firewall Configuration Tasks

To enable the Classic Firewall feature, follow these steps:

1. Identify traffic that will be allowed out through the firewall.

   You want to be sure that the ACLs permit legitimate traffic from the secure network to the unsecure network.

2. Configure ACLs to block traffic from the unsecure network.

3. Create inspection rules.

   Here you set global timeouts and thresholds for the dynamic connections that the firewall is going to open. This will also include the configuration of DoS protection and the definition of generic TCP and UDP inspection. In the inspection rule, you also configure application layer inspection for more granular inspection.

4. Configure IP packet-fragmentation options.
5. Configure application firewall settings for HTTP and IM applications.
6. Apply the rule inbound to the inside interface or outbound to the outside interface.
7. Configure audit trails and logging.

The following configuration enables a simple two-interface IOS Classic Firewall, as shown in Figure 5-1.

Define the access list to deny traffic sourced from an untrusted network into the trusted network:

```
SNRS_ROUTER(config)#access-list 101 deny ip any any
```

Create an inspection rule called SNRSFW that will inspect TCP, UDP, and ICMP traffic as it leaves the trusted network:

```
SNRS_ROUTER(config)#ip inspect name SNRSFW tcp
SNRS_ROUTER(config)#ip inspect name SNRSFW udp
SNRS_ROUTER(config)#ip inspect name SNRSFW icmp
```

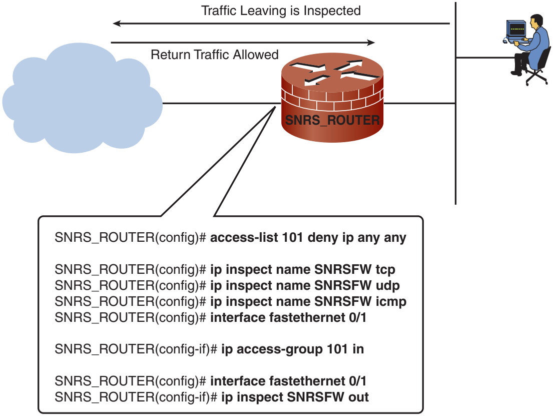
Apply the ACL to the outside interface:

```
SNRS_ROUTER(config)#interface fastEthernet 0/1
SNRS_ROUTER(config-if)#ip access-group 101 in
```

Apply the inspect rule to the interface that traffic is leaving:

```
SNRS_ROUTER(config)#interface fastEthernet 0/1
SNRS_ROUTER(config-if)#ip inspect SNRSFW out
```

FIGURE 5-1
Cisco IOS Firewall Example



```
SNRS_ROUTER(config)# access-list 101 deny ip any any

SNRS_ROUTER(config)# ip inspect name SNRSFW tcp
SNRS_ROUTER(config)# ip inspect name SNRSFW udp
SNRS_ROUTER(config)# ip inspect name SNRSFW icmp
SNRS_ROUTER(config)# interface fastethernet 0/1

SNRS_ROUTER(config-if)# ip access-group 101 in

SNRS_ROUTER(config)# interface fastethernet 0/1
SNRS_ROUTER(config-if)# ip inspect SNRSFW out
```

You can also add application firewall functions to the Cisco IOS router. This enhances the filtering capability and lets you look for IM applications and HTTP tunneling. To create the policy, use the **appfw policy-name** <*policy-name*> command, and define which application you will be filtering with the **application** [**http** | **im** {**aol**|**yahoo**|**msb**}] command.

The following configuration enables the application firewall called HTTP-Policy for HTTP.

Once in the **cfg-appfw-policy** command mode, you can enable protocol specific inspections. In the following configuration, the **strict-http** command instructs the firewall to allow noncompliant traffic but to send an alarm. The **content-length** command allows traffic through the firewall based on the message size, but sends an alarm if it's outside of the profile. The **content-type-verification**, **max-header-length**, and **max-uri-length** all set policies for alarming when traffic does not meet the configuration specifications. The **port-misuse** command specifies that peer-to-peer (P2P), IM, and tunneling applications that are seen in an HTTP message are going to be inspected, allowed, and alarmed on when they are seen. The **request-method** commands say that HTTP requests will be inspected for RFC compliance and for Remote Copy Protocol (RCP) extension methods. The **transfer-encoding** command says that all the transfer encoding types will be inspected:

```
router(config)#appfw policy-name HTTP-Policy
router(cfg-appfw-policy)#application http
router(cfg-appfw-policy-http)#strict-http action allow alarm
router(cfg-appfw-policy-http)#content-length maximum 1 action allow alarm
router(cfg-appfw-policy-http)#content-type-verification match-req-rsp action allow alarm
router(cfg-appfw-policy-http)#max-header-length request 1 response 1 action allow alarm
router(cfg-appfw-policy-http)#max-uri-length 1 action allow alarm
router(cfg-appfw-policy-http)#port-misuse default action allow alarm
router(cfg-appfw-policy-http)#request-method rfc default action allow alarm
router(cfg-appfw-policy-http)#request-method extension default action allow alarm
router(cfg-appfw-policy-http)#transfer-encoding type default action allow alarm
```

To apply the preceding application inspection policy, use the following configuration. This enables HTTP inspection on interface Fast Ethernet0/0:

```
router(config)#ip inspect name FW appfw HTTP-Policy
router(config)#ip inspect name FW http
router(config)#interface FastEthernet0/0
router(config-if)#ip inspect firewall in
```

You can use the following commands to verify the Cisco IOS Classic Firewall:

- **show ip access-lists** Displays the contents of all current IP ACLs
- **show ip inspect name** *inspection-name* Shows a particular configured inspection rule
- **show ip inspect config** Shows the complete Cisco IOS Classic Firewall inspection configuration
- **show ip inspect interfaces** Shows interface configuration with regard to applied inspection rules and ACLs
- **show ip inspect session** [**detail**] Shows existing sessions that are currently being tracked and inspected by Cisco IOS Classic Firewall
- **show ip inspect all** Shows all Cisco IOS classic firewall configuration and all existing sessions that are currently being tracked and inspected by Cisco IOS Classic Firewall

Now that you've spent all this time configuring Classic Firewall, you can remove it by just entering the command **no ip inspect**. This removes all configuration related to the firewall and resets all timeouts and thresholds.

Configuring Cisco IOS Zoned-Based Policy Firewall

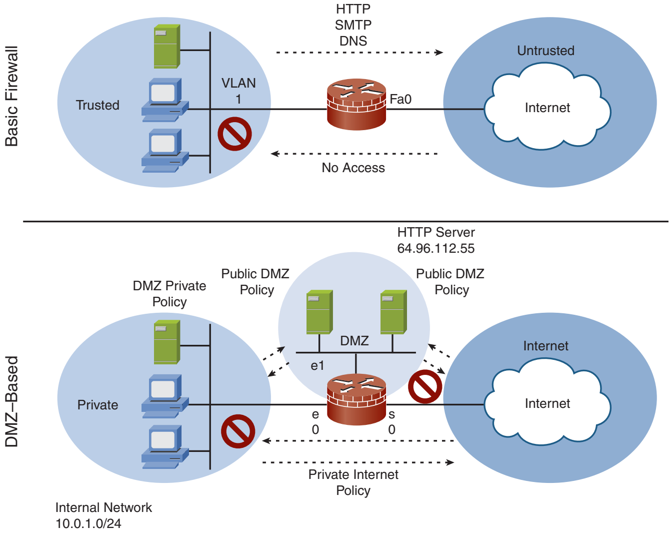
Cisco IOS Release 12.4(6)T introduced a new model for the Cisco IOS Firewall feature set. This new model has the following characteristics:

- Policies are applied to traffic moving between zones, not interfaces. Think of a zone as an area of trust. You could have an Internet zone, for instance, and a trusted zone. You will use this concept of a zone to determine how traffic flows and how firewall policy is defined.
- These policies are configured with what's known as the Cisco Policy Language (CPL).
- The default policy for interzone traffic is deny all. This means that traffic from one zone to another is blocked by default.
- Policies are subnet- and host-specific policies similar to an access list.

The benefit gained from this configuration model is that firewall policies can be more clearly understood because only policy from zone A to zone B impacts traffic and there is no interference between multiple inspection policies or ACLs. For example, you might configure a policy that says web traffic from the trusted zone to the Internet zone is allowed.

Figure 5-2 shows two topologies: a basic firewall and a demilitarized zone (DMZ)-based firewall.

FIGURE 5-2
Zone-Based Firewall Deployments



Zones

You should know a few zone rules:

- If two interfaces are not in zones, traffic flows freely between them.
- If one interface *is* in a zone, and another interface *is not* in a zone, traffic may *never* flow between them.
- If two interfaces are in two different zones, traffic will not flow between the interfaces until a policy is defined to allow the traffic.

Security Zone Firewall Policies

Security zone firewall policies have the following characteristics:

- The CPL framework, which is based on and similar to the Modular QoS CLI (MQC) framework, is used to configure them.
- The CPL consists of three constructs: class maps, which are used to match traffic; policy maps, which are used to associate that traffic with an action; and parameter maps, which are used to specify operating parameters for the classification and action application.

Zone firewall policies can be Layer 3, Layer 4, or Layer 7 policy types. A Layer 3/Layer 4 policy is a top-level policy that is attached to the zone pair (trusted zone – Internet zone). You use a class map to match protocol/ACLs selections that you configure, and then apply high-level actions such as drop, inspect, URL filter, and deep inspection. This is good for basic inspection. To get more granular with your policy, you use an application policy, or Layer 7 policy. These are optional and typically are applied to control the finer details of an application (for instance, HTTP and SMTP). Layer 7 policies are contained in a Layer 3/Layer 4 policy and cannot be directly attached to a target.

Configuring a Cisco IOS Zoned-Based Policy Firewall

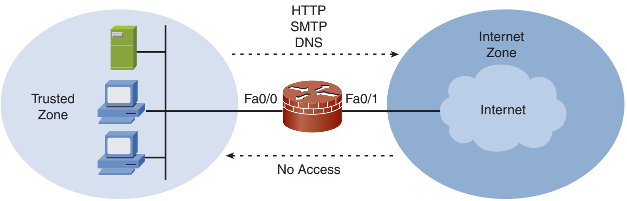
To configure a zone-based policy firewall, follow these steps:

1. Identify interfaces that share the same function security and group them into the same security zones.
2. Determine the required traffic flow between zones in both directions.
3. Set up zones.
4. Set up zone pairs for any policy other than deny all.
5. Define class maps to describe traffic between zones.
6. Associate class maps with policy maps to define actions applied to specific policies.
7. Assign policy maps to zone pairs.

Configuration Example

The network shown in Figure 5-3 is configured for zone-based firewall functionality. As shown in the figure, no inbound access is allowed from the Internet zone to the trusted zone. Traffic from the trusted zone to the Internet zone is allowed, but is restricted to HTTP, SMTP, and DNS.

FIGURE 5-3
Zone-Based Firewall Example Network



First, a class map is created that defines the list of the services in the firewall policy. This example includes HTTP, SMTP, and DNS:

```
class-map type inspect match-any snrsprotocols
 match protocol http
 match protocol smtp
 match protocol dns

!
```

Next an action is applied to the traffic matched in the class map snrsprotocols (inspect = stateful inspection):

```
!
policy-map type inspect snrsfwpolicy
 class type inspect snrsprotocols
  inspect
!
```

Next, create the “trusted” and “internet” zones.

```
!
zone security trusted
zone security internet
!
```

Now assign interface f0/0 to the “private” zone. If you compare this to the figure, it is clear as to what is happening here:

```
!

interface fastethernet 0/0
 zone-member security trusted
!
```

Next, assign the fa0/1 interface to the “internet” zone. Again, if you compare this to the figure, it is clear as to what is happening here:

```
interface fastethernet 0/1
 zone-member security internet
!
```

To apply the inspection, create the zone pair that will define the flow of traffic and the policy to be applied:

```
zone-pair security priv-to-internet source trusted destination internet
 service-policy type inspect snrsfwpolicy
```

Verifying Cisco IOS Zone-Based Policy Firewall

To verify the zone-based policy firewall, use the commands **show zone security**, **show zone-pair security**, **show policy-map type inspect**, **show policy-map type**, **inspect zone-pair sessions**, and **show class-map type inspect**.

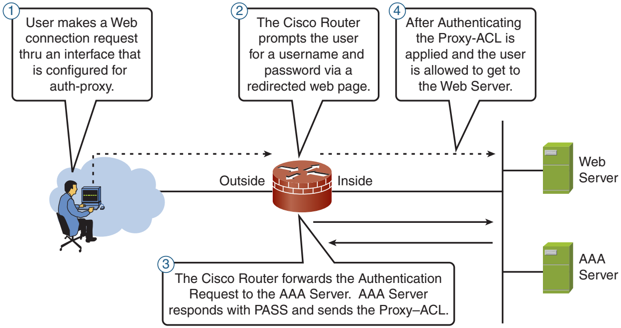
For a more detailed discussion on zone-based policy firewalls, refer to the digital Short Cut *Deploying Zone-Based Firewalls*, by Ivan Pepelnjak, published by Cisco Press at the following site:

http://safari.ciscopress.com/1587053101

Configuring Cisco IOS Firewall Authentication Proxy

The Cisco IOS Authentication Proxy enables HTTP, HTTPS, FTP, and Telnet authentication. This provides dynamic, per-user authentication and authorization control. After a user has authenticated, all authorized traffic can pass. Figure 5-4 shows how Authentication Proxy works for inbound connections.

FIGURE 5-4
Inbound Authentication Proxy


1
User makes a Web connection request thru an interface that is configured for auth-proxy.
2
The Cisco Router prompts the user for a username and password via a redirected web page.
4
After Authenticating the Proxy-ACL is applied and the user is allowed to get to the Web Server.
Outside
Inside
Web Server
AAA Server
3
The Cisco Router forwards the Authentication Request to the AAA Server. AAA Server responds with PASS and sends the Proxy–ACL.

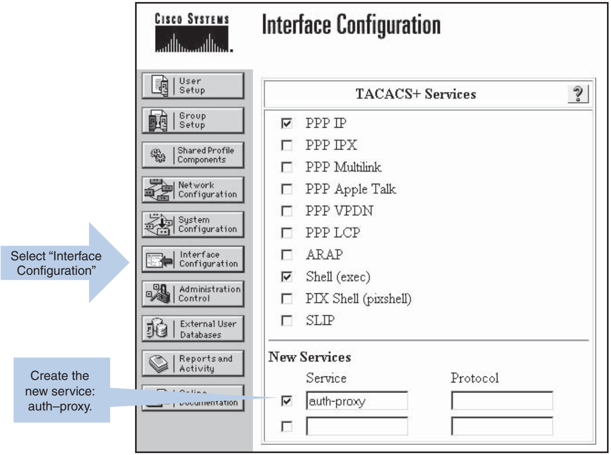
AAA Server Configuration

The Auth-Proxy service must be configured on the AAA server. To do so, follow these steps:

1. Log in to ACS.
2. Enable the Auth-Proxy service in Interface Configuration.
3. Build the proxy ACL in Group Setup.

Figure 5-5 shows the Auth-Proxy service being enabled on the AAA server.

FIGURE 5-5
Enable the Auth-Proxy Service in Cisco Secure ACS



Interface Configuration

TACACS+ Services

☑ PPP IP
☐ PPP IPX
☐ PPP Multilink
☐ PPP Apple Talk
☐ PPP VPDN
☐ PPP LCP
☐ ARAP
☑ Shell (exec)
☐ PIX Shell (pixshell)
☐ SLIP

New Services

	Service	Protocol
☑	auth-proxy	
☐		

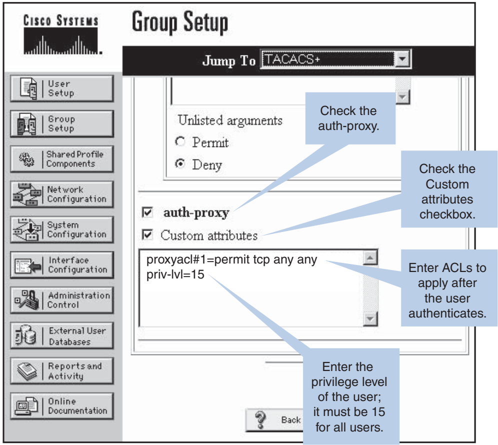
Select “Interface Configuration”

Create the new service: auth–proxy.

The next ACS configuration task is to build the proxy ACL in the Group Setup (see Figure 5-6). It's important to enable the privilege level of 15 because this level is required to apply an access list. When you build the proxy ACL, complete the configuration just as you would any extended access list on a Cisco router, with the exception of the line number. Each line number should increment in the list, as follows:

```
proxy-acl#1=permit tcp any any eq www
proxy-acl#2=permit tcp any any eq https
proxy-acl#3=permit tcp any any eq telnet
proxy-acl#4=permit tcp any any eq ftp
priv-lvl=15
```

FIGURE 5-6
Define the Proxy ACL



Group Setup

Jump To TACACS+

Unlisted arguments
○ Permit
● Deny

☑ auth-proxy
☑ Custom attributes

```
proxyacl#1=permit tcp any any
priv-lvl=15
```

Check the auth-proxy.

Check the Custom attributes checkbox.

Enter ACLs to apply after the user authenticates.

Enter the privilege level of the user; it must be 15 for all users.

Cisco IOS Firewall Authentication Proxy Configuration Task List

To configure Authentication Proxy on the Cisco IOS router, follow these steps.

1. Enable AAA.
2. Define a TACACS+ server and its key.
3. Allow AAA traffic to the router.
4. Enable the router HTTP or HTTPS server for AAA.
5. Set global timers
6. Apply Authentication Proxy rules with ACLs.

Cisco IOS Firewall Authentication Proxy Configuration on a Cisco Router

To configure Authentication Proxy, follow these steps:

1. Enable the AAA process on the router. Use the **aaa authorization auth-proxy** command to authorize traffic via the Authentication Proxy AAA server:

```
SNRS_ROUTER(config)#aaa new-model
SNRS_ROUTER(config)#aaa authentication login default group tacacs
SNRS_ROUTER(config)#aaa authorization auth-proxy default group tacacs+
SNRS_ROUTER(config)#aaa accounting auth-proxy default start-stop group tacacs+
```

2. Define the TACACS+ server IP and secret key that is used for message encryption between the router and the AAA server:

```
SNRS_ROUTER(config)#tacacs-server host 10.0.6.12
SNRS_ROUTER(config)#tacacs-server key cisco
```

3. Define Authentication Proxy parameters for cache timeout and the Authentication Proxy rule for HTTP. This rule is called SNRS-Proxy. This will later be applied to the interface where you want Authentication Proxy to be performed:

```
SNRS_ROUTER(config)#ip auth-proxy auth-cache-time 60
SNRS_ROUTER(config)#ip auth-proxy name SNRS-Proxy http
```

4. Enable the HTTP server so that you can present the authentication page to users. Without this, Authentication Proxy fails:

```
SNRS_ROUTER(config)#ip http server
```

5. Enable AAA authentication for the HTTP server. Because the HTTP server is enabled, you specify that anyone using it must be authenticated with this configuration:

```
SNRS_ROUTER(config)#ip http authentication aaa
```

6. Define the TACACS+ or RADIUS protocols that should be permitted on the interface that communicates with the AAA server. You could explicitly allow only the traffic that you want to pass, but that’s not the point here. The point is that the only thing that talks to the interface is TACACS+ *or* RADIUS, and maybe ICMP, until after the users authenticate with Authentication Proxy. After they authenticate, proxy ACLs will be applied at the router to let their web traffic pass:

```
SNRS_ROUTER(config)#access-list 102 permit tcp host 10.0.6.12 eq tacacs host 10.0.6.2  — -use this if you are
 using TACACS+
SNRS_ROUTER(config)#access-list 102 permit udp host 10.0.6.12 eq 1645 host 10.0.6.2    — -use this if you are
 using RADIUS
SNRS_ROUTER(config)#access-list 102 permit udp host 10.0.6.12 eq 1646 host 10.0.6.2    — -use this if you are
 using RADIUS
SNRS_ROUTER(config)#access-list 102 permit udp host 10.0.6.12 eq 1812 host 10.0.6.2    — -use this if you are
 using RADIUS
```

```
SNRS_ROUTER(config)#access-list 102 permit udp host 10.0.6.12 eq 1813 host 10.0.6.2    — -use this if you are
 using RADIUS
SNRS_ROUTER(config)#access-list 102 deny tcp any any                 — -deny TCP until after authenticated
SNRS_ROUTER(config)#access-list 102 deny   udp any any               — -deny UDP until after authenticated
SNRS_ROUTER(config)#access-list 102 permit ip any any                — -permit non UDP/TCP traffic such as ICMP
SNRS_ROUTER(config)#access-list 105 deny   tcp any any               — -deny TCP until after authenticated
SNRS_ROUTER(config)#access-list 105 deny   udp any any               — deny UDP until after authenticated
SNRS_ROUTER(config)#access-list 105 permit ip any any                — permit non UDP/TCP traffic such as ICMP
```

On the outside interface, define the Authentication Proxy rule that should be used:

```
SNRS_ROUTER(config)#interface Serial0
SNRS_ROUTER(config-if)#ip address 172.30.6.2 255.255.255.0
SNRS_ROUTER(config-if)#ip access-group 105 in
SNRS_ROUTER(config-if)#ip auth-proxy SNRS-Proxy
SNRS_ROUTER(config)#interface Ethernet0
SNRS_ROUTER(config-if)#ip address 10.0.6.2 255.255.255.0
SNRS_ROUTER(config-if)#ip access-group 102 in
```

Test and Verify

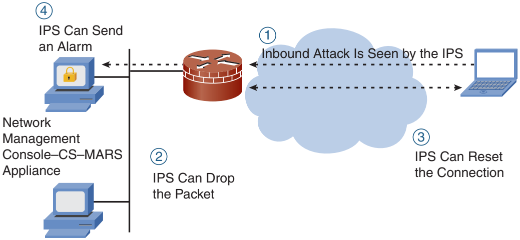
To verify the Authentication Proxy configuration, use the following commands:

- **show ip auth-proxy cache**
- **show ip auth-proxy configuration**
- **show ip auth-proxy watch list**

Configuring Cisco IOS IPS

The Cisco IOS IPS feature uses the underlying routing infrastructure to provide inline deep packet inspection that is software based. The IPS process provides signature-based packet scanning using the same signatures as the Cisco IPS appliances. The signature update process is now done without needing to update the Cisco IOS, and if you want to, you can configure your own custom signatures. The signatures have a variety of actions that they can take, and the signatures are scanned in parallel. Figure 5-7 shows the IPS process.

FIGURE 5-7
Cisco IOS IPS Process


4
IPS Can Send an Alarm
1
Inbound Attack Is Seen by the IPS
Network Management Console–CS–MARS Appliance
2
IPS Can Drop the Packet
3
IPS Can Reset the Connection

There are 135 signatures that are built in to the IOS, and more can be enabled by loading a Signature Definition File (SDF). These SDF files can be downloaded from Cisco.com, stored in flash, and loaded into memory on a router running the IOS IPS. There are three pretuned signature files: attack-drop.sdf, 128MB.sdf, and 256MB.sdf. These signatures are built based on Signature Micro-Engines (SME). The SME categorizes signatures; for example, if you want to look for a string of text in a TCP session, you write the signature using the STRING.TCP micro-engine. The following SMEs are supported in Cisco IOS IPS:

- ATOMIC.L3.IP
- ATOMIC.ICMP
- ATOMIC.IPOPTIONS

- ATOMIC.UDP
- ATOMIC.TCP
- SERVICE.DNS
- SERVICE.RPC
- SERVICE.SMTP
- SERVICE.HTTP
- SERVICE.FTP
- STRING.TCP
- STRING.UDP
- STRING.ICMP
- MULTI-STRING
- OTHER

For more information about the SMEs, refer to the following site:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cg/hsec_c/part15/sec_ips.htm#wp1154863

Cisco IOS Firewall IPS Configuration Tasks

To configure IPS on the router, follow these steps:

1. Specify the location of the SDF.
2. Create an IPS rule.

3. Attach a policy to a signature (optional).
4. Apply the IPS rule at an interface.
5. Configure logging via syslog or SDEE.
6. Verify the configuration.

Configuring the Cisco IOS IPS

The following sample configuration defines the SDF file to use, the action to take if the IPS software fails, and the IPS policy to apply to an interface:

```
! Define the location of the signature definition file that will be loaded.

SNRS_ROUTER(config)#ip ips sdf location flash:128MB.sdf

! Instruct the firewall to stop passing traffic if the IPS processes fail.

SNRS_ROUTER(config)#ip ips fail closed

! Create the IPS rule.

SNRS_ROUTER(config)#ip ips name SNRS-IPS

! Enter the interface that you want IPS to be enabled on.

SNRS_ROUTER(config)#interface FastEthernet0/1
```

```
! The following command enables the Virtual Fragment ReAssembly (VFR) functioun on the interface. Certain attacks
 take advantage of the time and memory required to reassemble packets that are fragmented by sending very high
 numbers of fragmented packets. The VFR feature allows the router to prevent these buffer overflow attacks.

SNRS_ROUTER(config-if)#ip virtual-reassembly

! Enable the IPS policy named SNRS-IPS in the inbound direction on the interface.  Packets leaving this interface
 will not be filtered by the IPS.

SNRS_ROUTER(config-if)#ip ips SNRS-IPS in

! Exit back to privilege mode.
SNRS_ROUTER(config-if)#end

! Once back in privilege mode the following message should appear indicating that the IPS is enabled and the sig-
 natures are loaded.

*Jan 28 01:18:04.664: %IPS-6-SDF_LOAD_SUCCESS: SDF loaded successfully from flash:128MB.sdf
. . . messages ommited ..........
*Jan 28 01:18:30.452: %IPS-6-ENGINE_BUILDING: ATOMIC.L3.IP - 5 signatures - 15 of 15 engines
```

You could also merge SDF files using the following configuration.

The following configuration merges the 128MB.sdf signature file with the snrs-signatures.sdf file:

```
SNRS_ROUTER#copy flash:128MB.sdf ips-sdf
SNRS_ROUTER#copy ips-sdf flash:snrs-signatures.sdf
SNRS_ROUTER#configure terminal
SNRS_ROUTER(config)#ip ips sdf location flash:snrs-signatures.sdf
```

Configure Logging via Syslog or SDEE

To monitor IPS, you have two options.

- SDEE (a pull mechanism using an SSL connection)
- Syslog (a push mechanism, with message sent in clear text)

To use syslog, enter **ip ips notify log**. To user SDEE, enter **ip ips notify sdee**. Your method of logging may vary. SDEE is a secure method that pulls the logs from the IPS device to the monitoring station, and syslog is a push mechanism where the IPS device sends the syslog message when the event occurs. The syslog message is sent in clear text.

Verifying IPS Configuration

You can verify the configuration by entering the **show ip ips configuration** command, as shown here:

```
SNRS_ROUTER#show ip ips configuration
Configured SDF Locations:
 flash:128MB.sdf
Builtin signatures are enabled but not loaded
Last successful SDF load time: 12:10:43 CST Oct 30 2006
IPS fail closed is disabled
Fastpath ips is enabled
Quick run mode is enabled
Event notification through syslog is enabled
Event notification through SDEE is enabled
Total Active Signatures: 303
Total Inactive Signatures: 0
Signature 50000:0 disable
Signature 50000:1 disable
Signature 50000:2 disable
IPS Rule Configuration
 IPS name SNRS-IPS
Interface Configuration
 Interface FastEthernet0/1
  Inbound IPS rule is SNRS-IPS
  Outgoing IPS rule is not set
```

To verify the signatures, enter **show ip ips signatures**:

```
SNRS_ROUTER#show ip ips signatures
Builtin signatures are configured
Signatures were last loaded from flash:128MB.sdf
Cisco SDF release version 128MB.sdf v2
Trend SDF release version V0.0

*=Marked for Deletion  Action=(A)larm,rop,(R)eset   Trait=AlarmTraits
MH=MinHits             AI=AlarmInterval                CT=ChokeThreshold
TI=ThrottleInterval    AT=AlarmThrottle                FA=FlipAddr
WF=WantFrag
Signature Micro-Engine: OTHER (4 sigs)
 SigID:SubID On Action  Sev Trait     MH    AI     CT     TI AT FA WF Version
 ——————. — ——— —— ——.  ——. ——. ——. ——. — — — ———.
 1203:0      Y   A     HIGH     0      0     0    30     15 FA  N  N 2.2.1.5
 1202:0      Y   A     HIGH     0      0     0   100     15 FA  N  N 2.2.1.5
 3050:0      Y   A     HIGH     0      0     0   100     15 FA  N    1.0
 1201:0      Y   A     HIGH     0      0     0    30     15 FA  N  N 2.2.1.5
Signature Micro-Engine: STRING.ICMP (1 sigs)
```

```
 SigID:SubID On Action  Sev Trait     MH    AI    CT    TI AT FA WF Version
 —————. — ——— —— ——.  ——. ——. ——. ——. — — — ———.
 2156:0      Y   A      MED     0      0     0   100    15 FA  N     S54
Signature Micro-Engine: STRING.UDP (16 sigs)
 SigID:SubID On Action  Sev Trait     MH    AI    CT    TI AT FA WF Version
 —————. — ——— —— ——.  ——. ——. ——. ——. — — — ———.
 11209:0     Y   A     INFO     0      0     0   100    15 FA  N     S139
 11208:0     Y   A     INFO     0      0     0   100    15 FA  N     S139
 4608:2      Y   A     HIGH     0      1     0   100    15 FA  N     S30b
```

To verify the configuration of IPS on an interface, enter the **show ip ips interfaces** command:

```
SNRS_ROUTER#show ip ips interfaces
Interface Configuration
 Interface FastEthernet0/1
  Inbound IPS rule is SNRS-IPS
  Outgoing IPS rule is not set
```

To view information about SDEE alerts, enter the **show ip sdee alerts** command. To see information about SDEE events, enter the **show ip sdee events** command.

If you want to remove the IOS IPS configuration, you can enter the **clear ip ips configuration** command.