Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide Using the CLI

Release 4.0

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Customer Order Number: N/A, Online only
Text Part Number: OL-16083-01

Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide using the CLI

CONTENTS

About This Guide

This preface describes the objectives and organization of this document and explains how to find additional information on related products and services.

This preface includes the following sections:

Audience

This guide is for network managers who perform any of the following tasks:

- Managing network security
- Installing and configuring firewalls
- Managing default and static routes, and TCP and UDP services

Objectives

This document contains instructions and procedures for configuring the Firewall Services Module (FWSM), a single-width services module supported on the Catalyst 6500 switch and the Cisco 7600 router, using the command-line interface. FWSM protects your network from unauthorized use. This guide does not cover every feature, but describes only the most common configuration scenarios.

You can also configure and monitor the FWSM by using ASDM, a web-based GUI application. ASDM includes configuration wizards to guide you through some common configuration scenarios, and online Help for less common scenarios. For more information, see:

http://www.cisco.com/en/US/products/ps6121/tsd_products_support_series_home.html.

Organization

This document contains the following chapters:

Chapter	Title	Description
Part 1: Getting Started and General Information		
1	Introduction to the Firewall Services Module	Provides a high-level overview of the FWSM.
2	Configuring the Switch for the Firewall Services Module	Describes how to configure the switch for use with the FWSM.
3	Connecting to the Firewall Services Module and Managing the Configuration	Describes how to access the command-line interface and work with the configuration.
4	Configuring Security Contexts	Describes how to use security contexts and enable multiple context mode.
5	Configuring the Firewall Mode	Describes in detail the two operation modes of the FWSM, routed and transparent mode, and how data is handled differently with each mode.
6	Configuring Interface Parameters	Describes how to configure the interface name, security level, and IP address. It also describes how to configure bridge groups for transparent firewall mode interfaces.
7	Configuring Basic Settings	Describes how to configure basic settings that are typically required for a functioning configuration.
8	Configuring IP Routing and DHCP Services	Describes how to configure IP routing and DHCP.
9	Configuring Multicast Routing	Describes how to configure multicast routing.
10	Configuring IPv6	Describes how to enable and configure IPv6.
11	Configuring AAA Servers and the Local Database	Describes how to configure AAA servers and the local database.
12	Identifying Traffic with Access Lists	Describes how to identify traffic with access lists.
13	Configuring Failover	Describes the failover feature, which lets you configure two FWSMs so that one will take over operation if the other one fails.
Part 2: Configuring the Security Policy		
14	Permitting or Denying Network Access	Describes how to control network access through the FWSM using access lists.
15	Configuring NAT	Describes how address translation is performed.
16	Applying AAA for Network Access	Describes how to enable AAA for network access.
17	Applying Filtering Services	Describes ways to filter web traffic to reduce security risks or prevent inappropriate use.
18	Configuring ARP Inspection and Bridging Parameters	Describes how to enable ARP inspection and how to customize bridging operations.

Chapter	Title	Description
19	Using Modular Policy Framework	Describes how to use the Modular Policy Framework to create security policies for TCP, general connection settings, and inspection.
20	Configuring Advanced Connection Features	Describes how to configure connection features.
21	Applying Application Layer Protocol Inspection	Describes how to use and configure application inspection.
Part 3: System Administration		
22	Configuring Management Access	Describes how to access the FWSM for system management through Telnet, SSH, HTTPS, and VPN.
23	Managing Software, Licenses, and Configurations	Describes how to enter license keys and download software and configurations files.
24	Monitoring the Firewall Services Module	Describes how to monitor the FWSM.
25	Troubleshooting the Firewall Services Module	Describes how to troubleshoot the FWSM.
Part 4: Reference		
A	Specifications	Describes the FWSM specifications.
B	Sample Configurations	Describes a number of common ways to implement the FWSM.
C	Using the Command-Line Interface	Describes how to use the CLI to configure the FWSM.
D	Mapping MIBs to CLI Commands	Lists MIB objects and the equivalent CLI commands.
E	Addresses, Protocols, and Ports	Provides a quick reference for IP addresses, protocols, and applications.
	Glossary	Provides a glossary for terms used in this guide.
	Index	Provides an index for this guide.

Document Conventions

The FWSM command syntax descriptions use the following conventions:

Command descriptions use these conventions:

- Braces ({ }) indicate a required choice.
- Square brackets ([ ]) indicate optional elements.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- **Boldface** indicates commands and keywords that are entered literally as shown.
- *Italics* indicate arguments for which you supply values.

Examples use these conventions:

- Examples depict screen displays and the command line in `screen` font.

- Information you need to enter in examples is shown in **boldface screen** font.
- Variables for which you must supply a value are shown in *italic screen* font.
- Examples might include output from different platforms; for example, you might not recognize an interface type in an example because it is not available on your platform. Differences should be minor.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

For information on modes, prompts, and syntax, see Appendix C "Using the Command-Line Interface."

Related Documentation

FWSM documentation is at the following URL:

http://www.cisco.com/en/US/products/hw/modules/ps2706/ps4452/tsd_products_support_model_home.html

ASDM documentation is at the following URL:

http://www.cisco.com/en/US/products/ps6121/tsd_products_support_series_home.html.

For more information, see the following documentation:

- *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*
- *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module System Log Messages*
- *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Installation and Verification Note*
- *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Release Notes*
- *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide using ASDM*
- *Release Notes for Cisco ASDM*
- *Open Source Software Licenses for FWSM*

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.

Quick Start Steps

The following sections describe the minimum configuration required for the FWSM in routed mode or transparent mode:

Routed Firewall Minimum Configuration Steps

To configure the FWSM in routed mode, perform the following steps:

	Task	Description
Step 1	Assigning VLANs to the Firewall Services Module, page 2-2	On the switch, you need to assign VLANs to the FWSM so that the FWSM can send and receive traffic on the switch.
Step 2	(Might be required) Adding Switched Virtual Interfaces to the MSFC, page 2-4	If you want the MSFC to route between VLANs that are assigned to the FWSM, complete this procedure.
Step 3	Connecting to the Firewall Services Module, page 3-1	From the switch CLI, you can session into the FWSM to access the FWSM CLI.
Step 4	(Might be required; multiple context mode only) Enabling or Disabling Multiple Context Mode, page 4-10	If you want to use multiple context mode and your FWSM is not already configured for it, or if you want to change back to single mode, follow this procedure.
Step 5	(Multiple context mode only) Configuring a Security Context, page 4-27	Add a security context.
Step 6	(Multiple context mode only) Changing Between Contexts and the System Execution Space, page 4-31	Because you must configure some settings in the system execution space and some settings within the context, you need to know how to switch between contexts and the system execution space.
Step 7	Configuring Interfaces for Routed Firewall Mode, page 6-2	For each VLAN interface, you must set a name (such as inside or outside), a security level, and an IP address.
Step 8	Configuring a Default Route, page 8-4	Create a default route to an upstream router.

	Task	Description
Step 9	Configure routing using one of these methods: • Configuring a Static Route, page 8-3 • Configuring BGP Stub Routing, page 8-6 • (Single context mode only) Configuring OSPF, page 8-9 • (Single context mode only) Configuring EIGRP, page 8-22 • (Single context mode only) Configuring RIP, page 8-21	In multiple context mode, static routing and stub BGP is the only routing method supported. In single mode, you have a choice of static, stub BGP, RIP, EIGRP, or OSPF.
Step 10	(Might be required) Use one or more of these NAT methods: • Using Dynamic NAT and PAT, page 15-18 • Using Static NAT, page 15-28 • Using Static PAT, page 15-30	Configure NAT if you use private addresses, or want the extra security.
Step 11	Adding an Extended ACE, page 12-7	Before any traffic can go through the FWSM, you must create an access list that permits traffic.
Step 12	Applying an Access List to an Interface, page 14-4	Apply the access list to an interface.

Transparent Firewall Minimum Configuration Steps

To configure the FWSM in transparent mode, perform the following steps:

	Task	Description
Step 1	Assigning VLANs to the Firewall Services Module, page 2-2	On the switch, you need to assign VLANs to the FWSM so that the FWSM can send and receive traffic on the switch.
Step 2	(Might be required) Adding Switched Virtual Interfaces to the MSFC, page 2-4	If you want the MSFC to route between VLANs that are assigned to the FWSM, complete this procedure.
Step 3	Connecting to the Firewall Services Module, page 3-1	From the switch CLI, you can session into the FWSM to access the FWSM CLI.
Step 4	(Might be required; multiple context mode only) Enabling or Disabling Multiple Context Mode, page 4-10	If you want to use multiple context mode and your FWSM is not already configured for it, or if you want to change back to single mode, follow this procedure.
Step 5	(Multiple context mode only) Configuring a Security Context, page 4-27	Add a security context.
Step 6	(Multiple context mode only) Changing Between Contexts and the System Execution Space, page 4-31	Because you must configure some settings in the system execution space and some settings within the context, you need to know how to switch between contexts and the system execution space.
Step 7	Setting Transparent or Routed Firewall Mode, page 5-17	Before you configure any settings, you must set the firewall mode to transparent mode. Changing the mode clears your configuration. In multiple context mode, set the mode in each context.

	Task	Description
Step 8	Configuring Transparent Firewall Interface Parameters, page 6-3	For each VLAN interface, you must set a name (such as inside or outside), a security level, and a bridge group.
Step 9	Assigning an IP Address to a Bridge Group, page 6-5	Assign an IP address to each bridge group.
Step 10	Configuring a Default Route, page 8-4	Create a default route to an upstream router for returning management traffic.
Step 11	Adding an Extended ACE, page 12-7	Before any traffic can go through the FWSM, you must create an access list that permits traffic.
Step 12	Applying an Access List to an Interface, page 14-4	Apply the access list to an interface.

PART 1

Getting Started and General Information

CHAPTER 1

Introduction to the Firewall Services Module

The FWSM is a high-performance, space-saving, stateful firewall module that installs in the Catalyst 6500 series switches and the Cisco 7600 series routers.

Firewalls protect inside networks from unauthorized access by users on an outside network. The firewall can also protect inside networks from each other, for example, by keeping a human resources network separate from a user network. If you have network resources that need to be available to an outside user, such as a web or FTP server, you can place these resources on a separate network behind the firewall, called a *demilitarized zone* (DMZ). The firewall allows limited access to the DMZ, but because the DMZ includes only the public servers, an attack there affects only the servers and does not affect the other inside networks. You can also control when inside users access outside networks (for example, access to the Internet), by allowing only certain addresses out, by requiring authentication or authorization, or by coordinating with an external URL filtering server.

The FWSM includes many advanced features, such as multiple security contexts (similar to virtualized firewalls), transparent (Layer 2) firewall or routed (Layer 3) firewall operation, hundreds of interfaces, and many more features.

When discussing networks connected to a firewall, the *outside* network is in front of the firewall, the *inside* network is protected and behind the firewall, and a *DMZ*, while behind the firewall, allows limited access to outside users. Because the FWSM lets you configure many interfaces with varied security policies, including many inside interfaces, many DMZs, and even many outside interfaces if desired, these terms are used in a general sense only.

This chapter includes the following sections:

New Features

This section lists new features for each maintenance release, and includes the following topics:

New Features in Release 4.0(4)

The following Cisco IOS-integrated features are now officially supported in FWSM:

Feature	Description
PISA integration	**Note** This feature depends on Cisco IOS Release 12.2(18)ZYA or later, and is only available on the Catalyst 6500 switch. The FWSM can leverage the high-performance deep packet inspection of the PISA card so that it can permit or deny traffic based on the application type.
Route Health Injection	**Note** This feature depends on Cisco IOS Release 12.2(33)SXI or later, and is only available on the Catalyst 6500 switch. Route Health Injection is used for injecting the connected and static routes and NAT pools configured on the FWSM into the MSFC routing table on a per context basis. MSFC can then redistribute the route or NAT pools to other router routing tables.
Virtual Switching System (VSS) support	**Note** This feature depends on Cisco IOS Release 12.2(33)SXI or later, and is only available on the Catalyst 6500 switch. VSS is a system virtualization technology that allows the pooling of multiple Catalyst 6500 switches into a single virtual switch. If you have the FWSM installed, FWSM traffic benefits from this feature. There is no configuration on the FWSM required.

New Features in Release 4.0(3)

The SCCP (Skinny) inspection has been enhanced to do the following:

- Support registrations of SCCP version 17 phones.
- Support SCCP version 17 media related messages for opening up pinholes for video/audio streams.

The following is not supported:

- Registrations of endpoints that have IPv6 addresses. The Register messages are dropped and a debug message is generated.
- If IPv6 messages are embedded in the SCCP messages, they are not NATed or PATed; they are left untranslated.

New Features in Release 4.0(2)

There were no new features in Release 4.0(2).

New Features in Release 4.0(1)

Table 1 lists the new features for Release 4.0(1).

Table 1 New Features for FWSM Release 4.0(1)

Feature	Description
Routing	
EIGRP	The following EIGRP features are supported in this release: • Summarization • Stub-routing • Route filtering • Manual Route summarization • Redistribution
Static route monitoring	If you configure multiple static routes to reach a network, the route monitoring feature can detect if a network goes down so that the next best route can be used.
DHCP	
DHCP Option 82 support	When the switch is acting as relay agent, to interoperate with HSRP, the FWSM will preserve the Option 82 field set up by the switch.
Modular Policy Framework	
Inspection policy maps and class maps	The following protocols support inspection policy and/or class maps: • DCERPC • ESMTP • HTTP • SIP
Regular expressions and regular expression class maps	You can create regular expressions and regular expression class maps for use in an inspection policy map or class map.
Filtering	

Table 1 New Features for FWSM Release 4.0(1) (continued)

Feature	Description
HTTPS support with Secure Computing SmartFilter	The FWSM now supports HTTPS filtering using Secure Computing SmartFilter.
Adding the context name to Websense version 4 requests	Because Websense requests initiated from the FWSM use the pre-NATted IP address of clients, which can be overlapping, this can lead to problems in defining policies in the Websense server. Adding the context name to Websense queries lets the Websense server use the context name for policy lookups.
Application Inspection	
DNS Guard configurability	You can now disable DNS Guard at the CLI.
SIP inspection enhancements	Numerous enhancements were added. You can now use an inspection policy map to configure special actions for inspection traffic; this method replaces the application map.
HTTP inspection enhancements	Numerous enhancements were added. You can now use an inspection policy map to configure special actions for inspection traffic; this method replaces the application map.
ESMTP inspection enhancements	Numerous enhancements were added. You can now use an inspection policy map to configure special actions for inspection traffic; this method replaces the application map.
DCERPC inspection enhancements	Numerous enhancements were added. You can now use an inspection policy map to configure special actions for inspection traffic; this method replaces the application map.
Access Lists	
Customizable memory partition sizes	In multiple context mode, you can change the size of memory partitions for rule use, so you can reallocate memory from one partition to another.
Rule reallocation per feature per partition	You can reallocate rules between features on a per-partition basis instead of just globally.
Access list optimization	The access list group optimization feature reduces the number of ACEs per group by merging and/or deleting redundant and conflicting ACEs without affecting the semantics of the access list.
Connections and Switch Integration	
Connection rate limiting	You can limit the connection rate for TCP and UDP traffic.
Monitoring	
New SNMP MIBs	For ACL entries and ACL hit counters (CISCO-IP-PROTOCOL-FILTER-MIB), and ARP table entries (IP-MIB).

Security Policy Overview

A security policy determines which traffic is allowed to pass through the firewall to access another network. The FWSM does not allow any traffic to pass through unless explicitly allowed by an access list. You can apply actions to traffic to customize the security policy. This section discusses some commonly-used features; not all features are listed here. This section includes the following topics:

Permitting or Denying Traffic with Access Lists

You can apply an access list to allow traffic through an interface. For transparent firewall mode, you can also apply an EtherType access list to allow non-IP traffic.

Applying NAT

Some of the benefits of NAT include the following:

- You can use private addresses on your inside networks. Private addresses are not routable on the Internet.
- NAT hides the local addresses from other networks, so attackers cannot learn the real address of a host.
- NAT can resolve IP routing problems by supporting overlapping IP addresses.

Protecting from IP Fragments

The FWSM provides IP fragment protection. This feature performs full reassembly of all ICMP error messages and virtual reassembly of the remaining IP fragments that are routed through the FWSM. Fragments that fail the security check are dropped and logged. Virtual reassembly cannot be disabled.

Using AAA for Through Traffic

You can require authentication and/or authorization for certain types of traffic, for example, for HTTP. The FWSM also sends accounting information to a RADIUS or TACACS+ server.

Applying Internet Filtering

Although you can use access lists to prevent outbound access to specific websites or FTP servers, configuring and managing web usage this way is not practical because of the size and dynamic nature of the Internet. We recommend that you use the FWSM in conjunction with a separate server running one of the following Internet filtering products:

- Websense Enterprise
- Sentian by N2H2

Applying Application Inspection

Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the FWSM to perform a deep packet inspection.

Applying Connection Limits

You can limit TCP and UDP connections and embryonic connections. Limiting the number of connections and embryonic connections protects you from a DoS attack. The FWSM uses the embryonic limit to trigger TCP Intercept, which protects inside systems from a DoS attack perpetrated by flooding an interface with TCP SYN packets. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination.

How the Firewall Services Module Works with the Switch

You can install the FWSM in the Catalyst 6500 series switches and the Cisco 7600 series routers with Cisco IOS software on both the switch supervisor and the integrated MSFC (known as "supervisor IOS").

Note The Catalyst Operating System (OS) is not supported.

The FWSM runs its own operating system.

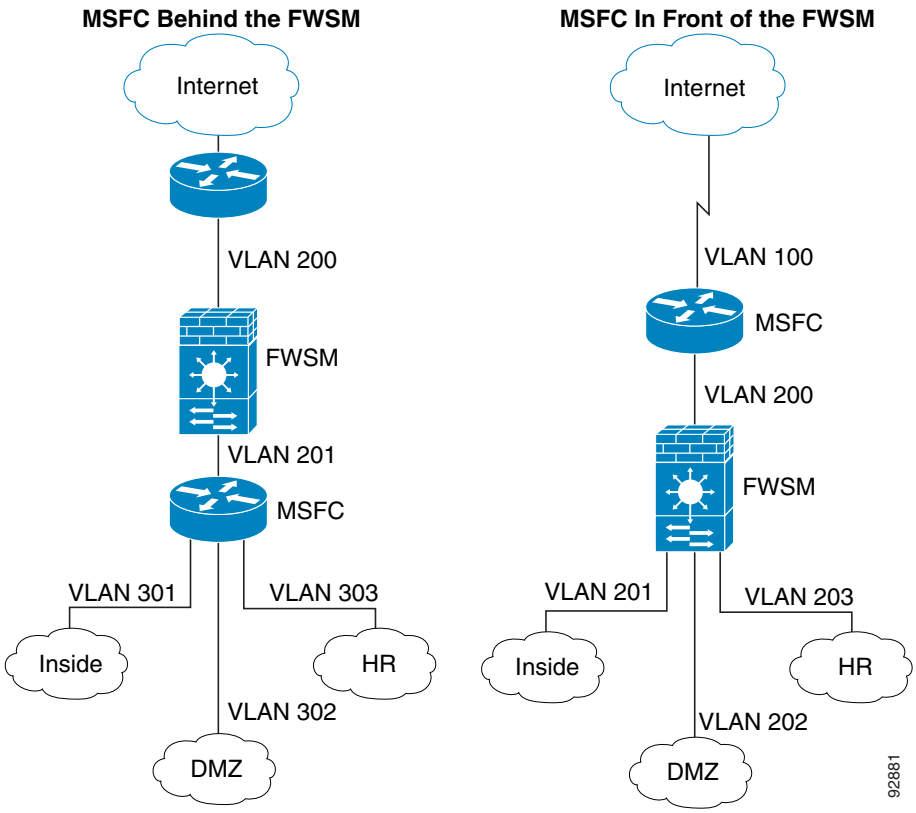
Using the MSFC

The switch includes a switching processor (the supervisor) and a router (the MSFC). Although you need the MSFC as part of your system, you do not have to use it. If you choose to do so, you can assign one or more VLAN interfaces to the MSFC (if your switch software version supports multiple SVIs; see Table A-1 on page A-2). In single context mode, you can place the MSFC in front of the firewall or behind the firewall (see Figure 1-1).

The location of the MSFC depends entirely on the VLANs that you assign to it. For example, the MSFC is behind the firewall in the example shown on the left side of Figure 1-1 because you assigned VLAN 201 to the inside interface of the FWSM. The MSFC is in front of the firewall in the example shown on the right side of Figure 1-1 because you assigned VLAN 200 to the outside interface of the FWSM.

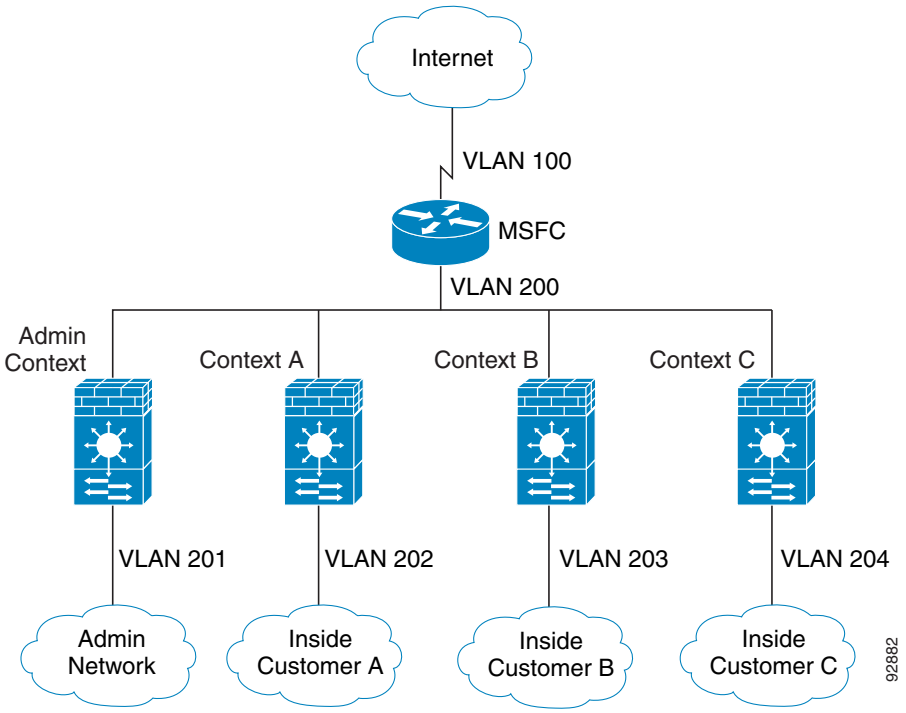
In the left-hand example, the MSFC routes between VLANs 201, 301, 302, and 303, and no inside traffic goes through the FWSM unless it is destined for the Internet. In the right-hand example, the FWSM processes and protects all traffic between the inside VLANs 201, 202, and 203.

Figure 1-1 MSFC Placement



For multiple context mode, if you place the MSFC behind the FWSM, you should only connect it to a single context. If you connect the MSFC to multiple contexts, the MSFC will route between the contexts, which might not be your intention. The typical scenario for multiple contexts is to use the MSFC in front of all the contexts to route between the Internet and the switched networks (see Figure 1-2).

Figure 1-2 MSFC Placement with Multiple Contexts



Firewall Mode Overview

The FWSM runs in two different firewall modes:

- Routed
- Transparent

In routed mode, the FWSM is considered to be a router hop in the network.

In transparent mode, the FWSM acts like a “bump in the wire,” or a “stealth firewall,” and is not considered a router hop. The FWSM connects to the same network on its inside and outside interfaces. You can configure up to eight pairs of interfaces (called bridge groups) to connect to eight different networks, per context.

You might use a transparent firewall to simplify your network configuration. Transparent mode is also useful if you want the firewall to be invisible to attackers. You can also use a transparent firewall for traffic that would otherwise be blocked in routed mode. For example, a transparent firewall can allow unsupported routing protocols.

In multiple context mode, you can choose the mode for each context independently, so some contexts can run in transparent mode while others can run in routed mode.

Stateful Inspection Overview

All traffic that goes through the firewall is inspected using the Adaptive Security Algorithm and is either allowed through or dropped. A simple packet filter can check for the correct source address, destination address, and ports, but it does not check that the packet sequence or flags are correct. A filter also checks every packet against the filter, which can be a slow process.

Note The following feature allows you to customize the packet flow: "Configuring TCP State Bypass" section on page 20-10.

A stateful firewall like the FWSM, however, takes into consideration the state of a packet:

- Is this a new connection?

  If it is a new connection, the firewall has to check the packet against access lists and perform other tasks to determine if the packet is allowed or denied. To perform this check, the first packet of the session goes through the "session management path," and depending on the type of traffic, it might also pass through the "control plane path."

  Note The first packet for a session cannot be comprised of fragments for a packet that is larger than 8500 Bytes. The session will be established, but only the first 8500 Bytes will be sent out. Subsequent packets for this session are not affected by this limitation.

  The session management path is responsible for the following tasks:

  - Performing the access list checks
  - Performing route lookups
  - Allocating NAT translations (xlates)
  - Establishing sessions in the "accelerated path"

  Some packets that require Layer 7 inspection (the packet payload must be inspected or altered) are passed on to the control plane path. Layer 7 inspection engines are required for protocols that have two or more channels: a data channel, which uses well-known port numbers, and a control channel, which uses different port numbers for each session. These protocols include FTP, H.323, and SNMP.

  Note The FWSM performs session management path and accelerated path processing on three specialized networking processors. The control plane path processing is performed in a general-purpose processor that also handles traffic directed to the FWSM and configuration and management tasks.

- Is this an established connection?

  If the connection is already established, the firewall does not need to recheck packets; most matching packets can go through the accelerated path in both directions. The accelerated path is responsible for the following tasks:

  - IP checksum verification
  - Session lookup
  - TCP sequence number check
  - NAT translations based on existing sessions

- Layer 3 and Layer 4 header adjustments

For UDP or other connectionless protocols, the FWSM creates connection state information so that it can also use the accelerated path.

Data packets for protocols that require Layer 7 inspection can also go through the accelerated path.

Some established session packets must continue to go through the session management path or the control plane path. Packets that go through the session management path include HTTP packets that require inspection or content filtering. Packets that go through the control plane path include the control packets for protocols that require Layer 7 inspection.

Note For QoS compatibility, the FWSM preserves the DSCP bits for all traffic that passes through the FWSM.

Security Context Overview

You can partition a single FWSM into multiple virtual devices, known as security contexts. Each context has its own security policy, interfaces, and administrators. Multiple contexts are similar to having multiple standalone devices. Many features are supported in multiple context mode, including routing tables, firewall features, and management. Some features are not supported, including dynamic routing protocols.

In multiple context mode, the FWSM includes a configuration for each context that identifies the security policy, interfaces, and almost all the options you can configure on a standalone device. The system administrator adds and manages contexts by configuring them in the system configuration, which, like a single mode configuration, is the startup configuration. The system configuration identifies basic settings for the FWSM. The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses one of the contexts that is designated as the admin context.

The admin context is just like any other context, except that when a user logs in to the admin context, then that user has system administrator rights and can access the system and all other contexts.

Note Multiple context mode supports static routing only.

CHAPTER 2

Configuring the Switch for the Firewall Services Module

This chapter describes how to configure the Catalyst 6500 series switch or the Cisco 7600 series router for use with the FWSM. Before completing the procedures in this chapter, configure the basic properties of your switch, including assigning VLANs to interfaces, according to the documentation that came with your switch.

This chapter includes the following sections:

Switch Overview

You can install the FWSM in the Catalyst 6500 series switches or the Cisco 7600 series routers. The configuration of both series is identical, and the series are referred to generically in this guide as the "switch." The switch includes a switch (the supervisor engine) as well as a router (the MSFC).

The switch supports Cisco IOS software on both the switch supervisor engine and the integrated MSFC router.

Note The Catalyst operating system software is not supported.

The FWSM runs its own operating system.

Note Because the FWSM runs its own operating system, upgrading the Cisco IOS software does not affect the operation of the FWSM.

See the "Using the MSFC" section on page 1-7 for more information about the MSFC.

Some FWSM features interact with Cisco IOS features, and require specific Cisco IOS software versions. See the "Switch Hardware and Software Compatibility" section on page A-1 for more information. The following features involve Cisco IOS software, and are described in the feature sections:

- Route Health Injection—See the "Configuring Route Health Injection" section on page 8-32.
- PISA integration—See the "Permitting or Denying Application Types with PISA Integration" section on page 20-4.
- Virtual Switching System (VSS) support—No FWSM configuration required.

Note For Cisco IOS software Version 12.2(18)SX6 and earlier, for each FWSM in a switch, the SPAN reflector feature is enabled. This feature enables multicast traffic (and other traffic that requires central rewrite engine) to be switched when coming from the FWSM. The SPAN reflector feature uses one SPAN session. To disable this feature, enter the following command:

```
Router(config)# no monitor session servicemodule
```

Verifying the Module Installation

To verify that the switch acknowledges the FWSM and has brought it online, view the module information using the following command:

```
Router> show module [mod-num | all]
```

The following is sample output from the **show module** command:

```
Router> show module
Mod Ports Card Type                              Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------
  1     2  Catalyst 6000 supervisor 2 (Active)    WS-X6K-SUP2-2GE    SAD0444099Y
  2    48  48 port 10/100 mb RJ-45 ethernet       WS-X6248-RJ-45     SAD03475619
  3     2  Intrusion Detection System             WS-X6381-IDS       SAD04250KV5
  4     6  Firewall Module                        WS-SVC-FWM-1       SAD062302U4
```

Note The **show module** command shows six ports for the FWSM; these are internal ports that are grouped together as an EtherChannel. See the "Customizing the FWSM Internal Interface" section on page 2-8 for more information.

Assigning VLANs to the Firewall Services Module

This section describes how to assign VLANs to the FWSM. The FWSM does not include any external physical interfaces. Instead, it uses VLAN interfaces. Assigning VLANs to the FWSM is similar to assigning a VLAN to a switch port; the FWSM includes an internal interface to the Switch Fabric Module (if present) or the shared bus.

Note See the switch documentation for information about adding VLANs to the switch and assigning them to switch ports.

This section includes the following topics:

VLAN Guidelines

See the following guidelines for using VLANs with the FWSM:

- You can use private VLANs with the FWSM. Assign the primary VLAN to the FWSM; the FWSM automatically handles secondary VLAN traffic.
- You cannot use reserved VLANs.
- You cannot use VLAN 1.
- If you are using FWSM failover within the same switch chassis, do not assign the VLAN(s) you are reserving for failover and stateful communications to a switch port. However, if you are using failover between chassis, you must include the VLANs in the trunk port between the chassis.
- If you do not add the VLANs to the switch before you assign them to the FWSM, the VLANs are stored in the supervisor engine database and are sent to the FWSM as soon as they are added to the switch.

Assigning VLANs to the FWSM

In Cisco IOS software, create up to 16 firewall VLAN groups, and then assign the groups to the FWSM. For example, you can assign all the VLANs to one group, or you can create an inside group and an outside group, or you can create a group for each customer. Each group can contain unlimited VLANs.

You cannot assign the same VLAN to multiple firewall groups; however, you can assign multiple firewall groups to an FWSM and you can assign a single firewall group to multiple FWSMs. VLANs that you want to assign to multiple FWSMs, for example, can reside in a separate group from VLANs that are unique to each FWSM.

To assign VLANs to the FWSM, perform the following steps:

Step 1 To assign VLANs to a firewall group, enter the following command:

```
Router(config)# firewall vlan-group firewall_group vlan_range
```

The *firewall_group* argument is an integer.

The *vlan_range* can be one or more VLANs (2 to 1000 and from 1025 to 4094) identified in one of the following ways:

- A single number (*n*)
- A range (*n-x*)

Separate numbers or ranges by commas. For example, enter the following numbers:

```
5,7-10,13,45-100
```

Note Routed ports and WAN ports consume internal VLANs, so it is possible that VLANs in the 1020-1100 range might already be in use.

If you configure the VLANs in the FWSM configuration, and then later assign the VLANs to the FWSM on the switch using this procedure, then those VLANs are brought administratively up on the FWSM even if they were configured to be shut down. To shut them down, enter the following commands at the FWSM CLI:

```
interface vlan number
shutdown
```

Step 2 To assign the firewall groups to the FWSM, enter the following command:

```
Router(config)# firewall module module_number vlan-group firewall_group
```

The *firewall_group* is one or more group numbers:

- A single number (*n*)
- A range (*n-x*)

Separate numbers or ranges by commas. For example, enter the following numbers:

```
5,7-10
```

The following example shows how you can create three firewall VLAN groups: one for each FWSM, and one that includes VLANs assigned to both FWSMs:

```
Router(config)# firewall vlan-group 50 55-57
Router(config)# firewall vlan-group 51 70-85
Router(config)# firewall vlan-group 52 100
Router(config)# firewall module 5 vlan-group 50,52
Router(config)# firewall module 8 vlan-group 51,52
```

The following is sample output from the **show firewall vlan-group** command:

```
Router# show firewall vlan-group
Group vlans
----- ------
   50 55-57
   51 70-85
   52 100
```

The following is sample output from the **show firewall module** command, which shows all VLAN groups:

```
Router# show firewall module
Module Vlan-groups
  5    50,52
  8    51,52
```

Adding Switched Virtual Interfaces to the MSFC

A VLAN defined on the MSFC is called a switched virtual interface. If you assign the VLAN used for the SVI to the FWSM (see the "Assigning VLANs to the Firewall Services Module" section on page 2-2), then the MSFC routes between the FWSM and other Layer 3 VLANs.

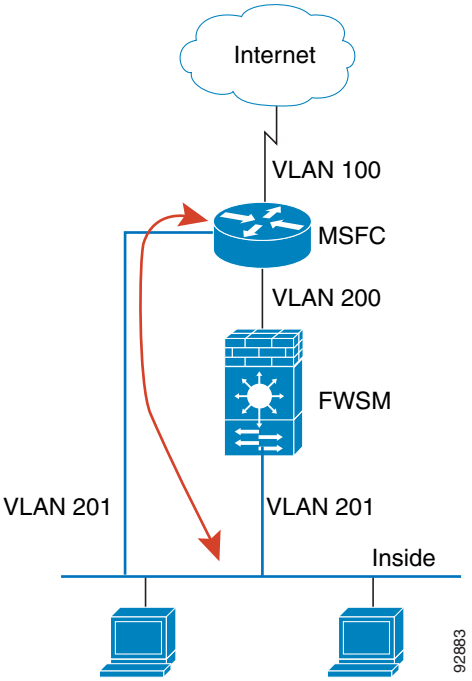
This section includes the following topics:

- Configuring SVIs, page 2-7

SVI Overview

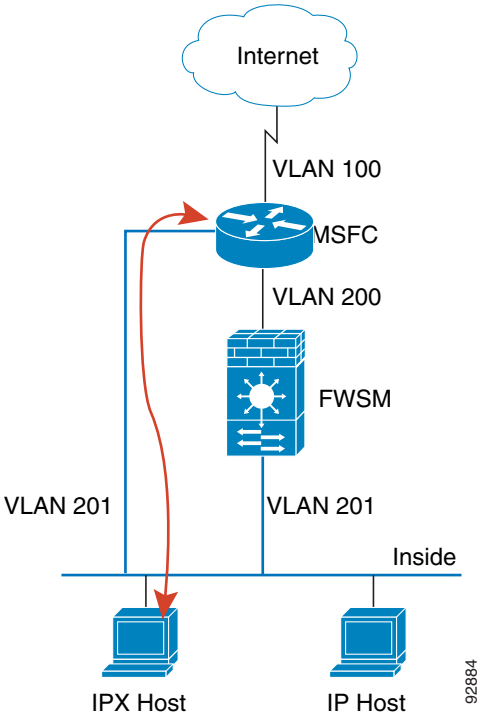
For security reasons, by default, only one SVI can exist between the MSFC and the FWSM. For example, if you misconfigure the system with multiple SVIs, you could accidentally allow traffic to pass around the FWSM by assigning both the inside and outside VLANs to the MSFC. (See Figure 2-1.)

Figure 2-1 Multiple SVI Misconfiguration



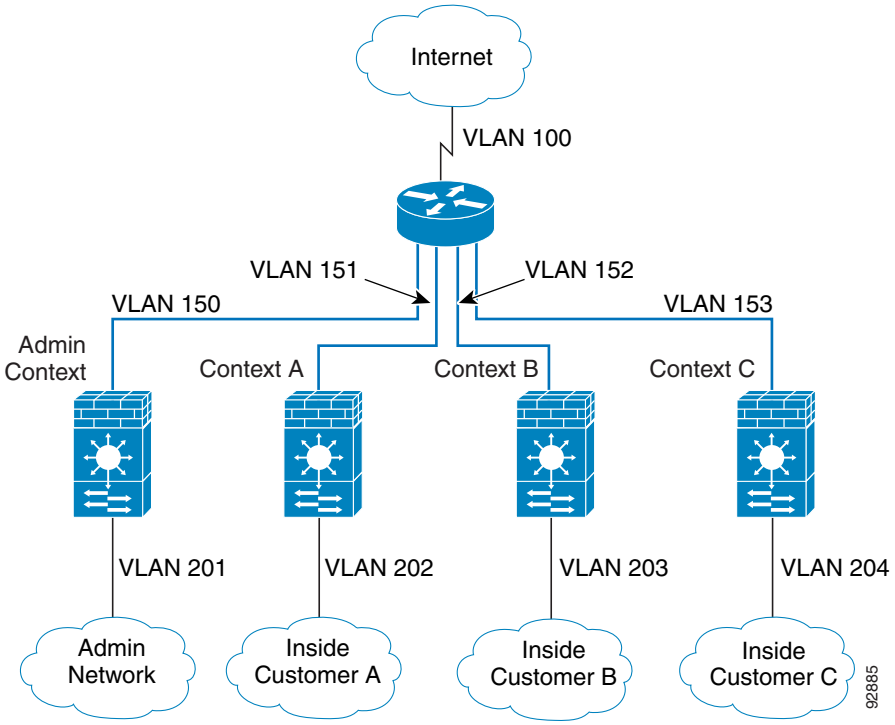
However, you might need to bypass the FWSM in some network scenarios. Figure 2-2 shows an IPX host on the same Ethernet segment as IP hosts. Because the FWSM in routed firewall mode only handles IP traffic and drops other protocol traffic like IPX (transparent firewall mode can optionally allow non-IP traffic), you might want to bypass the FWSM for IPX traffic. Make sure to configure the MSFC with an access list that allows only IPX traffic to pass on VLAN 201.

Figure 2-2 Multiple SVIs for IPX



For transparent firewalls in multiple context mode, you need to use multiple SVIs because each context requires a unique VLAN on its outside interface (See Figure 2-3). You might also choose to use multiple SVIs in routed mode so you do not have to share a single VLAN for the outside interface.

Figure 2-3 Multiple SVIs in Multiple Context Mode



Configuring SVIs

To add an SVI to the MSFC, perform the following steps:

Step 1 (Optional) To allow you to add more than one SVI to the FWSM, enter the following command:

```
Router(config)# firewall multiple-vlan-interfaces
```

Step 2 To add a VLAN interface to the MSFC, enter the following command:

```
Router(config)# interface vlan vlan_number
```

Step 3 To set the IP address for this interface on the MSFC, enter the following command:

```
Router(config-if)# ip address address mask
```

Step 4 To enable the interface, enter the following command:

```
Router(config-if)# no shutdown
```

The following example shows a typical configuration with multiple SVIs:

```
Router(config)# firewall vlan-group 50 55-57
```

```
Router(config)# firewall vlan-group 51 70-85
Router(config)# firewall module 8 vlan-group 50-51
Router(config)# firewall multiple-vlan-interfaces
Router(config)# interface vlan 55
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# interface vlan 56
Router(config-if)# ip address 10.1.2.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# end
Router#
```

The following is sample output from the **show interface** command:

```
Router# show interface vlan 55
Vlan55 is up, line protocol is up
  Hardware is EtherSVI, address is 0008.20de.45ca (bia 0008.20de.45ca)
  Internet address is 10.1.1.1/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type:ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Input queue:0/75/0/0 (size/max/drops/flushes); Total output drops:0
  Queueing strategy:fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  L2 Switched:ucast:196 pkt, 13328 bytes - mcast:4 pkt, 256 bytes
  L3 in Switched:ucast:0 pkt, 0 bytes - mcast:0 pkt, 0 bytes mcast
  L3 out Switched:ucast:0 pkt, 0 bytes
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     4 packets output, 256 bytes, 0 underruns
     0 output errors, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Customizing the FWSM Internal Interface

The connection between the FWSM and the switch is a 6-GB 802.1Q trunking EtherChannel. This EtherChannel is automatically created when you install the FWSM. On the FWSM side, two NPs connect to three Gigabit Ethernet interfaces each, and these interfaces comprise the EtherChannel. The switch distributes traffic to the interfaces in the EtherChannel according to a distribution algorithm based on session information; load sharing is not performed on a per-packet basis, but rather on a flow basis. In some cases, the algorithm assigns traffic unevenly between the interfaces and, therefore, between the two NPs. Aside from not utilizing the full processing potential of the FWSM, consistent inequity can result in unexpected behavior when you apply resource management to multiple contexts. (See the "Configuring a Class" section on page 4-24 for more information.)

To change the load-balancing method, enter the following command:

```
Router(config)# port-channel load-balance {dst-ip | dst-mac | dst-port | src-dst-ip |
src-dst-mac | src-dst-port | src-ip | src-mac | src-port}
```

The default is **src-dst-ip**.

Configuring the Switch for Failover

To configure the switch for failover, see the following topics:

Assigning VLANs to the Secondary Firewall Services Module

Because both units require the same access to the inside and outside networks, you must assign the same VLANs to both FWSMs on the switch(es). See the "Assigning VLANs to the Firewall Services Module" section on page 2-2.

Adding a Trunk Between a Primary Switch and Secondary Switch

If you are using inter-switch failover (see the "Intra- and Inter-Chassis Module Placement" section on page 13-3), then you should configure an 802.1Q VLAN trunk between the two switches to carry the failover and state links. The trunk should have QoS enabled so that failover VLAN packets, which have the CoS value of 5 (higher priority), are treated with higher priority in these ports.

To configure the EtherChannel and trunk, see the documentation for your switch.

Ensuring Compatibility with Transparent Firewall Mode

To avoid loops when you use failover in transparent mode, use switch software that supports BPDU forwarding. See the "Switch Hardware and Software Compatibility" section on page A-1 for more information about switch support for transparent firewall mode.

Do not enable LoopGuard globally on the switch if the FWSM is in transparent mode. LoopGuard is automatically applied to the internal EtherChannel between the switch and the FWSM, so after a failover and a failback, LoopGuard causes the secondary unit to be disconnected because the EtherChannel goes into the err-disable state.

Enabling Autostate Messaging for Rapid Link Failure Detection

Using Cisco IOS software Release 12.2(18)SXF5 and higher, the supervisor engine can send autostate messages to the FWSM about the status of physical interfaces associated with FWSM VLANs. For example, when all physical interfaces associated with a VLAN go down, the autostate message tells the FWSM that the VLAN is down. This information lets the FWSM declare the VLAN as down, bypassing the interface monitoring tests normally required for determining which side suffered a link failure. Autostate messaging provides a dramatic improvement in the time the FWSM takes to detect a link failure (a few milliseconds as compared to up to 45 seconds without autostate support).

The switch supervisor sends an autostate message to the FWSM when:

- The last interface belonging to a VLAN goes down.
- The first interface belonging to a VLAN comes up.

Note The switch supports autostate messaging only if you install a single FWSM in the chassis.

Autostate messaging is disabled by default. To enable autostate messaging in Cisco IOS software, enter the following command:

```
Router(config)# firewall autostate
```

Managing the Firewall Services Module Boot Partitions

This section describes how to reset the FWSM from the switch, and how to manage the boot partitions on the Flash memory card. This section includes the following topics:

Flash Memory Overview

The FWSM has a 128-MB Flash memory card that stores the operating system, configurations, and other data. The Flash memory includes six partitions, called **cf:***n* in Cisco IOS software commands:

- Maintenance partition (**cf:1**)—Contains the maintenance software. Use the maintenance software to upgrade or install application images if you cannot boot into the application partition, to reset the application image password, or to display the crash dump information.
- Network configuration partition (**cf:2**)—Contains the network configuration of the maintenance software. The maintenance software requires IP settings so that the FWSM can reach the TFTP server to download application software images.
- Crash dump partition (**cf:3**)—Stores the crash dump information.
- Application partitions (**cf:4** and **cf:5**)—Stores the application software image, system configuration, and ASDM. By default, Cisco installs the images on **cf:4**. You can use **cf:5** as a test partition. For example, if you want to upgrade your software, you can install the new software on **cf:5**, but maintain the old software as a backup in case you have problems. Each partition includes its own startup configuration.
- Security context partition (**cf:6**)—64 MB are dedicated to this partition, which stores security context configurations (if desired) and RSA keys in a navigable file system. Other partitions do not have file systems that allow you to perform common tasks such as listing files. This partition is called **disk** when using the **copy** command.

Setting the Default Boot Partition

By default, the FWSM boots from the **cf:4** application partition. However, you can choose to boot from the **cf:5** application partition or into the **cf:1** maintenance partition. Each application partition has its own startup configuration.

To change the default boot partition, enter the following command:

```
Router(config)# boot device module mod_num cf:n
```

Where *n* is 1 (maintenance), 4 (application), or 5 (application).

To view the current boot partition, enter the following command:

```
Router# show boot device [mod_num]
```

For example:

```
Router# show boot device
[mod:1 ]:
[mod:2 ]:
[mod:3 ]:
[mod:4 ]: cf:4
[mod:5 ]: cf:4
[mod:6 ]:
[mod:7 ]: cf:4
[mod:8 ]:
[mod:9 ]:
```

Resetting the FWSM or Booting from a Specific Partition

This section describes how to reset the FWSM or boot from a specific partition. You might need to reset the FWSM if you cannot reach it through the CLI or an external Telnet session. You might need to boot from a non-default boot partition if you need to access the maintenance partition or if you want to boot from a different software image in the backup application partition. The maintenance partition is valuable for troubleshooting.

The reset process might take several minutes.

When you reset the FWSM, you can also choose to run a full memory test. When the FWSM initially boots, it only runs a partial memory test. A full memory test takes approximately six minutes.

Note To reload the FWSM when you are logged into the FWSM, enter **reload** or **reboot**. You cannot boot from a non-default boot partition with these commands.

To reset the FWSM, enter the following command:

```
Router# hw-module module mod_num reset [cf:n] [mem-test-full]
```

The **cf:***n* argument is the partition, either 1 (maintenance), 4 (application), or 5 (application). If you do not specify the partition, the default partition is used (typically **cf:4**).

The **mem-test-full** option runs a full memory test, which takes approximately 6 minutes.

The following example shows how to reset the FWSM installed in slot 9. The default boot partition is used.

```
Router# hw-module module 9 reset

Proceed with reload of module? [confirm] y
% reset issued for module 9

Router#
00:26:55:%SNMP-5-MODULETRAP:Module 9 [Down] Trap
00:26:55:SP:The PC in slot 8 is shutting down. Please wait ...
```

CHAPTER 3

Connecting to the Firewall Services Module and Managing the Configuration

This chapter describes how to access the command-line interface and work with the configuration. This chapter includes the following sections:

Connecting to the Firewall Services Module

This section describes how to connect or "session" to the FWSM from the switch command line. It also describes how to log out of the FWSM to access the switch CLI. This section includes the following topics:

Logging in to the FWSM

The FWSM does not have an external console port, you must session in to the FWSM for initial configuration. Later, when you configure interfaces and IP addresses on the FWSM itself, you can access the FWSM CLI remotely through an FWSM interface. See Chapter 22, "Configuring Management Access," for more information.

Without any additional configuration for user authentication (see the "AAA for System Administrators" section on page 22-10), the login method consists of logging in as the default user:

1. The login password lets you access user EXEC mode.
2. To access configuration commands, you must enter privileged EXEC mode, which requires a second password.
3. From privileged EXEC mode, you can access global configuration mode, which does not require a password.

Caution Management access to the FWSM causes a degradation in performance. We recommend that you avoid accessing the FWSM when high network performance is critical.

To session in to the FWSM from the switch, log in, access privileged mode, and then configuration mode, perform the following steps:

Step 1 Session in to the FWSM from the switch using the command appropriate for your switch operating system:

- Cisco IOS software

```
Router# session slot number processor 1
```

- Catalyst operating system software

```
Console> (enable) session module_number
```

For multiple context mode, when you session in to the FWSM, you access the system configuration. See Chapter 4, "Configuring Security Contexts," for more information.

Step 2 Log in to the FWSM by entering the login password at the following prompt:

```
hostname passwd:
```

By default, the password is **cisco**.

To change the password, see the "Changing the Passwords" section on page 7-1.

Step 3 To access privileged EXEC mode, enter the following command:

```
hostname> enable
```

This command accesses the highest privilege level.

The following prompt appears:

```
Password:
```

Step 4 Enter the enable password at the prompt.

By default, the password is blank, and you can press the **Enter** key to continue. See the "Changing the Passwords" section on page 7-1 to change the enable password.

The prompt changes to:

```
hostname#
```

To exit privileged mode, enter **disable**. You can also enter **exit** or **quit** to exit the current access mode (privileged EXEC mode, global configuration mode, and so on).

Step 5 To access configuration mode, enter the following command:

```
hostname# configure terminal
```

The prompt changes to the following:

```
hostname(config)#
```

Logging out of the FWSM

To end the FWSM session and access the switch CLI, enter the following command:

```
hostname# exit

Logoff
```

```
[Connection to 127.0.0.31 closed by foreign host]
Router#
```

You might need to enter the **exit** command multiple times if you are in a configuration mode.

Managing the Configuration

This section describes how to work with the configuration. The FWSM loads the configuration from a text file, called the startup configuration.

When you enter a command, the change is made only to the running configuration in memory. You must manually save the running configuration to the startup configuration for your changes to remain after a reboot.

The information in this section applies to both single and multiple security contexts, except where noted. Additional information about contexts is in Chapter 4, "Configuring Security Contexts,"

This section includes the following topics:

Saving Configuration Changes

This section describes how to save your configuration, and includes the following topics:

Saving Configuration Changes in Single Context Mode

To save the running configuration to the startup configuration, enter the following command:

```
hostname# write memory
```

Note The **copy running-config startup-config** command is equivalent to the **write memory** command.

Saving Configuration Changes in Multiple Context Mode

You can save each context (and system) configuration separately, or you can save all context configurations at the same time. This section includes the following topics:

Saving Each Context and System Separately

To save the system or context configuration, enter the following command within the system or context:

```
hostname# write memory
```

Note The **copy running-config startup-config** command is equivalent to the **write memory** command.

For multiple context mode, context startup configurations can reside on external servers. In this case, the FWSM saves the configuration back to the server you identified in the context URL, except for an HTTP or HTTPS URL, which do not let you save the configuration to the server.

Saving All Context Configurations at the Same Time

To save all context configurations at the same time, as well as the system configuration, enter the following command in the system execution space:

```
hostname# write memory all [/noconfirm]
```

If you do not enter the **/noconfirm** keyword, you see the following prompt:

```
Are you sure [Y/N]:
```

After you enter **Y**, the FWSM saves the system configuration and each context. Context startup configurations can reside on external servers. In this case, the FWSM saves the configuration back to the server you identified in the context URL, except for an HTTP or HTTPS URL, which do not let you save the configuration to the server.

After the FWSM saves each context, the following message appears:

```
'Saving context 'b' ... ( 1/3 contexts saved ) '
```

Sometimes, a context is not saved because of an error. See the following information for errors:

- For contexts that are not saved because of low memory, the following message appears:

  ```
  The context 'context a' could not be saved due to Unavailability of resources
  ```

- For contexts that are not saved because the remote destination is unreachable, the following message appears:

  ```
  The context 'context a' could not be saved due to non-reachability of destination
  ```

- For contexts that are not saved because the context is locked, the following message appears:

  ```
  Unable to save the configuration for the following contexts as these contexts are
  locked.
  context 'a' , context 'x' , context 'z' .
  ```

  A context is only locked if another user is already saving the configuration or in the process of deleting the context.

- For contexts that are not saved because the startup configuration is read-only (for example, on an HTTP server), the following message report is printed at the end of all other messages:

  ```
  Unable to save the configuration for the following contexts as these contexts have
  read-only config-urls:
  context 'a' , context 'b' , context 'c' .
  ```

- For contexts that are not saved because of bad sectors in the Flash memory, the following message appears:

  ```
  The context 'context a' could not be saved due to Unknown errors
  ```

Copying the Startup Configuration to the Running Configuration

Copy the new startup configuration to the running configuration using one of these options:

- To merge the startup configuration with the current running configuration, enter the following command:

  ```
  hostname(config)# copy startup-config running-config
  ```

  A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results.

- To load the startup configuration and discard the running configuration, restart the FWSM by entering the following command:

  ```
  hostname# reload
  ```

  Alternatively, you can use the following commands to load the startup configuration and discard the running configuration without requiring a reboot:

  ```
  hostname(config)# clear configure all
  hostname(config)# copy startup-config running-config
  ```

Viewing the Configuration

The following commands let you view the running and startup configurations.

- To view the running configuration, enter the following command:

  ```
  hostname# show running-config
  ```

- To view the running configuration of a specific command, enter the following command:

  ```
  hostname# show running-config command
  ```

- To view the startup configuration, enter the following command:

  ```
  hostname# show startup-config
  ```

Clearing and Removing Configuration Settings

To erase settings, enter one of the following commands.

- To clear all the configuration for a specified command, enter the following command:

  ```
  hostname(config)# clear configure configurationcommand [level2configurationcommand]
  ```

This command clears all the current configuration for the specified configuration command. If you only want to clear the configuration for a specific version of the command, you can enter a value for *level2configurationcommand*.

For example, to clear the configuration for all **aaa** commands, enter the following command:

```
hostname(config)# clear configure aaa
```

To clear the configuration for only **aaa authentication** commands, enter the following command:

```
hostname(config)# clear configure aaa authentication
```

- To disable the specific parameters or options of a command, enter the following command:

  ```
  hostname(config)# no configurationcommand [level2configurationcommand] qualifier
  ```

  In this case, you use the **no** command to remove the specific configuration identified by *qualifier*.

  For example, to remove a specific **nat** command, enter enough of the command to identify it uniquely as follows:

  ```
  hostname(config)# no nat (inside) 1
  ```

- To erase the startup configuration, enter the following command:

  ```
  hostname(config)# write erase
  ```

- To erase the running configuration, enter the following command:

  ```
  hostname(config)# clear configure all
  ```

Note In multiple context mode, if you enter **clear configure all** from the system configuration, you also remove all contexts and stop them from running.

Creating Text Configuration Files Offline

This guide describes how to use the CLI to configure the FWSM; when you save commands, the changes are written to a text file. Instead of using the CLI, however, you can edit a text file directly on your PC and paste a configuration at the configuration mode command-line prompt in its entirety, or line by line. Alternatively, you can download a text file to the FWSM internal Flash memory. See Chapter 23, "Managing Software, Licenses, and Configurations," for information on downloading the configuration file to the FWSM.

In most cases, commands described in this guide are preceded by a CLI prompt. The prompt in the following example is "hostname(config)#":

```
hostname(config)# context a
```

In the text configuration file you are not prompted to enter commands, so the prompt is omitted as follows:

```
context a
```

For additional information about formatting the file, see Appendix C, "Using the Command-Line Interface."

CHAPTER 4

Configuring Security Contexts

This chapter describes how to configure multiple security contexts, and includes the following sections:

Security Context Overview

You can partition a single FWSM into multiple virtual devices, known as security contexts. Each context has its own security policy, interfaces, and administrators. Multiple contexts are similar to having multiple standalone devices. Many features are supported in multiple context mode, including routing tables, firewall features, and management. Some features are not supported, including most dynamic routing protocols.

This section provides an overview of security contexts, and includes the following topics:

Common Uses for Security Contexts

You might want to use multiple security contexts in the following situations:

- You are a service provider and want to sell security services to many customers. By enabling multiple security contexts on the FWSM, you can implement a cost-effective, space-saving solution that keeps all customer traffic separate and secure, and also eases configuration.
- You are a large enterprise or a college campus and want to keep departments completely separate.
- You are an enterprise that wants to provide distinct security policies to different departments.
- You have any network that requires more than one firewall.

Unsupported Features

Multiple context mode does not support the following features:

- Most dynamic routing protocols. BGP stub mode is supported.

  Security contexts support only static routes or BGP stub mode. You cannot enable OSPF or RIP in multiple context mode. You can, however, configure Route Health Injection, which lets you inject static, connected, and NAT addresses into the MSFC routing table. See the "Configuring Route Health Injection" section on page 8-32.
- Multicast routing. Multicast bridging is supported.

Context Configuration Files

This section describes how the FWSM implements multiple context mode configurations and includes the following topics:

Context Configurations

The FWSM includes a configuration for each context that identifies the security policy, interfaces, and almost all the options you can configure on a standalone device. You can store context configurations on the internal Flash memory or the external Flash memory card, or you can download them from a TFTP, FTP, or HTTP(S) server.

System Configuration

The system administrator adds and manages contexts by configuring each context configuration location, allocated interfaces, and other context operating parameters in the system configuration, which, like a single mode configuration, is the startup configuration. The system configuration identifies basic settings for the FWSM. The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses one of the contexts that is designated as the *admin context*. The system configuration does include a specialized failover interface for failover traffic only.

Admin Context Configuration

The admin context is just like any other context, except that when a user logs in to the admin context, then that user has system administrator rights and can access the system and all other contexts. The admin context is not restricted in any way, and can be used as a regular context. However, because logging into the admin context grants you administrator privileges over all contexts, you might need to restrict access to the admin context to appropriate users. The admin context must reside on Flash memory, and not remotely.

If your system is already in multiple context mode, or if you convert from single mode, the admin context is created automatically as a file on the internal Flash memory called admin.cfg. This context is named "admin." If you do not want to use admin.cfg as the admin context, you can change the admin context.

How the FWSM Classifies Packets

Each packet that enters the FWSM must be classified, so that the FWSM can determine to which context to send a packet. The FWSM uses only one global MAC address across all interfaces. A single MAC address is usually not a problem unless multiple contexts want to share an interface. A router cannot direct packets to IP addresses on the same network if all IP addresses resolve to the same MAC address. Moreover, the bridging table of the switch would constantly change as the MAC address moves from one interface to another. The purpose of the security context classifier is to resolve this situation.

This section includes the following topics:

Valid Classifier Criteria

If only one context is associated with the ingress interface, the FWSM classifies the packet into that context. In transparent firewall mode, unique interfaces for contexts are required, so this method is used to classify packets at all times.

If multiple contexts share an interface, then the classifier intercepts the packet and performs a destination IP address lookup. All other fields are ignored; only the destination IP address is used. To use the destination address for classification, the classifier must have knowledge about the subnets located behind each security context. The classifier relies on active NAT sessions to determine the subnets in each context. Active NAT sessions are created either by **static** commands, which create a permanent session, or by active dynamic NAT sessions.

For example, the classifier gains knowledge about subnets 10.10.10.0, 10.20.10.0 and 10.30.10.0 when the context administrators configure **static** commands in each context:

- Context A:

```
static (inside,shared) 10.10.10.0 10.10.10.0 netmask 255.255.255.0
```

- Context B:

```
static (inside,shared) 10.20.10.0 10.20.10.0 netmask 255.255.255.0
```

- Context C:

```
static (inside,shared) 10.30.10.0 10.30.10.0 netmask 255.255.255.0
```

If you use dynamic NAT, an active NAT session is created when the real host creates a connection through the shared interface. For traffic returning to the host, the active NAT session is used to classify the packet.

To quickly identify possible overlaps between different contexts, a situation that leads to connectivity problems, enter the **show np 3 static** command in the system execution space.

Note For management traffic destined for an interface, the interface IP address is used for classification.

Invalid Classifier Criteria

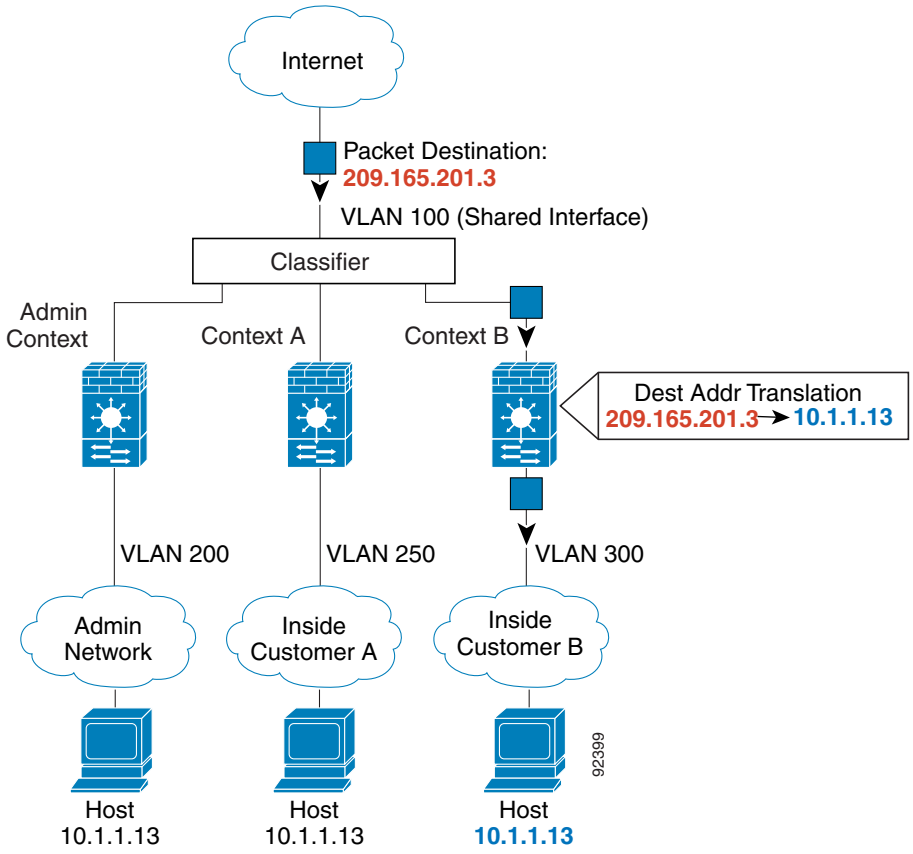
The following configurations are not used for packet classification:

- NAT exemption—The classifier does not use a NAT exemption configuration for classification purposes because NAT exemption does not identify the mapped (shared) interface.
- Routing table—The classifier does not use the routing table for classification. For example, if a context includes a static route that points to an external router as the next-hop to a subnet, and a different context includes a **static** command for the same subnet, then the classifier uses the **static** command to classify packets destined for that subnet and ignores the static route.

Classification Examples

Figure 4-1 shows multiple contexts sharing an outside interface, while the inside interfaces are unique, allowing overlapping IP addresses. The classifier assigns the packet to Context B because Context B includes the address translation that matches the destination address.

Figure 4-1 Packet Classification with a Shared Interface



Note that all new incoming traffic must be classified, even from inside networks. Figure 4-2 shows a host on the Context B inside network accessing the Internet. The classifier assigns the packet to Context B because the ingress interface is VLAN 300, which is assigned to Context B.

Figure 4-2 Incoming Traffic from Inside Networks

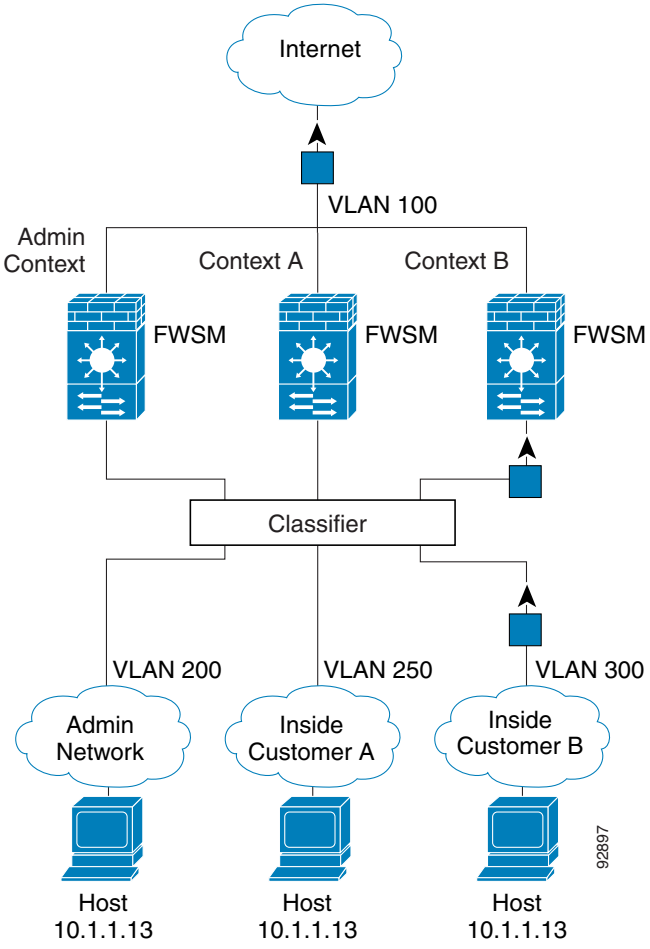
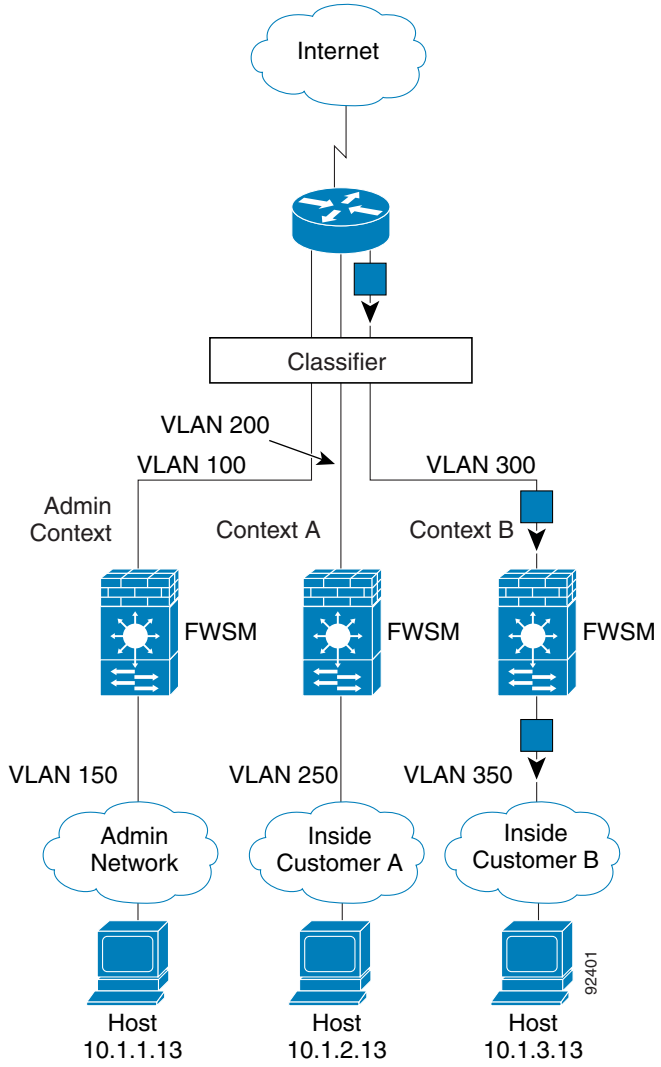


Figure 4-3 shows a transparent firewall with a host on the Context B inside network accessing the Internet. The classifier assigns the packet to Context B because the ingress interface is VLAN 300, which is assigned to Context B.

Figure 4-3 Transparent Firewall Contexts



Sharing Interfaces Between Contexts

Routed Mode Only

The FWSM lets you share an interface between contexts. However, packet classification requirements might make sharing interfaces impractical. Because the classifier relies on active NAT sessions to classify the destination addresses to a context, the classifier is limited by how you can configure NAT. If you do not want to perform NAT, you must use unique interfaces.

Note The FWSM does not support sharing the outside interface of one context with the inside interface of another context (known as cascading contexts). Traffic that is outbound from one context (from a higher to a lower security interface) can only enter another context as inbound traffic (lower to higher security); it cannot be outbound for both contexts, or inbound for both contexts.

This section includes the following topics:

NAT and Origination of Traffic

The type of NAT configured determines whether the traffic can originate on the shared interface or if it can only respond to an existing connection. When you use dynamic NAT, you cannot initiate a connection to the real addresses. Therefore, traffic from the shared interface must be in response to an existing connection. Static NAT, however, lets you initiate connections, so you can initiate connections on the shared interface.

Sharing an Outside Interface

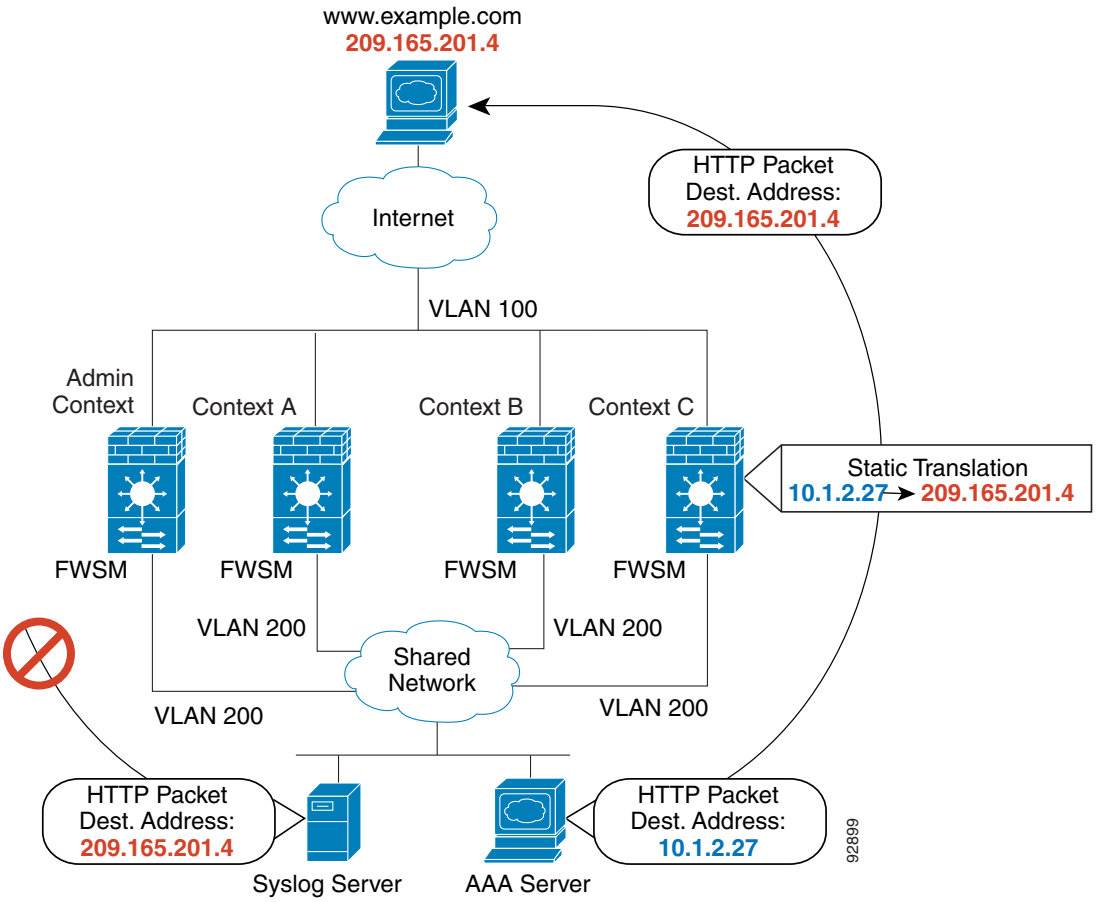
When you have an outside shared interface (connected to the Internet, for example), the destination addresses on the inside are limited, and are known by the system administrator, so configuring NAT for those addresses is easy, even if you want to configure static NAT.

Sharing an Inside Interface

Configuring an inside shared interface poses a problem, however, if you want to allow communication between the shared interface and the Internet, where the destination addresses are unlimited. For example, if you want to allow inside hosts on the shared interface to initiate traffic to the Internet, then you need to configure static NAT statements for each Internet address. This requirement necessarily limits the kind of Internet access you can provide for users on an inside shared interface. (If you intend to statically translate addresses for Internet servers, then you also need to consider DNS entry addresses and how NAT affects them. For example, if a server sends a packet to www.example.com, then the DNS server needs to return the translated address. Your NAT configuration determines DNS entry management.)

Figure 4-4 shows two servers on an inside shared interface. One server sends a packet to the translated address of a web server, and the FWSM classifies the packet to go through Context C because it includes a static translation for the address. The other server sends the packet to the real untranslated address, and the packet is dropped because the FWSM cannot classify it.

Figure 4-4 Originating Traffic on a Shared Interface



Management Access to Security Contexts

The FWSM provides system administrator access in multiple context mode as well as access for individual context administrators. The following topics describe logging in as a system administrator or as a context administrator:

- System Administrator Access, page 4-9
- Context Administrator Access, page 4-10

System Administrator Access

You can access the FWSM as a system administrator in two ways:

- Session to the FWSM from the switch.

  From the switch, you access the system execution space.

- Access the admin context using Telnet, SSH, or ASDM. You can have a maximum of 15 SSH or Telnet sessions in the admin context.

  See Chapter 22, "Configuring Management Access," to enable Telnet, SSH, and ASDM access.

As the system administrator, you can access all contexts.

When you change to a context from admin or the system, your username changes to the default "enable_15" username. If you configured command authorization in that context, you need to either configure authorization privileges for the "enable_15" user, or you can log in as a different name for which you provide sufficient privileges in the command authorization configuration for the context. To log in with a username, enter the **login** command.

For example, you log in to the admin context with the username "admin." The admin context does not have any command authorization configuration, but all other contexts include command authorization. For convenience, each context configuration includes a user "admin" with maximum privileges. When you change from the admin context to context A, your username is altered, so you must log in again as "admin" by entering the **login** command. When you change to context B, you must again enter the **login** command to log in as "admin."

Context Administrator Access

You can access a context using Telnet, SSH, or ASDM. If you log in to a non-admin context, you can only access the configuration for that context. You can provide individual logins to the context. See Chapter 22, "Configuring Management Access," to enable Telnet, SSH, and ASDM access and to configure management authentication.

Enabling or Disabling Multiple Context Mode

Your FWSM might already be configured for multiple security contexts depending on how you ordered it from Cisco. If you are upgrading, however, you might need to convert from single mode to multiple mode by following the procedures in this section. ASDM does not support changing modes, so you need to change modes using the CLI.

This section includes the following topics:

Backing Up the Single Mode Configuration

When you convert from single mode to multiple mode, the FWSM converts the running configuration into two files. The original startup configuration is not saved, so if it differs from the running configuration, you should back it up before proceeding.

Enabling Multiple Context Mode

The context mode (single or multiple) is not stored in the configuration file, even though it does endure reboots. If you need to copy your configuration to another device, set the mode on the new device to match using the **mode** command.

When you convert from single mode to multiple mode, the FWSM converts the running configuration into two files: a new startup configuration that comprises the system configuration, and admin.cfg that comprises the admin context (in the root directory of the internal Flash memory). The original running configuration is saved as old_running.cfg (in the root directory of the internal Flash memory). The original startup configuration is not saved. The FWSM automatically adds an entry for the admin context to the system configuration with the name "admin."

To enable multiple mode, enter the following command:

```
hostname(config)# mode multiple
```

You are prompted to reboot the FWSM.

Restoring Single Context Mode

If you convert from multiple mode to single mode, you might want to first copy a full startup configuration (if available) to the FWSM; the system configuration inherited from multiple mode is not a complete functioning configuration for a single mode device. For example, you can restore the old single-mode running configuration, if available, as the startup configuration. Because the system configuration does not have any network interfaces as part of its configuration, you must access the FWSM from a switch session to perform the copy.

To copy the old running configuration to the startup configuration and to change the mode to single mode, perform the following steps in the system execution space:

Step 1 To copy the backup version of your original running configuration to the current startup configuration, enter the following command in the system execution space:

```
hostname(config)# copy old_running.cfg startup-config
```

Step 2 To set the mode to single mode, enter the following command in the system execution space:

```
hostname(config)# mode single
```

The FWSM reboots.

Managing Memory for Rules

The FWSM supports a fixed number of rules for the entire system. In multiple context mode, the FWSM partitions the memory allocated to rule configuration, and assigns each context to a partition. This section describes how rule allocation works and how to manage memory partitions; it includes the following topics:

About Memory Partitions

In multiple context mode, the FWSM partitions the memory allocated to rule configuration, and assigns each context to a partition. By default, a context belongs to one of 12 partitions that offers a maximum number rules, including ACEs, AAA rules, and others. See the "Default Rule Allocation" section for a list of rule limits.

The FWSM assigns contexts to the partitions in the order they are loaded at startup. For example, if you have 12 contexts and the maximum number of rules is 14,103, each context is assigned to its own partition, and can use 14,103 rules. If you add one more context, then context number 1 and the new context number 13 are both assigned to partition 1, and can use 14,103 rules divided between them; the other 11 contexts continue to use 14,103 rules each. If you delete contexts, the partition membership does not shift, so you might have some unequal distribution until you reboot, at which time the contexts are evenly distributed.

Note Rules are used up on a first come, first served basis, so one context might use more rules than another context.

You can manage memory partitions by manually assigning a context to a partition (see the "Configuring a Security Context" section on page 4-27); reducing the number of partitions to better match the number of contexts you have (see the "Setting the Number of Memory Partitions" section on page 4-13); changing the size of a partition (see the "Changing the Memory Partition Size" section on page 4-14); and reallocating rules between features (see the "Reallocating Rules Between Features for a Specific Memory Partition" section on page 4-19).

Default Rule Allocation

Table 4-1 lists the default number of rules for each feature type in multiple context mode, for the default 12 memory partitions.

Note Some access lists use more memory than others. Depending on the type of access list, the actual limit the system can support will be less than the maximum. See the "Maximum Number of ACEs" section on page 12-6 for more information about ACEs and memory usage.

Table 4-1 Default Rule Allocation

Specification	Maximum per Partition (with 12[1] Partitions)
AAA Rules	1345
ACEs	14,801
established commands[2]	96
Filter Rules	576
ICMP, Telnet, SSH, and HTTP Rules	384
Policy NAT ACEs[3]	384
Inspect Rules	1537
Total Rules	**19,219**

1. Use the **show resource rule** command to view the default values for partitions other than 12.

2. Each **established** command creates a control and data rule, so this value is doubled in the Total Rules value.
3. This limit is lower than in release 2.3.

Setting the Number of Memory Partitions

When increasing the number of partitions, the default size of each partition is reduced. If you manually configured the partition sizes (see the "Changing the Memory Partition Size" section on page 4-14), the sizes you set might not be compatible with the new smaller partition sizes. If the current configured sizes do not fit into the new partitions, then the FWSM rejects the new memory partition configuration.

The FWSM also checks the rule allocation (see the "Reallocating Rules Between Features for a Specific Memory Partition" section on page 4-19). If you manually allocated rules between features so that the total number of rules allocated is now greater than those available, then the FWSM rejects the new memory partition configuration. Similarly, if the absolute maximum number of rules for a feature is now exceeded, then the FWSM rejects the new memory partition configuration.

Note Changing the number of partitions requires you to reload the FWSM.

Guidelines

Caution Failure to follow these guidelines might result in dropped access list configuration as well as other anomalies, including ACL tree corruption.

- The target partition and rule allocation settings must be carefully calculated, planned, and preferably tested in a non-production environment prior to making the change to ensure that all existing contexts and rules can be accommodated.
- When failover is used, both FWSMs need to be reloaded at the same time after making partition changes. Reloading both FWSMs causes an outage with no possibility for a zero-downtime reload. At no time should two FWSMs with a mismatched number of partitions or rule limits synchronize over failover.

Detailed Steps

To change the number of memory partitions, perform the following steps:

Step 1 To view the current mapping of contexts to memory partitions, enter the following command:

```
hostname(config)# show resource acl-partition
```

For example, the following output shows that 2 memory partitions are configured:

```
hostname(config)# show resource acl-partition
Total number of configured partitions = 2
Partition #0
        Mode                        :exclusive
        List of Contexts          :bandn, borders
        Number of contexts    :2(RefCount:2)
        Number of rules           :0(Max:53087)
Partition #1
        Mode                        :non-exclusive
        List of Contexts          :admin, momandpopA, momandpopB, momandpopC
                                    momandpopD
```

```
Number of contexts    :5(RefCount:5)
Number of rules          :6(Max:53087)
```

For information about exclusive and non-exclusive partitions, see the "Configuring a Security Context" section on page 4-27.

Step 2 To set the number of partitions, enter the following command in the system execution space:

```
hostname(config)# resource acl-partition number_of_partitions
```

Where *number_of_partitions* is between 1 and 12.

Note The partition numbering starts with 0. So if you have 12 partitions, the partition numbers are 0 through 11. The partition number is used for customizing the memory partition and for assigning a context to a partition.

If you later enter **clear configure all** to restore the default configuration, the **resource acl-partition** command is not changed back to the default. You must enter the **no resource acl-partition** command to restore the default for this command.

You see the following message:

```
WARNING: This command leads to re-partitioning of ACL Memory.
It will not take affect until you save the configuration and reboot.
```

Step 3 To reload the FWSM so your changes can take effect, enter the following command:

```
hostname(config)# reload
```

If you are using failover, wait a few seconds before reloading the standby unit as well; the standby unit does not reload automatically, and the memory partitions must match on both units. Traffic loss can occur because both units are down at the same time.

Note If you add a secondary unit at a later date, then after the new secondary unit synchronizes the configuration, immediately reload the secondary unit so that the memory partitions are the same. During the initial synchronization, the configuration might not fit properly in the secondary unit memory partitions, but after reloading, and another configuration synchronization, the secondary unit will be operational.

Changing the Memory Partition Size

The FWSM lets you set the memory size of each partition.

Note Changing the partition sizes requires you to reload the FWSM.

Guidelines

Caution Failure to follow these guidelines might result in dropped access list configuration as well as other anomalies, including ACL tree corruption.

- The target partition and rule allocation settings must be carefully calculated, planned, and preferably tested in a non-production environment prior to making the change to ensure that all existing contexts and rules can be accommodated.
- When failover is used, both FWSMs need to be reloaded at the same time after making partition changes. Reloading both FWSMs causes an outage with no possibility for a zero-downtime reload. At no time should two FWSMs with a mismatched number of partitions or rule limits synchronize over failover.
- Change the number of partitions before you set the partition sizes; changing the number of partitions affects the overall number of rules per partition. If you increase the number of partitions, for example, then the number of rules available per partition will be smaller. Therefore, your partition size configuration might be invalid, and you might need to reconfigure all your partition sizes. Changing the number of partitions requires you to reload the FWSM before you change the partition sizes; then changing the partition sizes requires a second reload.
- Allocate contexts to specific partitions before you set the partition sizes (see the "Configuring a Security Context" section on page 4-27). If you plan all your partition sizes based on the contexts currently assigned to a partition, but you did not specifically allocate the contexts, then you run the risk of context assignments shifting after a reload (for example if you add or subtract contexts).
- Reduce the size of partition(s) before increasing the size of other partition(s). The FWSM rejects any increases in size if there is not free space available.
- If the existing number of ACEs does not fit into the new partition size, then the resizing is rejected.
- In addition to the memory partitions to which the FWSM assigns contexts, the FWSM uses a backup tree partition to process changes to rules so traffic can continue to use the old configuration until the new configuration is ready. The backup tree must be as large as the largest partition. Therefore, some memory is automatically assigned to the backup tree in tandem with the largest partition; so be sure to include the backup tree in your calculations.
- If you reduce the size of a partition, the FWSM checks the rule allocation (see the "Reallocating Rules Between Features for a Specific Memory Partition" section on page 4-19). If you manually allocated rules between features so that the total number of rules allocated is now greater than those available, then the FWSM rejects the resizing of the partition. Similarly, if the absolute maximum number of rules for a feature is now exceeded, then the FWSM rejects the resizing of the partition.

Detailed Steps

To set the size of the memory partitions, perform the following steps:

Step 1 To view the current partition sizes, enter the following command:

```
hostname(config)# show resource partition
```

For example, the following output shows that each of 12 partitions have the default 19,219 rules (this is an example only, and might differ from the actual number of rules for your system). The backup tree always matches the largest partition size, so it also has 19,219 rules, for a total of 249,847 rules.

```
hostname(config)# show resource partition
```

```
                          Bootup     Current
 Partition    Default    Partition   Configured
  Number       Size       Size        Size
-----------+---------+----------+-----------
      0       19219      19219       19219
      1       19219      19219       19219
      2       19219      19219       19219
      3       19219      19219       19219
      4       19219      19219       19219
      5       19219      19219       19219
      6       19219      19219       19219
      7       19219      19219       19219
      8       19219      19219       19219
      9       19219      19219       19219
     10       19219      19219       19219
     11       19219      19219       19219
backup tree   19219      19219       19219
-----------+---------+----------+-----------
   Total     249847     249847      249847

Total Partition size - Configured size = Available to allocate
        249847       -     249847      =           0
```

You can also view the current mapping of contexts to partitions using the **show resource acl-partition** command.

Step 2 To identify the partition you want to reduce in size, enter the following command in the system execution space:

```
hostname(config)# resource partition number
```

Where *number* is between 0 and 11 by default. If you changed the number of partitions, the partition numbering starts with 0. So if you have 10 partitions, the partition numbers are 0 through 9.

Step 3 To reduce the partition size, enter the following command:

```
hostname(config-partition)# size number_of_rules
```

Where *number* is the number of rules you want to assign to the partition, in this case a lower number than was shown in the **show resource partition** command. Use the **no** form of this command to return to the default.

Step 4 To reduce the size of other partitions, repeat Steps 2 and 3.

Step 5 To view the rules now available for increasing partition sizes, enter the **show resource partition** command.

For example, if you reduced the sizes of partitions 0 through 5 to 15,000, then the output shows that you have 25,314 rules to reallocate to other partitions.

```
hostname(config)# show resource partition

                          Bootup     Current
 Partition    Default    Partition   Configured
  Number       Size       Size        Size
-----------+---------+----------+-----------
      0       19219      19219       15000
      1       19219      19219       15000
      2       19219      19219       15000
      3       19219      19219       15000
      4       19219      19219       15000
      5       19219      19219       15000
      6       19219      19219       19219
      7       19219      19219       19219
      8       19219      19219       19219
```

```
      9       19219      19219       19219
     10       19219      19219       19219
     11       19219      19219       19219
backup tree   19219      19219       19219
-----------+---------+----------+-----------
   Total     249847     249847      224533

Total Partition size - Configured size = Available to allocate
        249847       -     224533 =              25314
```

If you want to distribute the rules evenly across the other 6 partitions plus the backup tree, then you can add 3616 rules to each (with 2 left over). Remember that the backup tree must be as large as the largest partition, so you must consider the backup tree in your calculations. For example, if you want to make partition 6 have 24,001 rules, then you can allocate the rules like this:

Partition	Bootup Partition Size	Configured Size	Difference
6	19219	24001	4782
Backup Tree	19219	24001	4782
7	19219	22369	3150
8	19219	22369	3150
9	19219	22369	3150
10	19219	22369	3150
11	19219	22369	3150
			Total: 25314

Step 6 To identify the partition you want to increase in size, enter the following command in the system execution space:

```
hostname(config)# resource partition number
```

Where *number* is between 0 and 11 by default. If you changed the number of partitions, the partition numbering starts with 0. So if you have 10 partitions, the partition numbers are 0 through 9.

Step 7 To increase the partition size, enter the following command:

```
hostname(config-partition)# size number_of_rules
```

Where *number* is the number of rules you want to assign to the partition, in this case a higher number than was shown in the **show resource partition** command. Use the **no** form of this command to return to the default.

Step 8 To increase the size of other partitions, repeat Steps 6 and 7.

Step 9 To reload the FWSM so your changes can take effect, enter the following command:

```
hostname(config)# reload
```

If you are using failover, wait a few seconds before reloading the standby unit as well; the standby unit does not reload automatically, and the memory partition sizes must match on both units. Traffic loss can occur because both units are down at the same time.

Note If you add a secondary unit at a later date, then after the new secondary unit synchronizes the configuration, immediately reload the secondary unit so that the memory partitions are the same. During the initial synchronization, the configuration might not fit properly in the secondary unit memory partitions, but after reloading, and another configuration synchronization, the secondary unit will be operational.

For example, if you have 4 partitions, and you want to reduce partitions 0 and 1 to 40000, while increasing partitions 2 and 3 to 56616 and 56615 respectively, enter the following commands:

```
hostname(config)# show resource partition

                        Bootup     Current
 Partition   Default  Partition   Configured
  Number       Size      Size        Size
-----------+---------+----------+-----------
      0       49970      49970       49970
      1       49969      49969       49969
      2       49969      49969       49969
      3       49969      49969       49969
backup tree   49970      49970       49970
-----------+---------+----------+-----------
   Total     249847     249847      249847

Total Partition size - Configured size = Available to allocate
              249847 -          249847 =          0

hostname(config)# resource partition 0
hostname(config-partition)# size 40000
hostname(config-partition)# resource partition 1
hostname(config-partition)# size 40000

hostname(config-partition)# show resource partition

                        Bootup     Current
 Partition   Default  Partition   Configured
  Number       Size      Size        Size
-----------+---------+----------+-----------
      0       49970      49970       40000
      1       49969      49969       40000
      2       49969      49969       49969
      3       49969      49969       49969
backup tree   49970      49970       49969
-----------+---------+----------+-----------
   Total     249847     249847      249847

Total Partition size - Configured size = Available to allocate
              249847 -          229907 =          19940

hostname(config-partition)# resource partition 2
hostname(config-partition)# size 56616
hostname(config-partition)# resource partition 3
hostname(config-partition)# size 56615

hostname(config-partition)# show resource partition

                        Bootup     Current
 Partition   Default  Partition   Configured
  Number       Size      Size        Size
```

```
-----------+---------+----------+-----------
      0       49970      49970       40000
      1       49969      49969       40000
      2       49969      49969       56616
      3       49969      49969       56615
backup tree   49970      49970       56616
-----------+---------+----------+-----------
   Total     249847     249847      249847

Total Partition size - Configured size = Available to allocate
              249847 -          249847 =          0

hostname(config-partition)# reload
```

Reallocating Rules Between Features for a Specific Memory Partition

To set the rule allocation globally for all partitions, see the "Reallocating Rules Between Features" section on page A-8. Setting the rule allocation for a specific partition overrides the global setting.

Guidelines

Caution Failure to follow these guidelines might result in dropped access list configuration as well as other anomalies, including ACL tree corruption.

- The target partition and rule allocation settings must be carefully calculated, planned, and preferably tested in a non-production environment prior to making the change to ensure that all existing contexts and rules can be accommodated.
- When failover is used, both FWSMs need to be reloaded at the same time after making partition changes. Reloading both FWSMs causes an outage with no possibility for a zero-downtime reload. At no time should two FWSMs with a mismatched number of partitions or rule limits synchronize over failover.

Detailed Steps

To reallocate rules for a given partition, perform the following steps:

Step 1 To view the total number of rules available per partition, the default values, current rule allocation, and the absolute maximum number of rules you can allocate per feature, enter the following command:

```
hostname(config)# show resource rule partition [number]
```

For example, the following display shows the maximum rules as 19219 for partition 0 (this is an example only, and might differ from the actual number of rules for your system):

```
hostname(config)# show resource rule partition 0

             Default  Configured  Absolute
 CLS Rule     Limit      Limit      Max
-----------+---------+----------+---------
 Policy NAT     384        384       833
 ACL          14801      14801     14801
 Filter         576        576      1152
 Fixup         1537       1537      3074
 Est Ctl         96         96        96
```

```
 Est Data         96          96          96
 AAA            1345        1345        2690
 Console         384         384         768
-----------+---------+----------+---------
 Total         19219       19219

Partition Limit - Configured Limit = Available to allocate
      19219     -      19219       =           0
```

Note If you increase the size of a partition but have not yet reloaded, the maximum number of rules remains at the old smaller size. You have to reload to see the increased limits. If you decrease the size of a partition but have not yet reloaded, the new smaller number of rules is reflected right away.

Step 2 To view the number of rules currently being used so you can plan your reallocation, enter the following command:

```
hostname(config)# show np 3 acl count partition_number
```

Where *partition_number* is between 0 and 11 by default. If you changed the number of partitions, the partition numbering starts with 0. So if you have 10 partitions, the partition numbers are 0 through 9.

For example, the following is sample output from the **show np 3 acl count** command, and shows the number of inspections (Fixup Rule) close to the maximum of 9216. You might choose to reallocate some access list rules (ACL Rule) to inspections.

```
hostname(config)# show np 3 acl count 0

-------------- CLS Rule Current Counts --------------
CLS Filter Rule Count       :             0
CLS Fixup Rule Count        :          9001
CLS Est Ctl Rule Count      :             4
CLS AAA Rule Count          :            15
CLS Est Data Rule Count     :             4
CLS Console Rule Count      :            16
CLS Policy NAT Rule Count   :             0
CLS ACL Rule Count          :         30500
CLS ACL Uncommitted Add     :             0
CLS ACL Uncommitted Del     :             0
...
```

Note The **established** command creates two types of rules, control and data. Both of these types are shown in the display, but you allocate both rules by setting the number of **established** commands; you do not set each rule separately.

Step 3 To identify the partition you want to customize, enter the following command in the system execution space:

```
hostname(config)# resource partition number
```

Where *number* is between 0 and 11 by default. If you changed the number of partitions, the partition numbering starts with 0. So if you have 10 partitions, the partition numbers are 0 through 9.

Step 4 To reallocate rules between features, enter the following command. If you increase the value for one feature, then you must decrease the value by the same amount for one or more features so the total number of rules does not exceed the system limit. See Step 1 to use the **show resource rule** command for the total number of rules allowed.

```
hostname(config-partition)# rule nat {max_policy_nat_rules | current | default | max}
acl {max_ace_rules | current | default | max}
filter {max_filter_rules | current | default | max}
fixup {max_inspect_rules | current | default | max}
est {max_established_rules | current | default | max}
aaa {max_aaa_rules | current | default | max}
console {max_console_rules | current | default | max}
```

You must enter all arguments in this command. This command takes effect immediately.

The **nat** *max_nat_rules* arguments set the maximum number of policy NAT ACEs, between 0 and 10000.

The **acl** *max_nat_rules* arguments set the maximum number of ACEs, between 0 and the system limit. The system limit depends on how many memory partitions you configured. See Step 1 to use the **show resource rule** command.

The **filter** *max_nat_rules* arguments set the maximum number of filter rules, between 0 and 6000.

The **fixup** *max_nat_rules* arguments set the maximum number of inspect rules, between 0 and 10000.

The **est** *max_nat_rules* arguments set the maximum number of **established** commands, between 0 and 716. The established command creates two types of rules, control and data. Both of these types are shown in the **show np 3 acl count** and **show resource rules** display, but you set both rules using the **est** keyword, which correlates with the number of **established** commands. Be sure to double the value you enter here when comparing the total number of configured rules with the total number of rules shown in the **show** commands.

The **aaa** *max_nat_rules* arguments set the maximum number of AAA rules, between 0 and 10000.

The **console** *max_nat_rules* arguments set the maximum number of ICMP, Telnet, SSH, and HTTP rules, between 0 and 4000.

The **current** keyword keeps the current value set.

The **default** keyword sets the maximum rules to the default.

The **max** keyword sets the rules to the maximum allowed for the feature. Be sure to set other features lower to accommodate this value.

For example for partition 0, to reallocate 999 rules from the default 14,801 ACEs to inspections (default 9001), enter the following command:

```
hostname(config)# resource partition 0
hostname(config-partition)# rule nat default acl 13802 filter default fixup 10000 est
default aaa default console default
```

Configuring Resource Management

By default, all security contexts have unlimited access to the resources of the FWSM, except where maximum limits per context are enforced. However, if you find that one or more contexts use too many resources, and they cause other contexts to be denied connections, for example, then you can configure resource management to limit the use of resources per context.

Note The FWSM does not limit the bandwidth per context; however, the switch containing the FWSM can limit bandwidth per VLAN. See the switch documentation for more information.

This section includes the following topics:

Classes and Class Members Overview

The FWSM manages resources by assigning contexts to resource classes. Each context uses the resource limits set by the class. This section includes the following topics:

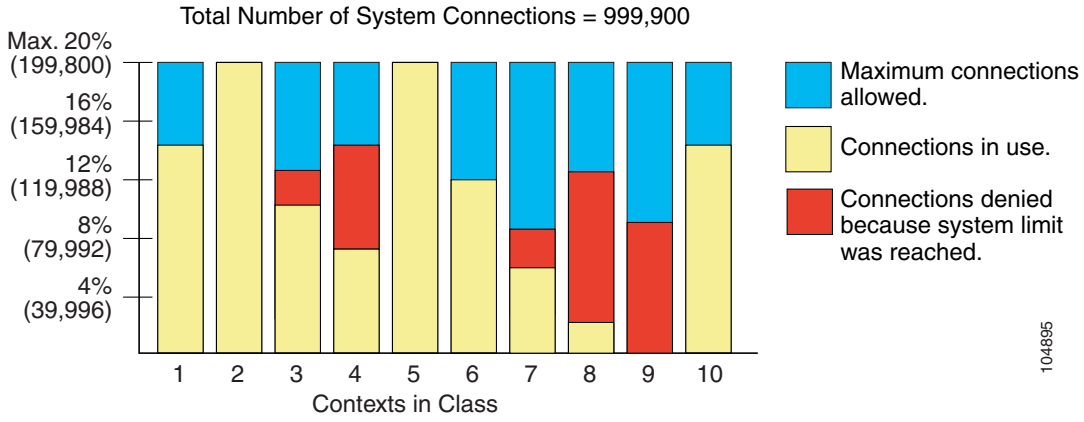
Resource Limits

When you create a class, the FWSM does not set aside a portion of the resources for each context assigned to the class; rather, the FWSM sets the maximum limit for a context. If you oversubscribe resources, or allow some resources to be unlimited, a few contexts can "use up" those resources, potentially affecting service to other contexts.

You can set the limit for all resources together as a percentage of the total available for the device. Also, you can set the limit for individual resources as a percentage or as an absolute value.

You can oversubscribe the FWSM by assigning more than 100 percent of the resources across all contexts. For example, you can set the Bronze class to limit connections to 20 percent per context, and then assign 10 contexts to the class for a total of 200 percent. If contexts concurrently use more than the system limit, then each context gets less than the 20 percent you intended. (See Figure 4-5.)

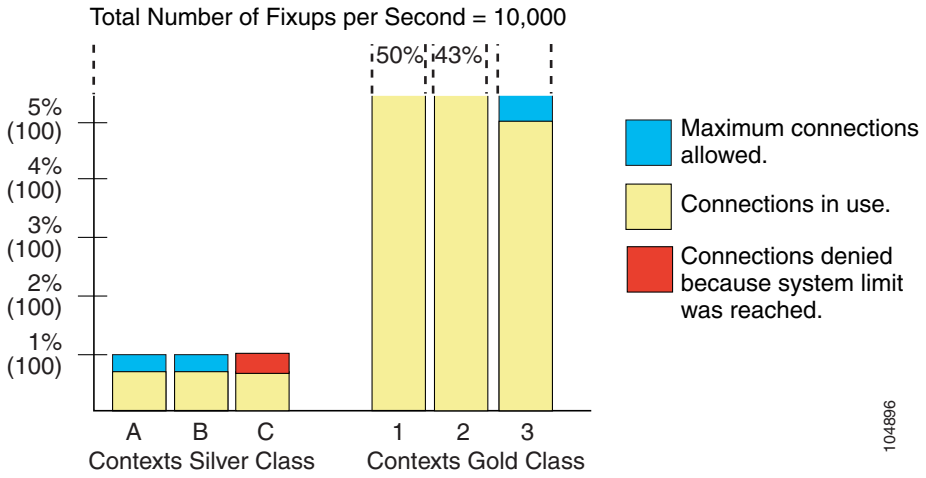
Figure 4-5 Resource Oversubscription



The FWSM lets you assign unlimited access to one or more resources in a class, instead of a percentage or absolute number. When a resource is unlimited, contexts can use as much of the resource as the system has available. For example, Context A, B, and C are in the Silver Class, which limits each class member to 1 percent of the system inspections per second, for a total of 3 percent; but the three contexts are currently only using 2 percent combined. Gold Class has unlimited access to inspections. The contexts in the Gold Class can use more than the 97 percent of "unassigned" inspections; they can also use the 1 percent of inspections not currently in use by Context A, B, and C, even if that means that Context A,

B, and C are unable to reach their 3 percent combined limit. (See Figure 4-6.) Setting unlimited access is similar to oversubscribing the FWSM, except that you have less control over how much you oversubscribe the system.

Figure 4-6 Unlimited Resources



Default Class

All contexts belong to the default class if they are not assigned to another class; you do not have to actively assign a context to the default class.

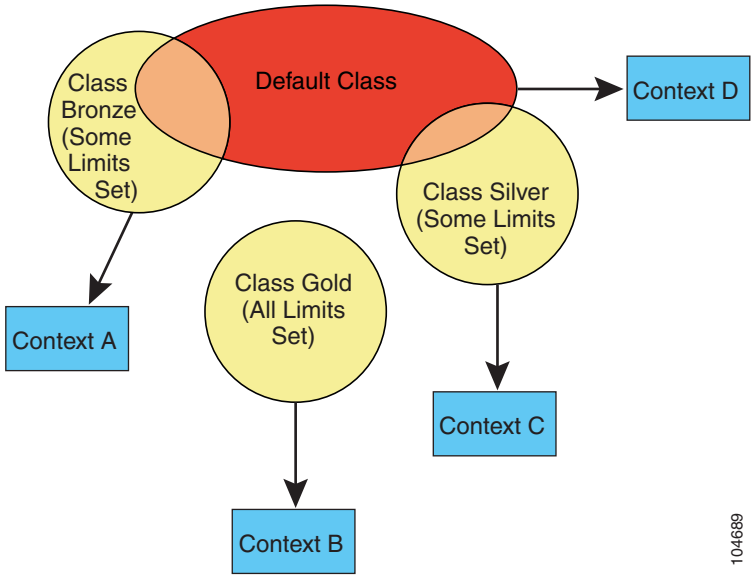
If a context belongs to a class other than the default class, those class settings always override the default class settings. However, if the other class has any settings that are not defined, then the member context uses the default class for those limits. For example, if you create a class with a 2 percent limit for all concurrent connections, but no other limits, then all other limits are inherited from the default class. Conversely, if you create a class with a 2 percent limit for *all* resources, the class uses no settings from the default class.

By default, the default class provides unlimited access to resources for all contexts, except for the following limits, which are by default set to the maximum allowed per context:

- Telnet sessions—5 sessions.
- SSH sessions—5 sessions.
- IPSec sessions—5 sessions.
- MAC addresses—65,535 entries.

Figure 4-7 shows the relationship between the default class and other classes. Contexts A and C belong to classes with some limits set; other limits are inherited from the default class. Context B inherits no limits from default because all limits are set in its class, the Gold class. Context D was not assigned to a class, and is by default a member of the default class.

Figure 4-7 Resource Classes



Class Members

To use the settings of a class, assign the context to the class when you define the context. All contexts belong to the default class if they are not assigned to another class; you do not have to actively assign a context to default. You can only assign a context to one resource class. The exception to this rule is that limits that are undefined in the member class are inherited from the default class; so in effect, a context could be a member of default plus another class.

Configuring a Class

To configure a class in the system configuration, perform the following steps. You can change the value of a particular resource limit by reentering the command with a new value.

Step 1 To specify the class name and enter the class configuration mode, enter the following command in the system execution space:

```
hostname(config)# class name
```

The *name* is a string up to 20 characters long. To set the limits for the default class, enter **default** for the name.

Step 2 To set the resource limits, see the following options:

- To set all resource limits (shown in Table 4-2), enter the following command:

```
hostname(config-resmgmt)# limit-resource all {number% | 0}
```

The *number* is an integer greater than or equal to 1. **0** (without a percent sign (%)) sets the resources to the system limit. You can assign more than 100 percent if you want to oversubscribe the device.

- To set a particular resource limit, enter the following command:

```
hostname(config-resmgmt)# limit-resource [rate] resource_name number[%]
```

For this particular resource, the limit overrides the limit set for **all**. Enter the **rate** argument to **s**et the rate per second for certain resources. See Table 4-2 for resources for which you can set the rate per second.

Table 4-2 lists the resource types and the limits. See also the **show resource types** command.

Table 4-2 Resource Names and Limits

Resource Name	Minimum and Maximum Number per Context	Total Number for System	Description
mac-addresses	N/A	65,535 concurrent	For transparent firewall mode, the number of MAC addresses allowed in the MAC address table.
conns	N/A	999,900 concurrent 102,400 per second (rate)	TCP or UDP connections between any two hosts, including connections between one host and multiple other hosts. **Note** For concurrent connections, the FWSM allocates half of the limit to each of two network processors that accept connections. Typically, the connections are divided evenly between the NPs. However, in some circumstances, the connections are not evenly divided, and you might reach the maximum connection limit on one NP before reaching the maximum on the other. In this case, the maximum connections allowed is less than the limit you set. The NP distribution is controlled by the switch based on an algorithm. You can adjust this algorithm on the switch, or you can adjust the connection limit upward to account for the inequity.
fixups	N/A	10,000 per second (rate)	Application inspection.
hosts	N/A	262,144 concurrent	Hosts that can connect through the FWSM.
ipsec	1 minimum 5 maximum concurrent	10 concurrent	IPSec sessions.
asdm	1 minimum 5 maximum concurrent	80 concurrent	ASDM management sessions. **Note** ASDM sessions use two HTTPS connections: one for monitoring that is always present, and one for making configuration changes that is present only when you make changes. For example, the system limit of 80 ASDM sessions represents a limit of 160 HTTPS sessions.
ssh	1 minimum 5 maximum concurrent	100 concurrent	SSH sessions.

Table 4-2 Resource Names and Limits (continued)

Resource Name	Minimum and Maximum Number per Context	Total Number for System	Description
syslogs	N/A	30,000 per second (rate)	System log messages. **Note** The FWSM can support 30,000 messages per second for messages sent to the FWSM terminal or buffer. If you send messages to a syslog server, the FWSM supports 25,000 per second.
telnet	1 minimum 5 maximum concurrent	100 concurrent	Telnet sessions.
xlates	N/A	266,144 concurrent	Address translations.

For example, to set the default class limit for conns to 10 percent instead of unlimited, enter the following commands:

```
hostname(config)# class default
hostname(config-class)# limit-resource conns 10%
```

All other resources remain at unlimited.

To add a class called gold with all resources set to 5 percent, except for fixups, with a setting of 10 percent, enter the following commands:

```
hostname(config)# class gold
hostname(config-class)# limit-resource all 5%
hostname(config-class)# limit-resource fixups 10%
```

To add a class called silver with all resources set to 3 percent, except for syslogs, with a setting of 500 per second, enter the following commands:

```
hostname(config)# class silver
hostname(config-class)# limit-resource all 3%
hostname(config-class)# limit-resource rate syslogs 500
```

Configuring a Security Context

The security context definition in the system configuration identifies the context name, configuration file URL, interfaces that a context can use, and other context parameters.

Note To assign a context to a failover group for active/active failover, see the "Using Active/Active Failover" section on page 13-26.

If you do not have an admin context (for example, if you clear the configuration) then you must first specify the admin context name by entering the following command:

```
hostname(config)# admin-context name
```

Although this context name does not yet exist in your configuration, you can subsequently enter the **context** *name* command to match the specified name to continue the admin context configuration.

To configure a context in the system configuration, perform the following steps:

Step 1 To configure a context, enter the following command in the system execution space:

```
hostname(config)# context name
```

The *name* is a string up to 32 characters long. This name is case sensitive, so you can have two contexts named "customerA" and "CustomerA," for example. You can use letters, digits, or hyphens, but you cannot start or end the name with a hyphen.

"System" or "Null" (in upper or lower case letters) are reserved names, and cannot be used.

Step 2 (Optional) To add a description for this context, enter the following command:

```
hostname(config-ctx)# description text
```

Step 3 To specify the interfaces you can use in the context, enter the following command:

```
hostname(config-ctx)# allocate-interface vlannumber[-vlannumber] [map_name[-map_name]
[invisible | visible]]
```

You can enter this command multiple times to specify different ranges. If you remove an allocation with the **no** form of this command, then any context commands that include this interface are removed from the running configuration.

Enter a VLAN number or a range of VLANs, typically from 2 to 1000 and from 1025 to 4094 (see the switch documentation for supported VLANs). To see a list of VLANs assigned to the FWSM, use the **show vlan** command. You can allocate a VLAN that is not yet assigned to the FWSM, but you need to assign them from the switch if you want them to pass traffic. When you allocate an interface, the FWSM automatically adds the **interface** command for each VLAN in the system configuration.

You can assign the same VLANs to multiple contexts in routed mode, if desired. See the "Sharing Interfaces Between Contexts" section on page 4-7 for more information about shared VLAN limitations.

The *map_name* is an alphanumeric alias for the interface that can be used within the context instead of the VLAN ID. If you do not specify a mapped name, the VLAN ID is used within the context. For security purposes, you might not want the context administrator to know which interfaces are being used by the context. You can use the same name in multiple contexts; the VLAN ID in multiple contexts can be the same or different for a given name. You cannot use the same name for different VLAN IDs in the same context.

A mapped name must start with a letter, end with a letter or digit, and have as interior characters only letters, digits, or an underscore. For example, you can use the following names:

```
int0

inta

int_0
```

If you specify a range of VLAN IDs, you can specify a matching range of mapped names. Follow these guidelines for ranges:

- The mapped name must consist of an alphabetic portion followed by a numeric portion. The alphabetic portion of the mapped name must match for both ends of the range. For example, enter the following range:

  ```
  int0-int10
  ```

- The numeric portion of the mapped name must include the same quantity of numbers as the **vlan***x*-**vlan***y* statement. For example, both ranges include 100 interfaces:

  ```
  vlan100-vlan199 int1-int100
  ```

  If you enter **vlan100-vlan199 int1-int15** or **vlan100-vlan199 happy1-sad5**, for example, the command fails.

If you set a mapped name, specify **visible** to see the VLAN ID in addition to the mapped name in the **show interface** command. The default **invisible** keyword specifies to only show the mapped name.

The following example shows VLANs 100, 200, and 300 through 305 assigned to the context. The mapped names are int1 through int8.

```
hostname(config-ctx)# allocate-interface vlan100 int1
hostname(config-ctx)# allocate-interface vlan200 int2
hostname(config-ctx)# allocate-interface vlan300-vlan305 int3-int8
```

Step 4 To identify the URL from which the system downloads the context configuration, enter the following command:

```
hostname(config-ctx)# config-url url
```

When you add a context URL, the system immediately loads the context so that it is running, if the configuration is available.

Note Enter the **allocate-interface** command(s) before you enter the **config-url** command. The FWSM must assign interfaces to the context before it loads the context configuration; the context configuration might include commands that refer to interfaces (**interface**, **nat**, **global**...). If you enter the **config-url** command first, the FWSM loads the context configuration immediately. If the context contains any commands that refer to interfaces, those commands fail.

See the following URL syntax:

- **disk:/**[*path*/]*filename*

  This URL indicates the internal Flash memory. The filename does not require a file extension, although we recommend using ".cfg". If the configuration file is not available, you see the following message:

  ```
  WARNING: Could not fetch the URL disk:/url
  INFO: Creating context with default config
  ```

  You can then change to the context, configure it at the CLI, and enter the **write memory** command to write the file to Flash memory.

  Note The admin context file must be stored on the internal Flash memory.

- **ftp://**[*user*[**:***password*]@]*server*[:*port*]**/**[*path*/]*filename*[**;type=***xx*]

  The **type** can be one of the following keywords:

  - **ap**—ASCII passive mode
  - **an**—ASCII normal mode
  - **ip**—(Default) Binary passive mode
  - **in**—Binary normal mode

The server must be accessible from the admin context. The filename does not require a file extension, although we recommend using ".cfg". If the configuration file is not available, you see the following message:

```
WARNING: Could not fetch the URL ftp://url
INFO: Creating context with default config
```

You can then change to the context, configure it at the CLI, and enter the **write memory** command to write the file to the FTP server.

- **http**[**s**]**://**[*user*[**:***password*]@]*server*[:*port*]**/**[*path/*]*filename*

  The server must be accessible from the admin context. The filename does not require a file extension, although we recommend using ".cfg". If the configuration file is not available, you see the following message:

  ```
  WARNING: Could not fetch the URL http://url
  INFO: Creating context with default config
  ```

  If you change to the context and configure the context at the CLI, you cannot save changes back to HTTP or HTTPS servers using the **write memory** command. You can, however, use the **copy tftp** command to copy the running configuration to a TFTP server.

- **tftp://**[*user*[**:***password*]@]*server*[:*port*]**/**[*path/*]*filename*[**;int=***interface_name*]

  The server must be accessible from the admin context. Specify the interface name if you want to override the route to the server address. The filename does not require a file extension, although we recommend using ".cfg". If the configuration file is not available, you see the following message:

  ```
  WARNING: Could not fetch the URL tftp://url
  INFO: Creating context with default config
  ```

  You can then change to the context, configure it at the CLI, and enter the **write memory** command to write the file to the TFTP server.

To change the URL, reenter the **config-url** command with a new URL.

See the "Changing the Security Context URL" section on page 4-33 for more information about changing the URL.

For example, enter the following command:

```
hostname(config-ctx)# config-url ftp://user1:passw0rd1@10.1.1.1/configlets/test.cfg
```

Step 5 (Optional) To assign the context to a resource class, enter the following command:

```
hostname(config-ctx)# member class_name
```

If you do not specify a class, the context belongs to the default class. You can only assign a context to one resource class.

For example, to assign the context to the gold class, enter the following command:

```
hostname(config-ctx)# member gold
```

Step 6 (Optional) To map a context to a specific memory partition, enter the following command:

```
hostname(config-ctx)# allocate-acl-partition partition_number
```

The *partition_number* is an integer from 0 to the number of partitions available, minus 1. The default is 12 partitions, so the range is 0 to 11. See the "Setting the Number of Memory Partitions" section on page 4-13 to configure the number of memory partitions.

When you assign a context to a partition, then the partition becomes *exclusive*. An exclusive partition only includes contexts that you specifically assign to it. Partitions that do not have contexts specifically assigned to them are non-exclusive and contexts are allocated to them in a round-robin fashion.

Note If you assign contexts to all partitions, then they are all exclusive. However, if you later add a context that is not assigned to a partition, then contexts are allocated to exclusive partitions in a round-robin fashion, and the first best-fit exclusive partition available is used for the allocation of the new context. However, if none of the exclusive partitions can accommodate the rules of the new context, then it is assigned to partition 0 by default, even though partition 0 also cannot accommodate the context rules. The context rules will not load completely, so you need to manually adjust the way contexts are assigned tomake room.

For example, to assign the context to the first partition, enter the following command:

```
hostname(config-ctx)# allocate-acl-partition 0
```

The following example sets the admin context to be "administrator," creates a context called "administrator" on the internal Flash memory, and then adds two contexts from an FTP server:

```
hostname(config)# admin-context administrator
hostname(config)# context administrator
hostname(config-ctx)# allocate-interface vlan10
hostname(config-ctx)# allocate-interface vlan11
hostname(config-ctx)# config-url disk:/admin.cfg

hostname(config-ctx)# context test
hostname(config-ctx)# allocate-interface vlan100 int1
hostname(config-ctx)# allocate-interface vlan102 int2
hostname(config-ctx)# allocate-interface vlan110-vlan115 int3-int8
hostname(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/test.cfg
hostname(config-ctx)# member gold
hostname(config-ctx)# allocate-acl-partition 0


hostname(config-ctx)# context sample
hostname(config-ctx)# allocate-interface vlan200 int1
hostname(config-ctx)# allocate-interface vlan212 int2
hostname(config-ctx)# allocate-interface vlan230-vlan235 int3-int8
hostname(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/sample.cfg
hostname(config-ctx)# member silver
```

Changing Between Contexts and the System Execution Space

If you log in to the system execution space (or the admin context using Telnet or SSH), you can change between contexts and perform configuration and monitoring tasks within each context. The running configuration that you edit in a configuration mode, or that is affected by the **copy** or **write** commands, depends on your location. When you are in the system execution space, the running configuration consists only of the system configuration; when you are in a context, the running configuration consists only of that context. For example, you cannot view all running configurations (system plus all contexts) by entering the **show running-config** command. Only the current configuration displays. You can, however, save all context running configurations from the system execution space using the **write memory all** command.

For information about command authorization when you change between contexts, see the "Management Access to Security Contexts" section on page 4-9.

To change between the system execution space and a context, or between contexts, see the following commands:

- To change to a context, enter the following command:

  ```
  hostname# changeto context name
  ```

  The prompt changes to the following:

  ```
  hostname/name#
  ```

- To change to the system execution space, enter the following command:

  ```
  hostname/admin# changeto system
  ```

  The prompt changes to the following:

  ```
  hostname#
  ```

Managing Security Contexts

This section describes how to manage security contexts, and includes the following topics:

Removing a Security Context

You can only remove a context by editing the system configuration. You cannot remove the current admin context, unless you remove all contexts using the **clear context** command.

Note If you use failover, there is a delay between when you remove the context on the active unit or group and when the context is removed on the standby unit or group. You might see an error message indicating that the number of interfaces on the active and standby units are not consistent; this error is temporary and can be ignored.

Use the following commands for removing contexts:

- To remove a single context, enter the following command in the system execution space:

  ```
  hostname(config)# no context name
  ```

- To remove all contexts (including the admin context), enter the following command in the system execution space:

  ```
  hostname(config)# clear context
  ```

Changing the Admin Context

The system configuration does not include any network interfaces or network settings for itself; rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses one of the contexts that is designated as the admin context.

The admin context is just like any other context, except that when a user logs in to the admin context, then that user has system administrator rights and can access the system and all other contexts. The admin context is not restricted in any way, and can be used as a regular context. However, because logging into the admin context grants you administrator privileges over all contexts, you might need to restrict access to the admin context to appropriate users.

You can set any context to be the admin context, as long as the configuration file is stored in the internal Flash memory. To set the admin context, enter the following command in the system execution space:

```
hostname(config)# admin-context context_name
```

Any remote management sessions, such as Telnet, SSH, or HTTPS, that are connected to the admin context are terminated. You must reconnect to the new admin context.

Note A few system commands identify an interface name that belongs to the admin context. If you change the admin context, and that interface name does not exist in the new admin context, be sure to update any system commands that refer to the interface.

Changing the Security Context URL

You cannot change the security context URL without reloading the configuration from the new URL.

The FWSM merges the new configuration with the current running configuration. Reentering the same URL also merges the saved configuration with the running configuration. A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results. If the running configuration is blank (for example, if the server was unavailable and the configuration was never downloaded), then the new configuration is used. If you do not want to merge the configurations, you can clear the running configuration, which disrupts any communications through the context, and then reload the configuration from the new URL.

To change the URL for a context, perform the following steps:

Step 1 If you do not want to merge the configuration, change to the context and clear its configuration by entering the following commands. If you want to perform a merge, skip to Step 2.

```
hostname# changeto context name
hostname/name# configure terminal
hostname/name(config)# clear configure all
```

Step 2 If required, change to the system execution space by entering the following command:

```
hostname/name(config)# changeto system
```

Step 3 To enter the context configuration mode for the context you want to change, enter the following command:

```
hostname(config)# context name
```

Step 4 To enter the new URL, enter the following command:

```
hostname(config)# config-url new_url
```

The system immediately loads the context so that it is running.

Reloading a Security Context

You can reload the context in two ways:

- Clear the running configuration and then import the startup configuration.

  This action clears most attributes associated with the context, such as connections and NAT tables.

- Remove the context from the system configuration.

  This action clears additional attributes, such as memory allocation, which might be useful for troubleshooting. However, to add the context back to the system requires you to respecify the URL and interfaces.

This section includes the following topics:

Reloading by Clearing the Configuration

To reload the context by clearing the context configuration, and reloading the configuration from the URL, perform the following steps:

Step 1 To change to the context that you want to reload, enter the following command:

```
hostname# changeto context name
```

Step 2 To access configuration mode, enter the following command:

```
hostname/name# configure terminal
```

Step 3 To clear the running configuration, enter the following command:

```
hostname/name(config)# clear configure all
```

This command clears all connections.

Step 4 To reload the configuration, enter the following command:

```
hostname/name(config)# copy startup-config running-config
```

The FWSM copies the configuration from the URL specified in the system configuration. You cannot change the URL from within a context.

Reloading by Removing and Readding the Context

To reload the context by removing the context and then readding it, perform the steps in the following topics:

1. "Removing a Security Context" section on page 4-32
2. "Configuring a Security Context" section on page 4-27

Monitoring Security Contexts

This section describes how to view and monitor context information, and includes the following topics:

Viewing Context Information

From the system execution space, you can view a list of contexts including the name, allocated interfaces, and configuration file URL.

From the system execution space, view all contexts by entering the following command:

```
hostname# show context [name | detail| count]
```

The **detail** option shows additional information. See the following sample displays for more information.

If you want to show information for a particular context, specify the *name*.

The **count** option shows the total number of contexts.

The following is sample output from the **show context** command. The following sample display shows three contexts:

```
hostname# show context
Context Name      Class       Interfaces          Mode          URL
*admin            default     Vlan100,101         Routed        disk:/admin.cfg
contexta          Gold        Vlan200,201         Transparent   disk:/contexta.cfg
contextb          Silver      Vlan300,301         Routed        disk:/contextb.cfg
Total active Security Contexts: 3
```

Table 4-3 shows each field description.

Table 4-3 show context Fields

Field	Description
Context Name	Lists all context names. The context name with the asterisk (*) is the admin context.
Class	Shows the resource class to which the context belongs.
Interfaces	Shows the interfaces assigned to the context.
Mode	Shows the firewall mode for each context, either Routed or Transparent.
URL	Shows the URL from which the FWSM loads the context configuration.

The following is sample output from the **show context detail** command:

```
hostname# show context detail

Context "admin", has been created, but initial ACL rules not complete
  Config URL: disk:/admin.cfg
  Real Interfaces: Vlan100
  Mapped Interfaces: Vlan100
  Class: default, Flags: 0x00000013, ID: 1

Context "ctx", has been created, but initial ACL rules not complete
  Config URL: disk:/ctx.cfg
  Real Interfaces: Vlan10,20,30
  Mapped Interfaces: int1, int2, int3
  Class: default, Flags: 0x00000011, ID: 2

Context "system", is a system resource
  Config URL: startup-config
  Real Interfaces:
  Mapped Interfaces: Vlan100,10,20,30
  Class: default, Flags: 0x00000019, ID: 257

Context "null", is a system resource
  Config URL: ... null ...
  Real Interfaces:
  Mapped Interfaces:
  Class: default, Flags: 0x00000009, ID: 258
```

See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about the **detail** output.

The following is sample output from the **show context count** command:

```
hostname# show context count
Total active contexts: 2
```

Viewing Resource Allocation

From the system execution space, you can view the allocation for each resource across all classes and class members.

To view the resource allocation, enter the following command:

```
hostname# show resource allocation [detail]
```

This command shows the resource allocation, but does not show the actual resources being used. See the "Viewing Resource Usage" section on page 4-39 for more information about actual resource usage.

The **detail** argument shows additional information. See the following sample displays for more information.

The following is sample output from the **show resource allocation** comamnd, and shows the total allocation of each resource as an absolute value and as a percentage of the available system resources:

```
hostname# show resource allocation
Resource                    Total       % of Avail
 Conns [rate]                35000           35.00%
 Fixups [rate]               35000           35.00%
 Syslogs [rate]              10500           35.00%
 Conns                      305000           30.50%
 Hosts                       78842           30.07%
 IPsec                           7           35.00%
 SSH                            35           35.00%
 Telnet                         35           35.00%
```

```
 Xlates                      91749           34.99%
 All                     unlimited
```

Table 4-4 shows each field description.

Table 4-4 show resource allocation Fields

Field	Description
Resource	The name of the resource that you can limit.
Total	The total amount of the resource that is allocated across all contexts. The amount is an absolute number of concurrent instances or instances per second. If you specified a percentage in the class definition, the FWSM converts the percentage to an absolute number for this display.
% of Avail	The percentage of the total system resources that is allocated across all contexts.

The following is sample putput from the **show resource allocation detail** command:

```
hostname# show resource allocation detail
Resource Origin:
    A    Value was derived from the resource 'all'
    C    Value set in the definition of this class
    D    Value set in default class
Resource         Class          Mmbrs  Origin      Limit      Total    Total %
Conns [rate]     default          all      CA  unlimited
                 gold               1       C      34000      34000     20.00%
                 silver             1      CA      17000      17000     10.00%
                 bronze             0      CA       8500
                 All Contexts:      3                         51000     30.00%

Fixups [rate]    default          all      CA  unlimited
                 gold               1      DA  unlimited
                 silver             1      CA      10000      10000     10.00%
                 bronze             0      CA       5000
                 All Contexts:      3                         10000     10.00%

Syslogs [rate]   default          all      CA  unlimited
                 gold               1       C       6000       6000     20.00%
                 silver             1      CA       3000       3000     10.00%
                 bronze             0      CA       1500
                 All Contexts:      3                          9000     30.00%

Conns            default          all      CA  unlimited
                 gold               1       C     200000     200000     20.00%
                 silver             1      CA     100000     100000     10.00%
                 bronze             0      CA      50000
                 All Contexts:      3                        300000     30.00%

Hosts            default          all      CA  unlimited
                 gold               1      DA  unlimited
                 silver             1      CA      26214      26214      9.99%
                 bronze             0      CA      13107
                 All Contexts:      3                         26214      9.99%

IPSec            default          all       C          5
                 gold               1       D          5          5     50.00%
                 silver             1      CA          1          1     10.00%
                 bronze             0      CA  unlimited
                 All Contexts:      3                            11    110.00%

SSH              default          all       C          5
```

```
                gold              1       D          5          5      5.00%
                silver            1      CA         10         10     10.00%
                bronze            0      CA          5
                All Contexts:     3                             20     20.00%

Telnet          default         all      C          5
                gold              1      D          5          5      5.00%
                silver            1      CA         10         10     10.00%
                bronze            0      CA          5
                All Contexts:     3                             20     20.00%

Xlates          default         all      CA  unlimited
                gold              1      DA  unlimited
                silver            1      CA       23040      23040     10.00%
                bronze            0      CA       11520
                All Contexts:     3                          23040     10.00%

mac-addresses   default         all      C        65535
                gold              1      D        65535      65535    100.00%
                silver            1      CA        6553       6553      9.99%
                bronze            0      CA        3276
                All Contexts:     3                         137623    209.99%
```

Table 4-5 shows each field description.

Table 4-5 show resource allocation detail Fields

Field	Description
Resource	The name of the resource that you can limit.
Class	The name of each class, including the default class. The All contexts field shows the total values across all classes.
Mmbrs	The number of contexts assigned to each class.
Origin	The origin of the resource limit, as follows: • A—You set this limit with the **all** option, instead of as an individual resource. • C—This limit is derived from the member class. • D—This limit was not defined in the member class, but was derived from the default class. For a context assigned to the default class, the value will be “C” instead of “D.” The FWSM can combine “A” with “C” or “D.”
Limit	The limit of the resource per context, as an absolute number. If you specified a percentage in the class definition, the FWSM converts the percentage to an absolute number for this display.
Total	The total amount of the resource that is allocated across all contexts in the class. The amount is an absolute number of concurrent instances or instances per second. If the resource is unlimited, this display is blank.
% of Avail	The percentage of the total system resources that is allocated across all contexts in the class. If the resource is unlimited, this display is blank.

Viewing Resource Usage

From the system execution space, you can view the resource usage for each context and display the system resource usage.

From the system execution space, view the resource usage for each context by entering the following command:

```
hostname# show resource usage [context context_name | top n | all | summary | system]
[resource {resource_name | all} | detail] [counter counter_name [count_threshold]]
```

By default, **all** context usage is displayed; each context is listed separately.

Enter the **top** *n* keyword to show the contexts that are the top *n* users of the specified resource. You must specify a single resource type, and not **resource all**, with this option.

The **summary** option shows all context usage combined.

The **system** option shows all context usage combined, but shows the system limits for resources instead of the combined context limits.

For the **resource** *resource_name*, see Table 4-2 for available resource names. See also the **show resource type** command. Specify **all** (the default) for all types.

The **detail** option shows the resource usage of all resources, including those you cannot manage. For example, you can view the number of TCP intercepts.

The **counter** *counter_name* is one of the following keywords:

- **current**—Shows the active concurrent instances or the current rate of the resource.
- **denied**—Shows the number of instances that were denied because they exceeded the resource allocation.
- **peak**—Shows the peak concurrent instances, or the peak rate of the resource since the statistics were last cleared, either using the **clear resource usage** command or because the device rebooted.
- **all**—(Default) Shows all statistics.

The *count_threshold* sets the number above which resources are shown. The default is 1. If the usage of the resource is below the number you set, then the resource is not shown. If you specify **all** for the counter name, then the *count_threshold* applies to the current usage.

Note To show all resources, set the *count_threshold* to **0**.

The following is sample output from the **show resource usage context** command, which shows the resource usage for the admin context:

```
hostname# show resource usage context admin

Resource              Current         Peak      Limit    Denied  Context
Telnet                      1            1          5         0  admin
Conns                      44           55        N/A         0  admin
Hosts                      45           56        N/A         0  admin
```

The following is sample output from the **show resource usage summary** command, which shows the resource usage for all contexts and all resources. This sample shows the limits for 6 contexts.

```
hostname# show resource usage summary

Resource              Current         Peak      Limit     Denied Context
Syslogs [rate]           1743         2132      12000(U)       0 Summary
Conns                     584          763     100000(S)       0 Summary
```

```
Xlates                   8526         8966      93400          0 Summary
Hosts                     254          254     262144          0 Summary
Conns [rate]              270          535      42200       1704 Summary
Fixups [rate]             270          535     100000(S)       0 Summary
U = Some contexts are unlimited and are not included in the total.
S = System limit: Combined context limits exceed the system limit; the system limit is
shown.
```

The following is sample output from the **show resource usage system counter all 0** command, which shows the resource usage for all contexts, but it shows the system limit instead of the combined context limits:

```
hostname# show resource usage system counter all 0

Resource              Current         Peak      Limit        Denied Context
Telnet                      0            0        100             0 System
SSH                         0            0        100             0 System
ASDM                        0            0         80             0 System
IPSec                       0            0         10             0 System
Syslogs [rate]              0            0      30000             0 System
Conns                       0            0    1000000             0 System
Xlates                      0            0     262144             0 System
Hosts                       0            0     262144             0 System
Conns [rate]                0            0     170000             0 System
Fixups [rate]               0            0     100000             0 System
Mac-addresses               0            0      65535             0 System
```

Monitoring SYN Attacks in Contexts

The FWSM prevents SYN attacks using TCP Intercept. TCP Intercept uses the SYN cookies algorithm to prevent TCP SYN-flooding attacks. A SYN-flooding attack consists of a series of SYN packets usually originating from spoofed IP addresses. The constant flood of SYN packets keeps the server SYN queue full, which prevents it from servicing connection requests. When the embryonic connection threshold of a connection is crossed, the FWSM acts as a proxy for the server and generates a SYN-ACK response to the client SYN request. When the FWSM receives an ACK back from the client, it can then authenticate the client and allow the connection to the server.

You can monitor the rate of attacks for individual contexts using the **show perfmon** command; you can monitor the amount of resources being used by TCP intercept for individual contexts using the **show resource usage detail** command; you can monitor the resources being used by TCP intercept for the entire system using the **show resource usage summary detail** command.

The following is sample output from the **show perfmon** command that shows the rate of TCP intercepts for a context called admin:

```
hostname/admin# show perfmon

Context:admin
PERFMON STATS:   Current      Average
Xlates               0/s          0/s
Connections          0/s          0/s
TCP Conns            0/s          0/s
UDP Conns            0/s          0/s
URL Access           0/s          0/s
URL Server Req       0/s          0/s
WebSns Req           0/s          0/s
TCP Fixup            0/s          0/s
HTTP Fixup           0/s          0/s
FTP Fixup            0/s          0/s
AAA Authen           0/s          0/s
```

```
AAA Author            0/s           0/s
AAA Account           0/s           0/s
TCP Intercept    322779/s      322779/s
```

The following is sample output from the **show resource usage detail** command that shows the amount of resources being used by TCP Intercept for individual contexts. (Sample text in italics shows the TCP intercept information.)

```
hostname(config)# show resource usage detail
Resource              Current         Peak      Limit        Denied Context
memory                 843732       847288  unlimited             0 admin
chunk:channels             14           15  unlimited             0 admin
chunk:fixup                15           15  unlimited             0 admin
chunk:hole                  1            1  unlimited             0 admin
chunk:ip-users             10           10  unlimited             0 admin
chunk:list-elem            21           21  unlimited             0 admin
chunk:list-hdr              3            4  unlimited             0 admin
chunk:route                 2            2  unlimited             0 admin
chunk:static                1            1  unlimited             0 admin
tcp-intercept-rate     328787       803610  unlimited             0 admin
np-statics                  3            3  unlimited             0 admin
statics                     1            1  unlimited             0 admin
ace-rules                   1            1        N/A             0 admin
console-access-rul          2            2        N/A             0 admin
fixup-rules                14           15        N/A             0 admin
memory                 959872       960000  unlimited             0 c1
chunk:channels             15           16  unlimited             0 c1
chunk:dbgtrace              1            1  unlimited             0 c1
chunk:fixup                15           15  unlimited             0 c1
chunk:global                1            1  unlimited             0 c1
chunk:hole                  2            2  unlimited             0 c1
chunk:ip-users             10           10  unlimited             0 c1
chunk:udp-ctrl-blk          1            1  unlimited             0 c1
chunk:list-elem            24           24  unlimited             0 c1
chunk:list-hdr              5            6  unlimited             0 c1
chunk:nat                   1            1  unlimited             0 c1
chunk:route                 2            2  unlimited             0 c1
chunk:static                1            1  unlimited             0 c1
tcp-intercept-rate      16056        16254  unlimited             0 c1
globals                     1            1  unlimited             0 c1
np-statics                  3            3  unlimited             0 c1
statics                     1            1  unlimited             0 c1
nats                        1            1  unlimited             0 c1
ace-rules                   2            2        N/A             0 c1
console-access-rul          2            2        N/A             0 c1
fixup-rules                14           15        N/A             0 c1
memory              232695716    232020648  unlimited             0 system
chunk:channels             17           20  unlimited             0 system
chunk:dbgtrace              3            3  unlimited             0 system
chunk:fixup                15           15  unlimited             0 system
chunk:ip-users              4            4  unlimited             0 system
chunk:list-elem          1014         1014  unlimited             0 system
chunk:list-hdr              1            1  unlimited             0 system
chunk:route                 1            1  unlimited             0 system
block:16384               510          885  unlimited             0 system
block:2048                 32           34  unlimited             0 system
```

The following sample output from the **show resource usage summary detail** command shows the resources being used by TCP intercept for the entire system. (Sample text in italics shows the TCP intercept information.)

```
hostname(config)# show resource usage summary detail
Resource              Current         Peak      Limit        Denied Context
```

```
memory              238421312    238434336  unlimited             0 Summary
chunk:channels              46           48  unlimited             0 Summary
chunk:dbgtrace               4            4  unlimited             0 Summary
chunk:fixup                 45           45  unlimited             0 Summary
chunk:global                 1            1  unlimited             0 Summary
chunk:hole                   3            3  unlimited             0 Summary
chunk:ip-users              24           24  unlimited             0 Summary
chunk:udp-ctrl-blk           1            1  unlimited             0 Summary
chunk:list-elem           1059         1059  unlimited             0 Summary
chunk:list-hdr              10           11  unlimited             0 Summary
chunk:nat                    1            1  unlimited             0 Summary
chunk:route                  5            5  unlimited             0 Summary
chunk:static                 2            2  unlimited             0 Summary
block:16384                510          885       8192(S)          0 Summary
block:2048                  32           35       1000(S)          0 Summary
tcp-intercept-rate      341306       811579  unlimited             0 Summary
globals                      1            1       1051(S)          0 Summary
np-statics                   6            6       4096(S)          0 Summary
statics                      2            2       2048(S)          0 Summary
nats                         1            1       2048(S)          0 Summary
ace-rules                    3            3     116448(S)          0 Summary
console-access-rul           4            4       4356(S)          0 Summary
fixup-rules                 43           44       8032(S)          0 Summary
S = System: Total exceeds the system limit; the system limit is shown
```

CHAPTER 5

Configuring the Firewall Mode

This chapter describes how to set the firewall mode, as well as how the firewall works in each firewall mode. You can set the firewall mode independently for each context in multiple context mode.

The FWSM (or each context in multiple mode) can run in one of two firewall modes:

- Routed mode
- Transparent mode

This chapter includes the following sections:

Routed Mode Overview

In routed mode, the FWSM is considered to be a router hop in the network. It can use OSPF or passive RIP (in single context mode). Routed mode supports many interfaces, and each interface is on a different subnet. You can share interfaces between contexts, with some limitations.

IP Routing Support

The FWSM acts as a router between connected networks, and each interface requires an IP address on a different subnet. In single context mode, the routed firewall supports OSPF and RIP (in passive mode). Multiple context mode supports static routes only. We recommend using the advanced routing capabilities of the upstream and downstream routers instead of relying on the FWSM for extensive routing needs.

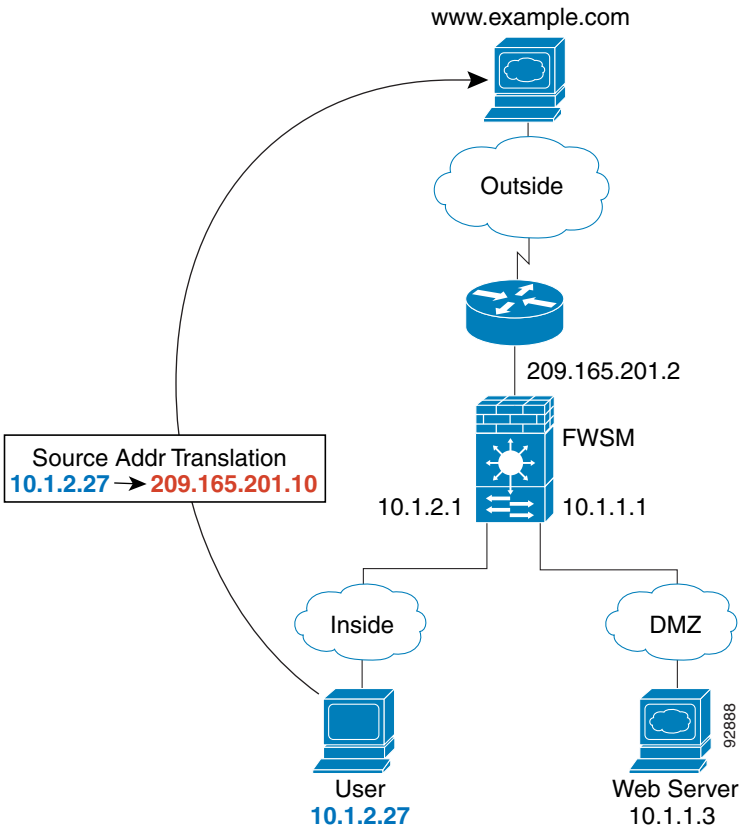
How Data Moves Through the FWSM in Routed Firewall Mode

This section describes how data moves through the FWSM in routed firewall mode, and includes the following topics:

An Inside User Visits a Web Server

Figure 5-1 shows an inside user accessing an outside web server.

Figure 5-1 Inside to Outside



The following steps describe how data moves through the FWSM (see Figure 5-1):

1. The user on the inside network requests a web page from www.example.com.
2. The FWSM receives the packet and because it is a new session, the FWSM verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

For multiple context mode, the FWSM first classifies the packet according to either a unique interface or a unique destination address associated with a context; the destination address is associated by matching an address translation in a context. In this case, the interface would be unique; the www.example.com IP address does not have a current address translation in a context.

3. The FWSM translates the real address (10.1.2.27) to the mapped address 209.165.201.10, which is on the outside interface subnet.

   The mapped address could be on any subnet, but routing is simplified when it is on the outside interface subnet.

4. The FWSM then records that a session is established and forwards the packet from the outside interface.

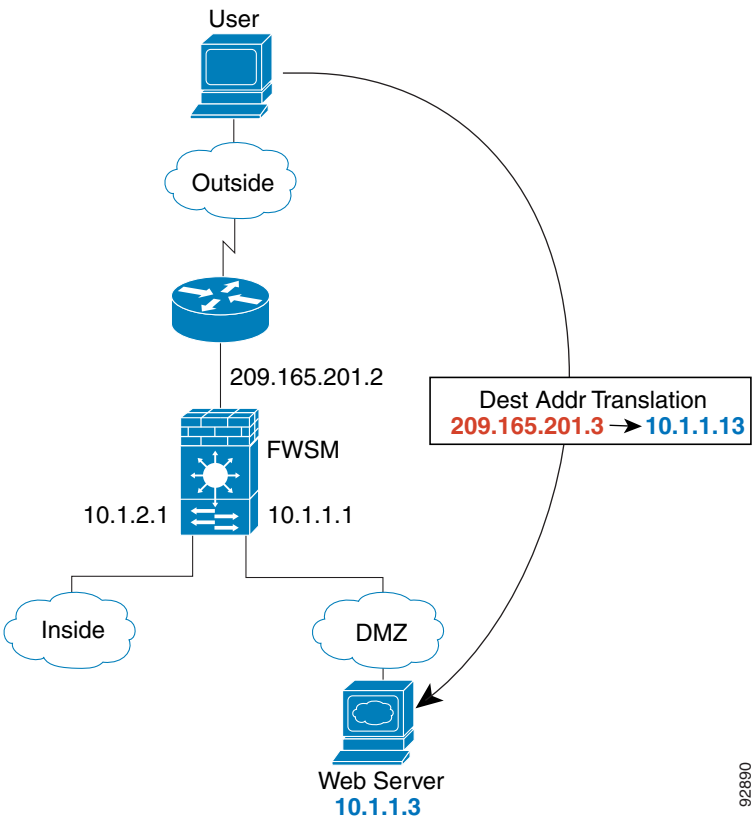
5. When www.example.com responds to the request, the packet goes through the FWSM, and because the session is already established, the packet bypasses the many lookups associated with a new connection. The FWSM performs NAT by translating the mapped address to the real address, 10.1.2.27.

6. The FWSM forwards the packet to the inside user.

An Outside User Visits a Web Server on the DMZ

Figure 5-2 shows an outside user accessing the DMZ web server.

Figure 5-2 Outside to DMZ



The following steps describe how data moves through the FWSM (see Figure 5-2):

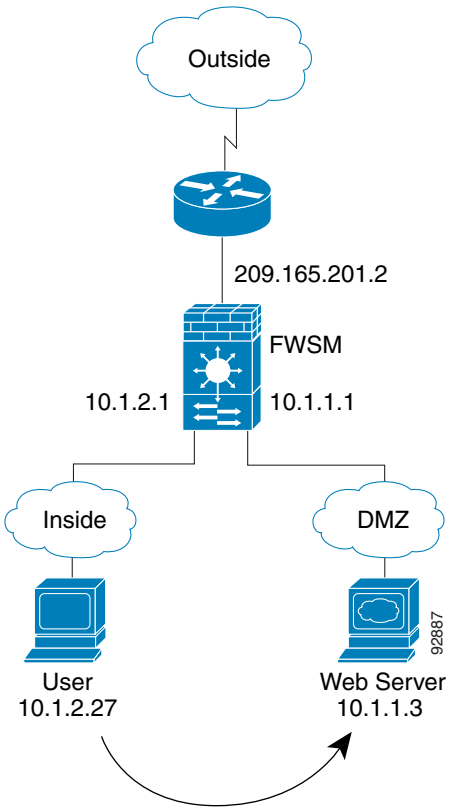
1. A user on the outside network requests a web page from the DMZ web server using the mapped address of 209.165.201.3, which is on the outside interface subnet.
2. The FWSM receives the packet and because it is a new session, the FWSM verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

   For multiple context mode, the FWSM first classifies the packet according to either a unique interface or a unique destination address associated with a context; the destination address is associated by matching an address translation in a context. In this case, the classifier "knows" that the DMZ web server address belongs to a certain context because of the server address translation.
3. The FWSM translates the destination address to the real address 10.1.1.3.
4. The FWSM then adds a session entry to the fast path and forwards the packet from the DMZ interface.
5. When the DMZ web server responds to the request, the packet goes through the FWSM and because the session is already established, the packet bypasses the many lookups associated with a new connection. The FWSM performs NAT by translating the real address to 209.165.201.3.
6. The FWSM forwards the packet to the outside user.

An Inside User Visits a Web Server on the DMZ

Figure 5-3 shows an inside user accessing the DMZ web server.

Figure 5-3 Inside to DMZ



The following steps describe how data moves through the FWSM (see Figure 5-3):

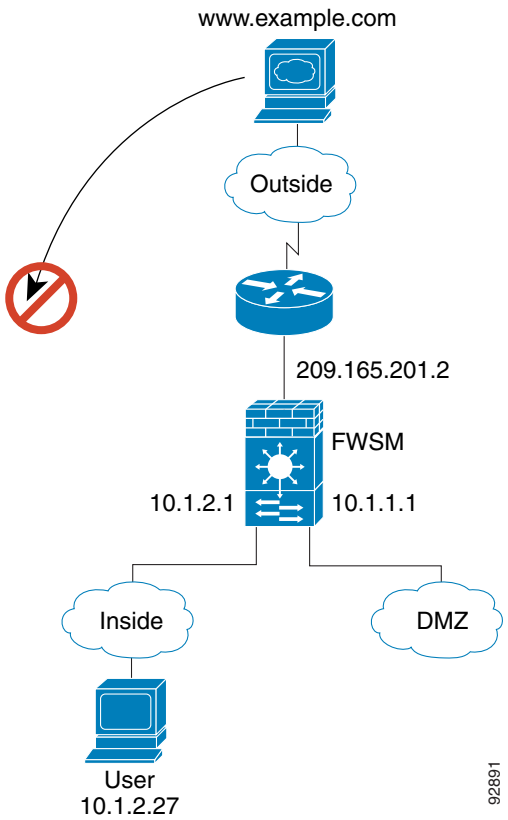
1. A user on the inside network requests a web page from the DMZ web server using the destination address of 10.1.1.3.
2. The FWSM receives the packet and because it is a new session, the FWSM verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

   For multiple context mode, the FWSM first classifies the packet according to either a unique interface or a unique destination address associated with a context; the destination address is associated by matching an address translation in a context. In this case, the interface is unique; the web server IP address does not have a current address translation.
3. The FWSM then records that a session is established and forwards the packet out of the DMZ interface.
4. When the DMZ web server responds to the request, the packet goes through the fast path, which lets the packet bypass the many lookups associated with a new connection.
5. The FWSM forwards the packet to the inside user.

An Outside User Attempts to Access an Inside Host

Figure 5-4 shows an outside user attempting to access the inside network.

Figure 5-4 Outside to Inside



The following steps describe how data moves through the FWSM (see Figure 5-4):

1. A user on the outside network attempts to reach an inside host (assuming the host has a routable IP address).

   If the inside network uses private addresses, no outside user can reach the inside network without NAT. The outside user might attempt to reach an inside user by using an existing NAT session.

2. The FWSM receives the packet and because it is a new session, the FWSM verifies if the packet is allowed according to the security policy (access lists, filters, AAA).

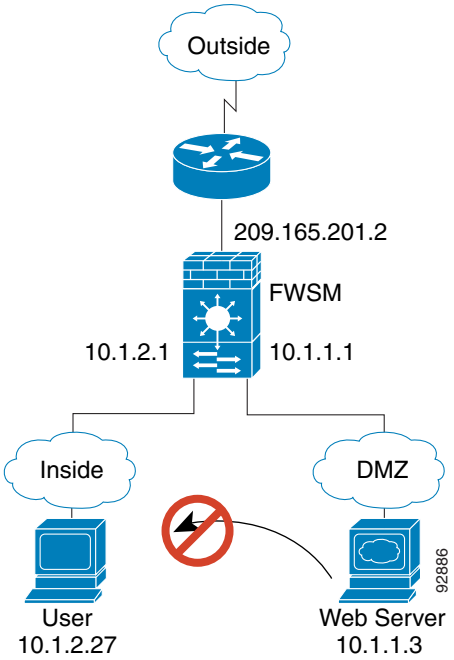
3. The packet is denied, and the FWSM drops the packet and logs the connection attempt.

   If the outside user is attempting to attack the inside network, the FWSM employs many technologies to determine if a packet is valid for an already established session.

A DMZ User Attempts to Access an Inside Host

Figure 5-5 shows a user in the DMZ attempting to access the inside network.

Figure 5-5 DMZ to Inside



The following steps describe how data moves through the FWSM (see Figure 5-5):

1. A user on the DMZ network attempts to reach an inside host. Because the DMZ does not have to route the traffic on the Internet, the private addressing scheme does not prevent routing.

2. The FWSM receives the packet and because it is a new session, the FWSM verifies if the packet is allowed according to the security policy (access lists, filters, AAA).

3. The packet is denied, and the FWSM drops the packet and logs the connection attempt.

Transparent Mode Overview

A transparent firewall is a Layer 2 firewall that acts like a "bump in the wire," or a "stealth firewall," and is not seen as a router hop to connected devices.

This section describes transparent firewall mode, and includes the following topics:

Transparent Firewall Network

The FWSM connects the same network on its inside and outside interfaces. Because the firewall is not a routed hop, you can easily introduce a transparent firewall into an existing network; IP readdressing is unnecessary.

You can optionally enable NAT for hosts connected to the transparent firewall.

Bridge Groups

If you do not want the overhead of security contexts, or want to maximize your use of security contexts, you can configure up to eight pairs of interfaces, called bridge groups. Each bridge group connects to a separate network. Bridge group traffic is isolated from other bridge groups; traffic is not routed to another bridge group within the FWSM, and traffic must exit the FWSM before it is routed by an external router back to another bridge group in the FWSM. Although the bridging functions are separate for each bridge group, many other functions are shared between all bridge groups. For example, all bridge groups share a system log server or AAA server configuration. For complete security policy separation, use security contexts with one bridge group in each context.

Because the firewall is not a routed hop, you can easily introduce a transparent firewall into an existing network; IP readdressing is unnecessary. Maintenance is facilitated because there are no complicated routing patterns to troubleshoot.

Note Each bridge group requires a management IP address. The FWSM uses this IP address as the source address for packets originating from the bridge group. The management IP address must be on the same subnet as the connected network.

Allowing Layer 3 Traffic

Even though transparent mode acts as a bridge, Layer 3 traffic, such as IP traffic, cannot pass through the FWSM unless you explicitly permit it with an extended access list. The only traffic allowed through the transparent firewall without an access list is ARP traffic. ARP traffic can be controlled by ARP inspection. See the "Adding an Extended Access List" section on page 12-6 for more information.

Allowed MAC Addresses

The following destination MAC addresses are allowed through the transparent firewall. Any MAC address not on this list is dropped.

- TRUE broadcast destination MAC address equal to FFFF.FFFF.FFFF
- IPv4 multicast MAC addresses from 0100.5E00.0000 to 0100.5EFE.FFFF
- IPv6 multicast MAC addresses from 3333.0000.0000 to 3333.FFFF.FFFF
- BPDU multicast address equal to 0100.0CCC.CCCD
- Appletalk multicast MAC addresses from 0900.0700.0000 to 0900.07FF.FFFF

Passing Traffic Not Allowed in Routed Mode

In routed mode, some types of traffic cannot pass through the FWSM even if you allow it in an access list. The transparent firewall, however, can pass most types of traffic through using either an extended access list (for IP traffic) or an EtherType access list (for non-IP traffic).

Note The transparent mode FWSM does not pass CDP packets, or any packets that do not have a valid EtherType greater than or equal to 0x600. For example, you cannot pass IS-IS packets. An exception is made for BPDUs, which are supported.

For example, you can establish routing protocol adjacencies through a transparent firewall; you can allow OSPF, RIP, EIGRP, or BGP traffic through based on an extended access list. Likewise, protocols like HSRP or VRRP can pass through the FWSM. See Table 12-2 on page 12-7 for more information about allowing special traffic.

Non-IP traffic (for example AppleTalk, IPX, BPDUs, and MPLS) can be configured to go through using an EtherType access list.

For features that are not directly supported on the transparent firewall, you can allow traffic to pass through so that upstream and downstream routers can support the functionality. For example, by using an extended access list, you can allow DHCP traffic (instead of the unsupported DHCP relay feature) or multicast traffic such as that created by IP/TV.

MAC Address vs. Route Lookups

When the FWSM runs in transparent mode without NAT, the outgoing interface of a packet is determined by performing a MAC address lookup instead of a route lookup. Route statements can still be configured, but they only apply to FWSM-originated traffic. For example, if your syslog server is located on a remote network, you must use a static route so the FWSM can reach that subnet.

An exception to this rule is when you use voice inspections and the endpoint is at least one hop away from the FWSM. For example, if you use the transparent firewall between a CCM and an H.323 gateway, and there is a router between the transparent firewall and the H.323 gateway, then you need to add a static route on the FWSM for the H.323 gateway for successful call completion.

If you use NAT, then the FWSM uses a route lookup instead of a MAC address lookup. In some cases, you will need static routes. For example, if the real destination address is not directly-connected to the FWSM, then you need to add a static route on the FWSM for the real destination address that points to the downstream router.

Using the Transparent Firewall in Your Network

Figure 5-6 shows a typical transparent firewall network where the outside devices are on the same subnet as the inside devices. The inside router and hosts appear to be directly connected to the outside router.

Figure 5-6 Transparent Firewall Network

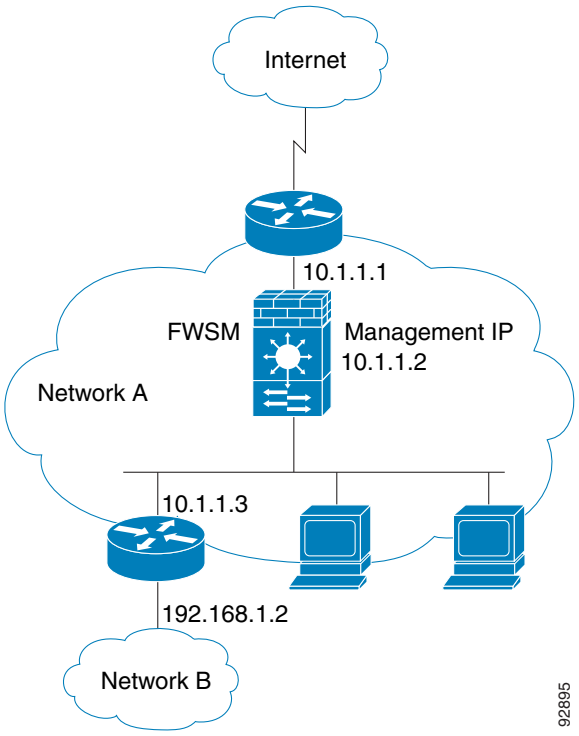
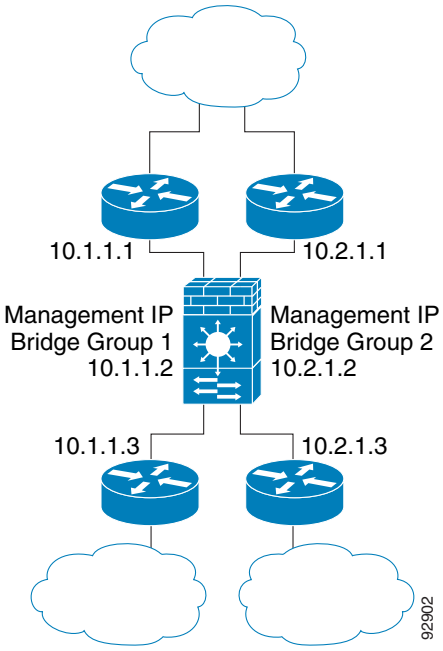


Figure 5-7 shows two networks connected to the FWSM, which has two bridge groups.

Figure 5-7 Transparent Firewall Network with Two Bridge Groups



Transparent Firewall Guidelines

Follow these guidelines when planning your transparent firewall network:

- A management IP address is required for each bridge group.

  Unlike routed mode, which requires an IP address for each interface, a transparent firewall has an IP address assigned to the entire bridge group. The FWSM uses this IP address as the source address for packets originating on the FWSM, such as system messages or AAA communications.

  The management IP address must be on the same subnet as the connected network. The FWSM does not support traffic on secondary networks; only traffic on the same network as the management IP address is supported. See the "Assigning an IP Address to a Bridge Group" section on page 6-5 for more information about management IP subnets.

- Each bridge group uses an inside interface and an outside interface only.
- Each directly-connected network must be on the same subnet.
- Do not specify the bridge group management IP address as the default gateway for connected devices; devices need to specify the router on the other side of the FWSM as the default gateway.
- The default route for the transparent firewall, which is required to provide a return path for management traffic, is only applied to management traffic from one bridge group network. This is because the default route specifies an interface in the bridge group as well as the router IP address on the bridge group network, and you can only define one default route. If you have management traffic from more than one bridge group network, you need to specify a static route that identifies the network from which you expect management traffic.
- For multiple context mode, each context must use different interfaces; you cannot share an interface across contexts.

- For multiple context mode, each context typically uses different subnets. You can use overlapping subnets, but your network topology requires router and NAT configuration to make it possible from a routing standpoint.
- You must use an extended access list to allow Layer 3 traffic, such as IP traffic, through the FWSM.

  You can also optionally use an EtherType access list to allow non-IP traffic through.

Unsupported Features in Transparent Mode

Table 5-1 lists features that are not supported in transparent mode.

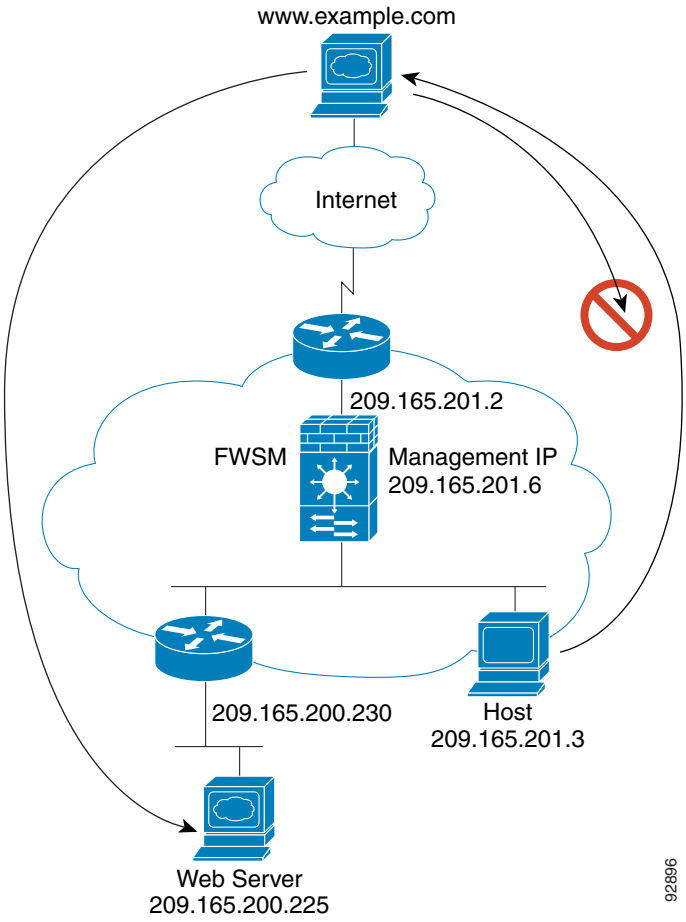
Table 5-1 Unsupported Features in Transparent Mode

Unsupported Feature	Description
DHCP relay	The transparent firewall can act as a DHCP server, but it does not support the DHCP relay commands. DHCP relay is not required because you can allow DHCP traffic to pass through using an extended access list.
Dynamic routing protocols	You can, however, add static routes for traffic originating on the FWSM. You can also allow dynamic routing protocols through the FWSM using an extended access list.
IPv6 for the bridge group IP address	You can, however, pass the IPv6 EtherType using an EtherType access list.
LoopGuard on the switch	Do not enable LoopGuard globally on the switch if the FWSM is in transparent mode. LoopGuard is automatically applied to the internal EtherChannel between the switch and the FWSM, so after a failover and a failback, LoopGuard causes the secondary unit to be disconnected because the EtherChannel goes into the err-disable state.
Multicast	You can, however, allow multicast traffic through the FWSM by allowing it in an extended access list.
Remote access VPN for management	You can use site-to-site VPN for management.

How Data Moves Through the Transparent Firewall

Figure 5-8 shows a typical transparent firewall implementation with an inside network that contains a public web server. The FWSM has an access list so that the inside users can access Internet resources. Another access list lets the outside users access only the web server on the inside network.

Figure 5-8 Typical Transparent Firewall Data Path



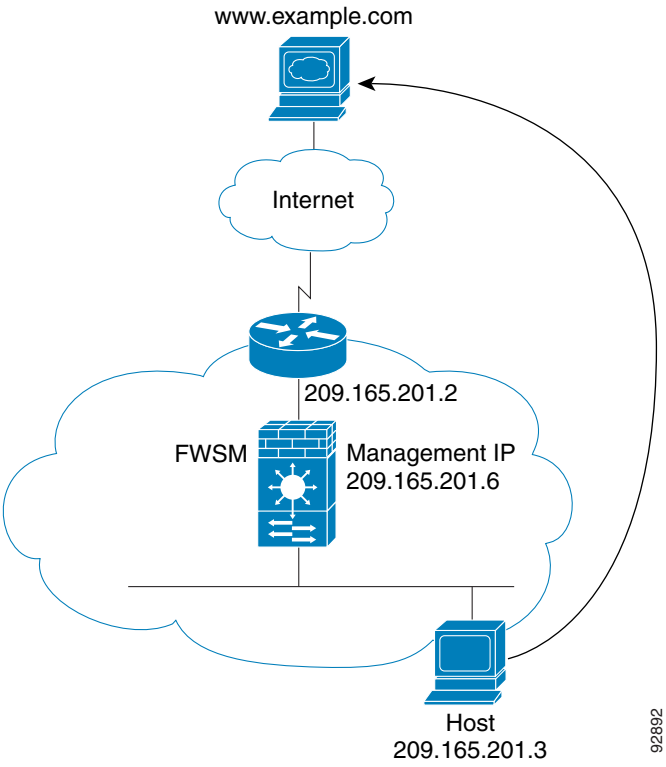
This section describes how data moves through the FWSM, and includes the following topics:

- An Inside User Visits a Web Server, page 5-13
- An Inside User Visits a Web Server Using NAT, page 5-14
- An Outside User Visits a Web Server on the Inside Network, page 5-15
- An Outside User Attempts to Access an Inside Host, page 5-16

An Inside User Visits a Web Server

Figure 5-9 shows an inside user accessing an outside web server.

Figure 5-9 Inside to Outside



The following steps describe how data moves through the FWSM (see Figure 5-9):

1. The user on the inside network requests a web page from www.example.com.
2. The FWSM receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

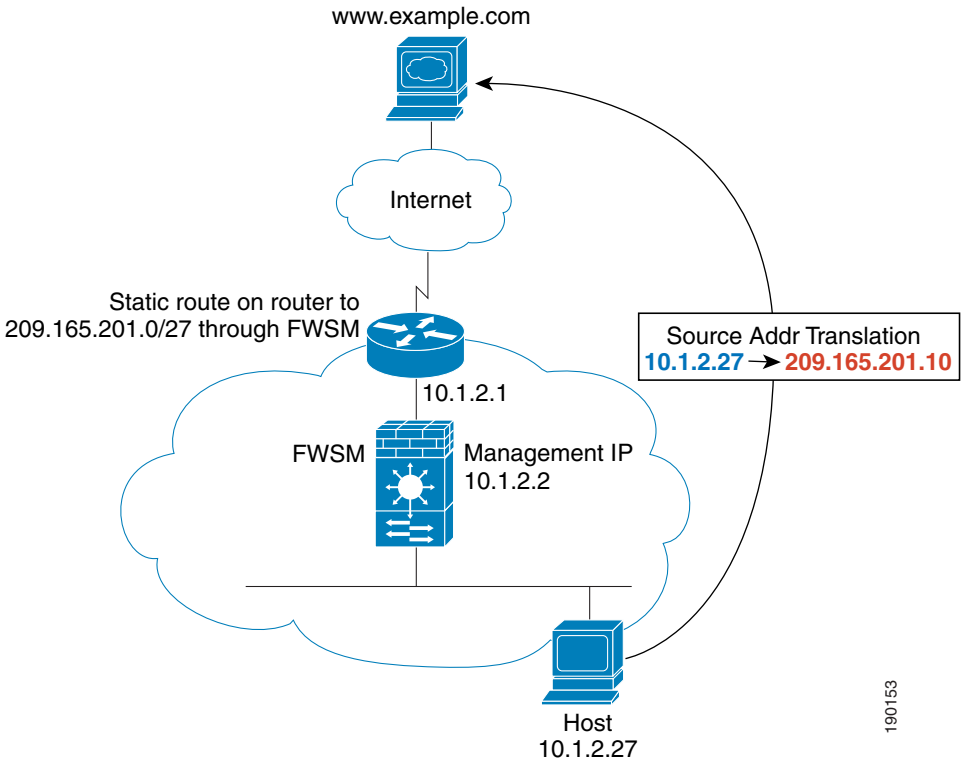
   For multiple context mode, the FWSM first classifies the packet according to a unique interface.
3. The FWSM records that a session is established.
4. If the destination MAC address is in its table, the FWSM forwards the packet out of the outside interface. The destination MAC address is that of the upstream router, 209.165.201.2.

   If the destination MAC address is not in the FWSM table, the FWSM attempts to discover the MAC address by sending an ARP request and a ping. The first packet is dropped.
5. The web server responds to the request; because the session is already established, the packet bypasses the many lookups associated with a new connection.
6. The FWSM forwards the packet to the inside user.

An Inside User Visits a Web Server Using NAT

Figure 5-9 shows an inside user accessing an outside web server.

Figure 5-10 Inside to Outside with NAT



The following steps describe how data moves through the FWSM (see Figure 5-9):

1. The user on the inside network requests a web page from www.example.com.
2. The FWSM receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

   For multiple context mode, the FWSM first classifies the packet according to a unique interface.
3. The FWSM translates the real address (10.1.2.27) to the mapped address 209.165.201.10.

   Because the mapped address is not on the same network as the outside interface, then be sure the upstream router has a static route to the mapped network that points to the FWSM.
4. The FWSM then records that a session is established and forwards the packet from the outside interface.
5. If the destination MAC address is in its table, the FWSM forwards the packet out of the outside interface. The destination MAC address is that of the upstream router, 209.165.201.2.

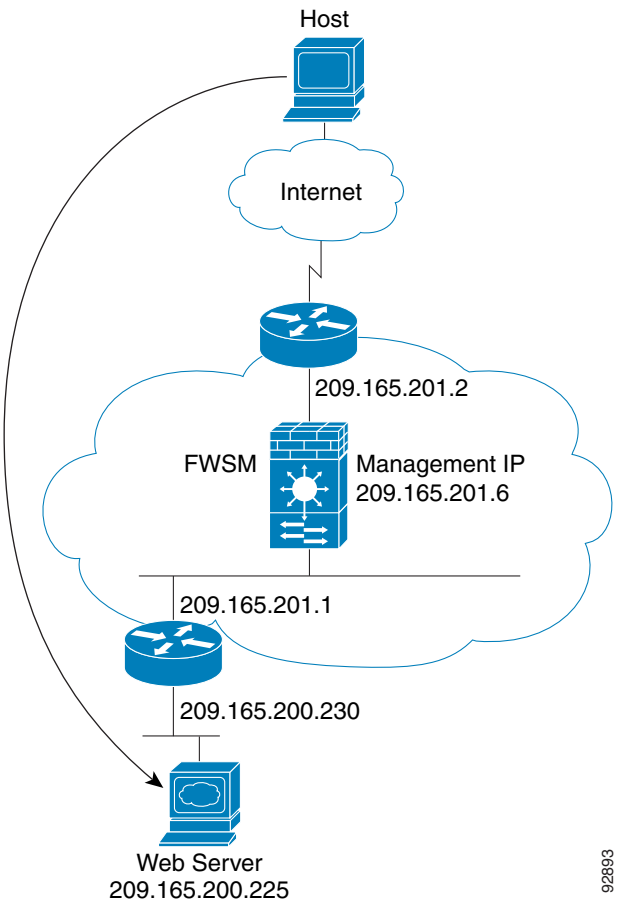
   If the destination MAC address is not in the FWSM table, the FWSM attempts to discover the MAC address by sending an ARP request and a ping. The first packet is dropped.
6. The web server responds to the request; because the session is already established, the packet bypasses the many lookups associated with a new connection.
7. The FWSM performs NAT by translating the mapped address to the real address, 10.1.2.27.

8. The FWSM forwards the packet to the inside user.

An Outside User Visits a Web Server on the Inside Network

Figure 5-11 shows an outside user accessing the inside web server.

Figure 5-11 Outside to Inside



The following steps describe how data moves through the FWSM (see Figure 5-11):

1. A user on the outside network requests a web page from the inside web server.
2. The FWSM receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies that the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

   For multiple context mode, the FWSM first classifies the packet according to a unique interface.
3. The FWSM records that a session is established.
4. If the destination MAC address is in its table, the FWSM forwards the packet out of the inside interface. The destination MAC address is that of the downstream router, 209.186.201.1.

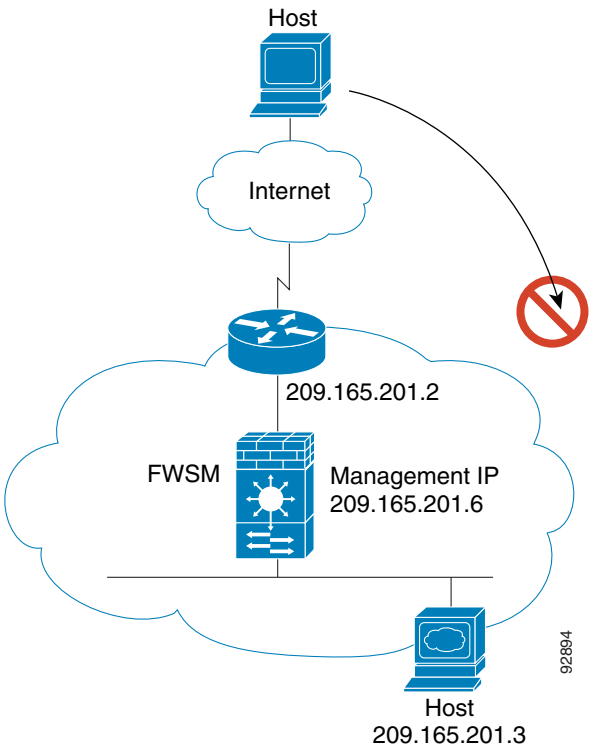
   If the destination MAC address is not in the FWSM table, the FWSM attempts to discover the MAC address by sending an ARP request and a ping. The first packet is dropped.

5. The web server responds to the request; because the session is already established, the packet bypasses the many lookups associated with a new connection.
6. The FWSM forwards the packet to the outside user.

An Outside User Attempts to Access an Inside Host

Figure 5-12 shows an outside user attempting to access a host on the inside network.

Figure 5-12 Outside to Inside



The following steps describe how data moves through the FWSM (see Figure 5-12):

1. A user on the outside network attempts to reach an inside host.
2. The FWSM receives the packet and adds the source MAC address to the MAC address table, if required. Because it is a new session, it verifies if the packet is allowed according to the terms of the security policy (access lists, filters, AAA).

   For multiple context mode, the FWSM first classifies the packet according to a unique interface.
3. The packet is denied, and the FWSM drops the packet.
4. If the outside user is attempting to attack the inside network, the FWSM employs many technologies to determine if a packet is valid for an already established session.

Setting Transparent or Routed Firewall Mode

You can set each context to run in routed firewall mode (the default) or transparent firewall mode.

When you change modes, the FWSM clears the configuration because many commands are not supported for both modes. If you already have a populated configuration, be sure to back up your configuration before changing the mode; you can use this backup for reference when creating your new configuration.

If you download a text configuration to the FWSM that changes the mode with the **firewall transparent** command, be sure to put the command at the top of the configuration; the FWSM changes the mode as soon as it reads the command and then continues reading the configuration you downloaded. If the command is later in the configuration, the FWSM clears all the preceding lines in the configuration.

- To set the mode to transparent, enter the following command in each context:

  ```
  hostname(config)# firewall transparent
  ```

- To set the mode to routed, enter the following command in each context:

  ```
  hostname(config)# no firewall transparent
  ```

CHAPTER 6

Configuring Interface Parameters

This chapter describes how to configure each interface for a name, security level, and IP address. For transparent firewall, you also need to configure a bridge group for each interface pair.

This chapter includes the following sections:

Security Level Overview

Each interface must have a security level from 0 (lowest) to 100 (highest). For example, you should assign your most secure network, such as the inside host network, to level 100. While the outside network connected to the Internet can be level 0. Other networks, such as DMZs can be in between. You can assign interfaces to the same security level. See the "Allowing Communication Between Interfaces on the Same Security Level" section on page 6-6 for more information.

The level controls the following behavior:

- Inspection engines—Some inspection engines are dependent on the security level. For same security interfaces, inspection engines apply to traffic in either direction.
  - NetBIOS inspection engine—Applied only for outbound connections.
  - OraServ inspection engine—If a control connection for the OraServ port exists between a pair of hosts, then only an inbound data connection is permitted through the FWSM.
- Filtering—HTTP(S) and FTP filtering applies only for outbound connections. For same security interfaces, you can filter traffic in either direction.
- NAT control—When you enable NAT control, you must configure NAT for hosts on a higher security interface (inside) when they access hosts on a lower security interface (outside).

  Without NAT control, or for same security interfaces, you can choose to use NAT between any interface, or you can choose not to use NAT. Keep in mind that configuring NAT for an outside interface might require a special keyword.

- **established** command—This command allows return connections from a lower security host to a higher security host if there is already an established connection from the higher level host to the lower level host.

  If you enable communication between same security interfaces (see the "Allowing Communication Between Interfaces on the Same Security Level" section on page 6-6), you can configure **established** commands for both directions.

Configuring Interfaces for Routed Firewall Mode

Before you can allow traffic through the FWSM, you need to configure an interface name and an IP address. You should also change the security level from the default, which is 0. If you name an interface "inside" and you do not set the security level explicitly, then the FWSM sets the security level to 100.

Note If you are using failover, do not use this procedure to name interfaces that you are reserving for failover and Stateful Failover communications. See Chapter 13, "Configuring Failover," to configure the failover and state links.

For multiple context mode, follow these guidelines:

- Configure the context interfaces from within each context.
- You can only configure context interfaces that you already assigned to the context in the system configuration.
- The system configuration only lets you configure failover interfaces; do not configure failover interfaces with this procedure. See Chapter 13, "Configuring Failover," for more information.
- If you change the security level of an interface, and you do not want to wait for existing connections to time out before the new security information is used, you can clear the connections using the **clear local-host** command.

You can add any VLAN ID to the configuration, but only VLANs that are assigned to the FWSM by the switch can pass traffic. To view all VLANs assigned to the FWSM, use the **show vlan** command.

To configure an interface, perform the following steps:

Step 1 To specify the interface you want to configure, enter the following command:

```
hostname(config)# interface {vlan number | mapped_name}
```

In multiple context mode, enter the mapped name if one was assigned using the **allocate-interface** command.

For example, enter the following command:

```
hostname(config)# interface vlan 101
```

Step 2 To name the interface, enter the following command:

```
hostname(config-if)# nameif name
```

The *name* is a text string up to 48 characters, and is not case-sensitive. You can change the name by reentering this command with a new value. Do not enter the **no** form, because that command causes all commands that refer to that name to be deleted.

Note After you set the name for an interface, the security-level is automatically changed to 0. However, if the name is "inside," then the security level becomes 100.

Step 3 To set the security level, enter the following command:

```
hostname(config-if)# security-level number
```

Where *number* is an integer between 0 (lowest) and 100 (highest).

Step 4 To set the IP address, enter the following command:

```
hostname(config-if)# ip address ip_address [mask] [standby ip_address]
```

The **standby** keyword and address is used for failover. See Chapter 13, "Configuring Failover," for more information.

Note To set an IPv6 address, see the "Configuring IPv6 on an Interface" section on page 10-2.

The following example configures parameters for VLAN 101:

```
hostname(config)# interface vlan 101
hostname(config-if)# nameif inside
hostname(config-if)# security-level 100
hostname(config-if)# ip address 10.1.1.1 255.255.255.0
```

The following example configures parameters in multiple context mode for the context configuration. The interface ID is a mapped name.

```
hostname/contextA(config)# interface int1
hostname/contextA(config-if)# nameif outside
hostname/contextA(config-if)# security-level 100
hostname/contextA(config-if)# ip address 10.1.2.1 255.255.255.0
```

Configuring Interfaces for Transparent Firewall Mode

Before you can allow traffic through the FWSM, you need to configure an interface name, security level, and bridge group association. Finally, assign a management IP address for each bridge group. This section includes the following topics:

Configuring Transparent Firewall Interface Parameters

A transparent firewall connects the same network on its inside and outside interfaces. Each pair of interfaces belongs to a bridge group, to which you must assign a management IP address (see the "Assigning an IP Address to a Bridge Group" section on page 6-5). You can configure up to eight bridge groups of two interfaces each. Each bridge group connects to a separate network. Bridge group traffic is

isolated from other bridge groups; traffic is not routed to another bridge group within the FWSM, and traffic must exit the FWSM before it is routed by an external router back to another bridge group in the FWSM.

Note The FWSM does not support traffic on secondary networks; only traffic on the same network as the management IP address is supported.

You might want to use more than one bridge group if you do not want the overhead of security contexts, or want to maximize your use of security contexts. Although the bridging functions are separate for each bridge group, many other functions are shared between all bridge groups. For example, all bridge groups share a system log server or AAA server configuration. For complete security policy separation, use security contexts with one bridge group in each context.

Note If you are using failover, do not use this procedure to name interfaces that you are reserving for failover and Stateful Failover communications.

For multiple context mode, follow these guidelines for configuring interfaces:

- You must configure the context interfaces from within each context.
- You can only configure context interfaces that you already assigned to the context in the system configuration.
- The system configuration only lets you configure failover interfaces; do not configure failover interfaces with this procedure.
- If you change the security level of an interface, and you do not want to wait for existing connections to time out before the new security information is used, you can clear the connections using the **clear local-host** command.

You can add any VLAN ID to the configuration, but only VLANs that are assigned to the FWSM by the switch can pass traffic. To view all VLANs assigned to the FWSM, use the **show vlan** command.

To assign an interface to a bridge group, set the name, and set the security level, perform the following steps:

Step 1 To identify the interface, enter the following command:

```
hostname(config)# interface {vlan number | mapped_name}
```

In multiple context mode, enter the mapped name if one was assigned using the **allocate-interface** command.

Step 2 To assign it to a bridge group, enter the following command:

```
hostname(config-if)# bridge-group number
```

Where *number* is an integer between 1 and 100. You can only assign two interfaces to a bridge group. You cannot assign the same interface to more than one bridge group.

Step 3 To name the interface, enter the following command:

```
hostname(config-if)# nameif name
```

The *name* is a text string up to 48 characters, and is not case-sensitive. You can change the name by reentering this command with a new value. Do not enter the **no** form, because that command causes all commands that refer to that name to be deleted. If you name an interface “inside” and you do not set the security level explicitly, then the FWSM sets the security level to 100.

Step 4 To set the security level, enter the following command:

```
hostname(config-if)# security-level number
```

Where *number* is an integer between 0 (lowest) and 100 (highest). By default, after you name the interface, the FWSM sets the security level to 0.

Assigning an IP Address to a Bridge Group

A transparent firewall does not participate in IP routing. The only IP configuration required for the FWSM is to set the management IP address for each bridge group. This address is required because the FWSM uses this address as the source address for traffic originating on the FWSM, such as system log messages or communications with AAA servers. You can also use this address for remote management access.

To set the management IP address, perform the following steps:

Step 1 Identify the bridge group by entering the following command:

```
hostname(config)# interface bvi bridge_group_number
```

Step 2 Specify the IP address by entering the following command:

```
hostname(config-if)# ip address ip_address [mask] [standby ip_address]
```

Do not assign a host address (/32 or 255.255.255.255) to the transparent firewall. Also, do not use other subnets that contain fewer than 3 host addresses (one each for the upstream router, downstream router, and transparent firewall) such as a /30 subnet (255.255.255.252). The FWSM drops all ARP packets to or from the first and last addresses in a subnet. Therefore, if you use a /30 subnet and assign a reserved address from that subnet to the upstream router, then the FWSM drops the ARP request from the downstream router to the upstream router.

The FWSM does not support traffic on secondary networks; only traffic on the same network as the management IP address is supported.

The **standby** keyword and address is used for failover. See Chapter 13, "Configuring Failover," for more information.

The following example assigns VLANs 300 and 301 to bridge group 1, then sets the management address and standby address of bridge group 1:

```
hostname(config)# interface vlan 300
hostname(config-if)# nameif inside
hostname(config-if)# security-level 100
hostname(config-if)# bridge-group 1
hostname(config-if)# interface vlan 301
hostname(config-if)# nameif outside
hostname(config-if)# security-level 0
hostname(config-if)# bridge-group 1
hostname(config-if)# interface bvi 1
hostname(config-if)# ip address 10.1.3.1 255.255.255.0 standby 10.1.3.2
```

Allowing Communication Between Interfaces on the Same Security Level

By default, interfaces on the same security level cannot communicate with each other, even if you configure NAT and access lists. Also, by default, traffic cannot enter and exit the same interface. This section describes how to configure inter-interface and intra-interface communication, and includes the following topics:

Configuring Inter-Interface Communication

Allowing communication between same security interfaces lets you configure more than 101 communicating interfaces. If you use different levels for each interface and do not assign any interfaces to the same security level, you can configure only one interface per level (0 to 100).

Note If you enable NAT control, you do not need to configure NAT between same security level interfaces. See the "NAT and Same Security Level Interfaces" section on page 15-14 for more information on NAT and same security level interfaces.

If you enable same security interface communication, you can still configure interfaces at different security levels as usual.

To enable interfaces on the same security level to communicate with each other, enter the following command:

```
hostname(config)# same-security-traffic permit inter-interface
```

To disable this setting, use the **no** form of this command.

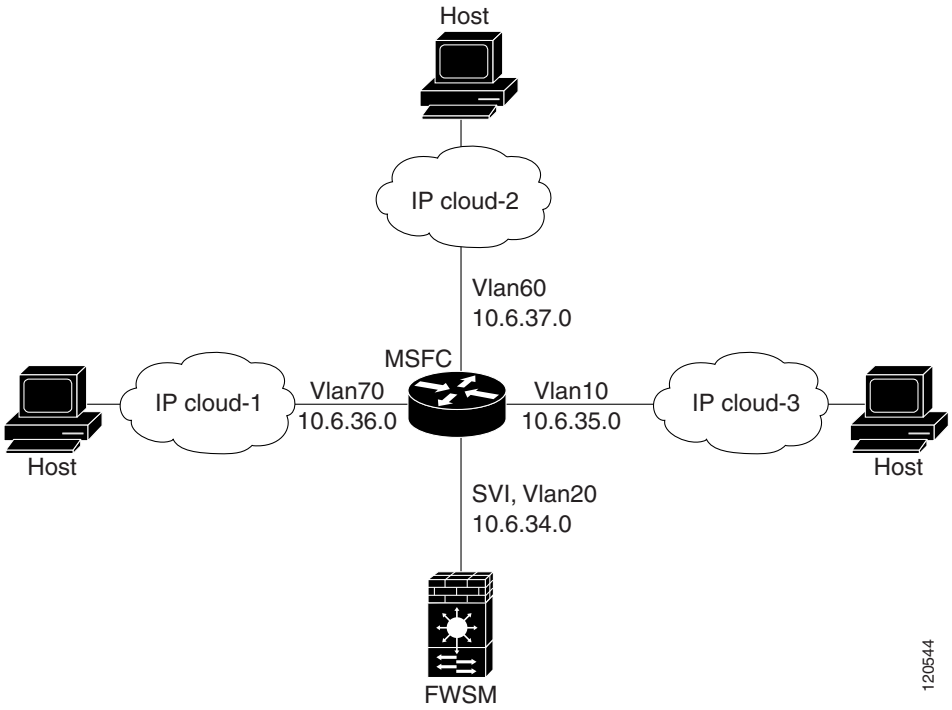
Note If you use a same-security interface for both the outside and inside interfaces, you might want to enable the **xlate-bypass** command; in some situations, you can exceed the maximum number of xlates using that configuration (see the "Managed System Resources" section on page A-4 for limits). For example, without **xlate-bypass**, the FWSM creates xlates for all connections (even if you do not configure NAT). In a same-security-traffic configuration, the FWSM randomly chooses which same-security interface is the "inside" interface for the sake of creating xlates. If the FWSM considers the outside same-security interface as the "inside" interface, it creates xlates for every Internet host being accessed through it. If there is any application (or a virus) on the internal network that scans thousands of Internet hosts, all entries in the xlate table may be quickly exhausted.

Configuring Intra-Interface Communication

You can configure the FWSM to enable communication between two hosts on the same interface. Before you can enable this feature, you must first correctly configure the MSFC so that packets are sent to the FWSM MAC address instead of being sent directly through the switch to the destination host. Figure 6-1 shows a network where hosts on the same interface need to communicate. The following samples show the **route-map** command used to enable policy routing on the MSFC in the network shown in Figure 6-1:

```
Router(config)# route-map intra-inter3 permit 0
Router(config-route-map)# match ip address 103
Router(config-route-map)# set interface Vlan20
Router(config-route-map)# set set ip next-hop 10.6.34.7

Router(config)# route-map intra-inter2 permit 20
Router(config-route-map)# match ip address 102
Router(config-route-map)# set interface Vlan20
Router(config-route-map)# set set ip next-hop 10.6.34.7

Router(config)# route-map intra-inter1 permit 10
Router(config-route-map)# match ip address 101
Router(config-route-map)# set interface Vlan20
Router(config-route-map)# set set ip next-hop 10.6.34.7
```

Figure 6-1 Communication Between Hosts on the Same Interface



When you enable communication between two hosts on the same interface, keep in mind the following requirements:

- Outside NAT is not supported.
- You can configure static routes from one interface to another on the same security level.

To enable communication between hosts on the same security level, enter the following command:

```
hostname(config)# same-security-traffic permit intra-interface
```

To disable these settings, add **no** before the command.

Turning Off and Turning On Interfaces

All interfaces are enabled by default. If you disable or reenable the interface within a context, only that context interface is affected. But if you disable or reenable the interface in the system execution space, then you affect that VLAN interface for all contexts.

To disable an interface or reenable it, perform the following steps:

Step 1 To enter the interface configuration mode, enter the following command:

```
hostname(config)# interface {vlan number | mapped_name}
```

In multiple context mode, enter the mapped name if one was assigned using the **allocate-interface** command.

Step 2 To disable the interface, enter the following command:

```
hostname(config)# shutdown
```

Step 3 To reenable the interface, enter the following command:

```
hostname(config)# no shutdown
```

CHAPTER 7

Configuring Basic Settings

This chapter describes how to configure basic settings on your FWSM that are typically required for a functioning configuration. This chapter includes the following sections:

Changing the Passwords

This section describes how to change the login and enable passwords and includes the following topics:

Note In multiple context mode, every context and the system execution space has its own login policies and passwords.

Changing the Login Password

The login password is used for sessions from the switch as well as Telnet and SSH connections. By default, the login password is "cisco." To change the password, enter the following command:

```
hostname(config)# {passwd | password} password
```

You can enter **passwd** or **password**. The *password* is a case-sensitive password of up to 16 alphanumeric and special characters. You can use any character in the password except a question mark or a space.

The password is saved in the configuration in encrypted form, so you cannot view the original password after you enter it. Use the **no password** command to restore the password to the default setting.

Changing the Enable Password

The enable password lets you enter privileged EXEC mode. By default, the enable password is blank. To change the enable password, enter the following command:

```
hostname(config)# enable password password
```

The *password* is a case-sensitive password of up to 16 alphanumeric and special characters. You can use any character in the password except a question mark or a space.

This command changes the password for the highest privilege level. If you configure local command authorization, you can set enable passwords for each privilege level from 0 to 15.

The password is saved in the configuration in encrypted form, so you cannot view the original password after you enter it. Enter the **enable password** command without a password to set the password to the default, which is blank.

Changing the Maintenance Software Passwords

The maintenance software is valuable for troubleshooting. For example, you can install new software to an application partition, reset passwords, or show crash dump information from the maintenance software. You can only access the maintenance software by sessioning in to the FWSM.

The maintenance software has two user levels with different access privileges:

- root—Lets you configure the network partition parameters, upgrade the software images on the application partitions, change the guest account password, and enable or disable the guest account.

  The default password is “cisco.”
- guest—Lets you configure the network partition parameters and show crash dump information.

  The default password is “cisco.”

To change the maintenance partition passwords for both users, perform the following steps:

Step 1 To reboot the FWSM into the maintenance partition, enter the following command at the switch prompt:

```
Router# hw-module module mod_num reset cf:1
```

Step 2 To session in to the FWSM, enter the following command:

```
Router# session slot mod_num processor 1
```

Step 3 Log in as root by entering the following command:

```
Login: root
```

Step 4 Enter the password at the prompt:

```
Password:
```

The default password is “cisco”.

Step 5 Change the root password by entering the following command:

```
root@localhost# passwd
```

Step 6 Enter the new password at the prompt:

```
Changing password for user root
New password:
```

Step 7 Enter the new password again:

```
Retype new password:
passwd: all authentication tokens updated successfully
```

Step 8 Change the guest password by entering the following command:

```
root@localhost# passwd-guest
```

Step 9 Enter the new password at the prompt:

```
Changing password for user guest
New password:
```

Step 10 Enter the new password again:

```
Retype new password:
passwd: all authentication tokens updated successfully
```

The following example shows how to set the password for the root account:

```
root@localhost# passwd
Changing password for user root
New password: *sh1p
Retype new password: *sh1p
passwd: all authentication tokens updated successfully
```

The following example shows how to set the password for the guest account:

```
root@localhost# passwd-guest
Changing password for user guest
New password: f1rc8t
Retype new password: f1rc8t
passwd: all authentication tokens updated successfully
```

Setting the Hostname

When you set a hostname for the FWSM, that name appears in the command-line prompt. If you establish sessions to multiple devices, the hostname helps you keep track of where you enter commands.

For multiple context mode, the hostname that you set in the system execution space appears in the command-line prompt for all contexts. The hostname that you optionally set within a context does not appear in the command line, but can be used by the **banner** command **$(hostname)** token.

To specify the hostname for the FWSM or for a context, enter the following command:

```
hostname(config)# hostname name
```

This name can be up to 63 characters. A hostname must start and end with a letter or digit, and have as interior characters only letters, digits, or a hyphen. The FWSM supports all 95 printable characters except the question mark (?). Avoid the use of non-ASCII characters.

This name appears in the command-line prompt. For example:

```
hostname(config)# hostname farscape
farscape(config)#
```

Setting the Domain Name

The FWSM appends the domain name as a suffix to unqualified names. For example, if you set the domain name to "example.com," and specify a syslog server by the unqualified name of "jupiter," then the FWSM qualifies the name to "jupiter.example.com."

The default domain name is default.domain.invalid.

For multiple context mode, you can set the domain name for each context, as well as within the system execution space.

To specify the domain name for the FWSM, enter the following command:

```
hostname(config)# domain-name name
```

For example, to set the domain as example.com, enter the following command:

```
hostname(config)# domain-name example.com
```

Setting the Prompt

You can configure the information shown in the CLI prompt, including the hostname, context name, domain name, slot, failover status, and failover priority. In multiple context mode, you can view the extended prompt when you log into the system execution space or the admin context. Within a non-admin context, you only see the default prompt, which is the hostname and the context name.

To configure the information included in the prompt, enter the following command:

```
hostname(config)# prompt [hostname] [context] [domain] [slot] [state] [priority]
```

The order in which you enter the keywords determines the order of the elements in the prompt, which are separated by a slash (/). See the following descriptions for the keywords:

- **hostname**—Displays the hostname.
- **domain**—Displays the domain name.
- **context**—(Multiple mode only) Displays the current context.
- **priority**—Displays the failover priority as pri (primary) or sec (secondary). Set the priority using the **failover lan unit** command.
- **slot**—Displays the slot location in the switch.
- **state**—Displays the traffic-passing state of the unit. The following values are displayed for the **state** keyword:
  - act—Failover is enabled, and the unit is actively passing traffic.
  - stby— Failover is enabled, and the unit is not passing traffic and is in a standby, failed, or other non-active state.
  - actNoFailover—Failover is not enabled, and the unit is actively passing traffic.
  - stbyNoFailover—Failover is not enabled, and the unit is not passing traffic. This might happen when there is an interface failure above the threshold on the standby unit.

For example, to show all available elements in the prompt, enter the following command:

```
hostname(config)# prompt hostname context priority slot state
```

The prompt changes to the following string:

```
hostname/admin/pri/6/act(config)#
```

Configuring a Login Banner

You can configure a message to display when a user connects to the FWSM, when a user logs in to the FWSM using Telnet, or when a user enters user EXEC mode.

To configure a login banner, enter the following command in the system execution space or within a context:

```
hostname(config)# banner {motd | login | exec} text
```

The **motd** keyword shows a banner when a user first connects.

The **login** keyword shows a banner when a user logs in to the FWSM using Telnet.

The **exec** keyword shows a banner when a user accesses user EXEC mode.

When a user connects to the FWSM, the message-of-the-day banner appears first, followed by the login banner and prompts. This banner does not appear for non-Telnet connections. After the user successfully logs in to the FWSM (for Telnet connections), the exec banner displays.

For the banner text, spaces are allowed but you cannot enter tabs using the CLI. You can dynamically add the hostname or domain name of the FWSM by including the strings **$(hostname)** and **$(domain)**. If you configure a banner in the system configuration, you can use that banner text within a context by using the **$(system)** string in the context configuration.

To add more than one line, precede each line by the banner command.

For example, to add a message-of-the-day banner, enter:

```
hostname(config)# banner motd Welcome to $(hostname)
hostname(config)# banner motd Contact me at admin@example.com for any
hostname(config)# banner motd issues
```

CHAPTER 8

Configuring IP Routing and DHCP Services

This chapter describes how to configure IP routing and DHCP on the FWSM. This chapter includes the following sections:

How Routing Behaves Within FWSM

FWSM uses both routing table and XLATE tables for routing decisions. To handle destination-ip-translated, that is, untranslated traffic, FWSM searches for existing XLATE, or static translation to select the egress interface. The selection process is as follows:

Egress Interface Selection Process

- If destination-ip-translating XLATE already exists, the egress interface for the packet is determined from the XLATE table, but not from the routing table.
- If destination-ip-translating XLATE does not exist, but a matching static translation exists, then the egress interface is determined from the static route and an XLATE is created, and the routing table is not used.
- If destination-ip-translating XLATE does not exist and no matching static translation exists, the packet is not destination-ip-translated. FWSM processes this packet by looking up the route to select egress interface, then source-ip translation is performed (if necessary).

Therefore, for regular dynamic outbound NAT, initial outgoing packets are routed using the route table and then create the XLATE. Incoming return packets are forwarded using existing XLATEs only. For static NAT, destination-translated incoming packets are always forwarded using existing XLATE or static translation rules.

Next Hop Selection Process

After selecting egress interface using any method described above, an additional route lookup is performed to find out suitable next hop(s) that belong to previously selected egress interface. If there are no routes in routing table that explicitly belong to selected interface, the packet is dropped with level 6 error message 110001 "`no route to host`", even if there is another route for a given destination network that belongs to different egress interface. If the route that belongs to selected egress interface is found, the packet is forwarded to corresponding next hop.

Load sharing on FWSM is possible only for multiple next-hops available using single egress interface. Load sharing cannot share multiple egress interfaces.

This is not ture if the following conditions exists:

- If dynamic routing is in use on FWSM and route table changes after XLATE creation, for example a route flap happens, then destination-translated traffic is still forwarded using old XLATE, not via route table, until XLATE times out. It may be either forwarded to wrong interface or dropped with message 110001 "`no route to host`" if old route was removed from the old interface and attached to another one by routing process.
- The same problem may happen when there is no route flaps on FWSM itself, but some routing process is flapping around it, sending source-translated packets that belong to the same flow through FWSM using different interfaces. Destination-translated return packets may be forwarded back using the wrong egress interface.

This issue has a high probability in same-security-traffic configuration, where virtually any traffic may be either source-translated or destination-translated, depending on direction of initial packet in the flow.

When this issue occurs after a route flap, it can be resolved manually by using the **clear xlate** command, or automatically resolved by an XLATE timeout. XLATE timeout may be decreased if necessary. To ensure that this rarely happens, make sure that there is no route flaps on FWSM and around it. That is, ensure that destination-translated packets that belong to the same flow are always forwarded the same way through FWSM.

Configuring Static and Default Routes

This section describes how to configure static and default routes on FWSM.

Multiple context mode does not support dynamic routing, so you must use static routes for any networks to which FWSM is not directly connected; for example, when there is a router between a network and FWSM.

You might want to use static routes in single context mode in the following cases:

- Your networks use a different router discovery protocol from RIP or OSPF.
- Your network is small and you can easily manage static routes.
- You do not want the traffic or CPU overhead associated with routing protocols.

The simplest option is to configure a default route to send all traffic to an upstream router, relying on the router to route the traffic for you. However, in some cases the default gateway might not be able to reach the destination network, so you must also configure more specific static routes. For example, if the default gateway is outside, then the default route cannot direct traffic to any inside networks that are not directly connected to FWSM.

In transparent firewall mode, for traffic that originates on FWSM and is destined for a non-directly connected network, you need to configure either a default route or static routes so FWSM knows out of which interface to send traffic. Traffic that originates on FWSM might include communications to a system log server, Websense or N2H2 server, or AAA server. If you have servers that cannot all be reached through a single default route, then you must configure static routes.

Note The default route for the transparent firewall, which is required to provide a return path for management traffic, is only applied to management traffic from one bridge group network. This is because the default route specifies an interface in the bridge group as well as the router IP address on the bridge group network, and you can only define one default route. If you have management traffic from more than one bridge group network, you need to specify a static route that identifies the network from which you expect management traffic.

The FWSM supports up to three equal cost routes to the same destination per interface for load balancing.

This section includes the following topics:

For information about configuring IPv6 static and default routes, see the "Configuring IPv6 Default and Static Routes" section on page 10-5.

Configuring a Static Route

To add a static route, enter the following command:

```
hostname(config)# route if_name dest_ip mask gateway_ip [distance]
```

The *dest_ip* and *mask* is the IP address for the destination network and the *gateway_ip* is the address of the next-hop router.

The *distance* is the administrative distance for the route. The default is 1 if you do not specify a value. Administrative distance is a parameter used to compare routes among different routing protocols. The default administrative distance for static routes is 1, giving it precedence over routes discovered by dynamic routing protocols but not directly connect routes. The default administrative distance for routes discovered by OSPF is 110. If a static route has the same administrative distance as a dynamic route, the static routes take precedence. Connected routes always take precedence over static or dynamically discovered routes.

Static routes remain in the routing table even if the specified gateway becomes unavailable. If the specified gateway becomes unavailable, you need to remove the static route from the routing table manually. However, static routes are removed from the routing table if the associated interface goes down. They are reinstated when the interface comes back up.

Note If you create a static route with an administrative distance greater than the administrative distance of the routing protocol running on the FWSM, then a route to the specified destination discovered by the routing protocol takes precedence over the static route. The static route is used only if the dynamically discovered route is removed from the routing table.

The following example creates a static route that sends all traffic destined for 10.1.1.0/24 to the router (10.1.2.45) connected to the inside interface:

```
hostname(config)# route inside 10.1.1.0 255.255.255.0 10.1.2.45 1
```

You can define up to three equal cost routes to the same destination per interface. ECMP is not supported across multiple interfaces. With ECMP, the traffic is not necessarily divided evenly between the routes; traffic is distributed among the specified gateways based on an algorithm that hashes the source and destination IP addresses.

The following example shows static routes that are equal cost routes that direct traffic to three different gateways on the outside interface. The FWSM distributes the traffic among the specified gateways.

```
hostname(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.1
hostname(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.2
hostname(config)# route outside 10.10.10.0 255.255.255.0 192.168.1.3
```

Configuring a Default Route

A default route identifies the gateway IP address to which FWSM sends all IP packets for which it does not have a learned or static route. A default route is simply a static route with 0.0.0.0/0 as the destination IP address. Routes that identify a specific destination take precedence over the default route.

You can define up to three equal cost default route entries per device. Defining more than one equal cost default route entry causes the traffic sent to the default route to be distributed among the specified gateways. When defining more than one default route, you must specify the same interface for each entry.

If you attempt to define more than three equal cost default routes, or if you attempt to define a default route with a different interface than a previously defined default route, you receive the message "ERROR: Cannot add route entry, possible conflict with existing routes."

To define the default route, enter the following command:

```
hostname(config)# route if_name 0.0.0.0 0.0.0.0 gateway_ip [distance]
```

Tip You can enter 0 0 instead of 0.0.0.0 0.0.0.0 for the destination network address and mask, for example:

```
hostname(config)# route outside 0 0 192.168.1 1
```

The following example shows an FWSM configured with three equal cost default routes. Traffic received by the FWSM for which there is no static or learned route is distributed among the gateways with the IP addresses 192.168.2.1, 192.168.2.2, 192.168.2.3.

```
hostname(config)# route outside 0 0 192.168.2.1
hostname(config)# route outside 0 0 192.168.2.2
hostname(config)# route outside 0 0 192.168.2.3
```

Monitoring a Static or Default Route

Note Currently, you can only monitor routes for one network as specified in the **route-monitor** command.

If you configured multiple static or default routes, FWSM lets you configure multiple routes to monitor whether there are any problems on the active route, and if so, switches to an alternate route on the network in the event a router goes down.

To do this, FWSM route monitoring process starts to send out ICMP queries to determine the best two static route for the destination network and a back up route at a configurable interval of time set. The interval of sending the ICMP query is set by the *interval* keyword; valid values are 100 to 3000, with the default value at 300 milliseconds. The query is always sent to both of the chosen routers, keeping the current available status locally.

The two routes chosen have the least metric distance, with the lowest chosen as the best path to send traffic. In the FWSM, the **route-monitor** command will automatically choose the best two routes among the static routes configured. The next best path always gets installed in the routing table when the current route goes down, and the current one becomes the backup router.

If the ICMP query does not receive a configurable threshold number set by the *failures* keyword, the router is determined to be unreachable. The *failures* keyword is the maximum number of ICMP queries that are not replied to before the router is determined to be down; the default value being five seconds. At this point the backup route takes precedence, provided this route was reachable, and becomes the best route. The original route then becomes the backup route.

If the original best route becomes reachable again, then FWSM switches back to that route and the current best route becomes the backup route. If in case both routes become unreachable, then both are made backup routes. However, there is no change in the routing table.

To monitor a static or default route, and to switch to an alternate path in the event a router goes down, use the Command Line Interface tool to enter the following command.

```
hostname(config-if)# route-monitor network_address network_mask [query_interval interval]
[max-failures failures]
```

Defining a Route Map

Route maps are used to redistribute routes between processes or for route health injection (RHI). To define a route map for use with supported features, perform the following steps:

Step 1 To create a route map entry, enter the following command:

```
hostname(config)# route-map name {permit | deny} [sequence_number]
```

Route map entries are read in order. You can identify the order using the *sequence_number* option, or the FWSM uses the order in which you add the entries.

Step 2 Enter one or more **match** commands:

- To match any routes that have a destination network that matches a standard access list, enter the following command:

  ```
  hostname(config-route-map)# match ip address acl_id [acl_id] [...]
  ```

If you specify more than one access list, then the route can match any of the access lists.

- To match any routes that have a specified metric, enter the following command:

```
hostname(config-route-map)# match metric metric_value
```

The *metric_value* can be from 0 to 4294967295.

- To match any routes that have a next hop router address that matches a standard access list, enter the following command:

```
hostname(config-route-map)# match ip next-hop acl_id [acl_id] [...]
```

If you specify more than one access list, then the route can match any of the access lists.

- To match any routes with the specified next hop interface, enter the following command:

```
hostname(config-route-map)# match interface if_name
```

If you specify more than one interface, then the route can match either interface.

- To match any routes that have been advertised by routers that match a standard access list, enter the following command:

```
hostname(config-route-map)# match ip route-source acl_id [acl_id] [...]
```

If you specify more than one access list, then the route can match any of the access lists.

- To match the route type, enter the following command:

```
hostname(config-route-map)# match route-type {internal | external [type-1 | type-2]}
```

Step 3 Enter one or more **set** commands.

If a route matches the **match** commands, then the following **set** commands determine the action to perform on the route before redistributing it.

- To set the metric, enter the following command:

```
hostname(config-route-map)# set metric metric_value
```

The *metric_value* can be a value between 0 and 294967295

- To set the metric type, enter the following command:

```
hostname(config-route-map)# set metric-type {type-1 | type-2}
```

The following example shows how to redistribute routes with a hop count equal to 1. The FWSM redistributes these routes as external LSAs with a metric of 5, metric type of Type 1, and a tag equal to 1.

```
hostname(config)# route-map 1-to-2 permit
hostname(config-route-map)# match metric 1
hostname(config-route-map)# set metric 5
hostname(config-route-map)# set metric-type type-1
```

Configuring BGP Stub Routing

The FWSM supports BGP stub routing. The BGP stub routing process advertises static and directly connected routes but does not accept routes advertised by the BGP peer.

BGP stub routing is a licensed feature. You must have or obtain a license key that supports BGP stub routing to configure this feature.

This section includes the following topics:

BGP Stub Limitations

The following limitations apply to configuring BGP stub routing on the FWSM:

- You can only configure one BGP routing process, even in multiple context mode.
- You can only configure one BGP neighbor, even in multiple context mode.
- The FWSM does not process UPDATE messages received from the BGP neighbor. It can only send routing updates to the BGP neighbor.
- The FWSM only advertises static routes and directly-connected networks. You cannot redistribute routes from other routing protocols into the BGP routing process.
- In multiple context mode, the FWSM can only advertise static routes and directly-connected networks for the context that contains the interface the BGP peer is reachable through and for which there are configured **network** commands. If the BGP neighbor is reachable through an interface that is shared across multiple contexts, then all of the static routes and directly-connected networks in the contexts sharing the interface are available to the BGP routing process.
- BGP stub does not support IPv6, VPN, or NLRI multicast.
- Only iBGP is supported; eBGP is not supported.

Configuring BGP Stub Routing

Before configuring BGP stub routing on the FWSM:

- You must enable route reflector on the BGP neighbor.
- If the FWSM is in multiple context mode, you must be in the admin context to configure BGP stub routing. Additionally, the admin context must be in routed mode.

Note Although in multiple context mode the BGP routing process is configured in the admin context, only the static routes and directly-connected networks for the context that the BGP peer is reachable through can be advertised.

To enable and configure a BGP routing process, perform the following steps:

Step 1 Create the BGP routing process by entering the following command:

```
hostname(config)# router bgp as-number
```

The *as-number* argument is the autonomous system number that identifies the FWSM to other BGP routers and tags the routing information passed along. It must be the same as the AS number of the BGP neighbor. After entering this command, the command prompt changes to `hostname(config-router)#` to indicate that you are now in router configuration mode for the specified routing process.

Step 2 (Optional) Specify the router ID for the FWSM by entering the following command. If you do not enter a router ID, the highest IP address configured on the FWSM is used.

```
hostname(config-router)# bgp router-id id
```

The *id* can be any IP address, including an IP address that is not configured on the FWSM. If this command is not specified, the router ID used is the highest IP address configured on the FWSM.

Step 3 Specify the BGP neighbor that BGP updates are sent to by entering the following command:

```
hostname(config-router)# neighbor ip-addr remote-as as-number
```

The *ip-addr* argument is the IP address of the BGP neighbor. The *as-number* is the autonomous system number of the BGP neighbor. This should be the same as the AS number configured on the FWSM with the **router** command.

Step 4 (Optional) Enter the password used to authenticate the BGP message to the neighbor. This password must be set on both the neighbor and the FWSM before BGP messages can be exchanged.

```
hostname(config-router)# neighbor ip-addr password [mode] password
```

The *ip-addr* argument is the IP address of the BGP neighbor defined with the **neighbor** command. The *mode* argument can be from 0 to 7. If used, the BGP neighbor must use the same mode. The *password* argument is an alphanumeric string that can contain keyboard symbols but cannot contain spaces.

Step 5 Specify the networks that the BGP routing process advertises using the **network** command. You can configure up to 200 network commands on the FWSM.

```
hostname(config-router)# network ip-addr mask mask
```

The BGP stub routing process only advertises static and directly-connected networks. The **network** command defines which of those networks are advertised in BGP updates.

Monitoring BGP Stub Routing

You can use the following commands to display information about the BGP routing process, neighbor, and advertised routes. In multiple context mode, these commands are entered in the admin context.

- To display information about the BGP routing process, enter the following command:

  ```
  hostname# show ip bgp summary
  ```

- To display BGP neighbor information, enter the following command:

  ```
  hostname# show ip bgp neighbors
  ```

- To display the routes advertised by the BGP routing process, enter the following command:

  ```
  hostname# show ip bgp neighbors advertised-routes
  ```

- To view debug messages for the BGP routing process, enter the following command:

  ```
  hostname# debug ip bgp
  ```

For more detailed information about the output from these commands, see the command information in the *Catalyst 6500 Series and Cisco 7600 Series Switch Firewall Services Module Command Reference*.

Restarting the BGP Stub Routing Process

To clear the BGP session established with the neighbor, clear the statistical counters associated with the session, and restart the BGP with the neighbor, enter the following command:

```
hostname(config)# clear ip bgp neighbor-addr
```

Configuring OSPF

This section describes how to configure OSPF. This section includes the following topics:

OSPF Overview

OSPF uses a link-state algorithm to build and calculate the shortest path to all known destinations. Each router in an OSPF area contains an identical link-state database, which is a list of each of the router usable interfaces and reachable neighbors.

The advantages of OSPF over RIP include the following:

- OSPF link-state database updates are sent less frequently than RIP updates, and the link-state database is updated instantly rather than gradually as stale information is timed out.
- Routing decisions are based on cost, which is an indication of the overhead required to send packets across a certain interface. FWSM calculates the cost of an interface based on link bandwidth rather than the number of hops to the destination. The cost can be configured to specify preferred paths.

The disadvantage of shortest path first algorithms is that they require a lot of CPU cycles and memory.

FWSM can run two processes of OSPF protocol simultaneously, on different sets of interfaces. You might want to run two processes if you have interfaces that use the same IP addresses (NAT allows these interfaces to coexist, but OSPF does not allow overlapping addresses). Or you might want to run one process on the inside, and another on the outside, and redistribute a subset of routes between the two processes. Similarly, you might need to segregate private addresses from public addresses.

Redistribution between the two OSPF processes is supported. Static and connected routes configured on OSPF-enabled interfaces on FWSM can also be redistributed into the OSPF process. You cannot enable RIP on FWSM if OSPF is enabled. Redistribution between RIP and OSPF is not supported.

FWSM supports the following OSPF features:

- Support of intra-area, interarea, and external (Type I and Type II) routes.
- Support of a virtual link.
- OSPF LSA flooding.
- Authentication to OSPF packets (both password and MD5 authentication).
- Support for configuring FWSM as a designated router or a designated backup router. FWSM also can be set up as an ABR; however, the ability to configure the FWSM as an ASBR is limited to default information only (for example, injecting a default route).
- Support for stub areas and not-so-stubby-areas.
- Area boundary router type-3 LSA filtering.
- Advertisement of static and global address translations.

Enabling OSPF

To enable OSPF, you need to create an OSPF routing process, specify the range of IP addresses associated with the routing process, then assign area IDs associated with that range of IP addresses.

Note You cannot enable OSPF if RIP is enabled.

To enable OSPF, perform the following steps:

Step 1 To create an OSPF routing process, enter the following command:

```
hostname(config)# router ospf process_id
```

This command enters the router configuration mode for this OSPF process.

The *process_id* is an internally used identifier for this routing process. It can be any positive integer. This ID does not have to match the ID on any other device; it is for internal use only. You can use a maximum of two processes.

Step 2 To define the IP addresses on which OSPF runs and to define the area ID for that interface, enter the following command:

```
hostname(config-router)# network ip_address mask area area_id
```

The following example shows how to enable OSPF:

```
hostname(config)# router ospf 2
```

```
hostname(config-router)# network 10.0.0.0 255.0.0.0 area 0
```

Redistributing Routes Between OSPF Processes

The FWSM can control the redistribution of routes between OSPF routing processes. The FWSM matches and changes routes according to settings in the **redistribute** command or by using a route map. See also the "Generating a Default Route" section on page 8-18 for another use for route maps.

Note Note: The maximum number of route entries for all types of routes (connected, static and dynamic) supported by FWSM is 32768, or 32K.

To redistribute static, connected, or OSPF routes from one process into another OSPF process, perform the following steps:

Step 1 To create a route map, see the "Defining a Route Map" section on page 8-5.

Step 2 If you have not already done so, enter the router configuration mode for the OSPF process you want to redistribute into by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 3 To specify the routes you want to redistribute, enter the following command:

```
hostname(config-router)# redistribute {ospf process_id
[match {internal | external 1 | external 2}] | static | connect} [metric metric-value]
[metric-type {type-1 | type-2}] [tag tag_value] [subnets] [route-map map_name]
```

The **ospf** *process_id*, **static**, and **connect** keywords specify from where you want to redistribute routes.

You can either use the options in this command to match and set route properties, or you can use a route map. The **tag** and **subnets** options do not have equivalents in the **route-map** command. If you use both a route map and options in the **redistribute** command, then they must match.

The following example shows route redistribution from OSPF process 1 into OSPF process 2 by matching routes with a metric equal to 1. The FWSM redistributes these routes as external LSAs with a metric of 5, metric type of Type 1, and a tag equal to 1.

```
hostname(config)# route-map 1-to-2 permit
hostname(config-route-map)# match metric 1
hostname(config-route-map)# set metric 5
hostname(config-route-map)# set metric-type type-1
hostname(config-route-map)# set tag 1
hostname(config-route-map)# router ospf 2
hostname(config-router)# redistribute ospf 1 route-map 1-to-2
```

The following example shows the specified OSPF process routes being redistributed into OSPF process 109. The OSPF metric is remapped to 100.

```
hostname(config)# router ospf 109
hostname(config-router)# redistribute ospf 108 metric 100 subnets
```

The following example shows route redistribution where the link-state cost is specified as 5 and the metric type is set to external, indicating that it has lower priority than internal metrics.

```
hostname(config)# router ospf 1
```

```
hostname(config-router)# redistribute ospf 2 metric 5 metric-type external
```

Configuring OSPF Interface Parameters

You can alter some interface-specific OSPF parameters as necessary. You are not required to alter any of these parameters, but the following interface parameters must be consistent across all routers in an attached network: **ospf hello-interval**, **ospf dead-interval**, and **ospf authentication-key**. Be sure that if you configure any of these parameters, the configurations for all routers on your network have compatible values.

To configure OSPF interface parameters, perform the following steps:

Step 1 To enter the interface configuration mode, enter the following command:

```
hostname(config)# interface if_name
```

Step 2 Enter any of the following commands:

- To specify the authentication type for an interface, enter the following command:

  ```
  hostname(config-interface)# ospf authentication [message-digest | null]
  ```

- To assign a password to be used by neighboring OSPF routers on a network segment that is using the OSPF simple password authentication, enter the following command:

  ```
  hostname(config-interface)# ospf authentication-key key
  ```

  The *key* can be any continuous string of characters up to 8 bytes in length.

  The password created by this command is used as a key that is inserted directly into the OSPF header when the FWSM software originates routing protocol packets. A separate password can be assigned to each network on a per-interface basis. All neighboring routers on the same network must have the same password to be able to exchange OSPF information.

- To explicitly specify the cost of sending a packet on an OSPF interface, enter the following command:

  ```
  hostname(config-interface)# ospf cost cost
  ```

  The *cost* is an integer from 1 to 65535.

- To set the number of seconds that a device must wait before it declares a neighbor OSPF router down because it has not received a hello packet, enter the following command:

  ```
  hostname(config-interface)# ospf dead-interval seconds
  ```

  The value must be the same for all nodes on the network.

- To specify the length of time between the hello packets that the FWSM sends on an OSPF interface, enter the following command:

  ```
  hostname(config-interface)# ospf hello-interval seconds
  ```

  The value must be the same for all nodes on the network.

- To enable OSPF MD5 authentication, enter the following command:

  ```
  hostname(config-interface)# ospf message-digest-key key_id md5 key
  ```

  Set the following values:

  – *key_id*—An identifier in the range from 1 to 255.

  – *key*—Alphanumeric password of up to 16 bytes.

  Usually, one key per interface is used to generate authentication information when sending packets and to authenticate incoming packets. The same key identifier on the neighbor router must have the same key value.

  We recommend that you not keep more than one key per interface. Every time you add a new key, you should remove the old key to prevent the local system from continuing to communicate with a hostile system that knows the old key. Removing the old key also reduces overhead during rollover.

- To set the priority to help determine the OSPF designated router for a network, enter the following command:

  ```
  hostname(config-interface)# ospf priority number_value
  ```

  The *number_value* is between 0 to 255.

- To specify the number of seconds between LSA retransmissions for adjacencies belonging to an OSPF interface, enter the following command:

  ```
  hostname(config-interface)# ospf retransmit-interval seconds
  ```

  The *seconds* must be greater than the expected round-trip delay between any two routers on the attached network. The range is from 1 to 65535 seconds. The default is 5 seconds.

- To set the estimated number of seconds required to send a link-state update packet on an OSPF interface, enter the following command:

  ```
  hostname(config-interface)# ospf transmit-delay seconds
  ```

  The *seconds* is from 1 to 65535 seconds. The default is 1 second.

The following example shows how to configure the OSPF interfaces:

```
hostname(config)# router ospf 2
hostname(config-router)# network 10.1.1.0 255.255.255.0 area 0
hostname(config-router)# interface inside
hostname(config-interface)# ospf cost 20
hostname(config-interface)# ospf retransmit-interval 15
hostname(config-interface)# ospf transmit-delay 10
hostname(config-interface)# ospf priority 20
hostname(config-interface)# ospf hello-interval 10
hostname(config-interface)# ospf dead-interval 40
hostname(config-interface)# ospf authentication-key cisco
hostname(config-interface)# ospf message-digest-key 1 md5 cisco
hostname(config-interface)# ospf authentication message-digest
```

The following is sample output from the **show ospf** command:

```
hostname(config)# show ospf

Routing Process "ospf 2" with ID 20.1.89.2 and Domain ID 0.0.0.2
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 5. Checksum Sum 0x 26da6
Number of opaque AS LSA 0. Checksum Sum 0x     0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
    Area BACKBONE(0)
```

```
Number of interfaces in this area is 1
Area has no authentication
SPF algorithm executed 2 times
Area ranges are
Number of LSA 5. Checksum Sum 0x 209a3
Number of opaque link LSA 0. Checksum Sum 0x     0
Number of DCbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0
```

Configuring OSPF Area Parameters

You can configure several area parameters. These area parameters (shown in the following task table) include setting authentication, defining stub areas, and assigning specific costs to the default summary route. Authentication provides password-based protection against unauthorized access to an area.

Stub areas are areas into which information on external routes is not sent. Instead, there is a default external route generated by the ABR, into the stub area for destinations outside the autonomous system. To take advantage of the OSPF stub area support, default routing must be used in the stub area. To further reduce the number of LSAs sent into a stub area, you can configure the **no-summary** keyword of the **area stub** command on the ABR to prevent it from sending summary link advertisement (LSA type 3) into the stub area.

To specify area parameters for your network, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 Enter any of the following commands:

- To enable authentication for an OSPF area, enter the following command:

  ```
  hostname(config-router)# area area-id authentication
  ```

- To enable MD5 authentication for an OSPF area, enter the following command:

  ```
  hostname(config-router)# area area-id authentication message-digest
  ```

- To define an area to be a stub area, enter the following command:

  ```
  hostname(config-router)# area area-id stub [no-summary]
  ```

- To assign a specific cost to the default summary route used for the stub area, enter the following command:

  ```
  hostname(config-router)# area area-id default-cost cost
  ```

  The *cost* is an integer from 1 to 65535. The default is 1.

The following example shows how to configure the OSPF area parameters:

```
hostname(config)# router ospf 2
hostname(config-router)# area 0 authentication
hostname(config-router)# area 0 authentication message-digest
hostname(config-router)# area 17 stub
```

```
hostname(config-router)# area 17 default-cost 20
```

Configuring OSPF NSSA

The OSPF implementation of an NSSA is similar to an OSPF stub area. NSSA does not flood type 5 external LSAs from the core into the area, but it can import autonomous system external routes in a limited way within the area.

NSSA imports type 7 autonomous system external routes within an NSSA area by redistribution. These type 7 LSAs are translated into type 5 LSAs by NSSA ABRs, which are flooded throughout the whole routing domain. Summarization and filtering are supported during the translation.

You can simplify administration if you are an ISP or a network administrator that must connect a central site using OSPF to a remote site that is using a different routing protocol using NSSA.

Before the implementation of NSSA, the connection between the corporate site border router and the remote router could not be run as an OSPF stub area because routes for the remote site could not be redistributed into the stub area, and two routing protocols needed to be maintained. A simple protocol such as RIP was usually run and handled the redistribution. With NSSA, you can extend OSPF to cover the remote connection by defining the area between the corporate router and the remote router as an NSSA.

To specify area parameters for your network as needed to configure OSPF NSSA, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 Enter any of the following commands:

- To define an NSSA area, enter the following command:

  ```
  hostname(config-router)# area area-id nssa [no-redistribution]
  [default-information-originate]
  ```

- To summarize groups of addresses, enter the following command:

  ```
  hostname(config-router)# summary address ip_address mask [not-advertise] [tag tag]
  ```

  This command helps reduce the size of the routing table. Using this command for OSPF causes an OSPF ASBR to advertise one external route as an aggregate for all redistributed routes that are covered by the address.

  OSPF does not support **summary-address 0.0.0.0 0.0.0.0**.

  In the following example, the summary address 10.1.0.0 includes address 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the address 10.1.0.0 is advertised in an external link-state advertisement.

  ```
  hostname(config-router)# summary-address 10.1.1.0 255.255.0.0
  ```

  Before you use this feature, consider these guidelines:

  – You can set a type 7 default route that can be used to reach external destinations. When configured, the router generates a type 7 default into the NSSA or the NSSA area boundary router.

– Every router within the same area must agree that the area is NSSA; otherwise, the routers will not be able to communicate.

Configuring a Point-To-Point, Non-Broadcast OSPF Neighbor

You need to define a static OSPF neighbor to advertise OSPF routes over a point-to-point, non-broadcast network. When an interface is configured as point-to-point, the following restrictions apply:

- You can define only one OSPF neighbor for the interface.
- You need to define a static route pointing to the OSPF neighbor if it is not on a directly connected network.
- The interface cannot form adjacencies unless neighbors are configured explicitly.

To define an OSPF neighbor on a point-to-point, non-broadcast network, perform the following tasks:

Step 1 If the OSPF neighbor is not on a directly-connected network, create a static route to the OSPF neighbor. Do not use the default route. See the "Configuring a Static Route" section on page 8-3 for more information about creating static routes.

Step 2 Define the OSPF neighbor by performing the following tasks:

a. Enter router configuration mode for the OSPF process. Enter the following command:

```
hostname(config)# router ospf pid
```

b. Define the OSPF neighbor by entering the following command:

```
hostname(config-router)# neighbor addr [interface if_name]
```

The *addr* argument is the IP address of the OSPF neighbor. The *if_name* is the interface used to communicate with the neighbor. If the OSPF neighbor is not on the same network as any of the directly-connected interfaces, you must specify the **interface**.

c. If not already configured, define the networks and associated area ID for the interface facing the OSPF neighbor by entering the following command:

```
hostname(config-router)# network addr mask area area_id
```

The *addr mask* pair must cover the IP address of the interface.

Step 3 Configure the interface through which the FWSM communicates with the neighbor by entering the following commands:

```
hostname(config)# interface vlan
hostname(config-if)# ospf network point-to-point non-broadcast
```

The following example shows how to configure OSPF across a point-to-point, non-broadcast network. The OSPF neighbor is not on a directly-connected network, so a static route is needed.

```
hostname(config)# route ospf_outside 10.3.3.0 255.255.255.0 10.1.1.99 1

hostname(config)# interface Vlan55
hostname(config-if)# nameif outside
hostname(config-if)# security-level 0
hostname(config-if)# ip address 10.1.1.1 255.255.255.0 standby 10.1.1.2
hostname(config-if)# ospf network point-to-point non-broadcast
```

```
hostname(config-if)# exit

hostname(config)# router ospf 1
hostname(config-router)# network 10.1.1.0 255.255.255.0 area 100
hostname(config-router)# neighbor 10.3.3.1 interface outside
hostname(config-router)# log-adj-changes
```

Configuring Route Summarization Between OSPF Areas

Route summarization is the consolidation of advertised addresses. This feature causes a single summary route to be advertised to other areas by an area boundary router. In OSPF, an area boundary router advertises networks in one area into another area. If the network numbers in an area are assigned in a way such that they are contiguous, you can configure the area boundary router to advertise a summary route that covers all the individual networks within the area that fall into the specified range.

To define an address range for route summarization, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 To set the address range, enter the following command:

```
hostname(config-router)# area area-id range ip-address mask [advertise | not-advertise]
```

The following example shows how to configure route summarization between OSPF areas:

```
hostname(config)# router ospf 1
hostname(config-router)# area 17 range 12.1.0.0 255.255.0.0
```

Configuring Route Summarization when Redistributing Routes into OSPF

When routes from other protocols are redistributed into OSPF, each route is advertised individually in an external LSA. However, you can configure the FWSM to advertise a single route for all the redistributed routes that are covered by a specified network address and mask. This configuration decreases the size of the OSPF link-state database.

To configure the software advertisement on one summary route for all redistributed routes covered by a network address and mask, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 To set the summary address, enter the following command:

```
hostname(config-router)# summary-address ip_address mask [not-advertise] [tag tag]
```

OSPF does not support **summary-address 0.0.0.0 0.0.0.0**.

The following example shows how to configure route summarization. The summary address 10.1.0.0 includes address 10.1.1.0, 10.1.2.0, 10.1.3.0, and so on. Only the address 10.1.0.0 is advertised in an external link-state advertisement:

```
hostname(config)# router ospf 1
hostname(config-router)# summary-address 10.1.0.0 255.255.0.0
```

Generating a Default Route

You can force an ASBR to generate a default route into an OSPF routing domain. Whenever you specifically configure redistribution of routes into an OSPF routing domain, the router automatically becomes an ASBR. However, an ASBR does not by default generate a default route into the OSPF routing domain.

To generate a default route, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 To force the ASBR to generate a default route, enter the following command:

```
hostname(config-router)# default-information originate [always] [metric metric-value]
[metric-type {1 | 2}] [route-map map-name]
```

The following example shows how to generate a default route:

```
hostname(config)# router ospf 2
hostname(config-router)# default-information originate always
```

Configuring Route Calculation Timers

You can configure the delay time between when OSPF receives a topology change and when it starts an SPF calculation. You also can configure the hold time between two consecutive SPF calculations.

To configure route calculation timers, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 To configure the route calculation time, enter the following command:

```
hostname(config-router)# timers spf spf-delay spf-holdtime
```

The *spf-delay* is the delay time (in seconds) between when OSPF receives a topology change and when it starts an SPF calculation. It can be an integer from 0 to 65535. The default time is 5 seconds. A value of 0 means that there is no delay; that is, the SPF calculation is started immediately.

The *spf-holdtime* is the minimum time (in seconds) between two consecutive SPF calculations. It can be an integer from 0 to 65535. The default time is 10 seconds. A value of 0 means that there is no delay; that is, two SPF calculations can be done, one immediately after the other.

The following example shows how to configure route calculation timers:

```
hostname(config)# router ospf 1
hostname(config-router)# timers spf 10 120
```

Logging Neighbors Going Up or Down

By default, the system sends a system log message when an OSPF neighbor goes up or down.

Configure this command if you want to know about OSPF neighbors going up or down without turning on the **debug ospf adjacency** command. The **log-adj-changes** router configuration command provides a higher level view of the peer relationship with less output. Configure **log-adj-changes detail** if you want to see messages for each state change.

To log neighbors going up or down, perform the following steps:

Step 1 If you have not already done so, enter the router configuration mode for the OSPF process you want to configure by entering the following command:

```
hostname(config)# router ospf process_id
```

Step 2 To configure logging for neighbors going up or down, enter the following command:

```
hostname(config-router)# log-adj-changes [detail]
```

Note Logging must be enabled for the neighbor up/down messages to be sent.

The following example shows how to log neighbors up/down messages:

```
hostname(config)# router ospf 1
hostname(config-router)# log-adj-changes detail
```

Displaying OSPF Update Packet Pacing

OSPF update packets are automatically paced so they are not sent less than 33 milliseconds apart. Without pacing, some update packets could get lost in situations where the link is slow, a neighbor could not receive the updates quickly enough, or the router could run out of buffer space. For example, without pacing packets might be dropped if either of the following topologies exist:

- A fast router is connected to a slower router over a point-to-point link.
- During flooding, several neighbors send updates to a single router at the same time.

Pacing is also used between resends to increase efficiency and minimize lost retransmissions. You also can display the LSAs waiting to be sent out an interface. The benefit of the pacing is that OSPF update and retransmission packets are sent more efficiently.

There are no configuration tasks for this feature; it occurs automatically.

To observe OSPF packet pacing by displaying a list of LSAs waiting to be flooded over a specified interface, enter the following command:

```
hostname# show ospf flood-list if_name
```

Monitoring OSPF

You can display specific statistics such as the contents of IP routing tables, caches, and databases. You can use the information provided to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path that your device packets are taking through the network.

To display various routing statistics, perform one of the following tasks, as needed:

- To display general information about OSPF routing processes, enter the following command:

  ```
  hostname# show ospf [process-id [area-id]]
  ```

- To display the internal OSPF routing table entries to the ABR and ASBR, enter the following command:

  ```
  hostname# show ospf border-routers
  ```

- To display lists of information related to the OSPF database for a specific router, enter the following command:

  ```
  hostname# show ospf [process-id [area-id]] database
  ```

- To display a list of LSAs waiting to be flooded over an interface (to observe OSPF packet pacing), enter the following command:

  ```
  hostname# show ospf flood-list if-name
  ```

- To display OSPF-related interface information, enter the following command:

  ```
  hostname# show ospf interface [if_name]
  ```

- To display OSPF neighbor information on a per-interface basis, enter the following command:

  ```
  hostname# show ospf neighbor [if-name] [neighbor-id] [detail]
  ```

- To display a list of all LSAs requested by a router, enter the following command:

  ```
  hostname# show ospf request-list neighbor if_name
  ```

- To display a list of all LSAs waiting to be resent, enter the following command:

  ```
  hostname# show ospf retransmission-list neighbor if_name
  ```

- To display a list of all summary address redistribution information configured under an OSPF process, enter the following command:

  ```
  hostname# show ospf [process-id] summary-address
  ```

- To display OSPF-related virtual links information, enter the following command:

  ```
  hostname# show ospf [process-id] virtual-links
  ```

Restarting the OSPF Process

To restart an OSPF process, clear redistribution, or counters, enter the following command:

```
hostname(config)# clear ospf pid {process | redistribution | counters
[neighbor [neighbor-interface] [neighbor-id]]}
```

Configuring RIP

This section describes how to configure RIP. This section includes the following topics:

RIP Overview

Devices that support RIP send routing-update messages at regular intervals and when the network topology changes. These RIP packets contain information about the networks that the devices can reach, as well as the number of routers or gateways that a packet must travel through to reach the destination address. RIP generates more traffic than OSPF, but is easier to configure initially.

RIP has advantages over static routes because the initial configuration is simple, and you do not need to update the configuration when the topology changes. The disadvantage to RIP is that there is more network and processing overhead than static routing. Additionally, RIP cannot be enabled on a global basis.

FWSM uses a limited version of RIP; it does not send out RIP updates that identify the networks that FWSM can reach. However, you can enable one or both of the following methods:

- Passive RIP—FWSM listens for RIP updates but does not send any updates about its networks out of the interface.

  Passive RIP allows FWSM to learn about networks to which it is not directly connected.

- Default Route Updates—Instead of sending normal RIP updates that describe all the networks reachable through FWSM, FWSM sends a default route to participating devices that identifies FWSM as the default gateway.

  You can use the default route option with passive RIP, or alone. You might use the default route option alone if you use static routes on FWSM, but do not want to configure static routes on downstream routers. Typically, you would not enable the default route option on the outside interface, because FWSM is not typically the default gateway for the upstream router.

Enabling RIP

To enable RIP on an interface, enter the following command:

```
hostname(config)# rip if_name {default | passive} [version {1 | 2
[authentication {text | md5} key key_id]}]
```

You can enable both the passive and default modes of RIP on an interface by entering the **rip** command twice, one time for each method. For example, enter the following commands:

```
hostname(config)# rip inside default version 2 authentication md5 scorpius 1
```

```
hostname(config)# rip inside passive version 2 authentication md5 scorpius 1
```

If you want to enable passive RIP on all interfaces, but only enable default routes on the inside interface, enter the following commands:

```
hostname(config)# rip inside default version 2 authentication md5 scorpius 1
hostname(config)# rip inside passive version 2 authentication md5 scorpius 1
hostname(config)# rip outside passive version 2 authentication md5 scorpius 1
```

Note Before testing your configuration, flush the ARP caches on any routers connected to the FWSM. For Cisco routers, use the **clear arp** command to flush the ARP cache.

You cannot enable RIP if OSPF is enabled.

Configuring EIGRP

This section describes the configuration and monitoring of EIGRP routing and includes the following topics:

EIGRP Routing Overview

EIGRP is an enhanced version of IGRP developed by Cisco. Unlike IGRP and RIP, EIGRP does not send out periodic route updates. EIGRP updates are sent out only when the network topology changes.

Neighbor discovery is the process that the FWSM uses to dynamically learn of other routers on directly attached networks. EIGRP routers send out multicast hello packets to announce their presence on the network. When the FWSM receives a hello packet from a new neighbor, it sends its topology table to the neighbor with an initialization bit set. When the neighbor receives the topology update with the initialization bit set, the neighbor sends its topology table back to the FWSM.

The hello packets are sent out as multicast messages. No response is expected to a hello message. The exception to this is for statically defined neighbors. If you use the **neighbor** command to configure a neighbor, the hello messages sent to that neighbor are sent as unicast messages. Routing updates and acknowledgements are sent out as unicast messages.

Once this neighbor relationship is established, routing updates are not exchanged unless there is a change in the network topology. The neighbor relationship is maintained through the hello packets. Each hello packet received from a neighbor contains a hold time. This is the time in which the FWSM can expect to receive a hello packet from that neighbor. If the FWSM does not receive a hello packet from that neighbor within the hold time advertised by that neighbor, the FWSM considers that neighbor to be unavailable.

The EIGRP uses an algorithm called DUAL for route computations. DUAL saves all routes to a destination in the topology table, not just the least-cost route. The least-cost route is inserted into the routing table. The other routes remain in the topology table. If the main route fails, another route is chosen from the feasible successors. A successor is a neighboring router used for packet forwarding that has a least-cost path to a destination. The feasibility calculation guarantees that the path is not part of a routing loop.

If a feasible successor is not found in the topology table, a route recomputation must occur. During route recomputation, DUAL queries the EIGRP neighbors for a route, who in turn query their neighbors. Routers that do no have a feasible successor for the route return an unreachable message.

During route recomputation, DUAL marks the route as active. By default, the FWSM waits for three minutes to receive a response from its neighbors. If the FWSM does not receive a response from a neighbor, the route is marked as stuck-in-active. All routes in the topology table that point to the unresponsive neighbor as a feasibility successor are removed.

Enabling and Configuring EIGRP Routing

You can only enable one EIGRP routing process on the FWSM.

To enable and configure EIGRP routing, perform the following tasks:

Step 1 Create the EIGRP routing process and enter router configuration mode for that process by entering the following command:

```
hostname(config)# router eigrp as-num
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 To configure the interfaces and networks that participate in EIGRP routing, configure one or more **network** statements by entering the following command:

```
hostname(config-router)# network ip-addr [mask]
```

Directly-connected networks that fall within the defined network are advertised by the FWSM. Additionally, only interfaces with an IP address that fall within the defined network participate in the EIGRP routing process.

If you have an interface that you do not want to participate in EIGRP routing, but that is attached to a network that you want advertised, configure a **network** command that covers the network the interface is attached to, and use the **passive-interface** command to prevent that interface from sending or receiving EIGRP updates.

Step 3 (Optional) To prevent an interface from sending or receiving EIGRP routing message, enter the following command:

```
hostname(config-router)# passive-interface {default | if-name}
```

Using the **default** keyword disables EIGRP routing updates on all interfaces. Specifying an interface name, as defined by the **nameif** command, disables EIGRP routing updates on the specified interface. You can have multiple **passive-interface** commands in your EIGRP router configuration.

Step 4 (Optional) To control the sending or receiving of candidate default route information, enter the following command:

```
hostname(config-router)# no default-information {in | out}
```

Configuring **no default-information in** causes the candidate default route bit to be blocked on received routes. Configuring **no default-information out** disables the setting of the default route bit in advertised routes.

Step 5 (Optional) To filter networks sent in EIGRP routing updates, perform the following steps:

a. Create a standard access list that defines the routes you want to advertise.

b. Enter the following command to apply the filter. You can specify an interface to apply the filter to only those updates sent by that interface.

```
hostname(config-router): distribute-list acl out [interface if_name]
```

You can enter multiple **distribute-list** commands in your EIGRP router configuration.

Step 6 (Optional) To filter networks received in EIGRP routing updates, perform the following steps:

a. Create a standard access list that defines the routes you want to filter from received updates.

b. Enter the following command to apply the filter. You can specify an interface to apply the filter to only those updates received by that interface.

```
hostname(config-router): distribute-list acl in [interface if_name]
```

You can enter multiple **distribute-list** commands in your EIGRP router configuration.

Enabling and Configuring EIGRP Stub Routing

You can configure the FWSM as an EIGRP stub router. Stub routing decreases memory and processing requirements on the FWSM. As a stub router, the FWSM does not need to maintain a complete EIGRP routing table because it forwards all nonlocal traffic to a distribution router. Generally, the distribution router need not send anything more than a default route to the stub router.

Only specified routes are propagated from the stub router to the distribution router. As a stub router, the FWSM responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message “inaccessible.” When the FWSM is configured as a stub, it sends a special peer information packet to all neighboring routers to report its status as a stub router. Any neighbor that receives a packet informing it of the stub status will not query the stub router for any routes, and a router that has a stub peer will not query that peer. The stub router depends on the distribution router to send the proper updates to all peers.

To enable and configure and EIGRP stub routing process, perform the following steps:

Step 1 Create the EIGRP routing process and enter router configuration mode for that process by entering the following command:

```
hostname(config)# router eigrp as-num
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Configure the interface connected to the distribution router to participate in EIGRP by entering the following command:

```
hostname(config-router)# network ip-addr [mask]
```

Step 3 Configure the stub routing process by entering the following command. You must specify which networks are advertised by the stub routing process to the distribution router. Static and connected networks are not automatically redistributed into the stub routing process.

```
hostname(config-router)# eigrp stub {receive-only | [connected] [redistributed] [static]
[summary]}
```

Enabling EIGRP Authentication

EIGRP route authentication provides MD5 authentication of routing updates from the EIGRP routing protocol. The MD5 keyed digest in each EIGRP packet prevents the introduction of unauthorized or false routing messages from unapproved sources.

EIGRP route authentication is configured on a per-interface basis. All EIGRP neighbors on interfaces configured for EIGRP message authentication must be configured with the same authentication mode and key for adjacencies to be established.

Before you can enable EIGRP route authentication, you must enable EIGRP.

To enable EIGRP authentication on an interface, perform the following steps:

Step 1 Enter interface configuration mode for the interface on which you are configuring EIGRP message authentication by entering the following command:

```
hostname(config)# interface phy_if
```

Step 2 Enable MD5 authentication of EIGRP packets by entering the following command:

```
hostname(config-if)# authentication mode eigrp as-num md5
```

The *as-num* argument is the autonomous system number of the EIGRP routing process configured on the FWSM. If EIGRP is not enabled or if you enter the wrong number, the FWSM returns the following error message:

```
% Asystem(100) specified does not exist
```

Step 3 Configure the key used by the MD5 algorithm by entering the following command:

```
hostname(config-if)# authentication key eigrp as-num key key-id key-id
```

The *as-num* argument is the autonomous system number of the EIGRP routing process configured on the FWSM. If EIGRP is not enabled or if you enter the wrong number, the FWSM returns the following error message:

```
% Asystem(100) specified does not exist
```

The *key* argument can contain up to 16 characters. The *key-id* argument is a number from 0 to 255.

Defining an EIGRP Neighbor

EIGRP hello packets are sent as multicast packets. If an EIGRP neighbor is located across a nonbroadcast network, such as a tunnel, you must manually define that neighbor. When you manually define an EIGRP neighbor, hello packets are sent to that neighbor as unicast messages.

To manually define an EIGRP neighbor, perform the following steps:

Step 1 Enter router configuration mode for the EIGRP routing process by entering the following command:

```
hostname(config)# router eigrp as-num
```

The *as-num* argument is the autonomous system number of the EIGRP routing process.

Step 2 Define the static neighbor by entering the following command:

```
hostname(config-router)# neighbor ip-addr interface if_name
```

The *ip-addr* argument is the IP address of the neighbor. The *if-name* argument is the name of the interface, as specified by the **nameif** command, through which that neighbor is available. You can define multiple neighbors for an EIGRP routing process.

Redistributing Routes Into EIGRP

You can redistribute routes discovered by OSPF into the EIGRP routing process. You can also redistribute static and connected routes into the EIGRP routing process. You do not need to redistribute static or connected routes if they fall within the range of a **network** statement in the EIGRP configuration.

To redistribute routes into the EIGRP routing process, perform the following steps:

Step 1 (Optional) Create a route-map to further define which routes from the specified routing protocol are redistributed in to the RIP routing process. See the "Defining a Route Map" section on page 8-5 for more information about creating a route map.

Step 2 Enter router configuration mode for the EIGRP routing process:

```
hostname(config)# router eigrp as-num
```

Step 3 (Optional) Specify the default metrics that should be applied to routes redistributed into the EIGRP routing process by entering the following command:

```
hostname(config-router)# default-metric bandwidth delay reliability loading mtu
```

If you do not specify a **default-metric** in the EIGRP router configuration, you must specify the metric values in each **redistribute** command. If you specify the EIGRP metrics in the **redistribute** command and have the **default-metric** command in the EIGRP router configuration, the metrics in the **redistribute** command are used.

Step 4 Choose one of the following options to redistribute the selected route type into the EIGRP routing process.

- To redistribute connected routes into the EIGRP routing process, enter the following command:

  ```
  hostname(config-router): redistribute connected [metric bandwidth delay reliability
  loading mtu] [route-map map_name]
  ```

- To redistribute static routes into the EIGRP routing process, enter the following command:

```
hostname(config-router): redistribute static [metric bandwidth delay reliability
loading mtu] [route-map map_name]
```

- To redistribute routes from an OSPF routing process into the EIGRP routing process, enter the following command:

```
hostname(config-router): redistribute ospf pid [match {internal | external [1 | 2] |
nssa-external [1 | 2]}] [metric bandwidth delay reliability loading mtu] [route-map
map_name]
```

You must specify the EIGRP metric values in the **redistribute** command if you do not have a **default-metric** command in the EIGRP router configuration.

Configuring the EIGRP Hello Interval and Hold Time

The FWSM periodically sends hello packets to discover neighbors and to learn when neighbors become unreachable or inoperative. By default, hello packets are sent every 5 seconds.

The hello packet advertises the FWSM hold time. The hold time indicates to EIGRP neighbors the length of time the neighbor should consider the FWSM reachable. If the neighbor does not receive a hello packet within the advertised hold time, then the FWSM is considered unreachable. By default, the advertised hold time is 15 seconds (three times the hello interval).

Both the hello interval and the advertised hold time are configured on a per-interface basis. We recommend setting the hold time to be at minimum three times the hello interval.

To configure the hello interval and advertised hold time, perform the following steps:

Step 1 Enter interface configuration mode for the interface on which you are configuring hello interval or advertised hold time by entering the following command:

```
hostname(config)# interface phy_if
```

Step 2 To change the hello interval, enter the following command:

```
hostname(config)# hello-interval eigrp as-num seconds
```

Step 3 To change the hold time, enter the following command:

```
hostname(config)# hold-time eigrp as-num seconds
```

Disabling Automatic Route Summarization

Automatic route summarization is enabled by default. The EIGRP routing process summarizes on network number boundaries. This can cause routing problems if you have non-contiguous networks.

For example, if you have a router with the networks 192.168.1.0, 192.168.2.0, and 192.168.3.0 connected to it, and those networks all participate in EIGRP, the EIGRP routing process creates the summary address 192.168.0.0 for those routes. If an additional router is added to the network with the networks 192.168.10.0 and 192.168.11.0, and those networks participate in EIGRP, they will also be summarized as 192.168.0.0. To prevent the possibility of traffic being routed to the wrong location, you should disable automatic route summarization on the routers creating the conflicting summary addresses.

To disable automatic router summarization, enter the following command in router configuration mode for the EIGRP routing process:

```
hostname(config-router)# no auto-summary
```

Note Automatic summary addresses have an adminstrative distance of 5. You cannot configure this value.

Configuring Summary Aggregate Addresses

You can configure a summary addresses on a per-interface basis. You need to manually define summary addresses if you want to create summary addresses that do not occur at a network number boundary or if you want to use summary addresses on a FWSM with automatic route summarization disabled. If any more specific routes are in the routing table, EIGRP will advertise the summary address out the interface with a metric equal to the minimum of all more specific routes.

To create a summary address, perform the following steps:

Step 1 Enter interface configuration mode for the interface on which you are creating a summary address by entering the following command:

```
hostname(config)# interface phy_if
```

Step 2 Create the summary address by entering the following command:

```
hostname(config-if)# summary-address eigrp as-num address mask [distance]
```

By default, EIGRP summary addresses that you define have an administrative distance of 5. You can change this value by specifying the optional *distance* argument in the **summary-address** command.

Disabling EIGRP Split Horizon

Split horizon controls the sending of EIGRP update and query packets. When split horizon is enabled on an interface, update and query packets are not sent for destinations for which this interface is the next hop. Controlling update and query packets in this manner reduces the possibility of routing loops.

By default, split horizon is enabled on all interfaces.

Split horizon blocks route information from being advertised by a router out of any interface from which that information originated. This behavior usually optimizes communications among multiple routing devices, particularly when links are broken. However, with nonbroadcast networks, there may be situations where this behavior is not desired. For these situations, including networks in which you have EIGRP configured, you may want to disable split horizon.

If you disable split horizon on an interface, you must disable it for all routers and access servers on that interface.

To disable EIGRP split-horizon, perform the following steps:

Step 1 Enter interface configuration mode for the interface on which you are disabling split horizon by entering the following command:

```
hostname(config)# interface phy_if
```

Step 2 To disable split horizon, enter the following command:

```
hostname(config-if)# no split-horizon eigrp as-number
```

Changing the Interface Delay Value

The interface delay value is used in EIGRP distance calculations. You can modify this value on a per-interface basis.

To change the delay value, perform the following steps:

Step 1 Enter interface configuration mode for the interface on which you are changing the delay value used by EIGRP by entering the following command:

```
hostname(config)# interface phy_if
```

Step 2 To change the delay value, enter the following command:

```
hostname(config-if)# delay value
```

The *value* entered is in tens of microseconds. So, to set the delay for 2000 microseconds, you would enter a *value* of 200.

Step 3 (Optional) To view the delay value assigned to an interface, use the **show interface** command.

Monitoring EIGRP

You can use the following commands to monitor the EIGRP routing process. For examples and descriptions of the command output, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

- To display the EIGRP event log, enter the following command:

  ```
  hostname# show eigrp [as-number] events [{start end} | type]
  ```

- To display the interfaces participating in EIGRP routing, enter the following command:

  ```
  hostname# show eigrp [as-number] interfaces [if-name] [detail]
  ```

- To display the EIGRP neighbor table, enter the following command:

  ```
  hostname# show eigrp [as-number] neighbors [detail | static] [if-name]
  ```

- To display the EIGRP topology table, enter the following command:

  ```
  hostname# show eigrp [as-number] topology [ip-addr [mask] | active | all-links |
  pending | summary | zero-successors]
  ```

- To display EIGRP traffic statistics, enter the following command:

  ```
  hostname# show eigrp [as-number] traffic
  ```

Disabling Neighbor Change and Warning Message Logging

By default neighbor change, and neighbor warning messages are logged. You can disable the logging of neighbor change message and neighbor warning messages.

- To disable the logging of neighbor change messages, enter the following command in router configuration mode for the EIGRP routing process:

```
hostname(config-router)# no eigrp log-neighbor-changes
```

- To disable the logging of neighbor warning messages, enter the following command in router configuration mode for the EIGRP routing process:

```
hostname(config-router)# no eigrp log-neighbor-warnings
```

Configuring Asymmetric Routing Support

In some situations, return traffic for a session may be routed through a different interface than it originated from. In failover configurations, return traffic for a connection that originated on one unit may return through the peer unit. This most commonly occurs when two interfaces on a single FWSM, or two FWSMs in a failover pair, are connected to different service providers and the outbound connection does not use a NAT address. By default, the FWSM drops the return traffic because there is no connection information for the traffic.

You can prevent the return traffic from being dropped using the **asr-group** command on interfaces where this is likely to occur. When an interface configured with the **asr-group** command receives a packet for which it has no session information, it checks the session information for the other interfaces that are in the same group.

Note In failover configurations, you must enable Stateful Failover for session information to be passed from the standby unit or failover group to the active unit or failover group.

If it does not find a match, the packet is dropped. If it finds a match, then one of the following actions occurs:

- If the incoming traffic originated on a peer unit in a failover configuration, some or all of the layer 2 header is rewritten and the packet is redirected to the other unit. This redirection continues as long as the session is active.
- If the incoming traffic originated on a different interface on the same unit, some or all of the layer 2 header is rewritten and the packet is re-injected into the stream.

This section contains the following topics:

Adding Interfaces to ASR Groups

Enter the following commands to add an interface to an asymmetric routing group. Stateful Failover must be enabled for asymmetric routing support to function properly between units in failover configurations.

```
hostname/ctx1(config)# interface if
hostname/ctx1(config-if)# asr-group num
```

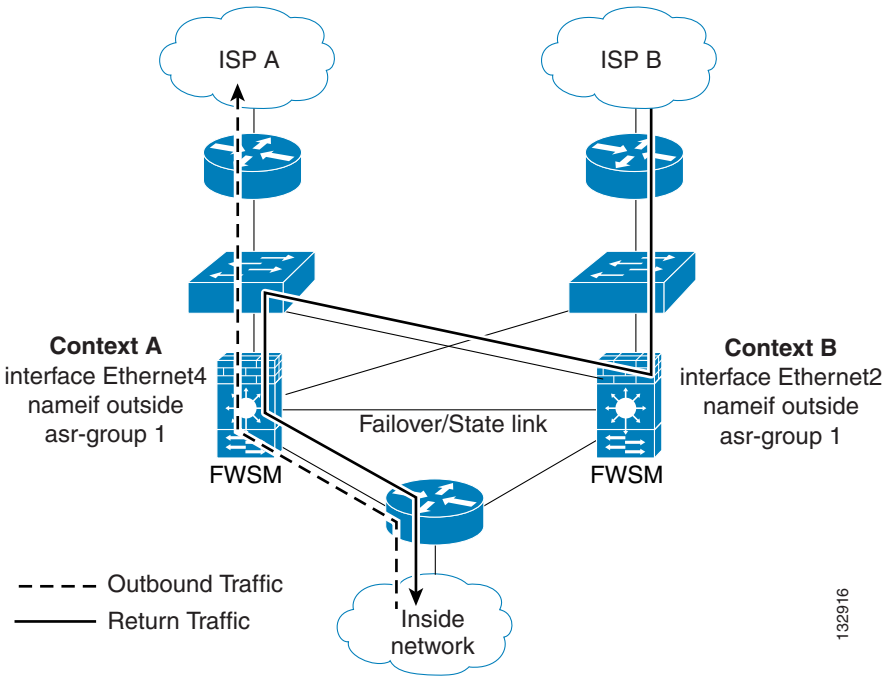
Valid values for *num* range from 1 to 32. You need to enter the command for each interface that will participate in the ASR group. You can create up to 32 ASR groups and assign a maximum of 8 interfaces to each group.

Note The upstream and downstream routers must use one MAC address per VLAN and have different MAC addresses for different VLANs to allow for the redirection of packets from a standby unit to an active unit in failover configurations.

Asymmetric Routing Support Example

Figure 8-1 shows an example of using the **asr-group** command for asymmetric routing support in an Active/Active failover configuration.

Figure 8-1 ASR Example with Active/Active Failover



Context A is active on one unit and context B is active on the other. Each context has an interface named "outside", both of which are configured as part of **asr-group** 1. The outbound traffic is routed through the unit where context A is active. However, the return traffic is being routed through the unit where context B is active. Normally, the return traffic would be dropped because there is no session information

for the traffic on the unit. However, because the interface is configured with an **asr-group** number, the unit looks at the session information for any other interfaces with the same **asr-group** assigned to it. It finds the session information in the outside interface for context A, which is in the standby state on the unit, and forwards the return traffic to the unit where context A is active.

The traffic is forwarded though the outside interface of context A on the unit where context A is in the standby state and returns through the outside interface of context A on the unit where context A is in the active state. This forwarding continues as needed until the session ends.

Configuring Route Health Injection

Note This feature depends on Cisco IOS Release 12.2(33)SXI or later, and is only available on the Catalyst 6500 switch.

Route Health Injection, or RHI, is used for injecting the connected routes, static routes, and NAT addresses configured on the FWSM into the MSFC routing table. In multiple context mode, this feature is especially valuable because of the lack of dynamic routing protocol support. The MSFC can then redistribute the route to other routing tables.

This section includes information on the following topics:

Route Health Injection Overview

For connected routes, static routes, and NAT addresses, the FWSM can inject routes into the routing table of the switch; these routes specify the IP address of the FWSM interface as the next hop IP address for each of these FWSM networks.

For example, when you configure NAT on the FWSM, the MSFC and other external routers do not know that those NAT addresses are connected to the FWSM unless you configure static routes on the MSFC to point to the FWSM interface. But by utilizing RHI, you can inject the NAT addresses to point to the FWSM interface so the MSFC can automatically forward that traffic to the FWSM.

Because the FWSM only supports OSPF or other dynamic routing protocols in single context mode, RHI can be used in multiple mode to inject routes to the MSFC, which can then redistribute these routes through OSPF or other dynamic routing protocols. This allows the FWSM to redistribute FWSM routes through OSPF or other dynamic routing protocols even when running in multiple mode, by utilizing the MSFC routing protocols and RHI.

In a failover scenario, RHI routes are injected from only Active FWSM (applicable in both Active/Standby and Active/Active scenario). If you have FWSM failover between two chassis in Active/Active failover mode, both of the FWSM networks inject routes to their corresponding MSFC, corresponding to the contexts that is in the Active state.

Additionally, if you have HSRP configured between two MSFCs on other interfaces which receive traffic targeted towards either of the two FWSMs, you must choose a routing protocol configured between the two MSFCs. This ensures that each MFSC knows the routes that can be reached through the other FWSM that is not in the same chassis. If there is no exchange of routing information between the two MSFCs, information will not be received and the system will not respond due to the fact that the HSRP Active

MFSC may receive a packet targeted towards a network that can be reached thru FWSM in the other chassis. In that case, the HSRP Active MSFC did not learn of this route from the other MSFC, it may drop the packet (or) incorrectly forwards it to it's default gateway.

The FWSM injects routes into the MSFC using SCP messages.

RHI Guidelines

- RHI is supported in both single and multiple context mode.
- RHI is supported in routed firewall mode; it is not supported in transparent mode.
- RHI is supported with failover (Active/Standby and Active/Active).
- The FWSM interface that you specify as the next hop interface must be an SVI between the FWSM and the MSFC. See the "Adding Switched Virtual Interfaces to the MSFC" section on page 2-4.

Enabling RHI

To configure RHI, perform the following steps:

Step 1 (Optional) If you want to limit the routes that you inject for each type (connected, static, and NAT), you can limit the routes you want to inject to those that match one of the following objects:

- **route-map**—See the "Defining a Route Map" section on page 8-5. Route maps are only available in single context mode.
- **access-list standard**—See the "Adding a Standard Access List" section on page 12-11.
- (NAT only) **global**—See the "Configuring Dynamic NAT or PAT" section on page 15-25.

Step 2 Enable RHI by entering the following command:

```
hostname(config)# route-inject
```

The CLI enters route-inject configuration mode. You can only configure one **route-inject** command.

Step 3 To inject NAT address routes, enter the following command:

```
hostname(config-route-inject)# redistribute nat [route-map map_name | access-list acl_id |
global-pool pool_id] interface interface_name
```

where the **interface** *interface_name* argument specifies the FWSM interface; this interface IP address is used as the next-hop IP address in the routes that are injected.

By default, all mapped addresses that you define in **static** and **global** commands are injected.

If you want to limit the NAT addresses injected, you can specify the **route-map**, **access-list**, or **global-pool** argument; only matching addresses are injected. For the **global-pool** argument, make sure the **global** command NAT ID that you specify is on the same interface as the **redistribute** command. If you use the same NAT ID for multiple **global** commands on multiple interfaces, only those commands on the matching interface as the **redistribute** command are used.

You can enter only one **redistribute nat** command.

Step 4 To inject connected routes, enter the following command:

```
hostname(config-route-inject)# redistribute connected [route-map map_name |
access-list acl_id] interface interface_name
```

where the **interface** *interface_name* argument specifies the FWSM interface; this interface IP address is used as the next-hop IP address in the routes that are injected.

By default, all connected routes are injected.

If you want to limit the routes injected, you can specify the **route-map** or **access-list** argument; only matching addresses are injected.

You can enter only one **redistribute connected** command.

Step 5 To inject static routes, enter the following command:

```
hostname(config-route-inject)# redistribute static [route-map map_name |
access-list acl_id] interface interface_name
```

where the **interface** *interface_name* argument specifies the FWSM interface; this interface IP address is used as the next-hop IP address in the routes that are injected.

By default, all static routes are injected.

If you want to limit the routes injected, you can specify the **route-map** or **access-list** argument; only matching addresses are injected.

You can enter only one **redistribute static** command.

The following example injects NAT addresses that match access list **acl1**; 209.165.201.0/30 is injected with a nexthop of 209.165.200.225 (the active IP address of the outside interface) on VLAN 20. The 209.165.201.10 through .16 addresses are not injected.

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 209.165.200.225 255.255.255.224 standby 209.165.200.226
hostname(config-if)# exit
hostname(config)# access-list acl1 standard permit 209.165.201.0 255.255.255.252
hostname(config)# global (outside) 10 209.165.201.1-209.165.201.2 netmask 255.255.255.0
hostname(config)# global (outside) 10 209.165.201.10-209.165.201.16 netmask 255.255.255.0
hostname(config)# route-inject
hostname(config-route-inject)# redistribute nat access-list acl1 interface outside
```

The following example injects 209.165.202.129 through .131 and 209.165.202.140 through .146 with a nexthop 209.165.200.250 on VLAN 20. The global pools on the dmz interface, and the global pool 20 on the outside interface are not included.

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 209.165.200.250 255.255.255.224 standby 209.165.200.251
hostname(config-if)# exit
hostname(config)# global (dmz) 10 209.165.201.1-209.165.201.10 netmask 255.255.255.0
hostname(config)# global (outside) 10 209.165.202.129-209.165.202.131 netmask
255.255.255.0
hostname(config)# global (outside) 10 209.165.202.140-209.165.202.146 netmask
255.255.255.0
hostname(config)# global (outside) 20 209.165.202.150-209.165.202.155 netmask
255.255.255.0
hostname(config)# route-inject
hostname(config-route-inject)# redistribute nat global-pool 10 interface outside
```

The following example injects 209.165.201.0/27 and 192.0.2.0/24 with a nexthop of 209.165.200.225 on VLAN 20. 209.165.202.128/27 is not injected.

```
hostname(config)# interface vlan20
hostname(config-if)# nameif outside
hostname(config-if)# ip address 209.165.200.225 255.255.255.224 standby 209.165.200.226
```

```
hostname(config-if)# exit
hostname(config)# access-list acl1 standard permit 209.165.201.0 255.255.255.224
hostname(config)# access-list acl2 standard permit 192.0.2.0 255.255.255.0
hostname(config)# route-map map1 permit 10
hostname(config-route-map)# match ip address acl1 acl2
hostname(config-route-map)# exit
hostname(config)# route inside 209.165.201.0 255.255.255.224 10.1.1.1
hostname(config)# route inside 192.0.2.0 255.255.255.0 10.1.1.1
hostname(config)# route inside 209.165.202.128 255.255.255.224 10.1.1.1
hostname(config)# route-inject
hostname(config-route-inject)# redistribute static route-map map1 interface outside
```

Configuring DHCP

DHCP provides network configuration parameters, such as IP addresses, to DHCP clients. The FWSM can provide a DHCP server or DHCP relay services to DHCP clients attached to FWSM interfaces. The DHCP server provides network configuration parameters directly to DHCP clients. DHCP relay passes DHCP requests received on one interface to an external DHCP server located behind a different interface.

This section includes the following topics:

Configuring a DHCP Server

This section describes how to configure DHCP server provided by the FWSM. This section includes the following topics:

Enabling the DHCP Server

The FWSM can act as a DHCP server. DHCP is a protocol that supplies network settings to hosts including the host IP address, the default gateway, and a DNS server.

Note The FWSM DHCP server does not support BOOTP requests.

In multiple context mode, you cannot enable the DHCP server or DHCP relay on an interface that is used by more than one context.

You can configure a DHCP server on each interface of the FWSM. Each interface can have its own pool of addresses to draw from. However the other DHCP settings, such as DNS servers, domain name, options, ping timeout, and WINS servers, are configured globally and used by the DHCP server on all interfaces.

You cannot configure a DHCP client or DHCP Relay services on an interface on which the server is enabled. Additionally, DHCP clients must be directly connected to the interface on which the server is enabled.

To enable the DHCP server on a given FWSM interface, perform the following steps:

Step 1 Create a DHCP address pool. Enter the following command to define the address pool:

```
hostname(config)# dhcpd address ip_address-ip_address interface_name
```

The FWSM assigns a client one of the addresses from this pool to use for a given length of time. These addresses are the local, untranslated addresses for the directly connected network.

The address pool must be on the same subnet as the FWSM interface.

Step 2 (Optional) To specify the IP address(es) of the DNS server(s) the client will use, enter the following command:

```
hostname(config)# dhcpd dns dns1 [dns2]
```

You can specify up to two DNS servers.

Step 3 (Optional) To specify the IP address(es) of the WINS server(s) the client will use, enter the following command:

```
hostname(config)# dhcpd wins wins1 [wins2]
```

You can specify up to two WINS servers.

Step 4 (Optional) To change the lease length to be granted to the client, enter the following command:

```
hostname(config)# dhcpd lease lease_length
```

This lease equals the amount of time (in seconds) the client can use its allocated IP address before the lease expires. Enter a value between 0 to 1,048,575. The default value is 3600 seconds.

Step 5 (Optional) To configure the domain name the client uses, enter the following command:

```
hostname(config)# dhcpd domain domain_name
```

Step 6 (Optional) To configure the DHCP ping timeout value, enter the following command:

```
hostname(config)# dhcpd ping_timeout milliseconds
```

To avoid address conflicts, the FWSM sends two ICMP ping packets to an address before assigning that address to a DHCP client. This command specifies the timeout value for those packets.

Step 7 (Transparent Firewall Mode) Define a default gateway. To define the default gateway that is sent to DHCP clients, enter the following command:

```
hostname(config)# dhcpd option 3 ip gateway_ip
```

If you do not use the DHCP option 3 to define the default gateway, DHCP clients use the IP address of the management interface. The management interface does not route traffic.

Step 8 To enable the DHCP daemon within the FWSM to listen for DHCP client requests on the enabled interface, enter the following command:

```
hostname(config)# dhcpd enable interface_name
```

For example, to assign the range 10.0.1.101 to 10.0.1.110 to hosts connected to the inside interface, enter the following commands:

```
hostname(config)# dhcpd address 10.0.1.101-10.0.1.110 inside
hostname(config)# dhcpd dns 209.165.201.2 209.165.202.129
hostname(config)# dhcpd wins 209.165.201.5
hostname(config)# dhcpd lease 3000
hostname(config)# dhcpd domain example.com
hostname(config)# dhcpd enable inside
```

Configuring DHCP Options

You can configure the FWSM to send information for the DHCP options listed in RFC 2132. The DHCP options fall into one of three categories:

- Options that return an IP address.
- Options that return a text string.
- Options that return a hexadecimal value.

The FWSM supports all three categories of DHCP options. To configure a DHCP option, do one of the following:

- To configure a DHCP option that returns one or two IP addresses, enter the following command:

  ```
  hostname(config)# dhcpd option code ip addr_1 [addr_2]
  ```

- To configure a DHCP option that returns a text string, enter the following command:

  ```
  hostname(config)# dhcpd option code ascii text
  ```

- To configure a DHCP option that returns a hexadecimal value, enter the following command:

  ```
  hostname(config)# dhcpd option code hex value
  ```

Note The FWSM does not verify that the option type and value that you provide match the expected type and value for the option code as defined in RFC 2132. For example, you can enter **dhcpd option 46 ascii hello**, and the FWSM accepts the configuration although option 46 is defined in RFC 2132 as expecting a single-digit, hexadecimal value. For more information about the option codes and their associated types and expected values, refer to RFC 2132.

Table 8-1 shows the DHCP options that are not supported by the **dhcpd option** command:

Table 8-1 Unsupported DHCP Options

Option Code	Description
0	DHCPOPT_PAD
1	HCPOPT_SUBNET_MASK
12	DHCPOPT_HOST_NAME
50	DHCPOPT_REQUESTED_ADDRESS
51	DHCPOPT_LEASE_TIME
52	DHCPOPT_OPTION_OVERLOAD
53	DHCPOPT_MESSAGE_TYPE
54	DHCPOPT_SERVER_IDENTIFIER
58	DHCPOPT_RENEWAL_TIME

Table 8-1 Unsupported DHCP Options

Option Code	Description
59	DHCPOPT_REBINDING_TIME
61	DHCPOPT_CLIENT_IDENTIFIER
67	DHCPOPT_BOOT_FILE_NAME
82	DHCPOPT_RELAY_INFORMATION
255	DHCPOPT_END

Specific options, DHCP option 3, 66, and 150, are used to configure Cisco IP Phones. See the "Using Cisco IP Phones with a DHCP Server" section on page 8-38 topic for more information about configuring those options.

Using Cisco IP Phones with a DHCP Server

Enterprises with small branch offices that implement a Cisco IP Telephony Voice over IP solution typically implement Cisco CallManager at a central office to control Cisco IP Phones at small branch offices. This implementation allows centralized call processing, reduces the equipment required, and eliminates the administration of additional Cisco CallManager and other servers at branch offices.

Cisco IP Phones download their configuration from a TFTP server. When a Cisco IP Phone starts, if it does not have both the IP address and TFTP server IP address preconfigured, it sends a request with option 150 or 66 to the DHCP server to obtain this information.

- DHCP option 150 provides the IP addresses of a list of TFTP servers.
- DHCP option 66 gives the IP address or the hostname of a single TFTP server.

Cisco IP Phones might also include DHCP option 3 in their requests, which sets the default route.

Cisco IP Phones might include both option 150 and 66 in a single request. In this case, the FWSM DHCP server provides values for both options in the response if they are configured on the FWSM.

You can configure the FWSM to send information for most options listed in RFC 2132. The following table shows the syntax for any option number, as well as the syntax for commonly-used options 66,150, and 3:

- To provide information for DHCP requests that include an option number as specified in RFC 2132, enter the following command:

  ```
  hostname(config)# dhcpd option number value
  ```

- To provide the IP address or name of a TFTP server for option 66, enter the following command:

  ```
  hostname(config)# dhcpd option 66 ascii server_name
  ```

- To provide the IP address or names of one or two TFTP servers for option 150, enter the following command:

  ```
  hostname(config)# dhcpd option 150 ip server_ip1 [server_ip2]
  ```

  The *server_ip1* is the IP address or name of the primary TFTP server while *server_ip2* is the IP address or name of the secondary TFTP server. A maximum of two TFTP servers can be identified using option 150.

- To provide set the default route, enter the following command:

  ```
  hostname(config)# dhcpd option 3 ip router_ip1
  ```

Configuring DHCP Relay Services

This section describes how to configure DHCP relay services provided by the FWSM. This section includes the following topics:

DHCP Relay Overview

You can configure a DHCP relay agent to forward DHCP requests received on an interface to one or more DHCP servers. When a DHCP request enters an interface, the DHCP servers to which the FWSM relays the request depends on your configuration. You can configure the following types of servers:

- Interface-specific DHCP servers—When a request enters a particular interface, then the FWSM relays the request only to the interface-specific servers.
- Global DHCP servers—When a request enters an interface that does not have interface-specific servers configured, then the FWSM relays the request to all global servers. If the interface has interface-specific servers, then the global servers are not used.

The following restrictions apply to the use of the DHCP relay agent:

- The relay agent cannot be enabled if the DHCP server feature is also enabled.
- DHCP Relay services are not available in transparent firewall mode. You can, however, allow DHCP traffic through using an access list. To allow DHCP requests and replies through the FWSM in transparent mode, you need to configure two access lists, one that allows DCHP requests from the inside interface to the outside, and one that allows the replies from the server in the other direction.
- Clients must be directly-connected to the FWSM and cannot send requests through another relay agent or a router.
- For multiple context mode, you cannot enable DHCP relay on an interface that is used by more than one context.

Configuring the DHCP Relay Agent

To enable DHCP relay, perform the following steps:

Step 1 Set the IP addresses of DHCP servers using one or both of the following methods:

- To configure an interface-specific server, enter the following commands:

```
hostname(config)# interface {vlan vlan_id | mapped_name}
hostname(config-if)# dhcprelay server ip_address
```

Where the **vlan** *vlan_id* or *mapped_interface* argument is the interface on which you want to enable DHCP relay.

You can enter the **dhcprelay server** command up to 4 times per interface, with a maximum of 10 servers allowed (including global servers) per context or in single mode.

The interface-specific servers take precedence over any global servers configured.

The DHCP servers cannot reside on the same interface on which you enable DHCP relay. (The FWSM determines which interface is connected to the DHCP server by using the routing table.)

Note If you configure an interface-specific server address after a connection has already been set up between a client and an existing global DHCP server, the client keeps using the global server until the server address lease expires. After the lease expires, new connections use the interface-specific server.

- To configure a global server, enter the following command:

  ```
  hostname(config)# dhcprelay server ip_address if_name
  ```

  Where the *if_name* argument is the interface connected to the DHCP server. The DHCP server must reside on a different interface from the DHCP clients where you enable DHCP relay.

  You can use this command up to 10 times to identify up to 10 servers, including any interface-specific servers.

Step 2 To enable DHCP relay on the interface connected to the clients, enter the following command:

```
hostname(config)# dhcprelay enable interface
```

You can enable DHCP relay on multiple interfaces; however, you cannot configure DHCP relay on any interfaces that are connected to the DHCP servers. For example, you can configure DHCP relay on inside1 and inside 2 interfaces, and configure DHCP servers on outside and dmz interfaces. You cannot configure any servers on inside1 or inside2.

Step 3 (Optional) To set the number of seconds allowed for relay address negotiation, enter the following command:

```
hostname(config)# dhcprelay timeout seconds
```

Step 4 (Optional) To change the first default router address in the packet sent from the DHCP server to the address of the FWSM interface, enter the following command:

```
hostname(config)# dhcprelay setroute if_name
```

This action allows the client to set its default route to point to the FWSM even if the DHCP server specifies a different router.

If there is no default router option in the packet, the FWSM adds one containing the interface address.

The following example enables the FWSM to forward DHCP requests from clients connected to the inside1 and inside2 interfaces to a global DHCP server on the outside interface and a global DHCP server on the DMZ interface:

```
hostname(config)# dhcprelay server 209.165.200.225 outside
hostname(config)# dhcprelay server 209.165.201.4 dmz
hostname(config)# dhcprelay enable inside1
hostname(config)# dhcprelay setroute inside1
hostname(config)# dhcprelay enable inside2
hostname(config)# dhcprelay setroute inside2
```

The following example enables the FWSM to forward DHCP requests from clients connected to the inside1 interface (on vlan 20) to an interface-specific DHCP server (on the outside interface). The inside2 interface uses the global DHCP servers on the outside and DMZ interfaces. Note that the global DHCP server on the outside interface is the same as the interface-specific server for inside1.

```
hostname(config)# interface vlan 20
hostname(config-if)# dhcprelay server 209.165.200.225
hostname(config)# dhcprelay server 209.165.200.225 outside
hostname(config)# dhcprelay server 209.165.201.4 dmz
hostname(config)# dhcprelay enable inside1
hostname(config)# dhcprelay setroute inside1
hostname(config)# dhcprelay enable inside2
hostname(config)# dhcprelay setroute inside2
```

Preserving DHCP Option 82

This section describes the DHCP option 82 feature. This feature enables the DHCP relay agent to include information about itself and the attached client when forwarding DHCP requests from a DHCP client to a DHCP server.

If the DHCP relay agent receives a DHCP packet from untrusted sources with option 82 already set, but the giaddr (the DHCP relay agent address that will be set by the relay agent before it forwards the packet to the server) field set to zero, or when it is behind DSLAM devices, then it will drop that packet by default. You can optionally preserve option 82 and forward the packet by identifying an interface as a trusted interface. This feature makes sure that DHCP snooping and IP source guard features on the switch work along with the FWSM.

DHCP snooping is a DHCP security feature that provides network security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding table.

You can enable this feature on interfaces configured with IPv4 and IPv6 addresses.

To configure a particular interface as a trusted interface that preserves option 82, enter the following commands:

```
hostname(config)# interface interface {vlan vlan_id | mapped_name}
hostname(config-if)# dhcprelay information trusted
```

To configure all interfaces as trusted interfaces, enter the following command:

```
hostname(config)# dhcprelay information trust-all
```

All the interfaces will be set as trusted interfaces except interfaces which are shared and the interface on which the DHCP server is configured.

The interface-specific trusted configuration and global trusted configuration can exist together. For example there are three interfaces A, B and C, and a user configures interface A as trusted using the interface-specific command.Then the user configures the global command also.

Now all the three interfaces A, B, and C are trusted interfaces. If you enter the **no dhcprelay information trust-all** command, then interfaces B and C will become non-trusted interfaces. Interface A will continue to be a trusted interface, since the interface-specific trusted configuration is not removed.

Verifying the DHCP Relay Configuration

To view the interface-specific DHCP relay configuration, enter the following command:

```
hostname# show running-config dhcprelay interface [vlan vlan_id | mapped_name]
```

To view the global DHCP relay configuration, enter the following command:

```
hostname# show running-config dhcprelay global
```

CHAPTER 9

Configuring Multicast Routing

This chapter describes how to configure multicast routing. This chapter includes the following sections:

Multicast Routing Overview

The FWSM supports both Stub Multicast Routing and PIM multicast routing. However, you cannot configure both concurrently on a single FWSM.

Note Only the UDP transport layer is supported for multicast routing.

Stub multicast routing provides dynamic host registration and facilitates multicast routing. When configured for Stub Multicast Routing, the FWSM acts as an IGMP proxy agent. Instead of fully participating in multicast routing, the FWSM forwards IGMP messages to an upstream multicast router, which sets up delivery of the multicast data. When configured for Stub Multicast Routing, the FWSM cannot be configured for PIM.

The FWSM supports both PIM-SM and bi-directional PIM. PIM-SM is a multicast routing protocol that uses the underlying unicast routing information base or a separate multicast-capable routing information base. It builds unidirectional shared trees rooted at a single Rendezvous Point per multicast group and optionally creates shortest-path trees per multicast source.

Bi-directional PIM is a variant of PIM-SM that builds bi-directional shared trees connecting multicast sources and receivers. Bi-directional trees are built using a DF election process operating on each link of the multicast topology. With the assistance of the DF, multicast data is forwarded from sources to the Rendezvous Point, and therefore along the shared tree to receivers, without requiring source-specific state. The DF election takes place during Rendezvous Point discovery and provides a default route to the Rendezvous Point.

Note If the FWSM is the PIM RP, use the untranslated outside address of the FWSM as the RP address.

Enabling Multicast Routing

Enabling multicast routing lets the FWSM forward multicast packets. Enabling multicast routing automatically enables PIM and IGMP on all interfaces. To enable multicast routing, enter the following command.

```
hostname(config)# multicast-routing
```

Note Multicast routing on the FWSM is limited to eight outgoing interfaces.

The number of entries in the multicast routing tables are limited by the amount of RAM on the system. Table 9-1 lists the maximum number of entries for specific multicast tables based on the amount of RAM on the FWSM. Once these limits are reached, any new entries are discarded.

Table 9-1 Entry Limits for Multicast Tables

Table	16 MB	128 MB	128+ MB
MFIB	1000	3000	5000
IGMP Groups	1000	3000	5000
PIM Routes	3000	7000	12000

Configuring IGMP Features

IP hosts use IGMP to report their group memberships to directly connected multicast routers. IGMP uses group addresses (Class D IP address) as group identifiers. Host group address can be in the range 224.0.0.0 to 239.255.255.255. The address 224.0.0.0 is never assigned to any group. The address 224.0.0.1 is assigned to all systems on a subnet. The address 224.0.0.2 is assigned to all routers on a subnet.

When you enable multicast routing on the FWSM, IGMP Version 2 is automatically enabled on all interfaces.

Note Only the **no igmp** command appears in the interface configuration when you use the **show run** command. If the **multicast-routing** command appears in the device configuration, then IGMP is automatically enabled on all interfaces.

This section describes how to configure optional IGMP setting on a per-interface basis. This section includes the following topics:

- Disabling IGMP on an Interface, page 9-3
- Configuring Group Membership, page 9-3
- Configuring a Statically Joined Group, page 9-3

Disabling IGMP on an Interface

You can disable IGMP on specific interfaces. This is useful if you know that you do not have any multicast hosts on a specific interface and you want to prevent the FWSM from sending host query messages on that interface.

To disable IGMP on an interface, enter the following command:

```
hostname(config-if)# no igmp
```

To reenable IGMP on an interface, enter the following command:

```
hostname(config-if)# igmp
```

Note Only the **no igmp** command appears in the interface configuration.

Configuring Group Membership

You can configure the FWSM to be a member of a multicast group. Configuring the FWSM to join a multicast group causes upstream routers to maintain multicast routing table information for that group and keep the paths for that group active.

To have the FWSM join a multicast group, enter the following command:

```
hostname(config-if)# igmp join-group group-address
```

Configuring a Statically Joined Group

Sometimes a group member cannot report its membership in the group, or there may be no members of a group on the network segment, but you still want multicast traffic for that group to be sent to that network segment. You can have multicast traffic for that group sent to the segment in one of two ways:

- Using the **igmp join-group** command (see Configuring Group Membership, page 9-3). This causes the FWSM to accept and to forward the multicast packets.
- Using the **igmp static-group** command. The FWSM does not accept the multicast packets but rather forwards them to the specified interface.

To configure a statically joined multicast group on an interface, enter the following command:

```
hostname(config-if)# igmp static-group group-address
```

Controlling Access to Multicast Groups

To control the multicast groups that hosts on the FWSM interface can join, perform the following steps:

Step 1 Create an access list for the multicast traffic. You can create more than one entry for a single access list. You can use extended or standard access lists.

- To create a standard access list, enter the following command:

  ```
  hostname(config)# access-list name standard [permit | deny] ip_addr mask
  ```

  The *ip_addr* argument is the IP address of the multicast group being permitted or denied.

- To create an extended access list, enter the following command:

  ```
  hostname(config)# access-list name extended [permit | deny] protocol src_ip_addr
  src_mask dst_ip_addr dst_mask
  ```

  The *dst_ip_addr* argument is the IP address of the multicast group being permitted or denied.

Step 2 Apply the access list to an interface by entering the following command:

```
hostname(config-if)# igmp access-group acl
```

The *acl* argument is the name of a standard or extended IP access list.

Limiting the Number of IGMP States on an Interface

You can limit the number of IGMP states resulting from IGMP membership reports on a per-interface basis. Membership reports exceeding the configured limits are not entered in the IGMP cache and traffic for the excess membership reports is not forwarded.

To limit the number of IGMP states on an interface, enter the following command:

```
hostname(config-if)# igmp limit number
```

Valid values range from 0 to 500, with 500 being the default value. Setting this value to 0 prevents learned groups from being added, but manually defined memberships (using the **igmp join-group** and **igmp static-group** commands) are still permitted. The **no** form of this command restores the default value.

Modifying the Query Interval and Query Timeout

The FWSM sends query messages to discover which multicast groups have members on the networks attached to the interfaces. Members respond with IGMP report messages indicating that they want to receive multicast packets for specific groups. Query messages are addressed to the all-systems multicast group, which has an address of 224.0.0.1, with a time-to-live value of 1.

These messages are sent periodically to refresh the membership information stored on the FWSM. If the FWSM discovers that there are no local members of a multicast group still attached to an interface, it stops forwarding multicast packet for that group to the attached network and it sends a prune message back to the source of the packets.

By default, the PIM designated router on the subnet is responsible for sending the query messages. By default, they are sent once every 125 seconds. To change this interval, enter the following command:

```
hostname(config-if)# igmp query-interval seconds
```

If the FWSM does not hear a query message on an interface for the specified timeout value (by default, 255 seconds), then the FWSM becomes the designated router and starts sending the query messages. To change this timeout value, enter the following command:

```
hostname(config-if)# igmp query-timeout seconds
```

Note The **igmp query-timeout** and **igmp query-interval** commands require IGMP Version 2.

Changing the Query Response Time

By default, the maximum query response time advertised in IGMP queries is 10 seconds. If the FWSM does not receive a response to a host query within this amount of time, it deletes the group.

To change the maximum query response time, enter the following command:

```
hostname(config-if)# igmp query-max-response-time seconds
```

Changing the IGMP Version

By default, the FWSM runs IGMP Version 2, which enables several additional features such as the **igmp query-timeout** and **igmp query-interval** commands.

All multicast routers on a subnet must support the same version of IGMP. The FWSM does not automatically detect version 1 routers and switch to version 1. However, a mix of IGMP Version 1 and 2 hosts on the subnet works; the FWSM running IGMP Version 2 works correctly when IGMP Version 1 hosts are present.

To control which version of IGMP is running on an interface, enter the following command:

```
hostname(config-if)# igmp version {1 | 2}
```

Configuring Stub Multicast Routing

An FWSM acting as the gateway to the stub area does not need to participate in PIM. Instead, you can configure it to act as an IGMP proxy agent and forward IGMP messages from hosts connected on one interface to an upstream multicast router on another. To configure the FWSM as an IGMP proxy agent, forward the host join and leave messages from the stub area interface to an upstream interface.

To forward the host join and leave messages, enter the following command from the interface attached to the stub area:

```
hostname(config-if)# igmp forward interface if_name
```

Note Stub Multicast Routing and PIM are not supported concurrently.

Configuring a Static Multicast Route

When using PIM, the FWSM expects to receive packets on the same interface where it sends unicast packets back to the source. In some cases, such as bypassing a route that does not support multicast routing, you may want unicast packets to take one path and multicast packets to take another.

Static multicast routes are not advertised or redistributed.

To configure a static multicast route for PIM, enter the following command:

```
hostname(config)# mroute src_ip src_mask {input_if_name | rpf_neighbor} [distance]
```

For example:

```
hostname(config)# mroute 10.1.1.1 255.255.255.255 192.168.1.2
```

where 10.1.1.1 is the server that is sending out the multicast traffic, and 192.168.1.2 is the RPF neighbor for FWSM.

Note You can specify the interface or the RPF neighbor, but not at the same time.

To configure a static multicast route for a stub area, enter the following command:

```
hostname(config)# mroute src_ip src_mask input_if_name [dense output_if_name] [distance]
```

Note The **dense** *output_if_name* keyword and argument pair is only supported for Stub Multicast Routing.

Configuring PIM Features

Routers use PIM to maintain forwarding tables for forwarding multicast diagrams. When you enable multicast routing on the FWSM, PIM and IGMP are automatically enabled on all interfaces.

Note PIM is not supported with PAT. The PIM protocol does not use ports and PAT only works with protocols that use ports.

This section describes how to configure optional PIM settings. This section includes the following topics:

Disabling PIM on an Interface

You can disable PIM on specific interfaces. To disable PIM on an interface, enter the following command:

```
hostname(config-if)# no pim
```

To reenable PIM on an interface, enter the following command:

```
hostname(config-if)# pim
```

Note Only the **no pim** command appears in the interface configuration.

Configuring a Static Rendezvous Point Address

All routers within a common PIM sparse mode or bidir domain require knowledge of the PIM RP address. The address is statically configured using the **pim rp-address** command.

Note The FWSM does not support Auto-RP or PIM BSR; you must use the **pim rp-address** command to specify the RP address.

You can configure the FWSM to serve as RP to more than one group. The group range specified in the access list determines the PIM RP group mapping. If an access list is not specified, then the RP for the group is applied to the entire multicast group range (224.0.0.0/4).

To configure the address of the PIM PR, enter the following command:

```
hostname(config)# pim rp-address ip_address [acl] [bidir]
```

The *ip_address* argument is the unicast IP address of the router to be a PIM RP. The *acl* argument is the name or number of an access list that defines which multicast groups the RP should be used with. Excluding the **bidir** keyword causes the groups to operate in PIM sparse mode.

Note The FWSM always advertises the bidir capability in the PIM hello messages regardless of the actual bidir configuration.

Configuring the Designated Router Priority

The DR is responsible for sending PIM register, join, and prune messaged to the RP. When there is more than one multicast router on a network segment, there is an election process to select the DR based on DR priority. If multiple devices have the same DR priority, then the device with the highest IP address becomes the DR.

By default, the FWSM has a DR priority of 1. You can change this value by entering the following command:

```
hostname(config-if)# pim dr-priority num
```

The *num* argument can be any number from 1 to 4294967294.

Filtering PIM Register Messages

You can configure the FWSM to filter PIM register messages. To filter PIM register messages, enter the following command:

```
hostname(config)# pim accept-register {list acl | route-map map-name}
```

Configuring PIM Message Intervals

Router query messages are used to elect the PIM DR. The PIM DR is responsible for sending router query messages. By default, router query messages are sent every 30 seconds. You can change this value by entering the following command:

```
hostname(config-if)# pim hello-interval seconds
```

Valid values for the *seconds* argument range from 1 to 3600 seconds.

Every 60 seconds, the FWSM sends PIM join/prune messages. To change this value, enter the following command:

```
hostname(config-if)# pim join-prune-interval seconds
```

Valid values for the *seconds* argument range from 10 to 600 seconds.

For More Information About Multicast Routing

The following RFCs from the IETF provide technical details about the IGMP and multicast routing standards used for implementing the SMR feature:

- RFC 2236 IGMPv2
- RFC 2362 PIM-SM
- RFC 2588 IP Multicast and Firewalls
- RFC 2113 IP Router Alert Option
- IETF draft-ietf-idmr-igmp-proxy-01.txt

CHAPTER 10

Configuring IPv6

This chapter describes how to enable and configure IPv6 on FWSM. IPv6 is available in routed firewall mode only.

This chapter includes the following sections:

For an example IPv6 configuration, see the "Example 4: IPv6 Configuration Example" section on page B-13.

IPv6-Enabled Commands

The following FWSM commands can accept and display IPv6 addresses:

- **capture**
- **configure**
- **copy**
- **http**
- **name**
- **object-group**
- **ping**
- **show conn**
- **show local-host**
- **show tcpstat**

- **ssh**
- **telnet**
- **tftp-server**
- **who**
- **write**

Note Failover does not support IPv6. The **ipv6 address** command does not support setting standby addresses for failover configurations. The **failover interface ip** command does not support using IPv6 addresses on the failover and Stateful Failover interfaces.

When entering IPv6 addresses in commands that support them, simply enter the IPv6 address using standard IPv6 notation, for example `ping fe80::2e0:b6ff:fe01:3b7a`. The FWSM correctly recognizes and processes the IPv6 address. However, you must enclose the IPv6 address in square brackets ([ ]) in the following situations:

- You need to specify a port number with the address, for example:

  ```
  [fe80::2e0:b6ff:fe01:3b7a]:8080
  ```

- The command uses a colon as a separator, such as the **write net** and **config net** commands. For example:

  ```
  configure net [fe80::2e0:b6ff:fe01:3b7a]:/tftp/config/pixconfig
  ```

The following commands were modified to work for IPv6:

- **debug**
- **fragment**
- **ip verify**
- **mtu**
- **icmp** (entered as **ipv6 icmp**)

The following inspection engines support IPv6:

- FTP
- HTTP
- ICMP
- SMTP
- TCP
- UDP

Configuring IPv6 on an Interface

At a minimum, each interface needs to be configured with an IPv6 link-local address. Additionally, you can add a site-local and global addresses to an interface.

Note FWSM does not support IPv6 anycast addresses.

You can configure both IPv6 and IPv4 addresses on an interface.

Note You cannot configure IPv6 on an interface that is used by more than one context (a shared VLAN).

To configure IPv6 on an interface, perform the following steps:

Step 1 Enter interface configuration mode for the interface for which you are configuring the IPv6 addresses:

```
hostname(config)# interface interface_name
```

Step 2 Configure an IPv6 address for the interface. You can assign several IPv6 addresses to an interface, such as an IPv6 link-local, site-local, and global address. However, at a minimum, you must configure a link-local address.

There are several methods for configuring IPv6 addresses for an interface. Pick the method that suits your needs from the following:

- The simplest method is to enable stateless autoconfiguration on the interface. Enabling stateless autoconfiguration on the interface configures IPv6 addresses based on prefixes received in Router Advertisement messages. A link-local address, based on the Modified EUI-64 interface ID, is automatically generated for the interface when stateless autoconfiguration is enabled. To enable stateless autoconfiguration, enter the following command:

  ```
  hostname(config-if)# ipv6 address autoconfig
  ```

- If you only need to configure a link-local address on the interface and are not going to assign any other IPv6 addresses to the interface, you have the option of manually defining the link-local address or generating one based on the interface MAC address (Modified EUI-64 format).

  Enter the following command to manually specify the link-local address:

  ```
  hostname(config-if)# ipv6 address ipv6-address link-local
  ```

Enter the following command to enable IPv6 on the interface and automatically generate the link-local address using the Modified EUI-64 interface ID based on the interface MAC address:

```
hostname(config-if)# ipv6 enable
```

Note You do not need to use the **ipv6 enable** command if you enter any other **ipv6 address** commands on an interface; IPv6 support is automatically enabled as soon as you assign an IPv6 address to the interface.

- Assign a site-local or global address to the interface. When you assign a site-local or global address, a link-local address is automatically created. Enter the following command to add a global or site-local address to the interface. Use the optional **eui-64** keyword to use the Modified EUI-64 interface ID in the low order 64 bits of the address.

  ```
  hostname(config-if)# ipv6 address ipv6-address [eui-64]
  ```

Step 3 (Optional) Suppress Router Advertisement messages on an interface. By default, Router Advertisement messages are automatically sent in response to router solicitation messages. You may want to disable these messages on any interface for which you do not want FWSM to supply the IPv6 prefix (for example, the outside interface).

Enter the following command to suppress Router Advertisement messages on an interface:

```
hostname(config-if)# ipv6 nd suppress-ra
```

See the "Example 4: IPv6 Configuration Example" section on page B-13 for an example of IPv6 addresses applied to an interface.

Configuring a Dual IP Stack on an Interface

FWSM supports the configuration of both IPv6 and IPv4 on an interface. You do not need to enter any special commands to do so; simply enter the IPv4 configuration commands and IPv6 configuration commands as you normally would. Make sure you configure the default route for both IPv4 and IPv6.

Configuring IPv6 Duplicate Address Detection

During the stateless autoconfiguration process, duplicate address detection verifies the uniqueness of new unicast IPv6 addresses before the addresses are assigned to interfaces (the new addresses remain in a tentative state while duplicate address detection is performed). Duplicate address detection is performed first on the new link-local address. When the link-local address is verified as unique, then duplicate address detection is performed all the other IPv6 unicast addresses on the interface.

Duplicate address detection is suspended on interfaces that are administratively down. While an interface is administratively down, the unicast IPv6 addresses assigned to the interface are set to a pending state. An interface returning to an administratively up state restarts duplicate address detection for all of the unicast IPv6 addresses on the interface.

When a duplicate address is identified, the state of the address is set to DUPLICATE and the address is not used. If the duplicate address is the link-local address of the interface, the processing of IPv6 packets is disabled on the interface and an error message is issued. If the duplicate address is a global address of the interface, the address is not used and an error message is issued. However, all configuration commands associated with the duplicate address remain as configured while the state of the address is set to DUPLICATE.

If the link-local address for an interface changes, duplicate address detection is performed on the new link-local address and all of the other IPv6 address associated with the interface are regenerated (duplicate address detection is performed only on the new link-local address).

FWSM uses neighbor solicitation messages to perform duplicate address detection. By default, the number of times an interface performs duplicate address detection is 1.

To change the number of duplicate address detection attempts, enter the following command:

```
hostname(config-if)# ipv6 nd dad attempts value
```

The *value* argument can be any value from 0 to 600. Setting the *value* argument to 0 disables duplicate address detection on the interface.

When you configure an interface to send out more than one duplicate address detection attempt, you can also use the **ipv6 nd ns-interval** command to configure the interval at which the neighbor solicitation messages are sent out. By default, they are sent out once every 1000 milliseconds.

To change the neighbor solicitation message interval, enter the following command:

```
hostname(config-if)# ipv6 nd ns-interval value
```

The *value* argument can be from 1000 to 3600000 milliseconds.

Note Changing this value changes it for all neighbor solicitation messages sent out on the interface, not just those used for duplicate address detection.

Configuring IPv6 Default and Static Routes

IPv6 unicast routing is always enabled. FWSM routes IPv6 traffic between interfaces as long as the interfaces are enabled for IPv6 and the IPv6 access lists allow the traffic. You can add a default route and static routes using the **ipv6 route** command.

To configure an IPv6 default route and static routes, perform the following steps:

Step 1 To add the default route, use the following command:

```
hostname(config)# ipv6 route interface_name ::/0 next_hop_ipv6_addr
```

The address ::/0 is the IPv6 equivalent of "any."

Step 2 (Optional) Define IPv6 static routes. Use the following command to add an IPv6 static route to the IPv6 routing table:

```
hostname(config)# ipv6 route if_name destination next_hop_ipv6_addr [admin_distance]
```

Note The **ipv6 route** command works like the **route** command used to define IPv4 static routes.

See the "Example 4: IPv6 Configuration Example" section on page B-13 for an example of the **ipv6 route** command used to configure the default route.

Configuring IPv6 Access Lists

Configuring an IPv6 access list is similar configuring an IPv4 access, but with IPv6 addresses.

To configure an IPv6 access list, perform the following steps:

Step 1 Create an access entry. To create an access list, use the **ipv6 access-list** command to create entries for the access list. There are two main forms of this command to choose from, one for creating access list entries specifically for ICMP traffic, and one to create access list entries for all other types of IP traffic.

- To create an IPv6 access list entry specifically for ICMP traffic, enter the following command:

  ```
  hostname(config)# ipv6 access-list id [line num] {permit | deny} icmp source
  destination [icmp_type]
  ```

- To create an IPv6 access list entry, enter the following command:

  ```
  hostname(config)# ipv6 access-list id [line num] {permit | deny} protocol source
  [src_port] destination [dst_port]
  ```

The following describes the arguments for the **ipv6 access-list** command:

- *id—The name of the access list. Use the same id in each command when you are entering multiple entries for an access list.*
- **line** *num*—When adding an entry to an access list, you can specify the line number in the list where the entry should appear.
- **permit | deny**—Determines whether the specified traffic is blocked or allowed to pass.
- **icmp**—Indicates that the access list entry applies to ICMP traffic.
- *protocol*—Specifies the traffic being controlled by the access list entry. This can be the name (**ip**, **tcp**, or **udp**) or number (1-254) of an IP protocol. Alternatively, you can specify a protocol object group using **object-group** *grp_id*.
- *source and destination*—Specifies the source or destination of the traffic. The source or destination can be an IPv6 prefix, in the format *prefix/length*, to indicate a range of addresses, the keyword **any, to specify any address**, or a specific host designated by **host** `host_ipv6_addr`.
- *src_port and dst_port*—The source and destination port (or service) argument. Enter an operator (**lt** for less than, **gt** for greater than, **eq** for equal to, **neq** for not equal to, or **range** for an inclusive range) followed by a space and a port number (or two port numbers separated by a space for the **range** keyword).
- *icmp_type*—Specifies the ICMP message type being filtered by the access rule. The value can be a valid ICMP type number (from 0 to 155) or one of the ICMP type literals as shown in Appendix E, "Addresses, Protocols, and Ports". Alternatively, you can specify an ICMP object group using **object-group** *id*.

Step 2 To apply the access list to an interface, enter the following command:

```
hostname(config)# access-group access_list_name {in | out} interface if_name
```

See the "Example 4: IPv6 Configuration Example" section on page B-13 for an example IPv6 access list.

Configuring IPv6 Neighbor Discovery

The IPv6 neighbor discovery process uses ICMPv6 messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and keep track of neighboring routers.

This section contains the following topics:

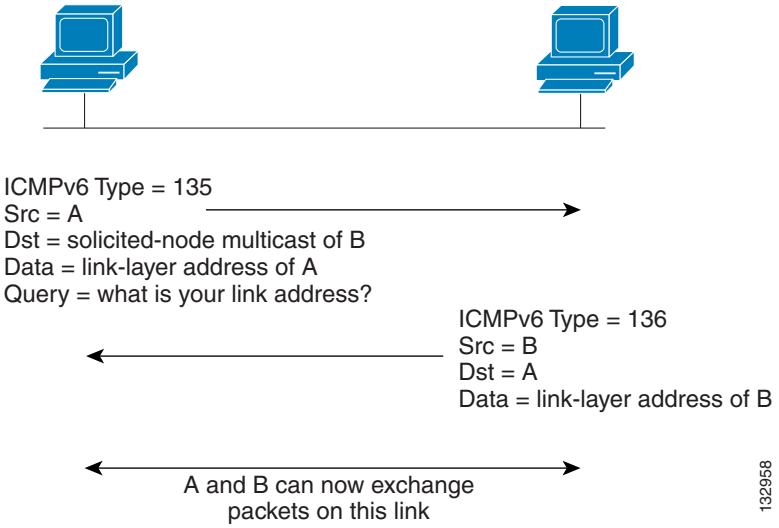
Configuring Neighbor Solicitation Messages

Neighbor solicitation messages (ICMPv6 Type 135) are sent on the local link by nodes attempting to discover the link-layer addresses of other nodes on the local link. The neighbor solicitation message is sent to the solicited-node multicast address.The source address in the neighbor solicitation message is the IPv6 address of the node sending the neighbor solicitation message. The neighbor solicitation message also includes the link-layer address of the source node.

After receiving a neighbor solicitation message, the destination node replies by sending a neighbor advertisement message (ICPMv6 Type 136) on the local link. The source address in the neighbor advertisement message is the IPv6 address of the node sending the neighbor advertisement message; the destination address is the IPv6 address of the node that sent the neighbor solicitation message. The data portion of the neighbor advertisement message includes the link-layer address of the node sending the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate. Figure 10-1 shows the neighbor solicitation and response process.

Figure 10-1 IPv6 Neighbor Discovery—Neighbor Solicitation Message



Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. When a node wants to verifying the reachability of a neighbor, the destination address in a neighbor solicitation message is the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is such a change, the destination address for the neighbor advertisement is the all-nodes multicast address.

You can configure the neighbor solicitation message interval and neighbor reachable time on a per-interface basis. See the following topics for more information:

- Configuring the Neighbor Solicitation Message Interval, page 10-7
- Configuring the Neighbor Reachable Time, page 10-8

Configuring the Neighbor Solicitation Message Interval

To configure the interval between IPv6 neighbor solicitation retransmissions on an interface, enter the following command:

```
hostname(config-if)# ipv6 nd ns-interval value
```

Valid values for the *value* argument range from 1000 to 3600000 milliseconds. The default value is 1000 milliseconds.

This setting is also sent in router advertisement messages.

Configuring the Neighbor Reachable Time

The neighbor reachable time enables detecting unavailable neighbors. Shorter configured times enable detecting unavailable neighbors more quickly; however, shorter times consume more IPv6 network bandwidth and processing resources in all IPv6 network devices. Very short configured times are not recommended in normal IPv6 operation.

To configure the amount of time that a remote IPv6 node is considered reachable after a reachability confirmation event has occurred, enter the following command:

```
hostname(config-if)# ipv6 nd reachable-time value
```

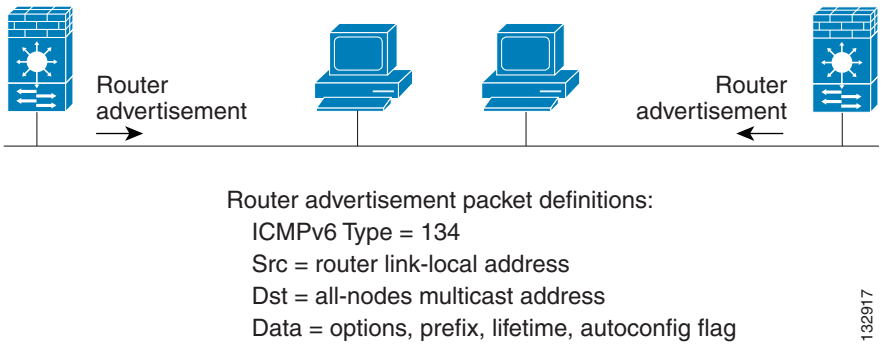
Valid values for the *value* argument range from 0 to 3600000 milliseconds. The default is 0.

This information is also sent in router advertisement messages.

Configuring Router Advertisement Messages

Router advertisement messages (ICMPv6 Type 134) are periodically sent out each IPv6 configured interface of FWSM. The router advertisement messages are sent to the all-nodes multicast address.

Figure 10-2 IPv6 Neighbor Discovery—Router Advertisement Message



Router advertisement messages typically include the following information:

- One or more IPv6 prefix that nodes on the local link can use to automatically configure their IPv6 addresses.
- Lifetime information for each prefix included in the advertisement.
- Sets of flags that indicate the type of autoconfiguration (stateless or stateful) that can be completed.
- Default router information (whether the router sending the advertisement should be used as a default router and, if so, the amount of time (in seconds) the router should be used as a default router).
- Additional information for hosts, such as the hop limit and MTU a host should use in packets that it originates.
- The amount of time between neighbor solicitation message retransmissions on a given link.
- The amount of time a node considers a neighbor reachable.

Router advertisements are also sent in response to router solicitation messages (ICMPv6 Type 133). Router solicitation messages are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled router advertisement message. Because router solicitation messages are usually sent by hosts at system startup, and the host does not have a

configured unicast address, the source address in router solicitation messages is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a configured unicast address, the unicast address of the interface sending the router solicitation message is used as the source address in the message. The destination address in router solicitation messages is the all-routers multicast address with a scope of the link. When a router advertisement is sent in response to a router solicitation, the destination address in the router advertisement message is the unicast address of the source of the router solicitation message.

You can configure the following settings for router advertisement messages:

- The time interval between periodic router advertisement messages.
- The router lifetime value, which indicates the amount of time IPv6 nodes should consider FWSM to be the default router.
- The IPv6 network prefixes in use on the link.
- Whether or not an interface transmits router advertisement messages.

Unless otherwise noted, the router advertisement message settings are specific to an interface and are entered in interface configuration mode. See the following topics for information about changing these settings:

- Configuring the Router Advertisement Transmission Interval, page 10-9
- Configuring the Router Lifetime Value, page 10-9
- Configuring the IPv6 Prefix, page 10-10
- Suppressing Router Advertisement Messages, page 10-10

Configuring the Router Advertisement Transmission Interval

By default, router advertisements are sent out every 200 seconds. To change the interval between router advertisement transmissions on an interface, enter the following command:

```
ipv6 nd ra-interval [msec] value
```

Valid values range from 3 to 1800 seconds (or 500 to 1800000 milliseconds if the **msec** keyword is used).

The interval between transmissions should be less than or equal to the IPv6 router advertisement lifetime if FWSM is configured as a default router by using the **ipv6 nd ra-lifetime** command. To prevent synchronization with other IPv6 nodes, randomly adjust the actual value used to within 20 percent of the desired value.

Configuring the Router Lifetime Value

The router lifetime value specifies how long nodes on the local link should consider FWSM as the default router on the link.

To configure the router lifetime value in IPv6 router advertisements on an interface, enter the following command:

```
hostname(config-if)# ipv6 nd ra-lifetime seconds
```

Valid values range from 0 to 9000 seconds. The default is 1800 seconds. Entering 0 indicates that FWSM should not be considered a default router on the selected interface.

Configuring the IPv6 Prefix

Stateless autoconfiguration uses IPv6 prefixes provided in router advertisement messages to create the global unicast address from the link-local address.

To configure which IPv6 prefixes are included in IPv6 router advertisements, enter the following command:

```
hostname(config-if)# ipv6 nd prefix ipv6-prefix/prefix-length
```

Note For stateless autoconfiguration to work properly, the advertised prefix length in router advertisement messages must always be 64 bits.

Suppressing Router Advertisement Messages

By default, Router Advertisement messages are automatically sent in response to router solicitation messages. You may want to disable these messages on any interface for which you do not want FWSM to supply the IPv6 prefix (for example, the outside interface).

To suppress IPv6 router advertisement transmissions on an interface, enter the following command:

```
hostname(config-if)# ipv6 nd suppress-ra
```

Configuring a Static IPv6 Neighbor

You can manually define a neighbor in the IPv6 neighbor cache. If an entry for the specified IPv6 address already exists in the neighbor discovery cache—learned through the IPv6 neighbor discovery process—the entry is automatically converted to a static entry. Static entries in the IPv6 neighbor discovery cache are not modified by the neighbor discovery process.

To configure a static entry in the IPv6 neighbor discovery cache, enter the following command:

```
hostname(config-if)# ipv6 neighbor ipv6_address if_name mac_address
```

The *ipv6_address* argument is the link-local IPv6 address of the neighbor, the *if_name* argument is the interface through which the neighbor is available, and the *mac_address* argument is the MAC address of the neighbor interface.

Note The **clear ipv6 neighbors** command does not remove static entries from the IPv6 neighbor discovery cache.

Verifying the IPv6 Configuration

This section describes how to verify your IPv6 configuration. You can use various show commands to verify your IPv6 settings.

This section includes the following topics:

Viewing IPv6 Interface Settings

To display the IPv6 interface settings, enter the following command:

```
hostname# show ipv6 interface [if_name]
```

Including the interface name, such as “outside”, displays the settings for the specified interface. Excluding the name from the command displays the setting for all interfaces that have IPv6 enabled on them. The output for the command shows the following:

- The name and status of the interface.
- The link-local and global unicast addresses.
- The multicast groups the interface belongs to.
- ICMP redirect and error message settings.
- Neighbor discovery settings.

The following is sample output from the **show ipv6 interface** command:

```
hostname# show ipv6 interface

ipv6interface is down, line protocol is down
  IPv6 is enabled, link-local address is fe80::20d:88ff:feee:6a82 [TENTATIVE]
  No global unicast address is configured
  Joined group address(es):
    ff02::1
    ff02::1:ffee:6a82
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
```

Note The **show interface** command only displays the IPv4 settings for an interface. To see the IPv6 configuration on an interface, you need to use the **show ipv6 interface** command. The **show ipv6 interface** command does not display any IPv4 settings for the interface (if both are configured on the interface).

Viewing IPv6 Routes

To display the routes in the IPv6 routing table, enter the following command:

```
hostname# show ipv6 route
```

The output from the **show ipv6 route** command is similar to the IPv4 **show route** command. It displays the following information:

- The protocol that derived the route.
- The IPv6 prefix of the remote network.
- The administrative distance and metric for the route.
- The address of the next-hop router.
- The interface through which the next hop router to the specified network is reached.

The following is sample output from the **show ipv6 route** command:

```
hostname# show ipv6 route
```

```
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
L   fe80::/10 [0/0]
     via ::, inside
L   fec0::a:0:0:a0a:a70/128 [0/0]
     via ::, inside
C   fec0:0:0:a::/64 [0/0]
     via ::, inside
L   ff00::/8 [0/0]
     via ::, inside
```

CHAPTER 11

Configuring AAA Servers and the Local Database

This chapter describes support for AAA (pronounced “triple A”) and how to configure AAA servers and the local database.

This chapter includes the following sections:

AAA Overview

AAA enables the FWSM to determine who the user is (authentication), what the user can do (authorization), and what the user did (accounting).

AAA provides an extra level of protection and control for user access than using access lists alone. For example, you can create an access list allowing all outside users to access Telnet on a server on an inside interface. If you want only some users to access the server and you might not always know IP addresses of these users, you can enable AAA to allow only authenticated and/or authorized users to make it through the FWSM. (The Telnet server enforces authentication, too; the FWSM prevents unauthorized users from attempting to access the server.)

You can use authentication alone or with authorization and accounting. Authorization always requires a user to be authenticated first. You can use accounting alone, or with authentication and authorization.

If you use multiple security contexts, AAA settings are discrete per context, not shared between contexts. This provides you the opportunity to control access, authorize resources and commands, and perform accounting differently among contexts.

This section includes the following topics:

About Authentication

Authentication controls access by requiring valid user credentials, which are typically a username and password. You can configure the FWSM to authenticate the following items:

- All administrative connections to the FWSM including the following sessions:
  - Telnet
  - SSH
  - Serial console
  - ASDM (using HTTPS)
  - VPN management access
- The **enable** command
- Network access

About Authorization

Authorization controls access *per user* after users authenticate. You can configure the FWSM to authorize the following items:

- Management commands
- Network access
- VPN access for management connections

Authorization controls the services and commands available to each authenticated user. Were you not to enable authorization, authentication alone would provide the same access to services for all authenticated users.

If you need the control that authorization provides, you can configure a broad authentication rule, and then have a detailed authorization configuration. For example, you authenticate inside users who attempt to access any server on the outside network and then limit the outside servers that a particular user can access using authorization.

The FWSM caches the first 16 authorization requests per user, so if the user accesses the same services during the current authentication session, the FWSM does not resend the request to the authorization server.

About Accounting

Accounting tracks traffic that passes through the FWSM, enabling you to have a record of user activity. If you enable authentication for that traffic, you can account for traffic per user. If you do not authenticate the traffic, you can account for traffic per IP address. Accounting information includes when sessions start and stop, username, the number of bytes that pass through the FWSM for the session, the service used, and the duration of each session.

AAA Server and Local Database Support

The FWSM supports a variety of AAA server types and a local database that is stored on the FWSM. This section describes support for each AAA server type and the local database.

This section includes the following topics:

Summary of Support

Table 11-1 summarizes the support for each AAA service by each AAA server type, including the local database. For more information about support for a specific AAA server type, see the topics following the table.

Table 11-1 Summary of AAA Support

	Database Type						
AAA Service	**Local**	**RADIUS**	**TACACS+**	**SDI**	**NT**	**Kerberos**	**LDAP**
Authentication of. . .							
VPN users[1]	Yes	Yes	Yes	Yes	Yes	Yes	No
Firewall sessions	Yes	Yes	Yes	No	No	No	No
Administrators	Yes	Yes	Yes	No	No	No	No
Authorization of. . .							
VPN users[1]	Yes	Yes	No	No	No	No	Yes
Firewall sessions	No	Yes[2]	Yes	No	No	No	No
Administrators	Yes[3]	No	Yes	No	No	No	No
Accounting of. . .							
VPN connections[1]	No	Yes	Yes	No	No	No	No
Firewall sessions	No	Yes	Yes	No	No	No	No
Administrators	No	No	Yes	No	No	No	No

1. VPN is available for management connections only.
2. For firewall sessions, RADIUS authorization is supported with user-specific access lists only, which are received or specified in a RADIUS authentication response.
3. Local command authorization is supported by privilege level only.

RADIUS Server Support

The FWSM supports RADIUS servers.

This section contains the following topics:

Authentication Methods

The FWSM supports the following authentication methods with RADIUS:

- PAP
- CHAP
- MS-CHAPv1
- MS-CHAPv2

  MS-CHAPv2 supports password management when the RADIUS server communicates with a Windows Active Directory server. When your password expires, you are prompted to change your password (see the **auth-prompt** command).

Attribute Support

The FWSM supports the following sets of RADIUS attributes:

- Authentication attributes defined in RFC 2138.
- Accounting attributes defined in RFC 2139.
- RADIUS attributes for tunneled protocol support, defined in RFC 2868.
- Cisco IOS VSAs, identified by RADIUS vendor ID 9.
- Cisco VPN-related VSAs, identified by RADIUS vendor ID 3076.
- Microsoft VSAs, defined in RFC 2548.

RADIUS Authorization Functions

The FWSM can use RADIUS servers for user authorization for network access using dynamic access lists or access list names per user. To implement dynamic access lists, you must configure the RADIUS server to support it. When the user authenticates, the RADIUS server sends a downloadable access list or access list name to the security appliance. Access to a given service is either permitted or denied by the access list. The security appliance deletes the access list when the authentication session expires.

TACACS+ Server Support

The security appliance supports TACACS+ authentication with ASCII, PAP, CHAP, and MS-CHAPv1.

SDI Server Support

The FWSM can use RSA SecureID servers for VPN authentication. These servers are also known as SDI servers. When a user attempts to establish VPN access and the applicable tunnel-group record specifies a SDI authentication server group, the FWSM sends to the SDI server the username and one-time password and grants or denies user access based on the response from the server.

This section contains the following topics:

SDI Version Support

The FWSM offers the following SDI version support:

- **Versions prior to Version 5.0**—SDI versions prior to 5.0 use the concept of an SDI master and an SDI slave server which share a single node secret file (SECURID).
- **Versions 5.0**—SDI Version 5.0 uses the concepts of an SDI primary and SDI replica servers. Each primary and its replicas share a single node secret file. The node secret file has its name based on the hexadecimal value of the ACE/Server IP address with .sdi appended.

  A Version 5.0 SDI server that you configure on the FWSM can be either the primary or any one of the replicas. See the "SDI Primary and Replica Servers" section on page 11-5 for information about how the SDI agent selects servers to authenticate users.

Two-step Authentication Process

SDI Version 5.0 uses a two-step process to prevent an intruder from capturing information from an RSA SecurID authentication request and using it to authenticate to another server. The SDI agent first sends a lock request to the SecurID server before sending the user authentication request. The server locks the username, preventing another (replica) server from accepting it. This means that the same user cannot authenticate to two FWSMs using the same authentication servers simultaneously. After a successful username lock, the FWSM sends the passcode.

SDI Primary and Replica Servers

The FWSM obtains the server list when the first user authenticates to the configured server, which can be either a primary or a replica. The FWSM then assigns priorities to each of the servers on the list, and subsequent server selection derives at random from those assigned priorities. The highest priority servers have a higher likelihood of being selected.

NT Server Support

The FWSM supports authentication of VPN-based management connections with Microsoft Windows server operating systems that support NTLM Version 1, which we collectively refer to as NT servers. When a user attempts to establish VPN access and the applicable tunnel-group record specifies an NT authentication server group, the FWSM uses NTLM Version 1 to for user authentication with the Microsoft Windows domain server. The FWSM grants or denies user access based on the response from the domain server.

Note NT servers have a maximum length of 14 characters for user passwords. Longer passwords are truncated. This is a limitation of NTLM Version 1.

Kerberos Server Support

The FWSM can use Kerberos servers for VPN-based management connections. When a user attempts to establish VPN access, and the traffic matches an authentication statement, the FWSM consults the Kerberos server for user authentication and grants or denies user access based on the response from the server.

The FWSM supports 3DES, DES, and RC4 encryption types.

Note The FWSM does not support changing user passwords during tunnel negotiation. To avoid this situation happening inadvertently, disable password expiration on the Kerberos/Active Directory server for users connecting to the FWSM.

LDAP Server Support

The FWSM can use LDAP servers for authorization of VPN-based management connections. When user authentication for VPN access has succeeded and the applicable tunnel-group record specifies an LDAP authorization server group, the FWSM queries the LDAP server and applies to the VPN session the authorizations it receives.

Local Database Support

The FWSM maintains a local database that you can populate with user profiles.

This section contains the following topics:

User Profiles

User profiles contain, at a minimum, a username. Typically, a password is assigned to each username, although passwords are optional.

The **username attributes** command enables you to enter the username mode. In this mode, you can add other information to a specific user profile. The information you can add includes VPN-related attributes, such as a VPN session timeout value.

Fallback Support

With the exception of fallback for network access authentication, the local database can act as a fallback method for the functions in Table 11-1. This behavior is designed to help you prevent accidental lockout from the FWSM.

For users who need fallback support, we recommend that their usernames and passwords in the local database match their usernames and passwords in the AAA servers. This provides transparent fallback support. Because the user cannot determine whether a AAA server or the local database is providing the service, using usernames and passwords on AAA servers that are different than the usernames and passwords in the local database means that the user cannot be certain which username and password should be given.

The local database supports the following fallback functions:

- **Console and enable password authentication**—When you use the **aaa authentication console** command, you can add the **LOCAL** keyword after the AAA server group tag. If the servers in the group all are unavailable, the FWSM uses the local database to authenticate administrative access. This can include enable password authentication, too.
- **Command authorization**—When you use the **aaa authorization command** command, you can add the **LOCAL** keyword after the AAA server group tag. If the TACACS+ servers in the group all are unavailable, the local database is used to authorize commands based on privilege levels.
- **VPN authentication and authorization**—VPN authentication and authorization are supported to enable remote access to the FWSM if AAA servers that normally support these VPN services are unavailable. The **authentication-server-group** command, available in tunnel-group general attributes mode, lets you specify the **LOCAL** keyword when you are configuring attributes of a tunnel group. When VPN client of an administrator specifies a tunnel group configured to fallback to the local database, the VPN tunnel can be established even if the AAA server group is unavailable, provided that the local database is configured with the necessary attributes.

Configuring the Local Database

This section describes how to manage users in the local database. You can use the local database for CLI access authentication, privileged mode authentication, command authorization, network access authentication, and VPN authentication and authorization. You cannot use the local database for network access authorization. The local database does not support accounting.

You cannot enter the **username** command in the system execution space. However, when you use the **login** command in system, or use Telnet authentication when you session to the FWSM from the switch, the FWSM uses the admin context username database (Telnet authentication for the system execution space is also configured in the admin context).

Caution If you add to the local database users who can gain access to the CLI but who should not be allowed to enter privileged mode, enable command authorization. (See the "Configuring Local Command Authorization" section on page 22-15.) Without command authorization, users can access privileged mode (and all commands) at the CLI using their own password if their privilege level is 2 or greater (2 is the default). Alternatively, you can use RADIUS or TACACS+ authentication so that the user will not be able to use the **login** command, or you can set all local users to level 1 so you can control who can use the system enable password to access privileged mode.

To define a user account in the local database, perform the following steps:

Step 1 Create the user account. To do so, enter the following command:

```
hostname(config)# username username {nopassword | password password} [privilege level]
```

where the options are as follows:

- *username*—A string from 4 to 64 characters long.
- **password** *password*—A string from 3 to 16 characters long.
- **privilege** *level*—The privilege level that you want to assign to the new user account (from 0 to 15). The default is 2. This privilege level is used with command authorization.
- **nopassword**—Creates a user account with no password.

The **encrypted** keyword is typically for display only. When you define a password in the **username** command, the FWSM encrypts it when it saves it to the configuration for security purposes. When you enter the **show running-config** command, the **username** command does not show the actual password; it shows the encrypted password followed by the **encrypted** keyword. For example, if you enter the password "test," the **show running-config** display would appear to be something like the following:

```
username pat password DLaUiAX3l78qgoB5c7iVNw== nt-encrypted
```

Step 2 To configure a local user account with VPN attributes, perform the following steps:

a. Enter the following command:

```
hostname(config)# username username attributes
```

When you enter the **username attributes** command, you enter username mode. The commands available in this mode are as follows:

- **group-lock**
- **password-storage**
- **vpn-access-hours**
- **vpn-filter**
- **vpn-framed-ip-address**
- **vpn-group-policy**
- **vpn-idle-timeout**
- **vpn-session-timeout**
- **vpn-simultaneous-logins**
- **vpn-tunnel-protocol**

Use these commands as needed to configure the user profile. For more information about these commands, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

b. When you have finished configuring the user profiles, enter **exit** to return to config mode.

For example, the following command assigns a privilege level of 15 to the admin user account:

```
hostname(config)# username admin password passw0rd privilege 15
```

The following command creates a user account with no password:

```
hostname(config)# username bcham34 nopassword
```

The following commands creates a user account with a password, enters username mode, and specifies a few VPN attributes:

```
hostname(config)# username user1 password gOgeOus
hostname(config)# username user1 attributes
```

```
hostname(config-username)# vpn-tunnel-protocol IPSec
hostname(config-username)# vpn-simultaneous-logins 6
hostname(config-username)# exit
```

Identifying AAA Server Groups and Servers

If you want to use an external AAA server for authentication, authorization, or accounting, you must first create at least one AAA server group per AAA protocol and add one or more servers to each group. You identify AAA server groups by name. Each server group is specific to one type of server: Kerberos, LDAP, NT, RADIUS, SDI, or TACACS+.

The FWSM contacts the first server in the group. If that server is unavailable, the FWSM contacts the next server in the group, if configured. If all servers in the group are unavailable, the FWSM tries the local database if you configured it as a fallback method (management authentication and authorization only). If you do not have a fallback method, the FWSM continues to try the AAA servers.

To create a server group and add AAA servers to it, perform the following steps:

Step 1 For each AAA server group you need to create, perform the following steps:

a. Identify the server group name and the protocol by entering the following command:

```
hostname(config)# aaa-server server_group protocol {kerberos | ldap | nt | radius |
sdi | tacacs+}
```

For example, to use RADIUS to authenticate network access and TACACS+ to authenticate CLI access, you need to create at least two server groups, one for RADIUS servers and one for TACACS+ servers.

You can have up to 15 AAA server groups in single mode or 4 AAA server groups per context in multiple mode. Each group can have up to 16 servers in single mode or 4 servers in multiple mode.

When you enter a **aaa-server** command, you enter group mode.

b. If you want to specify the maximum number of requests sent to a AAA server in the group before trying the next server, enter the following command:

```
hostname(config-aaa-server-group)# max-failed-attempts number
```

The *number* can be between 1 and 5. The default is 3.

If you configured a fallback method using the local database (for management access only; see the "AAA for System Administrators" section on page 22-10 and the "Configuring TACACS+ Command Authorization" section on page 22-18 to configure the fallback mechanism), and all the servers in the group fail to respond, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default) so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the **reactivation-mode** command in the following step.

If you do not have a fallback method, the FWSM continues to retry the servers in the group.

c. If you want to specify the method (reactivation policy) by which failed servers in a group are reactivated, use the **reactivation-mode** command. For more information about this command, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

d. If you want to indicate whether accounting messages are sent to a single server (single mode) or sent to all servers in the group (simultaneous mode), use the **accounting-mode** command. For more information about this command, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Step 2 For each AAA server on your network, perform the following steps:

a. Identify the server, including the AAA server group it belongs to by entering the following command:

```
hostname(config)# aaa-server server_tag [(interface_name)] host server_ip [key]
[timeout seconds]
```

When you enter a **aaa-server host** command, you enter aaa-server host configuration mode.

b. As needed, use host mode commands to further configure the AAA server.

The commands in host mode do not apply to all AAA server types. Table 11-2 lists the available commands, the server types they apply to, and whether a new AAA server definition has a default value for that command. Where a command is applicable to the server type you specified and no default value is provided (indicated by "—"), use the command to specify the value. For more information about these commands, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Table 11-2 Host Mode Commands, Server Types, and Defaults

Command	Applicable AAA Server Types	Default Value
accounting-port	RADIUS	1646
authentication-port	RADIUS	1645
kerberos-realm	Kerberos	—
key	RADIUS	—
	TACACS+	—
ldap-base-dn	LDAP	—
ldap-login-dn	LDAP	—
ldap-login-password	LDAP	—
ldap-naming-attribute	LDAP	—
ldap-scope	LDAP	—
nt-auth-domain-controller	NT	—
radius-common-pw	RADIUS	—
retry-interval	Kerberos	10 seconds
	RADIUS	10 seconds
sdi-pre-5-slave	SDI	—
sdi-version	SDI	sdi-5
server-port	Kerberos	88
	LDAP	389
	NT	139
	SDI	5500
	TACACS+	49
timeout	All	10 seconds

For example, to add one TACACS+ group with one primary and one backup server, one RADIUS group with a single server, and an NT domain server, enter the following commands:

```
hostname(config)# aaa-server AuthInbound protocol tacacs+
hostname(config-aaa-server-group)# max-failed-attempts 2
hostname(config-aaa-server-group)# reactivation-mode depletion deadtime 20
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthInbound (inside) host 10.1.1.1
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# aaa-server AuthInbound (inside) host 10.1.1.2
hostname(config-aaa-server-host)# key TACPlusUauthKey2
hostname(config-aaa-server-host)# exit
hostname(config)# aaa-server AuthOutbound protocol radius
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthOutbound (inside) host 10.1.1.3
hostname(config-aaa-server-host)# key RadUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# aaa-server NTAuth protocol nt
```

```
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server NTAuth (inside) host 10.1.1.4
hostname(config-aaa-server-host)# nt-auth-domain-controller primary1
```

CHAPTER 12

Identifying Traffic with Access Lists

This chapter describes how to identify traffic with access lists. Access lists are used in a variety of features. If your feature uses Modular Policy Framework, you can use an access list to identify traffic within a traffic class map. For more information on Modular Policy Framework, see Chapter 19, "Using Modular Policy Framework." This chapter includes the following sections:

For information about IPv6 access lists, see the "Configuring IPv6 Access Lists" section on page 10-5.

Access List Overview

Access lists are made up of one or more Access Control Entries. An ACE is a single entry in an access list that specifies a permit or deny rule, and is applied to a protocol, a source and destination IP address or network, and optionally the source and destination ports.

This section includes the following topics:

Access List Types

Table 12-1 lists the types of access lists and some common uses for them.

Table 12-1 Access List Types and Common Uses

Access List Use	Access List Type	Description
Control network access for IP traffic (routed and transparent mode)	Extended	The FWSM does not allow any traffic unless it is explicitly permitted by an extended access list. **Note** To access the FWSM interface for management access, you do not also need an access list allowing the host IP address. You only need to configure management access according to Chapter 22, "Configuring Management Access."
Identify traffic for AAA rules	Extended	AAA rules use access lists to identify traffic.
Control network access for IP traffic for a given user	Extended, downloaded from a AAA server per user	You can configure the RADIUS server to download a dynamic access list to be applied to the user, or the server can send the name of an access list that you already configured on the FWSM.
Identify addresses for NAT (policy NAT and NAT exemption)	Extended	Policy NAT lets you identify local traffic for address translation by specifying the source and destination addresses in an extended access list.
Establish VPN access	Extended	You can use an extended access list in VPN commands.
Identify traffic in a traffic class map for Modular Policy	Extended EtherType	Access lists can be used to identify traffic in a class map, which is used for features that support Modular Policy Framework. Features that support Modular Policy Framework include TCP and general connection settings, and inspection.
For transparent firewall mode, control network access for non-IP traffic	EtherType	You can configure an access list that controls traffic based on its EtherType.
Identify OSPF route redistribution	Standard	Standard access lists include only the destination address. You can use a standard access list to control the redistribution of OSPF routes.

Access Control Entry Order

An access list is made up of one or more Access Control Entries. Depending on the access list type, you can specify the source and destination addresses, the protocol, the ports (for TCP or UDP), the ICMP type (for ICMP), or the EtherType.

Each ACE that you enter for a given access list name is appended to the end of the access list unless you specify the line number in the ACE (extended access lists only).

The order of ACEs is important. When the FWSM decides whether to forward or drop a packet, the FWSM tests the packet against each ACE in the order in which the entries are listed. After a match is found, no more ACEs are checked. For example, if you create an ACE at the beginning of an access list that explicitly permits all traffic, no further statements are ever checked.

You can disable an ACE by making it inactive.

Access List Implicit Deny

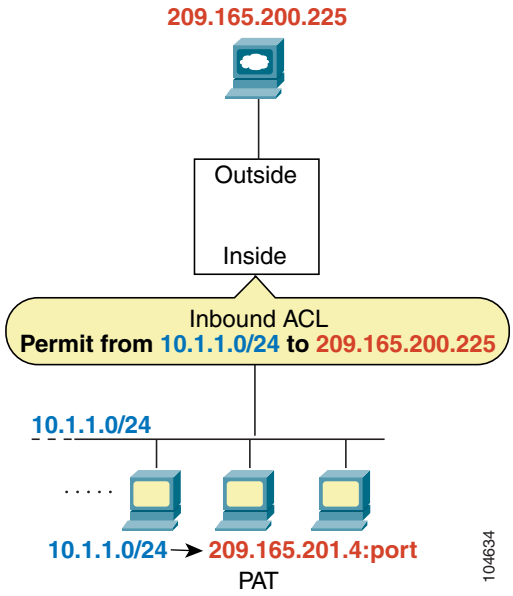
Access lists have an implicit deny at the end of the list, so unless you explicitly permit it, traffic cannot pass. For example, if you want to allow all users to access a network through the FWSM except for particular addresses, then you need to deny the particular addresses and then permit all others.

IP Addresses Used for Access Lists When You Use NAT

When you use NAT, the IP addresses you specify for an access list depend on the interface to which the access list is attached; you need to use addresses that are valid on the network connected to the interface. This guideline applies for both inbound and outbound access groups: the direction does not determine the address used, only the interface does.

For example, you want to apply an access list to the inbound direction of the inside interface. You configure the FWSM to perform NAT on the inside source addresses when they access outside addresses. Because the access list is applied to the inside interface, the source addresses are the original untranslated addresses. Because the outside addresses are not translated, the destination address used in the access list is the real address (see Figure 12-1).

Figure 12-1 IP Addresses in Access Lists: NAT Used for Source Addresses

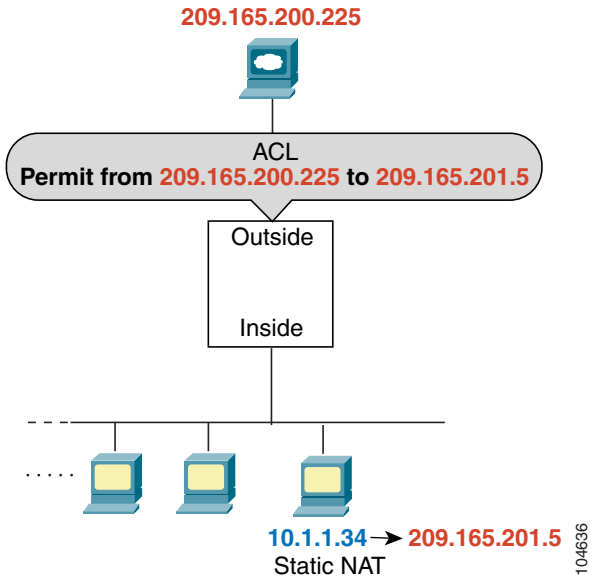


See the following commands for this example:

```
hostname(config)# access-list INSIDE extended permit ip 10.1.1.0 255.255.255.0 host
209.165.200.225
hostname(config)# access-group INSIDE in interface inside
```

If you want to allow an outside host to access an inside host, you can apply an inbound access list on the outside interface. You need to specify the translated address of the inside host in the access list because that address is the address that can be used on the outside network (see Figure 12-2).

Figure 12-2 IP Addresses in Access Lists: NAT used for Destination Addresses

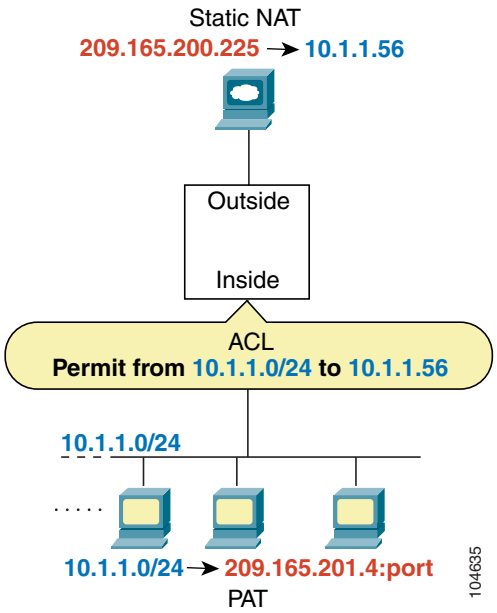


See the following commands for this example:

```
hostname(config)# access-list OUTSIDE extended permit ip host 209.165.200.225 host
209.165.201.5
hostname(config)# access-group OUTSIDE in interface outside
```

If you perform NAT on both interfaces, keep in mind the addresses that are visible to a given interface. In Figure 12-3, an outside server uses static NAT so that a translated address appears on the inside network.

Figure 12-3 IP Addresses in Access Lists: NAT used for Source and Destination Addresses



See the following commands for this example:

```
hostname(config)# access-list INSIDE extended permit ip 10.1.1.0 255.255.255.0 host
10.1.1.56
hostname(config)# access-group INSIDE in interface inside
```

Access List Commitment

When you add an ACE to an access list, the FWSM activates the access list by committing it to the network processors. The FWSM waits a short period of time after you last entered an **access-list** command and then commits the access list. If you enter an ACE after the commitment starts, the FWSM aborts the commitment and recommits the access list after a short waiting period. The FWSM displays a message similar to the following after it commits the access list:

```
Access Rules Download Complete: Memory Utilization: < 1%
```

Large access lists of approximately 60 K ACEs can take 3 to 4 minutes to commit, depending on the size.

Note To keep this message from displaying after every access list change and subsequent committal to the network processor, enter the **np acl-notify disable** command. This command is local and not saved in the startup configuration, so it does not replicate to the peer through failover, and you must re-enter the command after each reload.

For information about exceeding memory limits, see the "Maximum Number of ACEs" section.

Maximum Number of ACEs

The FWSM supports a maximum number of ACEs for the entire system. See the "Rule Limits" section on page A-6 for detailed information about rule limits, including for ACEs and other types of rules.

Some access lists use more memory than others, and these include access lists that use large port number ranges or overlapping networks (for example one ACE specifies 10.0.0.0/8 and another specifies 10.1.1.0/24, resulting in ACEs with overlapping networks). Depending on the type of access list, the actual limit the system can support will be less than the maximum.

If you use object groups in ACEs, the number of actual ACEs that you enter is fewer, but the number of *expanded* ACEs is the same as without object groups, and expanded ACEs count towards the system limit. To view the number of expanded ACEs in an access list, enter the **show access-list** command.

When you add an ACE, and the FWSM commits the access list, the console displays the memory used in a message similar to the following:

```
Access Rules Download Complete: Memory Utilization: < 1%
```

If you exceed the memory limitations, you receive an error message and a system log message (106024), and all the access lists that were added in this commitment are removed from the configuration. Only the set of access lists that were successfully committed in the previous commitment are used. For example, if you paste 1000 ACEs at the prompt, and the last ACE exceeds the memory limitations, all 1000 ACEs are rejected.

Adding an Extended Access List

This section describes how to add an extended access list, and includes the following topics:

Extended Access List Overview

An extended access list is made up of one or more ACEs, in which you can specify the line number to insert the ACE, source and destination addresses, and, depending on the ACE type, the protocol, the ports (for TCP or UDP), or the ICMP type (for ICMP). You can identify all of these parameters within the **access-list** command, or you can use object groups for each parameter. This section describes how to identify the parameters within the command. To use object groups, see the "Simplifying Access Lists with Object Grouping" section on page 12-11.

For information about logging options that you can add to the end of the ACE, see the "Logging Access List Activity" section on page 12-25. For information about time range options, see "Scheduling Extended Access List Activation" section on page 12-24.

For TCP and UDP connections for both routed and transparent mode, you do not need an access list to allow returning traffic, because the FWSM allows all returning traffic for established, bidirectional connections. For connectionless protocols such as ICMP, however, the FWSM establishes unidirectional sessions, so you either need access lists to allow ICMP in both directions (by applying access lists to the source and destination interfaces), or you need to enable the ICMP inspection engine. The ICMP inspection engine treats ICMP sessions as bidirectional connections.

You can apply only one access list of each type (extended and EtherType) to each direction of an interface. You can apply the same access lists on multiple interfaces. See Chapter 14, "Permitting or Denying Network Access," for more information about applying an access list to an interface.

Note If you change the access list configuration, and you do not want to wait for existing connections to time out before the new access list information is used, you can clear the connections using the **clear local-host** command.

Allowing Broadcast and Multicast Traffic through the Transparent Firewall

In routed firewall mode, broadcast and multicast traffic is blocked even if you allow it in an access list, including unsupported dynamic routing protocols and DHCP (unless you configure DHCP relay). Transparent firewall mode can allow any IP traffic through. This feature is especially useful in multiple context mode, which does not allow dynamic routing, for example.

Note Because these special types of traffic are connectionless, you need to apply an extended access list to both interfaces, so returning traffic is allowed through.

Table 12-2 lists common traffic types that you can allow through the transparent firewall.

Table 12-2 Transparent Firewall Special Traffic

Traffic Type	Protocol or Port	Notes
DHCP	UDP ports 67 and 68	If you enable the DHCP server, then the FWSM does not pass DHCP packets.
EIGRP	Protocol 88	—
OSPF	Protocol 89	—
Multicast streams	The UDP ports vary depending on the application.	Multicast streams are always destined to a Class D address (224.0.0.0 to 239.x.x.x).
RIP (v1 or v2)	UDP port 520	—

Adding an Extended ACE

When you enter the **access-list** command for a given access list name, the ACE is added to the end of the access list unless you specify the **line** number.

To add an ACE, enter the following command:

```
hostname(config)# access-list access_list_name [line line_number] [extended]
{deny | permit} protocol source_address mask [operator port] dest_address mask
[operator port | icmp_type] [inactive]
```

Tip Enter the access list name in upper case letters so the name is easy to see in the configuration. You might want to name the access list for the interface (for example, INSIDE), or for the purpose for which it is created (for example, NO_NAT or VPN).

Typically, you identify the **ip** keyword for the protocol, but other protocols are accepted. For a list of protocol names, see the "Protocols and Applications" section on page E-11.

Enter the **host** keyword before the IP address to specify a single address. In this case, do not enter a mask. Enter the **any** keyword instead of the address and mask to specify any address.

You can specify the source and destination ports only for the **tcp** or **udp** protocols. For a list of permitted keywords and well-known port assignments, see the "TCP and UDP Ports" section on page E-11. DNS, Discard, Echo, Ident, NTP, RPC, SUNRPC, and Talk each require one definition for TCP and one for UDP. TACACS+ requires one definition for port 49 on TCP.

Use an *operator* to match port numbers used by the source or destination. The permitted operators are as follows:

- **lt**—less than
- **gt**—greater than
- **eq**—equal to
- **neq**—not equal to
- **range**—an inclusive range of values. When you use this operator, specify two port numbers, for example:

  ```
  range 100 200
  ```

You can specify the ICMP type only for the **icmp** protocol. Because ICMP is a connectionless protocol, you either need access lists to allow ICMP in both directions (by applying access lists to the source and destination interfaces), or you need to enable the ICMP inspection engine (see the "Adding an ICMP Type Object Group" section on page 12-14). The ICMP inspection engine treats ICMP sessions as stateful connections. To control ping, specify **echo-reply** (**0**) (FWSM to host) or **echo** (**8**) (host to FWSM). See the "Adding an ICMP Type Object Group" section on page 12-14 for a list of ICMP types.

When you specify a network mask, the method is different from the Cisco IOS software **access-list** command. The FWSM uses a network mask (for example, 255.255.255.0 for a Class C mask). The Cisco IOS mask uses wildcard bits (for example, 0.0.0.255).

To make an ACE inactive, use the **inactive** keyword. To reenable it, enter the entire ACE without the **inactive** keyword. This feature lets you keep a record of an inactive ACE in your configuration to make reenabling easier.

See the following examples:

The following access list allows all hosts (on the interface to which you apply the access list) to go through the FWSM:

```
hostname(config)# access-list ACL_IN extended permit ip any any
```

The following sample access list prevents hosts on 192.168.1.0/24 from accessing the 209.165.201.0/27 network. All other addresses are permitted.

```
hostname(config)# access-list ACL_IN extended deny tcp 192.168.1.0 255.255.255.0
209.165.201.0 255.255.255.224
hostname(config)# access-list ACL_IN extended permit ip any any
```

If you want to restrict access to only some hosts, then enter a limited permit ACE. By default, all other traffic is denied unless explicitly permitted.

```
hostname(config)# access-list ACL_IN extended permit ip 192.168.1.0 255.255.255.0
209.165.201.0 255.255.255.224
```

The following access list restricts all hosts (on the interface to which you apply the access list) from accessing a website at address 209.165.201.29. All other traffic is allowed.

```
hostname(config)# access-list ACL_IN extended deny tcp any host 209.165.201.29 eq www
hostname(config)# access-list ACL_IN extended permit ip any any
```

Adding an EtherType Access List

Transparent firewall mode only

An EtherType access list is made up of one or more ACEs that specify an EtherType. This section includes the following topics:

Supported EtherTypes

An EtherType ACE controls any EtherType identified by a 16-bit hexadecimal number.

EtherType access lists support Ethernet V2 frames.

802.3-formatted frames are not handled by the access list because they use a length field as opposed to a type field.

BPDUs, which are handled by the access list, are the only exception: they are SNAP-encapsulated, and the FWSM is designed to specifically handle BPDUs.

The FWSM receives trunk port (Cisco proprietary) BPDUs because FWSM ports are trunk ports. Trunk BPDUs have VLAN information inside the payload, so the FWSM modifies the payload with the outgoing VLAN if you allow BPDUs.

Note If you use failover, you must allow BPDUs on both interfaces with an EtherType access list to avoid bridging loops.

Apply Access Lists in Both Directions

Because EtherTypes are connectionless, you need to apply the access list to both interfaces if you want traffic to pass in both directions.

Implicit Deny at the End of an Access List Does Not Affect IP or ARP Traffic

For EtherType access lists, the implicit deny at the end of the access list does not affect IPv4 traffic or ARPs; for example, if you allow EtherType 8037, the implicit deny at the end of the access list does not now block any IP traffic that you previously allowed with an extended access list. IPv4 and ARP traffic cannot be controlled with an EtherType access list.

Using Extended and EtherType Access Lists on the Same Interface

You can apply only one access list of each type (extended and EtherType) to each direction of an interface. You can also apply the same access lists on multiple interfaces.

Allowing MPLS

If you allow MPLS, ensure that Label Distribution Protocol and Tag Distribution Protocol TCP connections are established through the FWSM by configuring both MPLS routers connected to the FWSM to use the IP address on the FWSM interface as the router-id for LDP or TDP sessions. (LDP and TDP allow MPLS routers to negotiate the labels (addresses) used to forward packets.)

On Cisco IOS routers, enter the appropriate command for your protocol, LDP or TDP. The *interface* is the interface connected to the FWSM.

```
hostname(config)# mpls ldp router-id interface force
```

Or

```
hostname(config)# tag-switching tdp router-id interface force
```

Adding an EtherType ACE

To add an EtherType ACE, enter the following command:

```
hostname(config)# access-list access_list_name ethertype {permit | deny} {ipx | bpdu |
mpls-unicast | mpls-multicast | any | hex_number}
```

The *hex_number* is any EtherType that can be identified by a 16-bit hexadecimal number greater than or equal to 0x600. See RFC 1700, "Assigned Numbers," at http://www.ietf.org/rfc/rfc1700.txt for a list of EtherTypes.

When you enter the **access-list** command for a given access list name, the ACE is added to the end of the access list.

Tip Enter the *access_list_name* in upper case letters so the name is easy to see in the configuration. You might want to name the access list for the interface (for example, INSIDE), or for the purpose (for example, MPLS or IPX).

For example, the following sample access list allows common EtherTypes originating on the inside interface:

```
hostname(config)# access-list ETHER ethertype permit ipx
hostname(config)# access-list ETHER ethertype permit bpdu
hostname(config)# access-list ETHER ethertype permit mpls-unicast
hostname(config)# access-group ETHER in interface inside
```

The following access list allows some EtherTypes through the FWSM, but denies IPX:

```
hostname(config)# access-list ETHER ethertype deny ipx
hostname(config)# access-list ETHER ethertype permit 0x1234
hostname(config)# access-list ETHER ethertype permit bpdu
hostname(config)# access-list ETHER ethertype permit mpls-unicast
hostname(config)# access-group ETHER in interface inside
hostname(config)# access-group ETHER in interface outside
```

The following access list denies traffic with EtherType 0x1256, but allows all others on both interfaces:

```
hostname(config)# access-list nonIP ethertype deny 1256
hostname(config)# access-list nonIP ethertype permit any
hostname(config)# access-group ETHER in interface inside
hostname(config)# access-group ETHER in interface outside
```

Adding a Standard Access List

Standard access lists are used in some commands to identify the destination IP addresses only. For example, you use a standard access list to identify the destination addresses of OSPF routes for use in a route map for OSPF redistribution. Standard access lists cannot be applied to interfaces to control traffic.

The following command adds a standard ACE. To add another ACE at the end of the access list, enter another **access-list** command specifying the same access list name.

To add an ACE, enter the following command:

```
hostname(config)# access-list access_list_name standard {deny | permit} {any | ip_address mask}
```

The following sample access list identifies routes to 192.168.1.0/24:

```
hostname(config)# access-list OSPF standard permit 192.168.1.0 255.255.255.0
```

Simplifying Access Lists with Object Grouping

This section describes how to use object grouping to simplify access list creation and maintenance. This section includes the following topics:

How Object Grouping Works

By grouping like-objects together, you can use the object group in an ACE instead of having to enter an ACE for each object separately. You can create the following types of object groups:

- Protocol
- Network
- Service
- ICMP type

For example, consider the following three object groups:

- MyServices—Includes the TCP and UDP port numbers of the service requests that are allowed access to the internal network

- TrustedHosts—Includes the host and network addresses allowed access to the greatest range of services and servers
- PublicServers—Includes the host addresses of servers to which the greatest access is provided

After creating these groups, you could use a single ACE to allow trusted hosts to make specific service requests to a group of public servers.

You can also nest object groups in other object groups.

Note The ACE system limit applies to expanded access lists. If you use object groups in ACEs, the number of actual ACEs that you enter is fewer, but the number of expanded ACEs is the same as without object groups. In many cases, object groups create more ACEs than if you added them manually, because creating ACEs manually leads you to summarize addresses more than an object group does. To view the number of expanded ACEs in an access list, enter the **show access-list** command.

For example, consider a network object group with 100 sources, a network object group with 100 destinations, and a port object group with 5 ports. Permitting the ports from sources to destinations could result in 50,000 ACEs (5 x 100 x 100) in the expanded access list.

Adding Object Groups

This section describes how to add object groups, and includes the following topics:

Adding a Protocol Object Group

To add or change a protocol object group, perform the following steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

To add a protocol group, perform the following steps:

Step 1 To add a protocol group, enter the following command:

```
hostname(config)# object-group protocol grp_id
```

The *grp_id* is a text string up to 64 characters in length.

The prompt changes to protocol configuration mode.

Step 2 (Optional) To add a description, enter the following command:

```
hostname(config-protocol)# description text
```

The description can be up to 200 characters.

Step 3 To define the protocols in the group, enter the following command for each protocol:

```
hostname(config-protocol)# protocol-object protocol
```

The *protocol* is the numeric identifier of the specific IP protocol (1 to 254) or a keyword identifier (for example, **icmp**, **tcp**, or **udp**). To include all IP protocols, use the keyword **ip**. For a list of protocols you can specify, see the "Protocols and Applications" section on page E-11.

For example, to create a protocol group for TCP, UDP, and ICMP, enter the following commands:

```
hostname(config)# object-group protocol tcp_udp_icmp
hostname(config-protocol)# protocol-object tcp
hostname(config-protocol)# protocol-object udp
hostname(config-protocol)# protocol-object icmp
```

Adding a Network Object Group

To add or change a network object group, perform the following steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

Note A network object group supports IPv4 and IPv6 addresses, depending on the type of access list. For more information about IPv6 access lists, see "Configuring IPv6 Access Lists" section on page 10-5.

To add a network group, perform the following steps:

Step 1 To add a network group, enter the following command:

```
hostname(config)# object-group network grp_id
```

The *grp_id* is a text string up to 64 characters in length.

The prompt changes to network configuration mode.

Step 2 (Optional) To add a description, enter the following command:

```
hostname(config-network)# description text
```

The description can be up to 200 characters.

Step 3 To define the networks in the group, enter the following command for each network or address:

```
hostname(config-network)# network-object {host ip_address | ip_address mask}
```

For example, to create network group that includes the IP addresses of three administrators, enter the following commands:

```
hostname(config)# object-group network admins
hostname(config-network)# description Administrator Addresses
hostname(config-network)# network-object host 10.1.1.4
hostname(config-network)# network-object host 10.1.1.78
hostname(config-network)# network-object host 10.1.1.34
```

Adding a Service Object Group

To add or change a service object group, perform the following steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

To add a service group, perform the following steps:

Step 1 To add a service group, enter the following command:

```
hostname(config)# object-group service grp_id {tcp | udp | tcp-udp}
```

The *grp_id* is a text string up to 64 characters in length.

Specify the protocol for the services (ports) you want to add, either **tcp**, **udp**, or **tcp-udp** keywords. Enter **tcp-udp** keyword if your service uses both TCP and UDP with the same port number, for example, DNS (port 53).

The prompt changes to service configuration mode.

Step 2 (Optional) To add a description, enter the following command:

```
hostname(config-service)# description text
```

The description can be up to 200 characters.

Step 3 To define the ports in the group, enter the following command for each port or range of ports:

```
hostname(config-service)# port-object {eq port | range begin_port end_port}
```

For a list of permitted keywords and well-known port assignments, see the "Protocols and Applications" section on page E-11.

For example, to create service groups that include DNS (TCP/UDP), LDAP (TCP), and RADIUS (UDP), enter the following commands:

```
hostname(config)# object-group service services1 tcp-udp
hostname(config-service)# description DNS Group
hostname(config-service)# port-object eq domain

hostname(config-service)# object-group service services2 udp
hostname(config-service)# description RADIUS Group
hostname(config-service)# port-object eq radius
hostname(config-service)# port-object eq radius-acct

hostname(config-service)# object-group service services3 tcp
hostname(config-service)# description LDAP Group
hostname(config-service)# port-object eq ldap
```

Adding an ICMP Type Object Group

To add or change an ICMP type object group, perform the following steps. After you add the group, you can add more objects as required by following this procedure again for the same group name and specifying additional objects. You do not need to reenter existing objects; the commands you already set remain in place unless you remove them with the **no** form of the command.

To add an ICMP type group, perform the following steps:

Step 1 To add an ICMP type group, enter the following command:

```
hostname(config)# object-group icmp-type grp_id
```

The *grp_id* is a text string up to 64 characters in length.

The prompt changes to ICMP type configuration mode.

Step 2 (Optional) To add a description, enter the following command:

```
hostname(config-icmp-type)# description text
```

The description can be up to 200 characters.

Step 3 To define the ICMP types in the group, enter the following command for each type:

```
hostname(config-icmp-type)# icmp-object icmp_type
```

See the "ICMP Types" section on page E-15 for a list of ICMP types.

For example, to create an ICMP type group that includes echo-reply and echo (for controlling ping), enter the following commands:

```
hostname(config)# object-group icmp-type ping
hostname(config-service)# description Ping Group
hostname(config-icmp-type)# icmp-object echo
hostname(config-icmp-type)# icmp-object echo-reply
```

Nesting Object Groups

To nest an object group within another object group of the same type, first create the group that you want to nest according to the "Adding Object Groups" section on page 12-12. Then perform the following steps:

Step 1 To add or edit an object group under which you want to nest another object group, enter the following command:

```
hostname(config)# object-group {{protocol | network | icmp-type} grp_id | service grp_id
{tcp | udp | tcp-udp}}
```

Step 2 To add the specified group under the object group you specified in Step 1, enter the following command:

```
hostname(config-group_type)# group-object grp_id
```

The nested group must be of the same type.

You can mix and match nested group objects and regular objects within an object group.

For example, you create network object groups for privileged users from various departments:

```
hostname(config)# object-group network eng
hostname(config-network)# network-object host 10.1.1.5
hostname(config-network)# network-object host 10.1.1.9
hostname(config-network)# network-object host 10.1.1.89

hostname(config-network)# object-group network hr
```

```
hostname(config-network)# network-object host 10.1.2.8
hostname(config-network)# network-object host 10.1.2.12

hostname(config-network)# object-group network finance
hostname(config-network)# network-object host 10.1.4.89
hostname(config-network)# network-object host 10.1.4.100
```

You then nest all three groups together as follows:

```
hostname(config)# object-group network admin
hostname(config-network)# group-object eng
hostname(config-network)# group-object hr
hostname(config-network)# group-object finance
```

You only need to specify the admin object group in your ACE as follows:

```
hostname(config)# access-list ACL_IN extended permit ip object-group admin host
209.165.201.29
```

Using Object Groups with an Access List

To use object groups in an access list, replace the normal protocol (*protocol*), network (*source_address mask*, and so on), service (*operator port*), or ICMP type (*icmp_type*) parameter with **object-group** *grp_id* parameter.

For example, to use object groups for all available parameters in the **access-list** {**tcp** | **udp**} command, enter the following command:

```
hostname(config)# access-list access_list_name [line line_number] [extended] {deny |
permit} {tcp | udp} object-group nw_grp_id [object-group svc_grp_id] object-group
nw_grp_id [object-group svc_grp_id]
```

You do not have to use object groups for all parameters; for example, you can use an object group for the source address, but identify the destination address with an address and mask.

The following normal access list that does not use object groups restricts several hosts on the inside network from accessing several web servers. All other traffic is allowed.

```
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host 209.165.201.29
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host 209.165.201.29
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host 209.165.201.29
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host 209.165.201.16
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host 209.165.201.16
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host 209.165.201.16
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.4 host 209.165.201.78
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.78 host 209.165.201.78
eq www
hostname(config)# access-list ACL_IN extended deny tcp host 10.1.1.89 host 209.165.201.78
eq www
hostname(config)# access-list ACL_IN extended permit ip any any
hostname(config)# access-group ACL_IN in interface inside
```

If you make two network object groups, one for the inside hosts, and one for the web servers, then the configuration can be simplified and can be easily modified to add more hosts:

```
hostname(config)# object-group network denied
hostname(config-network)# network-object host 10.1.1.4
hostname(config-network)# network-object host 10.1.1.78
hostname(config-network)# network-object host 10.1.1.89

hostname(config-network)# object-group network web
hostname(config-network)# network-object host 209.165.201.29
hostname(config-network)# network-object host 209.165.201.16
hostname(config-network)# network-object host 209.165.201.78

hostname(config-network)# access-list ACL_IN extended deny tcp object-group denied
object-group web eq www
hostname(config)# access-list ACL_IN extended permit ip any any
hostname(config)# access-group ACL_IN in interface inside
```

Displaying Object Groups

To display a list of the currently configured object groups, enter the following command:

```
hostname(config)# show object-group [protocol | network | service | icmp-type | id grp_id]
```

If you enter the command without any parameters, the system displays all configured object groups.

The following is sample output from the **show object-group** command:

```
hostname# show object-group
object-group network ftp_servers
  description: This is a group of FTP servers
  network-object host 209.165.201.3
  network-object host 209.165.201.4
object-group network TrustedHosts
  network-object host 209.165.201.1
  network-object 192.168.1.0 255.255.255.0
  group-object ftp_servers
```

Removing Object Groups

To remove an object group, enter one of the following commands.

Note You cannot remove an object group or make an object group empty if it is used in an access list.

- To remove a specific object group, enter the following command:

  ```
  hostname(config)# no object-group grp_id
  ```

- To remove all object groups of the specified type, enter the following command:

  ```
  hostname(config)# clear object-group [protocol | network | services | icmp-type]
  ```

  If you do not enter a type, all object groups are removed.

Adding Remarks to Access Lists

You can include remarks about entries in any access list, including extended, EtherType, and standard access lists. The remarks make the access list easier to understand.

To add a remark to an access list, enter the following command:

```
hostname(config)# access-list access_list_name [line line_number] remark text
```

When you enter the **access-list remark** command for a given access list name, the remark is added to the end of the access list unless you specify the **line** number.

If you delete an access list using the **clear configure access-list** *access_list_name* command, then all the remarks are also removed.

The text can be up to 100 characters in length. You can enter leading spaces at the beginning of the text. Trailing spaces are ignored.

For example, you can add remarks before each ACE, and the remark appears in the access list in this location. Entering a dash (-) at the beginning of the remark helps set it apart from ACEs.

```
hostname(config)# access-list OUT remark - this is the inside admin address
hostname(config)# access-list OUT extended permit ip host 209.168.200.3 any
hostname(config)# access-list OUT remark - this is the hr admin address
hostname(config)# access-list OUT extended permit ip host 209.168.200.4 any
```

Access List Group Optimization

The access list optimization feature reduces the number of ACEs per group by merging and/or deleting redundant and conflicting ACEs without affecting the semantics of the access list.

This section includes the following topics:

How Access List Group Optimization Works

During optimization, four different cases are examined to determine whether the two rules can be merged (subset, superset, adjacency, and overlap):

- Subset—If rule x is a subset of rule y, rule x is merged down into rule y.

  Before optimization:

  ```
  access-list test extended permit tcp 10.1.1.1 255.255.255.255 any eq 80 [rule x]
  access-list test extended permit tcp 10.1.1.0 255.255.255.0 any  [rule y]
  ```

  After optimization:

  ```
  access-list test extended permit tcp 10.1.1.0 255.255.255.0 any [rule y]
  ```

- Superset—If rule x is a superset of rule y, rule y is merged up into rule x.

  Before optimization:

  ```
  access-list test extended permit udp 10.1.1.0 255.255.255.0 any [rule x]
  access-list test extended permit udp 10.1.1.1 255.255.255.255 any [rule y]
  ```

After optimization:

```
access-list test extended permit udp 10.1.1.0 255.255.255.0 any [rule x]
```

- Adjacency—If rule x is adjacent to rule y, rule y is merged up with rule x.

  Before optimization:

  ```
  access-list test extended permit ip 10.1.1.0 255.255.255.128 any [rule x]
  access-list test extended permit ip 10.1.1.128 255.255.255.128 any [rule y]
  ```

  After optimization:

  ```
  access-list test extended permit ip 10.1.1.0 255.255.255.0 any [rule x]
  ```

- Overlap—If rule x overlaps rule y, rule y is merged up with rule x.

  Before optimization:

  ```
  access-list test extended permit tcp any any range 50 100 [rule x]
  access-list test extended permit tcp any any range 60 120 [rule y]
  ```

  After optimization:

  ```
  access-list test extended permit tcp any any range 50 120 rule x]
  ```

Note Two redundant/overlapping rules cannot be merged if there exists a conflicting rule in the access list located in between the two rules.

- Permit/Deny—If rule x overlaps with rule y and rule z and rule y has an opposite permission/action, rule x cannot be merged with rule z even though both rules have the same permission/action.

  Before optimization:

  ```
  access-list test extended permit tcp any any range 50 100 [rule x]
  access-list test extended deny tcp any any range 80 130 [rule y]
  access-list test extended permit tcp any any range 60 120 [rule z]
  ```

  After optimization:

  ```
  access-list test extended permit tcp any any range 50 100 [rule x]
  access-list test extended deny tcp any any range 80 130 [rule y]
  access-list test extended permit tcp any any range 60 120 [rule z]
  ```

- Logging (default, disable keywords)—If rule x with a “log default” keyword overlaps with rule y with a “log disable” keyword, rule x can be merged with rule y only if both rules have a “permit” action.

  Before optimization:

  ```
  access-list test extended permit tcp any any range 50 100 log default [rule x]
  access-list test extended permit tcp any any range 80 130 log disable [rule y]
  ```

  After optimization:

  ```
  access-list test extended permit tcp any any range 50 130 log default [rule x]
  ```

  Before optimization:

  ```
  access-list test extended deny tcp any any range 50 100 log default [rule x]
  access-list test extended deny tcp any any range 80 130 log disable [rule y]
  ```

  After optimization:

  ```
  access-list test extended deny tcp any any range 50 100 log default [rule x]
  ```

```
access-list test extended deny tcp any any range 80 130 log disable [rule y]
```

- Logging syslog levels / time-range / inactive—Any rule with a log level, time-range or inactive defined cannot be merged with any other rules. It can also act as a blocking rule.

  Before optimization:

```
access-list test extended permit tcp any any range 50 100 [rule x]
access-list test extended permit tcp any any range 80 130 log critical [rule y]
access-list test extended permit tcp any any range 60 120 [rule z]
```

  After optimization:

```
access-list test extended permit tcp any any range 50 100 [rule x]
access-list test extended permit tcp any any range 80 130 log critical [rule y]
access-list test extended permit tcp any any range 60 120 [rule z]
```

Note Access list optimization is relevant to static extended access lists only. Dynamic access lists are not optimized. In addition, when an access list is bound to AAA, policy NAT, and fixup modules, two copies of the rules will coexist in the system. An optimized copy that would be used in case the access list is attached to an access group and the original non-optimized copy used for AAA, policy NAT and fixups.

Configuring Access List Group Optimization

To configure access list group optimization, perform the following steps:

Step 1 To enable access list group optimization, use the following command:

```
hostname(config)# access-list optimization enable
```

To disable access list group optimization, use the **no** form of the command.

Step 2 To show the optimized access list information, use the following command:

```
hostname(config)# show access-list [id] [optimization [detail] [range low high]]
```

The argument *id* identifies the specific access list. The **detail** keyword shows the optimization detail information. The **range** keyword lets you specify a specific low and high access list range arguments.

Step 3 To copy the optimized running configuration to a designated location, use the following command:

```
hostname(config)# copy optimized-running-config [url | running-config | startup-config |
system]
```

The argument *url* specifies the source or destination file to be copied (disk:, ftp:, or tftp:).

Note The **copy optimized-running-config** command overwrites the running configuration, and if you save the configuration, the object-group access list lines may be lost from the running config. Since optimized configurations usually contain more regular ACEs than object-group ACEs, this operation can increase the running configuration size. With a large number of access lists in a configuration, this operation can cause large configuration files that are over 3 MB in size. Therefore, use this command when you are sure that you will not exceed the start-up configuration size limit.

The following is an example of an optimized access list configuration.

Show the original access list configuration:

```
hostname(config)# sh access-list test
access-list test; 13 elements
access-list test line 1 extended permit tcp host 10.1.1.6 host 10.1.1.20 eq www (hitcnt=0) 0x1d3335f6
access-list test line 2 extended permit tcp any host 10.1.1.90 eq ssh (hitcnt=0) 0x9f0b14e0
access-list test line 3 extended permit tcp any host 10.1.1.90 eq ftp (hitcnt=0) 0x7d023e5f
access-list test line 4 extended permit tcp any object-group dns-servers eq domain 0xb4b0751d
access-list test line 4 extended permit tcp any host 10.10.10.5 eq domain (hitcnt=0) 0x9664696e
access-list test line 4 extended permit tcp any host 10.10.10.6 eq domain (hitcnt=0) 0xde9a7aec
access-list test line 4 extended permit tcp any host 10.10.10.7 eq domain (hitcnt=0) 0x5847c29a
access-list test line 4 extended permit tcp any host 10.10.10.8 eq domain (hitcnt=0) 0xa4246eba
access-list test line 4 extended permit tcp any host 10.10.10.9 eq domain (hitcnt=0) 0x85fc0e4a
access-list test line 5 extended permit udp any any eq domain (hitcnt=0) 0xbaf2384c
access-list test line 6 extended permit tcp 10.1.1.0 255.255.255.0 any (hitcnt=0) 0xd07a176b
access-list test line 7 extended permit icmp any any (hitcnt=0) 0xb422e9c2
access-list test line 8 extended permit udp any any neq domain (hitcnt=0) 0x8e2ee97e
access-list test line 9 extended permit tcp any host 10.10.10.5 (hitcnt=0) 0xaa819def
```

Enable access list group optimization:

```
hostname(config)# access-list optimization enable
ACL group optimization is enabled
hostname(config)#
Access Lists Optimization Complete
Access Rules Download Complete: Memory Utilization: < 1%
```

Note When optimization is enabled, rules are optimized and downloaded in the NPs. The original non-optimized rules become inactive. Any addition/deletion of any rule must take place on the original non-optimized access lists. Whenever a new rule is added/deleted, the optimization process is repeated and the message “Access Lists Optimization Complete” defines the end of the optimization process. During that processing time, some of the access lists information may not be accurate until the optimization process is complete.

Show the non-optimized (original) access list again:

```
hostname(config)# show access-list test
access-list test; 13 elements
access-list test line 1 extended permit tcp host 10.1.1.6 host 10.1.1.20 eq www (hitcnt=*) 0x1d3335f6
access-list test line 2 extended permit tcp any host 10.1.1.90 eq ssh (hitcnt=*) 0x9f0b14e0
access-list test line 3 extended permit tcp any host 10.1.1.90 eq ftp (hitcnt=*) 0x7d023e5f
access-list test line 4 extended permit tcp any object-group dns-servers eq domain 0xb4b0751d
access-list test line 4 extended permit tcp any host 10.10.10.5 eq domain (hitcnt=*) 0x9664696e
access-list test line 4 extended permit tcp any host 10.10.10.6 eq domain (hitcnt=*) 0xde9a7aec
access-list test line 4 extended permit tcp any host 10.10.10.7 eq domain (hitcnt=*) 0x5847c29a
access-list test line 4 extended permit tcp any host 10.10.10.8 eq domain (hitcnt=*) 0xa4246eba
access-list test line 4 extended permit tcp any host 10.10.10.9 eq domain (hitcnt=*) 0x85fc0e4a
access-list test line 5 extended permit udp any any eq domain (hitcnt=*) 0xbaf2384c
access-list test line 6 extended permit tcp 10.1.1.0 255.255.255.0 any (hitcnt=0) 0xd07a176b
access-list test line 7 extended permit icmp any any (hitcnt=0) 0xb422e9c2
access-list test line 8 extended permit udp any any neq domain (hitcnt=*) 0x8e2ee97e
access-list test line 9 extended permit tcp any host 10.10.10.5 (hitcnt=0) 0xaa819def
```

Note Some hit count values are represented with an asterisk ‘*’. An asterisk means that the rule has been merged with other rules and thus the hit count cannot be accurate. Hit counts for optimized rules represent the cumulative value of all of the hit counts of the merged or removed rules. There is no way to determine the hit count for every merged or removed rule.

Show the optimized access list:

```
hostname(config)# show access-list test optimization
access-list test;
13 elements before optimization
7 elements after optimization

Reduction rate = 46%
```

```
access-list test line 2 extended permit tcp any host 10.1.1.90 range ftp ssh (hitcnt=0) 0x9f0b14e0
access-list test line 4 extended permit tcp any 10.10.10.6 255.255.255.254 eq domain (hitcnt=0)
0xde9a7aec
access-list test line 4 extended permit tcp any 10.10.10.8 255.255.255.254 eq domain (hitcnt=0)
0xa4246eba
access-list test line 5 extended permit udp any any (hitcnt=0) 0xbaf2384c
access-list test line 6 extended permit tcp 10.1.1.0 255.255.255.0 any (hitcnt=0) 0xd07a176b
access-list test line 7 extended permit icmp any any (hitcnt=0) 0xb422e9c2
access-list test line 10 extended permit tcp any host 10.10.10.5 (hitcnt=0) 0xaa819def
```

Show the optimized access list in detail:

```
hostname(config)# show access-list test optimization detail
access-list test;
13 elements before optimization
7 elements after optimization

Reduction rate = 46%

SUBSET rules   : 2
ADJACENT rules : 5

access-list test line 1 extended permit tcp host 10.1.1.6 host 10.1.1.20 eq www (hitcnt=0) 0x00000000
[Merged to 6: SUBSET]
access-list test line 2 extended permit tcp any host 10.1.1.90 range ftp ssh (hitcnt=0) 0x9f0b14e0
[(3)]
access-list test line 3 extended permit tcp any host 10.1.1.90 eq ftp (hitcnt=0) 0x00000000 [Merged to
2: ADJACENT]
access-list test line 4 extended permit tcp any object-group dns-servers eq domain 0xb4b0751d
access-list test line 4.1 extended permit tcp any host 10.10.10.5 eq domain (hitcnt=0) 0x00000000
[Merged to 9: SUBSET]
access-list test line 4.2 extended permit tcp any 10.10.10.6 255.255.255.254 eq domain (hitcnt=0)
0xde9a7aec  [(4.3)]
access-list test line 4.3 extended permit tcp any host 10.10.10.7 eq domain (hitcnt=0) 0x00000000
[Merged to 4.2: ADJACENT]
access-list test line 4.4 extended permit tcp any 10.10.10.8 255.255.255.254 eq domain (hitcnt=0)
0xa4246eba  [(4.5)]
access-list test line 4.5 extended permit tcp any host 10.10.10.9 eq domain (hitcnt=0) 0x00000000
[Merged to 4.4: ADJACENT]
access-list test line 5 extended permit udp any any (hitcnt=0) 0xbaf2384c  [(8.1,8.2)]
access-list test line 6 extended permit tcp 10.1.1.0 255.255.255.0 any (hitcnt=0) 0xd07a176b  [(1)]
access-list test line 7 extended permit icmp any any (hitcnt=0) 0xb422e9c2
access-list test line 8.1 extended permit udp any any lt domain (hitcnt=0) 0x00000000 [Merged to 5:
ADJACENT]
access-list test line 8.2 extended permit udp any any gt domain (hitcnt=0) 0x00000000 [Merged to 5:
ADJACENT]
access-list test line 9 extended permit tcp any host 10.10.10.5 (hitcnt=0) 0xaa819def  [(4.1)]
```

Note Some rule information may change when merged. Rule 2 was modified because it was merged with rule 3. In order to view the original non-optimized rule 2, the user should refer to the non-optimized (original) access-list (for example, using the **show access-list test** command).

Show the optimized access list range 2 through 5:

```
hostname(config)# show access-list test optimization range 2 5
access-list test;
13 elements before optimization
7 elements after optimization

Reduction rate = 46%

access-list test line 2 extended permit tcp any host 10.1.1.90 range ftp ssh (hitcnt=0) 0x9f0b14e0
access-list test line 4 extended permit tcp any 10.10.10.6 255.255.255.254 eq domain (hitcnt=0)
0xde9a7aec
access-list test line 4 extended permit tcp any 10.10.10.8 255.255.255.254 eq domain (hitcnt=0)
0xa4246eba
access-list test line 5 extended permit udp any any (hitcnt=0) 0xbaf2384c
```

Show the optimized access list range 6 through 9 in detail:

```
hostname(config)# show access-list test optimization detail range 6 9
access-list test;
13 elements before optimization
```

```
7 elements after optimization

Reduction rate = 46%

SUBSET rules   : 2
ADJACENT rules : 5

access-list test line 6 extended permit tcp 10.1.1.0 255.255.255.0 any (hitcnt=0) 0xd07a176b  [(1)]
access-list test line 7 extended permit icmp any any (hitcnt=0) 0xb422e9c2
access-list test line 8.1 extended permit udp any any lt domain (hitcnt=0) 0x00000000 [Merged to 5:
ADJACENT]
access-list test line 8.2 extended permit udp any any gt domain (hitcnt=0) 0x00000000 [Merged to 5:
ADJACENT]
access-list test line 9 extended permit tcp any host 10.10.10.5 (hitcnt=0) 0xaa819def  [(4.1)]
```

Show the currently running optimized access-list

```
hostname(config)# show running-config access-list test optimization
access-list test extended permit tcp any host 10.1.1.90 range ftp ssh
access-list test extended permit tcp any 10.10.10.6 255.255.255.254 eq domain
access-list test extended permit tcp any 10.10.10.8 255.255.255.254 eq domain
access-list test extended permit udp any any
access-list test extended permit tcp 10.1.1.0 255.255.255.0 any
access-list test extended permit icmp any any
access-list test extended permit tcp any host 10.10.10.5
```

To replace original access lists with the optimized ones:

```
hostname(config)# copy optimized-running-config running-config

Destination filename [running-config]?

hostname(config)#
Access Lists Optimization Complete
Access Rules Download Complete: Memory Utilization: < 1%
```

Note Having access list optimization enabled at all time could be a waste of computational and memory resources. If you are satisfied with how the optimized access lists are merged, the original access lists can be replaced with the optimized ones. Note that this action will wipe out all of the original access lists. After copying the optimized access lists, the user may want to disable access list optimization because the newly copied optimized access lists may not be further optimized.

To disable the access list group optimization:

```
hostname(config)# no access-list optimization enable
Disabling ACL optimization will cause ACL rules get increased.
The non optimized rules might be more than the partition rule max
and might cause memory exhaustion to lose partial or all the
access-list configuration after disabling the optimization.
Please save a copy of your current optimized access-list config
before committing this command.
Continue ? [Y]es/[N]o:
ACL group optimization is disabled
hostname(config)# Access Rules Download Complete: Memory Utilization: < 1%

hostname(config)#
```

Note When disabling access list optimization, be aware that the number of the original non-optimized rules (which is often larger than to the number of optimized rules) may exceed the memory availaible to store them. This will cause some rules to be deleted. Thus, it is considered a good practice to back up the original configuration before proceeding with disabling access list group optimization.

Scheduling Extended Access List Activation

You can schedule each ACE to be activated at specific times of the day and week by applying a time range to the ACE. This section includes the following topics:

Adding a Time Range

To add a time range to implement a time-based access list, perform the following steps:

Step 1 Identify the time-range name by entering the following command:

```
hostname(config)# time-range name
```

Step 2 Specify the time range as either a recurring time range or an absolute time range.

Note Users could experience a delay of approximately 80 to 100 seconds after the specified end time for the ACL to become inactive. For example, if the specified end time is 3:50, because the end time is inclusive, the command is picked up anywhere between 3:51:00 and 3:51:59. After the command is picked up, the security appliance finishes any currently running task and then services the command to deactivate the ACL.

Multiple periodic entries are allowed per **time-range** command. If a **time-range** command has both **absolute** and **periodic** values specified, then the **periodic** commands are evaluated only after the **absolute** start time is reached, and are not further evaluated after the **absolute** end time is reached.

- Recurring time range:

  ```
  hostname(config-time-range)# periodic days-of-the-week time to [days-of-the-week] time
  ```

  You can specify the following values for *days-of-the-week*:

  - **monday**, **tuesday**, **wednesday**, **thursday**, **friday**, **saturday**, and **sunday**.
  - **daily**
  - **weekdays**
  - **weekend**

  The *time* is in the format *hh*:*mm*. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.

- Absolute time range:

  ```
  hostname(config-time-range)# absolute start time date [end time date]
  ```

  The *time* is in the format *hh*:*mm*. For example, 8:00 is 8:00 a.m. and 20:00 is 8:00 p.m.

  The *date* is in the format *day month year*; for example, **1 january 2006**.

The following is an example of an absolute time range beginning at 8:00 a.m. on January 1, 2006. Because no end time and date are specified, the time range is in effect indefinitely.

```
hostname(config)# time-range for2006
hostname(config-time-range)# absolute start 8:00 1 january 2006
```

The following is an example of a weekly periodic time range from 8:00 a.m. to 6:00 p.m on weekdays.:

```
hostname(config)# time-range workinghours
hostname(config-time-range)# periodic weekdays 8:00 to 18:00
```

Applying the Time Range to an ACE

To apply the time range to an ACE, enter the following command:

```
hostname(config)# access-list access_list_name [extended] {deny | permit}...[time-range
name]
```

See the "Adding an Extended Access List" section on page 12-6 for complete **access-list** command syntax.

Note If you also enable logging for the ACE, use the **log** keyword before the **time-range** keyword. If you disable the ACE using the **inactive** keyword, use the **inactive** keyword as the last keyword.

The following example binds an access list named "Sales" to a time range named "New_York_Minute."

```
hostname(config)# access-list Sales line 1 extended deny tcp host 209.165.200.225 host
209.165.201.1 time-range New_York_Minute
```

Logging Access List Activity

This section describes how to configure access list logging for extended access lists and Webtype access lists.

This section includes the following topics:

Access List Logging Overview

By default, when traffic is denied by an extended ACE, the FWSM generates system log message 106023 for each denied packet, in the following form:

```
%XXX-106023: Deny protocol src [interface_name:source_address/source_port] dst
interface_name:dest_address/dest_port [type {string}, code {code}] by access_group acl_id
```

If the FWSM is attacked, the number of system log messages for denied packets can be very large. We recommend that you instead enable logging using system log message 106100, which provides statistics for each ACE and lets you limit the number of system log messages produced. Alternatively, you can disable all logging.

Note Only ACEs in the access list generate logging messages; the implicit deny at the end of the access list does not generate a message. If you want all denied traffic to generate messages, add the implicit ACE manually to the end of the access list, as follows.

```
hostname(config)# access-list TEST deny ip any any log
```

The **log** options at the end of the extended **access-list** command lets you to set the following behavior:

- Enable message 106100 instead of message 106023
- Disable all logging
- Return to the default logging using message 106023

System log message 106100 is in the following form:

```
%XXX-n-106100: access-list acl_id {permitted | denied} protocol
interface_name/source_address(source_port) -> interface_name/dest_address(dest_port)
hit-cnt number ({first hit | number-second interval})
```

When you enable logging for message 106100, if a packet matches an ACE, the FWSM creates a flow entry to track the number of packets received within a specific interval. The FWSM generates a system log message at the first hit and at the end of each interval, identifying the total number of hits during the interval. At the end of each interval, the FWSM resets the hit count to 0. If no packets match the ACE during an interval, the FWSM deletes the flow entry.

Note An ACL only denies SYN packets, so if another type of packet comes in, that packet will not show up in the access-list hit counters. TCP packet types other than SYN packets (including RST, SYN-ACK, ACK, PSH, and FIN) are dropped by the FWSM before they can be dropped by an access list. Only SYN packets can create a session in the Adaptive Security Algorithm, so only SYN packets are assessed by the access list.

A flow is defined by the source and destination IP addresses, protocols, and ports. Because the source port might differ for a new connection between the same two hosts, you might not see the same flow increment because a new flow was created for the connection.

Permitted packets that belong to established connections do not need to be checked against access lists; only the initial packet is logged and included in the hit count. For connectionless protocols, such as ICMP, all packets are logged even if they are permitted, and all denied packets are logged.

See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module System Log Messages* for detailed information about this system log message.

Configuring Logging for an ACE

To configure logging for an ACE, see the following information about the **log** option:

```
hostname(config)# access-list access_list_name [extended] {deny | permit}...[log [[level]
[interval secs] | disable | default]]
```

See the "Adding an Extended Access List" section on page 12-6 for complete **access-list** command syntax.

Note If you also enable a time range for the ACE, use the **log** keyword before the **time-range** keyword. If you disable the ACE using the **inactive** keyword, use the **inactive** keyword as the last keyword.

If you enter the **log** option without any arguments, you enable system log message 106100 at the default level (6) and for the default interval (300 seconds). See the following options:

- *level*—A severity level between 0 and 7. The default is 6.
- **interval** *secs*—The time interval in seconds between system log messages, from 1 to 600. The default is 300. This value is also used as the timeout value for deleting an inactive flow.
- **disable**—Disables all access list logging.
- **default**—Enables logging to message 106023. This setting is the same as having no **log** option.

For example, you configure the following access list:

```
hostname(config)# access-list outside-acl permit ip host 1.1.1.1 any log 7 interval 600
hostname(config)# access-list outside-acl permit ip host 2.2.2.2 any
hostname(config)# access-list outside-acl deny ip any any log 2
hostname(config)# access-group outside-acl in interface outside
```

When a packet is permitted by the first ACE of outside-acl, the FWSM generates the following system log message:

```
%PIX-7-106100: access-list outside-acl permitted tcp outside/1.1.1.1(12345) ->
inside/192.168.1.1(1357) hit-cnt 1 (first hit)
```

Although 20 additional packets for this connection arrive on the outside interface, the traffic does not have to be checked against the access list, and the hit count does not increase.

If one more connection by the same host is initiated within the specified 10 minute interval (and the source and destination ports remain the same), then the hit count is incremented by 1 and the following message is displayed at the end of the 10 minute interval:

```
%PIX-7-106100: access-list outside-acl permitted tcp outside/1.1.1.1(12345)->
inside/192.168.1.1(1357) hit-cnt 2 (600-second interval)
```

When a packet is denied by the third ACE, then the FWSM generates the following system log message:

```
%PIX-2-106100: access-list outside-acl denied ip outside/3.3.3.3(12345) ->
inside/192.168.1.1(1357) hit-cnt 1 (first hit)
```

20 additional attempts within a 5 minute interval (the default) result in the following message at the end of 5 minutes:

```
%PIX-2-106100: access-list outside-acl denied ip outside/3.3.3.3(12345) ->
inside/192.168.1.1(1357) hit-cnt 21 (300-second interval)
```

Managing Deny Flows

When you enable logging for message 106100, if a packet matches an ACE, the FWSM creates a flow entry to track the number of packets received within a specific interval. The FWSM has a maximum of 64 K logging flows for ACEs. A large number of flows can exist concurrently at any point of time. To prevent unlimited consumption of memory and CPU resources, the FWSM places a limit on the number of concurrent *deny* flows; the limit is placed only on deny flows (and not permit flows) because they can indicate an attack. When the limit is reached, the FWSM does not create a new deny flow for logging until the existing flows expire.

For example, if someone initiates a DoS attack, the FWSM can create a large number of deny flows in a short period of time. Restricting the number of deny flows prevents unlimited consumption of memory and CPU resources.

When you reach the maximum number of deny flows, the FWSM issues system log message 106100:

```
%XXX-1-106101: The number of ACL log deny-flows has reached limit (number).
```

To configure the maximum number of deny flows and to set the interval between deny flow alert messages (106101), enter the following commands:

- To set the maximum number of deny flows permitted per context before the FWSM stops logging, enter the following command:

  ```
  hostname(config)# access-list deny-flow-max number
  ```

  The *number* is between 1 and 4096. 4096 is the default.

- To set the amount of time between system log messages (number 106101) that identify that the maximum number of deny flows was reached, enter the following command:

  ```
  hostname(config)# access-list alert-interval secs
  ```

  The *seconds* are between 1 and 3600. 300 is the default.

CHAPTER 13

Configuring Failover

This chapter describes the FWSM failover feature, which lets you configure two FWSMs so that one will take over operation if the other one fails. Failover is compatible with both routed and transparent firewall modes, and with single and multiple context modes.

This chapter includes the following sections:

For sample failover configurations, see the "Failover Example Configurations" section on page B-18.

Understanding Failover

The failover configuration requires two identical FWSMs connected to each other through a dedicated failover link and, optionally, a state link. The health of the active interfaces and units is monitored to determine if specific failover conditions are met. If those conditions are met, failover occurs.

FWSM supports two failover configurations, Active/Active failover and Active/Standby failover. Each failover configuration has its own method for determining and performing failover.

With Active/Active failover, both units can pass network traffic. This lets you configure load balancing on your network. Active/Active failover is only available on units running in multiple context mode.

With Active/Standby failover, only one unit passes traffic while the other unit waits in a standby state. Active/Standby failover is available on units running in either single or multiple context mode.

Both failover configurations support stateful or stateless (regular) failover.

This section includes the following topics:

Failover System Requirements

This section describes the software and license requirements for FWSMs in a failover configuration. This section includes the following topics:

Software Requirements

The two units in a failover configuration must have the same major (first number) and minor (second number) software version. However, you can use different versions of the software during an upgrade process; for example, you can upgrade one unit from Version 3.1(1) to Version 3.1(2) and have failover remain active. We recommend upgrading both units to the same version to ensure long-term compatibility.

License Requirements

Both units must have the same license. In the occurrence of a license mismatch, a failover pair enters pseudo-standby mode, a condition in which failover is disabled. FWSMs in an active/active configuration return to the active/standby state and do not pass any traffic.

Failover and State Links

This section describes the failover and the state links, which are dedicated connections between the two units in a failover configuration. This section includes the following topics:

Failover Link

The two units in a failover pair constantly communicate over a failover link to determine the operating status of each unit. The following information is communicated over the failover link:

- The unit state (active or standby).
- Hello messages (keep-alives).
- Network link status.
- MAC address exchange.
- Configuration replication and synchronization.

Caution All information sent over the failover and Stateful Failover links is sent in clear text unless you secure the communication with a failover key.

The failover link uses a special VLAN interface that you do not configure as a normal networking interface; rather, it exists only for failover communications. This VLAN should only be used for the failover link (and optionally for the state link). Sharing the failover link VLAN with any other VLANs can cause intermittent traffic problems and ping and ARP failures. For inter-chassis failover, use dedicated interfaces on the switch for the failover link.

On systems running in multiple context mode, the failover link resides in the system context. This interface and the state link, if used, are the only interfaces that you can configure in the system context. All other interfaces are allocated to and configured from within security contexts.

Note The IP address and MAC address for the failover link do not change at failover.

State Link

To use Stateful Failover, you must configure a state link to pass all state information. This link can be the same as the failover link, but we recommend that you assign a separate VLAN and IP address for the state link. The state traffic can be large, and performance is improved with separate links.

The state link interface is not configured as a normal networking interface; it exists only for Stateful Failover communications and, optionally, for the failover communication if you share the state and failover links.

In multiple context mode, the state link resides in the system context. This interface and the failover interface are the only interfaces in the system context. All other interfaces are allocated to and configured from within security contexts.

Note The IP address and MAC address for the state link do not change at failover.

Caution All information sent over the failover and Stateful Failover links is sent in clear text unless you secure the communication with a failover key.

Intra- and Inter-Chassis Module Placement

You can place the primary and secondary FWSMs within the same switch or in two separate switches. The following sections describe each option:

- Intra-Chassis Failover, page 13-3
- Inter-Chassis Failover, page 13-4

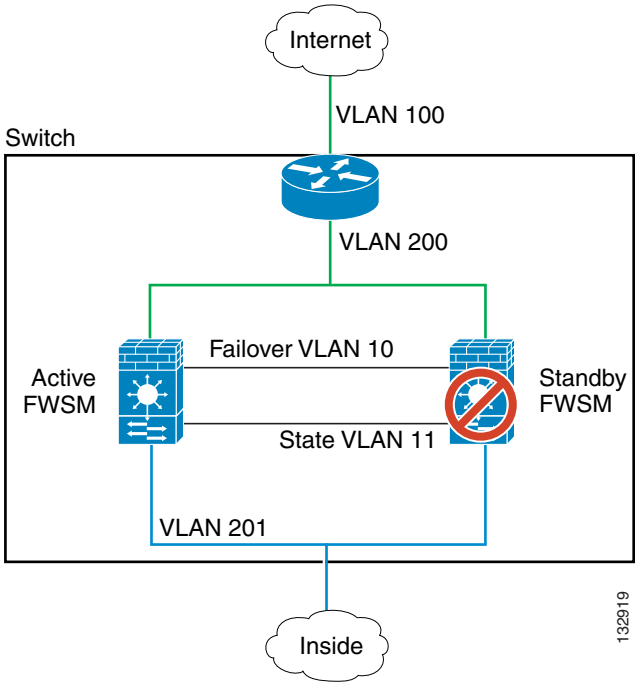
Intra-Chassis Failover

If you install the secondary FWSM in the same switch as the primary FWSM, you protect against module-level failure. To protect against switch-level failure, as well as module-level failure, see the "Inter-Chassis Failover" section on page 13-4.

Even though both FWSMs are assigned the same VLANs, only the active module takes part in networking. The standby module does not pass any traffic.

Figure 13-1 shows a typical intra-switch configuration.

Figure 13-1 Intra-Switch Failover



Inter-Chassis Failover

To protect against switch-level failure, you can install the secondary FWSM in a separate switch. FWSM does not coordinate failover directly with the switch, but it works harmoniously with the switch failover operation. See the switch documentation to configure failover for the switch.

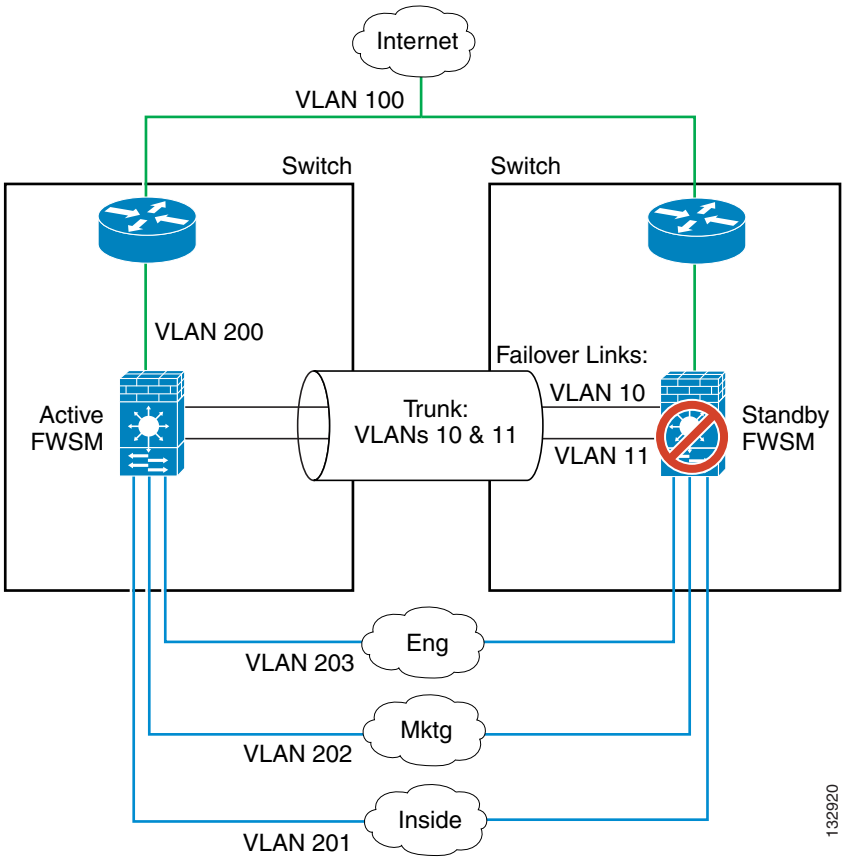
To accommodate the failover communications between FWSMs, we recommend that you configure a trunk port between the two switches that carries the failover and state VLANs. The trunk ensures that failover communication between the two units is subject to minimal failure risk.

For other VLANs, you must ensure that both switches have access to all firewall VLANs, and that monitored VLANs can successfully pass hello packets between both switches.

Figure 13-2 shows a typical switch and FWSM redundancy configuration. The trunk between the two switches carries the failover FWSM VLANs (VLANs 10 and 11).

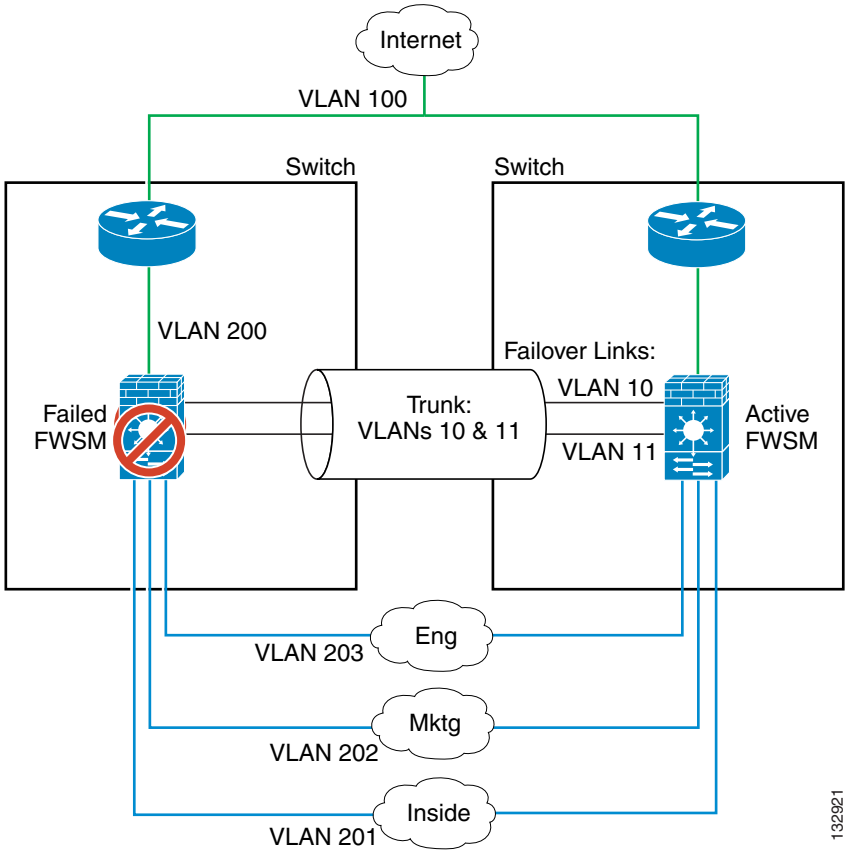
Note FWSM failover is independent of the switch failover operation; however, FWSM works in any switch failover scenario.

Figure 13-2 Normal Operation



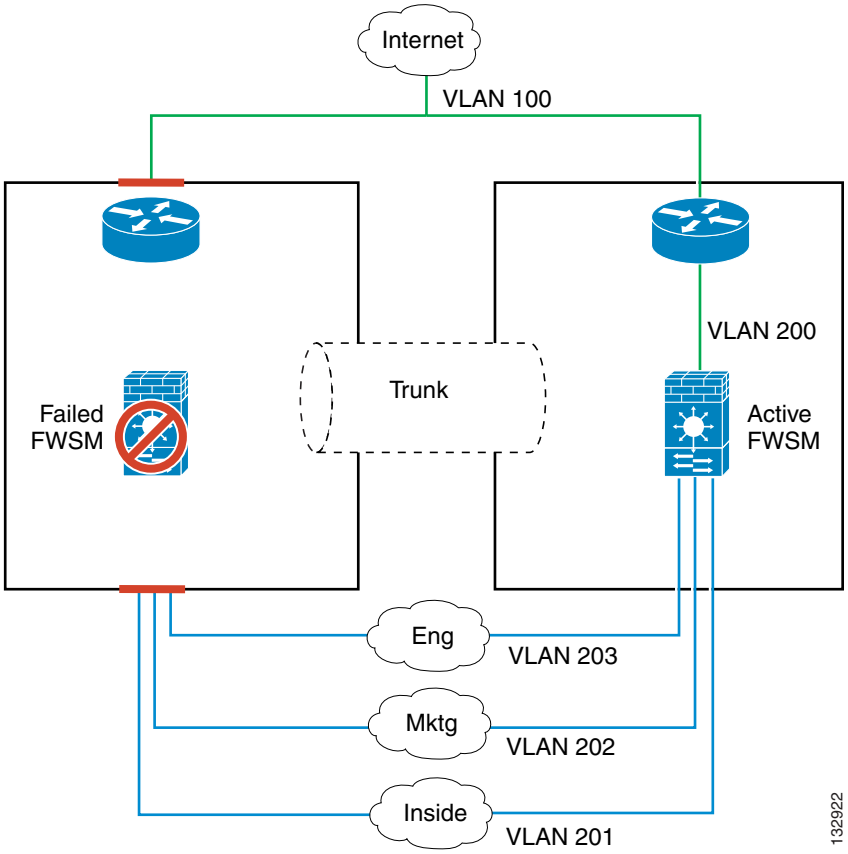
If the primary FWSM fails, then the secondary FWSM becomes active and successfully passes the firewall VLANs (Figure 13-3).

Figure 13-3 FWSM Failure



If the entire switch fails, as well as the FWSM (such as in a power failure), then both the switch and the FWSM fail over to their secondary units (Figure 13-4).

Figure 13-4 Switch Failure



Transparent Firewall Requirements

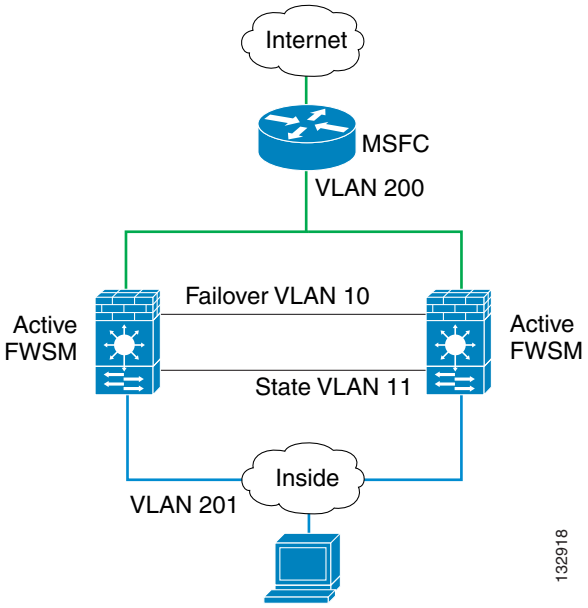
To avoid loops when you use failover in transparent mode, you must use switch software that supports BPDU forwarding, and you must configure the FWSM to allow BPDUs. See the "Switch Hardware and Software Compatibility" section on page A-1 for switch software versions that allow BPDUs automatically.

To allow BPDUs through the FWSM, configure an EtherType ACL and apply it to both interfaces according to the "Adding an EtherType Access List" section on page 12-9.

Loops can occur if both modules are active at the same time, such as when both modules are discovering the presence of the other module, or due to a bad failover link. Because the FWSMs bridge packets between the same two VLANs, loops can occur when inside packets destined for the outside get

endlessly replicated by both FWSMs (see Figure 13-5). The spanning tree protocol can break such loops if there is a timely exchange of BPDUs. To break the loop, BPDUs sent between VLAN 200 and VLAN 201 need to be bridged.

Figure 13-5 Potential Loops in Transparent Mode



Active/Standby and Active/Active Failover

This section describes each failover configuration in detail. This section includes the following topics:

- Active/Standby Failover, page 13-8
- Active/Active Failover, page 13-12
- Determining Which Type of Failover to Use, page 13-17

Active/Standby Failover

This section describes Active/Standby failover and includes the following topics:

- Active/Standby Failover Overview, page 13-9
- Primary/Secondary Status and Active/Standby Status, page 13-9
- Device Initialization and Configuration Synchronization, page 13-9
- Command Replication, page 13-11
- Failover Triggers, page 13-11
- Failover Actions, page 13-12

Active/Standby Failover Overview

Active/Standby failover lets you use a standby FWSM to take over the functionality of a failed unit. When the active unit fails, it changes to the standby state while the standby unit changes to the active state. The unit that becomes active assumes the IP addresses (or, for transparent firewall, the management IP address) and the MAC address of the failed unit and begins passing traffic. The unit that is now in standby state takes over the standby IP addresses and MAC address. Because network devices see no change in the MAC to IP address pairing, no ARP entries change or time out anywhere on the network.

Note For multiple context mode, FWSM can fail over the entire unit (including all contexts) but cannot fail over individual contexts separately.

Primary/Secondary Status and Active/Standby Status

The main difference between the two units in a failover pair is related to which unit is active and which unit is standby, namely which IP addresses are used and which unit actively passes traffic. However, a few differences also exist between the units based on which unit is primary (as specified in the configuration) and which unit is secondary.

- The primary unit always becomes the active unit if both units start up at the same time (and are of equal operational health).
- The primary unit MAC address is always coupled with the active IP addresses.
- By default, the MAC address used for the active FWSM comes from the Burned-in MAC address of the primary FWSM.
- Under certain circumstances, MAC addresses used for the active FWSMs are changed, such as in the following cases:
  - Case 1—The primary FWSM in a failover pair is replaced with a new FWSM
  - Case 2—The secondary FWSM boots and becomes active because it did not detect the primary FWSM.

In Case 1 above, if the primary FWSM is replaced, then as soon as it becomes part of the failover set, the secondary/active FWSM changes the MAC addresses to those of the new primary FWSM.

In Case 2 above, if the secondary FWSM boots without knowing the Burned-in MAC address of the primary FWSM, then it uses its own Burned-in MAC address until it hears from the primary, at which time it swaps the MAC addresses.

Any time the secondary/active FWSM applies new MAC addresses, it sends out gratuitous ARPS for the interface IP addresses but not for the other IP addresses that it owns. These other IP addresses consist of global IP addresses in static and global statements. Therefore, if the Burned-in MAC address of the secondary/active FWSM changes, you must clear the ARP table on the devices that Layer 2 adjacent to the FWSM. Otherwise, the ARP entries for the global IP addresses on those devices will be old and invalid.

Device Initialization and Configuration Synchronization

Configuration synchronization occurs when one or both devices in the failover pair boot. Configurations are always synchronized from the active unit to the standby unit. When the standby unit completes its initial startup, it clears its running configuration (except for the failover commands needed to communicate with the active unit), and the active unit sends its entire configuration to the standby unit.

The active unit is determined by the following:

- If a unit boots and detects a peer already running as active, it becomes the standby unit.
- If a unit boots and does not detect a peer, it becomes the active unit.
- If both units boot simultaneously, then the primary unit becomes the active unit and the secondary unit becomes the standby unit.

Note If the secondary unit boots without detecting the primary unit, it becomes the active unit. It uses its own MAC address for the active IP addresses. However, when the primary unit becomes available, the secondary unit changes the MAC address to that of the primary unit, which can cause an interruption in your network traffic.

When the configuration synchronization starts, the FWSM console on the active unit displays the message "Beginning configuration replication: Sending to mate," and when it is complete, the FWSM console displays the message "End Configuration Replication to mate." During the configuration synchronization, commands entered on the active unit may not replicate properly to the standby unit, and commands entered on the standby unit may be overwritten by the configuration being replicated from the active unit. Avoid entering commands on either unit in the failover pair during the configuration replication process. Depending upon the size of the configuration, replication can take from a few seconds to several minutes.

If you enter the **write standby** command on the active unit, the standby unit clears its running configuration (except for the failover commands used to communicate with the active unit), and the active unit sends its entire configuration to the standby unit.

In multiple context mode, when you enter the **write standby** command in the system execution space, all contexts are replicated. If you enter the **write standby** command within a context, the command replicates only the context configuration.

On the standby unit, the replicated configuration exists only in running memory. To save the configuration to Flash memory after synchronization:

- In single context mode, enter the **write memory** command on the active unit. The command is replicated to the standby unit, which proceeds to write its configuration to Flash memory.
- In multiple context mode, enter the **write memory all** command on the active unit from the system execution space. This command saves the system configuration and all context configurations. The command is replicated to the standby unit, which proceeds to write its configurations to Flash memory. Contexts with startup configurations on external servers are accessible from either unit over the network and do not need to be saved separately for each unit. Alternatively, you can copy the contexts on disk from the active unit to an external server, and then copy them to disk on the standby unit, where they become available when the unit reloads.

Note If you previously changed the number or size of memory partitions on the primary unit (see the "Managing Memory for Rules" section on page 4-11), then after the secondary unit synchronizes the configuration, immediately reload the secondary unit so that the memory partitions are the same. During the initial synchronization, the configuration might not fit properly in the secondary unit memory partitions, but after reloading, and another configuration synchronization, the secondary unit will be operational.

Command Replication

As commands are entered on the active unit, they are sent across the failover link to the standby unit. Command replication always flows from the active unit to the standby unit. Replicated commands are stored in the running configuration of the standby unit. Saving the running configuration to the startup configuration on the active unit causes the running configuration to be saved to the startup configuration on the standby unit; however, you do not have to save the active configuration to Flash memory to replicate the commands.

Note The RSA keys are not synchronized from the primary to the secondary unit in FWSM.

The following commands are replicated to the standby unit:

- all configuration commands except for the **mode** and **failover lan unit** commands
- **copy running-config startup-config**
- **delete**
- **mkdir**
- **rename**
- **rmdir**
- **write memory**

The following commands are not replicated to the standby unit:

- all forms of the **copy** command except for **copy running-config startup config**
- all forms of the **write** command except for **write memory**
- **asdm disconnect**
- **debug**
- **failover lan unit**
- **failover suspend-config-sync**
- **mode**
- **show**
- **ssh disconnect**

Changes made on the standby unit are not replicated to the active unit. If you enter a command on the standby unit, FWSM displays the message `**** WARNING **** Configuration Replication is NOT performed from Standby unit to Active unit. Configurations are no longer synchronized.` This message displays even when you enter many commands that do not affect the configuration.

Failover Triggers

The unit can fail if one of the following events occurs:

- The unit has a hardware failure or a power failure.
- The unit has a software failure.
- Too many monitored interfaces fail.
- The **no failover active** command is entered on the active unit or the **failover active** command is entered on the standby unit.

Failover Actions

In Active/Standby failover, failover occurs on a unit basis. Even on systems running in multiple context mode you cannot fail over individual or groups of contexts with Active/Standby failover.

Table 13-1 shows the failover action for each failure event. For each failure event, the table shows the failover policy (failover or no failover), the action taken by the active unit, the action taken by the standby unit, and any special notes about the failover condition and actions.

Table 13-1 Failover Behavior

Failure Event	Policy	Active Action	Standby Action	Notes
Active unit failed (power or hardware)	Failover	n/a	Become active Mark active as failed	No hello messages are received on any monitored interface or the failover link.
Formerly active unit recovers	No failover	Become standby	No action	None.
Standby unit failed (power or hardware)	No failover	Mark standby as failed	n/a	When the standby unit is marked as failed, then the active unit will not attempt to fail over, even if the interface failure threshold is surpassed.
Failover link failed during operation	No failover	Mark failover interface as failed	Mark failover interface as failed	You should restore the failover link as soon as possible because the unit cannot fail over to the standby unit while the failover link is down.
Failover link failed at startup	No failover	Mark failover interface as failed	Become active	If the failover link is down at startup, both units will become active.
State link failed	No failover	No action	No action	State information will become out of date, and sessions will be terminated if a failover occurs.
Interface failure on active unit above threshold	Failover	Mark active as failed	Become active	None.
Interface failure on standby unit above threshold	No failover	No action	Mark standby as failed	When the standby unit is marked as failed, then the active unit will not attempt to fail over even if the interface failure threshold is surpassed.

Active/Active Failover

This section describes Active/Active failover. This section includes the following topics:

- Failover Triggers, page 13-15
- Failover Actions, page 13-16

Active/Active Failover Overview

Active/Active failover is only available to FWSMs in multiple context mode. In an Active/Active failover configuration, both FWSMs can pass network traffic.

In Active/Active failover, you divide the security contexts on FWSM into *failover groups*. A failover group is simply a logical group of one or more security contexts. You can create a maximum of two failover groups on FWSM. The admin context is always a member of failover group 1, and any unassigned security contexts are also members of failover group 1 by default.

The failover group forms the base unit for failover in Active/Active failover. Interface failure monitoring, failover, and active/standby status are all attributes of a failover group rather than of the unit. The MAC address of the primary unit is used by all interfaces in the active contexts.

When an active failover group fails, it changes to the standby state while the associated standby failover group becomes active. The interfaces in the failover group that becomes active assume the MAC address and IP addresses of the interfaces in the failover group that failed. The interfaces in the failover group that is now in the standby state take over the standby MAC address and IP addresses.

Note A failover group failing on a unit does not mean that the unit has failed. The unit may still have another failover group passing traffic on it.

When creating the failover groups, you should create them on the unit that will have failover group 1 in the active state.

Primary/Secondary Status and Active/Standby Status

As in Active/Standby failover, one unit in an Active/Active failover pair is designated the primary unit, and the other unit the secondary unit. Unlike Active/Standby failover, this designation does not indicate which unit becomes active when both units start simultaneously. Instead, the primary/secondary designation determines which unit provides the running configuration to the pair and on which unit each failover group appears in the active state when both units start simultaneously.

Each failover group in the configuration is given a primary or secondary unit preference. This preference determines on which unit in the failover pair the contexts in the failover group appear in the active state when both units start simultaneously. You can have both failover groups be in the active state on a single unit in the pair, with the other unit containing the failover groups in the standby state. However, a more typical configuration is to assign each failover group a different role preference to make each one active on a different unit, balancing the traffic across the devices.

Note FWSM does not provide load balancing services. Load balancing must be handled by a router passing traffic to FWSM.

Device Initialization and Configuration Synchronization

Configuration synchronization occurs when one or both units in a failover pair boot.

When a unit boots while the peer unit is not available, then both failover groups become active on the unit regardless of the primary or secondary designation for the failover groups and the unit. Configuration synchronization does not occur. Some reasons a peer unit may not be available are that the peer unit is powered down, the peer unit is in a failed state, or the failover link between the units has not been established.

When a unit boots while the peer unit is active (with both failover groups active on it), the booting unit contacts the active unit to obtain the running configuration. By default, the failover groups will remain active on the active unit regardless of the primary or secondary preference of each failover group and unit designation (unless configured with the **preempt** command). The failover groups remain active on the first unit until one of the following occurs:

- A failover condition causes the failover group to become active on the peer unit.
- You manually force a failover group to become active on the peer unit using the **no failover active** command.
- The **preempt** command forces the failover group to become active on its preferred unit when that unit becomes available.

When both units boot at the same time, the primary unit becomes the active unit. The secondary unit obtains the running configuration from the primary unit. Once the configuration has been synchronized, each failover group becomes active on its preferred unit.

Note If you previously changed the number or size of memory partitions on the primary unit (see the "Managing Memory for Rules" section on page 4-11), then after the secondary unit synchronizes the configuration, immediately reload the secondary unit so that the memory partitions are the same. During the initial synchronization, the configuration might not fit properly in the secondary unit memory partitions, but after reloading, and another configuration synchronization, the secondary unit will be operational.

Command Replication

After both units are running, commands are replicated from one unit to the other as follows:

- Commands entered within a security context are replicated from the unit on which the security context appears in the active state to the peer unit.

Note A context is considered in the active state on a unit if the failover group to which it belongs is in the active state on that unit.

- Commands entered in the system execution space are replicated from the unit on which failover group 1 is in the active state to the unit on which failover group 1 is in the standby state.
- Commands entered in the admin context are replicated from the unit on which failover group 1 is in the active state to the unit on which failover group 1 is in the standby state.

Failure to enter the commands on the appropriate unit for command replication to occur will cause the configurations to become out of synchronization. Those changes may be lost the next time configuration synchronization occurs.

The following commands are replicated to the standby unit:

- all configuration commands except for the **mode** and **failover lan unit** commands
- **copy running-config startup-config**
- **delete**
- **mkdir**
- **rename**
- **rmdir**
- **write memory**

The following commands are not replicated to the standby unit:

- all forms of the **copy** command except for **copy running-config startup config**
- all forms of the **write** command except for **write memory**
- **asdm disconnect**
- **debug**
- **failover lan unit**
- **failover suspend-config-sync**
- **mode**
- **show**
- **ssh disconnect**

You can use the **write standby** command to resynchronize configurations that have become out of sync. For Active/Active failover, the **write standby** command behaves as follows:

- If you enter the **write standby** command in the system execution space, the system configuration and the configurations for all of the security contexts on FWSM is written to the peer unit. This includes configuration information for security contexts that are in the standby state. You must enter the command in the system execution space on the unit that has failover group 1 in the active state.
- If you enter the **write standby** command in a security context, only the configuration for the security context is written to the peer unit. You must enter the command in the security context on the unit where the security context appears in the active state.

Replicated commands are not saved to the Flash memory when replicated to the peer unit. They are added to the running configuration. To save replicated commands to Flash memory on both units, use the **write memory** or **copy running-config startup-config** command on the unit that you made the changes on. The command will be replicated to the peer unit and cause the configuration to be saved to Flash memory on the peer unit.

Failover Triggers

In Active/Active failover, failover can be triggered at the unit level if one of the following events occurs:

- The unit has a hardware failure.
- The unit has a power failure.
- The unit has a software failure.
- The **no failover active** or the **failover active** command is entered in the system execution space.

Failover is triggered at the failover group level when one of the following events occurs:

- Too many monitored interfaces in the contexts that belong to the failover group fail.
- The **no failover active group** *group_id* command is entered.

You configure the failover threshold for each failover group by specifying the number or percentage of interfaces within the failover group that must fail before the group fails. Because a failover group can contain multiple contexts, and each context can contain multiple interfaces, it is possible for all interfaces in a single context to fail without causing the associated failover group to fail.

See the "Failover Health Monitoring" section on page 13-19 for more information about interface and unit monitoring.

Failover Actions

In an Active/Active failover configuration, failover occurs on a failover group basis, not a system basis. For example, if you designate both failover groups as active on the primary unit, and failover group 1 fails, then failover group 2 remains active on the primary unit while failover group 1 becomes active on the secondary unit.

Note When configuring Active/Active failover, make sure that the combined traffic for both units is within the capacity of each unit.

Table 13-2 shows the failover action for each failure event. For each failure event, the policy (whether or not failover occurs), actions for the active failover group, and actions for the standby failover group are given.

Table 13-2 Failover Behavior for Active/Active Failover

Failure Event	Policy	Active Group Action	Standby Group Action	Notes
A unit experiences a power or software failure	Failover	Become standby Mark as failed	Become active Mark active as failed	When a unit in a failover pair fails, any active failover groups on that unit are marked as failed and become active on the peer unit.
Interface failure on active failover group above threshold	Failover	Mark active group as failed	Become active	None.
Interface failure on standby failover group above threshold	No failover	No action	Mark standby group as failed	When the standby failover group is marked as failed, then the active failover group will not attempt to fail over, even if the interface failure threshold is surpassed.
Formerly active failover group recovers	No failover	No action	No action	Unless configured with the **preempt** command, the failover groups remain active on their current unit.
Failover link failed at startup	No failover	Become active	Become active	If the failover link is down at startup, both failover groups on both units will become active.

Table 13-2 Failover Behavior for Active/Active Failover (continued)

Failure Event	Policy	Active Group Action	Standby Group Action	Notes
State link failed	No failover	No action	No action	State information will become out of date, and sessions will be terminated if a failover occurs.
Failover link failed during operation	No failover	n/a	n/a	Each unit marks the failover interface as failed. You should restore the failover link as soon as possible because the unit cannot fail over to the standby unit while the failover link is down.

Determining Which Type of Failover to Use

The type of failover you choose depends upon your FWSM configuration and how you plan to use FWSM.

If you are running FWSM in single mode, then you can only use Active/Standby failover; Active/Active failover is only available to FWSMs running in multiple context mode. If you are running the FWSM in multiple context mode, then you can configure either Active/Active failover or Active/Standby failover.

If you are using an upstream router to provide load balancing, use Active/Active failover. If you do not want to provide load balancing, use either Active/Standby or Active/Active failover.

Table 13-3 provides a comparison of some of the features supported by each type of failover configuration.

Table 13-3 Failover Configuration Feature Support

Feature	Active/Active	Active/Standby
Single Context Mode	No	Yes
Multiple Context Mode	Yes	Yes
Load Balancing Network Configurations	Yes	No
Unit Failover	Yes	Yes
Failover of Groups of Contexts	Yes	No
Failover of Individual Contexts	No	No

Regular and Stateful Failover

FWSM supports two types of failover, regular and stateful. This section includes the following topics:

- Regular Failover, page 13-18
- Stateful Failover, page 13-18

Regular Failover

When a failover occurs, all active connections are dropped. Clients need to reestablish connections when the new active unit takes over.

Stateful Failover

When Stateful Failover is enabled, the active unit continually passes per-connection state information to the standby unit. After a failover occurs, the same connection information is available at the new active unit. Supported end-user applications are not required to reconnect to keep the same communication session.

The state information passed to the standby unit includes the following:

- NAT translation table.
- TCP connection states.
- UDP connection states.
- The ARP table.
- The Layer 2 bridge table (when running in transparent firewall mode).
- The HTTP connection states (if HTTP replication is enabled).
- The ISAKMP and IPSec SA table.
- GTP PDP connection database.
- The user authentication (uauth) table (excluding state of inactivity timeout).

The information that is not passed to the standby unit when Stateful Failover is enabled includes the following:

- The HTTP connection table (unless HTTP replication is enabled).
- The routing tables.
- Multicast traffic information.

Note If failover occurs during an active Cisco IP SoftPhone session, the call will remain active because the call session state information is replicated to the standby unit. When the call is terminated, the IP SoftPhone client will lose connection with the CallManager. This occurs because there is no session information for the CTIQBE hangup message on the standby unit. When the IP SoftPhone client does not receive a response back from the CallManager within a certain time period, it considers the CallManager unreachable and unregisters itself.

OSPF databases and routing tables are not replicated by the HA process. In the event of a FWSM failure, allow time for the routing protocols to converge before traffic resumes flowing.

Note Because transparent FWSM relies on a Layer 2 MAC table for forwarding, the connection entry for a pair of hosts might still be active when the MAC table entries for one or both hosts have timed out due to inactivity. In such a situation, if a failover event occurs before either host sends another packet to re-populate the MAC address table, the peer FWSM is not able to generate switch CAM table refresh packets for the given endpoints. Therefore, if the CAM table entries on the switch for the given hosts are

still active and point to the formerly active unit, traffic is incorrectly switched to the standby FWSM and dropped there (if the idle connection starts passing traffic again after the failover event and before the CAM table entries age out on the switch).

Failover Health Monitoring

FWSM monitors each unit for overall health and for interface health. See the following sections for more information about how FWSM performs tests to determine the state of each unit:

- Unit Health Monitoring, page 13-19
- Interface Monitoring, page 13-19
- Rapid Link Failure Detection, page 13-20

Unit Health Monitoring

FWSM determines the health of the other unit by monitoring the failover link. When a unit does not receive hello messages on the failover link, then the unit sends an ARP request on all interfaces, including the failover interface. FWSM retries a user-configurable number of times. The action FWSM takes depends on the response from the other unit. See the following possible actions:

- If FWSM receives a response on any interface, then it does not fail over.
- If FWSM does not receive a response on any interface, then the standby unit switches to active mode and classifies the other unit as failed.
- If FWSM does not receive a response on the failover link only, then the unit does not failover. The failover link is marked as failed. You should restore the failover link as soon as possible because the unit cannot fail over to the standby while the failover link is down.

Note If a failed unit does not recover and you believe it should not be failed, you can reset the state by entering the **failover reset** command. If the failover condition persists, however, the unit will fail again.

Interface Monitoring

You can monitor up to 250 interfaces divided between all contexts. If an interface is shared among contexts, you can configure one context to monitor a shared interface. Because the interface is shared, all contexts benefit from the monitoring.

When a unit does not receive hello messages on a monitored interface, it runs the following tests:

1. Link Up/Down test—A test of the interface status. If the Link Up/Down test indicates that the interface is operational, then FWSM performs network tests. The purpose of these tests is to generate network traffic to determine which (if either) unit has failed. At the start of each test, each unit clears its received packet count for its interfaces. At the conclusion of each test, each unit looks to see if it has received any traffic. If it has, the interface is considered operational. If one unit receives traffic for a test and the other unit does not, the unit that received no traffic is considered failed. If neither unit has received traffic, then the next test is used.
2. Network Activity test—A received network activity test. The unit counts all received packets for up to 5 seconds. If any packets are received at any time during this interval, the interface is considered operational and testing stops. If no traffic is received, the ARP test begins.

3. ARP test—A reading of the unit ARP cache for the 2 most recently acquired entries. One at a time, the unit sends ARP requests to these machines, attempting to stimulate network traffic. After each request, the unit counts all received traffic for up to 5 seconds. If traffic is received, the interface is considered operational. If no traffic is received, an ARP request is sent to the next machine. If at the end of the list no traffic has been received, the ping test begins.
4. Broadcast Ping test—A ping test that consists of sending out a broadcast ping request. The unit then counts all received packets for up to 5 seconds. If any packets are received at any time during this interval, the interface is considered operational and testing stops.

If all network tests fail for an interface, but this interface on the other unit continues to successfully pass traffic, then the interface is considered to be failed. If the threshold for failed interfaces is met, then a failover occurs. If the other unit interface also fails all the network tests, then both interfaces go into the "Unknown" state and do not count towards the failover limit.

An interface becomes operational again if it receives any traffic. A failed FWSM returns to standby mode if the interface failure threshold is no longer met.

Note If a failed unit does not recover and you believe it should not be failed, you can reset the state by entering the **failover reset** command. If the failover condition persists, however, the unit will fail again.

Rapid Link Failure Detection

Detecting and responding to a failover condition can take up to 45 seconds. However, if you are using Catalyst operating system software Release 8.4(1) and higher or Cisco IOS software Release 12.2(18)SXF5 and higher on the switch, you can use the autostate feature to bypass the interface testing phase and provide sub-second failover times for interface failures.

With autostate enabled, the supervisor engine sends autostate messages to the FWSM about the status of physical interfaces associated with FWSM VLANs. For example, when all physical interfaces associated with a VLAN go down, the autostate message tells the FWSM that the VLAN is down. This information lets the FWSM declare the VLAN as down, bypassing the interface monitoring tests normally required for determining which side suffered a link failure.

In Cisco IOS software, autostate messaging is disabled by default. The Catalyst operating system software has autostate messaging enabled by default, and it is not configurable.

For more information about enabling autostate, see the "Enabling Autostate Messaging for Rapid Link Failure Detection" section on page 2-9.

Configuring Failover

This section describes how to configure failover and includes the following topics:

Failover Configuration Limitations

You cannot configure failover with the following type of IP addresses:

- IP addresses obtained through DHCP
- IPv6 addresses

Using Active/Standby Failover

This section provides step-by-step procedures for configuring Active/Standby failover. This section includes the following topics:

See the "Failover Example Configurations" section on page B-18 for examples of typical failover configurations.

Prerequisites

Before you begin, verify the following:

- Both units have the proper license.
- If the primary unit is in single context mode, the secondary unit must also be in single context mode and also be in the same firewall mode as the primary unit.
- If the primary unit is in multiple context mode, the secondary unit must also be in multiple context mode. You do not need configure the firewall mode of the security contexts on the secondary unit because the failover and state links reside in the system context. The secondary unit obtains the security context configuration from the primary unit.

Note The **mode** command does not get replicated to the secondary unit.

Configuring Active/Standby Failover

This section describes how to configure Active/Standby failover. You must configure the secondary unit to recognize the failover link before the secondary unit can obtain the running configuration from the primary unit.

This section includes the following topics:

Configuring the Primary Unit

Follow these steps to configure the primary unit in an Active/Standby failover configuration. These steps provide the minimum configuration needed to enable failover on the primary unit. For multiple context mode, all steps are performed in the system execution space unless otherwise noted.

To configure the primary unit in an Active/Standby failover pair, perform the following steps:

Step 1 If you have not done so already, configure the active and standby IP addresses for each interface (routed mode) or for the management address (transparent mode). The standby IP address is used on the FWSM that is currently the standby unit. It must be in the same subnet as the active IP address.

Note Do not configure an IP address for the failover link or for the state link (if you are going to use Stateful Failover).

```
hostname(config-if)# ip address active_addr netmask standby standby_addr
```

Note In multiple context mode, you must configure the interface addresses from within each context. Use the **changeto context** command to switch between contexts. The command prompt changes to `hostname/context(config-if)#`, where *context* is the name of the current context.

Step 2 Designate the unit as the primary unit:

```
hostname(config)# failover lan unit primary
```

Step 3 Define the failover interface.

a. Specify the interface to be used as the failover interface:

```
hostname(config)# failover lan interface if_name vlan vlan
```

The *if_name* argument assigns a name to the interface specified by the *vlan* argument.

b. Assign the active and standby IP address to the failover link:

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby address subnet mask.

The failover link IP address and MAC address do not change at failover. The active IP address for the failover link always stays with the primary unit, while the standby IP address stays with the secondary unit.

Step 4 (Optional) To enable Stateful Failover, configure the state link. The state link must be configured on an unused interface.

a. Specify the interface to be used as state link:

```
hostname(config)# failover link if_name [vlan vlan]
```

Note If the state link uses the failover link, then you only need to supply the *if_name* argument.

The *if_name* argument assigns a logical name to the interface specified by the *vlan* argument. This interface should not be used for any other purpose except, optionally, the failover link.

b. Assign an active and standby IP address to the state link.

Note If the state link uses the failover link, skip this step. You have already defined the failover link active and standby IP addresses.

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby address subnet mask.

The state link IP address and MAC address do not change at failover. The active IP address always stays with the primary unit, while the standby IP address stays with the secondary unit.

Step 5 To enable monitoring on an interface, enter the following command:

```
hostname(config)# monitor-interface interface_name
```

The maximum number of interfaces to monitor on the FWSM (divided between all contexts) is 250.

Note In multiple context mode, you must configure interface monitoring from within each context. Use the **changeto context** command to switch between contexts. The command prompt changes to `hostname/context(config)#`, where *context* is the name of the current context.

Step 6 Enable failover:

```
hostname(config)# failover
```

Step 7 Save the configuration:

```
hostname(config)# write memory
```

Note In multiple context mode, enter **write memory all** in the system execution space to save all context configurations.

Configuring the Secondary Unit

The only configuration required on the secondary unit is for the failover interface. The secondary unit requires these commands to initially communicate with the primary unit. After the primary unit sends its configuration to the secondary unit, the only permanent difference between the two configurations is the **failover lan unit** command, which identifies each unit as primary or secondary.

For multiple context mode, all steps are performed in the system execution space unless noted otherwise.

To configure the secondary unit, perform the following steps:

Step 1 Define the failover interface. Use the same settings as you used for the primary unit.

a. Specify the interface to be used as the failover interface:

```
hostname(config)# failover lan interface if_name vlan vlan
```

The *if_name* argument assigns a name to the interface specified by the *vlan* argument.

b. Assign the active and standby IP address to the failover link:

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

Note Enter this command exactly as you entered it on the primary unit when you configured the failover interface on the primary unit.

Step 2 (Optional) Designate this unit as the secondary unit:

```
hostname(config)# failover lan unit secondary
```

Note This step is optional because by default units are designated as secondary unless previously configured.

Step 3 Enable failover:

```
hostname(config)# failover
```

After you enable failover, the active unit sends the configuration in running memory to the standby unit. As the configuration synchronizes, the messages "Beginning configuration replication: Sending to mate" and "End Configuration Replication to mate" appear on the active unit console.

Step 4 After the running configuration has completed replication, save the configuration to Flash memory:

```
hostname(config)# write memory
```

Configuring Optional Active/Standby Failover Settings

You can configure the following optional Active/Standby failover setting when you are initially configuring failover or after failover has already been configured. Unless otherwise noted, the commands should be entered on the active unit.

This section includes the following topics:

Configuring Failover Preemption

When the primary unit in an Active/Standby failover configuration fails, or if the secondary unit boots before the primary unit, the secondary, standby unit becomes active. When the failover condition is resolved on the primary unit, it boots to the standby state by default and the secondary unit remains in the active state.

You can use the **failover preempt** command to cause the primary unit to become the active unit automatically after a specified amount of time. Enter the following command to configure preemption for the primary unit:

```
hostname(config)# failover preempt [delay]
```

The *delay* is the wait time, in seconds, before the secondary unit is preempted. Valid values are from 1 to 1200 seconds. If the *delay* is not specified, there is no delay.

When the primary unit becomes active, the secondary unit enters the standby state.

Enabling HTTP Replication with Stateful Failover

To allow HTTP connections to be included in the state information replication, you need to enable HTTP replication. Because HTTP connections are typically short-lived, and because HTTP clients typically retry failed connection attempts, HTTP connections are not automatically included in the replicated state information.

Enter the following command in global configuration mode to enable HTTP state replication when Stateful Failover is enabled:

```
hostname(config)# failover replication http
```

Configuring Interface and Unit Poll Times

FWSM monitors both unit and interface health for failover. You can configure the amount of time between hello messages when monitoring interface and unit health. Decreasing the poll time allows an interface or unit failure to be detected more quickly, but consumes more system resources.

To change the interface poll time, enter the following command in global configuration mode:

```
hostname(config)# failover polltime interface seconds
```

To change the unit poll time, enter the following command in global configuration mode:

```
hostname(config)# failover polltime seconds
```

To change the unit hold time, enter the following command in global configuration mode:

```
hostname(config)# failover holdtime seconds
```

The defaults are as follows:

- The interface **poll** *time* is 15 seconds.
- The unit **poll** *time* is 1 second.
- The **holdtime** time is 3 times the **poll** *time* (with a minimum value of 3 seconds) if you specify a **poll** *time* but do not specify a hold time with the **holdtime** keyword. If you specify a hold time using the **holdtime** keyword, it must be at least 3 times the **poll** *time*. If you enter the **clear configure failover** command, the hold time is 15 seconds.

Note You cannot enter a holdtime value that is less than 3 times the unit poll time. With a faster poll time, the FWSM can detect failure and trigger failover faster. However, faster detection can cause unnecessary switchovers when the network is temporarily congested.

Configuring Failover Criteria

By default, failure of 50% of monitored interfaces causes failover. You can specify a specific number of interfaces or a percentage of monitored interfaces that must fail before a failover occurs.

To change the default failover criteria, enter the following command in global configuration mode:

```
hostname(config)# failover interface-policy num[%]
```

When specifying a specific number of interfaces, the *num* argument can be from 1 to 250. When specifying a percentage of interfaces, the *num* argument can be from 1 to 100.

Using Active/Active Failover

This section describes how to configure Active/Active failover.

This section includes the following topics:

See the "Failover Example Configurations" section on page B-18 for examples of typical failover configurations.

Prerequisites

Before you begin, verify the following:

- Both units have the proper license.
- Both units are in multiple context mode. You do not need configure the firewall mode of the security contexts on the secondary unit because the failover and state links reside in the system context. The secondary unit obtains the security context configuration from the primary unit.

Note The **mode** command does not get replicated to the secondary unit.

Configuring Active/Active Failover

This section describes how to configure Active/Active failover. You must configure the secondary unit to recognize the failover link before the secondary unit can obtain the running configuration from the primary unit.

This section includes the following topics:

Configure the Primary Unit

To configure the primary unit in an Active/Active failover configuration, perform the following steps:

Step 1 If you have not done so already, configure the active and standby IP addresses for each interface (routed mode) or for the management address (transparent mode). The standby IP address is used on the FWSM that is currently the standby unit. It must be in the same subnet as the active IP address.

Note Do not configure an IP address for the failover link or for the state link (if you are going to use Stateful Failover).

```
hostname(config-if)# ip address active_addr netmask standby standby_addr
```

Note In multiple context mode, you must configure the interface addresses from within each context. Use the **changeto context** command to switch between contexts. The command prompt changes to `hostname/context(config-if)#`, where *context* is the name of the current context.

Step 2 Configure the basic failover parameters in the system execution space.

a. Designate the unit as the primary unit:

```
hostname(config)# failover lan unit primary
```

b. Specify the failover link:

```
hostname(config)# failover lan interface if_name vlan vlan
```

The *if_name* argument assigns a logical name to the interface specified by the *vlan* argument. This interface should not be used for any other purpose (except, optionally, the state link).

c. Specify the failover link active and standby IP addresses:

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby IP address subnet mask. The failover link IP address does not change at failover. The active IP address always stays with the primary unit, while the standby IP address stays with the secondary unit.

Step 3 (Optional) To enable Stateful Failover, configure the state link. The state link must be configured on an unused interface.

a. Specify the interface to be used as the state link:

```
hostname(config)# failover link if_name [vlan vlan]
```

The *if_name* argument assigns a logical name to the interface specified by the *vlan* argument. This interface should not be used for any other purpose (except, optionally, the failover link).

Note If the state link uses the failover link, then you only need to supply the *if_name* argument.

b. Assign an active and standby IP address to the state link.

Note If the state link uses the failover link, skip this step. You have already defined the failover link active and standby IP addresses.

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby address subnet mask.

The state link IP address does not change at failover. The active IP address always stays with the primary unit, while the standby IP address stays with the secondary unit.

Step 4 Configure the failover groups. You can have at most two failover groups. The **failover group** command creates the specified failover group if it does not exist and enters the failover group configuration mode.

For each failover group, you need to specify whether the failover group has primary or secondary preference using the **primary** or **secondary** command. You can assign the same preference to both failover groups. For load balancing configurations, you should assign each failover group a different unit preference.

The following example assigns failover group 1 a primary preference and failover group 2 a secondary preference:

```
hostname(config)# failover group 1
hostname(config-fover-group)# primary
hostname(config-fover-group)# exit
hostname(config)# failover group 2
hostname(config-fover-group)# secondary
hostname(config-fover-group)# exit
```

Step 5 Assign each context to a failover group using the **join-failover-group** command in context configuration mode.

Any unassigned contexts are automatically assigned to failover group 1. The admin context is always a member of failover group 1.

Enter the following commands to assign each context to a failover group:

```
hostname(config)# context context_name
hostname(config-context)# join-failover-group {1 | 2}
```

Step 6 Enable failover:

```
hostname(config)# failover
```

Step 7 To enable monitoring on an interface, change to the context and enter the following command:

```
hostname(config)# changeto context context_name
hostname(config)# monitor-interface interface_name
```

The maximum number of interfaces to monitor on the FWSM (divided between all contexts) is 250.

Configure the Secondary Unit

You need to configure the secondary unit to recognize the failover link. This allows the secondary unit to communicate with and receive the running configuration from the primary unit.

To configure the secondary unit in an Active/Active failover configuration, perform the following steps:

Step 1 Define the failover interface. Use the same settings as you used for the primary unit.

a. Specify the interface to be used as the failover interface:

```
hostname(config)# failover lan interface if_name vlan vlan
```

The *if_name* argument assigns a logical name to the interface specified by the *vlan* argument.

b. Assign the active and standby IP address to the failover link:

```
hostname(config)# failover interface ip if_name ip_addr mask standby ip_addr
```

Note Enter this command exactly as you entered it on the primary unit when you configured the failover interface.

The standby IP address must be in the same subnet as the active IP address. You do not need to identify the standby address subnet mask.

Step 2 (Optional) Designate this unit as the secondary unit:

```
hostname(config)# failover lan unit secondary
```

Note This step is optional because by default units are designated as secondary unless previously configured otherwise.

Step 3 Enable failover:

```
hostname(config)# failover
```

After you enable failover, the active unit sends the configuration in running memory to the standby unit. As the configuration synchronizes, the messages `Beginning configuration replication: Sending to mate` and `End Configuration Replication to mate` appear on the active unit console.

Step 4 After the running configuration has completed replication, enter the following command to save the configuration to Flash memory:

```
hostname(config)# write memory
```

Step 5 If necessary, force any failover group that is active on the primary to the active state on the secondary unit. To force a failover group to become active on the secondary unit, enter the following command in the system execution space on the primary unit:

```
hostname# no failover active group group_id
```

The *group_id* argument specifies the group you want to become active on the secondary unit.

Configuring Optional Active/Active Failover Settings

The following optional Active/Active failover settings can be configured when you are initially configuring failover or after you have already established failover. Unless otherwise noted, the commands should be entered on the unit that has failover group 1 in the active state.

This section includes the following topics:

Configuring Failover Group Preemption

Assigning a primary or secondary priority to a failover group specifies which unit the failover group becomes active on when both units boot simultaneously. However, if one unit boots before the other, then both failover groups become active on that unit. When the other unit comes online, any failover groups that have the unit as a priority do not become active on that unit unless manually forced over, a failover occurs, or the failover group is configured with the **preempt** command. The **preempt** command causes a failover group to become active on the designated unit automatically when that unit becomes available.

Enter the following commands to configure preemption for the specified failover group:

```
hostname(config)# failover group {1 | 2}
hostname(config-fover-group)# preempt [delay]
```

You can enter an optional *delay* value, which specifies the number of seconds the failover group remains active on the current unit before automatically becoming active on the designated unit.

Enabling HTTP Replication with Stateful Failover

To allow HTTP connections to be included in the state information, you need to enable HTTP replication. Because HTTP connections are typically short-lived, and because HTTP clients typically retry failed connection attempts, HTTP connections are not automatically included in the replicated state information. You can use the **replication http** command to cause a failover group to replicate HTTP state information when Stateful Failover is enabled.

To enable HTTP state replication for a failover group, enter the following command. This command only affects the failover group in which it was configured. To enable HTTP state replication for both failover groups, you must enter the following command in each group. This command should be entered in the system execution space.

```
hostname(config)# failover group {1 | 2}
hostname(config-fover-group)# replication http
```

Configuring Interface and Unit Poll Times

You can configure the amount of time between hello messages when monitoring the health of the interfaces in a failover group. Decreasing the interface poll time allows failover to occur faster when an interface fails, but consumes more system resources.

To change the default interface poll time, enter the following commands:

```
hostname(config)# failover group {1 | 2}
hostname(config-fover-group)# polltime interface seconds
```

The unit poll time specifies the amount of time between hello messages sent across the failover link to determine the health of the peer unit. Decreasing the unit poll time allows a failed unit to be detected faster, but consumes more system resources. To change the unit poll time, enter the following command in global configuration mode of the system execution space:

```
hostname(config)# failover polltime seconds
```

Configuring Failover Criteria

By default, failure of 50% of monitored interfaces causes failover. You can specify a specific number of interfaces or a percentage of monitored interfaces that must fail before a failover occurs. The failover criteria is specified on a failover group basis.

To change the default failover criteria for the specified failover group, enter the following commands:

```
hostname(config)# failover group {1 | 2}
hostname(config-fover-group)# interface-policy num[%]
```

When specifying a specific number of interfaces, the *num* argument can be from 1 to 250. When specifying a percentage of interfaces, the *num* argument can be from 1 to 100.

Configuring Failover Communication Authentication/Encryption

You can encrypt and authenticate the communication between failover peers by specifying a shared secret or hexadecimal key.

Caution All information sent over the failover and Stateful Failover links is sent in clear text unless you secure the communication with a failover key. If FWSM is used to terminate VPN tunnels, this information includes any usernames, passwords and preshared keys used for establishing the tunnels. Transmitting this sensitive data in clear text could pose a significant security risk. We recommend securing the failover communication with a failover key if you are using FWSM to terminate VPN tunnels.

Enter the following command on the active unit of an Active/Standby failover pair or on the unit that has failover group 1 in the active state of an Active/Active failover pair:

```
hostname(config)# failover key {secret | hex key}
```

The *secret* argument specifies a shared secret that is used to generate the encryption key. It can be from 1 to 63 characters. The characters can be any combination of numbers, letters, or punctuation. The **hex** *key* argument specifies a hexadecimal encryption key. The key must be 32 hexadecimal characters (0-9, a-f).

Note To prevent the failover key from being replicated to the peer unit in clear text for an existing failover configuration, disable failover on the active unit (or in the system execution space on the unit that has failover group 1 in the active state), enter the failover key on both units, and then reenable failover. When failover is reenabled, the failover communication will be encrypted with the key.

For new failover configurations, the **failover key** command should be part of the initial failover pair configuration.

Verifying the Failover Configuration

This section describes how to verify your failover configuration. This section includes the following topics:

Viewing Failover Status

This section describes how to view the failover status. On each unit you can verify the failover status by entering the **show failover** command. The information displayed depends upon whether you are using Active/Standby or Active/Active failover.

This section includes the following topics:

Viewing Failover Status for Active/Standby

The following is sample output from the **show failover** command for Active/Standby failover. Table 13-4 provides descriptions for the information shown.

```
hostname# show failover

Failover On
Failover unit Primary
Failover LAN Interface: fover Vlan 100(up)
Unit Poll frequency 1 seconds, holdtime 3 seconds
Interface Poll frequency 15 seconds
Interface Policy 1
Monitored Interfaces 2 of 250 maximum
failover replication http
Last Failover at: 22:44:03 UTC Dec 8 2004
        This host: Primary - Active
                Active time: 13434 (sec)
                Interface inside (10.130.9.3): Normal
                Interface outside (10.132.9.3): Normal
        Other host: Secondary - Standby Ready
                Active time: 0 (sec)
                Interface inside (10.130.9.4): Normal
                Interface outside (10.132.9.4): Normal

Stateful Failover Logical Update Statistics
        Link : fover Vlan100 (up)
        Stateful Obj    xmit       xerr       rcv        rerr
        General         1950       0          1733       0
        sys cmd         1733       0          1733       0
        up time         0          0          0          0
        RPC services    0          0          0          0
        TCP conn        6          0          0          0
        UDP conn        0          0          0          0
        ARP tbl         106        0          0          0
        Xlate_Timeout   0          0          0          0
        VPN IKE upd     15         0          0          0
        VPN IPSEC upd   90         0          0          0
        VPN CTCP upd    0          0          0          0
        VPN SDI upd     0          0          0          0
        VPN DHCP upd    0          0          0          0

        Logical Update Queue Information
                        Cur     Max     Total
        Recv Q:         0       2       1733
        Xmit Q:         0       2       15225
```

In multiple context mode, using the **show failover** command in a security context displays the failover information for that context. The information is similar to the information shown when using the command in single context mode. Instead of showing the active/standby status of the unit, it displays the active/standby status of the context. Table 13-4 provides descriptions for the information shown.

```
Failover On
Last Failover at: 04:03:11 UTC Jan 4 2003
        This context: Negotiation
                Active time: 1222 (sec)
                Interface outside (192.168.5.121): Normal
                Interface inside (192.168.0.1): Normal
        Peer context: Not Detected
                Active time: 0 (sec)
                Interface outside (192.168.5.131): Normal
                Interface inside (192.168.0.11): Normal
```

```
Stateful Failover Logical Update Statistics
        Status: Configured.
        Stateful Obj    xmit       xerr       rcv        rerr
        RPC services    0          0          0          0
        TCP conn        99          0          0          0
        UDP conn        0          0          0          0
        ARP tbl         22          0          0          0
        Xlate_Timeout   0          0          0          0
        GTP PDP         0          0          0          0
        GTP PDPMCB      0          0          0          0
```

Table 13-4 Show Failover Display Description

Field	Options
Failover	• On • Off
Failover Unit	Primary or Secondary.
Failover LAN Interface	Displays the name of the failover link.
Unit Poll frequency	Displays the number of seconds between hello messages sent to the peer unit and the number of seconds during which the unit must receive a hello message on the failover link before declaring the peer failed.
Interface Poll frequency	*n* seconds The number of seconds you set with the **failover polltime interface** command. The default is 15 seconds.
Interface Policy	Displays the number or percentage of interfaces that must fail to trigger failover.
Monitored Interfaces	Displays the number of interfaces monitored out of the maximum possible.
failover replication http	Displays if HTTP state replication is enabled for Stateful Failover.
Last Failover at:	The date and time of the last failover in the following form: *hh*:*mm*:*ss* **UTC** *DayName Month Day yyyy* UTC (Coordinated Universal Time) is equivalent to GMT (Greenwich Mean Time).
This host: Other host:	For each host, the display shows the following information.
Primary or Secondary	• Active • Standby
Active time:	*n* (sec) The amount of time the unit has been active. This time is cumulative, so the standby unit, if it was active in the past, will also show a value.

Table 13-4 Show Failover Display Description (continued)

Field	Options
Interface *name* (*n.n.n.n*):	For each interface, the display shows the IP address currently being used on each unit, as well as one of the following conditions: • Failed—The interface has failed. • No Link—The interface line protocol is down. • Normal—The interface is working correctly. • Link Down—The interface has been administratively shut down. • Unknown—FWSM cannot determine the status of the interface. • Waiting—Monitoring of the network interface on the other unit has not yet started.
Stateful Failover Logical Update Statistics	The following fields relate to the Stateful Failover feature. If the Link field shows an interface name, the Stateful Failover statistics are shown.
Link	• *interface_name*—The interface used for the Stateful Failover link. • Unconfigured—You are not using Stateful Failover. • up—The interface is up and functioning. • down—The interface is either administratively shutdown or is physically down. • failed—The interface has failed and is not passing stateful data.
Stateful Obj	For each field type, the following statistics are shown. They are counters for the number of state information packets sent between the two units; the fields do not necessarily show active connections through the unit. • xmit—Number of transmitted packets to the other unit. • xerr—Number of errors that occurred while transmitting packets to the other unit. • rcv—Number of received packets. • rerr—Number of errors that occurred while receiving packets from the other unit.
General	Sum of all stateful objects.
sys cmd	Logical update system commands; for example, LOGIN and Stay Alive.
up time	Up time, which the active unit passes to the standby unit.
RPC services	Remote Procedure Call connection information.
TCP conn	TCP connection information.
UDP conn	Dynamic UDP connection information.
ARP tbl	Dynamic ARP table information.
L2BRIDGE tbl	Layer 2 bridge table information (transparent firewall mode only).
Xlate_Timeout	Indicates connection translation timeout information.
VPN IKE upd	IKE connection information.

Table 13-4 Show Failover Display Description (continued)

Field	Options
VPN IPSEC upd	IPSec connection information.
VPN CTCP upd	cTCP tunnel connection information.
VPN SDI upd	SDI AAA connection information.
VPN DHCP upd	Tunneled DHCP connection information.
GTP PDP	GTP PDP update information. This information appears only if inspect GTP is enabled.
GTP PDPMCB	GTP PDPMCB update information. This information appears only if inspect GTP is enabled.
Logical Update Queue Information	For each field type, the following statistics are used: • Cur—Current number of packets • Max—Maximum number of packets • Total—Total number of packets
Recv Q	The status of the receive queue.
Xmit Q	The status of the transmit queue.

Viewing Failover Status for Active/Active

The following is sample output from the **show failover** command for Active/Active failover. Table 13-5 provides descriptions for the information shown.

```
hostname# show failover

Failover On
Failover unit Primary
Failover LAN Interface: fover Vlan 100 (up)
Unit Poll frequency 1 seconds, holdtime 15 seconds
Interface Poll frequency 4 seconds
Interface Policy 1
Monitored Interfaces 8 of 250 maximum
failover replication http
Group 1 last failover at: 13:40:18 UTC Dec 9 2004
Group 2 last failover at: 13:40:06 UTC Dec 9 2004

  This host:    Primary
  Group 1       State:          Active
                Active time:    2896 (sec)
  Group 2       State:          Standby Ready
                Active time:    0 (sec)

                admin Interface outside (10.132.8.5): Normal
                admin Interface third (10.132.9.5): Normal
                admin Interface inside (10.130.8.5): Normal
                admin Interface fourth (10.130.9.5): Normal
                ctx1 Interface outside (10.1.1.1): Normal
                ctx1 Interface inside (10.2.2.1): Normal
                ctx2 Interface outside (10.3.3.2): Normal
                ctx2 Interface inside (10.4.4.2): Normal

  Other host:   Secondary
  Group 1       State:          Standby Ready
                Active time:    190 (sec)
```

```
  Group 2       State:          Active
                Active time:    3322 (sec)

                admin Interface outside (10.132.8.6): Normal
                admin Interface third (10.132.9.6): Normal
                admin Interface inside (10.130.8.6): Normal
                admin Interface fourth (10.130.9.6): Normal
                ctx1 Interface outside (10.1.1.2): Normal
                ctx1 Interface inside (10.2.2.2): Normal
                ctx2 Interface outside (10.3.3.1): Normal
                ctx2 Interface inside (10.4.4.1): Normal

Stateful Failover Logical Update Statistics
        Link : fover Vlan100 (up)
        Stateful Obj    xmit       xerr       rcv        rerr
        General         1973       0          1895       0
        sys cmd         380        0          380        0
        up time         0          0          0          0
        RPC services    0          0          0          0
        TCP conn        1435       0          1450       0
        UDP conn        0          0          0          0
        ARP tbl         124        0          65         0
        Xlate_Timeout   0          0          0          0
        VPN IKE upd     15         0          0          0
        VPN IPSEC upd   90         0          0          0
        VPN CTCP upd    0          0          0          0
        VPN SDI upd     0          0          0          0
        VPN DHCP upd    0          0          0          0

        Logical Update Queue Information
                        Cur     Max     Total
        Recv Q:         0       1       1895
        Xmit Q:         0       0       1940
```

The following is sample output from the **show failover group** command for Active/Active failover. The information displayed is similar to that of the **show failover** command, but limited to the specified group. Table 13-5 provides descriptions for the information shown.

```
hostname# show failover group 1

  Last Failover at: 04:09:59 UTC Jan 4 2005

  This host:    Secondary
                State:          Active
                Active time:    186 (sec)

                admin Interface outside (192.168.5.121): Normal
                admin Interface inside (192.168.0.1): Normal


  Other host:   Primary
                State:          Standby
                Active time:    0 (sec)

                admin Interface outside (192.168.5.131): Normal
                admin Interface inside (192.168.0.11): Normal

  Stateful Failover Logical Update Statistics
        Status: Configured.
        RPC services    0          0          0          0
        TCP conn        33         0          0          0
        UDP conn        0          0          0          0
        ARP tbl         12         0          0          0
```

```
Xlate_Timeout   0           0           0           0
GTP PDP         0           0           0           0
GTP PDPMCB      0           0           0           0
```

Table 13-5 Show Failover Display Description

Field	Options
Failover	• On • Off
Failover Unit	Primary or Secondary.
Failover LAN Interface	Displays the name of the failover link.
Unit Poll frequency	Displays the number of seconds between hello messages sent to the peer unit and the number of seconds during which the unit must receive a hello message on the failover link before declaring the peer failed.
Interface Poll frequency	*n* seconds The number of seconds you set with the **failover polltime interface** command. The default is 15 seconds.
Interface Policy	Displays the number or percentage of interfaces that must fail before triggering failover.
Monitored Interfaces	Displays the number of interfaces monitored out of the maximum possible.
Group 1 Last Failover at: Group 2 Last Failover at:	The date and time of the last failover for each group in the following form: *hh*:*mm*:*ss* **UTC** *DayName Month Day yyyy* UTC (Coordinated Universal Time) is equivalent to GMT (Greenwich Mean Time).
This host: Other host:	For each host, the display shows the following information.
Role	Primary or Secondary
System State	• Active or Standby Ready • Active Time in seconds
Group 1 State Group 2 State	• Active or Standby Ready • Active Time in seconds
context Interface *name* (*n.n.n.n*):	For each interface, the display shows the IP address currently being used on each unit, as well as one of the following conditions: • Failed—The interface has failed. • No link—The interface line protocol is down. • Normal—The interface is working correctly. • Link Down—The interface has been administratively shut down. • Unknown—FWSM cannot determine the status of the interface. • Waiting—Monitoring of the network interface on the other unit has not yet started.

Table 13-5 Show Failover Display Description (continued)

Field	Options
Stateful Failover Logical Update Statistics	The following fields relate to the Stateful Failover feature. If the Link field shows an interface name, the Stateful Failover statistics are shown.
Link	• *interface_name*—The interface used for the Stateful Failover link. • Unconfigured—You are not using Stateful Failover. • up—The interface is up and functioning. • down—The interface is either administratively shutdown or is physically down. • failed—The interface has failed and is not passing stateful data.
Stateful Obj	For each field type, the following statistics are used. They are counters for the number of state information packets sent between the two units; the fields do not necessarily show active connections through the unit. • xmit—Number of transmitted packets to the other unit • xerr—Number of errors that occurred while transmitting packets to the other unit • rcv—Number of received packets • rerr—Number of errors that occurred while receiving packets from the other unit
General	Sum of all stateful objects.
sys cmd	Logical update system commands; for example, LOGIN and Stay Alive.
up time	Up time, which the active unit passes to the standby unit.
RPC services	Remote Procedure Call connection information.
TCP conn	TCP connection information.
UDP conn	Dynamic UDP connection information.
ARP tbl	Dynamic ARP table information.
L2BRIDGE tbl	Layer 2 bridge table information (transparent firewall mode only).
Xlate_Timeout	Indicates connection translation timeout information.
VPN IKE upd	IKE connection information.
VPN IPSEC upd	IPSec connection information.
VPN CTCP upd	cTCP tunnel connection information.
VPN SDI upd	SDI AAA connection information.
VPN DHCP upd	Tunneled DHCP connection information.
GTP PDP	GTP PDP update information. This information appears only if inspect GTP is enabled.
GTP PDPMCB	GTP PDPMCB update information. This information appears only if inspect GTP is enabled.

Table 13-5 Show Failover Display Description (continued)

Field	Options
Logical Update Queue Information	For each field type, the following statistics are used: • Cur—Current number of packets • Max—Maximum number of packets • Total—Total number of packets
Recv Q	The status of the receive queue.
Xmit Q	The status of the transmit queue.

Viewing Monitored Interfaces

To view the status of monitored interfaces, enter the following command. In single context mode, enter this command in global configuration mode. In multiple context mode, enter this command within a context.

```
primary/context(config)# show monitor-interface
```

For example:

```
hostname/context(config)# show monitor-interface
        This host: Primary - Active
                Interface outside (192.168.1.2): Normal
                Interface inside (10.1.1.91): Normal
        Other host: Secondary - Standby
                Interface outside (192.168.1.3): Normal
                Interface inside (10.1.1.100): Normal
```

Viewing the Failover Configuration

To view the failover commands in the running configuration, enter the following command:

```
hostname(config)# show running-config failover
```

All of the failover commands are displayed. On units running multiple context mode, enter this command in the system execution space. Entering **show running-config all failover** displays the failover commands in the running configuration and includes commands for which you have not changed the default value.

Testing the Failover Functionality

To test failover functionality, perform the following steps:

Step 1 Test that your active unit or failover group is passing traffic as expected by using FTP (for example) to send a file between hosts on different interfaces.

Step 2 Force a failover to the standby unit by entering the following command:

- For Active/Standby failover, enter the following command on the active unit:

  ```
  hostname(config)# no failover active
  ```

- For Active/Active failover, enter the following command on the unit where failover group containing the interface connecting your hosts is active:

```
hostname(config)# no failover active group group_id
```

Step 3 Use FTP to send another file between the same two hosts.

Step 4 If the test was not successful, enter the **show failover** command to check the failover status.

Step 5 When you are finished, you can restore the unit or failover group to active status by enter the following command:

- For Active/Standby failover, enter the following command on the active unit:

  ```
  hostname(config)# failover active
  ```

- For Active/Active failover, enter the following command on the unit where the failover group containing the interface connecting your hosts is active:

  ```
  hostname(config)# failover active group group_id
  ```

Controlling and Monitoring Failover

This section describes how to control and monitor failover. This section includes the following topics:

Forcing Failover

To force the standby unit or failover group to become active, enter one of the following commands:

- For Active/Standby failover:

  Enter the following command on the standby unit:

  ```
  hostname# failover active
  ```

  Or enter the following command on the active unit:

  ```
  hostname# no failover active
  ```

- For Active/Active failover:

  Enter the following command in the system execution space of the unit where failover group is in the standby state:

  ```
  hostname# failover active group group_id
  ```

  Or, enter the following command in the system execution space of the unit where the failover group is in the active state:

  ```
  hostname# no failover active group group_id
  ```

Entering the following command in the system execution space causes all failover groups to become active:

```
hostname# failover active
```

Disabling Failover

To disable failover, enter the following command:

```
hostname(config)# no failover
```

Disabling failover on an Active/Standby pair causes the active and standby state of each unit to be maintained until you restart. For example, the standby unit remains in standby mode so that both units do not start passing traffic. To make the standby unit active (even with failover disabled), see the "Forcing Failover" section on page 13-40.

Disabling failover on an Active/Active pair causes the failover groups to remain in the active state on whichever unit they are currently active on, no matter which unit they are configured to prefer. The **no failover** command should be entered in the system execution space.

Disabling Configuration Synchronization

Management applications may lose connectivity when upgrading the FWSM with complex configurations. This can result in incomplete configuration files being applied to the standby FWSM. You can disable the automatic configuration synchronization to avoid incomplete configurations being applied to the standby FWSM. You need to disable configuration synchronization when upgrading a software image or changing the configuration on the active FWSM to verify that the configuration files are complete before the configuration is synchronized with the standby FWSM configuration. After you verify that the configuration is complete, reenable configuration synchronization.

To disable configuration synchronization, enter the following command:

```
hostname(config)# failover suspend-config-sync
```

To reenable configuration synchronization, use the **no** form of the this command.

Restoring a Failed Unit or Failover Group

To restore a failed unit to an unfailed state, enter the following command:

```
hostname(config)# failover reset
```

To restore a failed Active/Active failover group to an unfailed state, enter the following command:

```
hostname(config)# failover reset group group_id
```

Restoring a failed unit or group to an unfailed state does not automatically make it active; restored units or groups remain in the standby state until made active by failover (forced or natural). An exception is a failover group configured with the **preempt** command. If previously active, a failover group will become active if it is configured with the **preempt** command and if the unit on which it failed is its preferred unit.

Monitoring Failover

When a failover occurs, both FWSMs send out system log messages. This section includes the following topics:

Failover System Log Messages

FWSM issues a number of system log messages related to failover at priority level 2, which indicates a critical condition. To view these messages, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module System Log Messages* to enable logging and to see descriptions of the system log messages.

Note During switchover, failover will logically shut down and then bring up interfaces, generating system log messages 411001 and 411002. This is normal activity.

Debug Messages

To see debug messages, enter the **debug fover** command. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

Note Because debugging output is assigned high priority in the CPU process, it can drastically affect system performance. For this reason, use the **debug fover** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC.

SNMP

To receive SNMP syslog traps for failover, configure the SNMP agent to send SNMP traps to SNMP management stations, define a syslog host, and compile the Cisco syslog MIB into your SNMP management station. See the **snmp-server** and **logging** commands in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

PART 2

Configuring the Security Policy

CHAPTER 14

Permitting or Denying Network Access

This chapter describes how to control network access through the FWSM using access lists. To create an extended access lists or an EtherType access list, see Chapter 12, "Identifying Traffic with Access Lists."

Note You use access lists to control network access in both routed and transparent firewall modes. In transparent mode, you can use both extended access lists (for Layer 3 traffic) and EtherType access lists (for Layer 2 traffic).

To access the FWSM interface for management access, you do not also need an access list allowing the host IP address. You only need to configure management access according to Chapter 22, "Configuring Management Access."

This chapter includes the following sections:

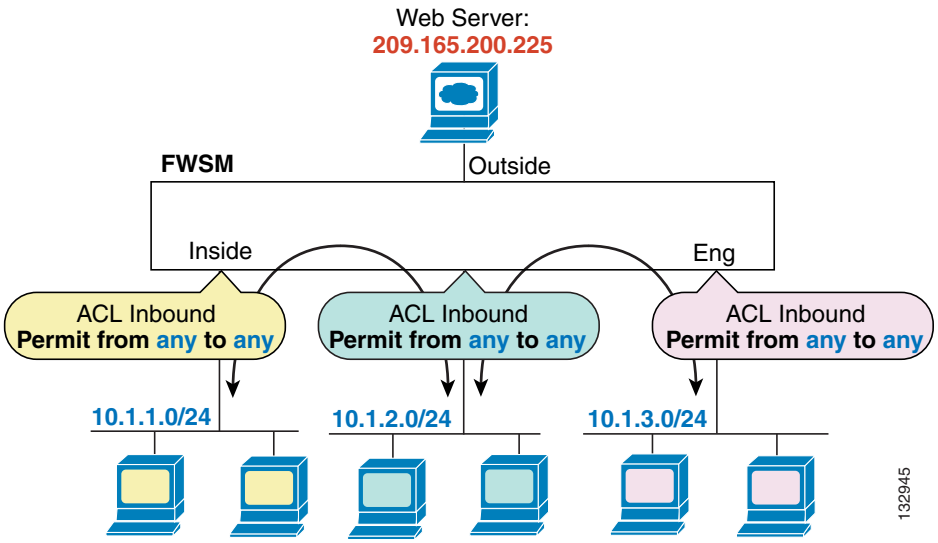
Inbound and Outbound Access List Overview

Traffic flowing across an interface in the FWSM can be controlled in two ways. Traffic that enters the FWSM can be controlled by attaching an inbound access list to the source interface. Traffic that exits the FWSM can be controlled by attaching an outbound access list to the destination interface. To allow any traffic to enter the FWSM, you must attach an inbound access list to an interface; otherwise, the FWSM automatically drops all traffic that enters that interface. By default, traffic can exit the FWSM on any interface unless you restrict it using an outbound access list, which adds restrictions to those already configured in the inbound access list.

Note "Inbound" and "outbound" refer to the application of an access list on an interface, either to traffic entering the FWSM on an interface or traffic exiting the FWSM on an interface. These terms do not refer to the movement of traffic from a lower security interface to a higher security interface, commonly known as inbound, or from a higher to lower interface, commonly known as outbound.

You might want to use an outbound access list to simplify your access list configuration. For example, if you want to allow three inside networks on three different interfaces to access each other, you can create a simple inbound access list that allows all traffic on each inside interface (see Figure 14-1).

Figure 14-1 Inbound Access Lists

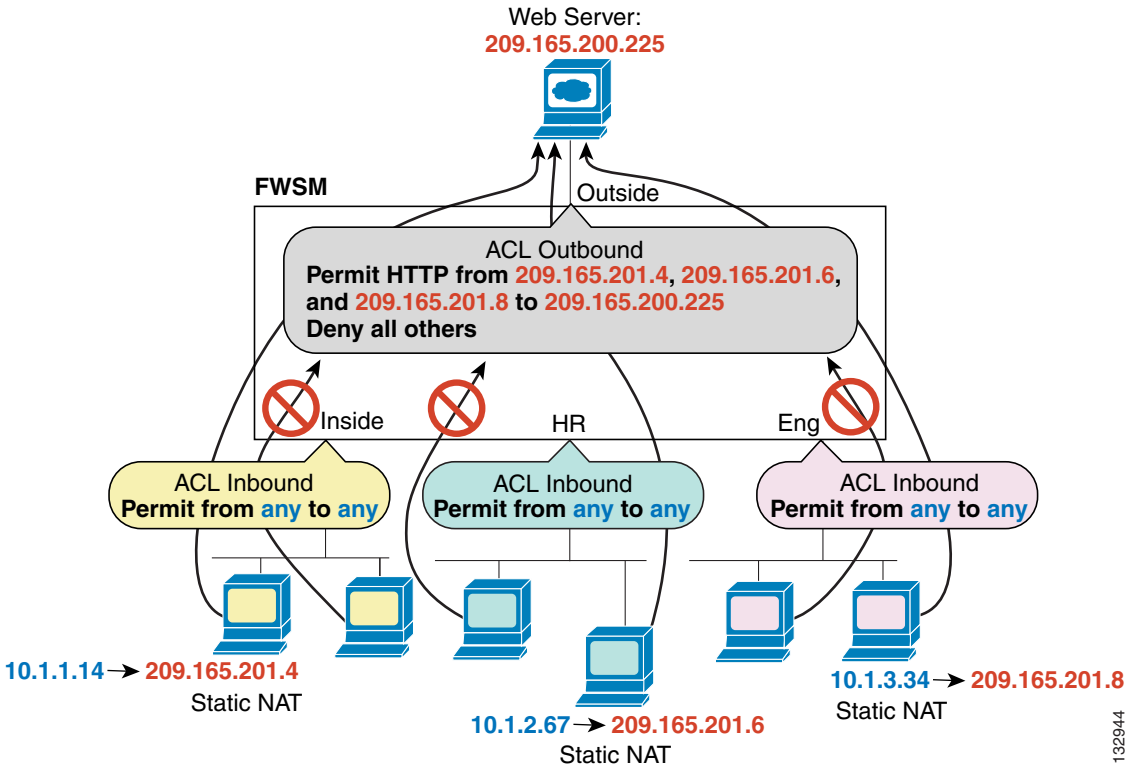


See the following commands for this example:

```
hostname(config)# access-list INSIDE extended permit ip any any
hostname(config)# access-group INSIDE in interface inside

hostname(config)# access-list HR extended permit ip any any
hostname(config)# access-group HR in interface hr

hostname(config)# access-list ENG extended permit ip any any
hostname(config)# access-group ENG in interface eng
```

Then, if you want to allow only certain hosts on the inside networks to access a web server on the outside network, you can create a more restrictive access list that allows only the specified hosts and apply it to the outbound direction of the outside interface (see Figure 14-1). See the "IP Addresses Used for Access Lists When You Use NAT" section on page 12-3 for information about NAT and IP addresses. The outbound access list prevents any other hosts from reaching the outside network.

Figure 14-2 Outbound Access List



See the following commands for this example:

```
hostname(config)# access-list INSIDE extended permit ip any any
hostname(config)# access-group INSIDE in interface inside

hostname(config)# access-list HR extended permit ip any any
hostname(config)# access-group HR in interface hr

hostname(config)# access-list ENG extended permit ip any any
hostname(config)# access-group ENG in interface eng

hostname(config)# access-list OUTSIDE extended permit tcp host 209.165.201.4
host 209.165.200.225 eq www
hostname(config)# access-list OUTSIDE extended permit tcp host 209.165.201.6
host 209.165.200.225 eq www
hostname(config)# access-list OUTSIDE extended permit tcp host 209.165.201.8
host 209.165.200.225 eq www
hostname(config)# access-group OUTSIDE out interface outside
```

Applying an Access List to an Interface

To apply an extended access list to the inbound or outbound direction of an interface, enter the following command:

```
hostname(config)# access-group access_list_name {in | out} interface interface_name
[per-user-override]
```

You can apply one access list of each type (extended and EtherType) to both directions of the interface. See the "Inbound and Outbound Access List Overview" section on page 14-1 for more information about access list directions.

The **per-user-override** keyword allows dynamic access lists that are downloaded for user authorization to override the access list assigned to the interface. For example, if the interface access list denies all traffic from 10.0.0.0, but the dynamic access list permits all traffic from 10.0.0.0, then the dynamic access list overrides the interface access list for that user. See the "Configuring RADIUS Authorization" section for more information about per-user access lists. The **per-user-override** keyword is only available for inbound access lists.

For connectionless protocols, you need to apply the access list to the source and destination interfaces if you want traffic to pass in both directions. For example, you can allow BGP in an EtherType access list in transparent mode, and you need to apply the access list to both interfaces.

The following example illustrates the commands required to enable access to an inside web server with the IP address 209.165.201.12 (this IP address is the address visible on the outside interface after NAT):

```
hostname(config)# access-list ACL_OUT extended permit tcp any host 209.165.201.12 eq www
hostname(config)# access-group ACL_OUT in interface outside
```

You also need to configure NAT for the web server.

The following access lists allow all hosts to communicate between the inside and hr networks, but only specific hosts to access the outside network:

```
hostname(config)# access-list ANY extended permit ip any any
hostname(config)# access-list OUT extended permit ip host 209.168.200.3 any
hostname(config)# access-list OUT extended permit ip host 209.168.200.4 any

hostname(config)# access-group ANY in interface inside
hostname(config)# access-group ANY in interface hr
hostname(config)# access-group OUT out interface outside
```

For example, the following sample access list allows common EtherTypes originating on the inside interface:

```
hostname(config)# access-list ETHER ethertype permit ipx
hostname(config)# access-list ETHER ethertype permit bpdu
hostname(config)# access-list ETHER ethertype permit mpls-unicast
hostname(config)# access-group ETHER in interface inside
```

The following access list allows some EtherTypes through the FWSM, but denies all others:

```
hostname(config)# access-list ETHER ethertype permit 0x1234
hostname(config)# access-list ETHER ethertype permit bpdu
hostname(config)# access-list ETHER ethertype permit mpls-unicast
hostname(config)# access-group ETHER in interface inside
hostname(config)# access-group ETHER in interface outside
```

The following access list denies traffic with EtherType 0x1256 but allows all others on both interfaces:

```
hostname(config)# access-list nonIP ethertype deny 1256
hostname(config)# access-list nonIP ethertype permit any
hostname(config)# access-group ETHER in interface inside
```

```
hostname(config)# access-group ETHER in interface outside
```

CHAPTER 15

Configuring NAT

This chapter describes Network Address Translation, and includes the following sections:

NAT Overview

This section describes how NAT works on the FWSM, and includes the following topics:

Introduction to NAT

Address translation substitutes the real address in a packet with a mapped address that is routable on the destination network. NAT is comprised of two steps: the process in which a real address is translated into a mapped address, and then the process to undo translation for returning traffic. NAT is supported in both routed and transparent firewall mode.

The FWSM translates an address when a NAT rule matches the traffic. If no NAT rule matches, processing for the packet continues. The exception is when you enable NAT control. NAT control requires that packets traversing from a higher security interface (inside) to a lower security interface (outside) match a NAT rule, or else processing for the packet stops. (See the "Security Level Overview" section on page 6-1 for more information about security levels, and see the "NAT Control" section on page 15-5 for more information about NAT control).

Note In this document, all types of translation are generally referred to as NAT. When discussing NAT, the terms *inside* and *outside* are relative, and represent the security relationship between any two interfaces. The higher security level is inside and the lower security level is outside; for example, interface 1 is at 60 and interface 2 is at 50, so interface 1 is "inside" and interface 2 is "outside."

Some of the benefits of NAT are as follows:

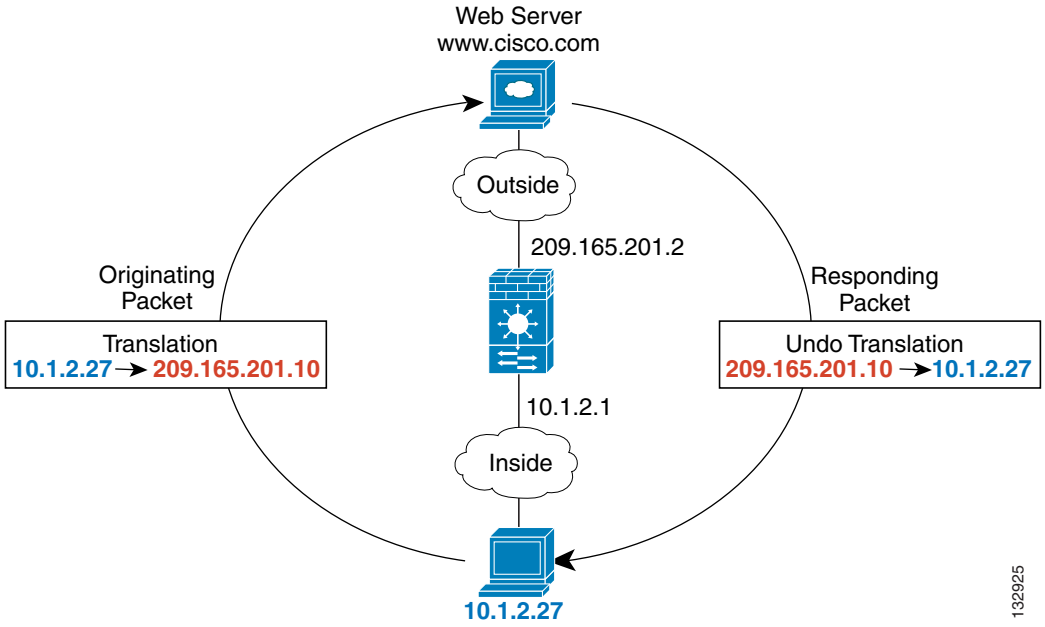
- You can use private addresses on your inside networks. Private addresses are not routable on the Internet. (See the "Private Networks" section on page E-2 for more information.)
- NAT hides the real addresses from other networks, so attackers cannot learn the real address of a host.
- You can resolve IP routing problems such as overlapping addresses.

Note See Table 21-1 on page 21-4 for information about protocols that do not support NAT.

NAT in Routed Mode

Figure 15-1 shows a typical NAT scenario in routed mode, with a private network on the inside. When the inside host at 10.1.1.27 sends a packet to a web server, the real source address, 10.1.1.27, of the packet is changed to a mapped address, 209.165.201.10. When the server responds, it sends the response to the mapped address, 209.165.201.10, and the FWSM receives the packet. The FWSM then undoes the translation of the mapped address, 209.165.201.10 back to the real address, 10.1.1.1.27 before sending it on to the host.

Figure 15-1 NAT Example: Routed Mode



See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.1-209.165.201.15
```

NAT in Transparent Mode

Using NAT in transparent mode eliminates the need for the upstream or downstream routers to perform NAT for their networks. For example, a transparent firewall FWSM is useful between two VRFs so you can establish BGP neighbor relations between the VRFs and the global table. However, NAT per VRF might not be supported. In this case, using NAT in transparent mode is essential.

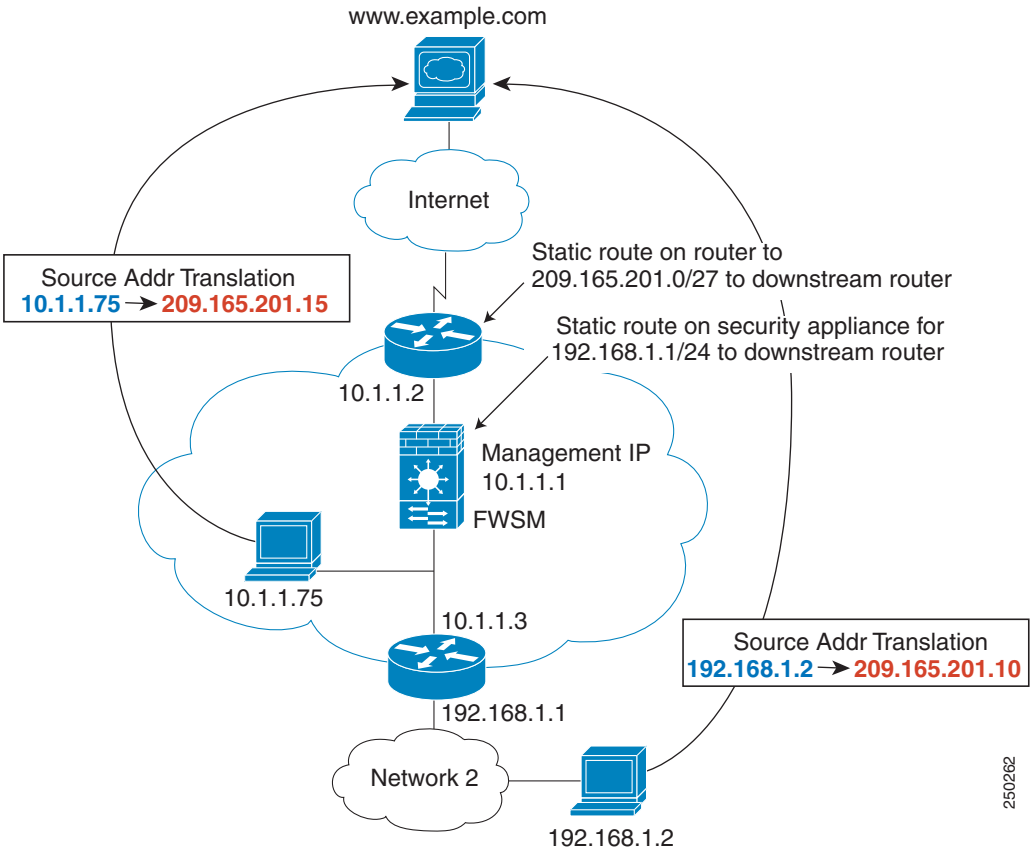
NAT in transparent mode has the following requirements and limitations:

- When the mapped addresses are not on the same network as the transparent firewall, then on the upstream router, you need to add a static route for the mapped addresses that points to the downstream router (through the FWSM).
- If the real destination address is not directly-connected to the FWSM, then you also need to add a static route on the FWSM for the real destination address that points to the downstream router. Without NAT, traffic from the upstream router to the downstream router does not need any routes on the FWSM because it uses the MAC address table. NAT, however, causes the FWSM to use a route lookup instead of a MAC address lookup, so it needs a static route to the downstream router.
- The **alias** command is not supported.
- Because the transparent firewall does not have any interface IP addresses, you cannot use interface PAT.
- ARP inspection is not supported. Moreover, if for some reason a host on one side of the firewall sends an ARP request to a host on the other side of the firewall, and the initiating host real address is mapped to a different address on the same subnet, then the real address remains visible in the ARP request.

- For ICMP support, you must enable ICMP inspection.

Figure 15-2 shows a typical NAT scenario in transparent mode, with the same network on the inside and outside interfaces. The transparent firewall in this scenario is performing the NAT service so that the upstream router does not have to perform NAT.

Figure 15-2 NAT Example: Transparent Mode



1. When the inside host at 10.1.1.75 sends a packet to a web server, the real source address of the packet, 10.1.1.75, is changed to a mapped address, 209.165.201.15.
2. When the server responds, it sends the response to the mapped address, 209.165.201.15, and the FWSM receives the packet because the upstream router includes this mapped network in a static route that is directed through the FWSM.
3. The FWSM then undoes the translation of the mapped address, 209.165.201.15, back to the real address, 10.1.1.1.75. Because the real address is directly-connected, the FWSM sends it directly to the host.
4. For host 192.168.1.2, the same process occurs, except that the FWSM looks up the route in its route table, and sends the packet to the downstream router at 10.1.1.3 based on the static route.

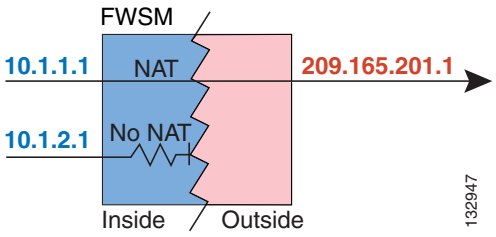
See the following commands for this example:

```
hostname(config)# route inside 192.168.1.0 255.255.255.0 10.1.1.3 1
hostname(config)# nat (inside) 1 10.1.1.0 255.255.255.0
hostname(config)# nat (inside) 1 192.168.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.1-209.165.201.15
```

NAT Control

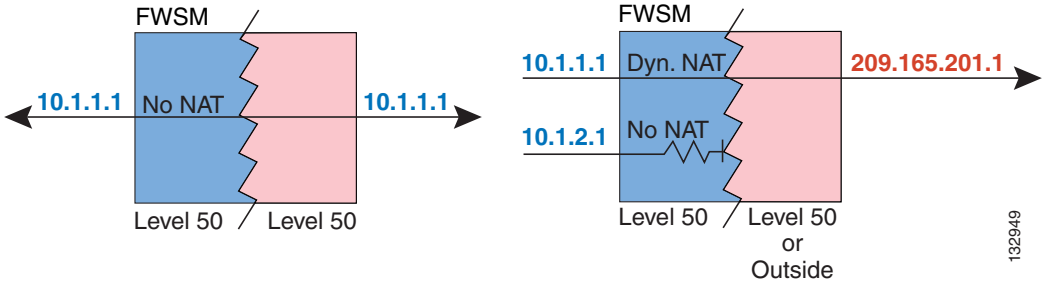
NAT control requires that packets traversing from an inside interface to an outside interface match a NAT rule; for any host on the inside network to access a host on the outside network, you must configure NAT to translate the inside host address (see Figure 15-3).

Figure 15-3 NAT Control and Outbound Traffic



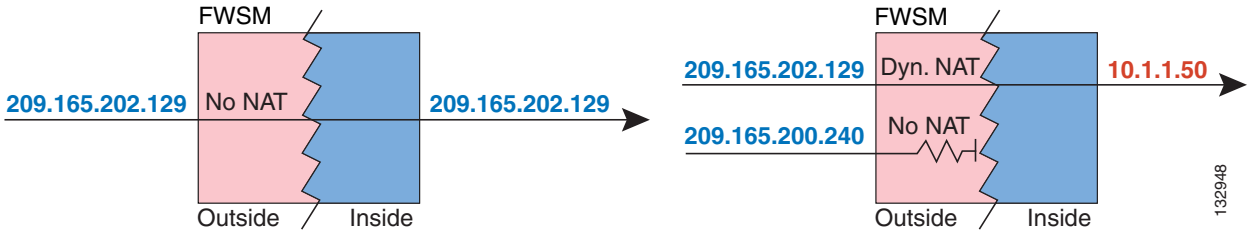
Interfaces at the same security level are not required to use NAT to communicate. However, if you configure dynamic NAT or PAT on a same security interface with NAT control enabled, then all traffic from the interface to a same security interface or an outside interface must match a NAT rule (see Figure 15-4).

Figure 15-4 NAT Control and Same Security Traffic



Similarly, if you enable outside dynamic NAT or PAT with NAT control, then all outside traffic must match a NAT rule when it accesses an inside interface (see Figure 15-5).

Figure 15-5 NAT Control and Inbound Traffic



Static NAT with NAT control does not cause these restrictions.

By default, NAT control is disabled, so you do not need to perform NAT on any networks unless you choose to perform NAT. If you upgraded from an earlier version of software, however, NAT control might be enabled on your system.

If you want the added security of NAT control but do not want to translate inside addresses in some cases, you can apply a NAT exemption or identity NAT rule on those addresses. (See the "Bypassing NAT" section on page 15-32 for more information).

To configure NAT control, see the "Configuring NAT Control" section on page 15-17.

Note In multiple context mode, the packet classifier relies on the NAT configuration in some cases to assign packets to contexts. If you do not perform NAT because NAT control is disabled, then the classifier might require changes in your network configuration. See the "How the FWSM Classifies Packets" section on page 4-3 for more information about the relationship between the classifier and NAT.

NAT Types

This section describes the available NAT types. You can implement address translation as dynamic NAT, Port Address Translation (PAT; also known as NAT overloading) , static NAT, or static PAT or as a mix of these types. You can also configure rules to bypass NAT, for example, if you enable NAT control but do not want to perform NAT. This section includes the following topics:

Dynamic NAT

Dynamic NAT translates a group of real addresses to a pool of mapped addresses that are routable on the destination network. The mapped pool can include fewer addresses than the real group. When a host you want to translate accesses the destination network, the FWSM assigns it an IP address from the mapped pool. The translation is added only when the real host initiates the connection. The translation is in place only for the duration of the connection, and a given user does not keep the same IP address after the translation times out (see the **timeout xlate** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*). Users on the destination network, therefore, cannot reliably initiate a connection to a host that uses dynamic NAT (even if the connection is allowed by an access list), and the FWSM rejects any attempt to connect to a real host address directly. See the following "Static NAT" or "Static PAT" sections for reliable access to hosts.

Figure 15-6 shows a remote host attempting to connect to the real address. The connection is denied because the FWSM only allows returning connections to the mapped address.

Figure 15-6 Remote Host Attempts to Connect to the Real Address

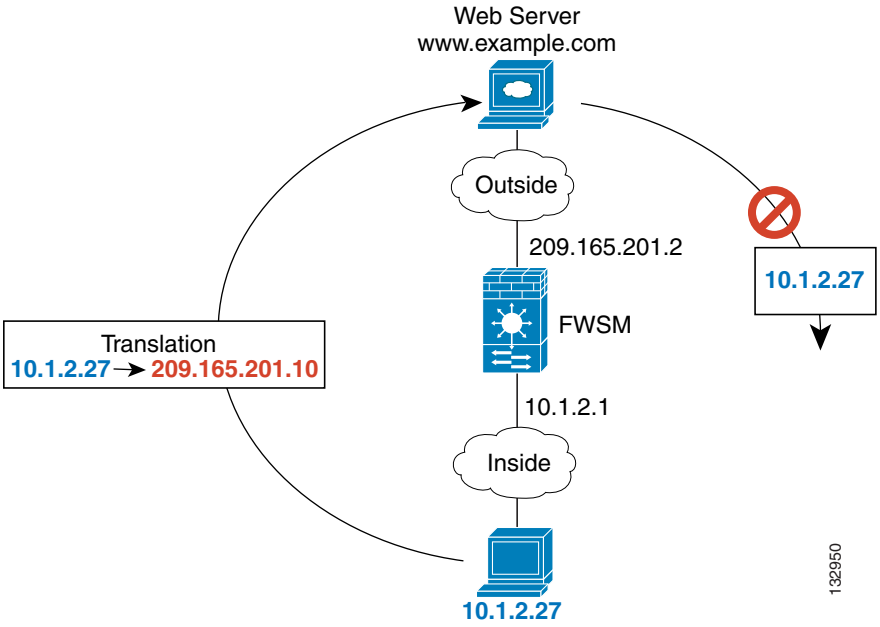
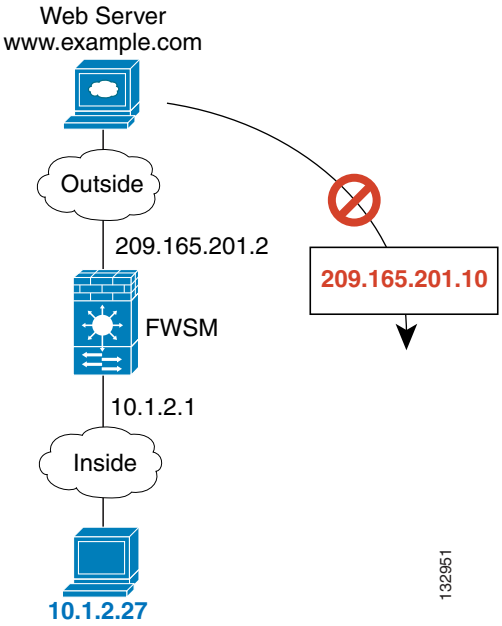


Figure 15-7 shows a remote host attempting to initiate a connection to a mapped address. This address is not currently in the translation table, so the FWSM drops the packet.

Figure 15-7 Remote Host Attempts to Initiate a Connection to a Mapped Address



Note For the duration of the translation, a remote host can initiate a connection to the translated host if an access list allows it. Because the address is unpredictable, a connection to the host is unlikely. However in this case, you can rely on the security of the access list.

Dynamic NAT has these disadvantages:

- If the mapped pool has fewer addresses than the real group, you could run out of addresses if the amount of traffic is more than expected.

  Use PAT if this event occurs often, because PAT provides over 64,000 translations using ports of a single address.

- You have to use a large number of routable addresses in the mapped pool; if the destination network requires registered addresses, such as the Internet, you might encounter a shortage of usable addresses.

The advantage of dynamic NAT is that some protocols cannot use PAT. For example, PAT does not work with IP protocols that do not have a port to overload, such as GRE version 0. PAT also does not work with some applications that have a data stream on one port and the control path on another and are not open standard, such as some multimedia applications. See the "Inspection Engine Overview" section on page 21-2 for more information about NAT and PAT support.

PAT

PAT (also known as NAT overloading) translates multiple real addresses to a single mapped IP address. Specifically, the FWSM translates the real address and source port (real socket) to the mapped address and a unique port above 1024 (mapped socket). Each connection requires a separate translation, because the source port differs for each connection. For example, 10.1.1.1:1025 requires a separate translation from 10.1.1.1:1026.

After the connection expires, the port translation also expires after 30 seconds of inactivity. The timeout is not configurable. Users on the destination network cannot reliably initiate a connection to a host that uses PAT (even if the connection is allowed by an access list). Not only can you not predict the real or mapped port number of the host, but the FWSM does not create a translation at all unless the translated host is the initiator. See the following "Static NAT" or "Static PAT" sections for reliable access to hosts.

PAT lets you use a single mapped address, thus conserving routable addresses. You can even use the FWSM interface IP address as the PAT address. PAT does not work with some multimedia applications that have a data stream that is different from the control path. See the "Inspection Engine Overview" section on page 21-2 for more information about NAT and PAT support.

Note For the duration of the translation, a remote host can initiate a connection to the translated host if an access list allows it. Because the port address (both real and mapped) is unpredictable, a connection to the host is unlikely. However in this case, you can rely on the security of the access list.

Static NAT

Static NAT creates a fixed translation of real address(es) to mapped address(es).With dynamic NAT and PAT, each host uses a different address or port for each subsequent translation. Because the mapped address is the same for each consecutive connection with static NAT, and a persistent translation rule exists, static NAT allows hosts on the destination network to initiate traffic to a translated host (if there is an access list that allows it).

The main difference between dynamic NAT and a range of addresses for static NAT is that static NAT allows a remote host to initiate a connection to a translated host (if there is an access list that allows it), while dynamic NAT does not. You also need an equal number of mapped addresses as real addresses with static NAT.

Static PAT

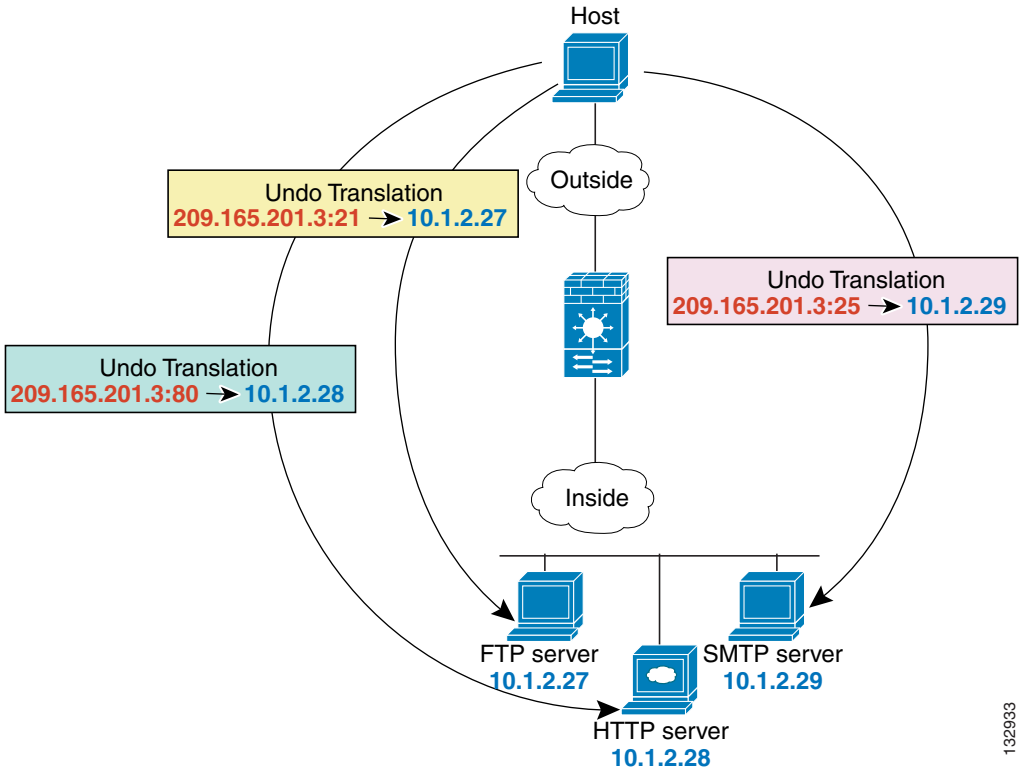
Static PAT is the same as static NAT, except it lets you specify the protocol (TCP or UDP) and port for the real and mapped addresses.

This feature lets you identify the same mapped address across many different static statements, so long as the port is different for each statement. However, you cannot use the same mapped address for multiple static *NAT* statements.

For applications that require application inspection for secondary channels (FTP, VoIP, and so on), the FWSM automatically translates the secondary ports.

For example, if you want to provide a single address for remote users to access FTP, HTTP, and SMTP, but these are all actually different servers on the real network, you can specify static PAT statements for each server that uses the same mapped IP address, but different ports (see Figure 15-8).

Figure 15-8 Static PAT



See the following commands for this example:

```
hostname(config)# static (inside,outside) tcp 209.165.201.3 ftp 10.1.2.27 ftp netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 209.165.201.3 http 10.1.2.28 http netmask
255.255.255.255
hostname(config)# static (inside,outside) tcp 209.165.201.3 smtp 10.1.2.29 smtp netmask
255.255.255.255
```

You can also use static PAT to translate a well-known port to a non-standard port or vice versa. For example, if your inside web servers use port 8080, you can allow outside users to connect to port 80, and then undo translation to the original port 8080. Similarly, if you want to provide extra security, you can tell your web users to connect to non-standard port 6785, and then undo translation to port 80.

Bypassing NAT when NAT Control is Enabled

If you enable NAT control, then inside hosts must match a NAT rule when accessing outside hosts. If you do not want to perform NAT for some hosts, then you can bypass NAT for those hosts (alternatively, you can disable NAT control). You might want to bypass NAT, for example, if you are using an application that does not support NAT (see the "Inspection Engine Overview" section on page 21-2 for information about inspection engines that do not support NAT).

You can configure traffic to bypass NAT using one of three methods. All methods achieve compatibility with inspection engines. However, each method offers slightly different capabilities, as follows:

- Identity NAT (**nat 0** command)—When you configure identity NAT (which is similar to dynamic NAT), you do not limit translation for a host on specific interfaces; you must use identity NAT for connections through all interfaces. Therefore, you cannot choose to perform normal translation on real addresses when you access interface A, but use identity NAT when accessing interface B. Regular dynamic NAT, on the other hand, lets you specify a particular interface on which to translate the addresses. Make sure that the real addresses for which you use identity NAT are routable on all networks that are available according to your access lists.

  For identity NAT, even though the mapped address is the same as the real address, you cannot initiate a connection from the outside to the inside (even if the interface access list allows it). Use static identity NAT or NAT exemption for this functionality.

- Static identity NAT (**static** command)—Static identity NAT lets you specify the interface on which you want to allow the real addresses to appear, so you can use identity NAT when you access interface A, and use regular translation when you access interface B. Static identity NAT also lets you use policy NAT, which identifies the real and destination addresses when determining the real addresses to translate (see the "Policy NAT" section on page 15-10 for more information about policy NAT). For example, you can use static identity NAT for an inside address when it accesses the outside interface and the destination is server A, but use a normal translation when accessing the outside server B.

- NAT exemption (**nat 0 access-list** command)—NAT exemption allows both translated and remote hosts to initiate connections. Like identity NAT, you do not limit translation for a host on specific interfaces; you must use NAT exemption for connections through all interfaces. However, NAT exemption does let you specify the real and destination addresses when determining the real addresses to translate (similar to policy NAT), so you have greater control using NAT exemption. However unlike policy NAT, NAT exemption does not consider the ports in the access list.

Policy NAT

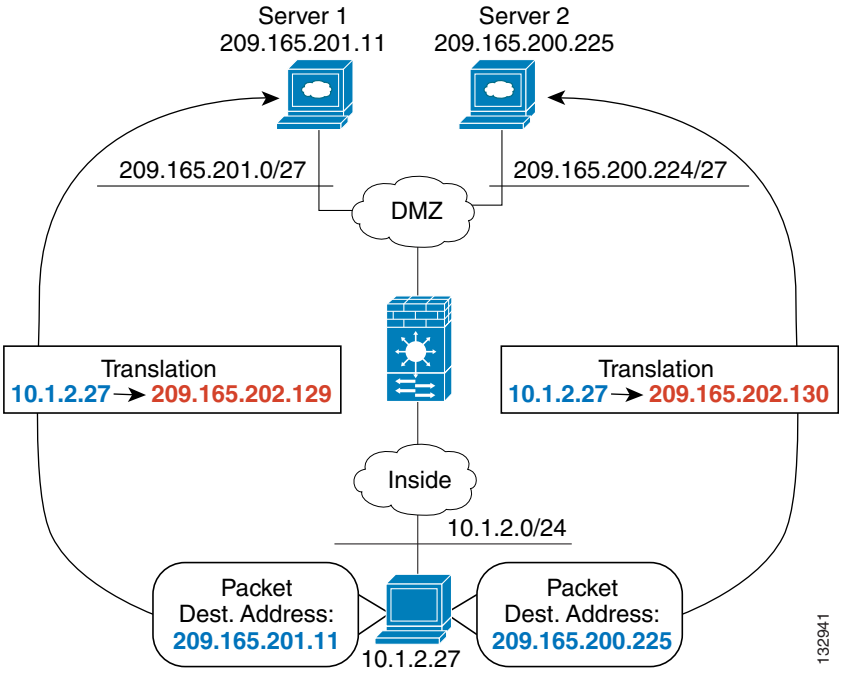
Policy NAT lets you identify real addresses for address translation by specifying the source and destination addresses in an extended access list. You can also optionally specify the source and destination ports. Regular NAT can only consider the source addresses, and not the destination. For example, with policy NAT, you can translate the real address to mapped address A when it accesses server A, but translate the real address to mapped address B when it accesses server B.

For applications that require application inspection for secondary channels (FTP, VoIP, and so on), the policy specified in the policy NAT statement should include the secondary ports. Or, when the ports cannot be predicted, the policy should specify only the IP addresses for the secondary channel. This way, the FWSM translates the secondary ports.

Note All types of NAT support policy NAT except for NAT exemption. NAT exemption uses an access list to identify the real addresses, but differs from policy NAT in that the ports are not considered. See the "Bypassing NAT" section on page 15-32 for other differences. You can accomplish the same result as NAT exemption using static identity NAT, which does support policy NAT.

Figure 15-9 shows a host on the 10.1.2.0/24 network accessing two different servers. When the host accesses the server at 209.165.201.11, the real address is translated to 209.165.202.129. When the host accesses the server at 209.165.200.225, the real address is translated to 209.165.202.130 so that the host appears to be on the same network as the servers, which can help with routing.

Figure 15-9 Policy NAT with Different Destination Addresses

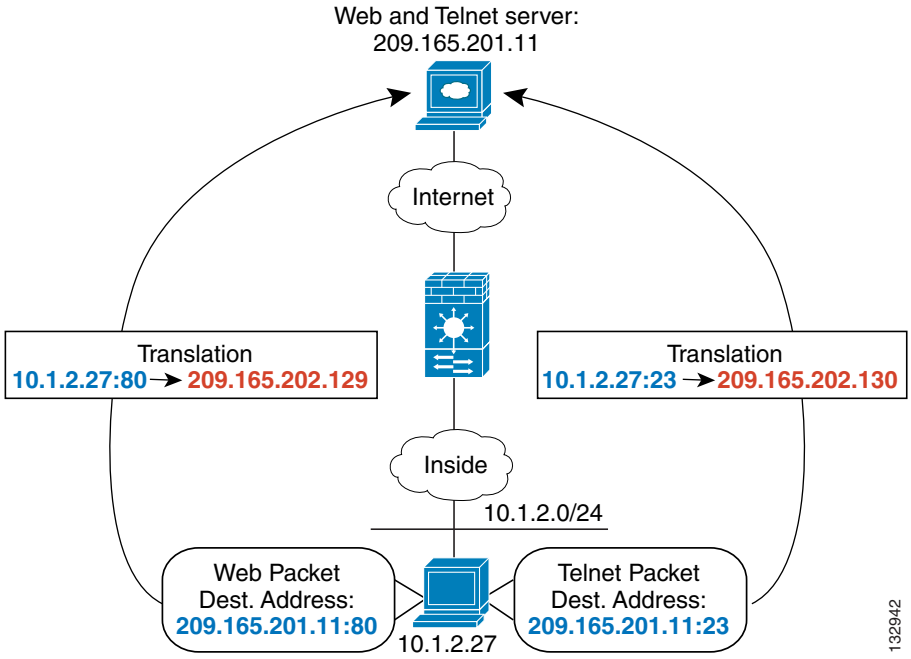


See the following commands for this example:

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
hostname(config)# access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
hostname(config)# nat (inside) 1 access-list NET1
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list NET2
hostname(config)# global (outside) 2 209.165.202.130
```

Figure 15-10 shows the use of source and destination ports. The host on the 10.1.2.0/24 network accesses a single host for both web services and Telnet services. When the host accesses the server for web services, the real address is translated to 209.165.202.129. When the host accesses the same server for Telnet services, the real address is translated to 209.165.202.130.

Figure 15-10 Policy NAT with Different Destination Ports



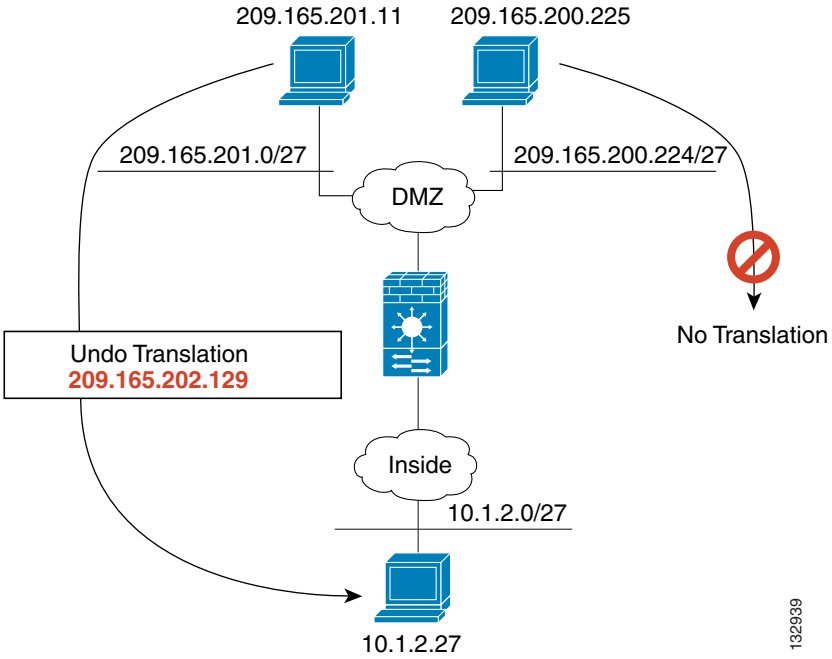
See the following commands for this example:

```
hostname(config)# access-list WEB permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 80
hostname(config)# access-list TELNET permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 23
hostname(config)# nat (inside) 1 access-list WEB
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list TELNET
hostname(config)# global (outside) 2 209.165.202.130
```

For policy static NAT (and for NAT exemption, which also uses an access list to identify traffic), both translated and remote hosts can originate traffic. For traffic originated on the translated network, the NAT access list specifies the real addresses and the *destination* addresses, but for traffic originated on the remote network, the access list identifies the real addresses and the *source* addresses of remote hosts who are allowed to connect to the host using this translation.

Figure 15-11 shows a remote host connecting to a translated host. The translated host has a policy static NAT translation that translates the real address only for traffic to and from the 209.165.201.0/27 network. A translation does not exist for the 209.165.200.224/27 network, so the translated host cannot connect to that network, nor can a host on that network connect to the translated host.

Figure 15-11 Policy Static NAT with Destination Address Translation



See the following commands for this example:

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.224 209.165.201.0
255.255.255.224
hostname(config)# static (inside,outside) 209.165.202.129 access-list NET1
```

Note For policy static NAT, in undoing the translation, the ACL in the **static** command is not used. If the destination address in the packet matches the mapped address in the static rule, the static rule is used to untranslate the address.

Note Policy NAT does not support SQL*Net, but it is supported by regular NAT. See the "Inspection Engine Overview" section on page 21-2 for information about NAT support for other protocols.

NAT Session (Xlate) Creation

By default, the FWSM creates NAT sessions for all connections even if you do not use NAT. For example, a session is created for each untranslated connection even if you do not enable NAT control, you use NAT exemption or identity NAT, or you use same security interfaces and do not configure NAT. Because there is a maximum number of NAT sessions (see the "Managed System Resources" section on page A-4), these kinds of NAT sessions might cause you to run into the limit.

To avoid running into the limit, you can disable NAT sessions for untranslated traffic (called xlate bypass). See the "Configuring Xlate Bypass" section on page 15-18 to enable xlate bypass. If you disable NAT control and have untranslated traffic or use NAT exemption, or you enable NAT control and use NAT exemption, then with xlate bypass, the FWSM does not create a session for these types of untranslated traffic. NAT sessions are still created in the following instances:

- You configure identity NAT (with or without NAT control). Identity NAT is considered to be a translation.
- You use same-security interfaces with NAT control. Traffic between same security interfaces create NAT sessions even when you do not configure NAT for the traffic. To avoid NAT sessions in this case, disable NAT control or use NAT exemption as well as xlate bypass.

NAT and Same Security Level Interfaces

NAT is not required between same security level interfaces even if you enable NAT control. You can optionally configure NAT if desired. However, if you configure dynamic NAT when NAT control is enabled, then NAT is required. See the "NAT Control" section on page 15-5 for more information. Also, when you specify a group of IP address(es) for dynamic NAT or PAT on a same security interface, then you must perform NAT on that group of addresses when they access any lower or same security level interface (even when NAT control is not enabled). Traffic identified for static NAT is not affected.

See the "Allowing Communication Between Interfaces on the Same Security Level" section on page 6-6 to enable same security communication.

Note The FWSM does not support VoIP inspection engines when you configure NAT on same security interfaces. These inspection engines include Skinny, SIP, and H.323. See the "Inspection Engine Overview" section on page 21-2 for supported inspection engines.

Order of NAT Commands Used to Match Real Addresses

The FWSM matches real addresses to NAT commands in the following order:

1. NAT exemption (**nat 0 access-list**)—In order, until the first match. Identity NAT is not included in this category; it is included in the regular static NAT or regular NAT category. We do not recommend overlapping addresses in NAT exemption statements because unexpected results can occur.
2. Static NAT and Static PAT (regular and policy) (**static**)—Best match. Static identity NAT is included in this category. In the case of overlapping addresses in **static** statements, a warning will be displayed, but they are supported. The order of the **static** commands does not matter; the **static** statement that best matches the real address is used.
3. Policy dynamic NAT (**nat access-list**)—In order, until the first match. Overlapping addresses are allowed.
4. Regular dynamic NAT (**nat**)—Best match. Regular identity NAT is included in this category. The order of the NAT commands does not matter; the NAT statement that best matches the real address is used. For example, you can create a general statement to translate all addresses (0.0.0.0) on an interface. If you want to translate a subset of your network (10.1.1.1) to a different address, then you can create a statement to translate only 10.1.1.1. When 10.1.1.1 makes a connection, the specific statement for 10.1.1.1 is used because it matches the real address best. We do not recommend using overlapping statements; they use more memory and can slow the performance of the FWSM.

Maximum Number of NAT Statements

The FWSM supports the following numbers of **nat**, **global**, and **static** commands divided between all contexts or in single mode:

- **nat** command—2 K
- **global** command—4 K
- **static** command—2 K

The FWSM also supports up to 3942 ACEs in access lists used for policy NAT for single mode, and 7272 ACEs for multiple mode.

Mapped Address Guidelines

When you translate the real address to a mapped address, you can use the following mapped addresses:

- Addresses on the same network as the mapped interface.

  If you use addresses on the same network as the mapped interface (through which traffic exits the FWSM), the FWSM uses proxy ARP to answer any requests for mapped addresses, and thus intercepts traffic destined for a real address. This solution simplifies routing, because the FWSM does not have to be the gateway for any additional networks. However, this approach does put a limit on the number of available addresses used for translations.

  For PAT, you can even use the IP address of the mapped interface.

- Addresses on a unique network.

  If you need more addresses than are available on the mapped interface network, you can identify addresses on a different subnet. The FWSM uses proxy ARP to answer any requests for mapped addresses, and thus intercepts traffic destined for a real address. If you use OSPF, and you advertise routes on the mapped interface, then the FWSM advertises the mapped addresses. If the mapped interface is passive (not advertising routes) or you are using static routing, then you need to add a static route on the upstream router that sends traffic destined for the mapped addresses to the FWSM.

DNS and NAT

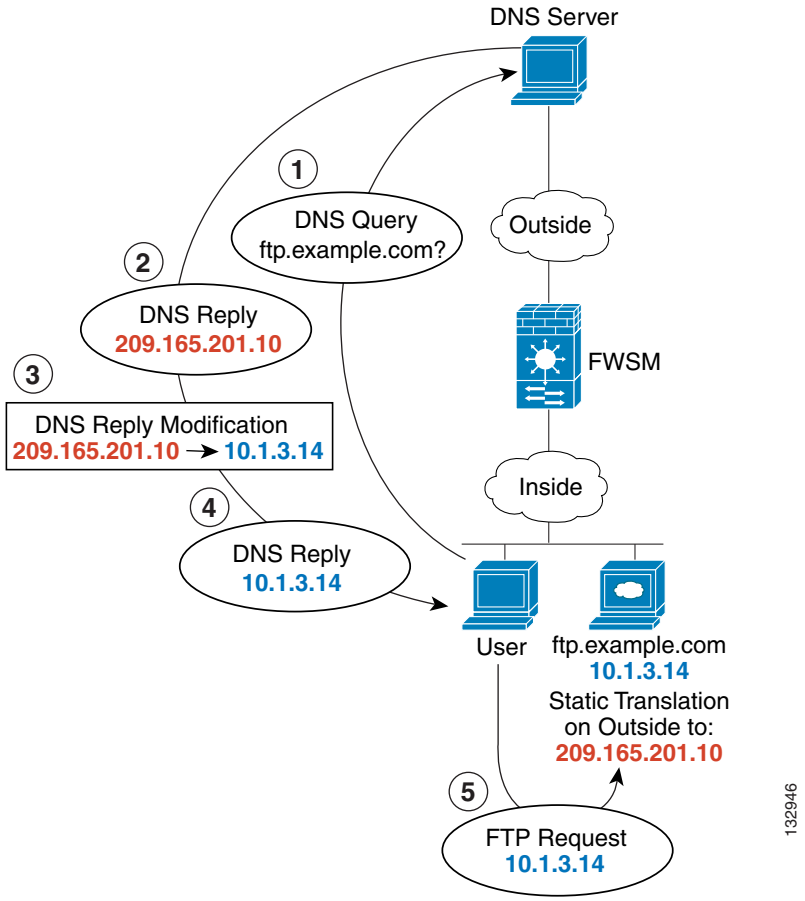
You might need to configure the FWSM to modify DNS replies by replacing the address in the reply with an address that matches the NAT configuration. You can configure DNS modification when you configure each translation.

For example, a DNS server is accessible from the outside interface. A server, ftp.example.com, is on the inside interface. You configure the FWSM to statically translate the ftp.example.com real address (10.1.3.14) to a mapped address (209.165.201.10) that is visible on the outside network (see Figure 15-12). In this case, you want to enable DNS reply modification on this static statement so that inside users who have access to ftp.example.com using the real address receive the real address from the DNS server, and not the mapped address.

When an inside host sends a DNS request for the address of ftp.example.com, the DNS server replies with the mapped address (209.165.201.10). The FWSM refers to the static statement for the inside server and translates the address inside the DNS reply to 10.1.3.14. If you do not enable DNS reply modification, then the inside host attempts to send traffic to 209.165.201.10 instead of accessing ftp.example.com directly.

Note A route needs to exist for the real IP address embedded in the DNS query response or the FWSM will not NAT it. The necessary route can be learned via static routing or by any other routing protocol, such as RIP or OSPF.

Figure 15-12 DNS Reply Modification



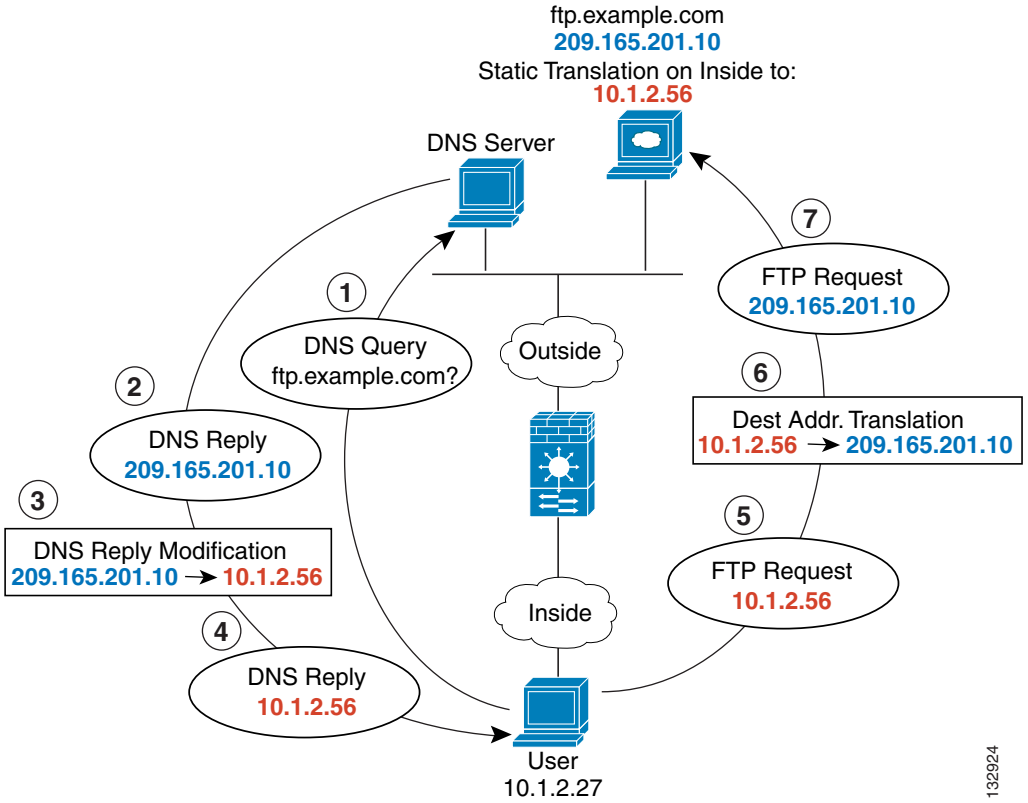
See the following command for this example:

```
hostname(config)# static (inside,outside) 209.165.201.10 10.1.3.14 netmask 255.255.255.255
dns
```

Note If a user on a different network (for example, DMZ) also requests the IP address for ftp.cisco.com from the outside DNS server, then the IP address in the DNS reply is also modified for this user, even though the user is not on the Inside interface referenced by the **static** command.

Figure 15-13 shows a web server and DNS server on the outside. The FWSM has a static translation for the outside server. In this case, when an inside user requests the address for ftp.example.com from the DNS server, the DNS server responds with the real address, 209.165.20.10. Because you want inside users to use the mapped address for ftp.example.com (10.1.2.56) you need to configure DNS reply modification for the static translation.

Figure 15-13 DNS Reply Modification Using Outside NAT



See the following command for this example:

```
hostname(config)# static (outside,inside) 10.1.2.56 209.165.201.10 netmask 255.255.255.255
dns
```

Configuring NAT Control

NAT control requires that packets traversing from an inside interface to an outside interface match a NAT rule. See the "NAT Control" section on page 15-5 for more information.

To enable NAT control, enter the following command:

```
hostname(config)# nat-control
```

To disable NAT control, enter the **no** form of the command.

Configuring Xlate Bypass

By default, the FWSM creates NAT sessions for all connections even if you do not use NAT. To avoid running into the maximum NAT session limit, you can disable NAT sessions for untranslated traffic (called xlate bypass). See the "NAT Session (Xlate) Creation" section on page 15-13 for more information.

To enable xlate bypass, enter the following command:

```
hostname(config)# xlate-bypass
```

To disable xlate bypass, enter the **no** form of the command.

The following sample output from the **show xlate detail** command shows xlate bypass disabled. The bolded display output shows that all 16 connections require identity NAT xlates even though NAT is not explicitly configured for any of the connections.

```
hostname# show xlate detail
Flags: D - DNS, d - dump, I - identity, i - inside, n - no random,
       o - outside, r - portmap, s - static
16 in use, 16 most used
NAT from inside:10.1.1.11 to outside:10.1.1.11 flags Ii
NAT from inside:10.1.1.12 to outside:10.1.1.12 flags Ii
NAT from inside:10.1.1.13 to outside:10.1.1.13 flags Ii
NAT from inside:10.1.1.14 to outside:10.1.1.14 flags Ii
NAT from inside:10.1.1.15 to outside:10.1.1.15 flags Ii
...
NAT from inside:10.1.1.25 to outside:10.1.1.25 flags Ii
NAT from inside:10.1.1.26 to outside:10.1.1.26 flags Ii.
```

The following sample output from the **show xlate detail** command shows xlate bypass enabled. The bolded display output shows that of the 16 connections active, none require xlates.

```
hostname# show xlate detail
Flags: D - DNS, d - dump, I - identity, i - inside, n - no random,
       o - outside, r - portmap, s - static
0 in use, 16 most used
```

The following sample output from the **show xlate detail** command shows xlate bypass enabled, but includes a static identity NAT configuration, which does require an xlate.

```
hostname(config)# static (inside,outside) 10.1.1.20 10.1.1.20 netmask 255.255.255.255
hostname(config)# show xlate detail
Flags: D - DNS, d - dump, I - identity, i - inside, n - no random,
       o - outside, r - portmap, s - static
1 in use, 16 most used
NAT from inside:10.1.1.20 to outside:10.1.1.20 flags Isi
```

Using Dynamic NAT and PAT

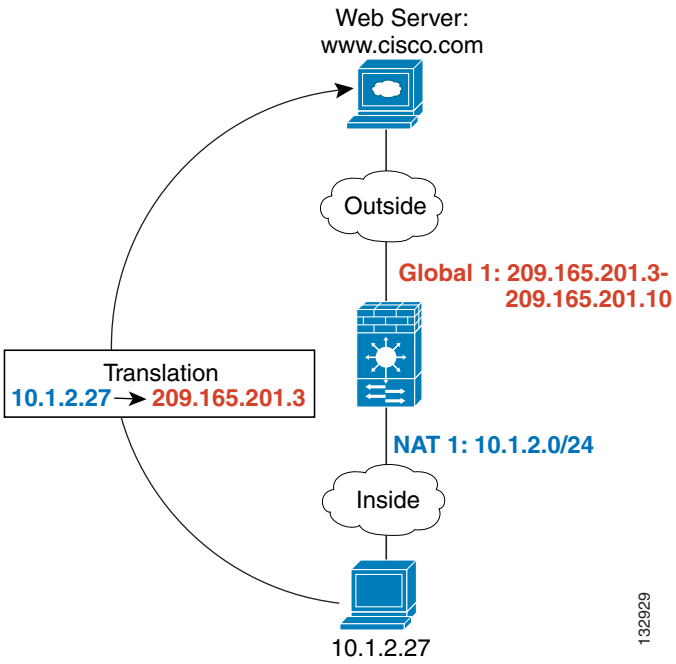
This section describes how to configure dynamic NAT and PAT, and includes the following topics:

- Dynamic NAT and PAT Implementation, page 15-19
- Configuring Dynamic NAT or PAT, page 15-25

Dynamic NAT and PAT Implementation

For dynamic NAT and PAT, you first configure a **nat** command identifying the real addresses on a given interface that you want to translate. Then you configure a separate **global** command to specify the mapped addresses when exiting another interface (in the case of PAT, this is one address). Each **nat** command matches a **global** command by comparing the NAT ID, a number that you assign to each command (see Figure 15-14).

Figure 15-14 nat and global ID Matching

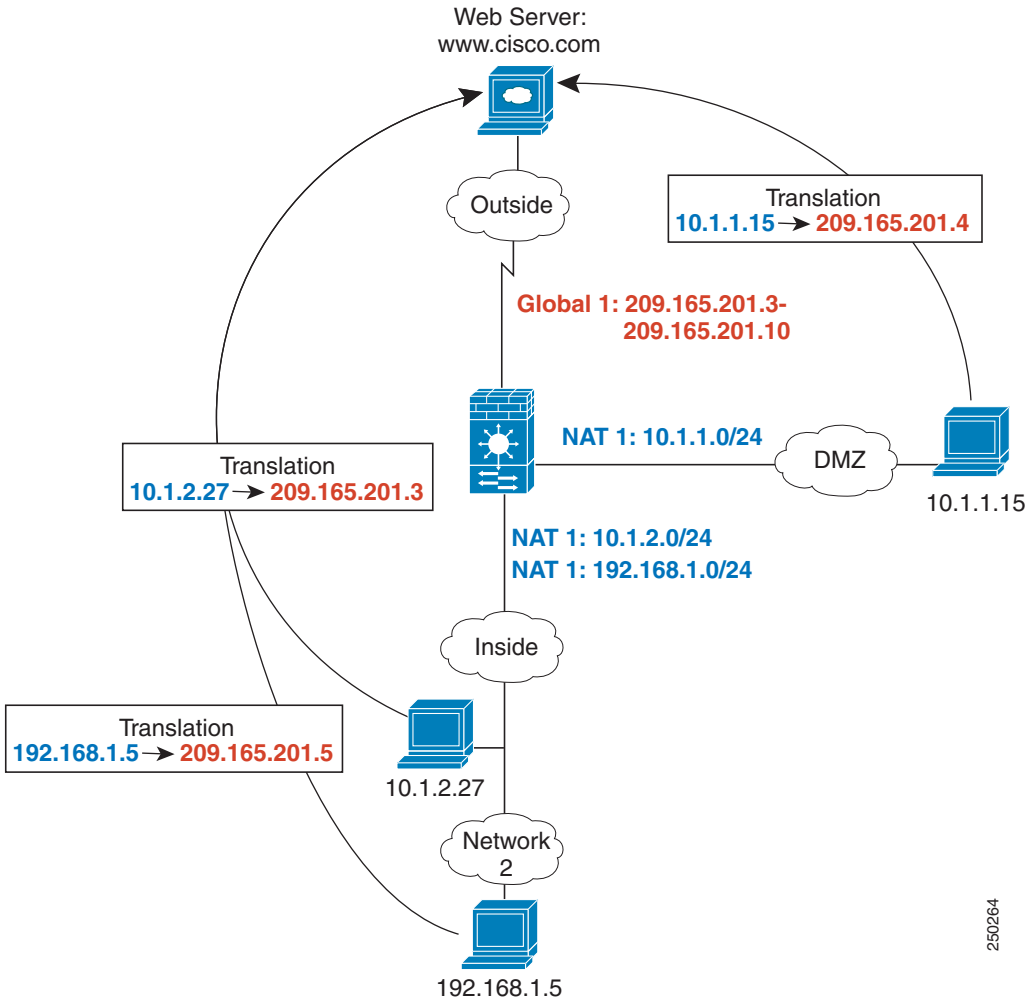


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
```

You can enter multiple **nat** commands using the same NAT ID on one or more interfaces; they all use the same **global** command when traffic exits a given interface. For example, you can configure **nat** commands for Inside and DMZ interfaces, both on NAT ID 1. Then you configure a **global** command on the Outside interface that is also on ID 1. Traffic from the Inside interface and the DMZ interface share a mapped pool or a PAT address when exiting the Outside interface (see Figure 15-15).

Figure 15-15 nat Commands on Multiple Interfaces

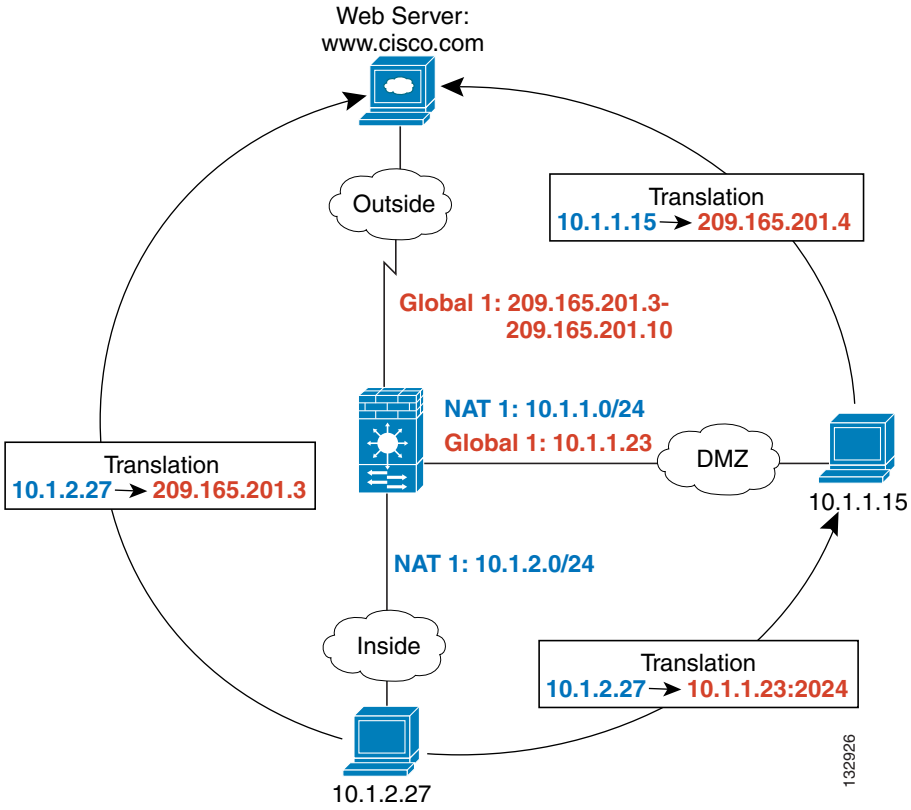


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# nat (inside) 1 192.168.1.0 255.255.255.0
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
```

You can also enter a **global** command for each interface using the same NAT ID. If you enter a **global** command for the Outside and DMZ interfaces on ID 1, then the Inside **nat** command identifies traffic to be translated when going to both the Outside and the DMZ interfaces. Similarly, if you also enter a **nat** command for the DMZ interface on ID 1, then the **global** command on the Outside interface is also used for DMZ traffic. (See Figure 15-16).

Figure 15-16 global and nat Commands on Multiple Interfaces

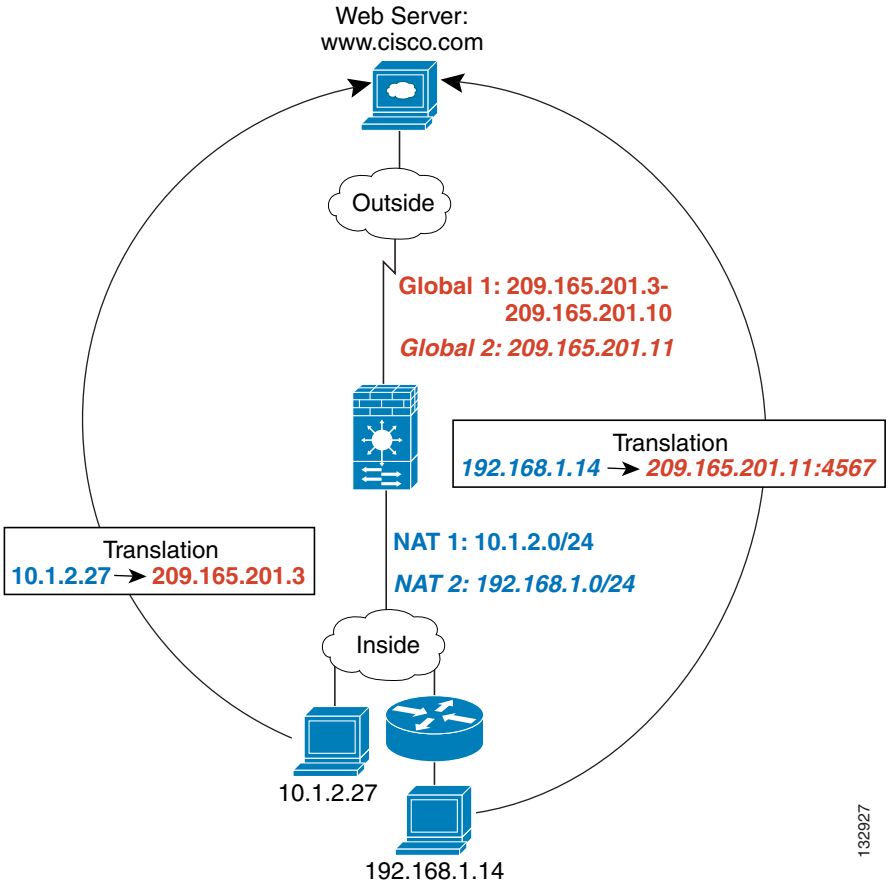


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
hostname(config)# global (dmz) 1 10.1.1.23
```

If you use different NAT IDs, you can identify different sets of real addresses to have different mapped addresses. For example, on the Inside interface, you can have two **nat** commands on two different NAT IDs. On the Outside interface, you configure two **global** commands for these two IDs. Then, when traffic from Inside network A exits the Outside interface, the IP addresses are translated to pool A addresses; while traffic from Inside network B are translated to pool B addresses (see Figure 15-17). If you use policy NAT, you can specify the same real addresses for multiple **nat** commands, as long as the the destination addresses and ports are unique in each access list.

Figure 15-17 Different NAT IDs

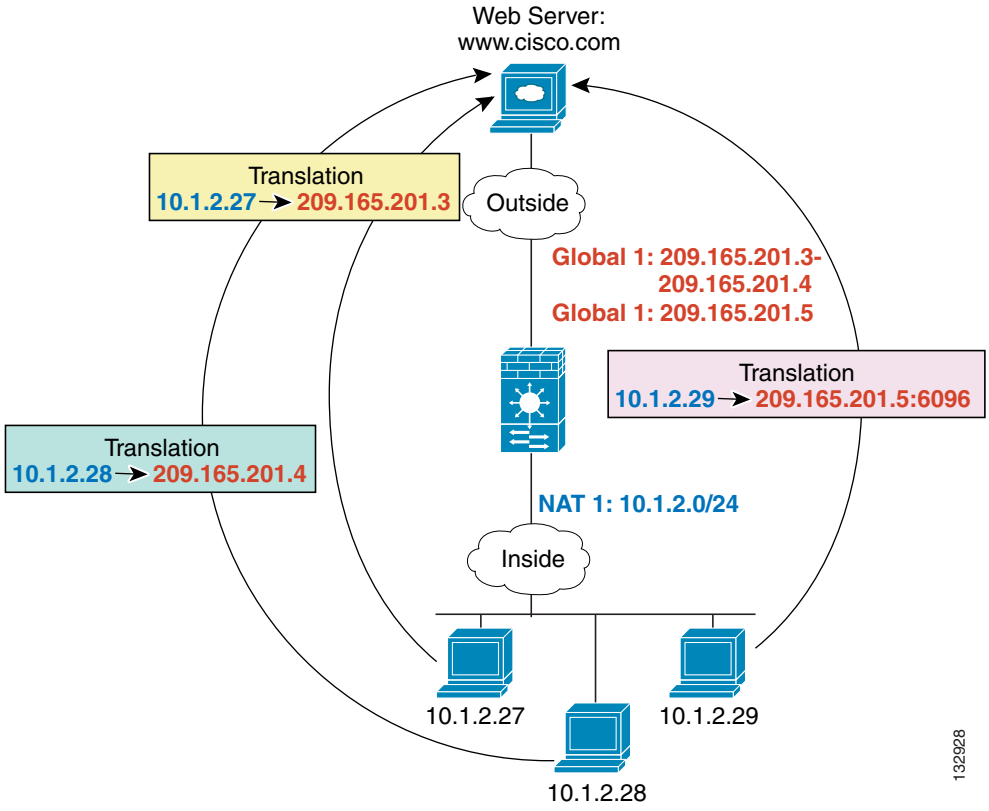


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# nat (inside) 2 192.168.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.10
hostname(config)# global (outside) 2 209.165.201.11
```

You can enter multiple **global** commands for one interface using the same NAT ID; the FWSM uses the dynamic NAT **global** commands first, in the order they are in the configuration, and then uses the PAT **global** commands in order. You might want to enter both a dynamic NAT **global** command and a PAT **global** command if you need to use dynamic NAT for a particular application, but want to have a backup PAT statement in case all the dynamic NAT addresses are depleted. Similarly, you might enter two PAT statements if you need more than the approximately 64,000 PAT sessions that a single PAT mapped statement supports (see Figure 15-18).

Figure 15-18 NAT and PAT Together

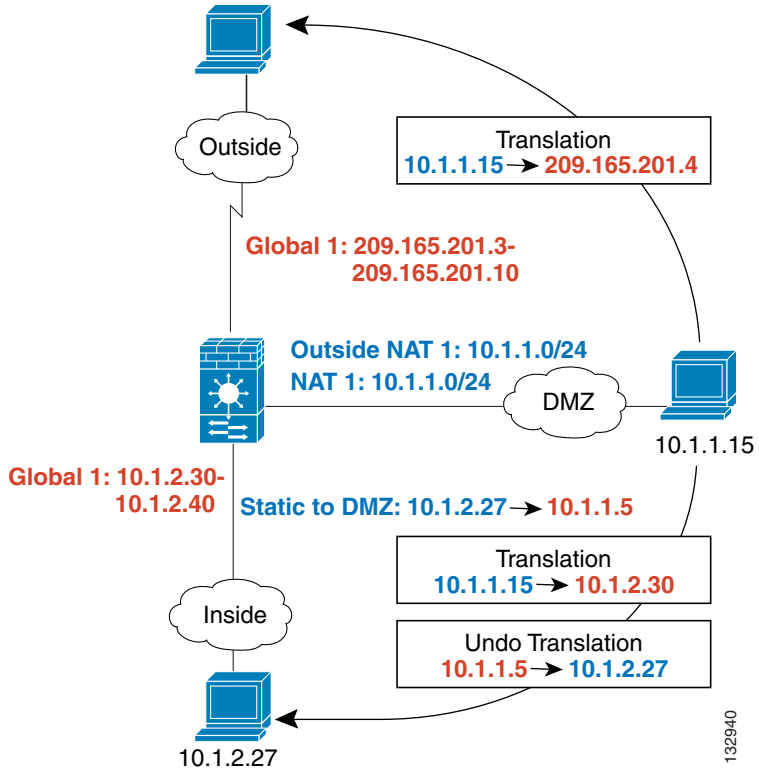


See the following commands for this example:

```
hostname(config)# nat (inside) 1 10.1.2.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.4
hostname(config)# global (outside) 1 209.165.201.5
```

For outside NAT (from outside to inside), you need to use the **outside** keyword in the **nat** command. If you also want to translate the same traffic when it accesses an outside interface (for example, traffic on a DMZ is translated when accessing the Inside and the Outside interfaces), then you must configure a separate **nat** command without the **outside** option. In this case, you can identify the same addresses in both statements and use the same NAT ID (see Figure 15-19). Note that for outside NAT (DMZ interface to Inside interface), the inside host uses a **static** command to allow outside access, so both the source and destination addresses are translated.

Figure 15-19 Outside NAT and Inside NAT Combined



See the following commands for this example:

```
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0 outside
hostname(config)# nat (dmz) 1 10.1.1.0 255.255.255.0
hostname(config)# static (inside,dmz) 10.1.1.5 10.1.2.27 netmask 255.255.255.255
hostname(config)# global (outside) 1 209.165.201.3-209.165.201.4
hostname(config)# global (inside) 1 10.1.2.30-1-10.1.2.40
```

When you specify a group of IP address(es) in a **nat** command, then you must perform NAT on that group of addresses when they access any lower or same security level interface; you must apply a **global** command with the same NAT ID on each interface, or use a **static** command. NAT is not required for that group when it accesses a higher security interface, because to perform NAT from outside to inside, you must create a separate **nat** command using the **outside** keyword. If you do apply outside NAT, then the NAT requirements preceding come into effect for that group of addresses when they access all higher security interfaces. Traffic identified by a **static** command is not affected.

Configuring Dynamic NAT or PAT

This section describes how to configure dynamic NAT or dynamic PAT. The configuration for dynamic NAT and PAT are almost identical; for NAT you specify a range of mapped addresses, and for PAT you specify a single address.

Figure 15-20 shows a typical dynamic NAT scenario. Only translated hosts can create a NAT session, and responding traffic is allowed back. The mapped address is dynamically assigned from a pool defined by the **global** command.

Figure 15-20 Dynamic NAT

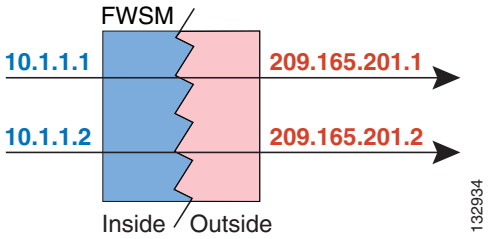
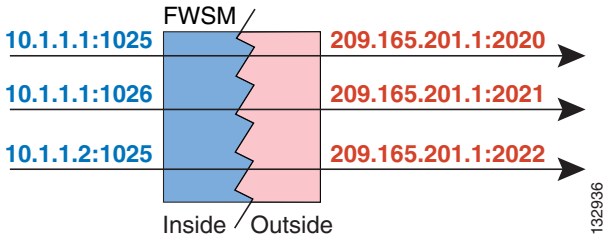


Figure 15-21 shows a typical dynamic PAT scenario. Only translated hosts can create a NAT session, and responding traffic is allowed back. The mapped address defined by the **global** command is the same for each translation, but the port is dynamically assigned.

Figure 15-21 Dynamic PAT



For more information about dynamic NAT, see the "Dynamic NAT" section on page 15-6. For more information about PAT, see the "PAT" section on page 15-8.

Note If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections that use translations.

To configure dynamic NAT or PAT, perform the following steps:

Step 1 To identify the real addresses that you want to translate, enter one of the following commands:

- Policy NAT:

```
hostname(config)# nat (real_interface) nat_id access-list acl_name [dns] [outside]
[[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns] [norandomseq]
```

You can identify overlapping addresses in other **nat** commands. For example, you can identify 10.1.1.0 in one command, but 10.1.1.1 in another. The traffic is matched to a policy NAT command in order, until the first match, or for regular NAT, using the best match.

See the following description about options for this command:

- **access-list** *acl_name*—Identify the real addresses and destination addresses using an extended access list. Create the extended access list using the **access-list extended** command (see the "Adding an Extended Access List" section on page 12-6). This access list should include only **permit** ACEs. You can optionally specify the real and destination ports in the access list using the **eq** operator. Policy NAT and static NAT consider the **inactive** or **time-range** keywords and stop working when an ACE is inactive.
- *nat_id*—An integer between 1 and 65535. The NAT ID should match a **global** command NAT ID. See the "Dynamic NAT and PAT Implementation" section on page 15-19 for more information about how NAT IDs are used. **0** is reserved for NAT exemption. (See the "Configuring NAT Exemption" section on page 15-35 for more information about NAT exemption.)
- **dns**—If your **nat** command includes the address of a host that has an entry in a DNS server, and the DNS server is on a different interface from a client, then the client and the DNS server need different addresses for the host; one needs the mapped address and one needs the real address. This option rewrites the address in the DNS reply to the client. The translated host needs to be on the same interface as either the client or the DNS server. Typically, hosts that need to allow access from other interfaces use a static translation, so this option is more likely to be used with the **static** command. (See the "DNS and NAT" section on page 15-15 for more information.)
- **outside**—If this interface is on a lower security level than the interface you identify by the matching **global** statement, then you must enter **outside** to identify the NAT instance as outside NAT.
- **tcp** *tcp_max_conns*—Sets the maximum number of simultaneous TCP connections for the entire subnet up to 65,536. The default is 0, which means the maximum connections.
- *emb_limit*—Sets the maximum number of embryonic connections per host up to 65,536. The default is 0, which means the maximum connections. You must enter the **tcp** *tcp_max_conns* before you enter the *emb_limit*. If you want to use the default value for *tcp_max_conns*, but change the *emb_limit*, then enter **0** for *tcp_max_conns*.

  An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. Limiting the number of embryonic connections protects you from a DoS attack. The FWSM uses the embryonic limit to trigger TCP Intercept. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. TCP Intercept uses the SYN cookies algorithm to prevent TCP SYN-flooding attacks. A SYN-flooding attack consists of a series of SYN packets usually originating from spoofed IP addresses. The constant flood of SYN packets keeps the server SYN queue full, which prevents it from servicing connection requests. When the embryonic connection threshold of a connection is crossed, the FWSM acts as a proxy for the server and generates a SYN-ACK response to the client's SYN request. When the FWSM receives an ACK back from the client, it can then authenticate the client and allow the connection to the server.
- **udp** *udp_max_conns*—Sets the maximum number of simultaneous UDP connections for the entire subnet up to 65,536. The default is 0, which means the maximum connections.
- **norandomseq**—Disables TCP Initial Sequence Number (ISN) randomization.TCP initial sequence number randomization can be disabled if another in-line firewall is also randomizing the initial sequence numbers, because there is no need for both firewalls to be performing this action. However, leaving ISN randomization enabled on both firewalls does not affect the

traffic. Each TCP connection has two ISNs: one generated by the client and one generated by the server. The security appliance randomizes the ISN of the TCP SYN passing in the outbound direction. If the connection is between two interfaces with the same security level, then the ISN will be randomized in the SYN in both directions. Randomizing the ISN of the protected host prevents an attacker from predecting the next ISN for a new connection and potentially hijacking the new session.

Note You can alternatively set connection limits (but not embryonic connection limits) using the Modular Policy Framework. See the "Configuring Connection Limits and Timeouts" section on page 20-1 for more information. You can only set embryonic connection limits using NAT. If you configure these settings for the same traffic using both methods, then the FWSM uses the lower limit. For TCP sequence randomization, if it is disabled using either method, then the FWSM disables TCP sequence randomization.

- Regular NAT:

```
hostname(config)# nat (real_interface) nat_id real_ip [mask [dns] [outside]
[[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns] [norandomseq]]
```

The *nat_id* is an integer between 1 and 2147483647. The NAT ID must match a **global** command NAT ID. See the "Dynamic NAT and PAT Implementation" section on page 15-19 for more information about how NAT IDs are used. **0** is reserved for identity NAT. See the "Configuring Identity NAT" section on page 15-33 for more information about identity NAT.

See the preceding policy NAT command for information about other options.

Step 2 To identify the mapped address(es) to which you want to translate the real addresses when they exit a particular interface, enter the following command:

```
hostname(config)# global (mapped_interface) nat_id {mapped_ip[-mapped_ip]}
```

This NAT ID should match a **nat** command NAT ID. The matching **nat** command identifies the addresses that you want to translate when they exit this interface.

You can specify a single address (for PAT) or a range of addresses (for NAT). The range can go across subnet boundaries if desired. For example, you can specify the following "supernet":

```
192.168.1.1-192.168.2.254
```

For example, to translate the 10.1.1.0/24 network on the inside interface, enter the following command:

```
hostname(config)# nat (inside) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.1-209.165.201.30
```

To identify a pool of addresses for dynamic NAT as well as a PAT address for when the NAT pool is exhausted, enter the following commands:

```
hostname(config)# nat (inside) 1 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 1 209.165.201.5
hostname(config)# global (outside) 1 209.165.201.10-209.165.201.20
```

To translate the lower security DMZ network addresses so they appear to be on the same network as the inside network (10.1.1.0), for example, to simplify routing, enter the following commands:

```
hostname(config)# nat (dmz) 1 10.1.2.0 255.255.255.0 outside dns
hostname(config)# global (inside) 1 10.1.1.45
```

To identify a single real address with two different destination addresses using policy NAT, enter the following commands (see Figure 15-9 on page 15-11 for a related figure):

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
hostname(config)# access-list NET2 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
hostname(config)# nat (inside) 1 access-list NET1 tcp 0 2000 udp 10000
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list NET2 tcp 1000 500 udp 2000
hostname(config)# global (outside) 2 209.165.202.130
```

To identify a single real address/destination address pair that use different ports using policy NAT, enter the following commands (see Figure 15-10 on page 15-12 for a related figure):

```
hostname(config)# access-list WEB permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 80
hostname(config)# access-list TELNET permit tcp 10.1.2.0 255.255.255.0 209.165.201.11
255.255.255.255 eq 23
hostname(config)# nat (inside) 1 access-list WEB
hostname(config)# global (outside) 1 209.165.202.129
hostname(config)# nat (inside) 2 access-list TELNET
hostname(config)# global (outside) 2 209.165.202.130
```

Note FWSM and ASA behave differently when you configure dynamic NAT without the **global** keyword. On FWSM, an identity xlate is created, and the the packet is forwarded. On the ASA, no xlate is created, and the packet is dropped due to the missing **global** keyword.
In the following example, a packet sourced from 10.1.1.0/24 with a destination behind the outside interface is forwarded on the FWSM but dropped on the ASA:

```
hostname(config)# nat (inside) 1 10.1.1.0 255.255.255.0 without global (outside) 1 X
```

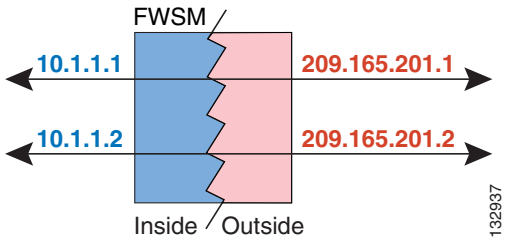
To avoid this situation, configure NAT exemption (nat 0), or specify the **global** keyword.

Using Static NAT

This section describes how to configure a static translation.

Figure 15-22 shows a typical static NAT scenario. The translation is always active so both translated and remote hosts can originate connections, and the mapped address is statically assigned by the **static** command.

Figure 15-22 Static NAT



You cannot use the same real or mapped address in multiple **static** commands between the same two interfaces unless you use static PAT (see the "Using Static PAT" section on page 15-30). Do not use a mapped address in the **static** command that is also defined in a **global** command for the same mapped interface.

For more information about static NAT, see the "Static NAT" section on page 15-8.

Note If you remove a **static** command, existing connections that use the translation are not affected. To remove these connections, enter the **clear local-host** or the **clear xlate** command. Static translations from the translation table can be removed using the **clear xlate** command; the translation table will be cleared and all current translations are deleted. The **clear xlate** command clears all connections, even when xlate-bypass is enabled and when a connection does not have an xlate.

For more information about these commands, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

To configure static NAT, enter one of the following commands.

- For policy static NAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) mapped_ip
access-list acl_name [dns] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
[norandomseq]
```

  Identify the real addresses and destination/source addresses using an extended access list. Create the extended access list using the **access-list extended** command (see the "Adding an Extended Access List" section on page 12-6). The first address in the access list is the real address; the second address is either the source or destiniation address, depending on where the traffic originates. For example, to translate the real address 10.1.1.1 to the mapped address 192.168.1.1 when 10.1.1.1 sends traffic to the 209.165.200.224 network, the **access-list** and **static** commands are:

```
hostname(config)# access-list TEST extended ip host 10.1.1.1 209.165.200.224
255.255.255.224
hostname(config)# static (inside,outside) 192.168.1.1 access-list TEST
```

  In this case, the second address is the destination address. However, the same configuration is used for hosts to originate a connection to the mapped address. For example, when a host on the 209.165.200.224/27 network initiates a connection to 192.168.1.1, then the second address in the access list is the source address.

  This access list should include only **permit** ACEs. You can optionally specify the real and destination ports in the access list using the **eq** operator. Policy NAT and static NAT consider the **inactive** or **time-range** keywords and stop working when an ACE is inactive. See the "Policy NAT" section on page 15-10 for more information.

  If you specify a network for translation (for example, 10.1.1.0 255.255.255.0), then the FWSM translates the .0 and .255 addresses. If you want to prevent access to these addresses, be sure to configure an access list to deny access.

  See the "Configuring Dynamic NAT or PAT" section on page 15-25 for information about the other options.

- To configure regular static NAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) mapped_ip real_ip
[netmask mask] [dns] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
[norandomseq]
```

See the "Configuring Dynamic NAT or PAT" section on page 15-25 for information about the options.

For example, the following policy static NAT example shows a single real address that is translated to two mapped addresses depending on the destination address (see Figure 15-9 on page 15-11 for a related figure):

```
hostname(config)# access-list NET1 permit ip host 10.1.2.27 209.165.201.0 255.255.255.224
hostname(config)# access-list NET2 permit ip host 10.1.2.27 209.165.200.224
255.255.255.224
hostname(config)# static (inside,outside) 209.165.202.129 access-list NET1
hostname(config)# static (inside,outside) 209.165.202.130 access-list NET2
```

The following command maps an inside IP address (10.1.1.3) to an outside IP address (209.165.201.12):

```
hostname(config)# static (inside,outside) 209.165.201.12 10.1.1.3 netmask 255.255.255.255
```

The following command maps the outside address (209.165.201.15) to an inside address (10.1.1.6):

```
hostname(config)# static (outside,inside) 10.1.1.6 209.165.201.15 netmask 255.255.255.255
```

The following command statically maps an entire subnet:

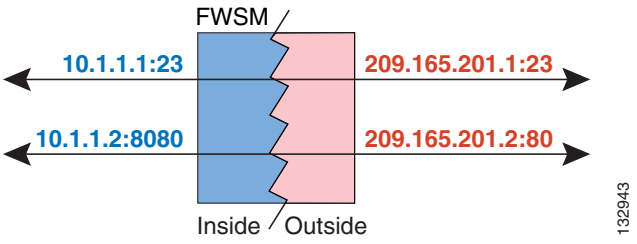
```
hostname(config)# static (inside,dmz) 10.1.1.0 10.1.2.0 netmask 255.255.255.0
```

Using Static PAT

This section describes how to configure a static port translation. Static PAT lets you translate the real IP address to a mapped IP address, as well as the real port to a mapped port. You can choose to translate the real port to the same port, which lets you translate only specific types of traffic, or you can take it further by translating to a different port.

Figure 15-23 shows a typical static PAT scenario. The translation is always active so that both translated and remote hosts can originate connections, and the mapped address and port is statically assigned by the **static** command.

Figure 15-23 Static PAT



For applications that require application inspection for secondary channels (FTP, VoIP, and so on), the FWSM automatically translates the secondary ports.

Do not use a mapped address in the **static** command that is also defined in a **global** command for the same mapped interface.

For more information about static PAT, see the "Static PAT" section on page 15-9.

Note If you remove a **static** command, existing connections that use the translation are not affected. To remove these connections, enter the **clear local-host** command.

Static translations from the translation table can be removed using the **clear xlate** command; the translation table will be cleared and all current translations are deleted.

To configure static PAT, enter one of the following commands.

- For policy static PAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) {tcp | udp} mapped_ip
mapped_port access-list acl_name [dns] [[tcp] tcp_max_conns [emb_limit]]
[udp udp_max_conns] [norandomseq]
```

  Identify the real addresses and destination/source addresses using an extended access list. Create the extended access list using the **access-list extended** command (see the "Adding an Extended Access List" section on page 12-6). The protocol in the access list must match the protocol you set in this command. For example, if you specify **tcp** in the **static** command, then you must specify **tcp** in the access list. Specify the port using the **eq** operator.

  The first address in the access list is the real address; the second address is either the source or destiniation address, depending on where the traffic originates. For example, to translate the real address 10.1.1.1/Telnet to the mapped address 192.168.1.1/Telnet when 10.1.1.1 sends traffic to the 209.165.200.224 network, the **access-list** and **static** commands are:

```
hostname(config)# access-list TEST extended tcp host 10.1.1.1 209.165.200.224
255.255.255.224 eq telnet
hostname(config)# static (inside,outside) tcp 192.168.1.1 telnet access-list TEST
```

  In this case, the second address is the destination address. However, the same configuration is used for hosts to originate a connection to the mapped address. For example, when a host on the 209.165.200.224/27 network initiates a Telnet connection to 192.168.1.1, then the second address in the access list is the source address.

  This access list should include only **permit** ACEs. Policy NAT and static NAT consider the **inactive** or **time-range** keywords and stop working when an ACE is inactive. See the "Policy NAT" section on page 15-10 for more information.

  If you specify a network for translation (for example, 10.1.1.0 255.255.255.0), then the FWSM translates the .0 and .255 addresses. If you want to prevent access to these addresses, be sure to configure an access list to deny access.

  See the "Configuring Dynamic NAT or PAT" section on page 15-25 for information about the other options.

- To configure regular static PAT, enter the following command:

```
hostname(config)# static (real_interface,mapped_interface) {tcp | udp} mapped_ip
mapped_port real_ip real_port [netmask mask] [dns] [[tcp] tcp_max_conns [emb_limit]]
[udp udp_max_conns] [norandomseq]
```

  See the "Configuring Dynamic NAT or PAT" section on page 15-25 for information about the options.

For example, for Telnet traffic initiated from hosts on the 10.1.3.0 network to the FWSM outside interface (10.1.2.14), you can redirect the traffic to the inside host at 10.1.1.15 by entering the following commands:

```
hostname(config)# access-list TELNET permit tcp host 10.1.1.15 10.1.3.0 255.255.255.0 eq
telnet
```

```
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet access-list TELNET
```

For HTTP traffic initiated from hosts on the 10.1.3.0 network to the FWSM outside interface (10.1.2.14), you can redirect the traffic to the inside host at 10.1.1.15 by entering:

```
hostname(config)# access-list HTTP permit tcp host 10.1.1.15 10.1.3.0 255.255.255.0 eq
http
hostname(config)# static (inside,outside) tcp 10.1.2.14 http access-list HTTP
```

To redirect Telnet traffic from the FWSM outside interface (10.1.2.14) to the inside host at 10.1.1.15, enter the following command:

```
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet netmask
255.255.255.255
```

If you want to allow the preceding real Telnet server to initiate connections, though, then you need to provide additional translation. For example, to translate all other types of traffic, enter the following commands. The original **static** command provides translation for Telnet to the server, while the **nat** and **global** commands provide PAT for outbound connections from the server.

```
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet netmask
255.255.255.255
hostname(config)# nat (inside) 1 10.1.1.15 255.255.255.255
hostname(config)# global (outside) 1 10.1.2.14
```

If you also have a separate translation for all inside traffic, and the inside hosts use a different mapped address from the Telnet server, you can still configure traffic initiated from the Telnet server to use the same mapped address as the **static** statement that allows Telnet traffic to the server. You need to create a more exclusive **nat** statement just for the Telnet server. Because **nat** statements are read for the best match, more exclusive **nat** statements are matched before general statements. The following example shows the Telnet **static** statement, the more exclusive **nat** statement for initiated traffic from the Telnet server, and the statement for other inside hosts, which uses a different mapped address.

```
hostname(config)# static (inside,outside) tcp 10.1.2.14 telnet 10.1.1.15 telnet netmask
255.255.255.255
hostname(config)# nat (inside) 1 10.1.1.15 255.255.255.255
hostname(config)# global (outside) 1 10.1.2.14
hostname(config)# nat (inside) 2 10.1.1.0 255.255.255.0
hostname(config)# global (outside) 2 10.1.2.78
```

To translate a well-known port (80) to another port (8080), enter the following command:

```
hostname(config)# static (inside,outside) tcp 10.1.2.45 80 10.1.1.16 8080 netmask
255.255.255.255
```

Bypassing NAT

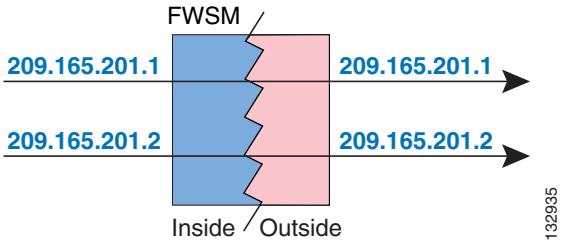
This section describes how to bypass NAT. You might want to bypass NAT when you enable NAT control. You can bypass NAT using identity NAT, static identity NAT, or NAT exemption. See the "Bypassing NAT when NAT Control is Enabled" section on page 15-10 for more information about these methods. This section includes the following topics:

Configuring Identity NAT

Identity NAT translates the real IP address to the same IP address. Only "translated" hosts can create NAT translations, and responding traffic is allowed back.

Figure 15-24 shows a typical identity NAT scenario.

Figure 15-24 Identity NAT



Note If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections that use translations.

To configure identity NAT, enter the following command:

```
hostname(config)# nat (real_interface) 0 real_ip [mask [dns] [outside]
[[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns] [norandomseq]]
```

See the "Configuring Dynamic NAT or PAT" section on page 15-25 for information about the options.

For example, to use identity NAT for the inside 10.1.1.0/24 network, enter the following command:

```
hostname(config)# nat (inside) 0 10.1.1.0 255.255.255.0
```

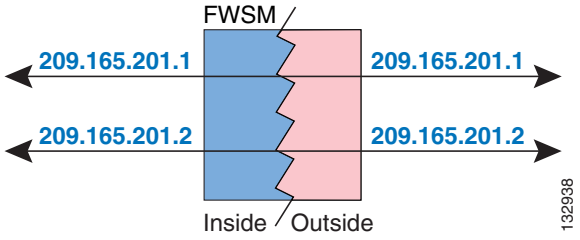
Configuring Static Identity NAT

Static identity NAT translates the real IP address to the same IP address. The translation is always active, and both "translated" and remote hosts can originate connections. Static identity NAT lets you use regular NAT or policy NAT. Policy NAT lets you identify the real and destination addresses when determining the real addresses to translate (see the "Policy NAT" section on page 15-10 for more

information about policy NAT). For example, you can use policy static identity NAT for an inside address when it accesses the outside interface and the destination is server A, but use a normal translation when accessing the outside server B.

Figure 15-25 shows a typical static identity NAT scenario.

Figure 15-25 Static Identity NAT



Note If you remove a **static** command, existing connections that use the translation are not affected. To remove these connections, enter the **clear local-host** command.

Static translations from the translation table can be removed using the **clear xlate** command; the translation table will be cleared and all current translations are deleted.

To configure static identity NAT, enter one of the following commands:

- To configure policy static identity NAT, enter the following command:

  ```
  hostname(config)# static (real_interface,mapped_interface) real_ip access-list acl_id
  [dns] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns] [norandomseq]
  ```

  Create the extended access list using the **access-list extended** command (see the "Adding an Extended Access List" section on page 12-6). This access list should include only **permit** ACEs. Make sure the source address in the access list matches the *real_ip* in this command. Policy NAT and static NAT consider the **inactive** or **time-range** keywords and stop working when an ACE is inactive. See the "Policy NAT" section on page 15-10 for more information.

  See the "Configuring Dynamic NAT or PAT" section on page 15-25 for information about the other options.

- To configure regular static identity NAT, enter the following command:

  ```
  hostname(config)# static (real_interface,mapped_interface) real_ip real_ip
  [netmask mask] [dns] [[tcp] tcp_max_conns [emb_limit]] [udp udp_max_conns]
  [norandomseq]
  ```

  Specify the same IP address for both *real_ip* arguments.

  See the "Configuring Dynamic NAT or PAT" section on page 15-25 for information about the other options.

For example, the following command uses static identity NAT for an inside IP address (10.1.1.3) when accessed by the outside:

```
hostname(config)# static (inside,outside) 10.1.1.3 10.1.1.3 netmask 255.255.255.255
```

The following command uses static identity NAT for an outside address (209.165.201.15) when accessed by the inside:

```
hostname(config)# static (outside,inside) 209.165.201.15 209.165.201.15 netmask
255.255.255.255
```

The following command statically maps an entire subnet:

```
hostname(config)# static (inside,dmz) 10.1.2.0 10.1.2.0 netmask 255.255.255.0
```

The following static identity policy NAT example shows a single real address that uses identity NAT when accessing one destination address, and a translation when accessing another:

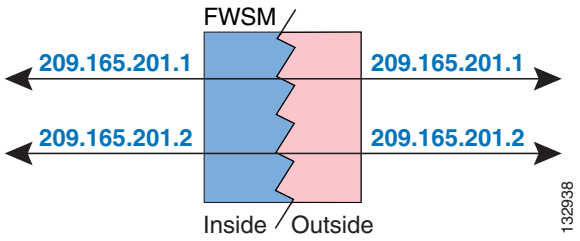
```
hostname(config)# access-list NET1 permit ip host 10.1.2.27 209.165.201.0 255.255.255.224
hostname(config)# access-list NET2 permit ip host 10.1.2.27 209.165.200.224
255.255.255.224
hostname(config)# static (inside,outside) 10.1.2.27 access-list NET1
hostname(config)# static (inside,outside) 209.165.202.130 access-list NET2
```

Configuring NAT Exemption

NAT exemption exempts addresses from translation and allows both real and remote hosts to originate connections. NAT exemption lets you specify the real and destination addresses when determining the real traffic to exempt (similar to policy NAT), so you have greater control using NAT exemption than identity NAT. However unlike policy NAT, NAT exemption does not consider the ports in the access list. Use static identity NAT to consider ports in the access list.

Figure 15-26 shows a typical NAT exemption scenario.

Figure 15-26 NAT Exemption



Note If you remove a NAT exemption configuration, existing connections that use NAT exemption are not affected. To remove these connections, enter the **clear local-host** command.

To configure NAT exemption, enter the following command:

```
hostname(config)# nat (real_interface) 0 access-list acl_name [outside] [[tcp]
tcp_max_conns [emb_limit]] [udp udp_max_conns] [norandomseq]
```

Create the extended access list using the **access-list extended** command (see the "Adding an Extended Access List" section on page 12-6). This access list can include both **permit** ACEs and **deny** ACEs. Do not specify the real and destination ports in the access list; NAT exemption does not consider the ports. NAT exemption also does not consider the **inactive** or **time-range** keywords; all ACEs are considered to be active for NAT exemption configuration.

See the "Configuring Dynamic NAT or PAT" section on page 15-25 for information about the other options.

By default, this command exempts traffic from inside to outside. If you want traffic from outside to inside to bypass NAT, then add an additional **nat** command and enter **outside** to identify the NAT instance as outside NAT. You might want to use outside NAT exemption if you configure dynamic NAT for the outside interface and want to exempt other traffic.

For example, to exempt an inside network when accessing any destination address, enter the following command:

```
hostname(config)# access-list EXEMPT permit ip 10.1.2.0 255.255.255.0 any
hostname(config)# nat (inside) 0 access-list EXEMPT
```

To use dynamic outside NAT for a DMZ network, and exempt another DMZ network, enter the following command:

```
hostname(config)# nat (dmz) 1 10.1.2.0 255.255.255.0 outside dns
hostname(config)# global (inside) 1 10.1.1.45
hostname(config)# access-list EXEMPT permit ip 10.1.3.0 255.255.255.0 any
hostname(config)# nat (dmz) 0 access-list EXEMPT
```

To exempt an inside address when accessing two different destination addresses, enter the following commands:

```
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.201.0
255.255.255.224
hostname(config)# access-list NET1 permit ip 10.1.2.0 255.255.255.0 209.165.200.224
255.255.255.224
hostname(config)# nat (inside) 0 access-list NET1
```

NAT Examples

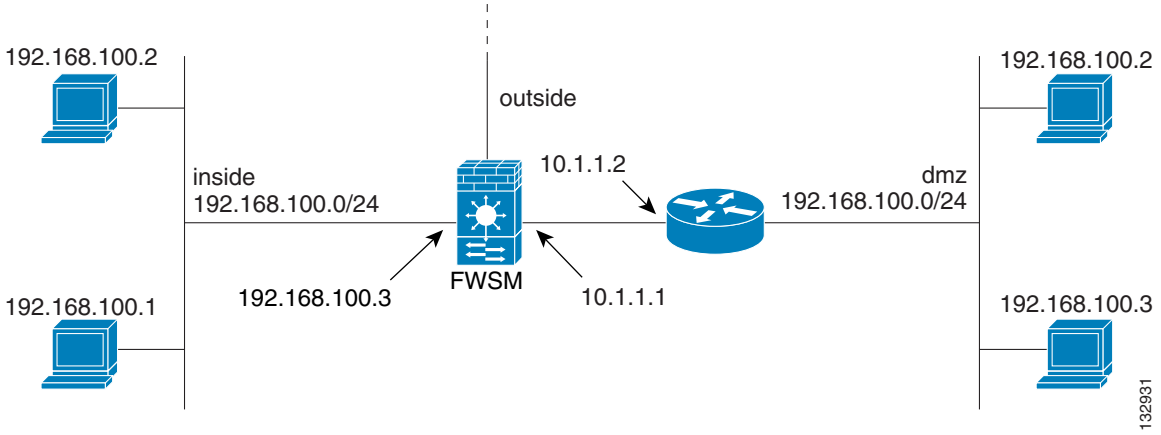
This section describes typical scenarios that use NAT solutions, and includes the following topics:

- Overlapping Networks, page 15-37
- Redirecting Ports, page 15-38

Overlapping Networks

In Figure 15-27, the FWSM connects two private networks with overlapping address ranges.

Figure 15-27 Using Outside NAT with Overlapping Networks



Two networks use an overlapping address space (192.168.100.0/24), but hosts on each network must communicate (as allowed by access lists). Without NAT, when a host on the inside network tries to access a host on the overlapping DMZ network, the packet never makes it past the FWSM, which sees the packet as having a destination address on the inside network. Moreover, if the destination address is being used by another host on the inside network, that host receives the packet.

To solve this problem, use NAT to provide non-overlapping addresses. If you want to allow access in both directions, use static NAT for both networks. If you only want to allow the inside interface to access hosts on the DMZ, then you can use dynamic NAT for the inside addresses, and static NAT for the DMZ addresses you want to access. This example shows static NAT.

To configure static NAT for these two interfaces, perform the following steps. The 10.1.1.0/24 network on the DMZ is not translated.

Step 1 Translate 192.168.100.0/24 on the inside to 10.1.2.0 /24 when it accesses the DMZ by entering the following command:

```
hostname(config)# static (inside,dmz) 10.1.2.0 192.168.100.0 netmask 255.255.255.0
```

Step 2 Translate the 192.168.100.0/24 network on the DMZ to 10.1.3.0/24 when it accesses the inside by entering the following command:

```
hostname(config)# static (dmz,inside) 10.1.3.0 192.168.100.0 netmask 255.255.255.0
```

Step 3 Configure the following static routes so that traffic to the DMZ network can be routed correctly by the FWSM:

```
hostname(config)# route dmz 192.168.100.128 255.255.255.128 10.1.1.2 1
hostname(config)# route dmz 192.168.100.0 255.255.255.128 10.1.1.2 1
```

The FWSM already has a connected route for the inside network. These static routes allow the FWSM to send traffic for the 192.168.100.0/24 network out the DMZ interface to the gateway router at 10.1.1.2. (You need to split the network into two because you cannot create a static route with the exact same network as a connected route.) Alternatively, you could use a more broad route for the DMZ traffic, such as a default route.

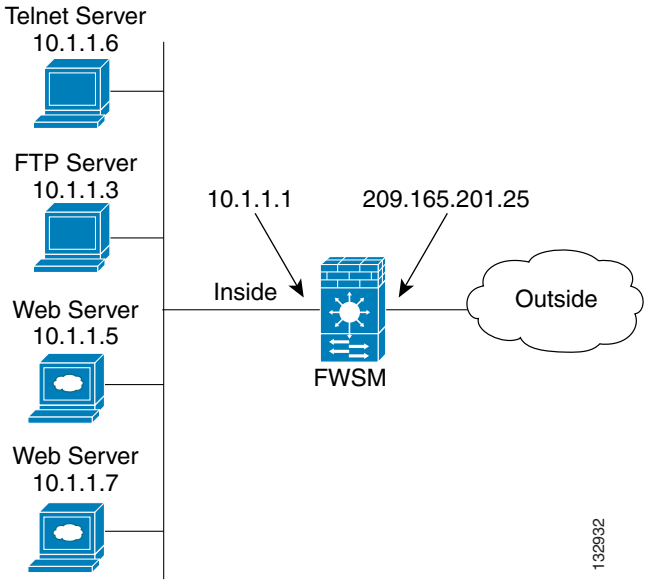
If host 192.168.100.2 on the DMZ network wants to initiate a connection to host 192.168.100.2 on the inside network, the following events occur:

1. The DMZ host 192.168.100.2 sends the packet to IP address 10.1.2.2.
2. When the FWSM receives this packet, the FWSM translates the source address from 192.168.100.2 to 10.1.3.2.
3. Then the FWSM translates the destination address from 10.1.2.2 to 192.168.100.2, and the packet is forwarded.

Redirecting Ports

Figure 15-28 illustrates a typical network scenario in which the port redirection feature might be useful.

Figure 15-28 Port Redirection Using Static PAT



In the configuration described in this section, port redirection occurs for hosts on external networks as follows:

- Telnet requests to IP address 209.165.201.5 are redirected to 10.1.1.6.
- FTP requests to IP address 209.165.201.5 are redirected to 10.1.1.3.
- HTTP request to FWSM outside IP address 209.165.201.25 are redirected to 10.1.1.5.
- HTTP port 8080 requests to PAT address 209.165.201.15 are redirected to 10.1.1.7 port 80.

To implement this scenario, perform the following steps:

Step 1 Configure PAT for the inside network by entering the following commands:

```
hostname(config)# nat (inside) 1 0.0.0.0 0.0.0.0 0 0
hostname(config)# global (outside) 1 209.165.201.15
```

Step 2 Redirect Telnet requests for 209.165.201.5 to 10.1.1.6 by entering the following command:

```
hostname(config)# static (inside,outside) tcp 209.165.201.5 telnet 10.1.1.6 telnet netmask
255.255.255.255
```

Step 3 Redirect FTP requests for IP address 209.165.201.5 to 10.1.1.3 by entering the following command:

```
hostname(config)# static (inside,outside) tcp 209.165.201.5 ftp 10.1.1.3 ftp netmask
255.255.255.255
```

Step 4 Redirect HTTP requests for the FWSM outside interface address to 10.1.1.5 by entering the following command:

```
hostname(config)# static (inside,outside) tcp interface www 10.1.1.5 www netmask
255.255.255.255
```

Step 5 Redirect HTTP requests on port 8080 for PAT address 209.165.201.15 to 10.1.1.7 port 80 by entering the following command:

```
hostname(config)# static (inside,outside) tcp 209.165.201.15 8080 10.1.1.7 www netmask
255.255.255.255
```

CHAPTER 16

Applying AAA for Network Access

This chapter describes how to enable AAA (pronounced “triple A”) for network access.

For information about AAA for management access, see the “AAA for System Administrators” section on page 22-10.

This chapter includes the following sections:

AAA Performance

The FWSM uses “cut-through proxy” to significantly improve performance compared to a traditional proxy server. The performance of a traditional proxy server suffers because it analyzes every packet at the application layer of the OSI model. The FWSM cut-through proxy challenges a user initially at the application layer and then authenticates against standard RADIUS, TACACS+, or the local database. After the FWSM authenticates the user, it shifts the session flow, and all traffic flows directly and quickly between the source and destination while maintaining session state information.

Configuring Authentication for Network Access

This section includes the following topics:

Authentication Overview

The FWSM lets you configure network access authentication using AAA servers. This section includes the following topics:

One-Time Authentication

A user at a given IP address only needs to authenticate one time for all rules and types, until the authentication session expires. (See the **timeout uauth** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for timeout values.) For example, if you configure the FWSM to authenticate Telnet and FTP, and a user first successfully authenticates for Telnet, then as long as the authentication session exists, the user does not also have to authenticate for FTP.

For HTTP or HTTPS authentication, once authenticated, a user never has to reauthenticate, no matter how low the **timeout uauth** command is set, because the browser caches the string "Basic=Uuhjksdkfhk==" in every subsequent connection to that particular site. This can be cleared only when the user exits *all* instances of the web browser and restarts. Flushing the cache is of no use.

Applications Required to Receive an Authentication Challenge

Although you can configure the FWSM to require authentication for network access to any protocol or service, users can authenticate directly with HTTP, HTTPS, Telnet, or FTP only. A user must first authenticate with one of these services before the FWSM allows other traffic requiring authentication.

The authentication ports that the FWSM supports for AAA are fixed:

- Port 21 for FTP
- Port 23 for Telnet
- Port 80 for HTTP
- Port 443 for HTTPS

FWSM Authentication Prompts

For Telnet and FTP, the FWSM generates an authentication prompt. After you authenticate correctly, the FWSM redirects you to your original destination. If the destination server also has its own authentication, you enter another username and password.

For HTTP, you log in using basic HTTP authentication supplied by the browser. For HTTPS, the FWSM generates custom login windows.

Note If you use HTTP authentication without using the **aaa authentication secure-http-client** command, the username and password are sent from the client to the FWSM in clear text. We recommend that you use the **aaa authentication secure-http-client** command whenever you enable HTTP authentication. For more information about the **aaa authentication secure-http-client** command, see the "Enabling Secure Authentication of Web Clients" section on page 16-6.

For FTP, a user has the option of entering the FWSM username followed by an at sign (@) and then the FTP username (name1@name2). For the password, the user enters the FWSM password followed by an at sign (@) and then the FTP password (password1@password2). For example, enter the following text.

```
name> user1@user2
password> letmein@he110
```

This feature is useful when you have cascaded firewalls that require multiple logins. You can separate several names and passwords by multiple at signs (@).

Static PAT and HTTP

For HTTP authentication, the FWSM checks real ports when static PAT is configured. If it detects traffic destined for real port 80, regardless of the mapped port, the FWSM intercepts the HTTP connection and enforces authentication.

For example, assume that outside TCP port 889 is translated to port 80 (www) and that any relevant access lists permit the traffic:

```
static (inside,outside) tcp 10.48.66.155 889 192.168.123.10 www netmask 255.255.255.255
```

Then when users try to access 10.48.66.155 on port 889, the FWSM intercepts the traffic and enforces HTTP authentication. Users see the HTTP authentication page in their web browsers before the FWSM allows HTTP connection to complete.

If the local port is different than port 80, as in the following example:

```
static (inside,outside) tcp 10.48.66.155 889 192.168.123.10 111 netmask 255.255.255.255
```

Then users do not see the authentication page. Instead, the FWSM sends to the web browser an error message indicating that the user must be authenticated prior using the requested service.

Authenticating Directly with the FWSM

If you do not want to allow HTTP(S), Telnet, or FTP through the FWSM but want to authenticate other types of traffic, you can configure virtual Telnet, virtual SSH, or virtual HTTP. With virtual Telnet, SSH, or HTTP, the user connects using Telnet, SSH, or HTTP to a given IP address configured on the FWSM, and the FWSM provides a prompt. For more information about the **virtual telnet**, **virtual ssh**, or **virtual http** commands, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Enabling Network Access Authentication

To enable network access authentication, perform the following steps:

Step 1 Using the **aaa-server** command, identify your AAA servers. If you have already identified your AAA servers, continue to the next step.

For more information about identifying AAA servers, see the "Identifying AAA Server Groups and Servers" section on page 11-9.

Step 2 Using the **access-list** command, create an access list that identifies the source addresses and destination addresses of traffic you want to authenticate. For steps, see the "Adding an Extended Access List" section on page 12-6.

The **permit** ACEs mark matching traffic for authentication, while **deny** entries exclude matching traffic from authentication. Be sure to include the destination ports for either HTTP(S), Telnet, or FTP in the access list because the user must authenticate with one of these services before other services are allowed through the FWSM.

Step 3 To configure authentication, enter the following command:

```
hostname(config)# aaa authentication match acl_name interface_name server_group
```

where *acl_name* is the name of the access list you created in Step 2, *interface_name* is the name of the interface as specified with the **nameif** command, and *server_group* is the AAA server group you created in Step 1.

Note You can alternatively use the **aaa authentication include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

Step 4 (Optional) If you are using the local database for network access authentication and you want to limit the number of consecutive failed login attempts that the FWSM allows any given user account, use the **aaa local authentication attempts max-fail** command. For example:

```
hostname(config)# aaa local authentication attempts max-fail 7
```

Tip To clear the lockout status of a specific user or all users, use the **clear aaa local user lockout** command.

Step 5 (Optional) When a user authentication times out or you clear the authentication sessions using the **clear uauth** command, you can force any active connections to close immediately by entering the following command:

```
hostname(config)# aaa authentication clear-conn interface_name source_ip source_mask
```

Without this command, active connections are not terminated even though the user authentication session expired.

For example, the following commands authenticate all inside HTTP traffic and SMTP traffic:

```
hostname(config)# aaa-server AuthOutbound protocol tacacs+
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp
hostname(config)# access-list MAIL_AUTH extended permit tcp any any eq www
hostname(config)# aaa authentication match MAIL_AUTH inside AuthOutbound
```

The following commands authenticate Telnet traffic from the outside interface to a particular server (209.165.201.5):

```
hostname(config)# aaa-server AuthInbound protocol tacacs+
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthInbound (inside) host 10.1.1.1
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# access-list TELNET_AUTH extended permit tcp any host 209.165.201.5 eq
telnet
hostname(config)# aaa authentication match TELNET_AUTH outside AuthInbound
```

Configuring Custom Login Prompts

By default, when a user authenticates with the FWSM, they see the following prompt:

- For HTTP—`HTTP Authentication`.
- For FTP—`FTP Authentication`.
- For Telnet—no prompt.

You can customize the login prompt, and also show prompts when a user is accepted or rejected. If you use a RADIUS server that communicates with a Windows Active Directory server, the reject prompt can be customized to show when a user was rejected due to invalid credentials (the wrong username or password) or because a password has expired. If a password expired, the user is prompted for a new password.

Note Customizing the login prompt causes the FWSM to use MSCHAPv2 for the user password. Please check for MSCHAPv2 compatibility with your RADIUS server and back-end database before enabling this feature.

To customize the login prompt, perform the following steps:

Step 1 To customize the login prompt, enter the following command:

```
hostname(config)# auth-prompt prompt text
```

Where *text* is a string of up to 235 alphanumeric characters or 31 words, limited by whichever maximum is first reached. Special characters, spaces, and punctuation characters are permitted. Entering a question mark or pressing the **Enter** key ends the string. (The question mark appears in the string.)

Step 2 To show text when a user is accepted, enter the following command:

```
hostname(config)# auth-prompt accept text
```

Step 3 To show text when a user is rejected, enter the following command:

```
hostname(config)# auth-prompt reject text
```

When you enter the **reject** keyword without the **invalid-credentials** or **reject expired-pwd** keywords, then this generic prompt is displayed for all rejections that are not due to invalid credentials or expired passwords. For a rejection due to an invalid credential or an expired password, then the prompt you set for the **invalid-credentials** or **reject expired-pwd** keyword displays. If you do not set any prompts for invalid credentials or expired passwords, then the generic reject prompt is shown in all cases.

Step 4 To show text when a user is rejected due to invalid credentials, enter the following command:

```
hostname(config)# auth-prompt reject invalid-credentials text
```

Step 5 To show text when a user is rejected due to an expired password, enter the following command:

```
hostname(config)# auth-prompt reject expired-pwd text
```

This prompt is only used if the RADIUS server uses a Windows Active Directory server for the username and password. You must configure a prompt using the **expired-pwd** keyword for the user to be prompted for a new password.

The following example sets the authentication prompt to the string "Please enter your username and password.":

```
hostname(config)# auth-prompt prompt Please enter your username and password
```

After this string is added to the configuration, users see the following:

```
Please enter your username and password
User Name:
Password:
```

You can also provide separate messages to display when the FWSM accepts or rejects the authentication attempt; for example:

```
hostname(config)# auth-prompt reject Authentication failed. Try again.
hostname(config)# auth-prompt accept Authentication succeeded.
```

To set rejection messages for invalid credentials, expired password, and for unknown rejection reasons, enter the following commands:

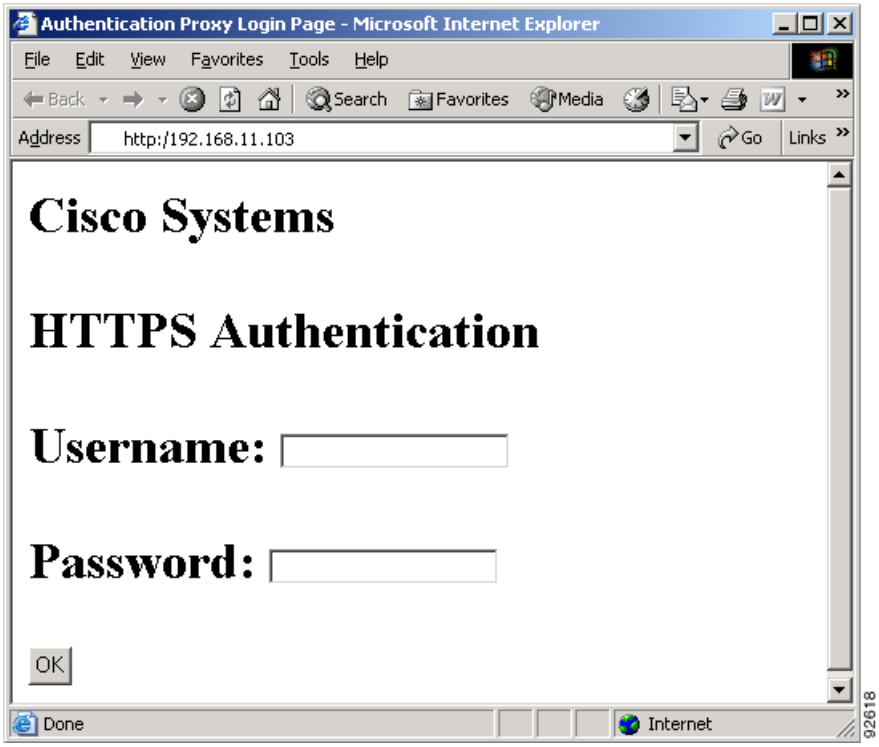
```
hostname(config)# auth-prompt reject Authentication failed. Try again.
hostname(config)# auth-prompt reject invalid-credentials Incorrect username or password
hostname(config)# auth-prompt reject expired-pwd Your password is expired. Reset your
password and try again.
```

Enabling Secure Authentication of Web Clients

The FWSM provides a method of securing HTTP authentication. Without securing HTTP authentication, usernames and passwords provided to the FWSM would be passed to the destination web server. By using the **aaa authentication secure-http-client** command, you enable the exchange of usernames and passwords between a web client and the FWSM with HTTPS. HTTPS encrypts the transmission, preventing the username and password from being passed to the external web server by HTTP.

After enabling this feature, when a user accesses a web page requiring authentication, the FWSM displays the Authentication Proxy Login Page shown in Figure 16-1.

Figure 16-1 Authentication Proxy Login Page



Note The Cisco Systems text field shown in this example was customized using the **auth-prompt** command. See the "Configuring Custom Login Prompts" section on page 16-5.

After the user enters a valid username and password, an "Authentication Successful" page appears and closes automatically. If the user fails to enter a valid username and password, an "Authentication Failed" page appears.

Secured web-client authentication has the following limitations:

- A maximum of 128 concurrent HTTPS authentication sessions are allowed. If all 128 HTTPS authentication processes are running, a new connection requiring authentication will not succeed.
- When **uauth timeout 0** is configured (the **uauth timeout** is set to 0), HTTPS authentication might not work. If a browser initiates multiple TCP connections to load a web page after HTTPS authentication, the first connection is let through, but the subsequent connections trigger authentication. As a result, users are continuously presented with an authentication page, even if the correct username and password are entered each time. To work around this, set the **uauth timeout** to 1 second with the **timeout uauth 0:0:1** command. However, this workaround opens a 1-second window of opportunity that might allow non-authenticated users to go through the firewall if they are coming from the same source IP address.
- Because HTTPS authentication occurs on the SSL port 443, users must not configure an **access-list** command statement to block traffic from the HTTP client to HTTP server on port 443. Furthermore, if static PAT is configured for web traffic on port 80, it must also be configured for the SSL port. In the following example, the first line configures static PAT for web traffic and the second line must be added to support the HTTPS authentication configuration.

```
static (inside,outside) tcp 10.132.16.200 www 10.130.16.10 www
```

```
static (inside,outside) tcp 10.132.16.200 443 10.130.16.10 443
```

- HTTP users see a pop-up window generated by the browser itself if **aaa authentication secure-http-client** is not configured. If **aaa authentication secure-http-client** is configured, a form loads in the browser to collect username and password. In either case, if a user enters an incorrect password, the user is prompted again. When the web server and the authentication server are on different hosts, use the **virtual http** command to get the correct authentication behavior.

To enable secure authentication of web clients, perform the following steps:

Step 1 Enable HTTP authentication. For more information about enabling authentication, see the "Enabling Network Access Authentication" section on page 16-3.

Step 2 To enable secure authentication of web clients, enter this command:

```
aaa authentication secure-http-client
```

Note Use of the **aaa authentication secure-http-client** command is not dependent upon enabling HTTP authentication. If you prefer, you can enter this command before you enable HTTP authentication so that if you later enable HTTP authentication, usernames and passwords are already protected by secured web-client authentication.

Disabling Authentication Challenge per Protocol

You can configure whether the FWSM challenges users for a username and password. By default, the FWSM prompts the user when a AAA rule enforces authentication for traffic in a new session and the protocol of the traffic is FTP, Telnet, HTTP, or HTTPS. In some cases, you may want to disable the authentication challenge for one or more of these protocols, using the following command:

```
hostname(config)# aaa authentication protocol challenge disable
```

For example, to disable the username and password challenge for new connections using FTP, enter the following command:

```
hostname(config)# aaa authentication ftp challenge disable
```

If you disable challenge authentication for a particular protocol, traffic using that protocol is allowed only if the traffic belongs to a session previously authenticated. This authentication can be accomplished by traffic using a protocol whose authentication challenge remains enabled. For example, if you disable challenge authentication for FTP, the FWSM denies new session using FTP if the traffic is included in an authentication rule. If the user establishes the session with a protocol whose authentication challenge is enabled (such as HTTP), FTP traffic is allowed.

Configuring Authorization for Network Access

After a user authenticates for a given connection, the FWSM can use authorization to further control traffic from the user.

This section includes the following topics:

Configuring TACACS+ Authorization

You can configure the FWSM to perform network access authorization with TACACS+.

After a user authenticates, the FWSM checks the authorization rules for matching traffic. If the traffic matches the authorization statement, the FWSM sends the username to the TACACS+ server. The TACACS+ server responds to the FWSM with information that the FWSM treats as a user-specific, dynamic access list for that traffic, based on the user profile.

Note If you have used the **access-group** command to apply access lists to interfaces, be aware of the following effects of the **per-user-override** keyword on authorization by dynamic access lists:

- Without the **per-user-override** keyword, traffic for a user session must be permitted by both the interface access list and the dynamic access list.
- With the **per-user-override** keyword, the dynamic access list determines what is permitted.

For more information, see the **access-group** command entry in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Authentication and authorization statements are independent; however, any unauthenticated traffic matched by an authorization statement will be denied. For authorization to succeed, a user must first authenticate with the FWSM.

Note We suggest that you identify the same traffic for authentication as for authorization. Due to the way the FWSM uses the dynamic access list, if you have a more restrictive authorization statement than authentication, then some connections are unexpectedly denied. When a user first authenticates, if the connection matches the authentication statement and not the authorization statement, then later connections for that user that match the authorization statement are denied (for as long as the uauth session exists). Conversely, if the first connection matches the authorization statement, then later connections that do not match the authorization statement but that match the authentication statement are denied. Therefore, you need to match the authentication and authorization configurations.

See the documentation for your TACACS+ server for information about configuring network access authorizations for a user.

To configure TACACS+ authorization, perform the following steps:

Step 1 Enable authentication. For more information, see the "Enabling Network Access Authentication" section on page 16-3. If you have already enabled authentication, continue to the next step.

Step 2 To enable authorization, enter the following command:

```
hostname(config)# aaa authorization match acl_name interface_name server_group
```

where *acl_name* is the name of the access list you created for authentication, *interface_name* is the name of the interface as specified with the **nameif** command or by default, and *server_group* is the AAA server group you created when you enabled authentication.

The following commands authenticate and authorize inside Telnet traffic.

```
hostname(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
hostname(config)# aaa-server AuthOutbound protocol tacacs+
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
hostname(config)# aaa authorization match TELNET_AUTH inside AuthOutbound
```

Configuring RADIUS Authorization

When authentication succeeds, the RADIUS protocol returns user authorizations in the access-accept packet sent by a RADIUS server. For more information about configuring authentication, see the "Configuring Authentication for Network Access" section on page 16-1.

When you configure the FWSM to authenticate users for network access, you are also implicitly enabling RADIUS authorizations; therefore, this section contains no information about configuring RADIUS authorization on the FWSM. It does provide information about how the FWSM handles dynamic, user-specific access list information received from RADIUS servers.

You can configure a RADIUS server to download an access list to the FWSM or an access list name at the time of authentication. The user is authorized to do only what is permitted in the dynamic access list.

Note If you have used the **access-group** command to apply access lists to interfaces, be aware of the following effects of the **per-user-override** keyword on authorization by dynamic access lists:

- Without the **per-user-override** keyword, traffic for a user session must be permitted by both the interface access list and the dynamic access list.
- With the **per-user-override** keyword, the dynamic access list determines what is permitted.

For more information, see the **access-group** command entry in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

This section includes the following topics:

Configuring a RADIUS Server to Download Per-User Access Control Lists

This section describes how to configure Cisco Secure ACS or a third-party RADIUS server, and includes the following topics:

- Configuring Any RADIUS Server for Downloadable Access Lists, page 16-12

Configuring Cisco Secure ACS for Downloadable Access Lists

You can configure downloadable access lists on Cisco Secure ACS as a shared profile component and then assign the access list to a group or to an individual user.

The access list definition consists of one or more FWSM commands that are similar to the extended **access-list** command, except without the following prefix:

access-list *acl_name* **extended**

The following example is a downloadable access list definition on Cisco Secure ACS Version 3.3:

```
+-------------------------------------------+
| Shared profile Components                 |
|                                           |
|      Downloadable IP ACLs Content         |
|                                           |
| Name:    acs_ten_acl                      |
|                                           |
|      ACL Definitions                      |
|                                           |
| permit tcp any host 10.0.0.254            |
| permit udp any host 10.0.0.254            |
| permit icmp any host 10.0.0.254           |
| permit tcp any host 10.0.0.253            |
| permit udp any host 10.0.0.253            |
| permit icmp any host 10.0.0.253           |
| permit tcp any host 10.0.0.252            |
| permit udp any host 10.0.0.252            |
| permit icmp any host 10.0.0.252           |
| permit ip any any                         |
+-------------------------------------------+
```

For more information about creating downloadable access lists and associating them with users, see the user guide for your version of Cisco Secure ACS.

On the FWSM, the downloaded access list has the following name:

```
#ACSACL#-ip-acl_name-number
```

The *acl_name* argument is the name that is defined on Cisco Secure ACS (acs_ten_acl in the preceding example), and *number* is a unique version ID generated by Cisco Secure ACS.

The downloaded access list on the FWSM consists of the following lines:

```
access-list #ACSACL#-ip-xxx-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.254
access-list #ACSACL#-ip-xxx-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.254
access-list #ACSACL#-ip-xxx-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.254
access-list #ACSACL#-ip-xxx-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.253
access-list #ACSACL#-ip-xxx-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.253
access-list #ACSACL#-ip-xxx-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.253
access-list #ACSACL#-ip-xxx-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.252
access-list #ACSACL#-ip-xxx-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.252
access-list #ACSACL#-ip-xxx-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.252
access-list #ACSACL#-ip-xxx-acs_ten_acl-3b5385f7 permit ip any any
```

Configuring Any RADIUS Server for Downloadable Access Lists

You can configure any RADIUS server that supports Cisco IOS RADIUS VSAs to send dynamic access lists to the FWSM in a Cisco IOS RADIUS cisco-av-pair VSA (VSA number 1). Cisco IOS RADIUS VSAs are identified by RADIUS vendor ID 9.

In the cisco-av-pair VSA, configure one or more ACEs that are similar to the **access-list extended** command, except that you replace the following command prefix:

access-list *acl_name* **extended**

with the following text:

ip:inacl#*nnn***=**

The *nnn* argument is a number in the range from 0 to 999999999 that identifies the order of the command statement to be configured on the FWSM. If this parameter is omitted, the sequence value is 0, and the order of the ACEs inside the cisco-av-pair RADIUS VSA is used.

The following example is an access list definition as it should be configured for a cisco-av-pair VSA on a RADIUS server:

```
ip:inacl#1=permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#99=deny tcp any any
ip:inacl#2=permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#100=deny udp any any
ip:inacl#3=permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
```

For information about making unique per user the access lists that are sent in the cisco-av-pair attribute, see the documentation for your RADIUS server.

On the FWSM, the downloaded access list name has the following format:

```
AAA-user-username
```

The *username* argument is the name of the user that is being authenticated.

The downloaded access list on the FWSM consists of the following lines. Notice the order based on the numbers identified on the RADIUS server.

```
access-list  AAA-user-bcham34-79AD4A08 permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-bcham34-79AD4A08 permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-bcham34-79AD4A08 permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-bcham34-79AD4A08 deny tcp any any
access-list  AAA-user-bcham34-79AD4A08 deny udp any any
```

Downloaded access lists have two spaces between the word “access-list” and the name. These spaces serve to differentiate a downloaded access list from a local access list. In this example, “79AD4A08” is a hash value generated by the FWSM to help determine when access list definitions have changed on the RADIUS server.

Configuring a RADIUS Server to Download Per-User Access Control List Names

To download a name for an access list that you already created on the FWSM from the RADIUS server when a user authenticates, configure the IETF RADIUS filter-id attribute (attribute number 11) as follows:

filter-id=*acl_name*

Note In Cisco Secure ACS, the value for filter-id attributes are specified in boxes in the HTML interface, omitting **filter-id=** and entering only *acl_name*.

For information about making unique per user the filter-id attribute value, see the documentation for your RADIUS server.

See the "Adding an Extended Access List" section on page 12-6 to create an access list on the FWSM.

Configuring Accounting for Network Access

The FWSM can send accounting information to a RADIUS or TACACS+ server about any TCP or UDP traffic that passes through the FWSM. If that traffic is also authenticated, then the AAA server can maintain accounting information by username. If the traffic is not authenticated, the AAA server can maintain accounting information by IP address. Accounting information includes when sessions start and stop, username, the number of bytes that pass through the FWSM for the session, the service used, and the duration of each session.

To configure accounting, perform the following steps:

Step 1 If you want the FWSM to provide accounting data per user, you must enable authentication. For more information, see the "Enabling Network Access Authentication" section on page 16-3. If you want the FWSM to provide accounting data per IP address, enabling authentication is not necessary and you can continue to the next step.

Step 2 Using the **access-list** command, create an access list that identifies the source addresses and destination addresses of traffic you want accounted. For steps, see the "Adding an Extended Access List" section on page 12-6.

The **permit** ACEs mark matching traffic for authorization, while **deny** entries exclude matching traffic from authorization.

Note If you have configured authentication and want accounting data for all the traffic being authenticated, you can use the same access list you created for use with the **aaa authentication match** command.

Step 3 To enable accounting, enter the following command:

```
hostname(config)# aaa accounting match acl_name interface_name server_group
```

Note Alternatively, you can use the **aaa accounting include** command (which identifies traffic within the command) but you cannot use both methods in the same configuration. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

The following commands authenticate, authorize, and account for inside Telnet traffic. Telnet traffic to servers other than 209.165.201.5 can be authenticated alone, but traffic to 209.165.201.5 requires authorization and accounting.

```
hostname(config)# aaa-server AuthOutbound protocol tacacs+
hostname(config-aaa-server-group)# exit
hostname(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
hostname(config-aaa-server-host)# key TACPlusUauthKey
hostname(config-aaa-server-host)# exit
hostname(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
hostname(config)# access-list SERVER_AUTH extended permit tcp any host 209.165.201.5 eq
telnet
hostname(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
hostname(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
hostname(config)# aaa accounting match SERVER_AUTH inside AuthOutbound
```

Using MAC Addresses to Exempt Traffic from Authentication and Authorization

The FWSM can exempt traffic from specific MAC addresses from being authenticated or authorized. This feature is particularly useful to exempt devices such as IP phones that cannot respond to authentication prompts.

Note This feature exempts the list of MAC addresses for through-the-box connections only. For connections like Telnet to the FWSM, the authentication or authorization is not exempted even if the MAC address of the device is specified.

To identify MAC addresses for exemption, perform the following steps:

Step 1 To configure a MAC list, enter the following command:

```
hostname(config)# mac-list id {deny | permit} mac macmask
```

Where the *id* argument is the hexadecimal number that you assign to the MAC list.

To exempt a MAC address, use the **permit** keyword. To allow a MAC address to be authenticated and authorized, use the **deny** keyword.

To group a set of MAC addresses, enter the **mac-list** command as many times as needed with the same ID value. Because you can only use one MAC list for AAA exemption, be sure that your MAC list includes all the MAC addresses you want to exempt. You can create multiple MAC lists, but you can only use one at a time.

The order of entries matters, because the packet uses the first entry it matches, as opposed to a best match scenario. If you have a **permit** entry, and you want to deny an address that is allowed by the **permit** entry, be sure to enter the **deny** entry before the **permit** entry.

The *mac* argument specifies the source MAC address in 12-digit hexadecimal form; that is, nnnn.nnnn.nnnn.

The *macmask* argument specifies the portion of the MAC address that should be used for matching. For example, ffff.ffff.ffff matches the MAC address exactly. ffff.ffff.0000 matches only the first 8 digits.

Step 2 To exempt traffic for the MAC addresses specified in a particular MAC list, enter the following command:

```
hostname(config)# aaa mac-exempt match id
```

Where *id* is the string identifying the MAC list containing the MAC addresses whose traffic is to be exempt from authentication and authorization. You can only enter one instance of the **aaa mac-exempt** command.

The following example bypasses authentication for a single MAC address:

```
hostname(config)# mac-list abc permit 00a0.c95d.0282 ffff.ffff.ffff
hostname(config)# aaa mac-exempt match abc
```

The following entry bypasses authentication for all Cisco IP Phones, which have the hardware ID 0003.E3:

```
hostname(config)# mac-list acd permit 0003.E300.0000 FFFF.FF00.0000
hostname(config)# aaa mac-exempt match acd
```

The following example bypasses authentication for a a group of MAC addresses except for 00a0.c95d.02b2. Enter the **deny** statement before the **permit** statement, because 00a0.c95d.02b2 matches the **permit** statement as well, and if it is first, the **deny** statement will never be matched.

```
hostname(config)# mac-list 1 deny 00a0.c95d.0282 ffff.ffff.ffff
hostname(config)# mac-list 1 permit 00a0.c95d.0000 ffff.ffff.0000
hostname(config)# aaa mac-exempt match 1
```

CHAPTER 17

Applying Filtering Services

This chapter describes ways to filter web traffic to reduce security risks or prevent inappropriate use. This chapter includes the following sections:

Filtering Overview

This section describes how filtering can provide greater control over traffic passing through the FWSM. Filtering can be used in two ways:

- Filtering ActiveX objects or Java applets
- Filtering URLs with an external filtering server

Instead of blocking access altogether, you can remove specific undesirable objects from HTTP traffic, such as ActiveX objects or Java applets, that may pose a security threat in certain situations.

You can also use URL filtering to direct specific traffic to an external filtering server, such an Secure Computing SmartFilter (formerly N2H2) or Websense filtering server. Filtering servers can block traffic to specific sites or types of sites, as specified by the security policy.

Because URL filtering is CPU-intensive, using an external filtering server ensures that the throughput of other traffic is not affected. However, depending on the speed of your network and the capacity of your URL filtering server, the time required for the initial connection may be noticeably slower when filtering traffic with an external filtering server.

Filtering ActiveX Objects

This section describes how to apply filtering to remove ActiveX objects from HTTP traffic passing through the firewall. This section includes the following topics:

ActiveX Filtering Overview

ActiveX objects may pose security risks because they can contain code intended to attack hosts and servers on a protected network. You can disable ActiveX objects with ActiveX filtering.

ActiveX controls, formerly known as OLE or OCX controls, are components you can insert in a web page or other application. These controls include custom forms, calendars, or any of the extensive third-party forms for gathering or displaying information. As a technology, ActiveX creates many potential problems for network clients including causing workstations to fail, introducing network security problems, or being used to attack servers.

The **filter activex** command blocks the HTML <object> commands by commenting them out within the HTML web page. ActiveX filtering of HTML files is performed by selectively replacing the <APPLET> and </APPLET> and <OBJECT CLASSID> and </OBJECT> tags with comments. Filtering of nested tags is supported by converting top-level tags to comments.

Caution This command also blocks any Java applets, image files, or multimedia objects that are embedded in object tags.

If the <object> or </object> HTML tags split across network packets or if the code in the tags is longer than the number of bytes in the MTU, FWSM cannot block the tag.

ActiveX blocking does not occur when users access an IP address referenced by the **alias** command.

Enabling ActiveX Filtering

This section describes how to remove ActiveX objects in HTTP traffic passing through the FWSM. To remove ActiveX objects, enter the following command in global configuration mode:

```
hostname(config)# filter activex {port[-port] | except} local_ip local_mask foreign_ip
foreign_mask
```

To use this command, replace *port* with the TCP port to which filtering is applied. Typically, this is port 80, but other values are accepted. The **http** or **url** literal can be used for port 80. You can specify a range of ports by using a hyphen between the starting port number and the ending port number.

To create an exception to a previous filter condition, specify the keyword **except**.

Note The filter exception rule works only when you use the default port.

The local IP address and mask identify one or more internal hosts that are the source of the traffic to be filtered. The foreign address and mask specify the external destination of the traffic to be filtered.

You can set either address to **0.0.0.0** (or in shortened form, **0**) to specify all hosts. You can use **0.0.0.0** for either mask (or in shortened form, **0**) to specify all hosts.

The following example specifies that ActiveX objects are blocked on all outbound connections:

```
hostname(config)# filter activex 80 0 0 0 0
```

This command specifies that the ActiveX object blocking applies to web traffic on port 80 from any local host and for connections to any foreign host.

To remove the configuration, use the **no** form of the command, as in the following example:

```
hostname(config)# no filter activex 80 0 0 0 0
```

Filtering Java Applets

This section describes how to apply filtering to remove Java applets from HTTP traffic passing through the firewall. Java applets may pose security risks because they can contain code intended to attack hosts and servers on a protected network. You can remove Java applets with the **filter java** command.

The **filter java** command filters out Java applets that return to the FWSM from an outbound connection. The user still receives the HTML page, but the web page source for the applet is commented out so that the applet cannot execute.

Note Use the **filter activex** command to remove Java applets that are embedded in <object> tags.

To remove Java applets in HTTP traffic passing through the FWSM, enter the following command in global configuration mode:

```
hostname(config)# filter java {port[-port] | except} local_ip local_mask foreign_ip
foreign_mask
```

To use this command, replace *port* with the TCP port to which filtering is applied. Typically, this is port 80, but other values are accepted. The **http** or **url** literal can be used for port 80. You can specify a range of ports by using a hyphen between the starting port number and the ending port number.

To create an exception to a previous filter condition, specify the keyword **except**.

Note The filter exception rule works only when you use the default port.

The local IP address and mask identify one or more internal hosts that are the source of the traffic to be filtered. The foreign address and mask specify the external destination of the traffic to be filtered.

You can set either address to **0.0.0.0** (or in shortened form, **0**) to specify all hosts. You can use **0.0.0.0** for either mask (or in shortened form, **0**) to specify all hosts.

You can set either address to **0.0.0.0** (or in shortened form, **0**) to specify all hosts. You can use **0.0.0.0** for either mask (or in shortened form, **0**) to specify all hosts.

The following example specifies that Java applets are blocked on all outbound connections:

```
hostname(config)# filter java 80 0 0 0 0
```

This command specifies that the Java applet blocking applies to web traffic on port 80 from any local host and for connections to any foreign host.

The following example blocks downloading of Java applets to a host on a protected network:

```
hostname(config)# filter java http 192.168.3.3 255.255.255.255 0 0
```

This command prevents host 192.168.3.3 from downloading Java applets.

To remove the configuration, use the **no** form of the command, as in the following example:

```
hostname(config)# no filter java http 192.168.3.3 255.255.255.255 0 0
```

Filtering URLs and FTP Requests with an External Server

This section describes how to filter URLs and FTP requests with an external server. This section includes the following topics:

URL Filtering Overview

You can apply filtering to connection requests originating from a more secure network to a less secure network. Although you can use access lists to prevent outbound access to specific content servers, managing usage this way is difficult because of the size and dynamic nature of the Internet. You can simplify configuration and improve FWSM performance by using a separate server running one of the following Internet filtering products:

- Websense Enterprise for filtering HTTP, HTTPS, FTP, and long URL filtering.
- Secure Computing SmartFilter (formerly N2H2) for filtering HTTP, and HTTPS filtering.

Although FWSM performance is less affected when using an external server, users may notice longer access times to websites or FTP servers when the filtering server is remote from the FWSM.

When filtering is enabled and a request for content is directed through the FWSM, the request is sent to the content server and to the filtering server at the same time. If the filtering server allows the connection, the FWSM forwards the response from the content server to the originating client. If the filtering server denies the connection, the FWSM drops the response and sends a message or return code indicating that the connection was not successful.

If user authentication is enabled on the FWSM, then the FWSM also sends the username to the filtering server. The filtering server can use username filtering settings or provide enhanced reporting regarding usage.

Identifying the Filtering Server

You can identify up to four filtering servers per context. The FWSM uses the servers in order until a server responds. You can only configure a single type of server (Websense or N2H2) in your configuration.

Note You must add the filtering server before you can configure filtering for HTTP or HTTPS with the **filter** command. You must also remove all filtering command before you remove the filtering servers from the configuration.

Identify the address of the filtering server using the **url-server** command:

For Websense:

```
hostname(config)# url-server (if_name) vendor websense host local_ip [timeout seconds]
[protocol {TCP | UDP | connections num_conns] | version 4][context-name]
```

Note The **context-name** option is only available with websense version 4.0 and not with verion 1.0, and this feature can be configured only in multiple context mode.

For Secure Computing SmartFilter (formerly N2H2):

```
hostname(config)# url-server (if_name) vendor {smartfilter | n2h2} host
<local_ip> [port <number>] [timeout <seconds>] [protocol {TCP [connections <number>]} |
UDP]
```

where <*if_name*> is the name of the security appliance interface connected to the filtering server.

For the **vendor** {smartfilter | n2h2}, you can use ‘ssmartfilter as a vendor string, however, ‘n2h2’ is acceptable for backward compatibility. When the configuration entries are generated, ‘smartfilter’ is saved as the vendor string.

The **host** <*local_ip*> is the IP address of the URL filtering server.

The **port** <*number*> is the Secure Computing SmartFilter server port number of the filtering server; the FWSM also listens for UDP replies on this port.

Note The default port is 4005. This is the default port used by the Secure Computing SmartFilter server to communicate to the FWSM via TCP or UDP. For information on changing the default port, see the *Filtering by N2H2 Administrator's Guide*.

The **timeout** <*seconds*> is the number of seconds the security appliance should keep trying to connect to the filtering server.

The **connections** <*number*> is the number of tries to attempt to make a connection between the host and server.

Context-name sends each websense query for policy lookups on the websense server.

For example, to identify a single Websense filtering server, enter the following command:

```
hostname(config)# url-server (perimeter) host 10.0.1.1 protocol TCP version 4
```

This identifies a Websense filtering server with the IP address 10.0.1.1 on a perimeter interface of the FWSM.Version 4, which is enabled in this example, is recommended by Websense because it supports caching.

To identify redundant Secure Computing SmartFilter servers, enter the following commands:

```
hostname(config)# url-server (perimeter) vendor n2h2 host 10.0.1.1
hostname(config)# url-server (perimeter) vendor n2h2 host 10.0.1.2
```

This identifies two Sentian filtering servers, both on a perimeter interface of the FWSM.

Buffering the Content Server Response

When a user issues a request to connect to a content server, the FWSM sends the request to the content server and to the filtering server at the same time. If the filtering server does not respond before the content server, the server response is dropped. This delays the web server response from the point of view of the web client because the client must reissue the request.

By enabling the HTTP response buffer, replies from web content servers are buffered and the responses are forwarded to the requesting client if the filtering server allows the connection. This prevents the delay that might otherwise occur.

To configure buffering for responses to HTTP or FTP requests, perform the following steps:

Step 1 To enable buffering of responses for HTTP or FTP requests that are pending a response from the filtering server, enter the following command:

```
hostname(config)# url-block block block-buffer-limit
```

Replace *block-buffer-limit* with the maximum number of blocks that will be buffered.

Note Buffering URLs longer than 1159 bytes is only supported for the Websense filtering server.

Step 2 To configure the maximum memory available for buffering pending URLs (and for buffering long URLs with Websense), enter the following command:

```
hostname(config)# url-block url-mempool memory-pool-size
```

Replace *memory-pool-size* with a value from 2 to 10240 for a maximum memory allocation of 2 KB to 10 MB.

Caching Server Addresses

After a user accesses a site, the filtering server can allow the FWSM to cache the server address for a certain amount of time, as long as every site hosted at the address is in a category that is permitted at all times. Then, when the user accesses the server again, or if another user accesses the server, the FWSM does not need to consult the filtering server again.

Note Requests for cached IP addresses are not passed to the filtering server and are not logged. As a result, this activity does not appear in any reports. You can accumulate Websense run logs before using the **url-cache** command.

Use the **url-cache** command if needed to improve throughput, as follows:

```
hostname(config)# url-cache {dst | src_dst} size
```

Replace *size* with a value for the cache size within the range 1 to 128 (KB).

Use the **dst** keyword to cache entries based on the URL destination address. Select this mode if all users share the same URL filtering policy on the Websense server.

Use the **src_dst** keyword to cache entries based on both the source address initiating the URL request as well as the URL destination address. Select this mode if users do not share the same URL filtering policy on the Websense server.

Filtering HTTP URLs

This section describes how to configure HTTP filtering with an external filtering server. This section includes the following topics:

Configuring HTTP Filtering

You must identify and enable the URL filtering server before enabling HTTP filtering.

When the filtering server approves an HTTP connection request, the FWSM allows the reply from the web server to reach the originating client. If the filtering server denies the request, the FWSM redirects the user to a block page, indicating that access was denied.

To enable HTTP filtering, enter the following command:

```
hostname(config)# filter url {http | port[-port] | except} local_ip local_mask foreign_ip
foreign_mask [allow][cgi-truncate][longurl-deny][longurl-truncate][proxy-block]
```

Replace *port* with one or more port numbers if a different port than the default port for HTTP (80) is used. Replace *local_ip* and *local_mask* with the IP address and subnet mask of a user or subnetwork making requests. Replace *foreign_ip* and *foreign_mask* with the IP address and subnet mask of a server or subnetwork responding to requests.

To create an exception to a previous filter condition, specify the keyword **except**.

Note The filter exception rule works only when you use the default port.

The **allow** option causes the FWSM to forward HTTP traffic without filtering when the primary filtering server is unavailable. Use the **proxy-block** command to drop all requests to proxy servers.

Enabling Filtering of Long HTTP URLs

By default, the FWSM considers an HTTP URL to be a long URL if it is greater than 1159 characters. For Websense servers, you can increase the maximum length allowed.

(Websense only) Configure the maximum size of a single URL with the following command:

```
hostname(config)# url-block url-size long_url_size
```

Replace *long_url_size* with a value from 2 to 4 for a maximum URL size of 2 KB to 4 KB. The default value is 2.

(Websense only) You can also configure the maximum size of the URL buffer memory pool with the following command:

```
hostname(config)# url-block url-mempool memory_pool_size
```

Replace *memory_pool_size* with a value from 2 to 10240 for a URL buffer memory pool size of 2 KB to 10,240 KB.

Truncating Long HTTP URLs

By default, if a URL exceeds the maximum permitted size, then it is dropped. To avoid this, you can set the FWSM to truncate a long URL by entering the following command:

```
hostname(config)# filter url [longurl-truncate | longurl-deny | cgi-truncate]
```

The **longurl-truncate** option causes the FWSM to send only the hostname or IP address portion of the URL for evaluation to the filtering server when the URL is longer than the maximum length permitted. Use the **longurl-deny** option to deny outbound URL traffic if the URL is longer than the maximum permitted.

Use the **cgi-truncate** option to truncate CGI URLs to include only the CGI script location and the script name without any parameters. Many long HTTP requests are CGI requests. If the parameters list is very long, waiting and sending the complete CGI request including the parameter list can use up memory resources and affect firewall performance.

Exempting Traffic from Filtering

To exempt specific traffic from filtering, enter the following command:

```
hostname(config)# filter url except source_ip source_mask dest_ip dest_mask
```

For example, the following commands cause all HTTP requests to be forwarded to the filtering server except for those from 10.0.2.54.

```
hostname(config)# filter url http 0 0 0 0
hostname(config)# filter url except 10.0.2.54 255.255.255.255 0 0
```

Note If you have the **filter java except** command configured and the **filter activex** command configured for the same source destination pair, then no filtering will occur on port 80 for this source destination pair.

Filtering HTTPS URLs

You must identify and enable the URL filtering server before enabling HTTPS filtering.

Because HTTPS content is encrypted, the FWSM sends the URL lookup without directory and filename information. When the filtering server approves an HTTPS connection request, the FWSM allows the completion of SSL connection negotiation and allows the reply from the web server to reach the originating client. If the filtering server denies the request, the FWSM prevents the completion of SSL connection negotiation. The browser displays an error message such as “The Page or the content cannot be displayed.”

Note The FWSM does not provide an authentication prompt for HTTPS, so a user must authenticate with the FWSM using HTTP or FTP before accessing HTTPS servers.

To enable HTTPS filtering, enter the following command:

```
hostname(config)# filter https port localIP local_mask foreign_IP foreign_mask [allow]
```

Replace *port* with the port number if a different port than the default port for HTTPS (443) is used. The filter exception rule works only when you use the default port.

Note Because both HTTPS and HTTP traffic have the same GET request, the HTTPS protocol inspector will also filter HTTP traffic on the port number that you specify.

Replace *local_ip* and *local_mask* with the IP address and subnet mask of a user or subnetwork making requests. Replace *foreign_ip* and *foreign_mask* with the IP address and subnet mask of a server or subnetwork responding to requests.

The **allow** option causes the FWSM to forward HTTPS traffic without filtering when the primary filtering server is unavailable.

Filtering FTP Requests

You must identify and enable the URL filtering server before enabling FTP filtering.

Note Secure Computing SmartFilter (formerly known as N2H2) does not support FTP filtering.

When the filtering server approves an FTP connection request, the FWSM allows the successful FTP return code to reach originating client. For example, a successful return code is "250: CWD command successful." If the filtering server denies the request, alters the FTP return code to show that the connection was denied. For example, the FWSM changes code 250 to "550 Requested file is prohibited by URL filtering policy."

To enable FTP filtering, enter the following command:

```
hostname(config)# filter ftp {port[-port] | except} localIP local_mask foreign_IP
foreign_mask [allow] [interact-block]
```

Replace *port* with the port number if a different port than the default port for FTP (21) is used. Replace *local_ip* and *local_mask* with the IP address and subnet mask of a user or subnetwork making requests. Replace *foreign_ip* and *foreign_mask* with the IP address and subnet mask of a server or subnetwork responding to requests.

To create an exception to a previous filter condition, specify the keyword **except**.

Note The filter exception rule works only when you use the default port.

The **allow** option causes the FWSM to forward FTP traffic without filtering when the primary filtering server is unavailable.

Use the **interact-block** option to prevent interactive FTP sessions that do not provide the entire directory path. An interactive FTP client allows the user to change directories without typing the entire path. For example, the user might enter **cd ./files** instead of **cd /public/files**.

Viewing Filtering Statistics and Configuration

This section describes how to monitor filtering statistics. This section includes the following topics:

Viewing Filtering Server Statistics

To show information about the filtering server, enter the following command:

```
hostname# show running-config url-server
```

The following is sample output from the **show running-config url-server** command:

```
hostname# show running-config url-server
url-server (outside) vendor n2h2 host 128.107.254.202 port 4005 timeout 5 protocol TCP
```

To show information about the filtering server or to show statistics, enter the following command:

```
hostname# show url-server statistics
```

The following is sample output from the **show url-server statistics** command, which shows filtering statistics:

```
hostname# show url-server statistics
URL Server Statistics:
----------------------
Vendor                          websense
URLs total/allowed/denied       50/35/15
HTTPSs total/allowed/denied     1/1/0
FTPs total/allowed/denied       3/1/2

URL Server Status:
------------------
10.130.28.18            UP

URL Packets Sent and Received Stats:
------------------------------------
Message                 Sent    Received
STATUS_REQUEST          65155   34773
LOOKUP_REQUEST          0       0
LOG_REQUEST             0       NA
------------------------------------
```

Viewing Buffer Configuration and Statistics

The **show running-config url-block** command displays the number of packets held in the url-block buffer and the number (if any) dropped due to exceeding the buffer limit or retransmission.

The following is sample output from the **show running-config url-block** command:

```
hostname# show running-config url-block
    url-block url-mempool 128
    url-block url-size 4
    url-block block 128
```

This shows the configuration of the URL block buffer.

The following is sample output from the **show url-block block statistics** command:

```
hostname# show url-block block statistics

URL Pending Packet Buffer Stats with max block  128
-----------------------------------------------------
Cumulative number of packets held:              896
Maximum number of packets held (per URL):       3
Current number of packets held (global):        38
Packets dropped due to
       exceeding url-block buffer limit:        7546
       HTTP server retransmission:              10
Number of packets released back to client:      0
```

This shows the URL block statistics.

Viewing Caching Statistics

The following is sample output from the **show url-cache** command:

```
hostname# show url-cache
URL Filter Cache Stats
----------------------
    Size :     128KB
 Entries :      1724
  In Use :       456
 Lookups :        45
    Hits :         8

This shows how the cache is used.
```

Viewing Filtering Performance Statistics

The following is sample output from the **show perfmon** command:

```
hostname# show perfmon
PERFMON STATS:    Current      Average
Xlates               0/s          0/s
Connections          0/s          2/s
TCP Conns            0/s          2/s
UDP Conns            0/s          0/s
URL Access           0/s          2/s
URL Server Req       0/s          3/s
TCP Fixup            0/s          0/s
TCPIntercept         0/s          0/s
HTTP Fixup           0/s          3/s
FTP Fixup            0/s          0/s
AAA Authen           0/s          0/s
AAA Author           0/s          0/s
AAA Account          0/s          0/s
```

This shows URL filtering performance statistics, along with other performance statistics. The filtering statistics are shown in the URL Access and URL Server Req rows.

Viewing Filtering Configuration

The following is sample output from the **show running-config filter** command:

```
hostname# show running-config filter
filter url http 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
```

CHAPTER 18

Configuring ARP Inspection and Bridging Parameters

Transparent Firewall Mode Only

This chapter describes how to enable ARP inspection and how to customize bridging operations for the FWSM. In multiple context mode, the commands in this chapter can be entered in a security context, but not the system.

This chapter includes the following sections:

Configuring ARP Inspection

This section describes ARP inspection and how to enable it, and includes the following topics:

ARP Inspection Overview

By default, all ARP packets are allowed through the FWSM. You can control the flow of ARP packets by enabling ARP inspection. ARP inspection settings apply to all bridge groups.

When you enable ARP inspection, the FWSM compares the MAC address, IP address, and source interface in all ARP packets to static entries in the ARP table, and takes the following actions:

- If the IP address, MAC address, and source interface match an ARP entry, the packet is passed through.
- If there is a mismatch between the MAC address, the IP address, or the interface, then the FWSM drops the packet.
- If the ARP packet does not match any entries in the static ARP table, then you can set the FWSM to either forward the packet out all interfaces (flood), or to drop the packet.

ARP inspection prevents malicious users from impersonating other hosts or routers (known as ARP spoofing). ARP spoofing can enable a "man-in-the-middle" attack. For example, a host sends an ARP request to the gateway router; the gateway router responds with the gateway router MAC address. The attacker, however, sends another ARP response to the host with the attacker MAC address instead of the router MAC address. The attacker can now intercept all the host traffic before forwarding it on to the router.

ARP inspection ensures that an attacker cannot send an ARP response with the attacker MAC address, so long as the correct MAC address and the associated IP address are in the static ARP table.

Adding a Static ARP Entry

ARP inspection compares ARP packets with static ARP entries in the ARP table. To add a static ARP entry, enter the following command:

```
hostname(config)# arp interface_name ip_address mac_address
```

Where the *interface_name* is the source interface for the ARP packets. The *ip_address* is the source address, and *mac_address* is the associated MAC address.

For example, to allow ARP responses from the router at 10.1.1.1 with the MAC address 0009.7cbe.2100 on the outside interface, enter the following command:

```
hostname(config)# arp outside 10.1.1.1 0009.7cbe.2100
```

Note The transparent firewall uses dynamic ARP entries in the ARP table for traffic to and from the FWSM, such as management traffic.

Enabling ARP Inspection

To enable ARP inspection, enter the following command:

```
hostname(config)# arp-inspection interface_name enable [flood | no-flood]
```

Where the *interface_name* is the interface on which you want to enable ARP inspection. The **flood** keyword forwards non-matching ARP packets out all interfaces, and **no-flood** drops non-matching packets.

Note The default setting is to flood non-matching packets. To restrict ARP through the FWSM to only static entries, then set this command to **no-flood**.

For example, to enable ARP inspection on the outside interface, and to drop all non-matching ARP packets, enter the following command:

```
hostname(config)# arp-inspection outside enable no-flood
```

To view the current settings for ARP inspection on all interfaces, enter the **show arp-inspection** command.

Customizing the MAC Address Table

This section describes the MAC address table, and includes the following topics:

MAC Address Table Overview

The FWSM learns and builds a MAC address table in a similar way as a normal bridge or switch: when a device sends a packet through the FWSM, the FWSM adds the MAC address to its table. The table associates the MAC address with the source interface and a bridge group so that the FWSM knows to send any packets addressed to the device out the correct interface. A MAC address can have more than one entry in the table if it sent traffic through more than one bridge group. When the FWSM needs to determine the egress interface to deliver a packet to that MAC address, then the FWSM uses the entry for the bridge group that contains the ingress interface for the packet.

Because the FWSM is a firewall, if the destination MAC address of a packet is not in the table, the FWSM does not flood the original packet on all interfaces of the bridge group as a normal bridge does. Instead, it generates the following packets for directly connected devices or for remote devices:

- Packets for directly connected devices—The FWSM generates an ARP request for the destination IP address, so that the FWSM can learn which interface receives the ARP response.
- Packets for remote devices—The FWSM generates a ping to the destination IP address so that the FWSM can learn which interface receives the ping reply.

The original packet is dropped.

Adding a Static MAC Address

Normally, MAC addresses are added to the MAC address table dynamically as traffic from a particular MAC address enters an interface. You can add static MAC addresses to the MAC address table if desired. One benefit to adding static entries is to guard against MAC spoofing. If a client with the same MAC address as a static entry attempts to send traffic to an interface that does not match the static entry, then the FWSM drops the traffic and generates a system log message.

To add a static MAC address to the MAC address table, enter the following command:

```
hostname(config)# mac-address-table static interface_name mac_address
```

The *interface_name* is the source interface.

Setting the MAC Address Timeout

The default timeout value for dynamic MAC address table entries is 5 minutes, but you can change the timeout. To change the timeout, enter the following command:

```
hostname(config)# mac-address-table aging-time timeout_value
```

The *timeout_value* (in minutes) is between 5 and 720 (12 hours). 5 minutes is the default.

Disabling MAC Address Learning

By default, each interface automatically learns the MAC addresses of entering traffic, and the FWSM adds corresponding entries to the MAC address table. You can disable MAC address learning if desired, however, unless you statically add MAC addresses to the table, no traffic can pass through the FWSM.

To disable MAC address learning, enter the following command:

```
hostname(config)# mac-learn interface_name disable
```

The **no** form of this command reenables MAC address learning. The **clear configure mac-learn** command reenables MAC address learning on all interfaces.

Viewing the MAC Address Table

You can view the entire MAC address table (including static and dynamic entries), the MAC address table for an interface, or the MAC address table for a bridge group. To view the MAC address table, enter the following command:

```
hostname# show mac-address-table [interface_name | bridge_group]
```

The following is sample output from the **show mac-address-table** command that shows the entire table:

```
hostname# show mac-address-table
interface          mac address       type      Age min)   Group
-----------------------------------------------------------------------
outside            0009.7cbe.2100    static      -        Eng
inside             0010.7cbe.6101    static      -        Eng
inside             0009.7cbe.5101    dynamic     10       Eng
```

The following is sample output from the **show mac-address-table** command that shows the table for the inside interface:

```
hostname# show mac-address-table inside
interface          mac address       type      Age min)   Group
-----------------------------------------------------------------------
inside             0010.7cbe.6101    static      -        Eng
inside             0009.7cbe.5101    dynamic     10       Eng
```

CHAPTER 19

Using Modular Policy Framework

This chapter describes how to use Modular Policy Framework to create security policies for supported features. This chapter includes the following sections:

Information About Modular Policy Framework

Modular Policy Framework provides a consistent and flexible way to configure FWSM features. For example, you can use Modular Policy Framework to create a timeout configuration that is specific to a particular TCP application, as opposed to one that applies to all TCP applications. This section includes the following topics:

Modular Policy Framework Supported Features

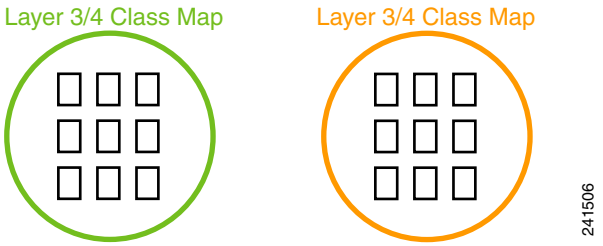
Modular Policy Framework supports the following features:

- TCP and UDP connection settings, TCP sequence number randomization, and TCP state bypass—See the "Configuring Connection Limits and Timeouts" section on page 20-1, and "Configuring TCP State Bypass" section on page 20-10.
- Application inspection—See Chapter 21, "Applying Application Layer Protocol Inspection."
- Permitting or Denying Application Types with PISA Integration—See the "Permitting or Denying Application Types with PISA Integration" section on page 20-4.

Modular Policy Framework Configuration Overview

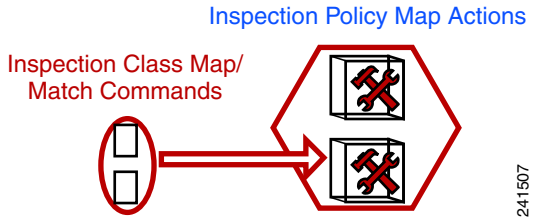
Configuring Modular Policy Framework consists of the following tasks:

1. Identify the traffic on which you want to perform Modular Policy Framework actions by creating Layer 3/4 class maps. For example, you might want to perform actions on all traffic that passes through the FWSM; or you might only want to perform certain actions on traffic from 10.1.1.0/24 to any destination address.



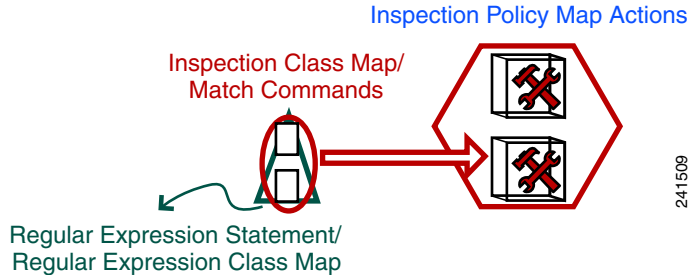
   See the "Identifying Traffic (Layer 3/4 Class Map)" section on page 19-4.

2. If one of the actions you want to perform is application inspection, and you want to perform additional actions on some inspection traffic, then create an inspection policy map. The inspection policy map identifies the traffic and specifies what to do with it. For example, you might want to drop all HTTP requests with a body length greater than 1000 bytes.



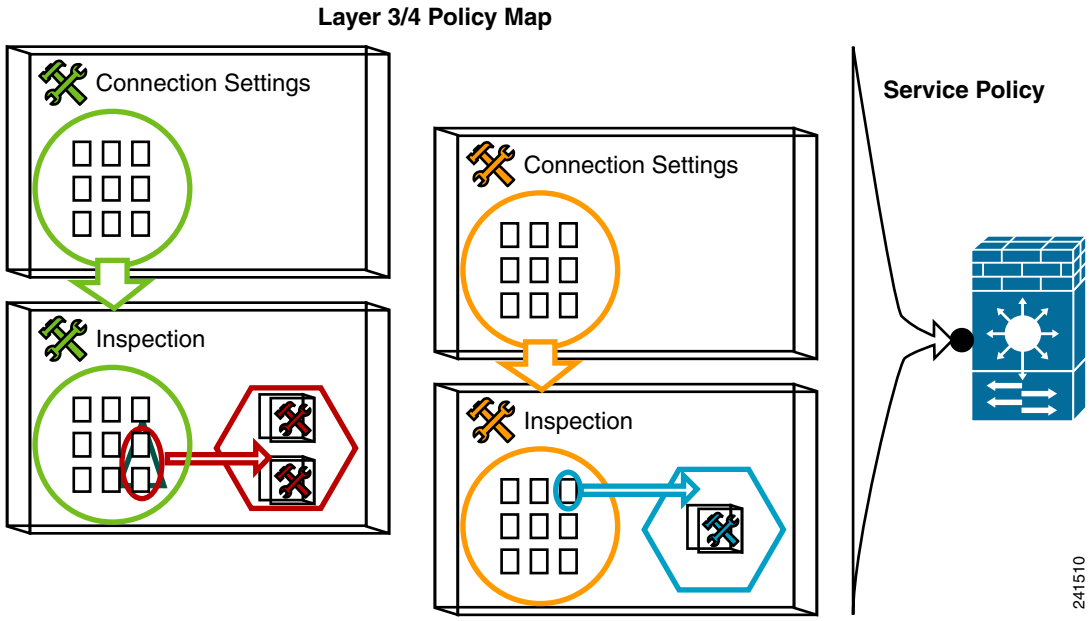
   You can create a self-contained inspection policy map that identifies the traffic directly with **match** commands, or you can create an inspection class map for reuse or for more complicated matching. See the "Defining Actions in an Inspection Policy Map" section on page 19-7 and the "Identifying Traffic in an Inspection Class Map" section on page 19-10.

3. If you want to match text with a regular expression within inspected packets, you can create a regular expression or a group of regular expressions (a regular expression class map). Then, when you define the traffic to match for the inspection policy map, you can call on an existing regular expression. For example, you might want to drop all HTTP requests with a URL including the text "example.com."



See the "Creating a Regular Expression" section on page 19-11 and the "Creating a Regular Expression Class Map" section on page 19-14.

4. Define the actions you want to perform on each Layer 3/4 class map by creating a Layer 3/4 policy map. Then, determine on which interfaces you want to apply the policy map using a service policy.



See the "Defining Actions (Layer 3/4 Policy Map)" section on page 19-14 and the "Applying Actions to an Interface (Service Policy)" section on page 19-20.

Default Global Policy

By default, the configuration includes a policy that matches all default application inspection traffic and applies certain inspections to the traffic on all interfaces (a global policy). Not all inspections are enabled by default. You can only apply one global policy, so if you want to alter the global policy, you need to either edit the default policy or disable it and apply a new one. (An interface policy overrides the global policy.)

The default policy configuration includes the following commands:

```
class-map inspection_default
 match default-inspection-traffic
policy-map global_policy
 class inspection_default
  inspect dns maximum-length 512
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect netbios
  inspect rsh
  inspect skinny
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
service-policy global_policy global
```

Note See the "Incompatibility of Certain Feature Actions" section on page 19-17 for more information about the special **match default-inspection-traffic** command used in the default class map.

Identifying Traffic (Layer 3/4 Class Map)

A Layer 3/4 class map identifies Layer 3 and 4 traffic to which you want to apply actions. You can create multiple Layer 3/4 class maps for each Layer 3/4 policy map.

This section includes the following topics:

Default Class Maps

The configuration includes many internally-created default class maps, including a default Layer 3/4 class map that the FWSM uses in the default global policy. It is called **inspection_default** and matches the default inspection traffic:

```
class-map inspection_default
 match default-inspection-traffic
```

Note See the "Incompatibility of Certain Feature Actions" section on page 19-17 for more information about the special **match default-inspection-traffic** command used in the default class map.

Another class map that exists in the default configuration is called class-default, and it matches all traffic:

```
class-map class-default
 match any
```

This class map appears at the end of all Layer 3/4 policy maps and essentially tells the FWSM to not perform any actions on all other traffic. You can use the class-default class map if desired, rather than making your own **match any** class map.

Default class maps also include inspection class maps.

To view all default class maps, as well as any user-created class maps, enter the **show running-config all class-map** command.

Maximum Class Maps

The maximum number of class maps of all types is 255 in single mode or per context in multiple mode. Class maps include the following types:

- Layer 3/4 class maps
- Inspection class maps
- Regular expression class maps
- **match** commands used directly underneath an inspection policy map

This limit also includes default class maps of all types. See the "Default Class Maps" section on page 19-4.

Creating a Layer 3/4 Class Map for Through Traffic

A Layer 3/4 class map matches traffic based on protocols, ports, IP addresses and other Layer 3 or 4 attributes.

To define a Layer 3/4 class map, perform the following steps:

Step 1 Create a Layer 3/4 class map by entering the following command:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

Where *class_map_name* is a string up to 40 characters in length. The name "class-default" is reserved. All types of class maps use the same name space, so you cannot reuse a name already used by another type of class map. The CLI enters class-map configuration mode.

Step 2 (Optional) Add a description to the class map by entering the following command:

```
hostname(config-cmap)# description string
```

Step 3 Define the traffic to include in the class by matching one of the following characteristics. Unless otherwise specified, you can include only one **match** command in the class map.

- Any traffic—The class map matches all traffic.

  ```
  hostname(config-cmap)# match any
  ```

- Access list—The class map matches traffic specified by an extended access list. If the FWSM is operating in transparent firewall mode, you can use an EtherType access list.

  ```
  hostname(config-cmap)# match access-list access_list_name
  ```

  For more information about creating access lists, see the "Adding an Extended Access List" section on page 12-6 or the "Adding an EtherType Access List" section on page 12-9.

  For information about creating access lists with NAT, see the "IP Addresses Used for Access Lists When You Use NAT" section on page 12-3.

- TCP or UDP destination ports—The class map matches a single port or a contiguous range of ports.

  ```
  hostname(config-cmap)# match port {tcp | udp} {eq port_num | range port_num port_num}
  ```

Tip For applications that use multiple, non-contiguous ports, use the **match access-list** command and define an ACE to match each port.

  For a list of ports you can specify, see the "TCP and UDP Ports" section on page E-11.

  For example, enter the following command to match TCP packets on port 80 (HTTP):

  ```
  hostname(config-cmap)# match tcp eq 80
  ```

- Default traffic for inspection—The class map matches the default TCP and UDP ports used by all applications that the FWSM can inspect.

  ```
  hostname(config-cmap)# match default-inspection-traffic
  ```

This command, which is used in the default global policy, is a special CLI shortcut that when used in a policy map, ensures that the correct inspection is applied to each packet, based on the destination port of the traffic. For example, when UDP traffic for port 69 reaches the FWSM, then the FWSM applies the TFTP inspection; when TCP traffic for port 21 arrives, then the FWSM applies the FTP inspection. So in this case only, you can configure multiple inspections for the same class map (with the exception of WAAS inspection, which can be configured with other inspections. See the "Incompatibility of Certain Feature Actions" section on page 19-17 for more information about combining actions). Normally, the FWSM does not use the port number to determine the inspection applied, thus giving you the flexibility to apply inspections to non-standard ports, for example.

See the "Default Inspection Policy" section on page 21-4 for a list of default ports. The FWSM includes a default global policy that matches the default inspection traffic, and applies common inspections to the traffic on all interfaces. Not all applications whose ports are included in the **match default-inspection-traffic** command are enabled by default in the policy map.

You can specify a **match access-list** command along with the **match default-inspection-traffic** command to narrow the matched traffic. Because the **match default-inspection-traffic** command specifies the ports and protocols to match, any ports or protocols in the access list are ignored.

The following is an example for the **class-map** command:

```
hostname(config)# access-list udp permit udp any any
hostname(config)# access-list tcp permit tcp any any
hostname(config)# access-list host_foo permit ip any 10.1.1.1 255.255.255.255

hostname(config)# class-map all_udp
hostname(config-cmap)# description "This class-map matches all UDP traffic"
hostname(config-cmap)# match access-list udp

hostname(config-cmap)# class-map all_tcp
hostname(config-cmap)# description "This class-map matches all TCP traffic"
hostname(config-cmap)# match access-list tcp

hostname(config-cmap)# class-map all_http
hostname(config-cmap)# description "This class-map matches all HTTP traffic"
hostname(config-cmap)# match port tcp eq http

hostname(config-cmap)# class-map to_server
hostname(config-cmap)# description "This class-map matches all traffic to server 10.1.1.1"
hostname(config-cmap)# match access-list host_foo
```

Configuring Special Actions for Application Inspections (Inspection Policy Map)

Modular Policy Framework lets you configure special actions for many application inspections. When you enable an inspection engine in the Layer 3/4 policy map, you can also optionally enable actions as defined in an *inspection policy map*. When the inspection policy map matches traffic within the Layer 3/4 class map for which you have defined an inspection action, then that subset of traffic will be acted upon as specified (for example, dropped or rate-limited).

This section includes the following topics:

Inspection Policy Map Overview

See the "Inspection Engine Overview" section on page 21-2 for a list of applications that support inspection policy maps.

An inspection policy map consists of one or more of the following elements. The exact options available for an inspection policy map depends on the application.

- Traffic matching command—You can define a traffic matching command directly in the inspection policy map to match application traffic to criteria specific to the application, such as a URL string, for which you then enable actions.
  - Some traffic matching commands can specify regular expressions to match text inside a packet. Be sure to create and test the regular expressions before you configure the policy map, either singly or grouped together in a regular expression class map.
- Inspection class map—(Not available for all applications. See the CLI help for a list of supported applications.) An inspection class map includes traffic matching commands that match application traffic with criteria specific to the application, such as a URL string. You then identify the class map in the policy map and enable actions. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that you can create more complex match criteria and you can reuse class maps.
  - Some traffic matching commands can specify regular expressions to match text inside a packet. Be sure to create and test the regular expressions before you configure the policy map, either singly or grouped together in a regular expression class map.
- Parameters—Parameters affect the behavior of the inspection engine.

Note There are default inspection policy maps such as **policy-map type inspect esmtp _default_esmtp_map**. These default policy maps are created implicitly by the command **inspect** *protocol*. For example, **inspect esmtp** implicitly uses the policy map "_default_esmtp_map." All the default policy maps can be shown by using the **show running-config all policy-map** command.

Defining Actions in an Inspection Policy Map

When you enable an inspection engine in the Layer 3/4 policy map, you can also optionally enable actions as defined in an inspection policy map.

To create an inspection policy map, perform the following steps:

Step 1 (Optional) Create an inspection class map according to the "Identifying Traffic in an Inspection Class Map" section on page 19-10. Alternatively, you can identify the traffic directly within the policy map.

Step 2 To create the inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect application policy_map_name
hostname(config-pmap)#
```

See the "Configuring Application Inspection" section on page 21-6 for a list of applications that support inspection policy maps.

The *policy_map_name* argument is the name of the policy map up to 40 characters in length. All types of policy maps use the same name space, so you cannot reuse a name already used by another type of policy map. The CLI enters policy-map configuration mode.

Step 3 To apply actions to matching traffic, perform the following steps:

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the inspection class map that you created in the "Identifying Traffic in an Inspection Class Map" section on page 19-10 by entering the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

  Not all applications support inspection class maps.

- Specify traffic directly in the policy map using one of the **match** commands described for each application in Chapter 21, "Applying Application Layer Protocol Inspection." If you use a **match not** command, then any traffic that matches the criterion in the **match not** command does not have the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop | drop-connection | mask | reset] [log] | log}
```

Not all options are available for each application. Other actions specific to the application might also be available. See Chapter 21, "Applying Application Layer Protocol Inspection." for the exact options available.

The **drop** keyword drops all packets that match.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

Note You can specify multiple **class** or **match** commands in the policy map.

If a packet matches multiple different **match** or **class** commands, then the order in which the FWSM applies the actions is determined by internal FWSM rules, and not by the order they are added to the policy map. The internal rules are determined by the application type and the logical progression of parsing a packet, and are not user-configurable. For example for HTTP traffic, parsing a Request Method field precedes parsing the Header Host Length field; an action for the Request Method field occurs before the action for the Header Host Length field. For example, the following match commands can be entered in any order, but the **match request method get** command is matched first.

```
match request header host length gt 100
  reset
match request method get
  log
```

If an action drops a packet, then no further actions are performed in the inspection policy map. For example, if the first action is to reset the connection, then it will never match any further **match** or **class** commands. If the first action is to log the packet, then a second action, such as resetting the connection, can occur. (You can configure both the **reset** (or **drop-connection**, and so on.) and the **log** action for the same **match** or **class** command, in which case the packet is logged before it is reset for a given match.)

If a packet matches multiple **match** or **class** commands that are the same, then they are matched in the order they appear in the policy map. For example, for a packet with the header length of 1001, it will match the first command below, and be logged, and then will match the second command and be reset. If you reverse the order of the two **match** commands, then the packet will be dropped and the connection reset before it can match the second **match** command; it will never be logged.

```
match request header length gt 100
  log
match request header length gt 1000
  reset
```

A class map is determined to be the same type as another class map or **match** command based on the lowest priority **match** command in the class map (the priority is based on the internal rules). If a class map has the same type of lowest priority **match** command as another class map, then the class maps are matched according to the order they are added to the policy map. If the lowest priority command for each class map is different, then the class map with the higher priority **match** command is matched first. For example, the following three class maps contain two types of **match** commands: **match content length** (higher priority) and **match content type** (lower priority). The sip3 class map includes both commands, but it is ranked according to the lowest priority command, **match content type**. The sip1 class map includes the highest priority command, so it is matched first, regardless of the order in the policy map. The sip3 class map is ranked as being of the same priority as the sip2 class map, which also contains the **match content type** command. They are matched according to the order in the policy map: sip3 and then sip2.

```
class-map inspect type sip match-all sip1
  match content length gt 1000
class-map inspect type sip match-all sip2
  match content type sdp
class-map inspect type sip match-all sip3
  match content length gt 1000
  match content type sdp

policy-map type inspect sip sip
  class sip3
    log
  class sip2
    log
  class sip1
    log
```

Step 4 To configure parameters that affect the inspection engine, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

The CLI enters parameters configuration mode. For the parameters available for each application, see Chapter 21, “Applying Application Layer Protocol Inspection.”

The following is an example of an HTTP inspection policy map and the related class maps. This policy map is activated by the Layer 3/4 policy map, which is enabled by the service policy.

```
hostname(config)# regex url_example example\.com
hostname(config)# regex url_example2 example2\.com
hostname(config)# class-map type regex match-any URLs
hostname(config-cmap)# match regex url_example
hostname(config-cmap)# match regex url_example2

hostname(config-cmap)# class-map type inspect http match-all http-traffic
hostname(config-cmap)# match req-resp content-type mismatch
hostname(config-cmap)# match request body length gt 1000
hostname(config-cmap)# match not request uri regex class URLs

hostname(config-cmap)# policy-map type inspect http http-map1
hostname(config-pmap)# class http-traffic
hostname(config-pmap-c)# drop-connection log
hostname(config-pmap-c)# match req-resp content-type mismatch
hostname(config-pmap-c)# reset log
hostname(config-pmap-c)# parameters
hostname(config-pmap-p)# protocol-violation action log

hostname(config-pmap-p)# policy-map test
hostname(config-pmap)# class test (a Layer 3/4 class map not shown)
hostname(config-pmap-c)# inspect http http-map1

hostname(config-pmap-c)# service-policy test interface outside
```

Identifying Traffic in an Inspection Class Map

This type of class map allows you to match criteria that is specific to an application. For example, for HTTP traffic, you can match a particular URL.

Note Not all applications support inspection class maps. See the CLI help for a list of supported applications.

A class map groups multiple traffic matches (in a match-all class map), or lets you match any of a list of matches (in a match-any class map). The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you group multiple match commands, and you can reuse class maps. For the traffic that you identify in this class map, you can specify actions such as dropping, resetting, and/or logging the connection in the inspection policy map. If you want to perform different actions on different types of traffic, you should identify the traffic directly in the policy map.

The maximum number of class maps (Layer 3/4, inspection, and regular expression) is 255 in single mode or per context in multiple mode. This limit includes default class maps. See the "Default Class Maps" section on page 19-4.

To define an inspection class map, perform the following steps:

Step 1 (Optional) If you want to match based on a regular expression, see the "Creating a Regular Expression" section on page 19-11 and the "Creating a Regular Expression Class Map" section on page 19-14.

Step 2 Create a class map by entering the following command:

```
hostname(config)# class-map type inspect application [match-all] class_map_name
hostname(config-cmap)#
```

Where the *application* is the application you want to inspect. For supported applications, see the CLI help for a list of supported applications or see Chapter 21, "Applying Application Layer Protocol Inspection."

The *class_map_name* argument is the name of the class map up to 40 characters in length.

The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map.

The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

Step 3 (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

Step 4 Define the traffic to include in the class by entering one or more **match** commands available for your application.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

To see the **match** commands available for each application, see Chapter 21, "Applying Application Layer Protocol Inspection."

The following example creates an HTTP class map that must match all criteria:

```
hostname(config-cmap)# class-map type inspect http match-all http-traffic
hostname(config-cmap)# match req-resp content-type mismatch
hostname(config-cmap)# match request body length gt 1000
hostname(config-cmap)# match not request uri regex class URLs
```

Creating a Regular Expression

A regular expression matches text strings either literally as an exact string, or by using *metacharacters* so you can match multiple variants of a text string. You can use a regular expression to match the content of certain application traffic; for example, you can match a URL string inside an HTTP packet.

Use **Ctrl+V** to escape all of the special characters in the CLI, such as question mark (?) or a tab. For example, type **d[Ctrl+V]g** to enter **d?g** in the configuration.

See the **regex** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for performance impact information when matching a regular expression to packets.

Note As an optimization, the FWSM searches on the deobfuscated URL. Deobfuscation compresses multiple forward slashes (/) into a single slash. For strings that commonly use double slashes, like "http://", be sure to search for "http:/" instead.

Table 19-1 lists the metacharacters that have special meanings.

Table 19-1 regex Metacharacters

Character	Description	Notes
.	Dot	Matches any single character. For example, **d.g** matches dog, dag, dtg, and any word that contains those characters, such as doggonnit.
(*exp*)	Subexpression	A subexpression segregates characters from surrounding characters, so that you can use other metacharacters on the subexpression. For example, **d(o|a)g** matches dog and dag, but **do|ag** matches do and ag. A subexpression can also be used with repeat quantifiers to differentiate the characters meant for repetition. For example, **ab(xy){3}z** matches abxyxyxyz.
|	Alternation	Matches either expression it separates. For example, **dog|cat** matches dog or cat.
?	Question mark	A quantifier that indicates that there are 0 or 1 of the previous expression. For example, **lo?se** matches lse or lose. **Note** You must enter **Ctrl+V** and then the question mark or else the help function is invoked.
*	Asterisk	A quantifier that indicates that there are 0, 1 or any number of the previous expression. For example, **lo*se** matches lse, lose, loose, and so on.
+	Plus	A quantifier that indicates that there is at least 1 of the previous expression. For example, **lo+se** matches lose and loose, but not lse.
{*x*} or {*x*,}	Minimum repeat quantifier	Repeat at least *x* times. For example, **ab(xy){2,}z** matches abxyxyz, abxyxyxyz, and so on.
[*abc*]	Character class	Matches any character in the brackets. For example, **[abc]** matches a, b, or c.
[^*abc*]	Negated character class	Matches a single character that is not contained within the brackets. For example, **[^abc]** matches any character other than a, b, or c. **[^A-Z]** matches any single character that is not an uppercase letter.
[*a*-*c*]	Character range class	Matches any character in the range. **[a-z]** matches any lowercase letter. You can mix characters and ranges: **[abcq-z]** matches a, b, c, q, r, s, t, u, v, w, x, y, z, and so does [**a-cq-z**]. The dash (-) character is literal only if it is the last or the first character within the brackets: **[abc-]** or **[-abc]**.
"""	Quotation marks	Preserves trailing or leading spaces in the string. For example, **" test"** preserves the leading space when it looks for a match.
^	Caret	Specifies the beginning of a line.
\	Escape character	When used with a metacharacter, matches a literal character. For example, **\[** matches the left square bracket.

Table 19-1 regex Metacharacters (continued)

Character	Description	Notes
char	Character	When character is not a metacharacter, matches the literal character.
\r	Carriage return	Matches a carriage return 0x0d.
\n	Newline	Matches a new line 0x0a.
\t	Tab	Matches a tab 0x09.
\f	Formfeed	Matches a form feed 0x0c.
\x*NN*	Escaped hexadecimal number	Matches an ASCII character using hexadecimal (exactly two digits).
NNN	Escaped octal number	Matches an ASCII character as octal (exactly three digits). For example, the character 040 represents a space.

To test and create a regular expression, perform the following steps:

Step 1 To test a regular expression to make sure it matches what you think it will match, enter the following command:

```
hostname(config)# test regex input_text regular_expression
```

Where the *input_text* argument is a string you want to match using the regular expression, up to 201 characters in length.

The *regular_expression* argument can be up to 100 characters in length.

Use **Ctrl+V** to escape all of the special characters in the CLI. For example, to enter a tab in the input text in the **test regex** command, you must enter **test regex "test[Ctrl+V Tab]" "test\t"**.

If the regular expression matches the input text, you see the following message:

```
INFO: Regular expression match succeeded.
```

If the regular expression does not match the input text, you see the following message:

```
INFO: Regular expression match failed.
```

Step 2 To add a regular expression after you tested it, enter the following command:

```
hostname(config)# regex name regular_expression
```

Where the *name* argument can be up to 40 characters in length.

The *regular_expression* argument can be up to 100 characters in length.

The following example creates two regular expressions for use in an inspection policy map:

```
hostname(config)# regex url_example example\.com
hostname(config)# regex url_example2 example2\.com
```

Creating a Regular Expression Class Map

A regular expression class map identifies one or more regular expressions. You can use a regular expression class map to match the content of certain traffic; for example, you can match URL strings inside HTTP packets.

The maximum number of class maps (Layer 3/4, inspection, and regular expression) is 255 in single mode or per context in multiple mode. This limit includes default class maps. See the "Default Class Maps" section on page 19-4.

To create a regular expression class map, perform the following steps:

Step 1 Create one or more regular expressions according to the "Creating a Regular Expression" section.

Step 2 Create a class map by entering the following command:

```
hostname(config)# class-map type regex match-any class_map_name
hostname(config-cmap)#
```

Where *class_map_name* is a string up to 40 characters in length. The name "class-default" is reserved. All types of class maps use the same name space, so you cannot reuse a name already used by another type of class map.

The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the regular expressions.

The CLI enters class-map configuration mode.

Step 3 (Optional) Add a description to the class map by entering the following command:

```
hostname(config-cmap)# description string
```

Step 4 Identify the regular expressions you want to include by entering the following command for each regular expression:

```
hostname(config-cmap)# match regex regex_name
```

The following example creates two regular expressions, and adds them to a regular expression class map. Traffic matches the class map if it includes the string "example.com" or "example2.com."

```
hostname(config)# regex url_example example\.com
hostname(config)# regex url_example2 example2\.com
hostname(config)# class-map type regex match-any URLs
hostname(config-cmap)# match regex url_example
hostname(config-cmap)# match regex url_example2
```

Defining Actions (Layer 3/4 Policy Map)

This section describes how to associate actions with Layer 3/4 class maps by creating a Layer 3/4 policy map. This section includes the following topics:

Information About Layer 3/4 Policy Maps

This section describes how Layer 3/4 policy maps work, and includes the following topics:

Policy Map Guidelines

See the following guidelines for using policy maps:

- You can only assign one policy map per interface. (However you can create up to 64 policy maps in the configuration.)
- You can apply the same policy map to multiple interfaces.
- You can identify multiple Layer 3/4 class maps in a Layer 3/4 policy map.
- For each class map, you can assign multiple actions from one or more feature types, if supported. See the "Incompatibility of Certain Feature Actions" section on page 19-17.

Feature Directionality

Actions are applied to traffic bidirectionally or unidirectionally depending on whether the service policy is applied to an interface or globally. For a service policy that is applied to an interface, all features are bidirectional; all traffic that enters or exits the interface to which you apply the policy map is affected if the traffic matches the class map for both directions. When you use a global policy, all features are unidirectional; features that are normally bidirectional when applied to a single interface only apply to the ingress of each interface when applied globally. Because the policy is applied to all interfaces, the policy will be applied in both directions so bidirectionality in this case is redundant.

Feature Matching Guidelines within a Policy Map

See the following guidelines for how a packet matches class maps in a policy map:

- A packet can match only one class map in the policy map for each feature type.
- When the packet matches a class map for a feature type, the FWSM does not attempt to match it to any subsequent class maps for that feature type.
- If the packet matches a subsequent class map for a different feature type, however, then the FWSM also applies the actions for the subsequent class map, if supported. See the "Incompatibility of Certain Feature Actions" section on page 19-17 for more information about unsupported combinations.

For example, if a packet matches a class map for connection limits, and also matches a class map for application inspection, then both class map actions are applied.

If a packet matches a class map for application inspection, but also matches another class map that includes application inspection, then the second class map actions are not applied.

Note Application inspection includes multiple inspection types, and each inspection type is a separate feature when you consider the matching guidelines above.

Order in Which Multiple Feature Actions are Applied

The order in which different types of actions in a policy map are performed is independent of the order in which the actions appear in the policy map. Actions are performed in the following order:

1. TCP and UDP connection settings, and TCP state bypass
2. Application inspection (multiple types)

   The order of application inspections applied when a class of traffic is classified for multiple inspections is as follows. Only one inspection type can be applied to the same traffic. WAAS inspection is an exception, because it can applied along with other inspections for the same traffic. See the "Incompatibility of Certain Feature Actions" section on page 19-17 for more information.

   a. CTIQBE
   b. DNS
   c. FTP
   d. GTP
   e. H323
   f. HTTP
   g. ICMP
   h. ICMP error
   i. ILS
   j. MGCP
   k. NetBIOS
   l. PPTP
   m. Sun RPC
   n. RSH
   o. RTSP
   p. SIP
   q. Skinny
   r. SMTP
   s. SNMP
   t. SQL*Net
   u. TFTP
   v. XDMCP
   w. DCERPC
3. Permitting or Denying Application Types with PISA Integration

Incompatibility of Certain Feature Actions

Some features are not compatible with each other for the same traffic. For example, you cannot configure PISA integration and inspections for the same set of traffic. Also, most inspections should not be combined with another inspection, so the FWSM only applies one inspection if you configure multiple inspections for the same traffic. In this case, the feature that is applied is the higher priority feature in the list in the "Order in Which Multiple Feature Actions are Applied" section on page 19-16.

For information about compatibility of each feature, see the chapter or section for your feature.

Note The **match default-inspection-traffic** command, which is used in the default global policy, is a special CLI shortcut to match the default ports for all inspections. When used in a policy map, this class map ensures that the correct inspection is applied to each packet, based on the destination port of the traffic. For example, when UDP traffic for port 69 reaches the FWSM, then the FWSM applies the TFTP inspection; when TCP traffic for port 21 arrives, then the FWSM applies the FTP inspection. So in this case only, you can configure multiple inspections for the same class map. Normally, the FWSM does not use the port number to determine the inspection applied, thus giving you the flexibility to apply inspections to non-standard ports, for example.

An example of a misconfiguration is if you configure multiple inspections in the same policy map and do not use the default-inspection-traffic shortcut. In Example 19-1, traffic destined to port 21 is mistakenly configured for both FTP and HTTP inspection. In Example 19-2, traffic destined to port 80 is mistakenly configured for both FTP and HTTP inspection. In both cases of misconfiguration examples, only the FTP inspection is applied, because FTP comes before HTTP in the order of inspections applied.

Example 19-1 Misconfiguration for FTP packets: HTTP Inspection Also Configured

```
class-map ftp
   match port tcp 21
class-map http
   match port tcp 21 [it should be 80]
policy-map test
   class ftp
     inspect ftp
   class http
     inspect http
```

Example 19-2 Misconfiguration for HTTP packets: FTP Inspection Also Configured

```
class-map ftp
   match port tcp 80 [it should be 21]
class-map http
   match port tcp 80
policy-map test
   class http
     inspect http
   class ftp
     inspect ftp
```

Feature Matching Guidelines for Multiple Policy Maps

For TCP and UDP traffic (and ICMP when you enable stateful ICMP inspection), service policies operate on traffic flows, and not just individual packets. If traffic is part of an existing connection that matches a feature in a policy on one interface, that traffic flow cannot also match the same feature in a policy on another interface; only the first policy is used.

For example, if HTTP traffic matches a policy on the inside interface to inspect HTTP traffic, and you have a separate policy on the outside interface for HTTP inspection, then that traffic is not also inspected on the egress of the outside interface. Similarly, the return traffic for that connection will not be inspected by the ingress policy of the outside interface, nor by the egress policy of the inside interface.

For traffic that is not treated as a flow, for example ICMP when you do not enable stateful ICMP inspection, returning traffic can match a different policy map on the returning interface. For example, if you configure connection limits on the inside and outside interfaces, but the inside policy sets the maximum connections to 2000 while the outside policy sets the maximum connections to 3000, then a non-stateful Ping might be denied at a lower level if it is outbound than if it is inbound.

Default Layer 3/4 Policy Map

The configuration includes a default Layer 3/4 policy map that the FWSM uses in the default global policy. It is called **global_policy** and performs inspection on the default inspection traffic. You can only apply one global policy, so if you want to alter the global policy, you need to either reconfigure the default policy or disable it and apply a new one.

The default policy map configuration includes the following commands:

```
policy-map global_policy
 class inspection_default
  inspect dns maximum-length 512
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect netbios
  inspect rsh
  inspect skinny
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
```

Note See the "Incompatibility of Certain Feature Actions" section on page 19-17 for more information about the special **match default-inspection-traffic** command used in the default class map.

Adding a Layer 3/4 Policy Map

The maximum number of policy maps is 64. To create a Layer 3/4 policy map, perform the following steps:

Step 1 Add the policy map by entering the following command:

```
hostname(config)# policy-map policy_map_name
```

The *policy_map_name* argument is the name of the policy map up to 40 characters in length. All types of policy maps use the same name space, so you cannot reuse a name already used by another type of policy map. The CLI enters policy-map configuration mode.

Step 2 (Optional) Specify a description for the policy map:

```
hostname(config-pmap)# description text
```

Step 3 Specify a previously configured Layer 3/4 class map using the following command:

```
hostname(config-pmap)# class class_map_name
```

where the *class_map_name* is the name of the class map you created earlier. See the "Identifying Traffic (Layer 3/4 Class Map)" section on page 19-4 to add a class map.

Step 4 Specify one or more actions for this class map.

- TCP and UDP connection limits and timeouts, and TCP sequence number randomization. See the "Configuring Connection Limits and Timeouts" section on page 20-1.
- TCP state bypass. See the "Configuring TCP State Bypass" section on page 20-10.
- Application inspection. See Chapter 21, "Applying Application Layer Protocol Inspection."
- Permitting or Denying Application Types with PISA Integration—See the "Permitting or Denying Application Types with PISA Integration" section on page 20-4.

Note If there is no **match default_inspection_traffic** command in a class map, then at most one **inspect** command is allowed to be configured under the class.

Step 5 Repeat Step 3 and Step 4 for each class map you want to include in this policy map.

The following is an example of a **policy-map** command for connection policy. It limits the number of connections allowed to the web server 10.1.1.1:

```
hostname(config)# access-list http-server permit tcp any host 10.1.1.1
hostname(config)# class-map http-server
hostname(config-cmap)# match access-list http-server

hostname(config)# policy-map global-policy
hostname(config-pmap)# description This policy map defines a policy concerning connection
to http server.
hostname(config-pmap)# class http-server
hostname(config-pmap-c)# set connection conn-max 256
```

The following example shows how multi-match works in a policy map:

```
hostname(config)# class-map inspection_default
hostname(config-cmap)# match default-inspection-traffic
hostname(config)# class-map http_traffic
hostname(config-cmap)# match port tcp eq 80

hostname(config)# policy-map outside_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect http http_map
hostname(config-pmap-c)# inspect sip
hostname(config-pmap)# class http_traffic
hostname(config-pmap-c)# set connection timeout tcp 0:10:0
```

The following example shows how traffic matches the first available class map, and will not match any subsequent class maps that specify actions in the same feature domain:

```
hostname(config)# class-map telnet_traffic
hostname(config-cmap)# match port tcp eq 23
hostname(config)# class-map ftp_traffic
hostname(config-cmap)# match port tcp eq 21
hostname(config)# class-map tcp_traffic
hostname(config-cmap)# match port tcp range 1 65535
hostname(config)# class-map udp_traffic
hostname(config-cmap)# match port udp range 0 65535
hostname(config)# policy-map global_policy
hostname(config-pmap)# class telnet_traffic
hostname(config-pmap-c)# set connection timeout tcp 0:0:0
hostname(config-pmap-c)# set connection conn-max 100
hostname(config-pmap)# class ftp_traffic
hostname(config-pmap-c)# set connection timeout tcp 0:5:0
hostname(config-pmap-c)# set connection conn-max 50
hostname(config-pmap)# class tcp_traffic
hostname(config-pmap-c)# set connection timeout tcp 2:0:0
hostname(config-pmap-c)# set connection conn-max 2000
```

When a Telnet connection is initiated, it matches **class telnet_traffic**. Similarly, if an FTP connection is initiated, it matches **class ftp_traffic**. For any TCP connection other than Telnet and FTP, it will match **class tcp_traffic**. Even though a Telnet or FTP connection can match **class tcp_traffic**, the FWSM does not make this match because they previously matched other classes.

Applying Actions to an Interface (Service Policy)

To activate the Layer 3/4 policy map, create a service policy that applies it to one or more interfaces or that applies it globally to all interfaces. Interface service policies take precedence over the global service policy for a given feature. For example, if you have a global policy with FTP inspection, and an interface policy with TCP connection settings, then both FTP inspection and TCP connection settings are applied to the interface. However, if you have a global policy with FTP inspection, and an interface policy with FTP inspection, then only the interface policy FTP inspection is applied to that interface.

- To create a service policy by associating a policy map with an interface, enter the following command:

  ```
  hostname(config)# service-policy policy_map_name interface interface_name
  ```

- To create a service policy that applies to all interfaces that do not have a specific policy, enter the following command:

  ```
  hostname(config)# service-policy policy_map_name global
  ```

  By default, the configuration includes a global policy that matches all default application inspection traffic and applies inspection to the traffic globally. You can only apply one global policy, so if you want to alter the global policy, you need to either edit the default policy or disable it and apply a new one.

  The default service policy includes the following command:

  ```
  service-policy global_policy global
  ```

For example, the following command enables the inbound_policy policy map on the outside interface:

```
hostname(config)# service-policy inbound_policy interface outside
```

The following commands disable the default global policy, and enables a new one called new_global_policy on all other FWSM interfaces:

```
hostname(config)# no service-policy global_policy global
```

```
hostname(config)# service-policy new_global_policy global
```

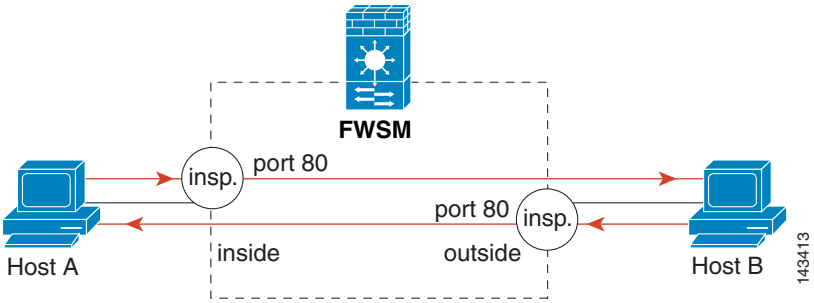
Modular Policy Framework Examples

This section includes several Modular Policy Framework examples, and includes the following topics:

Applying Inspection to HTTP Traffic Globally

In this example (see Figure 19-1), any HTTP connection (TCP traffic on port 80) that enters the FWSM through any interface is classified for HTTP inspection.

Figure 19-1 Global HTTP Inspection



See the following commands for this example:

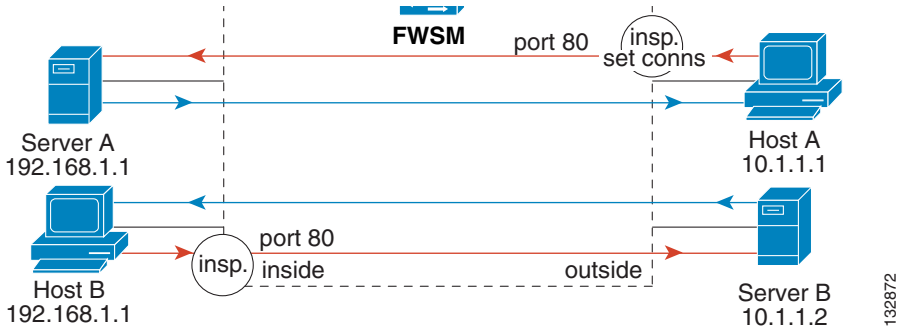
```
hostname(config)# class-map http_traffic
hostname(config-cmap)# match port tcp eq 80

hostname(config)# policy-map http_traffic_policy
hostname(config-pmap)# class http_traffic
hostname(config-pmap-c)# inspect http
hostname(config)# service-policy http_traffic_policy global
```

Applying Inspection and Connection Limits to HTTP Traffic to Specific Servers

In this example (see Figure 19-2), any HTTP connection destined for Server A (TCP traffic on port 80) that enters the FWSM through the outside interface is classified for HTTP inspection and maximum connection limits.

Any HTTP connection destined for Server B that enters the FWSM through the inside interface is classified for HTTP inspection.

Figure 19-2 HTTP Inspection and Connection Limits to Specific Servers



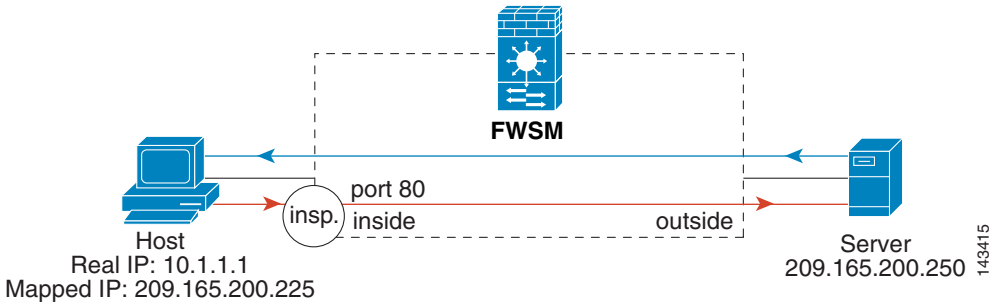
See the following commands for this example:

```
hostname(config)# access-list serverA extended permit tcp any host 192.168.1.1 eq 80
hostname(config)# access-list ServerB extended permit tcp any host 10.1.1.2 eq 80

hostname(config)# class-map http_serverA
hostname(config-cmap)# match access-list serverA
hostname(config)# class-map http_serverB
hostname(config-cmap)# match access-list serverB

hostname(config)# policy-map policy_serverA
hostname(config-pmap)# class http_serverA
hostname(config-pmap-c)# inspect http http_map_serverA
hostname(config-pmap-c)# set connection conn-max 100
hostname(config)# policy-map policy_serverB
hostname(config-pmap)# class http_serverB
hostname(config-pmap-c)# inspect http http_map_serverB

hostname(config)# service-policy policy_serverB interface inside
hostname(config)# service-policy policy_serverA interface outside
```

Applying Inspection to HTTP Traffic with NAT

In this example, the Host on the inside network has two addresses: one is the real IP address 10.1.1.1, and the other is a mapped IP address used on the outside network, 209.165.200.225 (see Figure 19-3). Because the policy is applied to the inside interface, where the real address is used, then you must use the real IP address in the access list in the class map. If you applied it to the outside interface, you would use the mapped addresses.

Figure 19-3 HTTP Inspection with NAT



See the following commands for this example:

```
hostname(config)# static (inside,outside) 209.165.200.225 10.1.1.1
hostname(config)# access-list http_client extended permit tcp host 10.1.1.1 any eq 80

hostname(config)# class-map http_client
hostname(config-cmap)# match access-list http_client

hostname(config)# policy-map http_client
hostname(config-pmap)# class http_client
hostname(config-pmap-c)# inspect http

hostname(config)# service-policy http_client interface inside
```

CHAPTER 20

Configuring Advanced Connection Features

This chapter describes how to customize connection features, and includes the following sections:

Configuring Connection Limits and Timeouts

This section describes how to set maximum TCP and UDP connections, the maximum connection rate, connection timeouts, and how to disable TCP sequence randomization.

Each TCP connection has two ISNs: one generated by the client and one generated by the server. The FWSM randomizes the ISN of the TCP SYN passing in both the inbound and outbound directions.

Randomizing the ISN of the protected host prevents an attacker from predicting the next ISN for a new connection and potentially hijacking the new session.

TCP initial sequence number randomization can be disabled if required. For example:

- If another in-line firewall is also randomizing the initial sequence numbers, there is no need for both firewalls to be performing this action, even though this action does not affect the traffic.
- If you use eBGP multi-hop through the FWSM, and the eBGP peers are using MD5. Randomization breaks the MD5 checksum.
- You use a WAAS device that requires the FWSM not to randomize the sequence numbers of connections.

Note Because of the way TCP sequence randomization is implemented, if you enable Xlate Bypass (see the "Configuring Xlate Bypass" section on page 15-18), then disabling TCP sequence randomization only works for control connections, and not data connections; for data connections, the TCP sequence continues to be randomized.

You can also configure maximum connections and TCP sequence randomization in the NAT

configuration. If you configure these settings for the same traffic using both methods, then the FWSM uses the lower limit. For TCP sequence randomization, if it is disabled using either method, then the FWSM disables TCP sequence randomization.

NAT also lets you configure embryonic connection limits, which triggers TCP Intercept to prevent a DoS attack. To configure connection limits, TCP randomization, and embryonic limits, see Chapter 15, "Configuring NAT."

To set connection limits and timeouts, perform the following steps:

Step 1 To identify the traffic, add a class map using the **class-map** command. See the "Identifying Traffic (Layer 3/4 Class Map)" section on page 19-4 for more information.

For example, you can match all traffic using the following commands:

```
hostname(config)# class-map CONNS
hostname(config-cmap)# match any
```

To match specific traffic, you can match an access list:

```
hostname(config)# access list CONNS extended permit ip any 10.1.1.1 255.255.255.255
hostname(config)# class-map CONNS
hostname(config-cmap)# match access-list CONNS
```

Note In 3.x, when you used the **set connection** command for an access list (**match access-list**), then connection settings were applied to each individual ACE; in 4.0, connection settings are applied to the access list as a whole.

Step 2 To add or edit a policy map that sets the actions to take with the class map traffic, enter the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where the *class_map_name* is the class map from Step 1.

For example:

```
hostname(config)# policy-map CONNS
hostname(config-pmap)# class CONNS
hostname(config-pmap-c)#
```

Step 3 To set maximum connection limits, connection rate limit, or whether TCP sequence randomization is enabled, enter the following command:

```
hostname(config-pmap-c)# set connection {[conn-max n] [conn-rate-limit n]
[random-sequence-number {enable | disable}]}
```

where the **conn-max** *n* argument sets the maximum number of simultaneous TCP and/or UDP connections that are allowed, between 0 and 65535. The default is 0, which means no limit on connections.

The **conn-rate-limit** *n* argument sets the maximum TCP and/or UDP connections per second between 0 and 65535. The default is 0, which means no limit on the connection rate.

The **random-sequence-number** {**enable** | **disable**} keyword enables or disables TCP sequence number randomization.

You can enter this command all on one line (in any order), or you can enter each attribute as a separate command. The FWSM combines the command into one line in the running configuration.

Step 4 To set the timeout for TCP embryonic connections (half-opened) or TCP half-closed connections, enter the following command:

```
hostname(config-pmap-c)# set connection timeout {[embryonic hh:mm:ss] [half-closed
hh:mm:ss]}
```

where the **embryonic** *hh***:***mm:ss* keyword sets the timeout period until a TCP embryonic (half-open) connection is closed, between 0:0:1 and 0:4:15. The default is 0:0:20. You can also set this value to 0, which means the connection never times out.

The **half-closed** *hh***:***mm:ss* keyword sets the idle timeout between 0:0:1 and 0:4:15. The default is 0:0:20. You can also set this value to 0, which means the connection never times out. The FWSM does not send a reset when taking down half-closed connections.

You can enter this command all on one line (in any order), or you can enter each attribute as a separate command. The command is combined onto one line in the running configuration.

Note This command does not affect secondary connections created by an inspection engine. For example, you cannot change the connection settings for secondary flows like SQL*Net, FTP data flows, and so on using the **set connection timeout** command. For these connections, use the global **timeout conn** command to change the idle time. Note that the **timeout conn** command affects *all* traffic flows unless you otherwise use the **set connection timeout** command for eligible traffic.

Step 5 To set the timeout for idle connections for all protocols, enter the following command:

```
hostname(config-pmap-c)# set connection timeout idle hh:mm:0
```

where the **idle** *hh:mm***:0** argument defines the idle time after which an established connection of any protocol closes, between 0:5:0 and 1092:15:0. The default is 0:60:0. You can also set the value to 0, which means the connection never times out.

Note This command ignores the value you set for seconds; you can only specify the hours and minutes. Therefore, you should set the seconds to be 0.

The **idle** keyword has replaced the **tcp** keyword in the **set connection timeout** command, but if your configuration includes the **tcp** command (for TCP connections only), it is still accepted. If your policy includes both the **idle** and **tcp** commands, then the **tcp** command takes precedence for TCP traffic only if the class map matches an access list that specifies TCP traffic explicitly. See the **set connection timeout** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

This command does not affect secondary connections created by an inspection engine. For example, you cannot change the connection settings for secondary flows like SQL*Net, FTP data flows, and so on using the **set connection timeout** command. For these connections, use the global **timeout conn** command to change the idle time. Note that the **timeout conn** command affects *all* traffic flows unless you otherwise use the **set connection timeout** command for eligible traffic.

Step 6 To activate the policy map on one or more interfaces, enter the following command:

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

where *policy_map_name* is the policy map you configured in Step 2. To apply the policy map to traffic on all the interfaces, use the **global** keyword. To apply the policy map to traffic on a specific interface, use the **interface** *interface_name* option, where *interface_name* is the name assigned to the interface with the **nameif** command.

Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

The following example sets the maximum TCP and UDP connections to 5000, the maximum connections per second to 500, and sets the maximum embryonic timeout to 40 seconds, the half-closed timeout to 20 minutes, and the idle timeout to 2 hours for traffic going to 10.1.1.1:

```
hostname(config)# access-list CONNS permit ip any host 10.1.1.1

hostname(config)# class-map conns
hostname(config-cmap)# match access-list CONNS

hostname(config-cmap)# policy-map conns
hostname(config-pmap)# class conns
hostname(config-pmap-c)# set connection conn-max 5000 conn-rate-limit 500
hostname(config-pmap-c)# set connection timeout embryonic 0:0:40 half-closed 0:20:0
hostname(config-pmap-c)# set connection timeout idle 2:0:0

hostname(config-pmap-c)# service-policy conns interface outside
```

You can enter **set connection** commands with multiple parameters or you can enter each parameter as a separate command. The FWSM combines the commands into one line in the running configuration. For example, if you entered the following two commands in class configuration mode:

```
hostname(config-pmap-c)# set connection timeout embryonic 0:0:40
hostname(config-pmap-c)# set connection timeout half-closed 0:20:0
```

the output of the **show running-config policy-map** command would display the result of the two commands in a single, combined command:

```
set connection timeout embryonic 0:0:40 half-closed 0:20:0
```

Permitting or Denying Application Types with PISA Integration

Note This feature depends on Cisco IOS Release 12.2(18)ZYA or later, and is only available on the Catalyst 6500 switch.

The Programmable Intelligent Services Accelerator (PISA) on the switch supervisor can quickly determine the application type of a given flow by performing deep packet inspection. This determination can be made even if the traffic is not using standard ports. The FWSM can leverage the high-performance deep packet inspection of the PISA card so that it can permit or deny traffic based on the application type. Unlike the FWSM inspection feature, which passes through the control plane path, traffic that the PISA tags can pass through the FWSM accelerated path. Another benefit of FWSM and PISA integration is to consolidate your security configuration on a single FWSM instead of having to configure multiple upstream switches with PISAs installed.

You might want to deny certain types of application traffic when you want to preserve bandwidth for critical application types. For example, you might deny the use of peer-to-peer (P2P) applications if they are affecting your other critical applications.

This section includes the following topics:

PISA Integration Overview

This section describes how the PISA works with the FWSM, and includes the following topics:

PISA Integration Guidelines and Limitations

The following guidelines and limitations apply to PISA integration:

- The PISA and the FWSM cannot be in the same switch chassis. You can, however, use multiple PISAs upstream and downstream of the FWSM if desired.
- There is a slight performance impact on the PISA for traffic sent to the FWSM, due to the need to tag the packets for the FWSM (see the "Using GRE for Tagging" section.)
- When a UDP packet is denied due to the FWSM service policy, the corresponding session is not immediately deleted. Instead, it is allowed to time out, and the packets that hit this session in the meantime are dropped.
- It is possible for an end-user application to use the special GRE key that is used between the FWSM and the PISA. In such instances, the PISA generates a syslog message and drops these packets.
- The PISA takes several packets to determine the application type; therefore a session starts to be established on the FWSM before the PISA tagging commences. When the PISA tagging commences, the FWSM security policy is then applied, and if the policy is to deny the flow, the session is prevented from completing.
- For fragmented packets, the PISA tags the first fragment, and the FWSM reassembles the packet and acts upon it based on the encapsulation included in the first fragment.

See also the "PISA Limitations and Restrictions" section on page 20-7.

Using GRE for Tagging

After the PISA identifies the application used by a given traffic flow, it encapsulates all packets using GRE and includes a tag informing the FWSM of the application type. In addition, an outer IP header almost identical (except for the Layer 4 protocol, which now indicates GRE) to the inner/original IP header is added. The original Layer 2 header is maintained. This preserves the original routing/switching paths for the modified packet. The GRE encapsulation adds 32 bytes (20 bytes for the outer IP header and 12 bytes for the GRE header).

After the FWSM receives the packet and acts on the information, it strips the GRE encapsulation from the packet.

When you configure the FWSM to deny traffic based on the PISA encapsulation, for the VLAN on which that traffic resides, the PISA encapsulates all traffic (including traffic that you did not specify for denial).

The GRE encapsulation increases the packet size slightly, so you should increase the MTU between the PISA and the FWSM according to the "Changing the MTU on the Switch to Support Longer Packet Length" section on page 20-8.

The GRE encapsulation causes a slight performance impact for PISA traffic sent to the FWSM.

Failover Support

Failover of the PISA is independent of failover of the FWSM. If you have Stateful Failover on the FWSM, then the session information is maintained across the failover.

Configuring the FWSM to Deny PISA Traffic

To identify traffic that you want to deny using PISA tagging, perform the following steps:

Step 1 To identify the traffic that you want to deny based on the application type, add a class map using the **class-map** command. See the "Identifying Traffic (Layer 3/4 Class Map)" section on page 19-4 for more information.

For example, you can match an access list:

```
hostname(config)# access list BAD_APPS extended permit any 10.1.1.1 255.255.255.255
hostname(config)# class-map denied_apps
hostname(config-cmap)# match access-list BAD_APPS
```

Step 2 To add or edit a policy map that sets the actions to take with the class map traffic, enter the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where the *class_map_name* is the class map from Step 1.

For example:

```
hostname(config)# policy-map denied_apps_policy
hostname(config-pmap)# class denied_apps
hostname(config-pmap-c)#
```

Step 3 Determine which applications are permitted or denied by entering the following commands:

```
hostname(config-pmap-c)# deny {all | protocol}
hostname(config-pmap-c)# permit protocol
```

Where the *protocol* argument is the protocol name or number. To see the supported protocol names, use the **permit ?** or **deny ?** command.

You can combine **permit** and **deny** statements to narrow the traffic that you want denied. You must enter at least one **deny** statement. Unlike access lists, which have an implicit deny at the end, PISA actions have an implicit permit at the end.

For example, to permit all traffic except for Skype, eDonkey, and Yahoo, enter the following commands:

```
hostname(config-pmap-c)# deny skype
hostname(config-pmap-c)# deny yahoo
hostname(config-pmap-c)# deny eDonkey
```

The following example denies all traffic except for Kazaa and eDonkey:

```
hostname(config-pmap-c)# deny all
hostname(config-pmap-c)# permit kazaa
hostname(config-pmap-c)# permit eDonkey
```

Note For a class map with the **permit** and **deny** commands, you cannot also include any **inspect** commands.

Step 4 Activate the policy map on one or more interfaces by entering the following command:

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

Where **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

The following is an example configuration for PISA integration:

```
hostname(config)# access-list BAD_APPS extended permit 10.1.1.0 255.255.255.0 10.2.1.0
255.255.255.0

hostname(config)# class-map denied_apps
hostname(config-cmap)# description "Apps to be blocked"
hostname(config-cmap)# match access-list BAD_APPS

hostname(config-cmap)# policy-map denied_apps_policy
hostname(config-pmap)# class denied_apps
hostname(config-pmap-c)# deny skype
hostname(config-pmap-c)# deny yahoo
hostname(config-pmap-c)# deny eDonkey

hostname(config-pmap-c)# service-policy denied_apps_policy inside
```

Configuring the Switch for PISA/FWSM Integration

This section describes how to configure the switch for PISA/FWSM integration and includes the following topics:

PISA Limitations and Restrictions

The following limitations and restrictions apply to the PISA:

- Network Based Application Recognition (NBAR) does not work on Layer 3 EtherChannels. Layer 2 EtherChannels are supported.
- The RP on the PISA does not support protocol tagging. So any packets going to the FWSM from the RP will not be tagged.
- NBAR implementation does not support IPv6. So protocol discovery and tagging are only applicable to IPv4. In addition to this restriction imposed by NBAR, the underlying PISA infrastructure also does not support acceleration of IPv6 packets.
- Currently there is a caveat in the L2 PISA implementation for VLANs that have been PISA-accelerated on an Layer 2 port (for example, a trunk); the SVI interfaces for VLANs passing through the accelerated Layer 2 port cannot be in an up state (they will become admin down).
- Multi-VLAN access ports are not supported.

See also the "PISA Integration Guidelines and Limitations" section on page 20-5.

Changing the MTU on the Switch to Support Longer Packet Length

Because of the GRE encapsulation, you should increase the MTU size on VLANs used between the PISA and the FWSM. The GRE encapsulation adds 32 bytes (20 bytes for the outer IP header and 12 bytes for the GRE header).

- To change the MTU on a routed switch port or a Layer 3 interface (SVI), enter the following command:

```
Router(config-if)# mtu mtu_size
```

  For an SVI, the *mtu_size* is between 64 and 9216 bytes. For a routed switch port, the *mtu_size* is between 1500 and 9216 bytes. The default MTU size is 1500 bytes.

- To configure the global LAN port MTU size for Layer 2 ports, enter the following command:

```
Router(config)# system jumbomtu mtu_size
```

  The *mtu_size* can be between 1500 and 921 bytes. The default size is 9216 bytes.

Configuring Classification on the PISA

- To enable classification on a Layer 2 switch port (access, trunk or EtherChannel configured on a physical port) or a Layer 3 interface (SVI, routed port, or subinterface), enter the following command in interface configuration mode.

```
Router(config-if)# ip nbar protocol-discovery
```

- To show protocol discovery statistics on a Layer 2 or Layer 3 interface, enter the following command:

```
Router# show ip nbar protocol-discovery interface ifname
```

Configuring Tagging on the PISA

After protocol discovery is enabled, enable egress packet tagging by entering the following commands.

Note Classification and tagging need to be enabled on the same port; for example, you cannot enable classification on access ports and tagging on a trunk port.

- To enable tagging on a switch port (access port) or a Layer 3 interface (SVI, routed port, or subinterface), enter the following command in interface configuration mode:

  ```
  Router(config-if)# ip nbar protocol-tagging
  ```

- To enable tagging on a trunk port, enter the following command in interface configuration mode:

  ```
  Router(config-if)# ip nbar protocol-tagging [vlan-list vlan-list]
  ```

  Where the **vlan-list** *vlan-list* argument specifies a list of VLANs to be tagged. If not specified, all active VLANs are tagged.

The following commands help you monitor the tagging by the PISA:

- The following command displays tagging configuration information:

  ```
  Router# show ip nbar protocol-tagging {key | interface ifname | summary}
  ```

  Where the **key** keyword shows the GRE key used for the tagging.

  The **interface** *ifname* argument shows if tagging is enabled on an interface.

  The **summary** keyword shows all interfaces with tagging enabled.

- The following command shows the mapping of protocol name to ID:

  ```
  Router# show ip nbar protocol-id [protocol_name]
  ```

  If you enter the *protocol_name*, the mapped ID is shown. When omitted, the complete list of protocol names and IDs is shown.

- To show the number of packets tagged on the PISA, enter the following command:

  ```
  Router# show platform pisa np tx counters
  ```

  For example:

  ```
  Router# show platform pisa np tx counters

  TX Statistics(ME1)
  ------------------
  Errors:  0
  ……
  TX NBAR Protocol tagged pkt:  9869
  ```

Sample Switch Configurations for PISA Integration

Example 20-1 Layer 3 Mode (Interface-based, Routed port/SVI)

```
Router(config)# interface vlan 100
Router(config-if)# ip nbar protocol-discovery
! enables discovery
Router(config-if)# ip nbar protocol-tagging
! enables tagging
Router(config-if)# mtu 9216
! Allows packet sizes up to 9216 bytes without fragmenting
```

Example 20-2 Layer 2 Mode (Interface-based, Protocol Discovery on Uplink Ports)

```
Router(config)# interface gigabitethernet 6/1
Router(config-if)# ip nbar protocol-discovery
! Classification
```

```
Router(config-if)# ip nbar protocol-tagging vlan-list 100
! Tagging
Router(config-if)# mtu 9216
! Allow packet size up to 9216 bytes without fragmenting
Router(config)# system jumbomtu 9216
! Set global LAN port MTU to 9216 bytes
```

Monitoring PISA Connections

This section includes the following topics:

Syslog Message for Dropped Connections

Syslog message 302014 (for TCP) and 302016 (for UDP) display when a PISA connection is denied. For example:

```
%FWSM-6-302014: Teardown TCP connection 144547133155839947 for inside:10.1.1.12/33407 to
outside:209.165.201.10/21 duration 0:00:00 bytes 160 PISA denied protocol
```

Viewing PISA Connections on the FWSM

To monitor connections from the PISA, use the **show conn** command. Connections that are tagged by the PISA are listed in the output with the “p” flag. The following is sample output from the **show conn** command:

```
hostname# show conn
2 in use, 3 most used
 Network Processor 1 connections
TCP out 10.1.1.10:21 in 209.165.201.12:33406 idle 0:00:04 Bytes 1668 FLAGS - UOIp
 Network Processor 2 connections
UDP out 10.1.1.255:137 in 10.1.1.11:137 idle 0:00:48 Bytes 288 FLAGS -
Multicast sessions:
 Network Processor 1 connections
 Network Processor 2 connections
IPv6 connections:
...
```

Configuring TCP State Bypass

This section describes how to configure TCP state bypass, and includes the following topics:

TCP State Bypass Overview

This section describes how to use TCP state bypass, and includes the following topics:

Allowing Outbound and Inbound Flows through Separate FWSMs

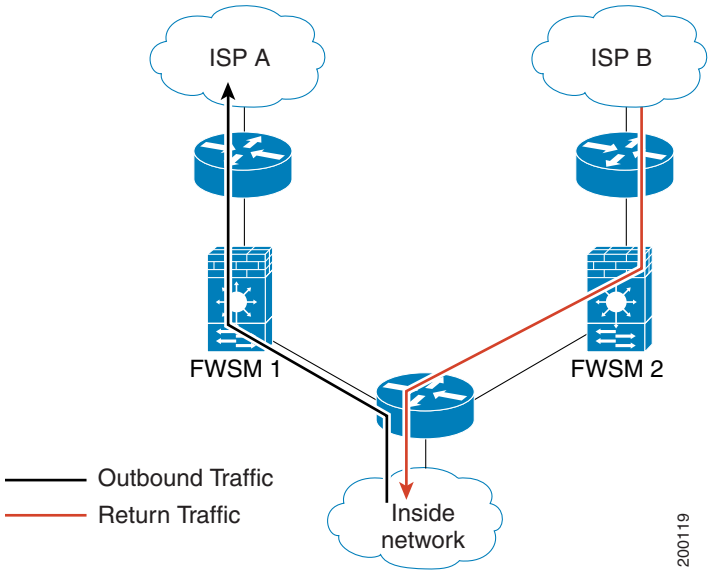
By default, all traffic that goes through the FWSM is inspected using the Adaptive Security Algorithm and is either allowed through or dropped based on the security policy. The FWSM maximizes the firewall performance by checking the state of each packet (is this a new connection or an established connection?) and assigning it to either the session management path (a new connection SYN packet), the accelerated path (an established connection), or the control plane path (advanced inspection). See the "Stateful Inspection Overview" section on page 1-9 for more detailed information about the stateful firewall.

TCP packets that match existing connections in the accelerated path can pass through the FWSM without rechecking every aspect of the security policy. This feature maximizes performance. However, the method of establishing the session in the accelerated path using the SYN packet, and the checks that occur in the accelerated path (such as TCP sequence number), can stand in the way of asymmetrical routing solutions: both the outbound and inbound flow of a connection must pass through the same FWSM.

For example, a new connection goes to FWSM 1. The SYN packet goes through the session management path, and an entry for the connection is added to the accelerated path table. If subsequent packets of this connection go through FWSM 1, then the packets will match the entry in the accelerated path, and are passed through. But if subsequent packets go to FWSM 2, where there was not a SYN packet that went

through the session management path, then there is no entry in the accelerated path for the connection, and the packets are dropped. Figure 20-1 shows an asymmetric routing example where the outbound traffic goes through a different FWSM than the inbound traffic:

Figure 20-1 Asymmetric Routing



If you have asymmetric routing configured on upstream routers, and traffic alternates between two FWSMs, then you can configure TCP state bypass for specific traffic. TCP state bypass alters the way sessions are established in the accelerated path and disables the accelerated path checks. This feature treats TCP traffic much as it treats a UDP connection: when a non-SYN packet matching the specified networks enters the FWSM, and there is not an accelerated path entry, then the packet goes through the session management path to establish the connection in the accelerated path. Once in the accelerated path, the traffic bypasses the accelerated path checks.

Unsupported Features

The following features are not supported when you use TCP state bypass:

- Application inspection—Application inspection requires both inbound and outbound traffic to go through the same FWSM, so application inspection is not supported with TCP state bypass.
- AAA authenticated sessions—When a user authenticates with one FWSM, traffic returning via the other FWSM will be denied because the user did not authenticate with that FWSM.

Compatibility with NAT

Because the translation session is established separately for each FWSM, be sure to configure static NAT on both FWSMs for TCP state bypass traffic; if you use dynamic NAT, the address chosen for the session on FWSM 1 will differ from the address chosen for the session on FWSM 2.

Connection Timeout

If there is no traffic on a given connection for 2 minutes, the connection times out. You can override this default using the **set connection timeout tcp** command. Normal TCP connections timeout by default after 60 minutes.

Enabling TCP State Bypass

To enable TCP state bypass, perform the following steps:

Step 1 To identify the traffic for which you want to disable stateful firewall inspection, add a class map using the **class-map** command. See the "Identifying Traffic (Layer 3/4 Class Map)" section on page 19-4 for more information.

For example, you can match an access list:

```
hostname(config)# access list bypass extended permit tcp any 10.1.1.1 255.255.255.255
hostname(config)# class-map bypass_traffic
hostname(config-cmap)# match access-list bypass
```

Step 2 To add or edit a policy map that sets the actions to take with the class map traffic, enter the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where the *class_map_name* is the class map from Step 1.

For example:

```
hostname(config)# policy-map tcp_bypass_policy
hostname(config-pmap)# class bypass_traffic
hostname(config-pmap-c)#
```

Step 3 Enable TCP state bypass by entering the following command:

```
hostname(config-pmap-c)# set connection advanced-options tcp-state-bypass
```

Step 4 Activate the policy map on one or more interfaces by entering the following command:

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

Where **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

Note If you use the **show conn** command, the display for connections that use TCP state bypass includes the flag "b."

The following is an example configuration for TCP state bypass:

```
hostname(config)# access-list tcp_bypass extended permit tcp 10.1.1.0 255.255.255.0
10.2.1.0 255.255.255.0

hostname(config)# class-map tcp_bypass
```

```
hostname(config-cmap)# description "TCP traffic that bypasses stateful firewall"
hostname(config-cmap)# match access-list tcp_bypass

hostname(config-cmap)# policy-map tcp_bypass_policy
hostname(config-pmap)# class tcp_bypass
hostname(config-pmap-c)# set connection advanced-options tcp-state-bypass

hostname(config-pmap-c)# service-policy tcp_bypass_policy outside
```

Disabling TCP Normalization

For traffic that passes through the control-plane path, such as packets that require Layer 7 inspection or management traffic, the FWSM sets the maximum number of out-of-order packets that can be queued for a TCP connection to 2 packets, which is not user-configurable. Other TCP normalization features that are supported on the PIX and ASA platforms are not enabled for FWSM. You can disable the limited TCP normalization support for the FWSM using the **no control-point tcp-normalizer** command.

Preventing IP Spoofing

This section lets you enable Unicast Reverse Path Forwarding on an interface. Unicast RPF guards against IP spoofing (a packet uses an incorrect source IP address to obscure its true source) by ensuring that all packets have a source IP address that matches the correct source interface according to the routing table.

Normally, the FWSM only looks at the destination address when determining where to forward the packet. Unicast RPF instructs the FWSM to also look at the source address; this is why it is called Reverse Path Forwarding. For any traffic that you want to allow through the FWSM, the FWSM routing table must include a route back to the source address. See RFC 2267 for more information.

For outside traffic, for example, the FWSM can use the default route to satisfy the Unicast RPF protection. If traffic enters from an outside interface, and the source address is not known to the routing table, the FWSM uses the default route to correctly identify the outside interface as the source interface.

If traffic enters the outside interface from an address that is known to the routing table, but is associated with the inside interface, then the FWSM drops the packet. Similarly, if traffic enters the inside interface from an unknown source address, the FWSM drops the packet because the matching route (the default route) indicates the outside interface.

Unicast RPF is implemented as follows:

- ICMP packets have no session, so each packet is checked.
- UDP and TCP have sessions, so the initial packet requires a reverse route lookup. Subsequent packets arriving during the session are checked using an existing state maintained as part of the session. Non-initial packets are checked to ensure they arrived on the same interface used by the initial packet.

To enable Unicast RPF, enter the following command:

```
hostname(config)# ip verify reverse-path interface interface_name
```

Configuring the Fragment Size

By default, the FWSM allows up to 24 fragments per IP packet, and up to 200 fragments awaiting reassembly. You might need to let fragments on your network if you have an application that routinely fragments packets, such as NFS over UDP. However, if you do not have an application that fragments traffic, we recommend that you do not allow fragments through the FWSM. Fragmented packets are often used as DoS attacks. To set disallow fragments, enter the following command:

```
hostname(config)# fragment chain 1 [interface_name]
```

Enter an interface name if you want to prevent fragmentation on a specific interface. By default, this command applies to all interfaces.

Blocking Unwanted Connections

If you know that a host is attempting to attack your network (for example, system log messages show an attack), then you can block (or shun) connections based on the source IP address and other identifying parameters. No new connections can be made until you remove the shun.

Note If you have an IPS that monitors traffic, then the IPS can shun connections automatically.

To shun a connection manually, perform the following steps:

Step 1 If necessary, view information about the connection by entering the following command:

```
hostname# show conn
```

The FWSM shows information about each connection, such as the following:

```
TCP out 64.101.68.161:4300 in 10.86.194.60:23 idle 0:00:00 bytes 1297 flags UIO
```

Step 2 To shun connections from the source IP address, enter the following command:

```
hostname(config)# shun src_ip [dst_ip src_port dest_port [protocol]] [vlan vlan_id]
```

This command drops an existing connection, as well as blocking future connections. By default, the protocol is 0 for IP.

For multiple context mode, you can enter this command in the admin context, and by specifying a VLAN ID that is assigned to an interface in other contexts, you can shun the connection in other contexts.

Step 3 To remove the shun, enter the following command:

```
hostname(config)# no shun src_ip [vlan vlan_id]
```

CHAPTER 21

Applying Application Layer Protocol Inspection

This chapter describes how to configure application layer protocol inspection. Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the FWSM to perform a deep packet inspection instead of passing the packet through the accelerated path (see the "Stateful Inspection Overview" section on page 1-9 for more information about the accelerated path). As a result, inspection engines can affect overall throughput.

Several common inspection engines are enabled on the FWSM by default, but you might need to enable others depending on your network. This chapter includes the following sections:

Inspection Engine Overview

This section includes the following topics:

When to Use Application Protocol Inspection

When a user establishes a connection, the FWSM checks the packet against access lists, creates an address translation, and creates an entry for the session in the accelerated path, so that further packets can bypass time-consuming checks. However, the accelerated path relies on predictable port numbers and does not perform address translations inside a packet.

Many protocols open secondary TCP or UDP ports. The initial session on a well-known port is used to negotiate dynamically-assigned port numbers.

Other applications embed an IP address in the packet that needs to match the source address that is normally translated when it goes through the FWSM.

If you use applications like these, then you need to enable application inspection.

When you enable application inspection for a service that embeds IP addresses, the FWSM translates embedded addresses and updates any checksum or other fields that are affected by the translation.

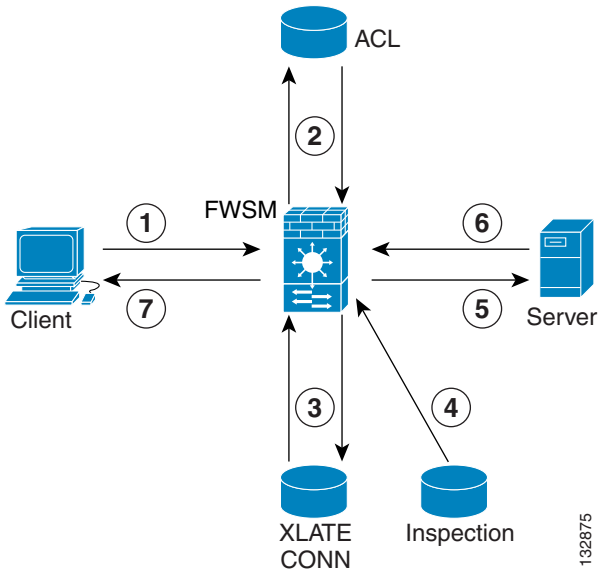
When you enable application inspection for a service that uses dynamically assigned ports, the FWSM monitors sessions to identify the dynamic port assignments, and permits data exchange on these ports for the duration of the specific session.

How Inspection Engines Work

As illustrated in Figure 21-2, the FWSM uses three databases for its basic operation:

- Access lists—Used for authentication and authorization of connections based on specific networks, hosts, and services (TCP/UDP port numbers).
- Inspections—Contains a static, predefined set of application-level inspection functions.
- Connections (XLATE and CONN tables)—Maintains state and other information about each established connection. This information is used by the Adaptive Security Algorithm and cut-through proxy to efficiently forward traffic within established sessions.

Figure 21-1 How Inspection Engines Work



In Figure 21-2, operations are numbered in the order they occur, and are described as follows:

1. A TCP SYN packet arrives at the FWSM to establish a new connection.
2. The FWSM checks the access list database to determine if the connection is permitted.
3. The FWSM creates a new entry in the connection database (XLATE and CONN tables).
4. The FWSM checks the Inspections database to determine if the connection requires application-level inspection.
5. After the application inspection engine completes any required operations for the packet, the FWSM forwards the packet to the destination system.
6. The destination system responds to the initial request.
7. The FWSM receives the reply packet, looks up the connection in the connection database, and forwards the packet because it belongs to an established session.

The default configuration of the FWSM includes a set of application inspection entries that associate supported protocols with specific TCP or UDP port numbers and that identify any special handling required.

Inspection Limitations

See the following limitations for application protocol inspection:

- State information for multimedia sessions that require inspection are not passed over the state link for stateful failover. The exception is GTP, which is replicated over the state link.
- Some inspection engines do not support PAT, NAT, outside NAT, or NAT between same security interfaces. See "Default Inspection Policy" for more information about NAT support.

- If you configure PAT for traffic that is being inspected, the FWSM performs application inspection on the translated port numbers rather than the real port numbers.

  Service policies applying inspection to traffic with translated port numbers should use class maps that identify traffic using the translated port numbers. For example, if you implement PAT to translate ports 2727 and 2427 to port 1400, you should configure MGCP inspection to match traffic sent to port 1400 rather than the well known ports 2427 and 2727.

- When application inspection is enabled on the FWSM for TCP flows (especially for application inspection of protocols like VoIP), the TCP sender segments the TCP packets based on the maximum segment size (MSS) advertised by the TCP receiver. The FWSM reassembles the TCP segments, performs the inspection, and transmits the packets to the TCP receiver based on its interface maximum transmission unit (MTU) and not the MSS advertised by the TCP receiver.

  For example, two SIP endpoints (Polycomm video conferencing units) advertise an MSS of 536 bytes. The FWSM proxies this connection and one video unit sends a H.245 setup message that is 761 bytes segmented into three packets. The FWSM reassembles these three segments and transmits them to the endpoint as one single 761 data byte packet instead of honoring the 536 byte MSS and resegmenting the message as appropriate.

  To account for this limitation, you must perform the following actions on the FWSM:

  - Increase the MSS on the TCP receiver.
  - Lower the MTU on the FWSM interface.
  - Only if possible, disable the advanced protocol inspection.

- When application inspection is enabled for a protocol and another application utilizes the same port as that inspected application protocol, the FWSM can exhibit unpredictable behavior (including packet loss) when inspecting that application protocol. When this situation occurs, you should disable the inspection engine for that application protocol.

Default Inspection Policy

By default, the configuration includes a policy that matches all default application inspection traffic and applies inspection to the traffic on all interfaces (a global policy). Default application inspection traffic includes traffic to the default ports for each protocol. You can only apply one global policy, so if you want to alter the global policy, for example, to apply inspection to non-standard ports, or to add inspections that are not enabled by default, you need to either edit the default policy or disable it and apply a new one.

Table 21-1 lists all inspections supported, the default ports used in the default class map, and the inspection engines that are on by default, shown in bold. This table also notes any NAT limitations.

Table 21-1 Supported Application Inspection Engines

Application[1]	Default Port	NAT Limitations	Standards[2]	Comments
CTIQBE	TCP/2748	—	—	—
DCERPC	TCP/135	—	—	Supports the map and lookup operations of the EPM for clients.
DNS over UDP	UDP/53	Only forward NAT. No NAT support is available for name resolution through WINS.	RFC 1123	No PTR records are changed. Default maximum packet length is 512 bytes.

Table 21-1 Supported Application Inspection Engines (continued)

Application[1]	Default Port	NAT Limitations	Standards[2]	Comments
ESMTP	TCP/25	—	RFC 821, 1123	—
FTP	TCP/21	—	RFC 959	Default FTP inspection does not enforce compliance with RFC standards. To do so, configure the **inspect ftp** command with the **strict** keyword.
GTP	UDP/3386 (V0) UDP/2123 (V1)	No NAT or PAT.	—	Requires a special license.
H.323	TCP/1720 UDP/1718 UDP (RAS) 1718-1719	No NAT on same security interfaces. No static PAT.	ITU-T H.323, H.245, H225.0, Q.931, Q.932	By default, both RAS and H.225 inspection are enabled.
HTTP	TCP/80	—	RFC 2616	Beware of MTU limitations stripping ActiveX and Java. If the MTU is too small to allow the Java or ActiveX tag to be included in one packet, stripping may not occur.
ICMP	—	—	—	All ICMP traffic is matched in the default class map.
ICMP ERROR	—	—	—	All ICMP traffic is matched in the default class map.
ILS (LDAP)	TCP/389	No PAT.	—	—
MGCP	UDP/2427, 2727	—	RFC 2705bis-05	—
NetBIOS Datagram Service / UDP	UDP/138		—	—
NetBIOS Name Service / UDP	UDP/137	No NAT No PAT	—	No WINS support.
PPTP	TCP/1723	—	RFC 2637	—
RSH	TCP/514	No PAT	Berkeley UNIX	—
RTSP	TCP/554	No PAT. No outside NAT.	RFC 2326, 2327, 1889	No handling for HTTP cloaking.
SIP	TCP/5060 UDP/5060	No outside NAT. No NAT on same security interfaces.	RFC 3261	—
SKINNY (SCCP)	TCP/2000	No outside NAT. No NAT on same security interfaces.	—	Does not handle TFTP uploaded Cisco IP Phone configurations under certain circumstances.
SMTP	TCP/25	—	RFC 821, 1123	—

Table 21-1 Supported Application Inspection Engines (continued)

Application[1]	Default Port	NAT Limitations	Standards[2]	Comments
SNMP	UDP/161, 162	No NAT or PAT.	RFC 1155, 1157, 1212, 1213, 1215	v.2 RFC 1902-1908; v.3 RFC 2570-2580.
SQL*Net	TCP/1521	—	—	v.1 and v.2.
SunRPC	UDP/111 TCP/111	No PAT. Payload not NATed.	—	The default class map includes UDP port 111; if you want to enable Sun RPC inspection for TCP port 111, you need to create a new class map that matches TCP port 111, add the class to the policy, and then apply the **inspect sunrpc** command to that class.
TFTP	TCP/69 UDP/69	Payload not NATed.	RFC 1530	—
WAAS	TCP	—	—	Enables the TCP option 33 parsing.
XDCMP	UDP/177	No NAT or PAT.	—	—

1. Inspection engines that are enabled by default for the default port are in bold.
2. The FWSM is in compliance with these standards, but it does not enforce compliance on packets being inspected. For example, FTP commands are supposed to be in a particular order, but the FWSM does not enforce the order.

The default policy configuration includes the following commands:

```
class-map inspection_default
 match default-inspection-traffic
policy-map global_policy
 class inspection_default
  inspect dns maximum-length 512
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect netbios
  inspect rsh
  inspect skinny
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
service-policy global_policy global
```

Configuring Application Inspection

This feature uses Modular Policy Framework, so that implementing application inspection consists of the following:

1. Identifying traffic.
2. Applying inspections to the traffic.

   For some applications, you can perform special actions when you enable inspection.
3. Activating inspections on an interface.

See Chapter 19, "Using Modular Policy Framework," for more information about Modular Policy Framework.

Inspection is enabled by default for some applications. See the "Default Inspection Policy" section on page 21-4 section for more information. Use this section to modify your inspection policy.

To configure application inspection, perform the following steps:

Step 1 To identify the traffic to which you want to apply inspections, add a Layer 3/4 class map. See the "Identifying Traffic (Layer 3/4 Class Map)" section on page 19-4 for detailed information.

The default Layer 3/4 class map for through traffic is called "inspection_default." It matches traffic using a special **match** command, **match default-inspection-traffic**, to match the default ports for each application protocol.

You can specify a **match access-list** command along with the **match default-inspection-traffic** command to narrow the matched traffic to specific IP addresses. Because the **match default-inspection-traffic** command specifies the ports to match, any ports in the access list are ignored.

If you want to match non-standard ports, then you need to create a new class map for the non-standard ports. See the "Default Inspection Policy" section on page 21-4 for the standard ports for each inspection engine. You can combine multiple class maps in the same policy if desired, so you can create one class map to match certain traffic, and another to match different traffic. However, if traffic matches a class map that contains an inspection command, and then matches another class map that also has an inspection command, only the first matching class is used. For example, SNMP matches the inspection_default class. To enable SNMP inspection, enable SNMP inspection for the default class in Step 5. Do not add another class that matches SNMP.

For example, to limit inspection to traffic from 10.1.1.0 to 192.168.1.0 using the default class map, enter the following commands:

```
hostname(config)# access-list inspect extended permit ip 10.1.1.0 255.255.255.0
192.168.1.0 255.255.255.0
hostname(config)# class-map inspection_default
hostname(config-cmap)# match access-list inspect
```

View the entire class map using the following command:

```
hostname(config-cmap)# show running-config class-map inspection_default
!
class-map inspection_default
 match default-inspection-traffic
 match access-list inspect
!
```

To inspect FTP traffic on port 21 as well as 1056 (a non-standard port), create an access list that specifies the ports, and assign it to a new class map:

```
hostname(config)# access-list ftp_inspect extended permit tcp any any eq 21
hostname(config)# access-list ftp_inspect extended permit tcp any any eq 1056
hostname(config)# class-map new_inspection
hostname(config-cmap)# match access-list ftp_inspect
```

Step 2 (Optional) Some inspection engines let you control additional parameters when you apply the inspection to the traffic. See the following sections to configure either an inspection policy map or an application map for your application. Both inspection policy maps and application maps let you customize the inspection engine. Inspection policy maps use Modular Policy Framework commands like **policy-map type inspect**, and others. Application maps use commands in the form *protocol*-**map**.

- DCERPC—See the "Configuring a DCERPC Inspection Policy Map for Additional Inspection Control" section on page 21-16.

- ESMTP—See the "Configuring an ESMTP Inspection Policy Map for Additional Inspection Control" section on page 21-26.
- FTP—See the "The request-command deny Command" section on page 21-31.
- GTP—See the "GTP Maps and Commands" section on page 21-36.
- H.323—See the "H.225 Map Commands" section on page 21-50.
- HTTP—See the "Configuring an HTTP Inspection Policy Map for Additional Inspection Control" section on page 21-60.
- MGCP—See the "Configuring and Enabling MGCP Inspection" section on page 21-67.
- SIP—See the "Configuring a SIP Inspection Policy Map for Additional Inspection Control" section on page 21-78.
- SNMP—See the "Enabling and Configuring SNMP Application Inspection" section on page 21-98.

Step 3 To add or edit a Layer 3/4 policy map that sets the actions to take with the class map traffic, enter the following command:

```
hostname(config)# policy-map name
hostname(config-pmap)#
```

The default policy map is called "global_policy." This policy map includes the default inspections listed in the "Default Inspection Policy" section on page 21-4. If you want to modify the default policy (for example, to add or delete an inspection, or to identify an additional class map for your actions), then enter **global_policy** as the name.

Step 4 To identify the class map from Step 1 to which you want to assign an action, enter the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

If you are editing the default policy map, it includes the inspection_default class map. You can edit the actions for this class by entering **inspection_default** as the name. To add an additional class map to this policy map, identify a different name. You can combine multiple class maps in the same policy if desired, so you can create one class map to match certain traffic, and another to match different traffic. However, if traffic matches a class map that contains an inspection command, and then matches another class map that also has an inspection command, only the first matching class is used. For example, SNMP matches the inspection_default class map.To enable SNMP inspection, enable SNMP inspection for the default class in Step 5. Do not add another class that matches SNMP.

Step 5 Enable application inspection by entering the following command:

```
hostname(config-pmap-c)# inspect protocol
```

Table 21-2 lists the *protocol* values.

Table 21-2 Protocol Keywords

Keywords	Notes
ctiqbe	—
dcerpc [*policy_map_name*]	If you added a DCERPC inspection policy map according to "Configuring a DCERPC Inspection Policy Map for Additional Inspection Control" section on page 21-16, identify the map name in this command.
dns [*map_name*]	—

Table 21-2 Protocol Keywords (continued)

Keywords	Notes
esmtp [*policy_map_name*]	If you added an ESMTP inspection policy map according to "Configuring an ESMTP Inspection Policy Map for Additional Inspection Control" section on page 21-26, identify the map name in this command.
ftp [**strict** [*map_name*]]	Use the **strict** keyword to increase the security of protected networks by preventing web browsers from sending embedded commands in FTP requests. See the "Using the strict Option" section on page 21-30 for more information. If you added an FTP application map according to "The request-command deny Command" section on page 21-31, identify the map name in this command.
gtp [*map_name*]	If you added a GTP application map according to the "GTP Maps and Commands" section on page 21-36, identify the map name in this command.
h323 h225 [*map_name*]	If you added an H.225 application map according to "H.225 Map Commands" section on page 21-50, identify the map name in this command.
h323 ras [*map_name*]	—
http [*policy_map_name*]	If you added an HTTP inspection policy map according to the "Configuring an HTTP Inspection Policy Map for Additional Inspection Control" section on page 21-60, identify the map name in this command.
icmp	—
icmp error	—
ils	—
mgcp [*map_name*]	If you added an MGCP inspection policy map according to "Configuring and Enabling MGCP Inspection" section on page 21-67, identify the map name in this command.
netbios	—
pptp	—
rsh	—
rtsp	—
sip [*policy_map_name*]	If you added a SIP inspection policy map according to "Configuring a SIP Inspection Policy Map for Additional Inspection Control" section on page 21-78, identify the map name in this command.
skinny	—
snmp [*map_name*]	If you added an SNMP application map according to "Enabling and Configuring SNMP Application Inspection" section on page 21-98, identify the map name in this command.
sqlnet	—

Table 21-2 Protocol Keywords (continued)

Keywords	Notes
sunrpc	The default class map includes UDP port 111; if you want to enable Sun RPC inspection for TCP port 111, you need to create a new class map that matches TCP port 111, add the class to the policy, and then apply the **inspect sunrpc** command to that class.
tftp	—
waas	—
xdmcp	—

Step 6 To activate the policy map on one or more interfaces, enter the following command:

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

Where **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. By default, the default policy map, "global_policy," is applied globally. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

CTIQBE Inspection

This section describes how to enable CTIQBE application inspection and change the default port configuration. This section includes the following topics:

CTIQBE Inspection Overview

The **inspect ctiqbe** command enables CTIQBE protocol inspection, which supports NAT, PAT, and bidirectional NAT. This enables Cisco IP SoftPhone and other Cisco TAPI/JTAPI applications to work successfully with Cisco CallManager for call setup across the FWSM.

TAPI and JTAPI are used by many Cisco VoIP applications. CTIQBE is used by Cisco TSP to communicate with Cisco CallManager.

Limitations and Restrictions

The following summarizes limitations that apply when using CTIQBE application inspection:

- CTIQBE application inspection does not support configurations with the **alias** command.

- Stateful Failover of CTIQBE calls is *not* supported.
- Entering the **debug ctiqbe** command may delay message transmission, which may have a performance impact in a real-time environment. When you enable this debugging or logging and Cisco IP SoftPhone seems unable to complete call setup through the FWSM, increase the timeout values in the Cisco TSP settings on the system running Cisco IP SoftPhone.

The following summarizes special considerations when using CTIQBE application inspection in specific scenarios:

- If two Cisco IP SoftPhones are registered with different Cisco CallManagers, which are connected to different interfaces of the FWSM, calls between these two phones fails.
- When Cisco CallManager is located on the higher security interface compared to Cisco IP SoftPhones, if NAT or outside NAT is required for the Cisco CallManager IP address, the mapping must be static as Cisco IP SoftPhone requires the Cisco CallManager IP address to be specified explicitly in its Cisco TSP configuration on the PC.
- When using PAT or Outside PAT, if the Cisco CallManager IP address is to be translated, its TCP port 2748 must be statically mapped to the **same port** of the PAT (interface) address for Cisco IP SoftPhone registrations to succeed. The CTIQBE listening port (TCP 2748) is fixed and is not user-configurable on Cisco CallManager, Cisco IP SoftPhone, or Cisco TSP.

Enabling and Configuring CTIQBE Inspection

To enable CTIQBE inspection or change the default port used for receiving CTIQBE traffic, perform the following steps:

Step 1 Create a class map or modify an existing class map to identify CTIQBE traffic. Use the **class-map** command to do so, as follows.

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 2 Use the **match port** command to identify CTIQBE traffic, as follows:

```
hostname(config-cmap)# match port tcp eq 2748
```

Step 3 Create a policy map or modify an existing policy map that you want to use to apply the CTIQBE inspection engine to FTP traffic. To do so, use the **policy-map** command, as follows.

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 4 Specify the class map, created in Step 1, that identifies the CTIQBE traffic. Use the **class** command to do so, as follows.

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where *class_map_name* is the name of the class map you created in Step 1. The CLI enters the policy map class configuration mode and the prompt changes accordingly.

Step 5 Enable CTIQBE application inspection.

```
hostname(config-pmap-c)# inspect ctiqbe
```

Step 6 Use the **service-policy** command to apply the policy map globally or to a specific interface, as follows:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname(config)#
```

where *policy_map_name* is the policy map you configured in Step 3. If you want to apply the policy map to traffic on all the interfaces, use the **global** option. If you want to apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

The FWSM begins inspecting CTIQBE traffic, as specified.

Example 21-1 Enabling and Configuring CTIQBE Inspection

The following example creates a class map to match CTIQBE traffic on the default port (2748) and enables CTIQBE inspection in the policy using the class matching CTIQBE traffic. The service policy is then applied to the outside interface.

```
hostname(config)# class-map ctiqbe_port
hostname(config-cmap)# match port tcp eq 2748
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class ctiqbe_port
hostname(config-pmap-c)# inspect ctiqbe
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

Verifying and Monitoring CTIQBE Inspection

The **show ctiqbe** command displays information regarding the CTIQBE sessions established across the FWSM. It shows information about the media connections allocated by the CTIQBE inspection engine.

The following is sample output from the **show ctiqbe** command under the following conditions. There is only one active CTIQBE session setup across the FWSM. It is established between an internal CTI device (for example, a Cisco IP SoftPhone) at local address 10.0.0.99 and an external Cisco CallManager at 209.165.201.2, where TCP port 2748 is the Cisco CallManager. The heartbeat interval for the session is 120 seconds.

```
hostname# # show ctiqbe

Total: 1
        LOCAL           FOREIGN         STATE   HEARTBEAT
-------------------------------------------------------------
1       10.0.0.99/1117  209.165.201.2/2748        1       120
        ---------------------------------------------
        RTP/RTCP: PAT xlates: mapped to 209.165.201.2(1028 - 1029)
        ---------------------------------------------
        MEDIA: Device ID 27     Call ID 0
               Foreign 209.165.201.2    (1028 - 1029)
               Local   209.165.201.3    (26822 - 26823)
        ---------------------------------------------
```

The CTI device has already registered with the CallManager. The device internal address and RTP listening port is PATed to 209.165.201.2 UDP port 1028. Its RTCP listening port is PATed to UDP 1029.

The line beginning with `RTP/RTCP: PAT xlates:` appears only if an internal CTI device has registered with an external CallManager and the CTI device address and ports are PATed to that external interface. This line does not appear if the CallManager is located on an internal interface, or if the internal CTI device address and ports are NATed to the same external interface that is used by the CallManager.

The output indicates a call has been established between this CTI device and another phone at 209.165.201.3. The RTP and RTCP listening ports of the other phone are UDP 26822 and 26823. The other phone locates on the same interface as the CallManager because the FWSM does not maintain a CTIQBE session record associated with the second phone and CallManager. The active call leg on the CTI device side can be identified with Device ID 27 and Call ID 0.

The following is sample output from the **show xlate debug** command for these CTIBQE connections:

```
hostname# show xlate debug
3 in use, 3 most used
Flags:  D - DNS, d - dump, I - identity, i - inside, n - no random,
        r - portmap, s - static
TCP PAT from inside:10.0.0.99/1117 to outside:209.165.201.2/1025 flags ri idle 0:00:22
timeout 0:00:30
UDP PAT from inside:10.0.0.99/16908 to outside:209.165.201.2/1028 flags ri idle 0:00:00
timeout 0:04:10
UDP PAT from inside:10.0.0.99/16909 to outside:209.165.201.2/1029 flags ri idle 0:00:23
timeout 0:04:10
```

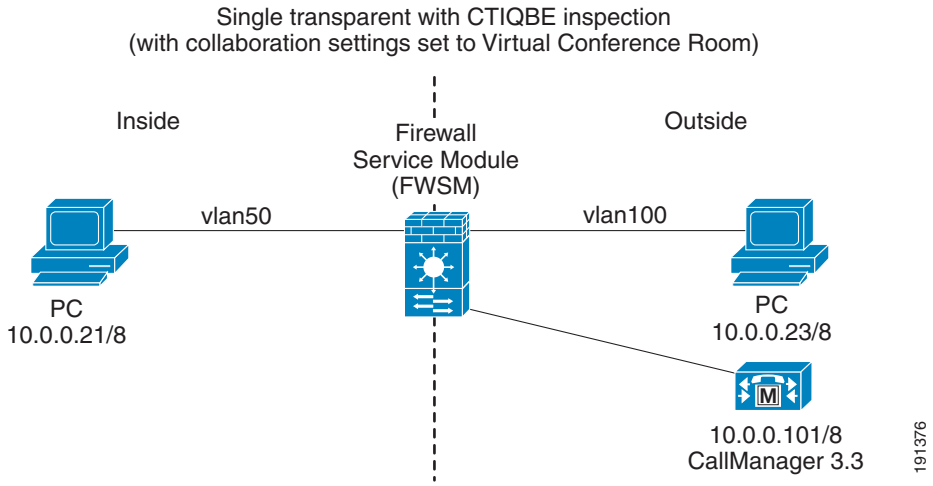
The **show conn state ctiqbe** command displays the status of CTIQBE connections. In the output, the media connections allocated by the CTIQBE inspection engine are denoted by a ‘C’ flag. The following is sample output from the **show conn state ctiqbe** command.

```
hostname# show conn state ctiqbe
1 in use, 10 most used
hostname# show conn state ctiqbe detail
1 in use, 10 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       B - initial SYN from outside, C - CTIQBE media, D - DNS, d - dump,
       E - outside back connection, F - outside FIN, f - inside FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
       i - incomplete, J - GTP, j - GTP data, k - Skinny media,
       M - SMTP data, m - SIP media, O - outbound data, P - inside back connection,
       q - SQL*Net data, R - outside acknowledged FIN,
       R - UDP RPC, r - inside acknowledged FIN, S - awaiting inside SYN,
       s - awaiting outside SYN, T - SIP, t - SIP transient, U - up
```

CTIQBE Sample Configurations

The following figure shows a sample configuration for a single transparent firewall for Cisco IP SoftPhone (Figure 21-2).

Figure 21-2 Single Transparent Firewall for Cisco IP SoftPhone (Virtual Conference)



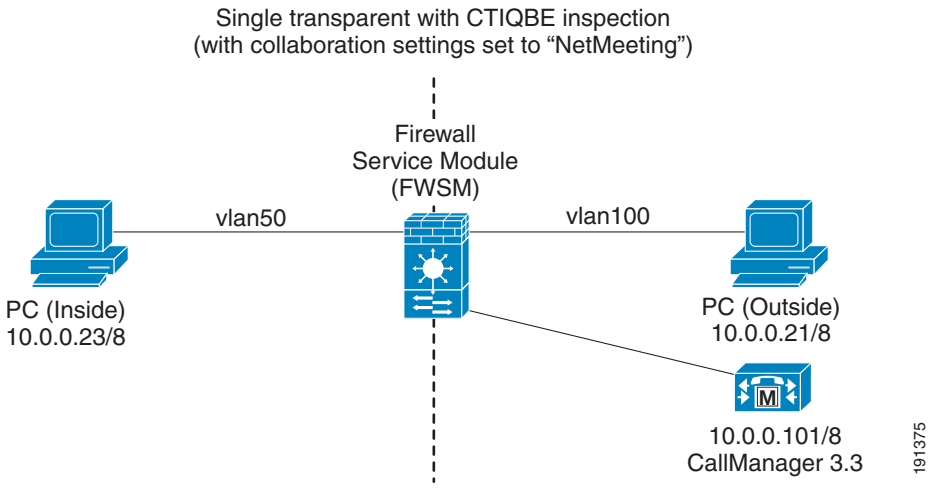
See the following configuration for this example:

```
firewall transparent
!
interface Vlan50
nameif inside
bridge-group 1
security-level 100
!
interface Vlan100
nameif outside
bridge-group 1
security-level 0
!
interface BVI1
ip address 10.0.0.30 255.0.0.0
!
access-list voice extended permit tcp any any eq ctiqbe
access-list voice extended permit tcp any any eq 1503
!
access-group voice in interface inside
access-group voice in interface outside
!
policy-map global_policy
class inspection_default
inspect ctiqbe
!
```

Note TCP port 1503 must be allowed to pass through the security appliance for virtual conference room collaboration to work with Cisco IP SoftPhone through the security appliance.

The following figure shows a sample configuration for a single transparent firewall for Cisco IP SoftPhone with NetMeeting enabled (Figure 21-3). Cisco IP SoftPhone is configured with the collaboration setting of NetMeeting.

Figure 21-3 Single Transparent Firewall for Cisco IP SoftPhone (Virtual Conference) with NetMeeting



See the following configuration for this example:

```
firewall transparent
!
interface Vlan50
nameif inside
bridge-group 1
security-level 100
!
interface Vlan100
nameif outside
bridge-group 1
security-level 0
!
interface BVI1
ip address 10.0.0.30 255.0.0.0
!
access-list voice extended permit tcp any any eq ctiqbe
access-list voice extended permit tcp any any eq h323
access-list voice extended permit tcp any any eq 1503
!
access-group voice in interface inside
access-group voice in interface outside
!
policy-map global_policy
class inspection_default
inspect ctiqbe
!
```

Note To allow successful collaboration and application sharing, TCP ports 1503 and 1720 must be allowed to pass through.

The following is sample output for the **show conn detail** command:

```
hostname# show conn detail
25 in use, 33 most used
Flags:  A - awaiting inside ACK to SYN,a - awaiting outside ACK to SYN
        B - initial SYN from outsideC - CTIQBE media, D - DNS, d - dump,
```

```
        E - outside back connection, F - outside FIN, f - inside FIN,
        G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
        i - incomplete, J - GTP, j - GTP data, k - Skinny media,
        M - SMTP data, m - SIP media, O - outbound data, P - inside back connection,
        q - SQL*Net data, R - outside acknowledged FIN,
        R - UDP SUNRPC, r - inside acknowledged FIN, S - awaiting inside SYN,
        s - awaiting outside SYN, T - SIP, t - SIP transient, U - up
Network Processor 1 connection
TCP out 10.0.0.101:2748 in 10.0.0.23:3598 idle 0:00:09 Bytes 103065 FLAGS - UOI
UDP out 10.0.0.21:30504 in 10.0.0.23:3650 idle 0:00:00 Bytes 4810406
 FLAGS - C
UDP out 10.0.0.21:1436 in 10.0.0.23:19972 idle 0:00:00 Bytes 4813240
 FLAGS - C
TCP out 10.0.0.21:1437 in 10.0.0.23:1720 idle 0:07:04 Bytes 1027 FLAGS - UBOIh
UDP out 10.0.0.21:49608 in 10.0.0.23:49608 idle 0:00:10 Bytes 241836
 FLAGS - H
UDP out 10.0.0.21:49609 in 10.0.0.23:49609 idle 0:00:01 Bytes 17480
 FLAGS - H
TCP out 10.0.0.21:1440 in 10.0.0.23:1503 idle 0:06:58 Bytes 4488 FLAGS - UBOI
TCP out 10.0.0.21:1441 in 10.0.0.23:1503 idle 0:04:50 Bytes 17888 FLAGS - UBOI
TCP out 10.0.0.21:1442 in 10.0.0.23:1503 idle 0:04:50 Bytes 471135 FLAGS - UBOI
 Network Processor 2 connections
Multicast sessions:
 Network Processor 1 connections
 Network Processor 2 connections
IPv6 connections:
```

DCERPC Inspection

DCERPC is a protocol widely used by Microsoft distributed client and server applications that allows software clients to execute programs on a server remotely.

This typically involves a client querying a server called the Endpoint Mapper listening on a well known port number for the dynamically allocated network information of a required service. The client then sets up a secondary connection to the server instance providing the service. The security appliance allows the appropriate port number and network address and also applies NAT, if needed, for the secondary connection.

DCERPC inspect maps inspect for native TCP communication between the EPM and client on well known TCP port 135. Map and lookup operations of the EPM are supported for clients. Client and server can be located in any security zone. The embedded server IP address and Port number are received from the applicable EPM response messages. Because a client may attempt multiple connections to the server port returned by EPM, multiple use of pinholes are allowed, which have user configurable timeouts.

Configuring a DCERPC Inspection Policy Map for Additional Inspection Control

To specify additional DCERPC inspection parameters, create a DCERPC inspection policy map. You can then apply the inspection policy map when you enable DCERPC inspection according to the "Configuring Application Inspection" section on page 21-6.

To create a DCERPC inspection policy map, perform the following steps:

Step 1 Create a DCERPC inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect dcerpc policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 2 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 3 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To configure the timeout for DCERPC pinholes and override the global system pinhole timeout of two minutes, enter the following command:

```
hostname(config-pmap-p)# timeout pinhole hh:mm:ss
```

Where the *hh:mm:ss* argument is the timeout for pinhole connections. Value is between 0:0:1 and 1193:0:0.

c. To configure options for the endpoint mapper traffic, enter the following command:

```
hostname(config-pmap-p)# endpoint-mapper [epm-service-only] [lookup-operation
[timeout hh:mm:ss]]
```

Where the *hh:mm:ss* argument is the timeout for pinholes generated from the lookup operation. If no timeout is configured for the lookup operation, the timeout pinhole command or the default is used. The **epm-service-only** keyword enforces endpoint mapper service during binding so that only its service traffic is processed. The **lookup-operation** keyword enables the lookup operation of the endpoint mapper service.

The following example shows how to define a DCERPC inspection policy map with the timeout configured for DCERPC pinholes.

```
hostname(config)# policy-map type inspect dcerpc dcerpc_map
hostname(config-pmap)# timeout pinhole 0:10:00

hostname(config)# class-map dcerpc
hostname(config-cmap)# match port tcp eq 135

hostname(config)# policy-map global-policy
hostname(config-pmap)# class dcerpc
hostname(config-pmap-c)# inspect dcerpc dcerpc-map

hostname(config)# service-policy global-policy global
```

DNS Inspection

This section describes how to manage DNS application inspection. This section includes the following topics:

How DNS Application Inspection Works

The FWSM tears down the DNS session associated with a DNS query as soon as the DNS reply is forwarded by the FWSM. The FWSM also monitors the message exchange to ensure that the ID of the DNS reply matches the ID of the DNS query.

When DNS inspection is enabled, which is the default, the FWSM performs the following additional tasks:

- Translates the DNS record based on the configuration completed using the **alias**, **static** and **nat** commands (DNS Rewrite). Translation only applies to the A-record in the DNS reply; therefore, DNS Rewrite does not affect reverse lookups, which request the PTR record.

Note DNS Rewrite is not applicable for PAT because multiple PAT rules are applicable for each A-record and the PAT rule to use is ambiguous.

- Enforces the maximum DNS message length (the default is 512 bytes and the maximum length is 65535 bytes). The FWSM performs reassembly as needed to verify that the packet length is less than the maximum length configured. The FWSM drops the packet if it exceeds the maximum length.

Note If you enter the **inspect dns** command without the **maximum-length** option, the DNS packet size is not checked.

- Enforces a domain-name length of 255 bytes and a label length of 63 bytes.
- Verifies the integrity of the domain-name referred to by the pointer if compression pointers are encountered in the DNS message.
- Checks to see if a compression pointer loop exists.

A single connection is created for multiple DNS sessions, as long as they are between the same two hosts, and the sessions have the same 5-tuple (source/destination IP address, source/destination port, and protocol). DNS identification is tracked by *app_id*, and the idle timer for each app_id runs independently.

Because the app_id expires independently, a legitimate DNS response can only pass through the FWSM within a limited period of time and there is no resource build-up. However, if you enter the **show conn** command, you will see the idle timer of a DNS connection being reset by a new DNS session. This is due to the nature of the shared DNS connection and is by design.

How DNS Rewrite Works

When DNS inspection is enabled, DNS Rewrite provides full support for NAT of DNS messages originating from any interface.

If a client on an inside network requests DNS resolution of an inside address from a DNS server on an outside interface, the DNS A-record is translated correctly. If the DNS inspection engine is disabled, the A-record is not translated.

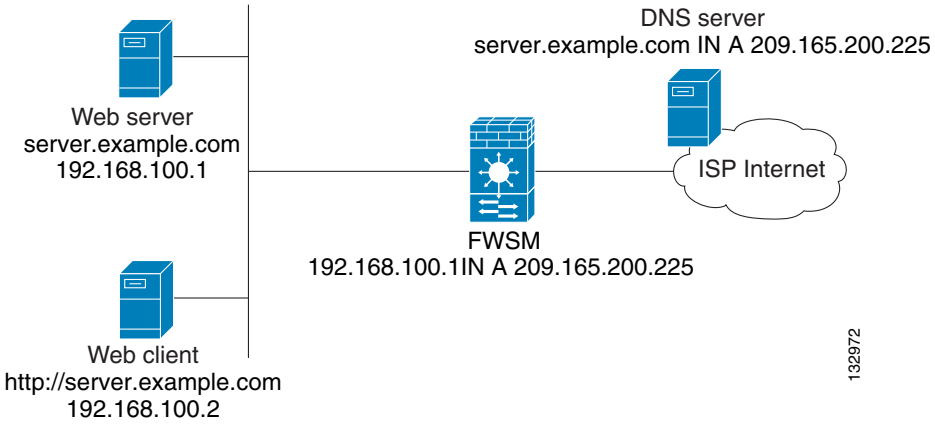
As long as DNS inspection remains enabled, you can configure DNS Rewrite using the **alias**, **static**, or **nat** commands. For details about the configuration required see the "Configuring DNS Rewrite" section on page 21-19.

DNS Rewrite performs two functions:

- Translating a public address (the routable or "mapped" address) in a DNS reply to a private address (the "real" address) when the DNS client is on a private interface.
- Translating a private address to a public address when the DNS client is on the public interface.

In Figure 21-4, the DNS server resides on the external (ISP) network. On the FWSM, a **static** command maps the real address of the web server (192.168.100.1) to the ISP-assigned address (209.165.201.5). When a web client on the inside interface attempts to access the web server with the URL http://server.example.com, the host running the web client sends a DNS request to the DNS server to resolve the IP address of the web server. The FWSM translates the non-routable source address in the IP header and forwards the request to the ISP network on its outside interface. When the DNS reply is returned, the FWSM applies address translation not only to the destination address, but also to the embedded IP address of the web server, which is contained in the A-record in the DNS reply. As a result, the web client on the inside network gets the correct address for connecting to the web server on the inside network. For the exact NAT and DNS configuration for this example, see Example 21-2. For configuration instructions for scenarios similar to this one, see the "Configuring DNS Rewrite with Two NAT Zones" section on page 21-21.

Figure 21-4 DNS Rewrite with Two NAT Zones



DNS Rewrite also works if the client making the DNS request is on a DMZ network and the DNS server is on an inside interface. For an illustration and configuration instructions for this scenario, see the "DNS Rewrite with Three NAT Zones" section on page 21-22.

Configuring DNS Rewrite

You configure DNS Rewrite using the **alias**, **static**, or **nat** commands. The **alias** and **static** command can be used interchangeably; however, we recommend using the **static** command for new deployments because it is more precise and unambiguous. Also, DNS Rewrite is optional when using the **static** command.

This section describes how to use the **alias** and **static** commands to configure DNS Rewrite. It provides configuration procedures for using the **static** command in a simple scenario and in a more complex scenario. Using the **nat** command is similar to using the **static** command except that DNS Rewrite is based on dynamic translation instead of a static mapping.

This section includes the following topics:

For detailed syntax and additional functions for the **alias**, **nat**, and **static** command, see the appropriate command page in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Using the Alias Command for DNS Rewrite

The **alias** command causes the FWSM to translate addresses on an IP network residing on *any* interface into addresses on another IP network connected through a different interface. The syntax for this command is as follows.

```
hostname(config)# alias (inside) mapped-address real-address
```

The following example specifies that the real address (192.168.100.10) on any interface *except* the inside interface will be translated to the mapped address (**209.165.200.225**) on the inside interface. Notice that the location of 192.168.100.10 is not precisely defined.

```
hostname(config)# alias (inside) 209.165.200.225 192.168.100.10
```

Note If you use the **alias** command to configure DNS Rewrite, proxy ARP will be performed for the mapped address. To prevent this, disable Proxy ARP by entering the **sysopt noproxyarp internal_interface** command after entering the **alias** command.

Using the Static Command for DNS Rewrite

The **static** command causes addresses on an IP network residing on a *specific* interface to be translated into addresses on another IP network on a different interface. The syntax for this command is as follows.

```
hostname(config)# static (inside,outside) mapped-address real-address dns
```

The following example specifies that the address 192.168.100.10 on the inside interface is translated into 209.165.201.5 on the outside interface:

```
hostname(config)# static (inside,outside) 209.165.200.225 192.168.100.10 dns
```

Note Using the **nat** command is similar to using the **static** command except that DNS Rewrite is based on dynamic translation instead of a static mapping.

Configuring DNS Rewrite with Two NAT Zones

To implement a DNS Rewrite scenario similar to the one shown in Figure 21-4, perform the following steps:

Step 1 Create a static translation for the web server, as follows:

```
hostname(config)# static (inside,outside) mapped-address real-address netmask
255.255.255.255 dns
```

where the arguments are as follows:

- *inside*—The name of the inside interface of the FWSM.
- *outside*—The name of the outside interface of the FWSM.
- *mapped-address*—The translated IP address of the web server.
- *real-address*—The real IP address of the web server.

Step 2 Create an access list that permits traffic to the port that the web server listens to for HTTP requests.

```
hostname(config)# access-list acl-name permit tcp any host mapped-address eq port
```

where the arguments are as follows:

acl-name—The name you give the access-list.

mapped-address—The translated IP address of the web server.

port—The TCP port that the web server listens to for HTTP requests.

Step 3 Apply the access list created in Step 2 to the outside interface. To do so, use the **access-group** command, as follows.

```
hostname(config)# access-group acl-name in interface outside
```

Step 4 If DNS inspection is disabled or if you want to change the maximum DNS packet length, configure DNS inspection. DNS application inspection is enabled by default with a maximum DNS packet length of 512 bytes. For configuration instructions, see the "Configuring DNS Inspection" section on page 21-24.

Step 5 On the public DNS server, add an A-record for the web server, such as:

```
domain-qualified-hostname. IN A mapped-address
```

where *domain-qualified-hostname* is the hostname with a domain suffix, as in server.example.com. The period after the hostname is important. *mapped-address* is the translated IP address of the web server.

The following example configures the FWSM for the scenario shown in Figure 21-4. It assumes DNS inspection is already enabled.

Example 21-2 DNS Rewrite with Two NAT Zones

```
hostname(config)# static (inside,outside) 209.165.200.225 192.168.100.1 netmask
255.255.255.255 dns
hostname(config)# access-list 101 permit tcp any host 209.165.200.225 eq www
hostname(config)# access-group 101 in interface outside
```

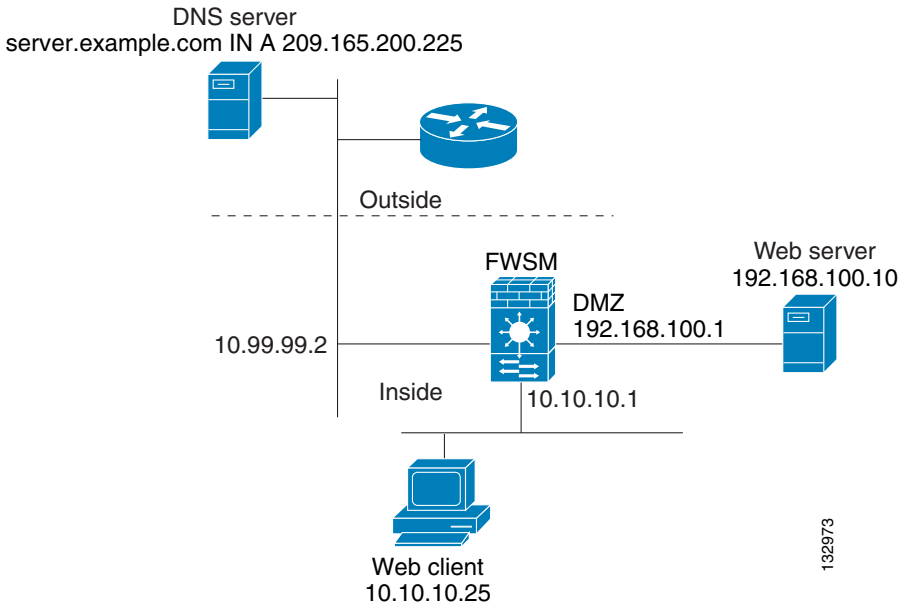
This configuration requires the following A-record on the DNS server:

```
server.example.com. IN A 209.165.200.225
```

DNS Rewrite with Three NAT Zones

Figure 21-5 illustrates a more complex scenario: how DNS inspection allows NAT to operate transparently with a DNS server with minimal configuration. For configuration instructions for scenarios like this one, see the "Configuring DNS Rewrite with Three NAT Zones" section on page 21-23.

Figure 21-5 DNS Rewrite with Three NAT Zones



In Figure 21-5, a web server, server.example.com, has the real address 192.168.100.10 on the DMZ interface of the FWSM. A web client with the IP address 10.10.10.25 is on the inside interface and a public DNS server is on the outside interface. The site NAT policies are as follows:

- The outside DNS server holds the authoritative address record for server.example.com.
- Hosts on the outside network can contact the web server with the domain name server.example.com through the outside DNS server or with the IP address 209.165.200.225.
- Clients on the inside network can access the web server with the domain name server.example.com through the outside DNS server or with the IP address 192.168.100.10.

When a host or client on any interface accesses the DMZ web server, it queries the public DNS server for the A-record of server.example.com. The DNS server returns the A-record showing that server.example.com binds to address 209.165.200.225.

When a web client on the *outside* network attempts to access http://server.example.com, the sequence of events is as follows:

1. The host running the web client sends the DNS server a request for the IP address of server.example.com.
2. The DNS server responds with the IP address 209.165.200.225 in the reply.
3. The web client sends its HTTP request to 209.165.200.225.
4. The packet from the outside host reaches the FWSM at the outside interface.
5. The static rule translates the address 209.165.200.225 to 192.168.100.10 and the FWSM directs the packet to the web server on the DMZ.

When a web client on the *inside* network attempts to access http://server.example.com, the sequence of events is as follows:

1. The host running the web client sends the DNS server a request for the IP address of server.example.com.
2. The DNS server responds with the IP address 209.165.200.225 in the reply.
3. The FWSM receives the DNS reply and submits it to the DNS application inspection engine.
4. The DNS application inspection engine does the following:
   a. Searches for any NAT rule to undo the translation of the embedded A-record address "[outside]:209.165.200.5". In this example, it finds the following static configuration.

      ```
      static (dmz,outside) 209.165.200.225 192.168.100.10 dns
      ```

   b. Uses the static rule to rewrite the A-record as follows because the **dns** option is included:

      ```
      [outside]:209.165.200.225 --> [dmz]:192.168.100.10
      ```

      Note If the **dns** option were not included with the **static** command, DNS Rewrite would not be performed and other processing for the packet continues.

   c. Searches for any NAT to translate the web server address, [dmz]:192.168.100.10, when communicating with the inside web client.

      No NAT rule is applicable, so application inspection completes.

      If a NAT rule (nat or static) were applicable, the **dns** option must also be specified. If the **dns** option were not specified, the A-record rewrite in step b would be reverted and other processing for the packet continues.
5. The FWSM sends the HTTP request to server.example.com on the DMZ interface.

Configuring DNS Rewrite with Three NAT Zones

To enable the NAT policies for the scenario in Figure 21-5, perform the following steps:

Step 1 Create a static translation for the web server on the DMZ network, as follows:

```
hostname(config)# static (dmz,outside) mapped-address real-address dns
```

where the arguments are as follows:

- *dmz*—The name of the DMZ interface of the FWSM.
- *outside*—The name of the outside interface of the FWSM.
- *mapped-address*—The translated IP address of the web server.
- *real-address*—The real IP address of the web server.

Step 2 Create an access list that permits traffic to the port that the web server listens to for HTTP requests.

```
hostname(config)# access-list acl-name permit tcp any host mapped-address eq port
```

where the arguments are as follows:

acl-name—The name you give the access-list.

mapped-address—The translated IP address of the web server.

port—The TCP port that the web server listens to for HTTP requests.

Step 3 Apply the access list created in Step 2 to the outside interface. To do so, use the **access-group** command, as follows:

```
hostname(config)# access-group acl-name in interface outside
```

Step 4 If DNS inspection is disabled or if you want to change the maximum DNS packet length, configure DNS inspection. DNS application inspection is enabled by default with a maximum DNS packet length of 512 bytes. For configuration instructions, see the "Configuring DNS Inspection" section on page 21-24.

Step 5 On the public DNS server, add an A-record for the web server, such as:

```
domain-qualified-hostname. IN A mapped-address
```

where *domain-qualified-hostname* is the hostname with a domain suffix, as in server.example.com. The period after the hostname is important. *mapped-address* is the translated IP address of the web server.

The following example configures the FWSM for the scenario shown in Figure 21-5. It assumes DNS inspection is already enabled.

Example 21-3 DNS Rewrite with Three NAT Zones

```
hostname(config)# static (dmz,outside) 209.165.200.225 192.168.100.10 dns
hostname(config)# access-list 101 permit tcp any host 209.165.200.225 eq www
hostname(config)# access-group 101 in interface outside
```

This configuration requires the following A-record on the DNS server:

```
server.example.com. IN A 209.165.200.225
```

Configuring DNS Inspection

DNS inspection is enabled by default.

To enable DNS inspection (if it has been previously disabled) or to change the default port used for receiving DNS traffic, perform the following steps:

Step 1 Create a class map or modify an existing class map to identify DNS traffic. Use the **class-map** command to do so, as follows.

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 2 Use the **match port** command to identify DNS traffic. The default port for DNS is UDP port 53.

```
hostname(config-cmap)# match port udp eq 53
```

Step 3 Create a policy map or modify an existing policy map that you want to use to apply the DNS inspection engine to FTP traffic. To do so, use the **policy-map** command, as follows.

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 4 Enable DNS application inspection. To do so, use the **inspect dns** command, as follows.

```
hostname(config-pmap-c)# inspect dns [maximum-length max-pkt-length]
```

To change the maximum DNS packet length from the default (512), use the **maximum-length** argument and replace *max-pkt-length* with a numeric value. Longer packets are dropped. To disable checking the DNS packet length, enter the **inspect dns** command without the **maximum-length** keyword.

Step 5 Use the **service-policy** command to apply the policy map globally or to a specific interface, as follows:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname(config)#
```

where *policy_map_name* is the policy map you configured in Step 3. If you want to apply the policy map to traffic on all the interfaces, use the **global** option. If you want to apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

The FWSM begins inspecting DNS traffic, as specified.

Example 21-4 Enabling and Configuring DNS Inspection

The following example creates a class map to match DNS traffic on the default port (53), and enables DNS inspection in the sample_policy policy map, and applies DNS inspection to the outside interface.

```
hostname(config)# class-map dns_port
hostname(config-cmap)# match port udp eq 53
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class dns_port
hostname(config-pmap-c)# inspect dns maximum-length 1500
hostname(config-pmap-c)# service-policy sample_policy interface outside
```

Verifying and Monitoring DNS Inspection

To view information about the current DNS connections, enter the following command:

```
hostname# show conn
```

For connections using a DNS server, the source port of the connection may be replaced by the IP address of DNS server in the **show conn** command output.

A single connection is created for multiple DNS sessions, as long as they are between the same two hosts, and the sessions have the same 5-tuple (source/destination IP address, source/destination port, and protocol). DNS identification is tracked by app_id, and the idle timer for each app_id runs independently.

Because the app_id expires independently, a legitimate DNS response can only pass through the FWSM within a limited period of time and there is no resource build-up. However, when you enter the **show conn** command, you see the idle timer of a DNS connection being reset by a new DNS session. This is due to the nature of the shared DNS connection and is by design.

To display the statistics for DNS application inspection, enter the **show service-policy** command. The following is sample output from the **show service-policy** command.

```
hostname# show service-policy
Interface outside:
```

```
Service-policy: sample_policy
  Class-map: dns_port
    Inspect: dns maximum-length 1500, packet 0, drop 0, reset-drop 0
```

DNS Guard

When a client sends a DNS request to an external DNS server, only the first response is accepted by the FWSM. All additional responses from other DNS servers are dropped by the FWSM.

After the client issues a DNS request, a dynamic hole allows UDP packets to return from the DNS server. When the FWSM receives a response from the first DNS server, the connection that was created in the accelerated path is dropped so that subsequent responses from other DNS servers are dropped by the FWSM. The UDP DNS connection is deleted immediately rather than marking the connection for deletion.

The FWSM creates a session-lookup key based on the source and destination IP address along with the protocol and the DNS ID instead of the source and destination ports.

If the DNS client and DNS server use TCP for DNS, the connection is cleared like a normal TCP connection.

However, if clients receive DNS responses from multiple DNS servers, you can disable the default DNS behavior on a per context basis. When DNS Guard is disabled, a response from the first DNS server does not delete the connection and the connection is treated as a normal UDP connection.

DNS Guard is enabled by default.

To disable DNS Guard, enter the following commands:

```
hostname(config)# no dns-guard
hostname(config)# show running-config | inc dns-guard
no dns-guard
hostname(config)#
```

ESMTP Inspection

ESMTP inspection detects attacks, including spam, phising, malformed message attacks, buffer overflow/underflow attacks. It also provides support for application security and protocol conformance, which enforce the sanity of the ESMTP messages as well as detect several attacks, block senders/receivers, and block mail relay.

Configuring an ESMTP Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an ESMTP inspection policy map. You can then apply the inspection policy map when you enable ESMTP inspection according to the "Configuring Application Inspection" section on page 21-6.

To create an ESMTP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 19-11. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 19-14.

Step 3 Create an ESMTP inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect esmtp policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 4 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 5 To apply actions to matching traffic, perform the following steps.

a. Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 19-7.

Step 6 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To configure a local domain name, enter the following command:

```
hostname(config-pmap-p)# mail-relay domain-name action [drop-connection / log]]
```

Where the **drop-connection** action closes the connection. The **log** action sends a system log message when this policy map matches traffic.

c. To enforce banner obfuscation, enter the following command:

```
hostname(config-pmap-p)# mask-banner
```

d. (Optional) To detect special characters in sender or receiver email addresses, enter the following command:

```
hostname(config-pmap-p)# special-character action [drop-connection | log]]
```

Using this command detects pipe (|), backquote (`) and null characters.

e. (Optional) To match the body length or body line length, enter the following command:

```
hostname(config-pmap-p)# match body [line] length gt length
```

Where *length* is the length of the message body or the length of a line in the message body.

f. (Optional) To match an ESMTP command verb, enter the following command:

```
hostname(config-pmap-p)# match cmd verb verb
```

Where *verb* is any of the following ESMTP commands:

```
AUTH|DATA|EHLO|ETRN||HELO|HELP|MAIL|NOOP|QUIT|RCPT|RSET|SAML|SOML|VRFY
```

g. (Optional) To match the number of recipient addresses, enter the following command:

```
hostname(config-pmap-p)# match cmd RCPT count gt count
```

Where *count* is the number of recipient addresses.

h. (Optional) To match the command line length, enter the following command:

```
hostname(config-pmap-p)# match cmd line length gt length
```

Where *length* is the command line length.

i. (Optional) To match the ehlo-reply-parameters, enter the following command:

```
hostname(config-pmap-p)# match ehlo-reply-parameter extensions
```

Where *extensions* are the ESMTP service extensions sent by the server in response to the EHLO message from the client. These extensions are implemented as a new command or as parameters to an existing command. *extensions* can be any of the following.

```
8bitmime|binarymime|checkpoint|dsn|ecode|etrn|others|pipelining|size|vrfy
```

j. (Optional) To match the header length or header line length, enter the following command:

```
hostname(config-pmap-p)# match header [line] length gt length
```

Where *length* is the number of characters in the header or line.

k. (Optional) To match the header to-fields count, enter the following command:

```
hostname(config-pmap-p)# match header to-fields count gt count
```

Where *count* is the number of recipients in the to-field of the header.

l. (Optional) To match the number of invalid recipients, enter the following command:

```
hostname(config-pmap-p)# match invalid-recipients count gt count
```

Where *count* is the number of invalid recipients.

m. (Optional) To match the type of MIME encoding scheme used, enter the following command:

```
hostname(config-pmap-p)# match mime encoding [7bit|8bit|base64|binary|others|
quoted-printable]
```

n. (Optional) To match the MIME filename length, enter the following command:

```
hostname(config-pmap-p)# match mime filename length gt length
```

Where *length* is the length of the *filename* in the range 1 to 1000.

o. (Optional) To match the MIME file type, enter the following command:

```
hostname(config-pmap-p)# match mime filetype regex [name | class name]
```

Where *name* or *class name* is the regular expression that matches a file type or a class map. The regular expression used to match a class map can select multiple file types.

p. (Optional) To match a sender address, enter the following command:

```
hostname(config-pmap-p)# match sender-address regex [name | class name]
```

Where *name* or *class name* is the regular expression that matches a sender address or a class map. The regular expression used to match a class map can select multiple sender addresses.

q. (Optional) To match the length of a sender's address, enter the following command:

```
hostname(config-pmap-p)# match sender-address length gt length
```

Where *length* is the number of characters in the sender's address.

The following example shows how to define an ESMTP inspection policy map.

```
hostname(config)# regex user1 "user1@cisco.com"
hostname(config)# regex user2 "user2@cisco.com"
hostname(config)# regex user3 "user3@cisco.com"
hostname(config)# class-map type regex senders_black_list
hostname(config-cmap)# description "Regular expressions to filter out undesired senders"
hostname(config-cmap)# match regex user1
hostname(config-cmap)# match regex user2
hostname(config-cmap)# match regex user3

hostname(config)# policy-map type inspect esmtp advanced_esmtp_map
hostname(config-pmap)# match sender-address regex class senders_black_list
hostname(config-pmap-c)# drop-connection log

hostname(config)# policy-map outside_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect esmtp advanced_esmtp_map

hostname(config)# service-policy outside_policy interface outside
```

FTP Inspection

This section describes how the FTP inspection engine works and how you can change its configuration. This section includes the following topics:

FTP Inspection Overview

The FTP application inspection inspects the FTP sessions and performs four tasks:

- Prepares dynamic secondary data connection
- Tracks **ftp** command-response sequence
- Generates an audit trail
- NATs embedded IP address

FTP application inspection prepares secondary channels for FTP data transfer. Ports for these channels are negotiated through PORT or PASV commands. The channels are allocated in response to a file upload, a file download, or a directory listing event.

Note If you disable FTP inspection engines with the **no inspect ftp** command, outbound users can start connections only in passive mode, and all inbound FTP is disabled.

Using the strict Option

Using the **strict** option with the **inspect ftp** command increases the security of protected networks by preventing web browsers from sending embedded commands in FTP requests.

Tip To specify FTP commands that are not permitted to pass through the FWSM, create an FTP map and enter the **request-command deny** command in FTP map configuration mode.

After you enable the **strict** option on an interface, FTP inspection enforces the following behavior:

- An FTP command must be acknowledged before the FWSM allows a new command.
- The FWSM drops connections that send embedded commands.
- The 227 and PORT commands are checked to ensure they do not appear in an error string.

Caution Using the **strict** option may cause the failure of FTP clients that are not strictly compliant with FTP RFCs.

If the **strict** option is enabled, each **ftp** command and response sequence is tracked for the following anomalous activity:

- Truncated command—Number of commas in the PORT and PASV reply command is checked to see if it is five. If it is not five, then the PORT command is assumed to be truncated and the TCP connection is closed.
- Incorrect command—Checks the **ftp** command to see if it ends with <CR><LF> characters, as required by the RFC. If it does not, the connection is closed.
- Size of RETR and STOR commands—These are checked against a fixed constant. If the size is greater, then an error message is logged and the connection is closed.
- Command spoofing—The PORT command should always be sent from the client. The TCP connection is denied if a PORT command is sent from the server.
- Reply spoofing—PASV reply command (227) should always be sent from the server. The TCP connection is denied if a PASV reply command is sent from the client. This prevents the security hole when the user executes "227 xxxxx a1, a2, a3, a4, p1, p2."
- TCP stream editing—The FWSM closes the connection if it detects TCP stream editing.
- Invalid port negotiation—The negotiated dynamic port value is checked to see if it is less than 1024. As port numbers in the range from 1 to 1024 are reserved for well-known connections, if the negotiated port falls in this range, then the TCP connection is freed.
- Command pipelining—The number of characters present after the port numbers in the PORT and PASV reply command is cross checked with a constant value of 8. If it is more than 8, then the TCP connection is closed.
- The FWSM replaces the FTP server response to the SYST command with a series of Xs to prevent the server from revealing its system type to FTP clients. To override this default behavior, use the **no mask-syst-reply** command in FTP map configuration mode.

The request-command deny Command

The **request-command deny** command lets you control which FTP commands the FWSM allows for FTP traffic through the FWSM. This command is available in FTP map configuration mode; therefore, to make use of it, you must create an FTP map and use that map when you enable FTP inspection, per "Configuring FTP Inspection" section on page 21-32.

Table 21-3 lists the FTP commands that you can disallow by using the **request-command deny** command.

Table 21-3 FTP Map request-command deny Options

request-command deny Option	Purpose
appe	Disallows the command that appends to a file.
cdup	Disallows the command that changes to the parent directory of the current working directory.
dele	Disallows the command that deletes a file on the server.
get	Disallows the client command for retrieving a file from the server.
help	Disallows the command that provides help information.
mkd	Disallows the command that makes a directory on the server.
put	Disallows the client command for sending a file to the server.

Table 21-3 FTP Map request-command deny Options (continued)

request-command deny Option	Purpose
rmd	Disallows the command that deletes a directory on the server.
rnfr	Disallows the command that specifies rename-from filename.
rnto	Disallows the command that specifies rename-to filename.
site	Disallows the command that are specific to the server system. Usually used for remote administration.
stou	Disallows the command that stores a file using a unique filename.

Configuring FTP Inspection

FTP application inspection is enabled default, so you only need to perform the procedures in this section if you want to change the default FTP configuration, in any of the following ways:

- Enable the **strict** option.
- Identify specific FTP commands that are not permitted to pass through the FWSM.
- Change the default port number.

To configure FTP inspection, perform the following steps:

Step 1 Determine the ports to which FTP servers behind your FWSM listen. The default FTP port is TCP port 21; however, alternate ports are often used as a simple means to thwart attacks. To ensure that all FTP traffic is inspected, check your FTP servers for use of ports other than TCP port 21.

Step 2 Create a class map or modify an existing class map to identify FTP traffic. Use the **class-map** command to do so, as follows.

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 3 Identify traffic sent to the FTP ports you determined in Step 1. To do so, use a **match port** or **match access-list** command.

If you need to identify two or more non-contiguous ports, create an access list with the **access-list extended** command, add an ACE to match each port, and then use the **match access-list** command. The following commands show how to use an access list to identify multiple TCP ports with an access list.

```
hostname(config)# access-list acl-name any any tcp eq port_number_1
hostname(config)# access-list acl-name any any tcp eq port_number_2
hostname(config)# class-map class_map_name
hostname(config-cmap)# match access-list acl-name
```

If you need to identify a single port, use the **match port** command, as follows:

```
hostname(config-cmap)# match port tcp port_number
```

where *port_number* is the only TCP port listened to by FTP servers behind the FWSM.

If you need to identify a range of contiguous ports for a single protocol, use **match port** command with the **range** keyword, as follows:

```
hostname(config-cmap)# match port tcp range begin_port_number end_port_number
```

where *begin_port_number* is the lowest port in the range of FTP ports and *end_port_number* is the highest port.

Step 4 (Optional) If you want FTP inspection to do the following:

- Allow FTP servers to reveal their system type to FTP clients.
- Limit the allowed FTP commands.

then create and configure an FTP map. To do so, perform the following steps.

a. Create an FTP map that contains the additional parameters of FTP inspection. Use the **ftp-map** command to do so, as follows.

```
hostname(config-cmap)# ftp-map map_name
hostname(config-ftp-map)#
```

where *map_name* is the name of the FTP map. The CLI enters FTP map configuration mode.

b. (Optional) If you want to allow FTP servers from revealing their system type to FTP clients in responses to SYST messages, use the **no** form of the **mask-syst-reply** command, as follows:

```
hostname(config-ftp-map)# no mask-syst-reply
hostname(config-ftp-map)#
```

Note By default, when FTP inspection is enabled, responses to SYST messages are masked. If you disable SYST response masking, you can reenable it with the **mask-syst-response** command.

c. (Optional) If you want to disallow specific FTP commands, use the **request-command deny** command and specify each FTP command that you want to disallow, as follows:

```
hostname(config-ftp-map)# request-command deny ftp_command [ftp_command...]
hostname(config-ftp-map)#
```

where *ftp_command* with one or more FTP commands that you want to restrict. See Table 21-3 for a list of the FTP commands that you can restrict.

Step 5 Create a policy map or modify an existing policy map that you want to use to apply the FTP inspection engine to FTP traffic. To do so, use the **policy-map** command, as follows.

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 6 Specify the class map, created in Step 2, that identifies the FTP traffic. Use the **class** command to do so, as follows.

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where *class_map_name* is the name of the class map you created in Step 2. The CLI enters the policy map class configuration mode and the prompt changes accordingly.

Step 7 Enable FTP application inspection with the options you want. To do so, do one of the following.

- If you want to enable strict FTP inspection, use the **inspect ftp** command with the **strict** keyword, as follows:

```
hostname(config-pmap-c)# inspect ftp strict
```

- If you want to enable strict FTP inspection with an optional FTP map you may have configured in Step 4, use the **inspect ftp** command with the **strict** keyword and the FTP map name, as follows:

```
hostname(config-pmap-c)# inspect ftp strict ftp_map_name
```

- If you want to revert to default FTP inspection, use the **inspect ftp** command with no keywords, as follows:

```
hostname(config-pmap-c)# inspect ftp
```

Step 8 Use the **service-policy** command to apply the policy map globally or to a specific interface, as follows:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname(config)#
```

where *policy_map_name* is the policy map you configured in Step 5. If you want to apply the policy map to traffic on all the interfaces, use the **global** option. If you want to apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

The FWSM begins inspecting FTP traffic, as specified.

The following example shows how to identify FTP traffic, define a FTP map, define a policy, and apply the policy to the outside interface.

Example 21-5 Enabling and Configuring Strict FTP Inspection

```
hostname(config)# class-map ftp_port
hostname(config-cmap)# match port tcp eq 21
hostname(config-cmap)# ftp-map sample_map
hostname(config-ftp-map)# request-command deny put stou appe
hostname(config-ftp-map)# policy-map sample_policy
hostname(config-pmap)# class ftp_port
hostname(config-pmap-c)# inspect ftp strict sample_map
hostname(config-pmap-c)# service-policy sample_policy interface outside
```

Verifying and Monitoring FTP Inspection

FTP application inspection generates the following log messages:

- An Audit record 302002 is generated for each file that is retrieved or uploaded.
- The FTP command is checked to see if it is RETR or STOR and the retrieve and store commands are logged.
- The username is obtained by looking up a table providing the IP address.
- The username, source IP address, destination IP address, NAT address, and the file operation are logged.
- Audit record 201005 is generated if the secondary dynamic channel preparation failed due to memory shortage.

In conjunction with NAT, the FTP application inspection translates the IP address within the application payload. This is described in detail in RFC 959.

GTP Inspection

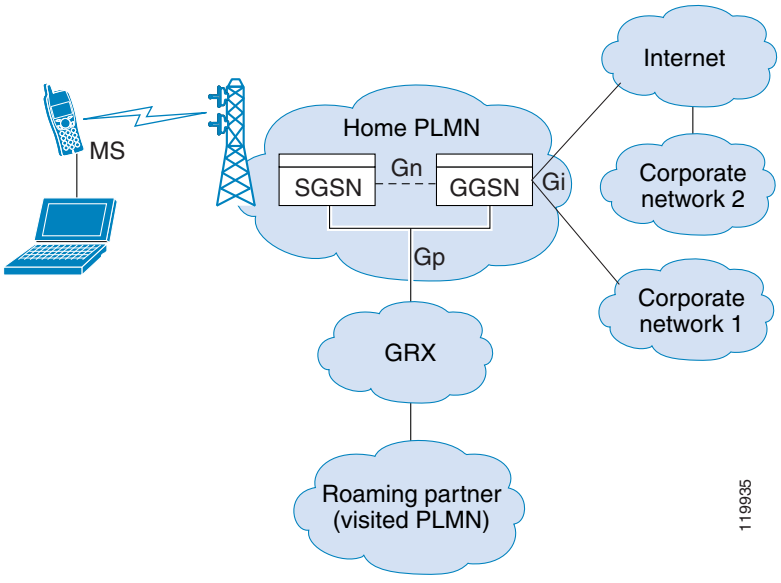
This section describes how the GTP inspection engine works and how you can change its configuration. This section includes the following topics:

Note GTP inspection requires a special license. If you enter GTP-related commands on a FWSM without the required license, the FWSM displays an error message.

GTP Inspection Overview

GPRS provides uninterrupted connectivity for mobile subscribers between GSM networks and corporate networks or the Internet. The GGSN is the interface between the GPRS wireless data network and other networks. The SGSN performs mobility, data session management, and data compression (See Figure 21-6).

Figure 21-6 GPRS Tunneling Protocol



The UMTS is the commercial convergence of fixed-line telephony, mobile, Internet and computer technology. UTRAN is the networking protocol used for implementing wireless networks in this system. GTP allows multi-protocol packets to be tunneled through a UMTS/GPRS backbone between a GGSN, an SGSN and the UTRAN.

GTP does not include any inherent security or encryption of user data, but using GTP with the FWSM helps protect your network against these risks.

The SGSN is logically connected to a GGSN using GTP. GTP allows multiprotocol packets to be tunneled through the GPRS backbone between GSNs. GTP provides a tunnel control and management protocol that allows the SGSN to provide GPRS network access for a mobile station by creating, modifying and deleting tunnels. GTP uses a tunneling mechanism to provide a service for carrying user data packets.

Note When using GTP with failover, if a GTP connection is established and the active unit fails before data is transmitted over the tunnel, the GTP data connection (with a "j" flag set) is not replicated to the standby unit. This occurs because the active unit does not replicate embryonic connections to the standby unit.

The GGSN load balancing feature allows any GSN belonging to a GSN pool to respond to an SGSN request to achieve load balancing on the GGSNs.

GTP Maps and Commands

You can enforce additional inspection parameters on GTP traffic. The **gtp-map** command lets you specify a set of such parameters. When you enable GTP inspection with the **inspect gtp** command, you have the option of specifying a GTP map.

If you do not specify a map with the **inspect gtp** command, the FWSM uses the default GTP map, which is preconfigured with the following default values:

- **request-queue** 200
- **timeout gsn** 0:30:00
- **timeout pdp-context** 0:30:00
- **timeout request** 0:01:00
- **timeout signaling** 0:30:00
- **timeout tunnel** 0:01:00
- **tunnel-limit** 500

Table 21-4 summarizes the commands that you use to configure GTP inspection parameters. These commands are available in GTP map configuration mode. For the detailed syntax of each command, see the applicable command page in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Table 21-4 GTP Map Configuration Commands

Command	Description
description	Specifies the GTP configuration map description.
drop	Specifies the message ID, APN, or GTP version to drop.
mcc	Specifies the three-digit Mobile Country Code (000 - 999). One-digit or two-digit entries will be prefixes with 0s.
message-length	Specifies the message length min and max.
permit errors	Permits packets with errors or different GTP versions.

Table 21-4 GTP Map Configuration Commands

Command	Description
permit response	Specifies an object group allowed to receive responses from another object group.
request-queue	Specifies the maximum requests allowed in the queue.
timeout (gtp-map)	Specifies the idle timeout for the GSN, PDP context, requests, signaling connections, and tunnels.
tunnel-limit	Specifies the maximum number of tunnels allowed.

Enabling and Configuring GTP Inspection

GTP application inspection is disabled by default, so you need to complete the procedures described in this section to enable GTP inspection.

Note GTP inspection requires a special license. If you enter GTP-related commands on a FWSM without the required license, the FWSM displays an error message.

To enable or change GTP configuration, perform the following steps:

Step 1 Define an access list with ACEs that identify the ports required for GTP traffic. The standard ports are UDP ports 2123 and 3386. To create the access list, use the **access-list extended** command once for each ACE, as follows.

```
hostname(config)# access-list acl-name permit {udp | tcp} any any eq port
```

where *acl-name* is the name you assign to the access list and *port* is the GTP port that the ACE identifies.

Step 2 Create a class map or modify an existing class map to identify GTP traffic. Use the **class-map** command to do so, as follows.

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 3 Use a **match access-list** command to identify GTP traffic with the access list you created in Step 1.

```
hostname(config-cmap)# match access-list acl-name
```

Step 4 (Optional) If you want to enforce additional parameters on GTP traffic, create and configure a GTP map. For more information about GTP maps and the default values enforced if you do not specify GTP map, see "GTP Maps and Commands" section on page 21-36. To create and configure a GTP map, perform the following steps.

a. Create a GTP map that will contain the additional parameters of GTP inspection. Use the **gtp-map** command to do so, as follows.

```
hostname(config-cmap)# gtp-map map_name
hostname(config-gtp-map)#
```

where *map_name* is the name of the GTP map. The CLI enters GTP map configuration mode.

b. Configure GTP inspection parameters. To do so, use the GTP map configuration mode commands that you want to enforce. For a list of commands, see Table 21-4.

Step 5 Create a policy map or modify an existing policy map that you want to use to apply the GTP inspection engine to GTP traffic. To do so, use the **policy-map** command, as follows.

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 6 Specify the class map, created in Step 2, that identifies the GTP traffic. Use the **class** command to do so, as follows.

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where *class_map_name* is the name of the class map you created in Step 2. The CLI enters the policy map class configuration mode and the prompt changes accordingly.

Step 7 Enable GTP application inspection. To do so, use the **inspect gtp** command, as follows:

```
hostname(config-pmap-c)# inspect gtp [map_name]
hostname(config-pmap-c)#
```

where *map_name* is the GTP map that you may have created in optional Step 4.

Step 8 Use the **service-policy** command to apply the policy map globally or to a specific interface, as follows:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname(config)#
```

where *policy_map_name* is the policy map you configured in Step 5. If you want to apply the policy map to traffic on all the interfaces, use the **global** option. If you want to apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

The FWSM begins inspecting GTP traffic, as specified.

Example 21-6 shows how to use access lists to identify GTP traffic, define a GTP map, define a policy, and apply the policy to the outside interface.

Example 21-6 Enabling and Configuring GTP Inspection

```
hostname(config)# access-list gtp_acl permit udp any any eq 3386
hostname(config)# access-list gtp_acl permit udp any any eq 2123
hostname(config)# class-map gtp-traffic
hostname(config-cmap)# match access-list gtp_acl
hostname(config-cmap)# gtp-map sample_map
hostname(config-gtp-map)# request-queue 300
hostname(config-gtp-map)# permit mcc 111 mnc 222
hostname(config-gtp-map)# message-length min 20 max 300
hostname(config-gtp-map)# drop message 20
hostname(config-gtp-map)# tunnel-limit 10000
hostname(config)# policy-map sample_policy
hostname(config-pmap)# class gtp-traffic
hostname(config-pmap-c)# inspect gtp sample_map
hostname(config)# service-policy sample_policy outside
```

Verifying and Monitoring GTP Inspection

To display GTP configuration, enter the **show service-policy inspect gtp** command in privileged EXEC mode. For the detailed syntax for this command, see the command page in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Use the **show service-policy inspect gtp statistics** command to show the statistics for GTP inspection. The following is sample output from the **show service-policy inspect gtp statistics** command.

```
hostname# show service-policy inspect gtp statistics
GPRS GTP Statistics:
  version_not_support               0     msg_too_short             0
  unknown_msg                       0     unexpected_sig_msg        0
  unexpected_data_msg               0     ie_duplicated             0
  mandatory_ie_missing              0     mandatory_ie_incorrect    0
  optional_ie_incorrect             0     ie_unknown                0
  ie_out_of_order                   0     ie_unexpected             0
  total_forwarded                   0     total_dropped             0
  signalling_msg_dropped            0     data_msg_dropped          0
  signalling_msg_forwarded          0     data_msg_forwarded        0
  total created_pdp                 0     total deleted_pdp         0
  total created_pdpmcb              0     total deleted_pdpmcb      0
  pdp_non_existent                  0
```

You can use the vertical bar (|) to filter the display. Type **?|** for more display filtering options.

Use the **show service-policy inspect gtp pdp-context** command to display PDP context-related information. The following is sample output from the **show service-policy inspect gtp pdp-context** command.

```
hostname# show service-policy inspect gtp pdp-context detail
1 in use, 1 most used, timeout 0:00:00

Version TID                 MS Addr        SGSN Addr    Idle      APN
v1      1234567890123425        10.0.1.1       10.0.0.2 0:00:13  gprs.example.com

    user_name (IMSI): 214365870921435    MS address:         10.5.1.1
    primary pdp: Y                       nsapi: 2
    sgsn_addr_signal:        10.0.0.2    sgsn_addr_data:          10.0.0.2
    ggsn_addr_signal:        10.1.1.1    ggsn_addr_data:          10.1.1.1
    sgsn control teid:     0x000001d1    sgsn data teid:        0x000001d3
    ggsn control teid:     0x6306ffa0    ggsn data teid:        0x6305f9fc
    seq_tpdu_up:                  0      seq_tpdu_down:                0
    signal_sequence:              0
    upstream_signal_flow:         0      upstream_data_flow:           0
    downstream_signal_flow:       0      downstream_data_flow:         0
    RAupdate_flow:                0
```

The PDP context is identified by the tunnel ID, which is a combination of the values for IMSI and NSAPI. A GTP tunnel is defined by two associated PDP contexts in different GSN nodes and is identified with a Tunnel ID. A GTP tunnel is necessary to forward packets between an external packet data network and a MS user.

You can use the vertical bar (|) to filter the display, as in the following example:

```
hostname# show service-policy gtp statistics | grep gsn
```

GGSN Load Balancing

GGSN load balancing (GSN pooling) allows any GSN the belongs to a GSN pool to respond to an SGSN request to achieve load balancing on the GGSN. To enable support for GNS pooling, use the **permit response** command.

If the security appliance performs GTP inspection, by default the security appliance drops GTP responses from GSNs that were not specified in the GTP request. This situation occurs when you use load balancing among a pool of GSNs to provide efficiency and scalability of GPRS.

You can enable support for GSN pooling by using the **permit response** command. This command configures the security appliance to allow responses from any of a designated set of GSNs, regardless of the GSN to which a GTP request was sent. You identify the pool of load-balancing GSNs as a network object. Likewise, you identify the SGSN as a network object. If the GSN responding belongs to the same object group as the GSN that the GTP request was sent to, and if the SGSN is in an object group that the responding GSN is permitted to send a GTP response to, the security appliance permits the response.

To create an object to represent the pool of load-balancing GSNs, perform the following steps:

Step 1 Define a new network object group representing the pool of load-balancing GSNs. To do so, use the **object-group** command.

```
hostname(config)# object-group network GSN-pool-name
hostname(config)#
```

where *GSN-pool-name* is the object group name for GGSNs.

Step 2 Specify the load-balancing GSNs using the **network-objec**t command. You can configure one **network-object** command per GSN using the **host** keyword. You can also specify a network containing GSNs that perform load balancing.

```
hostname(config)# network-object host IP-address
hostname(config)#
```

where *IP-address* is the IP address of the host.

Step 3 Create an object to represent the SGSN that the load-balancing GSNs are permitted to respond to. To do so, use the **object-group** command.

a. Define an SGSN network object group that sends GTP requests to the GSN pool. To do so, use the **object-group** command.

```
hostname(config)# object-group network SGSN-name
hostname(config)#
```

where *SGSN-name* is the SGSN network object group name.

b. Identify the SGSN. To do so, use the **network-object** command:

```
hostname(config)# network-object host IP-address
hostname(config)#
```

where *IP-address* is the SGSN.

Step 4 Allow GTP responses, from any GSN in the network object representing the GSN pool, to the network object representing the SGSN. To do so, use the **gtp-map** and **permit responses** commands.

```
hostname(config)# gtp-map map_name
hostname(config-gtp-map)# permit response to-object-group SGSN-name from-object-group
GSN-pool-name
```

where *map-name* is the name of the gtp map, *SGSN-name* is the name of the SGSN created in Step 3, and *GSN-pool-name* is the name of the GSN pool created in Step 1.

The following example shows how to support GSN pooling by defining network objects for the GSN pool and the SGSN. An entire Class C network is defined as the GSN pool in this case, but you can identify multiple individual IP addresses as well. The GTP map is configured to permit responses from the GSN pool to the SGSN.

Example 21-7 Enabling and Configuring GGSN Load Balancing

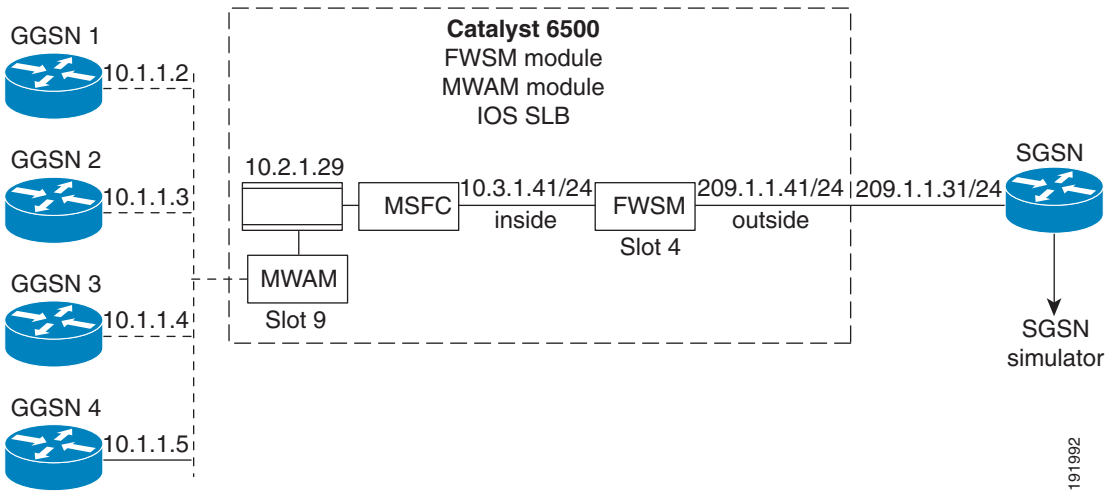
```
hostname(config)# object-group network GGSNS
hostname(config-network)# network-object 10.1.1.0 255.255.255.0
hostname(config)# object-group network SGSNS
hostname(config-network)# network-object hjost 192.168.1.1
hostname(config)# gtp-map GTPMAP
hostname(config-gtp-map)# permit response to-object-group SGSNS from-object-group GGSNS
```

For additional GGSN load balancing information and configurations, refer to the *Cisco GGSN Release 7.0 Configuration Guide* and the *Cisco MultiProcessor WAN Module User Guide*.

GTP Sample Configuration

Figure 21-7 shows a sample GTP inspection configuration.

Figure 21-7 GTP Inspection Setup



Sample configuration of SLB (IOS SLB, MSFC used), GGSN (MWAM module used) and FWSM. SLB and MWAM configuration on supervisor/MSFC.

The MWAM is a Cisco IOS application module that you can install in the Cisco Catalyst 6500 Series switch. Each MWAM contains three processor complexes, with two CPUs each and Each CPU can be used to run an independent IOS image. Several Cisco mobile wireless applications are supported. This sample configuration runs Cisco Gateway GPRS Support (General Packet Radio Service Packet Gateway application).

See the following documentation for installation and configuration of the MWAM module on the Cisco Catalyst 6500 Series switch.

http://www.cisco.com/en/US/docs/wireless/mwam/user/guide/install.html

The following configuration explains the config requirements for MWAM module as well as how to configure a gateway GPRS support node (GGSN) to support load balancing functions using the Cisco IOS software Server Load Balancing (SLB) feature.

See the following documentation for configuration of load balancing on GGSNs:

http://www.cisco.com/en/US/docs/ios/12_4/12_4y/12_4_22ye/ggsn9_0/cfg/ggsnslb.html

As per Figure 21-6, two GGSNs (GGSN1 and GGSN2) are configured on the MWAM module:

```
firewall multiple-vlan-interfaces
firewall module 4 vlan-group 1
firewall module 10 vlan-group 1
firewall vlan-group 1  3-40,44,84,115,119,172,200-202,400-500,800-900
mwam module 9 port 1 allowed-vlan 1,3-40,44,84,172,200-202,400-500,800-900
mwam module 9 port 2 allowed-vlan 1,3-40,44,84,172,200-202,400-500,800-900
mwam module 9 port 3 allowed-vlan 1,3-40,44,84,172,200-202,400-500,800-900
ip subnet-zero
!
no ip domain-lookup
ip slb timers gtp gsn 40000
ip slb probe PING ping
!
ip slb serverfarm GGSN-POOL
 nat server
 probe PING
 !
 real 10.4.1.32
  weight 1
  inservice
 !
 real 10.4.1.33
  weight 1
  inservice
!
ip slb serverfarm TGGSN-POOL
 nat server
 probe PING
 !
 real 10.4.1.34
  faildetect numconns 2
  inservice
 !
 real 10.4.1.35
  faildetect numconns 2
  inservice
!
ip slb vserver GTP-V0
 virtual 10.2.1.29 udp 3386 service gtp
 serverfarm GGSN-POOL
 inservice
!
ip slb vserver GTP-V1
 virtual 10.2.1.29 udp 2123 service gtp
 serverfarm GGSN-POOL
 inservice
!
ip slb vserver TGTP-V0
 virtual 10.2.1.28 udp 3386 service gtp
 serverfarm TGGSN-POOL
```

```
 inservice
!
ip slb vserver TGTP-V1
 virtual 10.2.1.28 udp 2123 service gtp
 serverfarm TGGSN-POOL
 inservice
!
ip slb dfp password 7 13061E010803
 agent 10.1.1.2 1111 30 0 10
 agent 10.1.1.3 1111 30 0 10
 agent 10.1.1.4 1111 30 0 10
 agent 10.1.1.5 1111 30 0 10
!
```

GGSN1 is configured as follows:

```
hostname# show running config
Building configuration...

Current configuration : 1460 bytes
!
! Last configuration change at 21:33:19 UTC Wed Dec 13 2006
! NVRAM config last updated at 01:54:40 UTC Sat Nov 18 2006
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service gprs ggsn
!
hostname GGSN2ADCTX
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
ip dfp agent gprs
 port 1111
 password cisco
 inservice
!
ip cef
no ip domain lookup
ip domain name cisco.com
no ip dhcp use vrf connected
!
interface GigabitEthernet0/0
 no ip address
!
interface GigabitEthernet0/0.1
!
interface GigabitEthernet0/0.8
 encapsulation dot1Q 8
 ip address 10.1.1.2 255.255.255.0
 no snmp trap link-status
!
interface Virtual-Template1
 ip address 10.4.1.32 255.255.255.0
 encapsulation gtp
```

```
 gprs access-point-list gtp-test
!
ip local pool localpool1 10.1.1.101 10.1.1.110
ip local pool localpool11 10.7.3.3 10.7.3.255
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.1
!
no ip http server
!
gprs access-point-list gtp-test
  access-point 1
   access-point-name sj-gtp.cisco.com
   ip-address-pool local localpool11
   !
control-plane
!
line con 0
line vty 0
 no login
line vty 1 4
 login
line vty 5 15
 login
!
end
```

GGSN3 is configured as follows:

```
hostname# show running-config
Building configuration...

Current configuration : 1533 bytes
!
! Last configuration change at 21:33:47 UTC Wed Dec 13 2006
! NVRAM config last updated at 01:56:07 UTC Sat Nov 18 2006
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service gprs ggsn
!
hostname GGSN3ADCTX
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
ip subnet-zero
ip dfp agent gprs
 port 1111
 password cisco
 inservice
!
ip cef
no ip domain lookup
ip domain name cisco.com
no ip dhcp use vrf connected
!
```

```
interface GigabitEthernet0/0
 no ip address
!
interface GigabitEthernet0/0.3
!
interface GigabitEthernet0/0.8
 encapsulation dot1Q 8
 ip address 10.4.1.3 255.255.255.0
 no snmp trap link-status
!
interface Virtual-Template1
 ip address 10.1.1.33 255.255.255.0
 encapsulation gtp
 gprs access-point-list gtp-test
!
ip local pool localpool2 10.1.1.111 10.1.1.120
ip local pool localpool22 10.8.4.4 10.8.4.254
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.1
!
no ip http server
!
gprs maximum-pdp-context-allowed 50000
gprs qos default-response requested
gprs access-point-list gtp-test
  access-point 1
   access-point-name sj-gtp.cisco.com
   ip-address-pool local localpool22
   !
control-plane
!
line con 0
line vty 0
 no login
line vty 1 4
 login
line vty 5 15
 login
!
!
end
```

The FWSM configuration for load balancing the GGSNs is as follows:

```
FWSM Version 3.2(0)45 <context>
!
hostname admin
domain-name cisco.com
enable password 9jNfZuG3TC5tCVH0 encrypted
no names
!
interface Vlan172
 nameif mgmt
 security-level 100
 ip address 172.21.64.35 255.255.255.128 standby 172.21.64.36
!
interface Vlan5
 nameif inside
 security-level 100
 ip address 10.2.1.41 255.255.255.0 standby 10.2.1.40
!
interface Vlan9
 nameif outside
 security-level 0
```

```
 ip address 209.165.201.41 255.255.255.0 standby 209.165.201.40
!
passwd 2KFQnbNIdI.2KYOU encrypted
same-security-traffic permit inter-interface
object-group network GGSNS ================================configured object group to
define GGSNs
 network-object host 10.4.1.32
 network-object host 10.4.1.33
object-group network SGSNS ================================configured object group to
define SGSNs
 network-object host 10.5.1.1
object-group network servers
 network-object 10.2.1.0 255.255.255.0
 network-object host 10.6.1.25
 network-object host 10.6.1.26
 network-object host 10.6.1.27
 network-object host 10.4.1.32
 network-object host 10.4.1.33
object-group network clients
 network-object 10.6.1.0 255.255.255.0
 network-object host 10.5.1.1
access-list gtpacl extended permit udp any any eq 2123
access-list gtpacl extended permit udp any any eq 3386
access-list gtpacl extended permit icmp any any
access-list gtpacl extended permit udp any any
access-list gtpacl extended permit tcp any any eq www
access-list gtpacl extended permit tcp any any eq ftp
access-list gtpacl extended permit tcp any any eq telnet
access-list gtpacl extended permit tcp any any eq ssh
access-list 112 extended permit tcp object-group servers object-group clients eq www
access-list 112 extended permit tcp object-group servers object-group clients eq https
access-list 112 extended permit tcp object-group servers object-group clients eq ftp
access-list 112 extended permit tcp object-group servers object-group clients eq telnet
access-list 112 extended permit udp object-group servers object-group clients eq 3386
access-list 112 extended permit udp object-group servers object-group clients eq 2123
access-list 112 extended permit tcp object-group servers object-group clients eq ssh
!
gtp-map GTPMAP ============================================================configured GTP
map to include the permit response cli
 permit response to-object-group SGSNS from-object-group GGSNS
 permit errors
!
pager lines 24
logging enable
logging timestamp
logging buffered debugging
mtu mgmt 1500
mtu inside 1500
mtu outside 1500
monitor-interface inside
monitor-interface outside
icmp permit any mgmt
icmp permit any inside
icmp permit any outside
asdm history enable
arp timeout 14400
nat-control
no xlate-bypass
static (outside,inside) 10.5.1.1 10.5.1.1 netmask 255.255.255.255
static (inside,outside) 10.4.1.31 10.4.1.31 netmask 255.255.255.255
static (inside,outside) 10.4.1.32 10.4.1.32 netmask 255.255.255.255
static (inside,outside) 10.4.1.33 10.4.1.33 netmask 255.255.255.255
access-group OO in interface mgmt
access-group 111 in interface outside per-user-override
```

```
route inside 10.4.1.32 255.255.255.255 10.1.1.2 1
route inside 10.4.1.33 255.255.255.255 10.1.1.3 1
route outside 10.5.1.1 255.255.255.255 209.165.201.31 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00
timeout mgcp-pat 0:05:00 sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect
0:02:00
timeout uauth 0:05:00 absolute
username cisco password 3USUcOPFUiMCO4Jk encrypted
no snmp-server location
no snmp-server contact
telnet timeout 5
ssh 171.69.42.198 255.255.255.255 mgmt
ssh timeout 5
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map global_policy
 class inspection_default
  inspect dns maximum-length 512
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect http
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect smtp
  inspect sunrpc
  inspect tftp
  inspect xdmcp
  inspect icmp
  inspect gtp GTPMAP ==========================================attached the GTP map to gtp
inspection in service policy
  !
service-policy global_policy global
Cryptochecksum:3b1c3373e908cb9163d9aa1387478fa4
: end
```

H.323 Inspection

This section describes how to enable H.323 application inspection and change the default port configuration. This section includes the following topics:

H.323 Inspection Overview

H.323 inspection provides support for H.323 compliant applications such as Cisco CallManager and VocalTec Gatekeeper. H.323 is a suite of protocols defined by the International Telecommunication Union for multimedia conferences over LANs. The FWSM supports H.323 through Version 4, including the H.323 v3 feature Multiple Calls on One Call Signaling Channel. The H.323 Gatekeeper Update Protocol inspection is also supported. Because GUP is a Cisco proprietary protocol, H.323-GUP inspection is relevant only in topologies where the Cisco Gatekeeper devices are employed.

With H.323 inspection enabled, the FWSM supports multiple calls on the same call signaling channel, a feature introduced with H.323 Version 3. This feature reduces call setup time and reduces the use of ports on the FWSM.

The two major functions of H.323 inspection are as follows:

- NAT the necessary embedded IPv4 addresses in the H.225 and H.245 messages. Because H.323 messages are encoded in PER encoding format, the FWSM uses an ASN.1 decoder to decode the H.323 messages.
- Dynamically allocate the negotiated H.245 and RTP/RTCP connections.

How H.323 Works

The H.323 protocols collectively may use up to two TCP connection and four to six UDP connections. FastConnect uses only one TCP connection. RAS uses a single UDP connection for registration, admissions, and status.

An H.323 client may initially establish a TCP connection to an H.323 server using TCP port 1720 to request Q.931 call setup. As part of the call setup process, the H.323 terminal supplies a port number to the client to use for an H.245 TCP connection. In environments where H.323 gatekeeper is in use, the initial packet is transmitted using UDP.

H.323 inspection monitors the Q.931 TCP connection to determine the H.245 port number. If the H.323 terminals are not using FastConnect, the FWSM dynamically allocates the H.245 connection based on the inspection of the H.225 messages.

Within each H.245 message, the H.323 endpoints exchange port numbers that are used for subsequent UDP data streams. H.323 inspection inspects the H.245 messages to identify these ports and dynamically creates connections for the media exchange. RTP uses the negotiated port number, while RTCP uses the next higher port number.

The H.323 control channel handles H.225 and H.245 and H.323 RAS. H.323 inspection uses the following ports.

- UDP port 1718—Gate Keeper Discovery
- UDP port 1719—RAS
- TCP port 1720—Control Port

You must permit traffic for the well-known H.323 port 1720 for the H.225 call signaling; however, the H.245 signaling ports are negotiated between the endpoints in the H.225 signaling. When an H.323 gatekeeper is used, the FWSM opens an H.225 connection based on inspection of the ACF message.

After inspecting the H.225 messages, the FWSM opens the H.245 channel and then inspects traffic sent over the H.245 channel as well. All H.245 messages passing through the FWSM undergo H.245 application inspection, which NATs embedded IP addresses and opens the media channels negotiated in H.245 messages.

The H.323 ITU standard requires that a TPKT header, defining the length of the message, precede the H.225 and H.245, before being passed on to the reliable connection. Because the TPKT header does not necessarily need to be sent in the same TCP packet as H.225 and H.245 messages, the FWSM must remember the TPKT length to process and decode the messages properly. For each connection, the FWSM keeps a record that contains the TPKT length for the next expected message.

If the FWSM needs to perform NAT on IP addresses in messages, it changes the checksum, the UUIE length, and the TPKT, if it is included in the TCP packet with the H.225 message. If the TPKT is sent in a separate TCP packet, the FWSM proxy ACKs that TPKT and appends a new TPKT to the H.245 message with the new length.

Note The FWSM does not support TCP options in the Proxy ACK for the TPKT.

Each UDP connection with a packet going through H.323 inspection is marked as an H.323 connection and times out with the H.323 timeout as configured with the **timeout** command.

Limitations and Restrictions

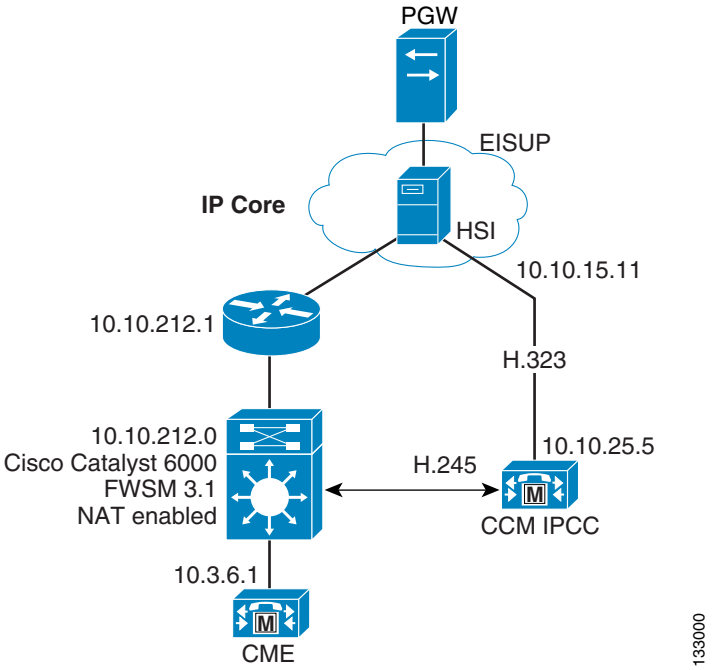
Some of the known issues and limitations of H.323 application inspection are as follows:

- Static PAT may not properly translate IP addresses embedded in optional fields within H.323 messages. If you experience this kind of problem, do not use static PAT with H.323.
- When a NetMeeting client registers with an H.323 gatekeeper and tries to call an H.323 gateway that is also registered with the H.323 gatekeeper, the connection is established but no voice is heard in either direction. This problem is unrelated to the FWSM.
- If you configure a network static address where the network static address is the same as a third-party netmask and address, then any outbound H.323 connection fails.
- Dynamic NAT (PAT) is not supported for H.323-GUP inspection.

Topologies Requiring H.225 Configuration

Some additional H.225 configuration may be required in a topology where call control happens between H.323 endpoints connecting through an FWSM (see Figure 21-8).

Figure 21-8 Topology Requiring H.225 Configuration



In this topology, call signaling occurs between the Cisco CallManager and the HSI on one side of the FWSM and between the HSI and the Cisco CallManager endpoint on the other side. Afterwards, call control happens directly between the Cisco CallManager and the Cisco CallManager endpoint. When the HSI and one endpoint is on a network protected by the FWSM and the other endpoint is on another network, the call control may not go through without additional H.225 configuration.

The FWSM is not aware of the existence of the Cisco CallManager in this topology. With only the packet flows that happen through the security appliance, the FWSM cannot open a proper pinhole to allow such a call to be successful. For this reason, some additional H.225 configuration is required in this scenario.

To provide the necessary configuration, you identify an HSI and its associated endpoints within an HSI group. After this configuration is completed, when the FWSM sees the HSI as one of the communicating hosts in an H.225 connection, it opens H.245 holes between the endpoints in the HSI group. The actual H.245 connection will match one of these pinholes and will go through properly.

H.225 Map Commands

The H.225 map allows the FWSM to open dynamic, port-specific pinholes for an H.245 connection when an HSI is involved in H.225 call-signalling. The H.225 map provides information about the HSI and its associated endpoints, which is required to establish this connection without compromising the security of the network protected by the FWSM.

The **h225-map** command lets you create an H.225 map. One H225 map can contain a maximum of five HSI groups. Table 21-5 lists the commands available in H.225 map configuration mode.

Table 21-5 H.225 Configuration Commands

Command	Configuration mode	Description
hsi-group	H.225 map configuration mode	Defines an HSI group and enables HSI group configuration mode. Each HSI group can contain a maximum of ten endpoints.
hsi	HSI group configuration mode	Identifies the HSI.
endpoint	HSI group configuration mode	Identifies one or more endpoints within the HSI group.

Enabling and Configuring H.323 Inspection

H.323 inspection is enabled by default.

To enable H.323 inspection, including the optional use of an H.225 map, perform the following steps:

Step 1 To define an access list with ACEs that identify the ports required for H.323 traffic, enter the following command for each ACE:

```
hostname(config)# access-list acl-name permit {udp | tcp} any any eq port
```

where *acl-name* is the name you assign to the access list and *port* is the H.323 port that the ACE identifies.

The standard ports are UDP ports 1718 and 1719 and TCP port 1720.

Step 2 Create a class map or modify an existing class map to identify H.323 traffic. Use the **class-map** command to do so, as follows.

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 3 Use a **match access-list** command to identify H.323 traffic with the access list you created in Step 1.

```
hostname(config-cmap)# match access-list acl-name
```

Step 4 (Optional) If required by your network topology, configure an H.225 map. For more information about whether your network requires an H.225 map, see the "Topologies Requiring H.225 Configuration" section on page 21-50. To create and configure an H.225 map, perform the following steps.

a. Create an H.225 map.

```
hostname(config)# h225-map map_name
hostname(config-h225-map)#
```

The system enters H.225 map configuration mode and the CLI prompt changes accordingly.

b. Identify an HSI group. To do so, use the **hsi-group** command, as follows.

```
hostname(config-h225-map)# hsi-group group_ID
hostname(config-h225-map-hsi-grp)#
```

where *group_ID* is a number, from 0 to 2147483647, that identifies the HSI group.

Note The maximum number of HSI groups allowed per H.225 map is five.

The system enters HSI group configuration mode and the CLI prompt changes accordingly.

c. Define an HSI for the group.

```
hostname(config-h225-map-hsi-grp)# hsi ip_address
```

where *ip_address* is the addresses of the HSI.

d. Define up to ten endpoints. To do so, use the **endpoint** command once per endpoint, as follows.

```
hostname(config-h225-map-hsi-grp)# endpoint ip_address interface
```

where *interface* with the interface on the FWSM that is connected to the endpoint and *ip_address* is the addresses of the endpoint.

e. If you need to create additional HSI groups, repeat step b. through d.

Step 5 Create a policy map or modify an existing policy map that you want to use to apply the H.323 inspection engine to H.323 traffic. To do so, use the **policy-map** command, as follows.

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 6 Specify the class map, created in Step 2, that identifies the H.323 traffic. Use the **class** command to do so, as follows.

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where *class_map_name* is the name of the class map you created in Step 2. The CLI enters the policy map class configuration mode and the prompt changes accordingly.

Step 7 Enable H.323 application inspection. To do so, use the **inspect h323** command, as follows.

```
hostname(config-pmap-c)# inspect h323 [h225 map_name]
hostname(config-pmap-c)#
```

where *map_name* is the H.225 map that you may have created in optional Step 4.

Step 8 Use the **service-policy** command to apply the policy map globally or to a specific interface, as follows:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname(config)#
```

where *policy_map_name* is the policy map you configured in Step 5. If you want to apply the policy map to traffic on all the interfaces, use the **global** option. If you want to apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

The FWSM begins inspecting H.323 traffic, as specified.

Example 21-8 Configuring H.323 Inspection without an H.225 Map

You enable the H.323 inspection engine as shown in the following example, which creates a class map to match H.323 traffic on the default port (1720). The service policy is then applied to the outside interface.

```
hostname(config)# access-list h323_acl permit udp any any eq 1718
hostname(config)# access-list h323_acl permit udp any any eq 1719
hostname(config)# access-list h323_acl permit tcp any any eq 1720
hostname(config)# class-map h323-traffic
hostname(config-cmap)# match access-list h323_acl
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class h323_port
hostname(config-pmap-c)# inspect h323 ras
hostname(config-pmap-c)# inspect h323 h225
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

Example 21-9 includes an H.225 map with two HSI groups, as part of the overall H.323 configuration.

Example 21-9 Configuring H.323 Inspection with an H.225 Map

```
hostname(config)# access-list h323_acl permit udp any any eq 1718
hostname(config)# access-list h323_acl permit udp any any eq 1719
hostname(config)# access-list h323_acl permit tcp any any eq 1720
hostname(config)# class-map h323-traffic
hostname(config-cmap)# match access-list h323_acl
hostname(config-cmap)# h225-map sample_map
hostname(config-h225-map)# hsi-group 1
hostname(config-h225-map-hsi-grp)# hsi 10.10.15.11
hostname(config-h225-map-hsi-grp)# endpoint 10.3.6.1 inside
hostname(config-h225-map-hsi-grp)# endpoint 10.10.25.5 outside
hostname(config-h225-map-hsi-grp)# policy-map sample_policy
hostname(config-pmap)# class h323_port
hostname(config-pmap-c)# inspect h323 ras
hostname(config-pmap-c)# inspect h323 h225 sample_map
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

Configuring H.323 and H.225 Timeout Values

To configure the idle time after which an H.225 signalling connection is closed, use the **timeout h225** command. The default for H.225 timeout is one hour.

To configure the idle time after which an H.323 control connection is closed, use the **timeout h323** command. The default is five minutes.

Verifying and Monitoring H.323 Inspection

This section describes how to display information about H.323 sessions. This section includes the following topics:

- Monitoring H.225 Sessions, page 21-54
- Monitoring H.245 Sessions, page 21-54
- Monitoring H.323 RAS Sessions, page 21-55

Monitoring H.225 Sessions

The **show h225** command displays information for H.225 sessions established across the FWSM. Along with the **debug h323 h225 event**, **debug h323 h245 event**, and **show local-host** commands, this command is used for troubleshooting H.323 inspection engine issues.

Before entering the **show h225**, **show h245**, or **show h323-ras** commands, we recommend that you configure the **pager** command. If there are a lot of session records and the **pager** command is not configured, it may take a while for the **show** command output to reach its end. If there is an abnormally large number of connections, check that the sessions are timing out based on the default timeout values or the values set by you. If they are not, then there is a problem that needs to be investigated.

The following is sample output from the **show h225** command:

```
hostname# show h225
Total H.323 Calls: 1
1 Concurrent Call(s) for
    Local:   10.130.56.3/1040   Foreign: 172.30.254.203/1720
    1. CRV 9861
    Local:   10.130.56.3/1040   Foreign: 172.30.254.203/1720
0 Concurrent Call(s) for
    Local:   10.130.56.4/1050   Foreign: 172.30.254.205/1720
```

This output indicates that there is currently 1 active H.323 call going through the FWSM between the local endpoint 10.130.56.3 and foreign host 172.30.254.203, and for these particular endpoints, there is 1 concurrent call between them, with a CRV for that call of 9861.

For the local endpoint 10.130.56.4 and foreign host 172.30.254.205, there are 0 concurrent calls. This means that there is no active call between the endpoints even though the H.225 session still exists. This could happen if, at the time of the **show h225** command, the call has already ended but the H.225 session has not yet been deleted. Alternately, it could mean that the two endpoints still have a TCP connection opened between them because they set “maintainConnection” to TRUE, so the session is kept open until they set it to FALSE again, or until the session times out based on the H.225 timeout value in your configuration.

Monitoring H.245 Sessions

The **show h245** command displays information for H.245 sessions established across the FWSM by endpoints using slow start. Slow start is when the two endpoints of a call open another TCP control channel for H.245. Fast start is where the H.245 messages are exchanged as part of the H.225 messages on the H.225 control channel.) Along with the **debug h323 h245 event**, **debug h323 h225 event**, and **show local-host** commands, this command is used for troubleshooting H.323 inspection engine issues.

The following is sample output from the **show h245** command:

```
hostname# show h245
Total: 1
        LOCAL           TPKT    FOREIGN         TPKT
1       10.130.56.3/1041        0       172.30.254.203/1245    0
        MEDIA: LCN 258 Foreign 172.30.254.203 RTP 49608 RTCP 49609
                       Local   10.130.56.3 RTP 49608 RTCP 49609
        MEDIA: LCN 259 Foreign 172.30.254.203 RTP 49606 RTCP 49607
                       Local   10.130.56.3 RTP 49606 RTCP 49607
```

There is currently one H.245 control session active across the FWSM. The local endpoint is 10.130.56.3, and we are expecting the next packet from this endpoint to have a TPKT header because the TPKT value is 0. The TKTP header is a 4-byte header preceding each H.225/H.245 message. It gives the length of the message, including the 4-byte header. The foreign host endpoint is 172.30.254.203, and we are expecting the next packet from this endpoint to have a TPKT header because the TPKT value is 0.

The media negotiated between these endpoints have an LCN of 258 with the foreign RTP IP address/port pair of 172.30.254.203/49608 and an RTCP IP address/port of 172.30.254.203/49609 with a local RTP IP address/port pair of 10.130.56.3/49608 and an RTCP port of 49609.

The second LCN of 259 has a foreign RTP IP address/port pair of 172.30.254.203/49606 and an RTCP IP address/port pair of 172.30.254.203/49607 with a local RTP IP address/port pair of 10.130.56.3/49606 and RTCP port of 49607.

Monitoring H.323 RAS Sessions

The **show h323-ras** command displays information for H.323 RAS sessions established across the FWSM between a gatekeeper and its H.323 endpoint. Along with the **debug h323 ras event** and **show local-host** commands, this command is used for troubleshooting H.323 RAS inspection engine issues.

The **show h323-ras** command displays connection information for troubleshooting H.323 inspection engine issues. The following is sample output from the **show h323-ras** command.

```
hostname# show h323-ras
Total: 1
        GK                      Caller
        172.30.254.214 10.130.56.14
```

This output shows that there is one active registration between the gatekeeper 172.30.254.214 and its client 10.130.56.14.

H.323 GUP Support

The H.323-GUP feature is used for creation of the secondary channel for the H.323-GUP connection from the H.323-RAS connection, and for translation (NAT) of the embedded addresses in the GUP messages. It enables Gatekeepers to communicate with each other through the firewall.

You do not need to enable H.323-GUP explicitly. To utilize this feature, enable H.323-RAS inspection with the appropriate access list (allowing UDP port 1719).

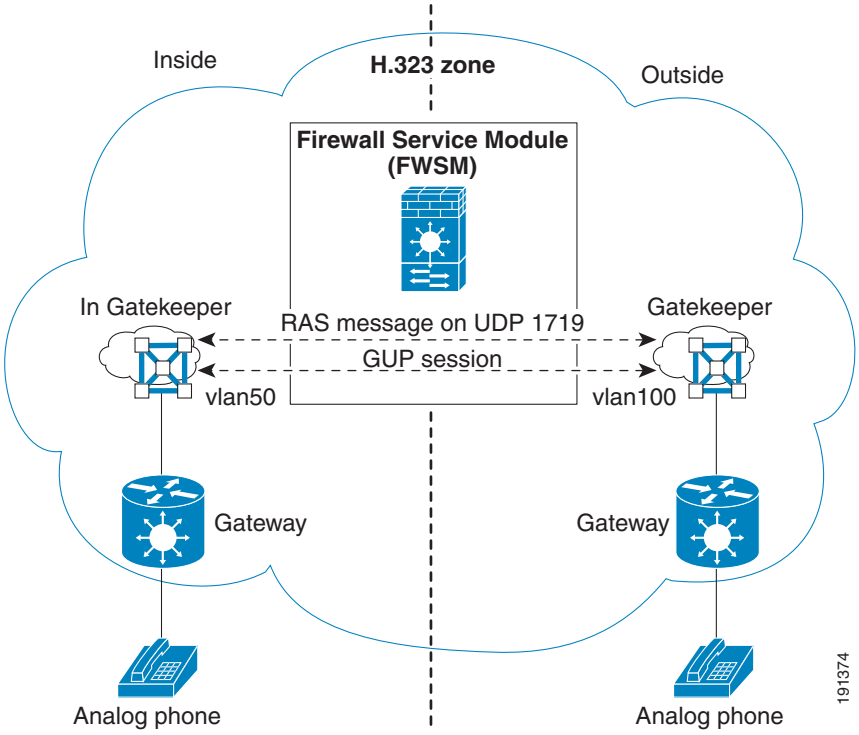
Limitations:

- H.323-GUP inspection is relevant only in topologies where the Cisco Gatekeeper devices are employed because GUP is a Cisco proprietary protocol.
- Dynamic NAT and dynamic PAT are not supported in H.323 GUP inspection.

H.323 GUP Configuration

Figure 21-9 illustrates an H.323 inspection topology configured with H.323 GUP support.

Figure 21-9 H.323 Inspection Configuration with H.323 GUP Support



The following configuration applies to Figure 21-9.

```
firewall transparent
hostname FWSM
!
interface Vlan50
nameif inside
bridge-group 1
security-level 100
!
interface Vlan100
nameif outside
bridge-group 1
security-level 0
!
interface BVI1
ip address 10.0.0.8 255.255.255.0
!
access-list h323-gup-permit extended permit udp any any eq 1719
access-group h323-gup-permit in interface inside
access-group h323-gup-permit in interface outside
```

Note RAS inspection should be turned on for interfaces through which the gatekeeper running GUP protocol is reachable. In this example, RAS inspection is turned on for both inside and outside interfaces.

Outside gatekeeper configuration (GK):

```
gatekeeper
zone local GK cisco.com 10.0.0.6
zone cluster local gup-cluster GK
element inGK 10.0.0.7 1719
```

Inside gatekeeper configuration (inGK):

```
gatekeeper
zone local inGK cisco.com 10.0.0.7
zone cluster local gup-cluster inGK
element GK 10.0.0.6 1719
```

When the H.323 GUP session is established in this configuration, the following is output from the **show h323 gup** command:

```
hostname(config)# show h323 gup

No.     LOCAL                       FOREIGN
1       inside:10.0.0.7/15970       Outside:209.165.201.6/22754
```

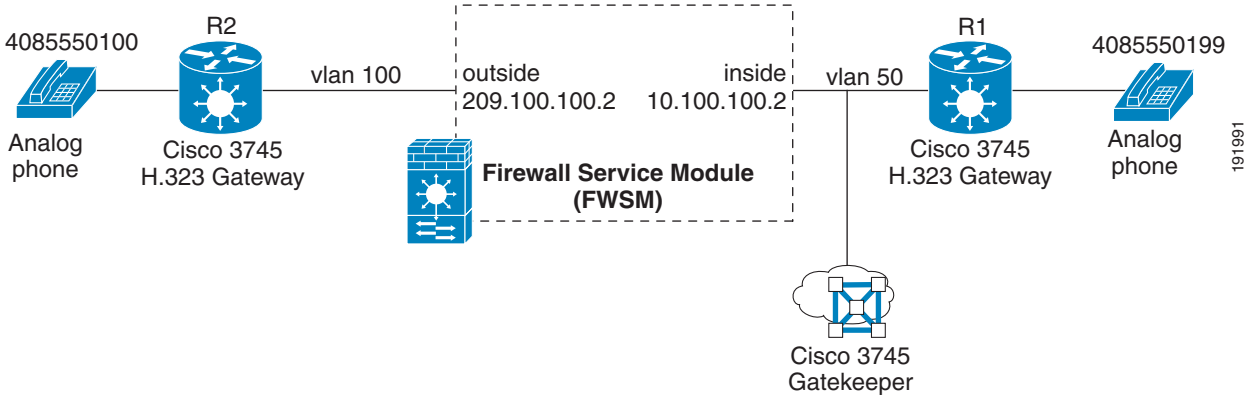
The following output from the **show conn** command shows the secondary channel established between the H.323 Gatekeepers and the H.323 GUP connections marked with the flag n.

```
hostname(config)# show conn

3 in use, 37 most used
Network Processor 1 connection
UDP out 209.165.201.6:1719 in 10.0.0.7:1719 idle 0:00:45 Bytes 672
FLAGS - H
TCP out 209.165.201.6:22754 in 10.0.0.7:15970 idle 0:00:04 Bytes 1188 FLAGS - UBIn
Network Processor 2 connections

Multicast sessions:
Network Processor 1 connection
Network Processor 2 connections
IPv6 connections:
```

H.323 Sample Configuration

Figure 21-10 shows a sample configuration for H.323 inspection.

Figure 21-10 H.323 Inspection Setup



Configuration of the IOS H.323 Gateway (Router R2) on the outside interface:

```
hostname(config)#hostname R2
hostname(config)#interface FastEthernet0/1
hostname(config-if)# ip address 209.165.201.1 255.0.0.0
hostname(config-if)#no shut
hostname(config-if)#exit
hostname(config)#ip route 10.0.0.0 255.0.0.0 209.165.201.2
hostname(config)#voice-port 1/0/0
hostname(config-voiceport)#no shut
hostname(config-voiceport)#exit
hostname(config)#gateway
hostname(config-gateway)#
hostname(config-gateway)#exit
hostname(config)#ip cef
hostname(config)#int f0/1
hostname(config-if)#h323-gateway voip interface
hostname(config-if)#h323-gateway voip id inGK ipaddr 10.0.0.6
hostname(config-if)#h323-gateway voip h323-id R2
```

Configure dial peer to forward voice calls to destination number 408555010991 using the H.323 protocol:

```
hostname(config)#dial-peer voice 101 voip
hostname(config-dial-peer)#destination-pattern 40855501099
hostname(config-dial-peer)#session target ras
hostname(config-dial-peer)#exit
```

Configure dial peer to forward voice calls to 4085550100 to voice port 1/0/0 in router R2:

```
hostname(config)#dial-peer voice 102 pots
hostname(config-dial-peer)#destination-pattern 4085550100
hostname(config-dial-peer)#port 1/0/0
hostname(config-dial-peer)#exit
hostname(config)#exit
```

Configuration of the IOS H.323 gateway (router R1) on the inside interface:

```
hostname(config)#hostname R1
hostname(config)#interface FastEthernet0/1
hostname(config-if)# ip address 10.100.100.1 255.0.0.0
hostname(config-if)#no shut
hostname(config-if)#ip route 209.165.201.0 255.0.0.0 10.100.100.2
hostname(config)#
hostname(config)#voice-port 3/0/0
hostname(config-voiceport)#no shut
hostname(config-voiceport)#exit
hostname(config)#gateway
hostname(config-gateway)#exit
hostname(config)#ip cef
hostname(config)#int fastethernet0/1
hostname(config-if)#h323-gateway voip interface
hostname(config-if)#h323-gateway voip id inGK ipaddr 10.0.0.6
hostname(config-if)#h323-gateway voip h323-id R1
hostname(config-if)#exit
```

Forward all voice calls destined to 4085550100 using the H.323 protocol:

```
hostname(config)#dial-peer voice 101 voip
hostname(config-dial-peer)#destination-pattern 4085550100
hostname(config-dial-peer)#session target ras
```

Forward all voice calls destined to 4085550199 to voice port 3/0/0:

```
hostname(config)#dial-peer voice 102 pots
```

```
hostname(config-dial-peer)#destination-pattern 4085550199
hostname(config-dial-peer)#port 3/0/0
hostname(config-dial-peer)#^Z
```

Configuration of the IOS H.323 Gatekeeper (router inGK) on the inside interface:

```
hostname(config)#hostname inGK
hostname(config)#interface FastEthernet0/1
hostname(config-if)# ip address 10.0.0.6 255.0.0.0
hostname(config-if)#no shut
hostname(config-if)#exit
hostname(config)#gatekeeper
hostname(config-gk)#zone local inGK cisco.com 10.0.0.6
hostname(config-gk)#no shut
hostname(config)#
hostname(config)#ip route 209.165.201.0 255.0.0.0 10.100.100.2
```

Configuration of the FWSM for H.323 inspection:

```
hostname# config t
hostname(config)# interface Vlan100
hostname(config-if)#  nameif outside
hostname(config-if)#  security-level 0
hostname(config-if)#  ip address 209.165.201.2 255.0.0.0
hostname(config-if)#
hostname(config-if)# interface Vlan50
hostname(config-if)#  nameif inside
hostname(config-if)#  security-level 100
hostname(config-if)#  ip address 10.100.100.2 255.0.0.0
hostname(config-if)#
hostname(config-if)# access-list voice extended permit udp any any eq 1719
hostname(config)# access-list voice extended permit tcp any any eq h323
hostname(config)#
hostname(config)# access-group voice in interface outside
hostname(config)# access-group voice in interface inside
hostname(config)#
hostname(config)# policy-map global_policy
hostname(config-pmap)#  class inspection_default
hostname(config-pmap-c)# inspect h323 h225
hostname(config-pmap-c)#   inspect h323 ras
hostname(config-pmap-c)#
```

Output of **show conn** shows H.323 media connections and control (connections flagged by h and output of **show h225**):

```
FWSM/admin# show conn
4 in use, 7 most used
 Network Processor 1 connections
UDP out 209.165.201.1:52906 in 10.0.0.6:1719 idle 0:00:07 Bytes 5162
 FLAGS - H
TCP out 209.165.201.1:1720 in 10.100.100.1:12139 idle 0:00:54 Bytes 1307 FLAGS - UOIh
UDP out 209.165.201.1:19253 in 10.100.100.1:17815 idle 0:00:03 Bytes 13012
 FLAGS - H
UDP out 209.165.201.1:19252 in 10.100.100.1:17814 idle 0:00:00 Bytes 1370400
 FLAGS - H
 Network Processor 2 connections
Multicast sessions:
 Network Processor 1 connections
 Network Processor 2 connections
IPv6 connections:

FWSM/admin# show h225
```

```
Total: 1
1  Concurrent Call(s) for
    Local: 10.100.100.1/12139 Foreign: 209.165.201.1/1720
0  CRV: 2
    Local: 10.100.100.1/12139  TPKT: 211        Foreign: 209.165.201.1/1720  TPKT: 113
```

HTTP Inspection

This section describes how the HTTP inspection engine works and how you can change its configuration. This section includes the following topics:

HTTP Inspection Overview

Use the HTTP inspection engine to protect against specific attacks and other threats that may be associated with HTTP traffic. HTTP inspection performs several functions.

- Enhanced HTTP inspection
- Java and ActiveX filtering

The second feature is configured in conjunction with the **filter** command. For more information about filtering, see Chapter 17, "Applying Filtering Services."

Note The **no inspect http** command also disables the **filter url** command.

The enhanced HTTP inspection feature, which is also known as an application firewall and is available when you configure an HTTP map (see "Configuring an HTTP Inspection Policy Map for Additional Inspection Control"), can help prevent attackers from using HTTP messages for circumventing network security policy. It verifies the following for all HTTP messages.

- Conformance to RFC 2616
- Use of RFC-defined methods only.
- Compliance with the additional criteria.

Configuring an HTTP Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an HTTP inspection policy map. You can then apply the inspection policy map when you enable HTTP inspection according to the "Configuring Application Inspection" section on page 21-6.

Note When you enable HTTP inspection with an inspection policy map, strict HTTP inspection with the action reset and log is enabled by default. You can change the actions performed in response to inspection failure, but you cannot disable strict inspection as long as the inspection policy map remains enabled.

To create an HTTP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 19-11. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 19-14.

Step 3 (Optional) Create an HTTP inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop, drop-connection, reset, mask, set the rate limit, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
hostname(config)# class-map type inspect http [match-all | match-any] class_map_name
hostname(config-cmap)#
```

Where *class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

c. (Optional) To match traffic with a content-type field in the HTTP response that does not match the accept field in the corresponding HTTP request message, enter the following command:

```
hostname(config-cmap)# match [not] req-resp content-type mismatch
```

d. (Optional) To match text found in the HTTP request message arguments, enter the following command:

```
hostname(config-cmap)# match [not] request args regex [regex_name | class
regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

e. (Optional) To match text found in the HTTP request message body or to match traffic that exceeds the maximum HTTP request message body length, enter the following command:

```
hostname(config-cmap)# match [not] request body {regex [regex_name | class
regex_class_name] | length gt max_bytes}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes.

f. (Optional) To match text found in the HTTP request message header, or to restrict the count or length of the header, enter the following command:

```
hostname(config-cmap)# match [not] request header {[field]
[regex [regex_name | class regex_class_name]] |
[length gt max_length_bytes | count gt max_count_bytes]}
```

Where the *field* is the predefined message header keyword. The **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes. The **count gt** *max_count* is the maximum number of header fields.

g. (Optional) To match text found in the HTTP request message method, enter the following command:

```
hostname(config-cmap)# match [not] request method {[method] |
[regex [regex_name | class regex_class_name]]
```

Where the *method* is the predefined message method keyword. The **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

h. (Optional) To match text found in the HTTP request message URI, enter the following command:

```
hostname(config-cmap)# match [not] request uri {regex [regex_name | class
regex_class_name] | length gt max_bytes}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes.

i. (Optional) To match text found in the HTTP response message body, or to comment out Java applet and Active X object tags in order to filter them, enter the following command:

```
hostname(config-cmap)# match [not] response body {[active-x] | [java-applet] |
[regex [regex_name | class regex_class_name]] | length gt max_bytes}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes.

j. (Optional) To match text found in the HTTP response message header, or to restrict the count or length of the header, enter the following command:

```
hostname(config-cmap)# match [not] response header {[field]
[regex [regex_name | class regex_class_name]] |
[length gt max_length_bytes | count gt max_count]}
```

Where the *field* is the predefined message header keyword. The **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes. The **count gt** *max_count* is the maximum number of header fields.

k. (Optional) To match text found in the HTTP response message status line, enter the following command:

```
hostname(config-cmap)# match [not] response status-line {regex [regex_name | class
regex_class_name]}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

Step 4 Create an HTTP inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect http policy_map_name
```

```
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the HTTP class map that you created in Step 3 by entering the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop-connection [send-protocol-error]| reset] [log]}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for the exact options available.

The **drop-connection** keyword drops the packet and closes the connection.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 19-7.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To check for HTTP protocol violations, enter the following command:

```
hostname(config-pmap-p)# protocol-violation [action [drop-connection | reset | log]]
```

Where the **drop-connection** action closes the connection. The **reset** action closes the connection and sends a TCP reset to the client. The **log** action sends a system log message when this policy map matches traffic.

c. To enforce banner obfuscation, enter the following command:

```
hostname(config-pmap-p)# mask-banner
```

The **mask-banner** command is the default when **policy-map type inspect** is configured, however **no mask-banner** can be entered to change the default.

d. To substitute a string for the server header field, enter the following command:

```
hostname(config-pmap-p)# spoof-server string
```

Where the *string* argument is the string to substitute for the server header field. Note: WebVPN streams are not subject to the **spoof-server** command.

The following example shows how to define an HTTP inspection policy map that will allow and log any HTTP connection that attempts to access “www\.example.com/.*\.asp" or "www\.example[0-9][0-9]\.com" with methods "GET" or "PUT." All other URL/Method combinations will be silently allowed.

```
hostname(config)# regex url1 “www\.example.com/.*\.asp”
hostname(config)# regex url2 “www\.example[0-9][0-9]\.com”
hostname(config)# regex get “GET”
hostname(config)# regex put “PUT”

hostname(config)# class-map type regex match-any url_to_log
hostname(config-cmap)# match regex url1
hostname(config-cmap)# match regex url2
hostname(config-cmap)# exit

hostname(config)# class-map type regex match-any methods_to_log
hostname(config-cmap)# match regex get
hostname(config-cmap)# match regex put
hostname(config-cmap)# exit

hostname(config)# class-map type inspect http http_url_policy
hostname(config-cmap)# match request uri regex class url_to_log
hostname(config-cmap)# match request method regex class methods_to_log
hostname(config-cmap)# exit

hostname(config)# policy-map type inspect http http_policy
hostname(config-pmap)# class http_url_policy
hostname(config-pmap-c)# log
```

ICMP Inspection

ICMP inspection is disabled by default.

For information about ICMP inspection, see the **inspect icmp** and **inspect icmp error** command pages in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

ILS Inspection

ILS inspection is disabled by default.

For information about ILS inspection, see the **inspect ils** command page in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

MGCP Inspection

This section describes how to enable and configure MGCP application inspection and change the default port configuration. This section includes the following topics:

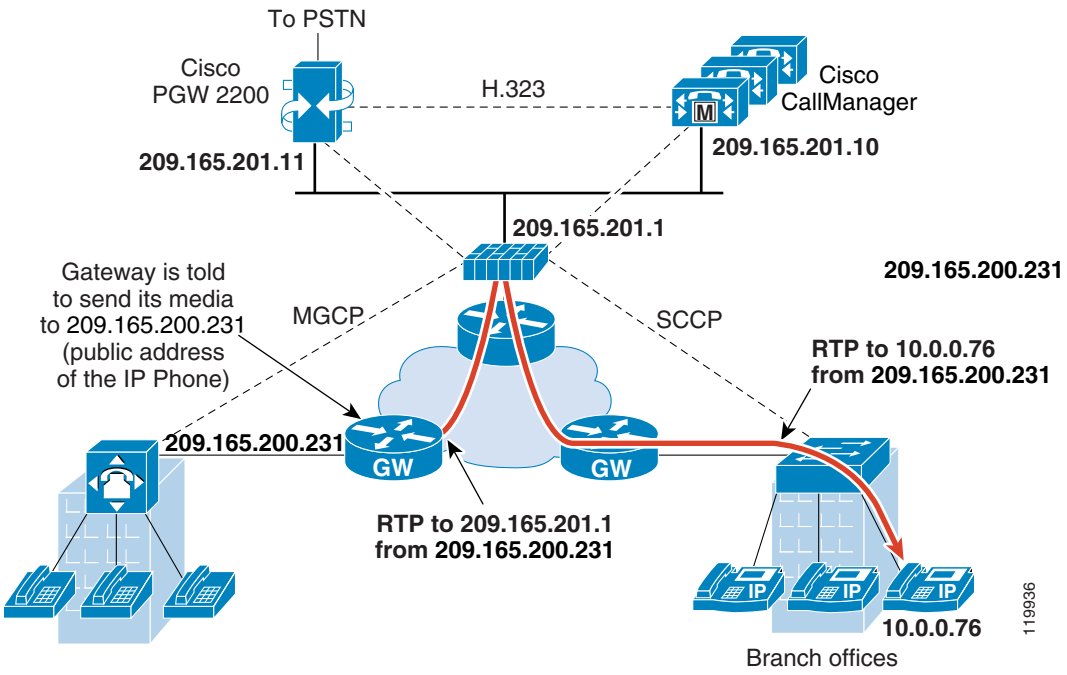
MGCP Inspection Overview

MGCP is a master/slave protocol used to control media gateways from external call control elements called media gateway controllers or call agents. A media gateway is typically a network element that provides conversion between the audio signals carried on telephone circuits and data packets carried over the Internet or over other packet networks. Using NAT and PAT with MGCP lets you support a large number of devices on an internal network with a limited set of external (global) addresses. Examples of media gateways are as follows.

- Trunking gateways that interface between the telephone network and a VoIP network. Such gateways typically manage a large number of digital circuits.
- Residential gateways that provide a traditional analog (RJ11) interface to a VoIP network. Examples of residential gateways include cable modem/cable set-top boxes, xDSL devices, broad-band wireless devices.
- Business gateways that provide a traditional digital PBX interface or an integrated soft PBX interface to a VoIP network.

MGCP messages are transmitted over UDP. A response is sent back to the source (IP address and UDP port number) of the command, but the response may not arrive from the same address as the command was sent to. This can happen when multiple call agents are being used in a failover configuration and the call agent that received the command has passed control to a backup call agent, which then sends the response. Figure 21-11 illustrates how NAT can be used with MGCP.

Figure 21-11 Using NAT with MGCP



MGCP endpoints are physical or virtual sources and destinations for data. Media gateways contain endpoints on which the call agent can create, modify and delete connections to establish and control media sessions with other multimedia endpoints. Also, the call agent can instruct the endpoints to detect certain events and generate signals. The endpoints automatically communicate changes in service state to the call agent.

MGCP transactions are composed of a command and a mandatory response. There are eight types of commands.

- CreateConnection
- ModifyConnection
- DeleteConnection
- NotificationRequest
- Notify
- AuditEndpoint
- AuditConnection
- RestartInProgress

The first four commands are sent by the call agent to the gateway. The Notify command is sent by the gateway to the call agent. The gateway may also send a DeleteConnection command. The registration of the MGCP gateway with the call agent is achieved by the RestartInProgress command. The AuditEndpoint and the AuditConnection commands are sent by the call agent to the gateway.

All commands are composed of a Command header, optionally followed by a session description. All responses are composed of a Response header, optionally followed by a session description.

To use MGCP, you usually need to configure inspection for traffic sent to two ports:

- The port on which the gateway receives commands from the call agent. Gateways usually listen to UDP port 2427.
- The port on which the call agent receives commands from the gateway. Call agents usually listen to UDP port 2727.

Note MGCP inspection does not support the use of different IP addresses for MGCP signaling and RTP data. A common and recommended practice is to send RTP data from a resilient IP address, such as a loopback or virtual IP address; however, the FWSM requires the RTP data to come from the same address as MGCP signalling.

Configuring MGCP Call Agents and Gateways

Use the **call-agent** command to specify a group of call agents that can manage one or more gateways. The call agent group information is used to open connections for the call agents in the group (other than the one a gateway sends a command to) so that any of the call agents can send the response. Call agents with the same *group_id* belong to the same group. A call agent may belong to more than one group. The *group_id* option is a number from 0 to 4294967295. The *ip_address* option specifies the IP address of the call agent.

To specify a group of call agents, enter the **call-agent** command in MGCP map configuration mode, which is accessible by entering the **mgcp-map** command in global configuration mode.

Note Using call agents to control the MGCP gateways does not restrict calls between the gateways. For example, the FWSM does not deny voice calls based on the call agent or gateway IP addresses configured by using the **mgcp-map** command. The gateways can make voice calls even when they are not configured by using the **mgcp-map** command.

Use the **gateway** command to specify which group of call agents are managing a particular gateway. The IP address of the gateway is specified with the *ip_address* option. The *group_id* option is a number from 0 to 4294967295 that must correspond with the *group_id* of the call agents that are managing the gateway. A gateway may only belong to one group.

Note MGCP call agents send AUEP messages to determine if MGCP end points are present. This establishes a flow through the FWSM and allows MGCP end points to register with the call agent.

Configuring and Enabling MGCP Inspection

To enable and configure MGCP application inspection, perform the following steps:

Step 1 Define an access list with ACEs that identify the two ports required for receiving MGCP traffic. The standard ports are UDP ports 2427 and 2727. To create the access list, use the **access-list extended** command, as follows:

```
hostname(config)# access-list acl-name permit udp any any eq port-1
hostname(config)# access-list acl-name permit udp any any eq port-2
```

where *acl-name* is the name you assign to the access list, *port-1* is the first MGCP port and *port-2* is the second MGCP port.

Step 2 Create a class map or modify an existing class map to identify MGCP traffic. Use the **class-map** command to do so, as follows.

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 3 Use a **match access-list** command to identify MGCP traffic with the access list you created in Step 1.

```
hostname(config-cmap)# match access-list acl-name
```

Step 4 (Optional) If the network has multiple call agents and gateways for which the FWSM has to open pinholes, create an MGCP map. To do so, perform the following steps.

a. Create an MGCP map by using the **mgcp-map** command. The **mgcp-map** command lets you create parameters for MGCP inspection. Use the **mgcp-map** command as follows.

```
hostname(config-cmap)# mgcp-map map_name
hostname(config-mgcp-map)#
```

where *map_name* is the name of the MGCP map. The system enters MGCP map configuration mode and the CLI prompt changes accordingly.

b. Configure the call agents. To do so, use the **call-agent** command once per call agent, as follows.

```
hostname(config-mgcp-map)# call-agent ip_address group_id
```

c. Configure the gateways. To do so, use the **gateway** command once per gateway, as follows.

```
hostname(config-mgcp-map)# gateway ip_address group_id
```

d. (Optional) If you want to change the maximum number of commands allowed in the MGCP command queue, use the **command-queue** command, as follows:

```
hostname(config-mgcp-map)# command-queue command_limit
```

Step 5 Create a policy map or modify an existing policy map that you want to use to apply the MGCP inspection engine to MGCP traffic. To do so, use the **policy-map** command, as follows.

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 6 Specify the class map, created in Step 2, that identifies the MGCP traffic. Use the **class** command to do so, as follows.

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where *class_map_name* is the name of the class map you created in Step 2. The CLI enters the policy map class configuration mode and the prompt changes accordingly.

Step 7 Enable MGCP application inspection. To do so, use the **inspect mgcp** command, as follows.

```
hostname(config-pmap-c)# inspect mgcp [map_name]
hostname(config-pmap-c)#
```

where *map_name* is the MGCP map that you may have created in optional Step 4.

Step 8 Use the **service-policy** command to apply the policy map globally or to a specific interface, as follows:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname(config)#
```

where *policy_map_name* is the policy map you configured in Step 5. If you want to apply the policy map to traffic on all the interfaces, use the **global** option. If you want to apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

The FWSM begins inspecting MGCP traffic, as specified.

Example 21-10 shows how to identify MGCP traffic, define a MGCP map, define a policy, and apply the policy to the outside interface. This creates a class map to match MGCP traffic on the default ports (2427 and 2727). This configuration allows call agents 10.10.11.5 and 10.10.11.6 to control gateway 10.10.10.115, and allows call agents 10.10.11.7 and 10.10.11.8 to control both gateways 10.10.10.116 and 10.10.10.117. The maximum number of MGCP commands that can be queued is 150. The service policy is then applied to the outside interface.

Example 21-10 Enabling and Configuring MGCP Inspection

```
hostname(config)# access-list mgcp_acl permit udp any any eq 2427
hostname(config)# access-list mgcp_acl permit udp any any eq 2727
hostname(config)# class-map mgcp-traffic
hostname(config-cmap)# match access-list mgcp_acl
hostname(config-cmap)# mgcp-map sample_map
hostname(config-mgcp-map)# call-agent 10.10.11.5 101
hostname(config-mgcp-map)# call-agent 10.10.11.6 101
hostname(config-mgcp-map)# call-agent 10.10.11.7 102
hostname(config-mgcp-map)# call-agent 10.10.11.8 102
hostname(config-mgcp-map)# gateway 10.10.10.115 101
hostname(config-mgcp-map)# gateway 10.10.10.116 102
hostname(config-mgcp-map)# gateway 10.10.10.117 102
hostname(config-mgcp-map)# command-queue 150
hostname(config-mgcp-map)# policy-map sample_policy
hostname(config-pmap)# class mgcp_port
hostname(config-pmap-c)# inspect mgcp sample_map
hostname(config-pmap-c)# service-policy sample_policy interface outside
```

Configuring MGCP Timeout Values

The **timeout mgcp** command lets you set the interval for inactivity after which an MGCP media connection is closed. The default is five minutes.

The **timeout mgcp-pat** command lets you set the timeout for PAT xlates. Because MGCP does not have a keepalive mechanism, if you use non-Cisco MGCP gateways (call agents), the PAT xlates are torn down after the default timeout interval, which is 30 seconds.

Verifying and Monitoring MGCP Inspection

The **show mgcp commands** command lists the number of MGCP commands in the command queue. The **show mgcp sessions** command lists the number of existing MGCP sessions. The **detail** option includes additional information about each command (or session) in the output. The following is sample output from the **show mgcp commands** command.

```
hostname# show mgcp commands
1 in use, 1 most used, 200 maximum allowed
CRCX, gateway IP: host-pc-2, transaction ID: 2052, idle: 0:00:07
```

The following is sample output from the **show mgcp commands detail** command:

```
hostname# show mgcp commands detail
1 in use, 1 most used, 200 maximum allowed
CRCX, idle: 0:00:10
        Gateway IP      host-pc-2
        Transaction ID  2052
        Endpoint name   aaln/1
        Call ID         9876543210abcdef
        Connection ID
        Media IP        192.168.5.7
        Media port      6058
```

The following is sample output from the **show mgcp sessions** command:

```
hostname# show mgcp sessions
1 in use, 1 most used
Gateway IP host-pc-2, connection ID 6789af54c9, active 0:00:11
```

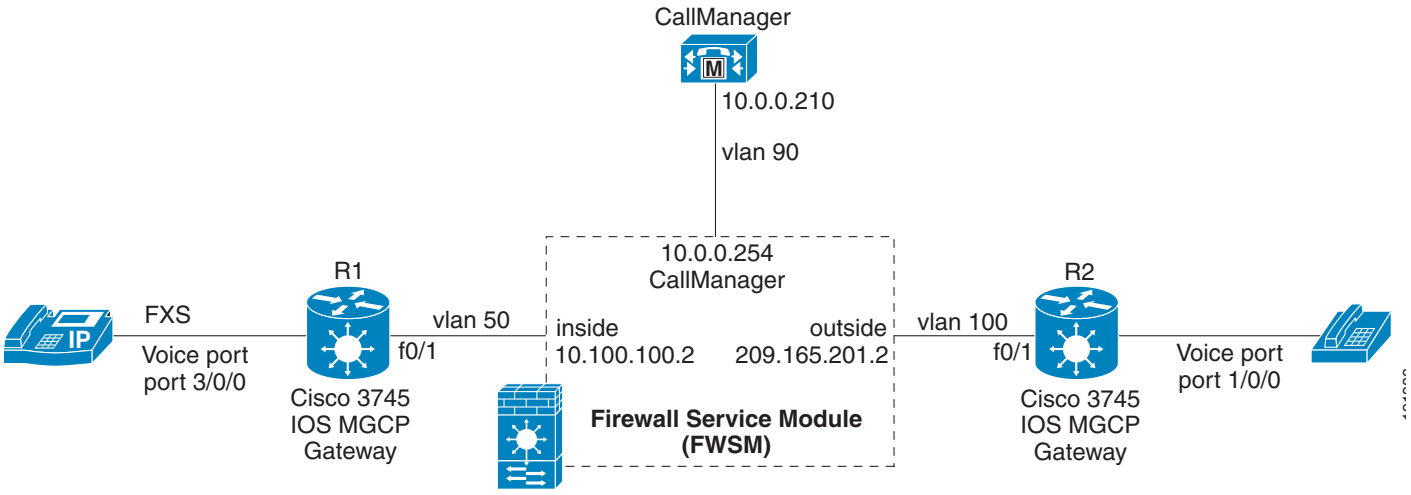
The following is sample output from the **show mgcp sessions detail** command:

```
hostname# show mgcp sessions detail
1 in use, 1 most used
Session active 0:00:14
        Gateway IP      host-pc-2
        Call ID         9876543210abcdef
        Connection ID   6789af54c9
        Endpoint name   aaln/1
        Media lcl port  6166
        Media rmt IP    192.168.5.7
        Media rmt port  6058
```

MGCP Sample Configuration

Figure 21-12 shows a sample configuration for MGCP inspection:

Figure 21-12 MGCP Inspection Setup



See the following configuration for this example:

```
hostname(config)# interface Vlan100
hostname(config-if)# nameif outside
hostname(config-if)# security-level 0
hostname(config-if)# ip address 209.165.201.2 255.0.0.0
hostname(config-if)# !
hostname(config-if)# interface Vlan50
hostname(config-if)# nameif inside
hostname(config-if)# security-level 100
hostname(config-if)# ip address 10.100.100.2 255.0.0.0
hostname(config-if)# !
hostname(config-if)# interface Vlan90
hostname(config-if)# nameif callmgr
hostname(config-if)# security-level 75
hostname(config-if)# ip address 10.0.0.254 255.0.0.0
```

TFTP port is enabled so that IOS MGCP gateway can download configuration files from the Cisco CallManager. MGCP control protocol over UDP port 2427 is enabled for pass through. MGCP backup port TCP 2428 is enabled.

```
hostname(config-if)# access-list mgcp extended permit udp any host 10.0.0.210 eq 2428
hostname(config)# access-list mgcp extended permit udp any any eq 2427
hostname(config)# access-list mgcp extended permit udp any any eq tftp
```

Apply the above access lists on the inside and outside interfaces for incoming traffic:

```
hostname(config)# access-group mgcp in interface outside
hostname(config)# access-group mgcp in interface inside
```

Configure call agent (IP address of the Cisco CallManager) and the IP address of the IOS MGCP gateway in an MGCP map:

```
hostname(config)# mgcp-map mgcp-inspect
hostname(config-mgcp-map)# call-agent 15.0.0.210 101
hostname(config-mgcp-map)# gateway 10.100.100.1 101
hostname(config-mgcp-map)# gateway 209.165.201.1 101
hostname(config-mgcp-map)# command-queue 150
hostname(config-mgcp-map)# exit
```

Apply MGCP inspection with MGCP map:

```
hostname(config)# policy-map global_policy
```

```
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect mgcp mgcp-inspect
```

Output of **show conn** in admin context, which shows the connections established between IOS MGCP gateways and call agents:

```
hostname(config)# show conn
6in use, 48 most used
    Network Processor 1 connections
    Network Processor 2 connections
UDP out 15.0.0.210:2427 in 10.100.100.1:2427 idle 0:00:04 Bytes 5790 FLAGS - g
UDP out 209.165.201.1:2427 in 10.0.0.210:2427 idle 0:00:00 Bytes 8221 FLAGS - g
UDP out 209.165.201.1:18199 in 10.100.100.1:19247 idle 0:00:03 Bytes 14080 FLAGS - g
UDP out 209.165.201.1:18199 in 10.100.100.1:19246 idle 0:00:00 Bytes 4030838 FLAGS - g
TCP out 10.0.0.210:2428 in 10.100.100.1:12695 idle 0:00:04 Bytes 1346 FLAGS - UOI
TCP out 209.165.201.1:15954 in 10.0.0.210:2428 idle 0:00:00 Bytes 1346 FLAGS - UBOI
Multicast sessions:
    Network Processor 1 connections
    Network Processor 2 connections
IPV6 connections:
```

IOS MGCP gateway configuration of router R2:

```
hostname(config)# interface FastEthernet 0/1
hostname(config-if)# ip address 209.165.201.1 255.0.0.0
hostname(config-if)# no shut
hostname(config-if)# ip host FWSM-CCM-14 10.0.0.210
hostname(config-if)# exit
hostname(config)# ip route 10.0.0.0 255.0.0.0 209.165.201.2
hostname(config)# mgcp
hostname(config)# mgcp call-agent FWSM-CCM-14
hostname(config)# mgcp dtmf-relay voip codec all mode out-of-band
hostname(config)# ccm-manager mgcp
hostname(config)# ccm-manager config server 10.0.0.210
hostname(config)# ccm-manager config
hostname(config)# dial-peer voice 100 pots
hostname(config-dial-peer)# application mgcpapp
hostname(config-dial-peer)# port 1/0/0
hostname(config-dial-peer)# no shut
```

IOS MGCP gateway configuration of router R1:

```
hostname(config)# interface FastEthernet 0/1
hostname(config-if)# ip address 10.100.100.1 255.0.0.0
hostname(config-if)# no shut
hostname(config-if)# exit
hostname(config)# ip route 10.0.0.0 255.0.0.0 10.100.100.2
hostname(config)# ccm-manager mgcp
hostname(config)# ccm-manager music-on-hold
hostname(config)# ccm-manager config server 10.0.0.210
hostname(config)# ccm-manager config
hostname(config)# mgcp
hostname(config)# mgcp call-agent FWSM-CCM-14
hostname(config)# mgcp dtmf-relay voip codec all mode out-of-band
hostname(config)# dial-peer voice 101 pots
hostname(config-dial-peer)# application mgcpapp
hostname(config-dial-peer)# port 3/0/0
```

NetBIOS Inspection

NetBIOS inspection is enabled by default.

For information about NetBIOS inspection, see the **inspect netbios** command page in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

PPTP Inspection

PPTP inspection is disabled by default.

For information about PPTP inspection, see the **inspect pptp** command page in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

RSH Inspection

RSH inspection is enabled by default.

For information about RSH inspection, see the **inspect rsh** command page in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

RTSP Inspection

This section describes how to enable RTSP application inspection and change the default port configuration. This section includes the following topics:

RTSP Inspection Overview

You control RTSP application inspection with the **inspect rtsp** command, available in policy map class configuration mode. This command is disabled by default. The **inspect rtsp** command lets the FWSM pass RTSP packets. RTSP is used by RealAudio, RealNetworks, Apple QuickTime 4, RealPlayer, and Cisco IP/TV connections.

Note For Cisco IP/TV, use RTSP TCP port 554 and TCP 8554.

RTSP applications use the well-known port 554 with TCP (rarely UDP) as a control channel. The FWSM supports TCP only, in conformity with RFC 2326. This TCP control channel is used to negotiate the data channels that is used to transmit audio/video traffic, depending on the transport mode that is configured on the client.

The supported RDT transports are: rtp/avp, rtp/avp/udp, x-real-rdt, x-real-rdt/udp, and x-pn-tng/udp.

The FWSM parses SETUP response messages with a status code of 200. If the response message is travelling inbound, the server is outside relative to the FWSM and dynamic channels need to be opened for connections coming inbound from the server. If the response message is outbound, then the FWSM does not need to open dynamic channels.

Because RFC 2326 does not require that the client and server ports must be in the SETUP response message, the FWSM keeps state and remembers the client ports in the SETUP message. QuickTime places the client ports in the SETUP message and then the server responds with only the server ports.

RTSP inspection supports PAT. It does not support dual-NAT, however. Also, the FWSM cannot recognize HTTP cloaking, which hides RTSP messages in the HTTP messages.

Using RealPlayer

When using RealPlayer, it is important to properly configure transport mode. For the FWSM, add an **access-list** command from the server to the client or vice versa. For RealPlayer, change transport mode by choosing **Options > Preferences > Transport > RTSP Settings**.

If using TCP mode on the RealPlayer, check the **Use TCP to Connect to Server** and **Attempt to use TCP for all content** check boxes. On the FWSM, there is no need to configure the inspection engine.

If using UDP mode on the RealPlayer, check the **Use TCP to Connect to Server** and **Attempt to use UDP for static content** check boxes, and for live content not available via Multicast. On the FWSM, add an **inspect rtsp** *port* command.

Restrictions and Limitations

The following restrictions apply to RTSP inspection:

- The FWSM does not support multicast RTSP or RTSP messages over UDP.
- The FWSM does not have the ability to recognize HTTP cloaking, which hides RTSP messages in the HTTP messages.
- The FWSM cannot perform NAT on RTSP messages because the embedded IP addresses are contained in the SDP files as part of HTTP or RTSP messages. Packets could be fragmented and FWSM cannot perform NAT on fragmented packets.
- With Cisco IP/TV, the number of NATs the FWSM performs on the SDP part of the message is proportional to the number of program listings in the Content Manager (each program listing can have at least six embedded IP addresses).
- You can configure NAT for Apple QuickTime 4 or RealPlayer. Cisco IP/TV only works with NAT if the Viewer and Content Manager are on the outside network and the server is on the inside network.

Enabling and Configuring RTSP Inspection

To enable or configure RTSP application inspection, perform the following steps:

Step 1 Determine the ports receiving RTSP SETUP messages behind the FWSM. The default ports are TCP ports 554 and 8554.

Step 2 Create an access list to identify the RTSP SETUP messages. Use the **access-list extended** command to do so, adding an ACE to match each port, as follows.

```
hostname(config)# access-list acl-name any any tcp eq port_number
```

Tip If you allow RTSP SETUP messages on one port only or on a contiguous range or ports, you can skip creating the access list and, in Step 4, use the **match port** command instead of the **match access-list** command.

Step 3 Create a class map or modify an existing class map to identify RTSP traffic. Use the **class-map** command to do so, as follows.

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 4 Identify traffic sent to the RTSP ports you determined in Step 1. To do so, use a **match access-list** command, as follows.

```
hostname(config-cmap)# match access-list acl-name
```

Step 5 Create a policy map or modify an existing policy map that you want to use to apply the RTSP inspection engine to RTSP traffic. To do so, use the **policy-map** command, as follows.

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 6 Specify the class map, created in Step 3, that identifies the RTSP traffic. Use the **class** command to do so, as follows.

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where *class_map_name* is the name of the class map you created. The CLI enters the policy map class configuration mode and the prompt changes accordingly.

Step 7 Enable RTSP application inspection. To do so, use the **inspect rtsp** command, as follows.

```
hostname(config-pmap-c)# inspect rtsp
hostname(config-pmap-c)#
```

Step 8 Use the **service-policy** command to apply the policy map globally or to a specific interface, as follows:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname(config)#
```

where *policy_map_name* is the policy map you configured in Step 5. If you want to apply the policy map to traffic on all the interfaces, use the **global** option. If you want to apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

The FWSM begins inspecting RTSP traffic, as specified.

Example 21-11 shows how to enable the RTSP inspection engine RTSP traffic on the default ports (554 and 8554). The service policy is then applied to the outside interface.

Example 21-11 Enabling and Configuring RTSP Inspection

```
hostname(config)# access-list rtsp_acl permit tcp any any eq 554
hostname(config)# access-list rtsp_acl permit tcp any any eq 8554
hostname(config)# class-map rtsp-traffic
hostname(config-cmap)# match access-list rtsp_acl
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class rtsp_port
hostname(config-pmap-c)# inspect rtsp
hostname(config-pmap-c)# service-policy sample_policy interface outside
```

SIP Inspection

This section describes how to enable SIP application inspection and change the default port configuration. This section includes the following topics:

SIP Inspection Overview

SIP, as defined by the IETF, enables call handling sessions, particularly two-party audio conferences, or "calls." SIP works with SDP for call signalling. SDP specifies the ports for the media stream. Using SIP, the FWSM can support any SIP VoIP gateways and VoIP proxy servers. SIP and SDP are defined in the following RFCs.

- SIP: Session Initiation Protocol, RFC 2543
- SDP: Session Description Protocol, RFC 2327

Supporting SIP calls through the FWSM requires inspection of signaling messages for the media connection addresses, media ports, and embryonic connections for the media. While SIP signalling is sent over a well-known destination port (UDP/TCP 5060), the media streams use dynamically allocated ports. Also, SIP embeds IP addresses in the user-data portion of the IP packet and SIP inspection applies NAT for these embedded IP addresses.

The following limitations and restrictions apply when using PAT with SIP:

- If a remote endpoint tries to register with a SIP proxy on a network protected by the FWSM, the registration fails under very specific conditions, as follows:
  - PAT is configured for the remote endpoint.
  - The SIP registrar server is on the outside network.
  - The port is missing in the contact field in the REGISTER message sent by the endpoint to the proxy server.

- If a SIP device transmits a packet in which the SDP portion has an IP address in the owner/creator field (o=) that is different than the IP address in the connection field (c=), the IP address in the o= field may not be properly translated. This is due to a limitation in the SIP protocol, which does not provide a port value in the o= field.

SIP Instant Messaging

Instant Messaging refers to the transfer of messages between users in near real-time. SIP supports the Chat feature on Windows XP using Windows Messenger RTC Client version 4.7.0105 only. The MESSAGE/INFO methods and 202 Accept response are used to support IM as defined in the following RFCs.

- Session Initiation Protocol (SIP)-Specific Event Notification, RFC 3265
- Session Initiation Protocol (SIP) Extension for Instant Messaging, RFC 3428

MESSAGE/INFO requests can come in at any time after registration/subscription. For example, two users can be online at any time, but not chat for hours. Therefore, the SIP inspection engine opens pinholes that time out according to the configured SIP timeout value. This value must be configured at least five minutes longer than the subscription duration. The subscription duration is defined in the Contact Expires value and is typically 30 minutes.

Because MESSAGE/INFO requests are typically sent using a dynamically allocated port other than port 5060, they are required to go through the SIP inspection engine.

Note Only the Chat feature is currently supported. Whiteboard, File Transfer, and Application Sharing are not supported. RTC Client 5.0 is not supported.

SIP inspection NATs the SIP text-based messages, recalculates the content length for the SDP portion of the message, and recalculates the packet length and checksum. It dynamically opens media connections for ports specified in the SDP portion of the SIP message as address/ports on which the endpoint should listen.

SIP inspection has a database with indices CALL_ID/FROM/TO from the SIP payload. These indices identify the call, the source, and the destination. This database contains the media addresses and media ports found in the SDP media information fields and the media type. There can be multiple media addresses and ports for a session. The FWSM opens RTP/RTCP connections between the two endpoints using these media addresses/ports.

The well-known port 5060 must be used on the initial call setup (INVITE) message; however, subsequent messages may not have this port number. The SIP inspection engine opens signaling connection pinholes, and marks these connections as SIP connections. This is done for the messages to reach the SIP application and be NATed.

As a call is set up, the SIP session is in the “transient” state until the media address and media port is received from the called endpoint in a Response message indicating the RTP port the called endpoint listens on. If there is a failure to receive the response messages within one minute, the signaling connection is torn down.

Once the final handshake is made, the call state is moved to active and the signaling connection remains until a BYE message is received.

If an inside endpoint initiates a call to an outside endpoint, a media hole is opened to the outside interface to allow RTP/RTCP UDP packets to flow to the inside endpoint media address and media port specified in the INVITE message from the inside endpoint. Unsolicited RTP/RTCP UDP packets to an inside interface does not traverse the FWSM, unless the FWSM configuration specifically allows it.

IP Address Privacy

When IP Address Privacy is enabled, if any two SIP endpoints participating in an IP phone call or instant messaging session use the same internal firewall interface to contact their SIP proxy server on an external firewall interface, all SIP signaling messages go through the SIP proxy server.

IP Address Privacy can be enabled when SIP over TCP or UDP application inspection is enabled. By default, this feature is disabled. If IP Address Privacy is enabled, the FWSM does not translate internal and external host IP addresses embedded in the TCP or UDP payload of inbound SIP traffic, ignoring translation rules for those IP addresses.

You control whether this feature is enabled by using the **ip-address-privacy** command in SIP map configuration mode.

Configuring a SIP Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create a SIP inspection policy map. You can then apply the inspection policy map when you enable SIP inspection according to the "Configuring Application Inspection" section on page 21-6.

To create a SIP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the "Creating a Regular Expression" section on page 19-11. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the "Creating a Regular Expression Class Map" section on page 19-14.s

Step 3 (Optional) Create a SIP inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match all of the match commands to match the class map if match-all is specified. Traffic must match any one of the match commands to match the class map if the match-any keyword is used.

You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop-connection, reset, drop, drop-connection log, reset log, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
hostname(config)# class-map type inspect sip [match-all | match-any] class_map_name
hostname(config-cmap)#
```

Where *the class_map_name* is the name of the class map. The name can be a string up to 40 characters. The match-all keyword is the default, and specifies that traffic must match all criteria to match the class map if match-all is specified. The match-any keyword specifies that the traffic matches the class map if any of the match commands in the class map is matched.

The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
hostname(config-cmap)# description string
```

Where *string* is the description of the class map (up to 200 characters).

c. (Optional) To match a called party, as specified in the To header or Contact header, enter the following command:

```
hostname(config-cmap)# match [not] called-party regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

d. (Optional) To match a calling party, as specified in the From header, enter the following command:

```
hostname(config-cmap)# match [not] calling-party regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

e. (Optional) To match a content length in the SIP header, enter the following command:

```
hostname(config-cmap)# match [not] content length gt length
```

Where *length* is the number of bytes the content length is greater than. 0 to 65536.

f. (Optional) To match an SDP content type or regular expression, enter the following command:

```
hostname(config-cmap)# match [not] content type {sdp | regex {class class_name |
regex_name}}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

g. (Optional) To match a SIP IM subscriber, enter the following command:

```
hostname(config-cmap)# match [not] im-subscriber regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

h. (Optional) To match a SIP via header, enter the following command:

```
hostname(config-cmap)# match [not] message-path regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

i. (Optional) To match a SIP request method, enter the following command:

```
hostname(config-cmap)# match [not] request-method method
```

Where *method* is the type of method to match (ack, bye, cancel, info, invite, message, notify, options, prack, refer, register, subscribe, unknown, update).

j. (Optional) To match the requester of a third-party registration by matching the From header in SIP REGISTER messages, enter the following command. This command only matches the requestor when the contents of the To and From fields in a SIP REGISTER message are different.

```
hostname(config-cmap)# match [not] third-party-registration regex {class class_name |
regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

k. (Optional) To match an URI in the SIP headers, enter the following command:

```
hostname(config-cmap)# match [not] uri {sip | tel} length gt length
```

Where *length* is the number of bytes the URI is greater than. 0 to 65536.

Step 4 Create a SIP inspection policy map, enter the following command:

```
hostname(config)# policy-map type inspect sip policy_map_name
hostname(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
hostname(config-pmap)# description string
```

The string for the description can contain up to 200 characters.

Step 6 To apply actions to matching traffic, perform the following steps.

a. (Optional) Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the SIP class map that you created in Step 3 by entering the following command:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
hostname(config-pmap-c)# {[drop | drop-connection | mask | reset] [log]}
```

Not all options are available for each **match** or **class** command. See the CLI help or the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for the exact options available.

The **drop** keyword drops all packets that match.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 19-7.

Step 7 (Optional) To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. To enable or disable instant messaging, enter the following command. Instant messaging is enabled by default.

```
hostname(config-pmap-p)# im
```

c. To enable or disable IP address privacy, enter the following command:

```
hostname(config-pmap-p)# ip-address-privacy
```

d. To enable check on Max-forwards header field being 0 (which cannot be 0 before reaching the destination), enter the following command:

```
hostname(config-pmap-p)# max-forwards-validation action {drop | drop-connection |
reset | log} [log]
```

e. To enable check on RTP packets flowing on the pinholes for protocol conformance, enter the following command:

```
hostname(config-pmap-p)# rtp-conformance [enforce-payloadtype]
```

Where the **enforce-payloadtype** keyword enforces the payload type to be audio or video based on the signaling exchange.

f. To identify the Server and User-Agent header fields, which expose the software version of either a server or an endpoint, enter the following command:

```
hostname(config-pmap-p)# software-version action {mask | log} [log]
```

Where the **mask** keyword removes the SIP header containing the software version and the Alert-Info and Call-Info header fields in the SIP messages.

g. To enable state checking validation, enter the following command:

```
hostname(config-pmap-p)# state-checking action {drop | drop-connection | reset | log}
[log]
```

h. To enable strict verification of the header fields in the SIP messages according to RFC 3261, enter the following command:

```
hostname(config-pmap-p)# strict-header-validation action {drop | drop-connection |
reset | log} [log]
```

Note To send a TCP reset from the universal access concentrator (UAC) to the user agent server (UAS) when there is a violation in SIP message header, you must configure the **service resetinbound** command in addition to entering the **reset log** keywords for the **strict-header-validation** command.

When the security level is different on the inside and outside interfaces, the reset is sent to the inside host only. To send the reset to the outside, you must configure the **service resetinbound** command and enter the **reset log** keywords for the **strict-header-validation** command.

i. To allow non SIP traffic using the well-known SIP signaling port, enter the following command. Allowing non SIP traffic using the well-known SIP signaling port is enabled by default.

```
hostname(config-pmap-p)# traffic-non-sip
```

j. To identify the presence of SIP headers such as the Alert-Info and Call-Info header fields in SIP messages, enter the following command:

```
hostname(config-pmap-p)# uri-non-sip action {mask | log} [log]
```

The following example shows how to disable instant messaging over SIP:

```
hostname(config)# policy-map type inspect sip mymap
hostname(config-pmap)# parameters
hostname(config-pmap-p)# no im

hostname(config)# policy-map global_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# no inspect sip
hostname(config-pmap-c)# inspect sip mymap

hostname(config)# service-policy global_policy global
```

Configuring SIP Timeout Values

The media connections are torn down within two minutes after the connection becomes idle. This is, however, a configurable timeout and can be set for a shorter or longer period of time. To configure the timeout for the SIP control connection, use the following command:

```
hostname(config)# timeout sip hh:mm:ss
```

This command configures the idle timeout after which a SIP control connection is closed.

To configure the timeout for the SIP disconnect, use the following command:

```
hostname(config)# timeout sip_disconnect hh:mm:ss
```

This command configures the idle timeout after which SIP media is deleted and media xlates are closed. Range is from 1 to 10 minutes.

To configure the timeout for the SIP invite, use the following command:

```
hostname(config)# timeout sip_invite hh:mm:ss
```

In the absence of the EXPIRE field in the SIP header, this command configures the idle timeout after which pinholes for provisional responses and media xlates are closed. Range is from 1 to 30 minutes. Default is 30 minutes.

To configure the timeout for the SIP media timer, use the following command:

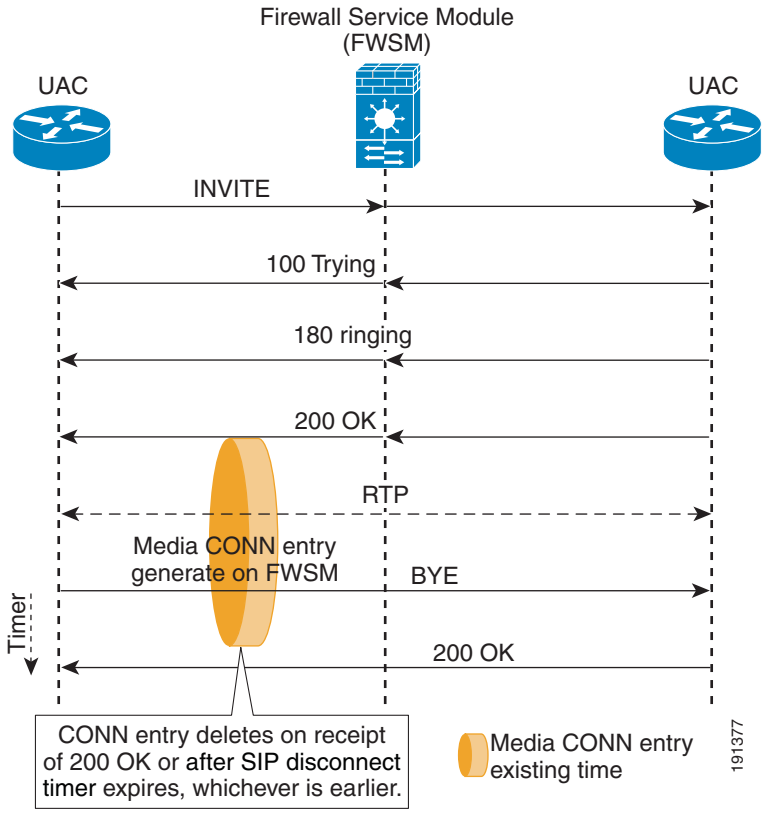
```
hostname(config)# timeout sip_media hh:mm:ss
```

This command modifies the SIP media idle timer, which determines the time after which the SIP RTP/RTCP connections are torn down due to inactivity. This command overrides the UDP inactivity timeout **timeout udp** command for SIP media connections only. Default timeout value is 2 minutes. Minimum timeout value is 1 minute. Maximum timeout value is 1193 hours. Value 0 causes SIP media connections not to timeout.

SIP Inspection Enhancement

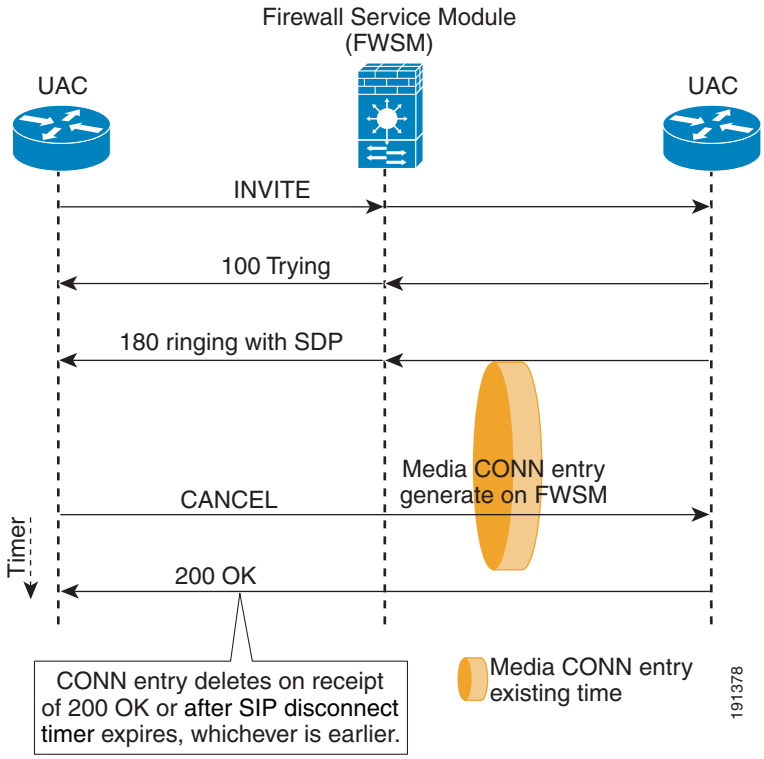
The SIP inspection enhancement removes media connections after receiving 200 OK for the BYE SIP message, 200 OK for the CANCEL SIP message, and 200 OK for 4xx/5xx/6xx SIP messages, instead of waiting for the idle timeout.

Figure 21-13 Media Connection Clear on BYE Message



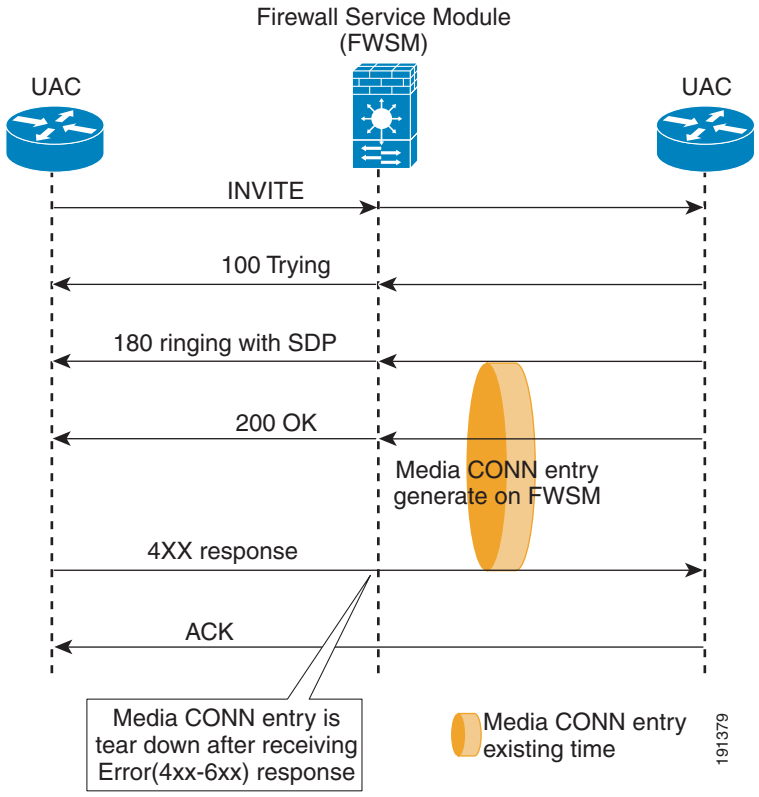
In Figure 21-13, when 200 OK is not received for the BYE message, media connections are removed after the **timeout sip-disconnect** occurs.

Figure 21-14 Media Connection Clear on CANCEL Message



In Figure 21-14, the media connection is cleared after 200 OK is received for the CANCEL message. If 200 OK is not received for the CANCEL SIP message, the media connection is cleared after the **timeout sip-disconnect** occurs.

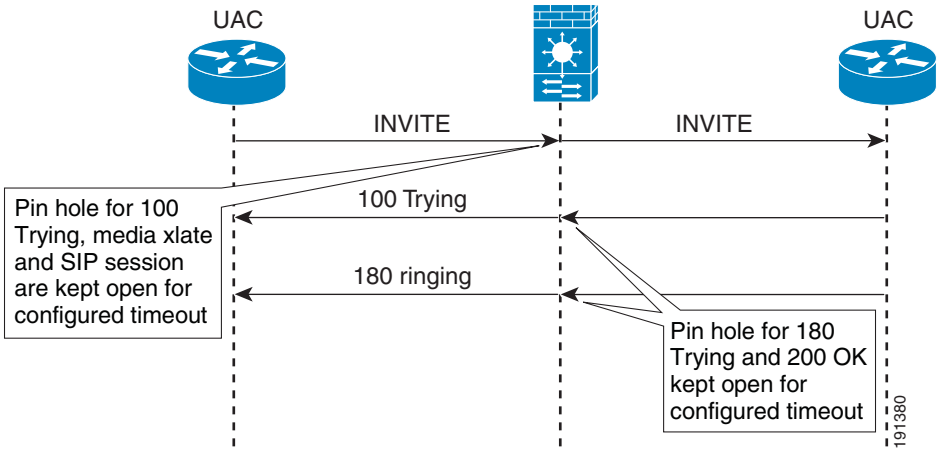
Figure 21-15 Media Connection Clear on 4xx/5xx/6xx SIP Messages



In Figure 21-15, the media connections are cleared immediately when 200 OK is received in response for 4xx/5xx/6xx SIP messages. If 200 OK is not received, the media connection is cleared after the **timeout sip-disconnect** occurs.

It is possible to extend the timeout for provisional responses. The timeout for pinholes receiving 1xx/2xx responses is configurable, or configured based on the EXPIRE field. When the EXPIRE field exists in the SIP INVITE message, and is less than 30 minutes, the timeout for pinholes for receiving 1xx/2xx responses is set to the EXPIRE field value. If the EXPIRE field value in the SIP INVITE header is greater than 30 minutes, the timeout for provisional responses is set to 30 minutes. In the absence of the EXPIRE field in the SIP INVITE message, the timeout for provisional responses is set to the value configured using the **timeout sip-invite** command.

Figure 21-16 Extending Timeout for Provisional Responses



Restrictions include:

- SIP media connections are not cleared when 200 OK is received for the BYE message, CANCEL message, or 4xx/5xx/6xx SIP message for those SIP sessions established prior to failover. SIP sessions established after failover are cleared.
- Pinholes opened for SIP responses are not cleared along with SIP media connections when 200 OK is received for the BYE SIP message, CANCEL SIP message, or 4xx/5xx/6xx SIP message. Instead, pinhole connections are cleared eventually due to timeout.

Verifying and Monitoring SIP Inspection

The **show sip** command assists in troubleshooting SIP inspection engine issues and is described with the **inspect protocol sip udp 5060** command. The **show timeout sip** command displays the timeout value of the designated protocol.

The **show sip** command displays information for SIP sessions established across the FWSM. Along with the **debug sip, show local-host,** and **show service-policy inspect sip** commands, this command is used for troubleshooting SIP inspection engine issues.

Note We recommend that you configure the **pager** command before entering the **show sip** command. If there are a lot of SIP session records and the **pager** command is not configured, it takes a while for the **show sip** command output to reach its end.

The following is sample output from the **show sip** command:

```
hostname# show sip
Total: 2
call-id c3943000-960ca-2e43-228f@10.130.56.44
    state Call init, idle 0:00:01
call-id c3943000-860ca-7e1f-11f7@10.130.56.45
    state Active, idle 0:00:06
```

This sample shows two active SIP sessions on the FWSM (as shown in the Total field). Each call-id represents a call.

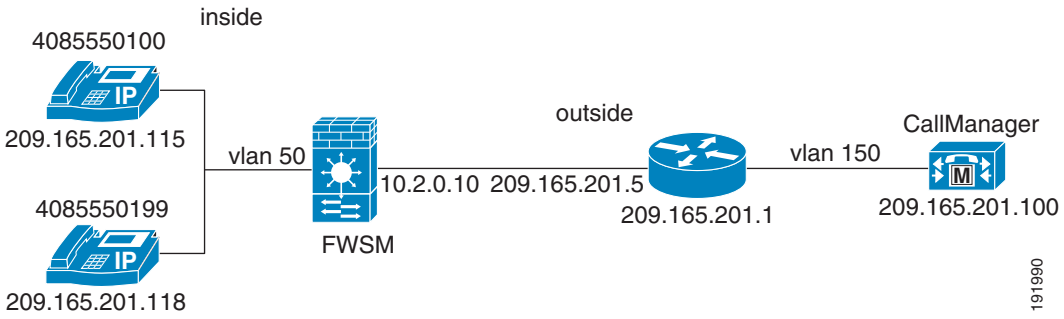
The first session, with the call-id c3943000-960ca-2e43-228f@10.130.56.44, is in the state Call Init, which means the session is still in call setup. Call setup is not complete until a final response to the call has been received. For instance, the caller has already sent the INVITE, and maybe received a 100 Response, but has not yet seen the 200 OK, so the call setup is not complete yet. Any non-1xx response message is considered a final response. This session has been idle for 1 second.

The second session is in the state Active, in which call setup is complete and the endpoints are exchanging media. This session has been idle for 6 seconds.

SIP Sample Configuration

Figure 21-17 shows a sample configuration for SIP inspection:

Figure 21-17 SIP IP Address Privacy Setup



See the following configuration for this example:

```
hostname(config)# interface Vlan12
hostname(config-if)# nameif outside
hostname(config-if)# security-level 0
hostname(config-if)# ip address 10.2.0.10 255.0.0.0
hostname(config-if)# !
```

Vlan 12 is an outside Vlan that routes all packets to 10.x.x.x network back to the FWSM with the next hop IP address set to 10.2.0.10. This is done by configuring policy-based routing at the up-stream router.

```
hostname(config-if)# interface Vlan50
hostname(config-if)# nameif inside
hostname(config-if)# security-level 100
hostname(config-if)# ip address 10.100.100.7 255.255.255.0
```

The two phones 4085550100 and 4085550199 are used in the same 209.165.201.0 subnet.

```
hostname(config)# access-list voice extended permit udp any any eq sip
hostname(config)# access-list voice extended permit tcp any any eq sip
hostname(config)# access-list voice extended permit udp any any eq tftp
hostname(config)# !
hostname(config)# sip-map privacy
hostname(config-if)# ip-address-privacy
hostname(config)# !
hostname(config)# nat-control
hostname(config)# static (inside, outside) 10.3.100.115 209.165.201.115 netmask
255.255.255.255
hostname(config)# static (inside, outside) 10.3.100.118 209.165.201.118 netmask
255.255.255.255
```

Each phone IP address is translated to an external dummy IP address that is not in a network connected to the FWSM. The translated IP address should not be in a network connected to the FWSM.

```
hostname(config)# access-group voice in interface outside
hostname(config)# access-group voice in interface inside
hostname(config)# route outside 10.3.0.0 255.0.0.0 10.2.0.5 1
```

Configure route to10.3.0.0 via 10.2.0.5 (IP address of the next hop router):

```
hostname(config)# route outside 209.165.201.0 255.0.0.0 10.2.0.5 1
hostname(config)# !
hostname(config)# policy-map global_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect dns maximum-length 512
hostname(config-pmap-c)# inspect ftp
hostname(config-pmap-c)# inspect h323 h225
hostname(config-pmap-c)# inspect h323 ras
hostname(config-pmap-c)# inspect rsh
hostname(config-pmap-c)# inspect smtp
hostname(config-pmap-c)# inspect sqlnet
hostname(config-pmap-c)# inspect skinny
hostname(config-pmap-c)# inspect sunrpc
hostname(config-pmap-c)# inspect xdmcp
hostname(config-pmap-c)# inspect netbios
hostname(config-pmap-c)# inspect tftp
hostname(config-pmap-c)# inspect sip privacy
```

Router configuration:

```
hostname(config)# interface GigabitEthernet0/2
hostname(config-if)# ip address 10.2.0.5 255.0.0.0
hostname(config-if)# ip policy route-map privacy
hostname(config-if)# duplex auto
hostname(config-if)# speed auto
hostname(config-if)# media-type rj45
hostname(config-if)# no negotiation auto
hostname(config-if)# !
hostname(config)# interface GigabitEthernet0/3
hostname(config-if)# ip address 209.165.201.1 255.0.0.0
hostname(config-if)# duplex auto
hostname(config-if)# speed auto
hostname(config-if)# media-type rj45
hostname(config-if)# no negotiation auto
hostname(config-if)# !
hostname(config-if)# ip route 10.3.0.0 255.0.0.0 10.2.0.10
hostname(config-if)# ip route 50.0.0.0 255.0.0.0 12.0.0.10
```

Configured route to 10.3.0.0 and 209.165.201.0 reachable via 10.2.0.10 (IP address of FWSM).

```
hostname(config)# access-list 10 permit ip any 10.3.0.0 0.255.255.255
hostname(config)# !
hostname(config)# route-map privacy permit 10
hostname(config-if)# match ip address 100
hostname(config-if)# set ip next-hop 10.2.0.10
```

Route map is configured to capture any packets destined for 10.3.0.0 network to be sent to 10.2.0.10 (IP address of FWSM).

The **show conn** output at the FWSM shows that voice traffic is getting switched via the FWSM module and hiding each phone IP address. RTP traffic is not switched via the same subnet. Instead it is getting routed via the FWSM.

```
hostname(config)# show conn
6 in use, 28 most used
```

```
    Network Processor 1 connections
UDP out 209.165.201.100:5060 in 209.165.201.118:5060 idle 0:00:21 Bytes 67358 FLAGS - T
UDP out 10.3.100.115:16384 in 209.165.201.118:16384 idle 0:00:00 Bytes 6042324 FLAGS - m
UDP out 209.165.201.100:0 in 209.165.201.118:5060 idle 0:00:21 Bytes 18 FLAGS - t
 Network Processor 2 connections
UDP out 209.165.201.100:5060 in 209.165.201.115:5060 idle 0:00:25 Bytes 80181 FLAGS - T
UDP out 10.3.100.118:16384 in 209.165.201.115:16384 idle 0:00:00 Bytes 6047556 FLAGS - m
UDP out 209.165.201.100:0 in 209.165.201.115:5060 idle 0:00:25 Bytes 33 FLAGS - t
Multicast sessions:
 Network Processor 1 connections
 Network Processor 2 connections
IPv6 connections:

hostname(config)# show xlate
2 in use, 2 most used
Global 30.100.100.115 Local 209.165.201.115
Global 30.100.100.118 Local 209.165.201.118
```

Skinny (SCCP) Inspection

This section describes how to enable SCCP application inspection and change the default port configuration. This section includes the following topics:

SCCP Inspection Overview

Skinny (SCCP) is a simplified protocol used in VoIP networks. Cisco IP Phones using SCCP can coexist in an H.323 environment. When used with Cisco CallManager, the SCCP client can interoperate with H.323 compliant terminals. The FWSM supports all versions through Version 17 including:

- Registrations of SCCP version 17 phones.
- SCCP version 17 media related messages for opening up pinholes for video/audio streams.

The following actions are not supported:

- Registrations of endpoints that have IPv6 addresses. The Register messages are dropped and a debug message is generated.
- If IPv6 messages are embedded in the SCCP messages, they are not NATed or PATed; they are left untranslated.

The FWSM supports PAT and NAT for SCCP. PAT is necessary if you have more IP phones than global IP addresses for the IP phones to use. By supporting NAT and PAT of SCCP Signaling packets, Skinny application inspection ensures that all SCCP signalling and media packets can traverse the FWSM.

Normal traffic between Cisco CallManager and Cisco IP Phones uses SCCP and is handled by SCCP inspection without any special configuration. The FWSM also supports DHCP options 150 and 66, which it accomplishes by sending the location of a TFTP server to Cisco IP Phones and other DHCP clients. Cisco IP Phones might also include DHCP option 3 in their requests, which sets the default route. For more information, see the "Using Cisco IP Phones with a DHCP Server" section on page 8-38.

Supporting Cisco IP Phones

In topologies where Cisco CallManager is located on the higher security interface with respect to the Cisco IP Phones, if NAT is required for the Cisco CallManager IP address, the mapping must be **static** because a Cisco IP Phone requires the Cisco CallManager IP address to be specified explicitly in its configuration. A static identity entry allows the Cisco CallManager on the higher security interface to accept registrations from the Cisco IP Phones. Cisco IP Phones require access to a TFTP server to download the configuration information they need to connect to the Cisco CallManager server.

When the Cisco IP Phones are on a lower security interface compared to the TFTP server, you must use an access list to connect to the protected TFTP server on UDP port 69. While you do need a static identity entry for the TFTP server, this does not have to be an identity static entry. When you use NAT, a static identity entry maps to the same IP address. When you use PAT, it maps to the same IP address and port.

When the Cisco IP Phones are on a *higher* security interface compared to the TFTP server and Cisco CallManager, no access list or static identity entry is required to allow the Cisco IP Phones to initiate the connection.

Restrictions and Limitations

The following are limitations that apply to the current version of PAT and NAT support for SCCP:

- PAT does not work with configurations containing the **alias** command.
- Outside NAT or PAT is *not* supported.

If the address of an internal Cisco CallManager is configured for NAT or PAT to a different IP address or port, registrations for external Cisco IP Phones fail because the FWSM currently does not support NAT or PAT for the file content transferred over TFTP. Although the FWSM supports NAT of TFTP messages and opens a pinhole for the TFTP file, the FWSM cannot translate the Cisco CallManager IP address and port embedded in the Cisco IP Phone configuration files that are transferred by TFTP during phone registration.

The following is not supported for SCCP version 17 phones:

- Registrations of endpoints that have IPv6 addresses. The Register messages are dropped and a debug message is generated.
- If IPv6 messages are embedded in the SCCP messages, they are not NATed or PATed; they are left untranslated.

Note The FWSM supports stateful failover of SCCP calls except for calls that are in the middle of call setup.

Configuring and Enabling SCCP Inspection

SCCP inspection is enabled by default.

To enable SCCP inspection or change the default port used for receiving SCCP traffic, perform the following steps:

Step 1 Name the traffic class by entering the following command in global configuration mode:

```
hostname(config)# class-map class_map_name
```

Replace *class_map_name* with the name of the traffic class, for example:

```
hostname(config)# class-map sccp_port
```

When you enter the **class-map** command, the CLI enters the class map configuration mode, and the prompt changes, as in the following example:

```
hostname(config-cmap)#
```

Step 2 In the class map configuration mode, define the **match** command, as in the following example:

```
hostname(config-cmap)# match port tcp eq 2000
hostname(config-cmap)# exit
hostname(config)#
```

To assign a range of continuous ports, enter the **range** keyword, as in the following example:

```
hostname(config-cmap)# match port tcp range 2000-2010
```

To assign more than one non-contiguous port for SCCP inspection, enter the **access-list extended** command and define an ACE to match each port. Then enter the **match** command to associate the access lists with the SCCP traffic class.

Step 3 Name the policy map by entering the following command:

```
hostname(config)# policy-map policy_map_name
```

Replace *policy_map_name* with the name of the policy map, as in the following example:

```
hostname(config)# policy-map sample_policy
```

The CLI enters the policy map configuration mode and the prompt changes accordingly, as follows:

```
hostname(config-pmap)#
```

Step 4 Specify the traffic class defined in Step 1 to be included in the policy map by entering the following command:

```
hostname(config-pmap)# class class_map_name
```

For example, the following command assigns the sccp_port traffic class to the current policy map:

```
hostname(config-pmap)# class sccp_port
```

The CLI enters the policy map class configuration mode and the prompt changes accordingly, as follows:

```
hostname(config-pmap-c)#
```

Step 5 (Optional) To change the default port used by the FWSM for receiving SCCP traffic, enter the following command:

```
hostname(config-pmap-c)# inspect skinny
```

Step 6 Return to policy map configuration mode by entering the following command:

```
hostname(config-pmap-c)# exit
hostname(config-pmap)#
```

Step 7 Return to global configuration mode by entering the following command:

```
hostname(config-pmap)# exit
hostname(config)#
```

Step 8 Apply the policy map globally or to a specific interface by entering the following command:

```
hostname(config)# service-policy policy_map_name [global | interface interface_ID
```

Replace policy_map_name with the policy map you configured in Step 3, and identify all the interfaces with the **global** option or a specific interface using the name assigned with the **nameif** command.

For example, the following command applies the sample_policy to the outside interface:

```
hostname(config)# service-policy sample_policy interface outside
```

The following command applies the sample_policy to the all the FWSM interfaces:

```
hostname(config)# service-policy sample_policy global
```

You enable the SCCP inspection engine as shown in Example 21-12, which creates a class map to match SCCP traffic on the default port (2000). The service policy is then applied to the outside interface.

Example 21-12 Enabling SCCP Application Inspection

```
hostname(config)# class-map sccp_port
hostname(config-cmap)# match port tcp eq 2000
hostname(config-cmap)# exit
hostname(config)# policy-map sample_policy
hostname(config-pmap)# class sccp_port
hostname(config-pmap-c)# inspect skinny
hostname(config-pmap-c)# exit
hostname(config)# service-policy sample_policy interface outside
```

Verifying and Monitoring SCCP Inspection

The **show skinny** command assists in troubleshooting SCCP (Skinny) inspection engine issues. The following is sample output from the **show skinny** command under the following conditions. There are two active Skinny sessions set up across the FWSM. The first one is an audio connection established between an internal Cisco IP Phone at local address 10.0.0.11 and an external Cisco CallManager at 172.18.1.33. TCP port 2000 is the CallManager. The second one is a video connection established between another internal Cisco IP Phone at local address 10.0.0.22 and the same Cisco CallManager.

```
hostname# show skinny
        LOCAL                   FOREIGN                 STATE
-----------------------------------------------------------------
1       10.0.0.11/52238         172.18.1.33/2000                1
  AUDIO 10.0.0.11/22948         172.18.1.22/20798
2       10.0.0.22/52232         172.18.1.33/2000                1
  VIDEO 10.0.0.22/20798         172.18.1.11/22948
```

The output indicates that a call has been established between two internal Cisco IP Phones. The RTP listening ports of the first and second phones are UDP 22948 and 20798 respectively.

The following is sample output from the **show xlate debug** command for these Skinny connections:

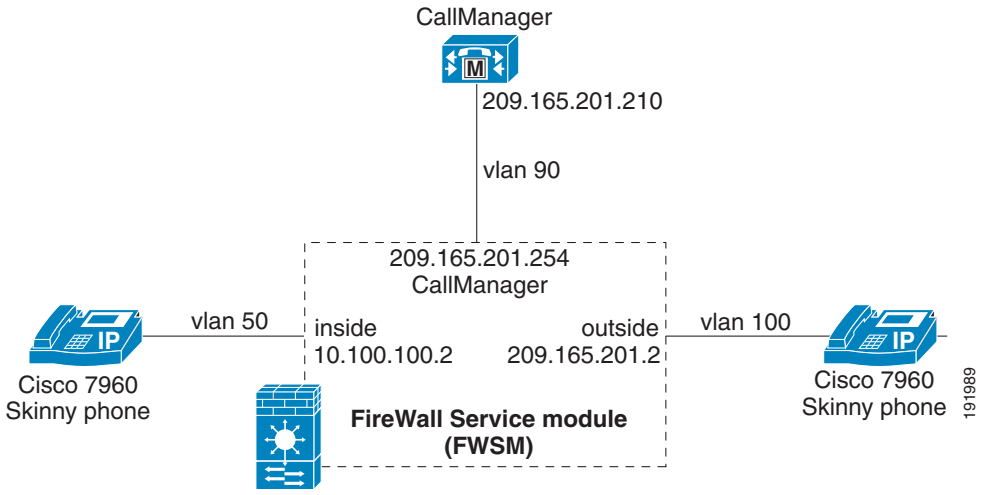
```
hostname# show xlate debug
2 in use, 2 most used
Flags:  D - DNS, d - dump, I - identity, i - inside, n - no random,
```

```
        r - portmap, s - static
NAT from inside:10.0.0.11 to outside:172.18.1.11 flags si idle 0:00:16 timeout 0:05:00
NAT from inside:10.0.0.22 to outside:172.18.1.22 flags si idle 0:00:14 timeout 0:05:00
```

SCCP (Skinny) Sample Configuration

Figure 21-18 shows a sample configuration for SCCP (Skinny) inspection:

Figure 21-18 SCCP (Skinny) Inspection Setup



See the following configuration for this example:

```
hostname(config)# interface Vlan100
hostname(config-if)# nameif outside
hostname(config-if)# security-level 0
hostname(config-if)# ip address 209.165.201.2 255.0.0.0
hostname(config-if)# !
hostname(config-if)# interface Vlan50
hostname(config-if)# nameif inside
hostname(config-if)# security-level 100
hostname(config-if)# ip address 10.100.100.2 255.0.0.0
hostname(config-if)# !
hostname(config-if)# interface Vlan90
hostname(config-if)# nameif callmgr
hostname(config-if)# security-level 75
hostname(config-if)# ip address 209.165.201.254 255.0.0.0
```

TFTP port is enabled for the IP address of the CallManager so that phones on the inside and outside can download configuration files from the CallManager for initial setup. TCP Port 2000 is enabled for the IP address of the CallManager so that skinny signaling can pass the module between the phone and the CallManager through firewall module.

```
hostname(config-if)# access-list voice extended permit udp any host 209.165.201.210 eq
tftp
hostname(config)# access-list voice extended permit tcp any host 209.165.201.210 eq 2000
```

Apply the above access lists on the inside and outside interfaces for incoming traffic:

```
hostname(config)# access-group voice in interface outside
hostname(config)# access-group voice in interface inside
```

Configure SCCP (Skinny) inspection:

```
hostname(config)# policy-map global_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect skinny
```

Output of **show skinny** when Skinny phone call through the firewall module is active:

```
hostname(config)# show skinny
            LOCAL                  FOREIGN                STATE
------------------------------------------------------------------------
1           10.0.0.2/49723         209.165.201.210/2000       1
    AUDIO   209.165.201.2/24002    209.165.201.211/19212
2           209.165.201.210/2000   209.165.201.211/49692      1
    AUDIO   10.0.0.2/24002         209.165.201.211/19212
```

Output of **show conn** when there is one active phone call between Skinny phones, each reachable through inside and outside interfaces. Skinny connections are marked by a k flag.

```
hostname(config)# show conn
3 in use, 26 most used
    Network Processor 1 connections
TCP out 209.165.201.210:2000 in 10.0.0.2:49723 idle 0:00:07 Bytes 10232 FLAGS - UOI
    Network Processor 2 connections
TCP out 209.165.201.211:49692 in 209.165.201.210:49723 idle 0:00:27 Bytes 12394 FLAGS -
UBOI
UDP out 209.165.201.211:19212 in 10.0.0.2:24002 idle 0:00:00 Bytes 3575654 FLAGS - K
Multicast sessions:
    Network Processor 1 connections
    Network Processor 2 connections
IPV6 connections:
```

SMTP and Extended SMTP Inspection

This section describes how to enable SMTP and ESMTP application inspection and change the default port configuration. This section includes the following topics:

- SMTP and Extended SMTP Inspection Overview, page 21-94
- Configuring and Enabling SMTP and Extended SMTP Application Inspection, page 21-96

SMTP and Extended SMTP Inspection Overview

The FWSM supports application inspection for SMTP and ESMTP. Application inspection for these protocols protects against attacks by restricting the types of SMTP or ESMTP commands that can pass through the FWSM and by adding monitoring capabilities.

ESMTP is an enhancement to the SMTP protocol and is similar to SMTP. For convenience, the term SMTP is used in this document to refer to both SMTP and ESMTP. The application inspection process for ESMTP includes support for SMTP sessions. Most commands used in an ESMTP session are the same as those used in an SMTP session but an ESMTP session is considerably faster and offers more options related to reliability and security, such as delivery status notification.

The **inspect smtp** command supports seven RFC 821 commands (DATA, HELO, MAIL, NOOP, QUIT, RCPT, RSET). The **inspect esmtp** command supports those seven commands and supports the following extended SMTP commands: AUTH, HELP, EHLO, ETRN, SAML, SEND, SOML and VRFY.

Other SMTP or ESMTP commands and private extensions to ESMTP and are not supported. Unsupported commands are translated into Xs, which are rejected by the SMTP server protected by the FWSM. This results in a message such as “500 Command unknown: 'XXX'.” Incomplete commands are discarded.

SMTP application inspection, as enabled by the **inspect smtp** command, occurs in fast path processing; therefore, it occurs on one of the three network processors on the FWSM. ESMTP application inspection, as enabled by the **inspect esmtp** command, occurs in control plane path processing; therefore, it occurs on the single, general purpose processor on the FWSM.

Note If a policy map contains both the **inspect smtp** command and the **inspect esmtp** command, only the first command listed in the policy map is applied to matching traffic.

Inspection changes the characters in the server SMTP banner to asterisks except for the “2”, “0”, “0” characters. Carriage return (CR) and linefeed (LF) characters are ignored.

With SMTP inspection enabled, a Telnet session used for interactive SMTP may hang if the following rules are not observed: SMTP commands must be at least four characters in length; must be terminated with carriage return and line feed; and must wait for a response before issuing the next reply.

An SMTP server responds to client requests with numeric reply codes and optional human-readable strings. SMTP application inspection controls and reduces the commands that the user can use as well as the messages that the server returns. SMTP inspection performs three primary tasks:

- Restricts SMTP requests to seven basic SMTP commands and eight extended commands.
- Monitors the SMTP command-response sequence.
- Generates an audit trail—Audit record 108002 is generated when invalid character embedded in the mail address is replaced. For more information, see RFC 821.

SMTP inspection monitors the command and response sequence for the following anomalous signatures:

- Truncated commands.
- Incorrect command termination (not terminated with <CR><LR>).
- The MAIL and RCPT commands specify who are the sender and the receiver of the mail. Mail addresses are scanned for strange characters. The pipeline character (|) is deleted (changed to a blank space) and “<” ,”>” are only allowed if they are used to define a mail address (“>” must be preceded by “<”).
- Unexpected transition by the SMTP server.
- For unknown commands, the FWSM changes all the characters in the packet to X. In this case, the server generates an error code to the client. Because of the change in the packed, the TCP checksum has to be recalculated or adjusted.
- TCP stream editing.
- Command pipelining.

Configuring and Enabling SMTP and Extended SMTP Application Inspection

SMTP inspection is enabled by default.

To enable SMTP or extended SMTP inspection, perform the following steps:

Step 1 Determine the ports that SMTP servers behind the FWSM listen to for SMTP traffic. The default port is TCP port 25 but your SMTP servers may be configured to listen to other ports.

Step 2 Create a class map or modify an existing class map to identify SMTP traffic. Use the **class-map** command to do so, as follows:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 3 Use a match command to identify traffic sent to the SMTP ports you determined in Step 1.

If the port mapper process listens to a single port, you can use the **match port** command to identify traffic sent to that port, as follows:

```
hostname(config-cmap)# match port tcp eq port_number
```

where *port_number* is the port to which the port mapper process listens. If you need to assign a range of contiguous ports, use the **range** keyword, as in the following example:

```
hostname(config-cmap)# match port tcp range begin_port_number end_port_number
```

Tip To identify two or more non-contiguous ports, enter the **access-list extended** command and define an ACE to match each port. Then, rather than the **match port** command, use the **match access-list** command to associate the access list with the SMTP traffic class.

Step 4 Create a policy map that you want to use to apply the SMTP inspection engine to the SMTP traffic. To do so, use the **policy-map** command, as follows:

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 5 Specify the class map, created in Step 2, that identifies the SMTP traffic. Use the **class** command to do so, as follows:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where *class_map_name* is the name of the class map you created in Step 2. The CLI enters the policy map class configuration mode and the prompt changes accordingly.

Step 6 Do one of the following:

a. To enable extended SMTP application inspection, enter the following command:

```
hostname(config-pmap-c)# inspect esmtp
```

b. To enable SMTP application inspection, enter the following command:

```
hostname(config-pmap-c)# inspect smtp
```

Note For information about the differences between the **inspect smtp** and **inspect esmtp** commands, see the "SMTP and Extended SMTP Inspection Overview" section on page 21-94.

Step 7 Use the **service-policy** command to apply the policy map globally or to a specific interface, as follows:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname(config)#
```

where *policy_map_name* is the policy map you configured in Step 4. If you want to apply the policy map to traffic on all the interfaces, use the **global** option. If you want to apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

The FWSM begins inspecting SMTP traffic, as specified.

Example 21-13 Configuring and Enabling ESMTP Inspection

```
hostname(config)# class-map smtp_port
hostname(config-cmap)# match port tcp eq 25
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class smtp_port
hostname(config-pmap-c)# inspect esmtp
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

SNMP Inspection

This section describes how to enable SNMP application inspection and change the default port configuration. This section includes the following topics:

- SNMP Inspection Overview, page 21-97
- Enabling and Configuring SNMP Application Inspection, page 21-98

SNMP Inspection Overview

SNMP application inspection lets you restrict SNMP traffic to a specific version of SNMP. Earlier versions of SNMP are less secure; therefore, denying certain SNMP versions may be required by your security policy. The FWSM can deny SNMP versions 1, 2, 2c, or 3. You control the versions permitted by using the **deny version** command in SNMP map configuration mode.

Enabling and Configuring SNMP Application Inspection

To change the default configuration for SNMP inspection, perform the following steps:

Step 1 Determine the ports that network devices behind the FWSM listen to for SNMP traffic. The default ports are TCP ports 161 and 162.

Step 2 Create a class map or modify an existing class map to identify SNMP traffic. Use the **class-map** command to do so, as follows:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 3 Use a match command to identify traffic sent to the SNMP ports you determined in Step 1.

If you need to assign a range of contiguous ports, use the **range** keyword, as in the following example:

```
hostname(config-cmap)# match port tcp range begin_port_number end_port_number
```

where *begin_port_number* is the lowest port in the range of SNMP ports and *end_port_number* is the highest port.

Tip To identify two or more non-contiguous ports, enter the **access-list extended** command and define an ACE to match each port. Then, rather than the **match port** command, use the **match access-list** command to associate the access list with the SNMP traffic class.

Step 4 Create an SNMP map that will contain the parameters of SNMP inspection. Use the **snmp-map** command to do so, as follows:

```
hostname(config-cmap)# snmp-map map_name
hostname(config-snmp-map)#
```

where *map_name* is the name of the SNMP map. The CLI enters SNMP map configuration mode.

Step 5 Specify the versions of SNMP permitted by the SNMP map. To do so, use the **deny version** command to disallow the versions that you do not want to permit, as follows:

```
hostname(config-snmp-map)# deny version version
hostname(config-snmp-map)#
```

where *version* with an SNMP version that you want to restrict. Valid values of *version* are 1, 2, 2c, and 3. You can enter as many **deny version** commands as needed.

Step 6 Create a policy map or modify an existing policy map that you want to use to apply the SNMP inspection engine to the SNMP traffic. To do so, use the **policy-map** command, as follows:

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 7 Specify the class map, created in Step 2, that identifies the SNMP traffic. Use the **class** command to do so, as follows:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where *class_map_name* is the name of the class map you created in Step 2. The CLI enters the policy map class configuration mode and the prompt changes accordingly.

Step 8 Enable SNMP application inspection. To do so, use the **inspect snmp** command, as follows:

```
hostname(config-pmap-c)# inspect snmp snmp_map_name
hostname(config-pmap-c)#
```

where *snmp_map_name* is the SNMP map that you created in Step 4.

Step 9 Use the **service-policy** command to apply the policy map globally or to a specific interface, as follows:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname(config)#
```

where *policy_map_name* is the policy map you configured in Step 6. If you want to apply the policy map to traffic on all the interfaces, use the **global** option. If you want to apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

The FWSM begins inspecting SNMP traffic, as specified.

Example 21-14 enables SNMP application inspection on traffic sent to TCP ports 161 and 162 from the outside interface:

Example 21-14 Configuring SNMP Application Inspection

```
hostname(config)# class-map snmp_port
hostname(config-cmap)# match port tcp range 161 162
hostname(config-cmap)# snmp-map sample_map
hostname(config-snmp-map)# deny version 1
hostname(config-snmp-map)# deny version 2
hostname(config-snmp-map)# policy-map sample_policy
hostname(config-pmap)# class snmp_port
hostname(config-pmap-c)# inspect snmp sample_map
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

SQL*Net Inspection

SQL*Net inspection is enabled by default.

For information about SQL*Net inspection, see the **inspect sqlnet** command page in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Sun RPC Inspection

This section describes how to enable Sun RPC application inspection, change the default port configuration, and manage the Sun RPC service table. This section includes the following topics:

- Sun RPC Inspection Overview, page 21-100
- Enabling and Configuring Sun RPC Inspection, page 21-100
- Managing Sun RPC Services, page 21-102

Sun RPC Inspection Overview

To enable Sun RPC application inspection or to change the ports to which the FWSM listens, use the **inspect sunrpc command** in policy map class configuration mode, which is accessible by using the **class** command within policy map configuration mode. To remove the configuration, use the **no** form of this command.

The **inspect sunrpc** command enables or disables application inspection for the Sun RPC protocol. Sun RPC is used by NFS and NIS. Sun RPC services can run on any port. When a client attempts to access an Sun RPC service on a server, it must learn the port that service is running on. It does this by querying the port mapper process, usually rpcbind, on the well-known port of 111.

The client sends the Sun RPC program number of the service and the port mapper process responds with the port number of the service. The client sends its Sun RPC queries to the server, specifying the port identified by the port mapper process. When the server replies, the FWSM intercepts this packet and opens both embryonic TCP and UDP connections on that port.

Note NAT or PAT of Sun RPC payload information is not supported.

Enabling and Configuring Sun RPC Inspection

Sun RPC inspection is enabled by default.

Note To enable or configure Sun RPC inspection over UDP, you do not have to define a separate traffic class or a new policy map. You simply add the **inspect sunrpc** command into a policy map whose traffic class is defined by the default traffic class. An example of this configuration is shown in Example 21-16 on page 21-102.

To enable Sun RPC inspection or change the default port used for receiving Sun RPC traffic using TCP, perform the following steps:

Step 1 Determine the port or ports that the port mapper process listens to. While this is most often port 111, it can differ between operating systems and implementations.

Step 2 Create a class map or modify an existing class map to identify Sun RPC traffic. Use the **class-map** command to do so, as follows:

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

where *class_map_name* is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.

Step 3 Use a **match** command to identify traffic sent to the port or ports that you determined in Step 1.

If the port mapper process listens to a single port, you can use the **match port** command to identify traffic sent to that port, as follows:

```
hostname(config-cmap)# match port tcp eq port_number
```

where *port_number* is the port to which the port mapper process listens. If you need to assign a range of contiguous ports, use the **range** keyword, as in the following example:

```
hostname(config-cmap)# match port tcp range begin_port_number end_port_number
```

Tip To identify two or more non-contiguous ports, enter the **access-list extended** command and define an ACE to match each port. Then, rather than the **match port** command, use the **match access-list** command to associate the access list with the Sun RPC traffic class.

Step 4 Create a policy map or modify an existing policy map that you want to use to apply the Sun RPC inspection engine to the Sun RPC traffic. To do so, use the **policy-map** command, as follows:

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

where *policy_map_name* is the name of the policy map. The CLI enters the policy map configuration mode and the prompt changes accordingly.

Step 5 Specify the class map, created in Step 2, that identifies the Sun RPC traffic. Use the **class** command to do so, as follows:

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where *class_map_name* is the name of the class map you created in Step 2. The CLI enters the policy map class configuration mode and the prompt changes accordingly.

Step 6 Enable Sun RPC application inspection. To do so, enter the following command:

```
hostname(config-pmap-c)# inspect sunrpc
hostname(config-pmap-c)#
```

Step 7 Use the **service-policy** command to apply the policy map globally or to a specific interface, as follows:

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface interface_ID]
hostname(config)#
```

where *policy_map_name* is the policy map you configured in Step 4. If you want to apply the policy map to traffic on all the interfaces, use the **global** option. If you want to apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* option, where *interface_ID* is the name assigned to the interface with the **nameif** command.

The FWSM begins inspecting Sun RPC traffic, as specified.

Example 21-15 Enabling and Configuring TCP-based Sun RPC Inspection

The following example enables Sun RPC application inspection on traffic sent to TCP port 111 from the outside interface:

```
hostname(config)# class-map sunrpc_port
hostname(config-cmap)# match port tcp eq 111
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class sunrpc_port
hostname(config-pmap-c)# inspect sunrpc
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

Example 21-16 enables Sun RPC over UDP, which you do by adding the **inspect sunrpc** command to a policy map that applies actions to the default traffic class:

Example 21-16 Enabling and Configuring UDP-based Sun RPC Inspection

```
hostname(config)# policy-map asa_global_fw_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect sunrpc
hostname(config-pmap-c)#
```

Managing Sun RPC Services

The FWSM maintains a Sun RPC services table to control established Sun RPC sessions. To create entries in the Sun RPC services table, use the **sunrpc-server** command in global configuration mode.

You can use the **sunrpc-server** command to specify the timeout after which the FWSM closes a pinhole opened by Sun RPC application inspection. For example, to create a timeout of 30 minutes for the Sun RPC server with the IP address 192.168.100.2, enter the following command:

```
hostname(config)# sunrpc-server inside 192.168.100.2 255.255.255.255 service 100003
protocol tcp 111 timeout 00:30:00
```

This command specifies that the pinhole that was opened by Sun RPC application inspection will be closed after 30 minutes. In this example, the Sun RPC server is on the inside interface using TCP port 111. You can also specify UDP, a different port number, or a range of ports. To specify a range of ports, separate the starting and ending port numbers in the range with a hyphen (for example, 111-113).

The service type identifies the mapping between a specific service type and the port number used for the service. To determine the service type, which in this example is 100003, use the **sunrpcinfo** command at the UNIX or Linux command line on the Sun RPC server machine.

To clear the Sun RPC configuration, enter the following command.

```
hostname(config)# clear configure sunrpc-server
```

This removes the configuration performed using the **sunrpc-server** command. The **sunrpc-server** command allows pinholes to be created with a specified timeout.

To clear the active Sun RPC services, enter the following command:

```
hostname(config)# clear sunrpc-server active
```

This clears the pinholes open because Sun RPC application inspection enabled the traffic based on service requests to the port mapper service.

Verifying and Monitoring Sun RPC Inspection

The sample output in this section is for a Sun RPC server with an IP address of 192.168.100.2 on the inside interface and a Sun RPC client with an IP address of 209.165.201.5 on the outside interface.

To view information about the current Sun RPC connections, enter the **show conn** command. The following is sample output from the **show conn** command:

```
hostname# show conn
15 in use, 21 most used
UDP out 209.165.200.5:800 in 192.168.100.2:2049 idle 0:00:04 flags -
UDP out 209.165.200.5:714 in 192.168.100.2:111 idle 0:00:04 flags -
UDP out 209.165.200.5:712 in 192.168.100.2:647 idle 0:00:05 flags -
```

```
UDP out 192.168.100.2:0 in 209.168.201.5:714 idle 0:00:05 flags i
hostname(config)#
```

To display the information about the Sun RPC service table configuration, enter the **show running-config sunrpc-server** command. The following is sample output from the **show running-config sunrpc-server** command:

```
hostname(config)# show running-config sunrpc-server
sunrpc-server inside 192.168.100.2 255.255.255.255 service 100003 protocol UDP port 111
timeout 0:30:00
sunrpc-server inside 192.168.100.2 255.255.255.255 service 100005 protocol UDP port 111
timeout 0:30:00
```

This output shows that a timeout interval of 30 minutes is configured on UDP port 111 for the Sun RPC server with the IP address 192.168.100.2 on the inside interface.

To display the pinholes open for Sun RPC services, enter the **show sunrpc-server active** command. The following is sample output from **show sunrpc-server active** command:

```
hostname# show sunrpc-server active
LOCAL FOREIGN SERVICE TIMEOUT
-----------------------------------------------
1 209.165.201.5/0 192.168.100.2/2049 100003 0:30:00
2 209.165.201.5/0 192.168.100.2/2049 100003 0:30:00
3 209.165.201.5/0 192.168.100.2/647 100005 0:30:00
4 209.165.201.5/0 192.168.100.2/650 100005 0:30:00
```

The entry in the LOCAL column shows the IP address of the client or server on the inside interface, while the value in the FOREIGN column shows the IP address of the client or server on the outside interface.

To view information about the Sun RPC services running on a Sun RPC server, enter the **rpcinfo -p** command from the Linux or UNIX server command line. The following is sample output from the **rpcinfo -p** command:

```
sunrpcserver:~ # rpcinfo -p
program vers proto port
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 632 status
100024 1 tcp 635 status
100003 2 udp 2049 nfs
100003 3 udp 2049 nfs
100003 2 tcp 2049 nfs
100003 3 tcp 2049 nfs
100021 1 udp 32771 nlockmgr
100021 3 udp 32771 nlockmgr
100021 4 udp 32771 nlockmgr
100021 1 tcp 32852 nlockmgr
100021 3 tcp 32852 nlockmgr
100021 4 tcp 32852 nlockmgr
100005 1 udp 647 mountd
100005 1 tcp 650 mountd
100005 2 udp 647 mountd
100005 2 tcp 650 mountd
100005 3 udp 647 mountd
100005 3 tcp 650 mountd
```

In this output, port 647 corresponds to the mountd daemon running over UDP. The mountd process would more commonly be using port 32780, but it uses TCP port 650 in this example.

TFTP Inspection

TFTP inspection is enabled by default.

For information about TFTP inspection, see the **inspect tftp** command page in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

XDMCP Inspection

XDMCP inspection is enabled by default; however, the XDMCP inspection engine is dependent upon proper configuration of the **established** command.

For information about XDMCP inspection, see the **established** and **inspect pptp** and command pages in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

PART 3

System Administration

CHAPTER 22

Configuring Management Access

This chapter describes how to access the FWSM for system management through Telnet, SSH, HTTPS, and VPN. It also describes how to authenticate and authorize users.

This chapter includes the following sections:

Note To access the FWSM interface for management access, you do not also need an access list allowing the host IP address. You only need to configure management access according to the sections in this chapter.

Allowing Telnet Access

The FWSM allows Telnet connections to the FWSM for management purposes. You cannot use Telnet to the lowest security interface unless you use Telnet inside an IPSec tunnel.

The FWSM allows a maximum of 5 concurrent Telnet connections per context, if available, with a maximum of 100 connections divided between all contexts. You can control the number of Telnet sessions allowed per context using resource classes. (See the "Configuring a Class" section on page 4-24.) In admin context only, you can have up to 15 Telnet and 15 SSH sessions concurrently.

Note Please note that if you have two or more concurrent Telnet or SSH sessions and one of the sessions is at the **More** prompt, the other sessions may hang until the **More** prompt is dismissed. To disable the **More** prompt and avoid this situation, enter the **pager lines 0** command.

Please note that concurrent access to the FWSM is not recommended. In some cases, two Telnet sessions issuing the same commands might cause one of the sessions to hang until a key is depressed on the other session.

To configure Telnet access to the FWSM, perform the following steps:

Step 1 To identify the IP addresses from which the FWSM accepts connections, enter the following command for each address or subnet:

```
hostname(config)# telnet source_IP_address mask source_interface
```

If there is only one interface, you can configure Telnet to access that interface as long as the interface has a security level of 100.

Step 2 (Optional) To set the duration for how long a Telnet session can be idle before the FWSM disconnects the session, enter the following command:

```
hostname(config)# telnet timeout minutes
```

Set the timeout from 1 to 1440 minutes. The default is 5 minutes. The default duration is too short in most cases and should be increased until all pre-production testing and troubleshooting has been completed.

For example, to let a host on the inside interface with an address of 192.168.1.2 access the FWSM, enter the following command:

```
hostname(config)# telnet 192.168.1.2 255.255.255.255 inside
hostname(config)# telnet timeout 30
```

To allow all users on the 192.168.3.0 network to access the FWSM on the inside interface, enter the following command:

```
hostname(config)# telnet 192.168.3.0 255.255.255.0 inside
```

Allowing SSH Access

The FWSM allows SSH connections to the FWSM for management purposes. The FWSM allows a maximum of 5 concurrent SSH connections per context, if available, with a maximum of 100 connections divided between all contexts. You can control the number of SSH sessions allowed per context using resource classes. (See the "Configuring a Class" section on page 4-24.) In admin context only, you can have up to 15 Telnet and 15 SSH sessions concurrently.

Note Please note that if you have two or more concurrent Telnet or SSH sessions and one of the sessions is at the **More** prompt, the other sessions may hang until the **More** prompt is dismissed. To disable the **More** prompt and avoid this situation, enter the **pager lines 0** command.

SSH is an application running on top of a reliable transport layer, such as TCP/IP, that provides strong authentication and encryption capabilities. The FWSM supports the SSH remote shell functionality provided in SSH Versions 1 and 2 and supports DES and 3DES ciphers.

Note XML management over SSL and SSH are not supported.

This section includes the following topics:

Configuring SSH Access

To configure SSH access to the FWSM, perform the following steps:

Step 1 To generate an RSA key pair, which is required for SSH, enter the following command:

```
hostname(config)# crypto key generate rsa modulus modulus_size
```

The modulus (in bits) is 512, 768, 1024, or 2048. The larger the key modulus size you specify, the longer it takes to generate an RSA. We recommend a value of 1024.

Step 2 To save the RSA keys to persistent Flash memory, enter the following command:

```
hostname(config)# write memory
```

Step 3 To identify the IP addresses from which the FWSM accepts connections, enter the following command for each address or subnet:

```
hostname(config)# ssh source_IP_address mask source_interface
```

The FWSM accepts SSH connections from all interfaces, including the one with the lowest security level.

Step 4 (Optional) To set the duration for how long an SSH session can be idle before the FWSM disconnects the session, enter the following command:

```
hostname(config)# ssh timeout minutes
```

Set the timeout from 1 to 60 minutes. The default is 5 minutes. The default duration is too short in most cases and should be increased until all pre-production testing and troubleshooting has been completed.

Step 5 (Optional) To restrict the version of SSH accepted by the FWSM, enter the following command. By default, the FWSM accepts both versions.

```
hostname(config)# ssh version {1 | 2)
```

For example, to generate RSA keys and let a host on the inside interface with an address of 192.168.1.2 access the FWSM, enter the following command:

```
hostname(config)# crypto key generate rsa modulus 1024
hostname(config)# write mem
hostname(config)# ssh 192.168.1.2 255.255.255.255 inside
hostname(config)# ssh 192.168.1.2 255.255.255.255 inside
hostname(config)# ssh timeout 30
```

To allow all users on the 192.168.3.0 network to access the FWSM on the inside interface, the following command:

```
hostname(config)# ssh 192.168.3.0 255.255.255.0 inside
```

Using an SSH Client

To gain access to the FWSM console using SSH, at the SSH client enter the username **pix** and enter the login password set by the **password** command (see the "Changing the Login Password" section on page 7-1). By default, the password is "cisco."

When starting an SSH session, a dot (.) displays on the FWSM console before the SSH user authentication prompt appears, as follows:

```
hostname(config)# .
```

The display of the dot does not affect the functionality of SSH. The dot appears at the console when generating a server key or decrypting a message using private keys during SSH key exchange before user authentication occurs. These tasks can take up to two minutes or longer. The dot is a progress indicator that verifies that the FWSM is busy and has not hung.

Allowing HTTPS Access for ASDM

To use ASDM, you need to enable the HTTPS server, and allow HTTPS connections to the FWSM. These tasks are completed if you use the **setup** command. This section describes how to manually configure ASDM access.

The FWSM allows a maximum of 5 concurrent ASDM instances per context, if available, with a maximum of 80 ASDM instances between all contexts. You can control the number of ASDM sessions allowed per context using resource classes. (See the "Configuring a Class" section on page 4-24.)

To configure ASDM access, perform the following steps:

Step 1 To identify the IP addresses from which the FWSM accepts HTTPS connections, enter the following command for each address or subnet:

```
hostname(config)# http source_IP_address mask source_interface
```

Step 2 To enable the HTTPS server, enter the following command:

```
hostname(config)# http server enable
```

For example, to enable the HTTPS server and let a host on the inside interface with an address of 192.168.1.2 access ASDM, enter the following commands:

```
hostname(config)# http server enable
hostname(config)# http 192.168.1.2 255.255.255.255 inside
```

To allow all users on the 192.168.3.0 network to access ASDM on the inside interface, enter the following command:

```
hostname(config)# http 192.168.3.0 255.255.255.0 inside
```

Allowing a VPN Management Connection

The FWSM supports IPSec for management access. An IPSec VPN ensures that IP packets can safely travel over insecure networks such as the Internet. All communication between two VPN peers occurs over a secure tunnel, which means the packets are encrypted and authenticated by the peers.

The FWSM can connect to another VPN concentrator, such as a Cisco PIX firewall or a Cisco IOS router, using a site-to-site tunnel. You specify the peer networks that can communicate over the tunnel. In the case of the FWSM, the only address available on the FWSM end of the tunnel is the interface itself.

In routed mode, the FWSM can also accept connections from VPN clients, either hosts running the Cisco VPN client, or VPN concentrators such as the Cisco PIX firewall or Cisco IOS router running the Easy VPN client. Unlike a site-to-site tunnel, you do not know in advance the IP address of the client. Instead, you rely on client authentication. Transparent firewall mode does not support remote clients. Transparent mode does support site-to-site tunnels.

The FWSM can support 5 concurrent IPSec connections, with a maximum of 10 concurrent connections divided between all contexts. You can control the number of IPSec sessions allowed per context using resource classes. (See the "Configuring a Class" section on page 4-24.)

This section describes the following topics:

Configuring Basic Settings for All Tunnels

The following steps are required for both VPN client access and for site-to-site tunnels, and include setting the IKE policy (IKE is part of the ISAKMP) and the IPSec transforms.

To configure basic settings for all tunnels, perform the following steps:

Step 1 To set the IKE encryption algorithm, enter the following command:

```
hostname(config)# isakmp policy priority encryption {des | 3des}
```

The **3des** keyword is more secure than **des**.

You can have multiple IKE policies. The FWSM tries each policy in order of the *priority* until the policy matches the peer policy. The *priority* can be an integer from 1 to 65,534, with 1 being the highest priority and 65,534 the lowest. Use this same priority number for the following **isakmp** commands.

Step 2 To set the Diffie-Hellman group used for key exchange, enter the following command:

```
hostname(config)# isakmp policy priority group {1 | 2}
```

Group 1 is 768 bits, and Group 2 is 1024 bits (and therefore more secure).

Step 3 To set the authentication algorithm, enter the following command:

```
hostname(config)# isakmp policy priority hash {md5 | sha}
```

The **sha** keyword is more secure than **md5**.

Step 4 To set the IKE authentication method as a shared key, enter the following command:

```
hostname(config)# isakmp policy priority authentication pre-share
```

You can alternatively use certificates instead of a shared key by specifying the **rsa-sig** option. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about this method.

Step 5 To enable IKE on the tunnel interface, enter the following command:

```
hostname(config)# isakmp enable interface_name
```

Step 6 To set the authentication and encryption methods used for IPSec tunnels in a transform set, enter the following command:

```
hostname(config)# crypto ipsec transform-set transform_name [esp-md5-hmac | esp-sha-hmac]
{esp-aes-256 | esp-aes-192 | esp-aes | esp-des | esp-3des}
```

Although you can specify authentication alone, or encryption alone, these methods are not secure.

You refer to this transform set when you configure the VPN client group or a site-to-site tunnel.

You can refer to up to 6 transform sets for the tunnel, and the sets are checked in order until the transforms match.

The authentication and encryption algorithms of this transform typically match the IKE policy (**isakmp policy** commands). For site-to-site tunnels, this transform must match the peer transform.

Authentication options include the following (from most secure to least secure):

- **esp-sha-hmac**
- **esp-md5-hmac**

Encryption options include the following (from most secure to least secure):

- **esp-aes-256**
- **esp-aes-192**
- **esp-aes**
- **esp-3des**
- **esp-des**

Note **esp-null** (no encryption) is for testing purposes only.

For example, to configure the IKE policy and the IPSec transform sets, enter the following commands:

```
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)# isakmp policy 1 group 2
hostname(config)# isakmp policy 1 hash sha
hostname(config)# isakmp enable outside
hostname(config)# crypto ipsec transform-set vpn_client esp-3des esp-sha-hmac
hostname(config)# crypto ipsec transform-set site_to_site esp-3des ah-sha-hmac
```

Configuring VPN Client Access

In routed mode, a host with Version 3.0 or 4.0 of the Cisco VPN client can connect to the FWSM for management purposes over a public network, such as the Internet.

Transparent firewall mode does not support remote clients. Transparent mode does support site-to-site tunnels.

To allow remote clients to connect to the FWSM for management access, first configure basic VPN settings (see "Configuring Basic Settings for All Tunnels"), and then perform the following steps:

Step 1 To specify the transform sets (defined in the "Configuring Basic Settings for All Tunnels" section on page 22-5) allowed for client tunnels, enter the following command:

```
hostname(config)# crypto dynamic-map dynamic_map_name priority set transform-set
transform_set1 [transform_set2] [...]
```

List multiple transform sets in order of priority (highest priority first).

This dynamic crypto map allows unknown IP addresses to connect to the FWSM.

The **dynamic-map** name is used in Step 2.

The *priority* specifies the order in which multiple commands are evaluated. If you have a command that specifies one set of transforms, and another that specifies others, then the priority number determines the command that is evaluated first.

Step 2 To assign the dynamic crypto map (from Step 1) to a static tunnel, enter the following command:

```
hostname(config)# crypto map crypto_map_name priority ipsec-isakmp dynamic
dynamic_map_name
```

Step 3 To specify the interface at which you want the client tunnels to terminate, enter the following command:

```
hostname(config)# crypto map crypto_map_name interface interface_name
```

You can apply only one **crypto map** name to an interface, so if you want to terminate both a site-to-site tunnel and VPN clients on the same interface, they need to share the same **crypto map** name.

Step 4 To specify the range of addresses that VPN clients use on the FWSM enter the following command:

```
hostname(config)# ip local pool pool_name first_ip_address-last_ip_address [mask mask]
```

All tunneled packets from the client use one of these addresses as the source address.

Step 5 To specify the traffic that is destined for the FWSM, so you can tunnel only that traffic according to the **tunnel group** command in Step 7, enter the following command:

```
hostname(config)# access-list acl_name [extended] permit {protocol} host
fwsm_interface_address pool_addresses mask
```

This access list identifies traffic from the local pool (see Step 4) destined for the FWSM interface. See the "Adding an Extended Access List" section on page 12-6 for more information about access lists.

Step 6 To assign the VPN address pool to a tunnel group, enter the following command:

```
hostname(config)# tunnel-group name general-attributes address-pool pool_name
```

This group specifies VPN characteristics for connecting clients. When a client connects to the FWSM, they need to enter the tunnel group name and password in Step 8.

Step 7 To specify that only traffic destined for the FWSM is tunneled, enter the following commands:

```
hostname(config)# group-policy name attributes
hostname(config-group-policy)# split-tunnel-policy tunnelall
```

This command is required.

Step 8 To set the VPN group password, enter the following command:

```
hostname(config)# group-policy group_name external server-group server_group_name password
server_password
```

Step 9 To allow Telnet or SSH access, see the "Allowing Telnet Access" section on page 22-1 and the "Allowing SSH Access" section on page 22-2.

Specify the VPN pool addresses in the **telnet** and **ssh** commands.

For example, the following commands allow VPN clients to use Telnet on the outside interface (209.165.200.225). The user authentication is the local database, so users with the tunnel group name and password, as well as the username "admin" and the password "passw0rd" can connect to the FWSM.

```
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)# isakmp policy 1 group 2
```

```
hostname(config)# isakmp policy 1 hash sha
hostname(config)# isakmp enable outside
hostname(config)# username admin password passw0rd
hostname(config)# crypto ipsec transform-set vpn esp-3des esp-sha-hmac
hostname(config)# crypto dynamic-map vpn_client 1 set transform-set vpn
hostname(config)# crypto map telnet_tunnel 1 ipsec-isakmp dynamic vpn_client
hostname(config)# crypto map telnet_tunnel interface outside
hostname(config)# crypto map telnet_tunnel client authentication LOCAL
hostname(config)# ip local pool Firstpool 10.1.1.1-10.1.1.2
hostname(config)# access-list VPN_SPLIT extended permit ip host 209.165.200.225 host 10.1.1.1
hostname(config)# access-list VPN_SPLIT extended permit ip host 209.165.200.225 host 10.1.1.2
hostname(config)# tunnel-group StocktonAAA general-attributes address-pool Firstpool
hostname(config)# group-policy name attributes
hostname(config-group-policy)# split-tunnel-policy tunnelall
hostname(config)# group-policy ExternalGroup external server-group LodiAAA password $ecure23
hostname(config)# telnet 10.1.1.1 255.255.255.255 outside
hostname(config)# telnet 10.1.1.2 255.255.255.255 outside
hostname(config)# telnet timeout 30
```

Configuring a Site-to-Site Tunnel

To configure a site-to-site tunnel, first configure basic VPN settings (see "Configuring Basic Settings for All Tunnels"), and then perform the following steps:

Step 1 To set the shared key used by both peers, enter the following command:

```
hostname(config)# isakmp key keystring address peer-address
```

Step 2 To identify the traffic allowed to go over the tunnel, enter the following command:

```
hostname(config)# access-list acl_name [extended] {deny | permit} {protocol} host
fwsm_interface_address dest_address mask
```

For the destination address, specify the addresses that are allowed to access the FWSM.

See the "Adding an Extended Access List" section on page 12-6 for more information about access lists.

Step 3 To create an IPSec tunnel, enter the following command:

```
hostname(config)# crypto map crypto_map_name priority ipsec-isakmp
```

All tunnel attributes are identified by the same **crypto map** name.

The *priority* specifies the order in which multiple commands are evaluated. If you have a command for this **crypto map** name that specifies **ipsec-isakmp**, and another that specifies **ipsec-isakmp dynamic** (for VPN client connections), then the priority number determines the command that is evaluated first.

Step 4 To assign the access list from Step 2 to this tunnel, enter the following command:

```
hostname(config)# crypto map crypto_map_name priority match address acl_name
```

Step 5 To specify the remote peer on which this tunnel terminates, enter the following command:

```
hostname(config)# crypto map crypto_map_name priority set peer ip_address
```

Step 6 To specify the transform sets for this tunnel (defined in the "Configuring Basic Settings for All Tunnels" section on page 22-5), enter the following command:

```
hostname(config)# crypto map crypto_map_name priority set transform-set transform_set1
[transform_set2] [...]
```

List multiple transform sets in order of priority (highest priority first). You can specify up to six transform sets.

Step 7 To specify the interface at which you want this tunnel to terminate, enter the following command:

```
hostname(config)# crypto map crypto_map_name interface interface_name
```

You can apply only one **crypto map** name to an interface, so if you want to terminate both a site-to-site tunnel and VPN clients on the same interface, they need to share the same **crypto map** name.

This command must be entered after all other **crypto map** commands. If you change any **crypto map** settings, remove this command with the **no** prefix, and reenter it.

Step 8 To allow Telnet or SSH access, see the "Allowing Telnet Access" section on page 22-1 and the "Allowing SSH Access" section on page 22-2.

For example, the following commands allow hosts connected to the peer router (209.165.202.129) to use Telnet on the outside interface (209.165.200.225).

```
hostname(config)# isakmp policy 1 authentication pre-share
hostname(config)# isakmp policy 1 encryption 3des
hostname(config)# isakmp policy 1 group 2
hostname(config)# isakmp policy 1 hash sha
hostname(config)# isakmp enable outside
hostname(config)# crypto ipsec transform-set vpn esp-3des esp-sha-hmac
hostname(config)# isakmp key 7mfi02lirotn address 209.165.200.223
hostname(config)# access-list TUNNEL extended permit ip host 209.165.200.225 209.165.201.0
255.255.255.224
hostname(config)# crypto map telnet_tunnel 2 ipsec-isakmp
hostname(config)# crypto map telnet_tunnel 1 match address TUNNEL
hostname(config)# crypto map telnet_tunnel 1 set peer 209.165.202.129
hostname(config)# crypto map telnet_tunnel 1 set transform-set vpn
hostname(config)# crypto map telnet_tunnel interface outside
hostname(config)# telnet 209.165.201.0 255.255.255.224 outside
hostname(config)# telnet timeout 30
```

Allowing ICMP to and from the FWSM

By default, ICMP (including ping) is not allowed to an FWSM interface (or through the FWSM. To allow ICMP *through* the FWSM, see Chapter 14, "Permitting or Denying Network Access."). ICMP is an important tool for testing your network connectivity; however, it can also be used to attack the FWSM or your network. We recommend allowing ICMP during your initial testing, but then disallowing it during normal operation.

See the "Rule Limits" section on page A-6 for information about the maximum number of ICMP rules allowed for the entire system.

To permit or deny address(es) to reach an FWSM interface with ICMP (either from a host to the FWSM, or from the FWSM to a host, which requires the ICMP reply to be allowed back), enter the following command:

```
hostname(config)# icmp {permit | deny} {host ip_address | ip_address mask | any}
[icmp_type] interface_name
```

If you do not specify an *icmp_type*, all types are identified. You can enter the number or the name. To control ping, specify **echo-reply** (**0**) (FWSM to host) or **echo** (**8**) (host to FWSM). See the "ICMP Types" section on page E-15 for a list of ICMP types.

Like access lists, the FWSM matches a packet to each **icmp** statement in order. You should use specific statements first, and general statements later. There is an implicit deny at the end. For example, if you allow all addresses first, then deny a specific address after, then that address will be unintentionally allowed because it matched the first statement.

Note If you only want to allow the FWSM to ping a host (and thus allow the echo reply back to the interface), and not allow hosts to ping the FWSM, you can enable the ICMP inspection engine instead of entering the command above. See Chapter 21, "Applying Application Layer Protocol Inspection."

For example, to allow all hosts except the one at 10.1.1.15 to use ICMP to the inside interface, enter the following commands:

```
hostname(config)# icmp deny host 10.1.1.15 inside
hostname(config)# icmp permit any inside
```

To allow the host at 10.1.1.15 to use only ping to the inside interface, enter the following commands:

```
hostname(config)# icmp permit host 10.1.1.15 inside
```

AAA for System Administrators

This section describes how to enable CLI authentication, command authorization, and command accounting for system administrators. Before you configure AAA for system administrators, first configure the local database or AAA server according to Chapter 11, "Configuring AAA Servers and the Local Database."

Note In multiple context mode, you cannot configure any AAA commands in the system configuration. However, if you configure Telnet authentication in the admin context, then authentication also applies to sessions from the switch to the FWSM (which enters the system execution space). See the "Configuring Authentication for CLI and ASDM Access" section on page 22-10 for more information.

This section includes the following topics:

Configuring Authentication for CLI and ASDM Access

This section explains how to configure CLI authentication when you use Telnet or SSH, and how to configure ASDM authentication. This section includes the following topics:

CLI Access Overview

Before the FWSM can authenticate a Telnet or SSH user, you must first configure access to the FWSM using the **telnet** or **ssh** commands (see the "Allowing Telnet Access" section on page 22-1 and "Allowing SSH Access" section on page 22-2). These commands identify the IP addresses that are allowed to communicate with the FWSM. The exception is for access to the system in multiple context mode; a session from the switch to the FWSM is a Telnet session, but the **telnet** command is not required.

After you connect to the FWSM, you log in and access user EXEC mode.

- If you do not enable any authentication for Telnet, you do not enter a username; you enter the login password (set with the **password** command). For SSH, you enter "pix" as the username, and enter the login password.
- If you enable Telnet or SSH authentication according to this section, you enter the username and password as defined on the AAA server or local user database.

To enter privileged EXEC mode, enter the **enable** command or the **login** command (if you are using the local database only).

- If you do not configure enable authentication, enter the system enable password when you enter the **enable** command (set by the **enable password** command). However, if you do not use enable authentication, after you enter the **enable** command, you are no longer logged in as a particular user. To maintain your username, use enable authentication.
- If you configure enable authentication (see the "Configuring Authentication for the Enable Command" section on page 22-13), the FWSM prompts you for your username and password.

For authentication using the local database, you can use the **login** command, which maintains the username but requires no configuration to turn on authentication.

ASDM Access Overview

By default, you can log into ASDM with a blank username and the enable password set by the **enable password** command. However, if you enter a username and password at the login screen (instead of leaving the username blank), ASDM checks the local database for a match.

Although you can configure HTTP authentication according to this section and specify the local database, that functionality is always enabled by default. You should only configure HTTP authentication if you want to use a RADIUS or TACACS+ server for authentication.

Authenticating Sessions from the Switch to the FWSM

In multiple context mode, you cannot configure any AAA commands in the system configuration. However, if you configure Telnet authentication in the admin context, then authentication also applies to sessions from the switch to the FWSM (which enters the system execution space). The admin context AAA server or local user database are used in this instance.

Enabling CLI or ASDM Authentication

To authenticate users who access the CLI or ASDM, enter the following command:

```
hostname(config)# aaa authentication {telnet | ssh | http} console {LOCAL |
server_group [LOCAL]}
```

The **telnet** keyword enables authentication for Telnet sessions, and when you configure this command in the admin context, for sessioning from the switch to the FWSM.

The **ssh** keyword enables authentication for SSH sessions.

The **http** keyword authenticates the ASDM client that accesses the FWSM using HTTPS.

If you use a TACACS+ or RADIUS server group for authentication, you can configure the FWSM to use the local database as a fallback method if the AAA server is unavailable. Specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the AAA server because the FWSM prompt does not give any indication which method is being used.

You can alternatively use the local database as your main method of authentication (with no fallback) by entering **LOCAL** alone.

For example, to enable authentication for sessions from the switch to the FWSM system execution space, enter the following commands starting from the switch CLI:

```
Router# session slot 1 processor 1 (for an FWSM in slot 1)
The default escape character is Ctrl-^, then x.
You can also type 'exit' at the remote prompt to end the session
Trying 127.0.0.41 ... Open

User Access Verification

Password: cisco (the default login password)
Type help or '?' for a list of available commands.
hostname> enable
hostname# configure terminal
hostname(config)# changeto context admin (change from the system execution space to the admin
context called "admin")
hostname/admin(config)# aaa-server RADS protocol radius (adds a server group called RADS)
hostname/admin(config-aaa-server-group)# aaa-server RADS (mgmt) host 192.168.1.4 cisco
(adds a RADIUS server to the RADS server group)
hostname/admin(config-aaa-server-group)# exit
hostname/admin(config)# aaa authentication telnet console RADS (enables Telnet authentication
using the RADS server group)
```

The next time you session from the switch to the FWSM, you are prompted for a username and password defined on the RADIUS server:

```
Router# session slot 1 processor 1
The default escape character is Ctrl-^, then x.
You can also type 'exit' at the remote prompt to end the session
Trying 127.0.0.41 ... Open

User Access Verification

Username: myRADIUSusername
Password: myRADIUSpassword
Type help or '?' for a list of available commands.
hostname>
```

Configuring Authentication to Access Privileged EXEC Mode

You can configure the FWSM to authenticate users with a AAA server or the local database when they enter the **enable** command. Alternatively, users are automatically authenticated with the local database when they enter the **login** command, which also accesses privileged EXEC mode depending on the user level in the local database.

This section includes the following topics:

Configuring Authentication for the Enable Command

You can configure the FWSM to authenticate users when they enter the **enable** command. If you do not authenticate the **enable** command, when you enter **enable**, the FWSM prompts for the enable password (set by the **enable password** command), and you are no longer logged in as a particular user. Applying authentication to the **enable** command maintains the username. This feature is particularly useful when you perform command authorization, where usernames are important to determine the commands a user can enter.

To authenticate users who enter the **enable** command, enter the following command:

```
hostname(config)# aaa authentication enable console {LOCAL | server_group [LOCAL]}
```

The user is prompted for the username and password.

If you use a TACACS+ or RADIUS server group for authentication, you can configure the FWSM to use the local database as a fallback method if the AAA server is unavailable. Specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the AAA server because the FWSM prompt does not give any indication which method is being used.

You can alternatively use the local database as your main method of authentication (with no fallback) by entering **LOCAL** alone.

Authenticating Users Using the Login Command

From user EXEC mode, you can log in as any username in the local database using the **login** command.

Unlike enable authentication, this method is available in the system execution space in multiple context mode. The system execution space uses the admin context local user database when you enter the **login** command; the system configuration does not contain a local user database (you cannot enter the **username** command).

The login feature allows users to log in with their own username and password to access privileged EXEC mode, so you do not have to give out the system enable password to everyone. To allow users to access privileged EXEC mode (and all commands) when they log in, set the user privilege level to 2 (the default) through 15. If you configure local command authorization, then the user can only enter commands assigned to that privilege level or lower. See the "Configuring Local Command Authorization" section on page 22-15 for more information.

Caution If you add users to the local database who can gain access to the CLI and whom you do not want to enter privileged EXEC mode, you should configure command authorization. Without command authorization, users can access privileged EXEC mode (and all commands) at the CLI using their own password if their

privilege level is 2 or greater (2 is the default). Alternatively, you can use RADIUS or TACACS+ authentication, or you can set all local users to level 1 so you can control who can use the system enable password to access privileged EXEC mode.

To log in as a user from the local database, enter the following command:

```
hostname> login
```

The FWSM prompts for your username and password. After you enter your password, the FWSM places you in the privilege level that the local database specifies. You can only enter the **login** command in user EXEC mode. If you are in privileged EXEC mode, enter the **disable** command to return to user EXEC mode.

Configuring Command Authorization

By default when you log in, you can access user EXEC mode, which offers only minimal commands. When you enter the **enable** command (or the **login** command when you use the local database), you can access privileged EXEC mode and advanced commands, including configuration commands. If you want to control the access to commands, the FWSM lets you configure command authorization, where you can determine which commands are available to a user.

This section includes the following topics:

Command Authorization Overview

You can use one of two command authorization methods:

- Local database—Configure the command privilege levels on the FWSM. When a local user authenticates with the **enable** command (or logs in with the **login** command), the FWSM places that user in the privilege level that is defined by the local database. The user can then access commands at the user privilege level and below.

  You can use local command authorization without any users in the local database and without CLI or enable authentication. To do so, when you enter the **enable** command, use the system enable password, and the FWSM places you in level 15 as the default “enable_15” username. You can create enable passwords for every level, so that when you enter **enable** *n* (2 to 15), the FWSM places you in level *n*. These levels are not used unless you turn on local command authorization (see “Configuring Local Command Authorization”). (See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information about the **enable** command.)

- TACACS+ server—On the TACACS+ server, configure the commands that a user or group can use after they authenticate for CLI access. Every command that a user enters at the CLI is checked with the TACACS+ server.

Security Contexts and Command Authorization

The following are important points to consider when implementing command authorization with multiple security contexts:

- AAA settings are discrete per context, not shared between contexts.

  When configuring command authorization, you must configure each security context separately. This provides you the opportunity to enforce different command authorizations for different security contexts.

  When switching between security contexts, administrators should be aware that the commands permitted for the username specified when they login may be different in the new context session or that command authorization may not be configured at all in the new context. Failure to understand that command authorizations may differ between security contexts could confuse an administrator. This behavior is further complicated by the next point.

- New context sessions started with the **changeto** command always use the default "enable_15" username as the administrator identity, regardless of what username was used in the previous context session. This behavior can lead to confusion if command authorization is not configured for the enable_15 user or if authorizations are different for the enable_15 user than for the user in the previous context session.

  This behavior also affects command accounting, which is useful only if you can accurately associate each command that is issued with a particular administrator. Because all administrators with permission to use the **changeto** command can use the enable_15 username in other contexts, command accounting records may not readily identify who was logged in as the enable_15 username. If you use different accounting servers for each context, tracking who was using the enable_15 username requires correlating the data from several servers.

  When configuring command authorization, consider the following:

  – An administrator with permission to use the **changeto** command effectively has permission to use all commands permitted to the enable_15 user in each of the other contexts.

  – If you intend to authorize commands differently per context, ensure that in each context the enable_15 username is denied use of commands that are also denied to administrators who are permitted use of the **changeto** command.

  When switching between security contexts, administrators can exit privileged EXEC mode and enter the **enable** command again to use the username they need.

Note The system execution space does not support AAA commands; therefore, command authorization is not available in the system execution space.

Configuring Local Command Authorization

Local command authorization places each user at a privilege level, and each user can enter any command at their privilege level or below. The FWSM lets you assign commands to one of 16 privilege levels (0 to 15). By default, each command is assigned either to privilege level 0 or 15.

This section includes the following topics:

Local Command Authorization Prerequisites

Complete the following tasks as part of your command authorization configuration:

- Configure **enable** authentication. (See the "Configuring Authentication to Access Privileged EXEC Mode" section on page 22-13.)

  Alternatively, you can use the **login** command (which is the same as the **enable** command with authentication), which requires no configuration. We do not recommend this option because it is not as secure as enable authentication.

  You can also use CLI authentication, but it is not required.

- Configure each user in the local database at a privilege level from 0 to 15. (See the "Configuring the Local Database" section on page 11-7.)

Default Command Privilege Levels

By default, the following commands are assigned to privilege level 0. All other commands are at level 15.

- **show checksum**
- **show curpriv**
- **enable** (enable mode)
- **help**
- **show history**
- **login**
- **logout**
- **pager**
- **show pager**
- **quit**
- **show version**

If you move any configure mode commands to a lower level than 15, be sure to move the **configure** command to that level as well, otherwise, the user will not be able to enter configuration mode.

To view all privilege levels, see the "Viewing Command Privilege Levels" section on page 22-18.

Assigning Privilege Levels to Commands and Enabling Authorization

To assign a command to a new privilege level and enable authorization, perform the following steps:

Step 1 To assign a command to a privilege level, enter the following command:

```
hostname(config)# privilege [show | clear | cmd] level level [mode {enable | cmd}] command
command
```

Repeat this command for each command you want to reassign.

See the following information about the options in this command:

- **show | clear | cmd**—These optional keywords let you set the privilege only for the show, clear, or configure form of the command. The configure form of the command is typically the form that causes a configuration change, either as the unmodified command (without the **show** or **clear** prefix) or as the **no** form. If you do not use one of these keywords, all forms of the command are affected.

- **level** *level*—A level between 0 and 15.
- **mode** {**enable** | **configure**}—If a command can be entered in user EXEC/privileged EXEC mode as well as configuration mode, and the command performs different actions in each mode, you can set the privilege level for these modes separately:
  - **enable**—Specifies both user EXEC mode and privileged EXEC mode.
  - **configure**—Specifies configuration mode, accessed using the **configure terminal** command.
- **command** *command*—The command you are configuring. You can only configure the privilege level of the *main* command. For example, you can configure the level of all **aaa** commands, but not the level of the **aaa authentication** command and the **aaa authorization** command separately.

  Also, you cannot configure the privilege level of commands that are in a configuration mode entered by the main command separately from the main command. For example, you can configure the **context** command, but not the **allocate-interface** command, which inherits the settings from the **context** command.

Step 2 To enable local command authorization, enter the following command:

```
hostname(config)# aaa authorization command LOCAL
```

Even if you set command privilege levels, command authorization does not take place unless you enable command authorization with this command.

For example, the **filter** command has the following forms:

- **filter** (represented by the **configure** option)
- **show running-config filter**
- **clear configure filter**

You can set the privilege level separately for each form, or set the same privilege level for all forms by omitting this option. For example, set each form separately as follows.

```
hostname(config)# privilege show level 5 command filter
hostname(config)# privilege clear level 10 command filter
hostname(config)# privilege cmd level 10 command filter
```

Alternatively, you can set all filter commands to the same level:

```
hostname(config)# privilege level 5 command filter
```

The **show privilege** command separates the forms in the display.

The following example shows the use of the **mode** keyword. The **enable** command must be entered from user EXEC mode, while the **enable password** command, which is accessible in configuration mode, requires the highest privilege level.

```
hostname(config)# privilege cmd level 0 mode enable command enable
hostname(config)# privilege cmd level 15 mode cmd command enable
hostname(config)# privilege show level 15 mode cmd command enable
```

The following example shows an additional command, the **configure** command, that uses the **mode** keyword:

```
hostname(config)# privilege show level 5 mode cmd command configure
hostname(config)# privilege clear level 15 mode cmd command configure
hostname(config)# privilege cmd level 15 mode cmd command configure
hostname(config)# privilege cmd level 15 mode enable command configure
```

Note This last line is for the **configure terminal** command.

Viewing Command Privilege Levels

The following commands let you view privilege levels for commands.

- To show all commands, enter the following command:

  ```
  hostname(config)# show running-config all privilege all
  ```

- To show commands for a specific level, enter the following command:

  ```
  hostname(config)# show running-config privilege level level
  ```

  The *level* is an integer between 0 and 15.

- To show the level of a specific command, enter the following command:

  ```
  hostname(config)# show running-config privilege command command
  ```

The following is sample output from the **show running-config all privilege all** command. The system displays the current assignment of each CLI command to a privilege level.

```
hostname(config)# show running-config all privilege all
privilege show level 15 command aaa
privilege clear level 15 command aaa
privilege configure level 15 command aaa
privilege show level 15 command aaa-server
privilege clear level 15 command aaa-server
privilege configure level 15 command aaa-server
privilege show level 15 command access-group
privilege clear level 15 command access-group
privilege configure level 15 command access-group
privilege show level 15 command access-list
privilege clear level 15 command access-list
privilege configure level 15 command access-list
privilege show level 15 command activation-key
privilege configure level 15 command activation-key
....
```

The following command displays the command assignments for privilege level 10:

```
hostname(config)# show running-config privilege level 10
privilege show level 10 command aaa
```

The following command displays the command assignment for the **access-list** command:

```
hostname(config)# show running-config privilege command access-list
privilege show level 15 command access-list
privilege clear level 15 command access-list
privilege configure level 15 command access-list
```

Configuring TACACS+ Command Authorization

If you enable TACACS+ command authorization, and a user enters a command at the CLI, the FWSM sends the command and username to the TACACS+ server to determine if the command is authorized.

When configuring command authorization with a TACACS+ server, do not save your configuration until you are sure it works the way you want. If you get locked out because of a mistake, you can usually recover access by restarting the FWSM. If you still get locked out, see the "Recovering from a Lockout" section on page 22-23.

Be sure that your TACACS+ system is completely stable and reliable. The necessary level of reliability typically requires that you have a fully redundant TACACS+ server system and fully redundant connectivity to the FWSM. For example, in your TACACS+ server pool, include one server connected to interface 1, and another to interface 2. You can also configure local command authorization as a fallback method if the TACACS+ server is unavailable. In this case, you need to configure local users and command privilege levels according to the "Configuring Command Authorization" section on page 22-14.

This section includes the following topics:

TACACS+ Command Authorization Prerequisites

Complete the following tasks as part of your command authorization configuration:

- Configure CLI authentication (see the "Configuring Authentication for CLI and ASDM Access" section on page 22-10).
- Configure **enable** authentication (see the "Configuring Authentication to Access Privileged EXEC Mode" section on page 22-13).

Configuring Commands on the TACACS+ Server

You can configure commands on a Cisco Secure Access Control Server (ACS) as a shared profile component, for a group, or for individual users. For third-party TACACS+ servers, see your server documentation for more information about command authorization support.

See the following guidelines for configuring commands in Cisco Secure ACS Version 3.1; many of these guidelines also apply to third-party servers:

- The FWSM sends the commands to be authorized as "shell" commands, so configure the commands on the TACACS+ server as shell commands.

Note Cisco Secure ACS might include a command type called "pix-shell." Do not use this type for FWSM command authorization.

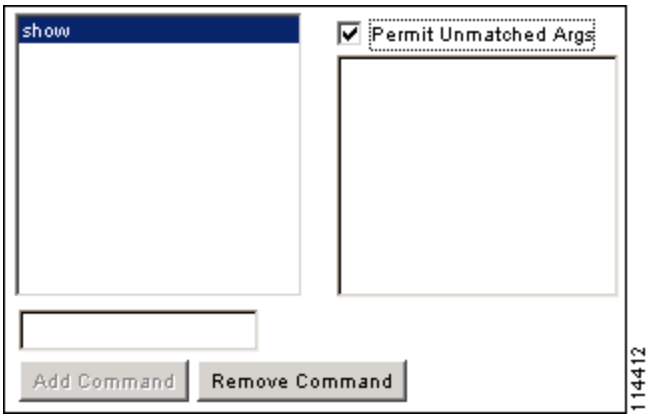
- The first word of the command is considered to be the main command. All additional words are considered to be arguments, which need to be preceded by **permit** or **deny**.

  For example, to allow the **show running-configuration aaa-server** command, add **show running-configuration** to the command field, and type **permit aaa-server** in the arguments field.

- You can permit all arguments of a command that you do not explicitly deny by checking the **Permit Unmatched Args** check box.

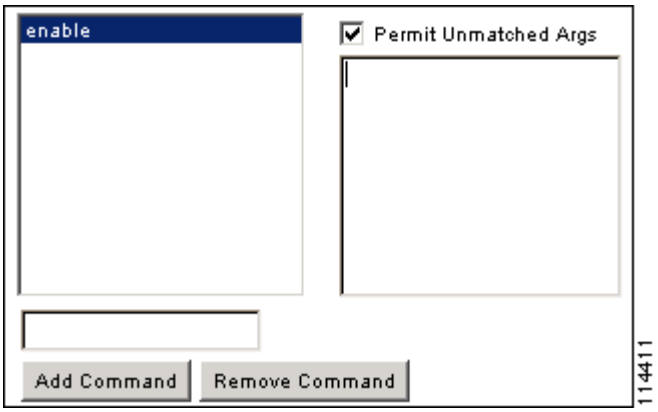
  For example, you can configure just the **show** command, and then all the **show** commands are allowed. We recommend using this method so that you do not have to anticipate every variant of a command, including abbreviations and **?**, which shows CLI usage (see Figure 22-1).

Figure 22-1 Permitting All Related Commands



- For commands that are a single word, you *must* permit unmatched arguments, even if there are no arguments for the command, for example **enable** or **help** (see Figure 22-2).

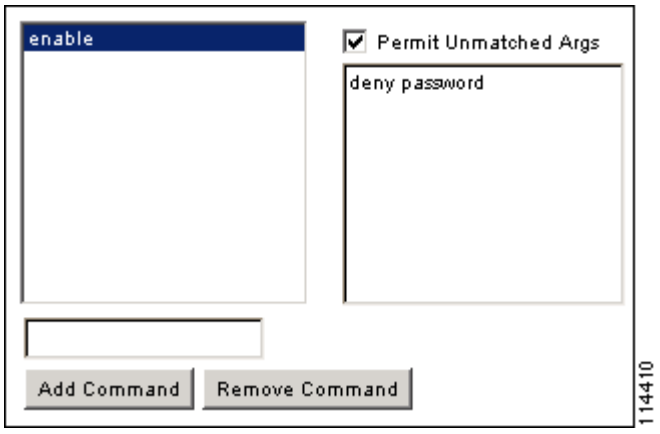
Figure 22-2 Permitting Single Word Commands



- To disallow some arguments, enter the arguments preceded by **deny**.

  For example, to allow **enable**, but not **enable password**, enter **enable** in the commands field, and **deny password** in the arguments field. Be sure to check the **Permit Unmatched Args** check box so that **enable** alone is still allowed (see Figure 22-3).

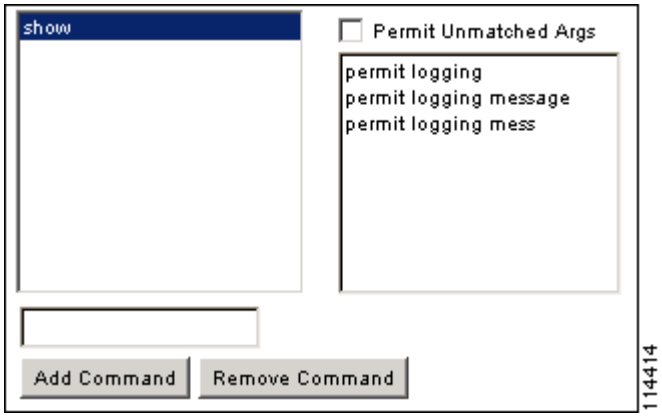
Figure 22-3 Disallowing Arguments



- When you abbreviate a command at the command line, the FWSM expands the prefix and main command to the full text, but it sends additional arguments to the TACACS+ server as you enter them.

  For example, if you enter **sh log**, then the FWSM sends the entire command to the TACACS+ server, **show logging**. However, if you enter **sh log mess**, then the FWSM sends **show logging mess** to the TACACS+ server, and not the expanded command **show logging message**. You can configure multiple spellings of the same argument to anticipate abbreviations (see Figure 22-4).

Figure 22-4 Specifying Abbreviations



- We recommend that you allow the following basic commands for all users:
  - **show checksum**
  - **show curpriv**
  - **enable**
  - **help**
  - **show history**
  - **login**
  - **logout**
  - **pager**

- **show pager**
- **quit**
- **show version**

Enabling TACACS+ Command Authorization

Before you enable TACACS+ command authorization, be sure that you are logged in to the FWSM as a user that is defined on the TACACS+ server, and that you have the necessary command authorization to continue configuring the FWSM. For example, you should log in as an admin user with all commands authorized. Otherwise, you could become unintentionally locked out.

To perform command authorization using a TACACS+ server, enter the following command:

```
hostname(config)# aaa authorization command tacacs+_server_group [LOCAL]
```

You can configure the FWSM to use the local database as a fallback method if the TACACS+ server is unavailable. To enable fallback, specify the server group name followed by **LOCAL** (**LOCAL** is case sensitive). We recommend that you use the same username and password in the local database as the TACACS+ server because the FWSM prompt does not give any indication which method is being used. Be sure to configure users in the local database (see the "Configuring the Local Database" section on page 11-7) and command privilege levels (see the "Configuring Local Command Authorization" section on page 22-15).

Configuring Command Accounting

You can send accounting messages to the TACACS+ accounting server when you enter any command other than **show** commands at the CLI. If you customize the command privilege level using the **privilege** command (see the "Assigning Privilege Levels to Commands and Enabling Authorization" section on page 22-16), you can limit which commands the FWSM accounts for by specifying a minimum privilege level. The FWSM does not account for commands that are below the minimum privilege level.

To enable command accounting, enter the following command:

```
hostname(config)# aaa accounting command [privilege level] server-tag
```

Where *level* is the minimum privilege level and *server-tag* is the name of the TACACS+ server group that to which the FWSM should send command accounting messages. The TACACS+ server group configuration must already exist. For information about configuring a AAA server group, see the "Identifying AAA Server Groups and Servers" section on page 11-9.

Viewing the Current Logged-In User

To view the current logged-in user, enter the following command:

```
hostname# show curpriv
```

See the following sample **show curpriv** command output. A description of each field follows.

```
hostname# show curpriv
Username : admin
Current privilege level : 15
Current Mode/s : P_PRIV
```

Table 22-1 describes the **show curpriv** command output.

Table 22-1 show curpriv Display Description

Field	Description
Username	Username. If you are logged in as the default user, the name is enable_1 (user EXEC) or enable_15 (privileged EXEC).
Current privilege level	Level from 0 to 15. Unless you configure local command authorization and assign commands to intermediate privilege levels, levels 0 and 15 are the only levels that are used.
Current Mode/s	Shows the access modes: • P_UNPR—User EXEC mode (levels 0 and 1) • P_PRIV—Privileged EXEC mode (levels 2 to 15) • P_CONF—Configuration mode

Recovering from a Lockout

In some circumstances, when you turn on command authorization or CLI authentication, you can be locked out of the FWSM CLI. You can usually recover access by restarting the FWSM. However, if you already saved your configuration, you might be locked out. Table 22-2 lists the common lockout conditions and how you might recover from them.

Table 22-2 CLI Authentication and Command Authorization Lockout Scenarios

Feature	Lockout Condition	Description	Workaround: Single Mode	Workaround: Multiple Mode
Local CLI authentication	No users in the local database	If you have no users in the local database, you cannot log in, and you cannot add any users.	Log in and reset the passwords and **aaa** commands.	Session in to the FWSM from the switch. From the system execution space, you can change to the context and add a user.
TACACS+ command authorization TACACS+ CLI authentication RADIUS CLI authentication	Server down or unreachable and you do not have the fallback method configured	If the server is unreachable, then you cannot log in or enter any commands.	1. Log in and reset the passwords and AAA commands. 2. Configure the local database as a fallback method so you do not get locked out when the server is down.	1. If the server is unreachable because the network configuration is incorrect on the FWSM, session in to the FWSM from the switch. From the system execution space, you can change to the context and reconfigure your network settings. 2. Configure the local database as a fallback method so you do not get locked out when the server is down.

Table 22-2 CLI Authentication and Command Authorization Lockout Scenarios (continued)

Feature	Lockout Condition	Description	Workaround: Single Mode	Workaround: Multiple Mode
TACACS+ command authorization	You are logged in as a user without enough privileges or as a user that does not exist	You enable command authorization, but then find that the user cannot enter any more commands.	Fix the TACACS+ server user account. If you do not have access to the TACACS+ server and you need to configure the FWSM immediately, then log into the maintenance partition and reset the passwords and **aaa** commands.	Session in to the FWSM from the switch. From the system execution space, you can change to the context and complete the configuration changes. You can also disable command authorization until you fix the TACACS+ configuration.
Local command authorization	You are logged in as a user without enough privileges	You enable command authorization, but then find that the user cannot enter any more commands.	Log in and reset the passwords and **aaa** commands.	Session in to the FWSM from the switch. From the system execution space, you can change to the context and change the user level.

CHAPTER 23

Managing Software, Licenses, and Configurations

This chapter describes how to install new software on the FWSM from an FTP, TFTP, HTTP, or HTTPS server. You can upgrade the application software, the maintenance software, and ASDM management software. You can also enable Auto Update support. This chapter includes the following sections:

Note Because the FWSM runs its own operating system, upgrading the Cisco IOS software does not affect the operation of the FWSM.

Managing Licenses

When you install the software, the existing activation key is extracted from the original image and stored in a file in the FWSM file system. This section includes the following topics:

Obtaining an Activation Key

To obtain an activation key, you will need a Product Authorization Key, which you can purchase from your Cisco account representative. After obtaining the Product Authorization Key, register it on the Cisco website to obtain an activation key by performing the following steps:

Step 1 Obtain the serial number for your FWSM by entering the following command:

```
hostname> show version | include Number
```

Enter the pipe character (|) as part of the command.

Step 2 Connect a web browser to one of the following websites (the URLs are case-sensitive):

Use the following website if you are a registered user of Cisco.com:

```
http://www.cisco.com/go/license
```

Use the following website if you are not a registered user of Cisco.com:

```
http://www.cisco.com/go/license/public
```

Step 3 When prompted, enter the following information:

- Your Product Authorization Key
- The serial number of your FWSM.
- Your e-mail address.

The activation key will be automatically generated and sent to the e-mail address that you provide.

Entering a New Activation Key

To enter the activation key, enter the following command:

```
hostname(config)# activation-key key
```

The key is a four-element hexadecimal string with one space between each element. For example, a key in the correct form might look like the following key:

0xe02888da 0x4ba7bed6 0xf1c123ae 0xffd8624e

The leading 0x specifier is optional; all values are assumed to be hexadecimal.

If you are already in multiple context mode, enter this command in the system execution space.

Note The activation key is not stored in your configuration file. The key is tied to the serial number of the device.

This example shows how to change the activation key on the FWSM:

```
hostname(config)# activation-key 0xe02888da 0x4ba7bed6 0xf1c123ae 0xffd8624e
```

Entering Activation Keys in a Failover Pair

To enter activation keys in a failover configuration, perform the following steps:

Step 1 Disable failover by entering the **no failover** command on the active FWSM:

```
hostname(config)# no failover
```

The active FWSM remains active, and the standby FWSM moves to a pseudo-standby state.

Note Disabling failover does not affect transient traffic.

Step 2 Apply different activation keys to the active FWSM and to the standby FWSM. Each device has his own unique key that is tied to the serial number on the FWSM.

Step 3 Re-enable failover by entering the **failover** command on the active FWSM:

```
hostname(config)# failover
```

Entering the command on the active FWSM re-enables failover between both units and brings up the failover pair.

Installing Application or ASDM Software

This section contains the following topics:

Installation Overview

For application software, you can use one of two methods to upgrade:

- Installing to the current application partition from the FWSM CLI

  The benefit of this method is you do not have to boot in to the maintenance partition; instead you log in as usual and copy the new software.

  This method supports downloading from a TFTP, FTP, HTTP, or HTTPS server.

  You cannot copy software to the other application partition. You might want to copy to the other partition if you want to keep the old version of software as a backup in the current partition.

  You must have an operational configuration with network access. For multiple context mode, you need to have network connectivity through the admin context.

- Installing to any application partition from the maintenance partition

  The benefit of this method is you can copy software to both application partitions, and you do not have to have an operational configuration. You just need to configure some routing parameters in the maintenance partition so you can reach the server on VLAN 1.

  The disadvantage is that you need to boot in to the maintenance partition, which might not be convenient if you have an operational application partition.

  This method supports downloading from an FTP server only.

To upgrade ASDM, you can only install to the current application partition from the FWSM CLI.

See the "Managing the Firewall Services Module Boot Partitions" section on page 2-10 for more information about application and maintenance partitions.

Installing Application Software from the FWSM CLI

When you log in to the FWSM during normal operation, you can copy the application software to the current application partition from a TFTP, FTP, HTTP, or HTTPS server.

For multiple context mode, you must be in the system execution space.

To upgrade software to the current application partition from an FTP, TFTP, or HTTP(S) server, perform the following steps:

Step 1 Enter the following command to confirm access to the selected FTP, TFTP, or HTTP(S) server:

```
hostname# ping ip_address
```

Step 2 To copy the application software, enter one of the following commands, directed to the appropriate download server.

- To copy from a TFTP server, enter the following command:

  ```
  hostname# copy tftp://server[/path]/filename flash:
  ```

  The **flash** keyword refers to the application partition on the FWSM. You can only copy an image and ASDM software to the **flash** partition. Configuration files are copied to the **disk** partition.

- To copy from an FTP server, enter the following command:

  ```
  hostname# copy ftp://[user[:password]@]server[/path]/filename flash:
  ```

- To copy from an HTTP or HTTPS server, enter the following command:

  ```
  hostname# copy http[s]://[user[:password]@]server[:port][/path]/filename flash:
  ```

- To use secure copy, first enable SSH, then enter the following command:

  ```
  hostname# ssh scopy enable
  ```

  Then from a Linux client, enter the following command:

  ```
  scp filename username@fwsm_address:disk:
  ```

For example, to copy the application software from an FTP server, enter the following command:

```
hostname# copy ftp://10.94.146.80/tftpboot/user1/cdisk flash:

copying ftp://10.94.146.80/tftpboot/user1/cdisk to flash:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!
Received 6128128 bytes.
Erasing current image.This may take some time..
Writing 6127616 bytes of image.
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!
Image installed.
```

Step 3 To run the new software, you need to reload the system. If you do not have a failover pair, enter the following command:

```
hostname# reload
Proceed with reload? [confirm]
```

At the "Proceed with reload?" prompt, press **Enter** to confirm the command.

```
Rebooting...
```

If you have a failover pair, see the "Upgrading Failover Pairs" section on page 23-9.

Installing Application Software from the Maintenance Partition

If you log in to the maintenance partition, you can install application software to either application partition (cf:4 or cf:5).

Note The FWSM maintenance partition can only use VLAN 1 on the switch. The FWSM does not support 802.1Q tagging on VLAN 1. Using an address in the 10.3.1.0/24 subnet for the maintenance partition IP address can cause communication problems with other hosts on that subnet; the FWSM uses 10.3.1.1 for internal diagnostics.

You must use maintenance software Release 2.1(2) or later with the FWSM. See the "Installing Maintenance Software" section on page 23-12 to upgrade.

Note **If upgrading to a new maintenance release using failover:** If you have an Active/Standby failover pair, first perform this procedure on the standby unit; after the standby unit reloads, force the active unit to fail over to the standby unit using the **no failover active** command in the system execution space of the active unit; then upgrade the active unit.

For Active/Active failover, make both failover groups active on the primary unit by entering the **failover active** command in the system execution space of the primary unit. Then perform this procedure on the secondary unit. After you complete the upgrade procedure for the secondary unit, make both failover groups active on the secondary unit using the **no failover active** command in the system execution space of the primary unit. Then upgrade the active unit.

If upgrading to a minor or major release using failover: If you have a failover pair, upgrade the

primary unit first, but then be sure to start the upgrade on the secondary unit before the primary unit comes online with the new version. If both units are running, and the major version number does not match (3.1 vs. 3.2), then both units become active. Two active units can cause networking problems.

To install application software from an FTP server while logged in to the maintenance partition, perform the following steps:

Step 1 Each application partition has its own startup configuration, so you need to make the current configuration available to copy to the backup application partition, if desired. You can either copy it to an available TFTP, FTP, or HTTP(S) server, or you can enter the **show running-config** command and cut and paste the configuration from the terminal.

Step 2 If necessary, end the FWSM session by entering the following command:

```
hostname# exit

Logoff

[Connection to 127.0.0.31 closed by foreign host]
Router#
```

You might need to enter the **exit** command multiple times if you are in a configuration mode.

Step 3 To view the current boot partition, enter the command for your operating system. Note the current boot partition so you can set a new default boot partition.

- Cisco IOS software

  ```
  Router# show boot device [mod_num]
  ```

  For example:

  ```
  Router# show boot device
  [mod:1 ]:
  [mod:2 ]:
  [mod:3 ]:
  [mod:4 ]: cf:4
  [mod:5 ]: cf:4
  [mod:6 ]:
  [mod:7 ]: cf:4
  [mod:8 ]:
  [mod:9 ]:
  ```

- Catalyst operating system software

  ```
  Console> (enable) show boot device mod_num
  ```

  For example:

  ```
  Console> (enable) show boot device 4
  Device BOOT variable = cf:4
  ```

Step 4 To change the default boot partition to the backup, enter the command for your operating system:

- Cisco IOS software

  ```
  Router(config)# boot device module mod_num cf:{4 | 5}
  ```

- Catalyst operating system software

  ```
  Console> (enable) set boot device cf:{4 | 5} mod_num
  ```

Step 5 To boot the FWSM into the maintenance partition, enter the command for your operating system at the switch prompt:

- For Cisco IOS software, enter the following command:

```
Router# hw-module module mod_num reset cf:1
```

- For Catalyst operating system software, enter the following command:

```
Console> (enable) reset mod_num cf:1
```

Step 6 To session in to the FWSM, enter the command for your operating system:

- Cisco IOS software

```
Router# session slot number processor 1
```

- Catalyst operating system software

```
Console> (enable) session module_number
```

Step 7 To log in to the FWSM maintenance partition as root, enter the following command:

```
Login: root
Password:
```

By default, the password is **cisco**.

Step 8 To set network parameters, perform the following steps:

a. To assign an IP address to the maintenance partition, enter the following command:

```
root@localhost# ip address ip_address netmask
```

This address is the address for VLAN 1, which is the only VLAN used by the maintenance partition. Using an address in the 10.3.1.0/24 subnet for the maintenance partition IP address can cause communication problems with other hosts on that subnet; the FWSM uses 10.3.1.1 for internal diagnostics.

b. To assign a default gateway to the maintenance partition, enter the following command:

```
root@localhost# ip gateway ip_address
```

c. (Optional) To ping the FTP server to verify connectivity, enter the following command:

```
root@localhost# ping ftp_address
```

Step 9 To download the application software from the FTP server, enter the following command:

```
root@localhost# upgrade ftp://[user[:password]@]server[/path]/filename cf:{4 | 5}
```

cf:4 and **cf:5** are the application partitions on the FWSM. Install the new software to the backup partition.

Follow the screen prompts during the upgrade.

Step 10 To log out of the maintenance partition, enter the following command:

```
root@localhost# logout
```

Step 11 To reboot the FWSM into the backup application partition (that you set as the default in Step 4), enter the command for your operating system:

- For Cisco IOS software, enter the following command:

```
Router# hw-module module mod_num reset
```

- For Catalyst operating system software, enter the following command:

```
Console> (enable) reset mod_num
```

Step 12 To session in to the FWSM, enter the command for your operating system:

- Cisco IOS software

```
Router# session slot number processor 1
```

- Catalyst operating system software

```
Console> (enable) session module_number
```

By default, the password to log in to the FWSM is **cisco** (set by the **password** command). If this partition does not have a startup configuration, the default password is used.

Step 13 Enter privileged EXEC mode using the following command:

```
hostname> enable
```

The default password is blank (set by the **enable password** command). If this partition does not have a startup configuration, the default password is used.

Step 14 Each application partition has its own startup configuration, so you might need to copy a current configuration to the application partition. If you have an old configuration running on this partition, you might want to clear it before copying to the running configuration. To clear the running configuration, enter the **clear configure all** command. To copy the configuration to the running configuration, use one of the following methods:

- Paste the configuration at the command line.
- To copy from a TFTP server, enter the following command:

```
hostname# copy tftp://server[/path]/filename running-config
```

- To copy from an FTP server, enter the following command:

```
hostname# copy ftp://[user[:password]@]server[/path]/filename running-config
```

- To copy from an HTTP or HTTPS server, enter the following command:

```
hostname# copy http[s]://[user[:password]@]server[:port][/path]/filename
running-config
```

- To copy from the local flash memory, enter the following command:

```
hostname# copy disk:[path/]filename running-config
```

Step 15 Save the running configuration to the startup configuration using the following command:

```
hostname# write memory
```

Step 16 The default context mode is single mode, so if you are running in multiple context mode, set the mode to multiple in the new application partition using the following command:

```
hostname# configuration terminal
hostname(config)# mode multiple
WARNING: This command will change the behavior of the device
WARNING: This command will initiate a Reboot
Proceed with change mode? [confirm]
```

Confirm to reload the FWSM.

Installing ASDM from the FWSM CLI

When you log in to the FWSM during normal operation, you can copy ASDM software to the current application partition from a TFTP, FTP, HTTP, or HTTPS server.

For multiple context mode, you must be in the system execution space.

To check connectivity, use the **ping** command.

To copy ASDM software, enter one of the following commands for the appropriate download server:

- To copy from a TFTP server, enter the following command:

  ```
  hostname# copy tftp://server[/path]/filename flash:asdm
  ```

  The **flash** keyword represents the application partition on the FWSM. You can only copy an image and ASDM software to the **flash** partition. Configuration files are copied to the **disk** partition.

- To copy from an FTP server, enter the following command:

  ```
  hostname# copy ftp://[user[:password]@]server[/path]/filename flash:asdm
  ```

- To copy from an HTTP or HTTPS server, enter the following command:

  ```
  hostname# copy http[s]://[user[:password]@]server[:port][/path]/filename flash:asdm
  ```

- To use secure copy, first enable SSH, and then enter the following command:

  ```
  hostname# ssh scopy enable
  ```

  Then from a Linux client, enter the following command:

  ```
  scp -v -pw password filename username@fwsm_address
  ```

  Where **-v** is for verbose, and if **-pw** is not specified, you will be prompted for a password.

For example, to copy ASDM from a TFTP server, enter:

```
hostname# copy tftp://209.165.200.226/cisco/asdm.bin flash:asdm
```

To copy to the ASDM from an HTTPS server, enter:

```
hostname# copy http://admin:letmein@209.165.200.228/adsm/asdm.bin flash:asdm
```

Upgrading Failover Pairs

The two units in a failover configuration should have the same major (first number), minor (second number), and maintenance (third number) software version.

However, you can have different *maintenance* versions of the software running on each unit and still maintain failover support. You can upgrade from any maintenance release to any other maintenance release within a minor release. For example, you can upgrade from 3.1(1) to 3.1(3) without first installing the maintenance releases in between.

To ensure long-term compatibility and stability, we recommend upgrading both units to the same version as soon as possible.

The FWSM does not support upgrading from between major or minor releases, for example, from 2.3 to 3.1, without downtime.

Note To upgrade failover pairs from the maintenance partition, see the "Installing Application Software from the Maintenance Partition" section on page 23-5.

This section includes the following topics:

- Upgrading Failover Pairs to a New Maintenance Release, page 23-10
- Upgrading Failover Pairs to a New Minor or Major Release, page 23-12

Upgrading Failover Pairs to a New Maintenance Release

You can upgrade from any maintenance release to any other maintenance release within a minor release without downtime.

For example, you can upgrade from 3.1(1) to 3.1(3) without first installing the maintenance releases in between.

This section includes the following topics:

- Upgrading an Active/Standby Failover Pair to a New Maintenance Release, page 23-10
- Upgrading an Active/Active Failover Pair to a New Maintenance Release, page 23-11

Upgrading an Active/Standby Failover Pair to a New Maintenance Release

To upgrade two units in an Active/Standby failover configuration to a new maintenance release, perform the following steps.

Step 1 Download the new software to both units. See the "Installing Application Software from the FWSM CLI" section on page 23-4.

Step 2 Ensure that the secondary unit has a configuration saved to memory by entering the following command:

```
secondary(config)# write memory
```

The saved configuration will load when you restart the secondary unit. This step is useful if the primary unit fails to start up correctly.

In multiple context mode, enter the **write memory all** command from the system execution space. This command saves all context configurations to which the FWSM has write access.

Step 3 Reload the standby unit to boot the new image by entering the following command on the active unit:

```
primary# failover reload-standby
```

Step 4 When the standby unit has finished reloading, and is in the Standby Ready state, force the active unit to fail over to the standby unit by entering the following command on the active unit.

Note Use the **show failover** command to verify that the standby unit is in the Standby Ready state.

```
primary# no failover active
```

Step 5 Reload the former active unit (now the new standby unit) by entering the following command:

```
primary# reload
```

Step 6 (Optional) When the new standby unit has finished reloading, and is in the Standby Ready state, return the original active unit to active status by entering the following command:

```
primary# failover active
```

Upgrading an Active/Active Failover Pair to a New Maintenance Release

To upgrade two units in an Active/Active failover configuration to a new maintenance release, perform the following steps.

Step 1 Download the new software to both units. See the "Installing Application Software from the FWSM CLI" section on page 23-4.

Step 2 Ensure that the secondary unit has a configuration saved to memory by entering the following command:

```
secondary(config)# write memory
```

The saved configuration will load when you restart the secondary unit. This step is useful if the primary unit fails to start up correctly.

In multiple context mode, enter the **write memory all** command from the system execution space. This command saves all context configurations to which the FWSM has write access.

Step 3 Make both failover groups active on the primary unit by entering the following command in the system execution space of the primary unit:

```
primary# failover active
```

Step 4 Reload the secondary unit to boot the new image by entering the following command in the system execution space of the primary unit:

```
primary# failover reload-standby
```

Step 5 When the secondary unit has finished reloading, and both failover groups are in the Standby Ready state on that unit, make both failover groups active on the secondary unit using the following command in the system execution space of the primary unit:

```
primary# no failover active
```

Note Use the **show failover** command to verify that both failover groups are in the Standby Ready state on the secondary unit.

Step 6 Make sure both failover groups are in the Standby Ready state on the primary unit, and then reload the primary unit using the following command:

```
primary# reload
```

If the failover groups are configured with the **preempt** command, they will automatically become active on their designated unit after the preempt delay has passed. If the failover groups are not configured with the **preempt** command, you can return them to active status on their designated units using the **failover active group** command.

Upgrading Failover Pairs to a New Minor or Major Release

To upgrade two units in an Active/Active or Active/Standby failover configuration to a new minor or major release, perform the following steps.

Step 1 Download the new software to both units. See the "Installing Application Software from the FWSM CLI" section on page 23-4.

Step 2 Ensure that the secondary unit has a configuration saved to memory by entering the following command:

```
secondary(config)# write memory
```

The saved configuration will load when you restart the secondary unit. This step is useful if the primary unit fails to start up correctly.

In multiple context mode, enter the **write memory all** command from the system execution space. This command saves all context configurations to which the FWSM has write access.

Step 3 To load the new software, reload the primary unit and then reload the secondary unit before the primary unit comes online. Enter the following command separately on each unit:

```
primary(config)# reload
Proceed with reload? [confirm]
```

At the "Proceed with reload?" prompt, press **Enter** to confirm the command.

```
Rebooting...

secondary(config)# reload
Proceed with reload? [confirm]
```

While the units reload, all active connections are terminated. We recommend reloading both units at the same time because if both units are running, and the major or minor version number does not match (3.1 vs. 3.2), then both units become active. Two active units can cause networking problems.

Installing Maintenance Software

You must install maintenance software Release 2.1(2) or later before you upgrade to FWSM Release 4.0. This section includes the following topics:

Checking the Maintenance Software Release

To determine the maintenance software release, you must boot in to the maintenance partition and view the release by performing the following steps:

Step 1 If necessary, end the FWSM session by entering the following command:

```
hostname# exit

Logoff
```

```
[Connection to 127.0.0.31 closed by foreign host]
Router#
```

You might need to enter the **exit** command multiple times if you are in a configuration mode.

Step 2 To boot the FWSM into the maintenance partition, enter the command for your operating system at the switch prompt:

- For Cisco IOS software, enter the following command:

  ```
  Router# hw-module module mod_num reset cf:1
  ```

- For Catalyst operating system software, enter the following command:

  ```
  Console> (enable) reset mod_num cf:1
  ```

Step 3 To session in to the FWSM, enter the command for your operating system:

- Cisco IOS software

  ```
  Router# session slot number processor 1
  ```

- Catalyst operating system software

  ```
  Console> (enable) session module_number
  ```

Step 4 To log in to the FWSM maintenance partition as root, enter the following command:

```
Login: root

Password:
```

By default, the password is **cisco**.

The FWSM shows the version when you first log in:

```
Maintenance image version: 2.1(2)
```

Step 5 To view the maintenance version after you log in, enter the following command:

```
root@localhost# show version

Maintenance image version: 3.2(1)
mp.3-2-1.bin : Thu Nov 18 11:41:36 PST 2007 : integ@kplus-build-lx.cisco.com

Line Card Number :WS-SVC-FWM-1
Number of Pentium-class Processors :       2
BIOS Vendor: Phoenix Technologies Ltd.
BIOS Version: 4.0-Rel 6.0.9
Total available memory: 1004 MB
Size of compact flash: 123 MB
Daughter Card Info: Number of DC Processors: 3
Size of DC Processor Memory (per proc): 32 MB
```

Upgrading the Maintenance Software

If you need to upgrade the maintenance software, perform the following steps:

Step 1 Download the maintenance software from Cisco.com at the following URL:

http://www.cisco.com/cgi-bin/tablebuild.pl/cat6000-serv-maint

Put the software on a TFTP, HTTP, or HTTPS server that is accessible from the FWSM admin context.

Step 2 If required, log out of the maintenance partition and reload the application partition by performing the following steps:

a. Log out of the maintenance partition by entering the following command:

```
root@localhost# logout
```

b. If required, reboot the FWSM into the application partition by entering the command for your operating system:

- For Cisco IOS software, enter the following command:

```
Router# hw-module module mod_num reset
```

- For Catalyst operating system software, enter the following command:

```
Console> (enable) reset mod_num
```

c. To session in to the FWSM, enter the command for your operating system:

- Cisco IOS software

```
Router# session slot number processor 1
```

- Catalyst operating system software

```
Console> (enable) session module_number
```

Step 3 To upgrade the maintenance partition software, enter one of the following commands for the appropriate download server.

For multiple context mode, you must be in the system execution space.

- To download the maintenance software from a TFTP server, enter the following command:

```
hostname# upgrade-mp tftp[://server[:port][/path]/filename]
```

You are prompted to confirm the server information, or if you do not supply it in the command, you can enter it at the prompts.

- To download the maintenance software from an HTTP or HTTPS server, enter the following command:

```
hostname# upgrade-mp http[s]://[user[:password]@]server[:port][/path]/filename
```

Passwords for the root and guest accounts of the maintenance partition are retained after the upgrade.

Step 4 Reload the FWSM to load the new maintenance software by entering the following command:

```
hostname# reload
```

Alternatively, you can log out of the FWSM in preparation for booting in to the maintenance partition; from the maintenance partition, you can install application software to both application partitions. To end the FWSM session, enter the following command:

```
hostname# exit

Logoff

[Connection to 127.0.0.31 closed by foreign host]
Router#
```

You might need to enter the **exit** command multiple times if you are in a configuration mode.

See the "Installing Application Software from the Maintenance Partition" section on page 23-5 to reload the FWSM into the maintenance partition.

The following example shows the prompts for the TFTP server information:

```
hostname# upgrade-mp tftp
Address or name of remote host [127.0.0.1]? 10.1.1.5
Source file name [cdisk]? mp.3-2-1-3.bin.gz
copying tftp://10.1.1.5/mp.3-2-1-3.bin.gz to flash
[yes|no|again]? yes
!!!!!!!!!!!!!!!!!!!!!!
Received 1695744 bytes.
Maintenance partition upgraded.
```

Downloading and Backing Up Configuration Files

This section describes how to download and back up configuration files, and includes the following sections:

Viewing Files in Flash Memory

You can view files in flash memory and see information about the files.

- To view the files in Flash memory, enter the following command:

  ```
  hostname# dir disk:
  ```

  For example:

  ```
  hostname# dir

  Directory of disk:/

  9      -rw-  1411        08:53:42 Oct 06 2005  old_running.cfg
  10     -rw-  959         09:21:50 Oct 06 2005  admin.cfg
  11     -rw-  1929        08:23:44 May 07 2005  admin_backup.cfg
  ```

- To view extended information about a specific file, enter the following command:

  ```
  hostname# show file information [path:/]filename
  ```

  The default path is the root directory of the internal flash memory (disk:/).

  For example:

  ```
  hostname# show file info admin.cfg

  disk:/admin.cfg:
    type is ascii text
    file size is 959 bytes
  ```

Downloading a Text Configuration to the Startup or Running Configuration

You can download a text file from the following server types to the single mode configuration or the multiple mode system configuration:

- TFTP
- FTP
- HTTP
- HTTPS

For a multiple mode context, see the "Downloading a Context Configuration to Disk" section on page 23-17.

Note When you copy a configuration to the running configuration, you merge the two configurations. A merge adds any new commands from the new configuration to the running configuration. If the configurations are the same, no changes occur. If commands conflict or if commands affect the running of the context, then the effect of the merge depends on the command. You might get errors, or you might have unexpected results.

To copy the startup configuration or running configuration from the server to the FWSM, enter one of the following commands for the appropriate download server:

- To copy from a TFTP server, enter the following command:

  ```
  hostname# copy tftp://server[/path]/filename {startup-config | running-config}
  ```

- To copy from an FTP server, enter the following command:

  ```
  hostname# copy ftp://[user[:password]@]server[/path]/filename[;type=xx]
  {startup-config | running-config}
  ```

  The **type** can be one of the following keywords:

  - **ap**—ASCII passive mode
  - **an**—ASCII normal mode
  - **ip**—(Default) Binary passive mode
  - **in**—Binary normal mode

  You can use ASCII or binary for configuration files.

- To copy from an HTTP or HTTPS server, enter the following command:

  ```
  hostname# copy http[s]://[user[:password]@]server[:port][/path]/filename
  {startup-config | running-config}
  ```

For example, to copy the configuration from a TFTP server, enter the following command:

```
hostname# copy tftp://209.165.200.226/configs/startup.cfg startup-config
```

To copy the configuration from an FTP server, enter the following command:

```
hostname# copy ftp://admin:letmein@209.165.200.227/configs/startup.cfg;type=an
startup-config
```

To copy the configuration from an HTTP server, enter the following command:

```
hostname# copy http://209.165.200.228/configs/startup.cfg startup-config
```

Downloading a Context Configuration to Disk

To copy context configurations to disk, including the admin configuration, enter one of the following commands for the appropriate download server from the system execution space:

- To copy from a TFTP server, enter the following command:

  ```
  hostname# copy tftp://server[/path]/filename disk:[path/]filename
  ```

- To copy from a FTP server, enter the following command:

  ```
  hostname# copy ftp://[user[:password]@]server[/path]/filename disk:[path/]filename
  ```

- To copy from an HTTP or HTTPS server, enter the following command:

  ```
  hostname# copy http[s]://[user[:password]@]server[:port][/path]/filename
  disk:[path/]filename
  ```

Backing Up the Configuration

To back up your configuration, use one of the following methods:

- Backing up the Single Mode Configuration or Multiple Mode System Configuration, page 23-17
- Backing Up a Context Configuration in Flash Memory, page 23-18
- Backing Up a Context Configuration within a Context, page 23-18
- Copying the Configuration from the Terminal Display, page 23-18

Backing up the Single Mode Configuration or Multiple Mode System Configuration

In single context mode or from the system configuration in multiple mode, you can copy the startup configuration or running configuration to an external server or to the local flash memory:

- To copy to a TFTP server, enter the following command:

  ```
  hostname# copy {startup-config | running-config} tftp://server[/path]/filename
  ```

- To copy to a FTP server, enter the following command:

  ```
  hostname# copy {startup-config | running-config}
  ftp://[user[:password]@]server[/path]/filename
  ```

- To copy to local flash memory, enter the following command:

  ```
  hostname# copy {startup-config | running-config} disk:[path/]filename
  ```

  Be sure the destination directory exists. If it does not exist, first create the directory using the **mkdir** command.

Backing Up a Context Configuration in Flash Memory

In multiple context mode, copy context configurations that are on the local flash memory by entering one of the following commands in the system execution space:

- To copy to a TFTP server, enter the following command:

  ```
  hostname# copy disk:[path/]filename tftp://server[/path]/filename
  ```

- To copy to a FTP server, enter the following command:

  ```
  hostname# copy disk:[path/]filename ftp://[user[:password]@]server[/path]/filename
  ```

- To copy to local flash memory, enter the following command:

  ```
  hostname# copy disk:[path/]filename disk:[path/]newfilename
  ```

  Be sure that the destination directory exists. If it does not, create the directory using the **mkdir** command.

Backing Up a Context Configuration within a Context

In multiple context mode, from within a context, you can perform the following backups:

- To copy the running configuration to the startup configuration server (connected to the admin context), enter the following command:

  ```
  hostname/contexta# copy running-config startup-config
  ```

- To copy the running configuration to a TFTP server connected to the context network, enter the following command:

  ```
  hostname/contexta# copy running-config tftp:/server[/path]/filename
  ```

Copying the Configuration from the Terminal Display

To print the configuration to the terminal, enter the following command:

```
hostname# show running-config
```

Copy the output from this command, and then paste the configuration in to a text file.

Configuring Auto Update Support

Auto Update is a protocol specification that allows an Auto Update Server to download configurations and software images to many FWSMs, and can provide basic monitoring of the FWSMs from a central location. The FWSM periodically polls the Auto Update Server for updates to software images and configuration files.

Note Auto Update is supported in single context mode only.

This section includes the following topics:

Configuring Communication with an Auto Update Server

To configure an Auto Update Server, perform the following steps:

Step 1 To specify the URL of the AUS, use the following command:

```
hostname(config)# auto-update server url [source interface] [verify-certificate]
```

Where *url* has the following syntax:

```
http[s]://[user:password@]server_ip[:port]/pathname
```

You can configure only one server. SSL is used when **https** is specified. The *user* and *password* arguments of the URL are used for basic authentication when logging in to the server. If you use the **write terminal**, **show configuration** or **show tech-support** commands to view the configuration, the user and password are replaced with "********".

The default port is 80 for HTTP and 443 for HTTPS.

The **source** *interface* argument specifies which interface to use when sending requests to the AUS. If you specify the same interface specified by the **management-access** command, the Auto Update Server requests travel over the same IPSec VPN tunnel used for management access.

The **verify-certificate** keyword verifies the certificate returned by the AUS.

Step 2 (Optional) To identify the device ID to send when communicating with the AUS, enter the following command:

```
hostname(config)# auto-update device-id {hardware-serial | hostname | ipaddress [if-name]
| mac-address [if-name] | string text}
```

The device ID used is determined according to one of the following parameters:

- **hardware-serial**—Use the FWSM serial number.
- **hostname**—Use the FWSM hostname.
- **ipaddress**—Use the IP address of the specified interface. If the interface name is not specified, the device uses the IP address of the interface used to communicate with the AUS.
- **mac-address**—Use the MAC address of the specified interface. If the interface name is not specified, the device uses the MAC address of the interface used to communicate with the AUS.
- **string**—Use the specified text identifier, which cannot contain white space or the characters ‘, “, , >, & and ?.

Step 3 (Optional) To specify how often to poll the AUS for configuration or image updates, enter the following command:

```
hostname(config)# auto-update poll-period poll-period [retry-count [retry-period]]
```

The *poll-period* argument specifies how often (in minutes) to check for an update. The default is 720 minutes (12 hours).

The *retry-count* argument specifies how many times to try reconnecting to the server if the first attempt fails. The default is 0.

The *retry-period* argument specifies how long to wait (in minutes) between retries. The default is 5.

Step 4 (Optional) If the Auto Update Server has not been contacted for a certain period of time, the following command will cause it to cease passing traffic:

```
hostname(config)# auto-update timeout period
```

Where *period* specifies the timeout period in minutes between 1 and 35791. The default is to never time out (0). To restore the default, enter the **no** form of this command.

Use this command to ensure that the FWSM has the most recent image and configuration. This condition is reported with system log message 201008.

In the following example, a FWSM is configured to poll an AUS with IP address 209.165.200.224, at port number 1742, from the outside interface, with certificate verification.

The FWSM is also configured to use the hostname of the FWSM as the device ID, and the polling period has been decreased from the default of 720 minutes to 600 minutes. On a failed polling attempt, the FWSM will try to reconnect to the AUS 10 times, and wait 3 minutes between attempts at reconnecting.

```
hostname(config)# auto-update server
https://jcrichton:farscape@209.165.200.224:1742/management source outside
verify-certificate
hostname(config)# auto-update device-id hostname
hostname(config)# auto-update poll-period 600 10 3
```

Viewing Auto Update Server Status

To view the Auto Update Server status, enter the following command:

```
hostname(config)# show auto-update
```

The following is sample output from the **show auto-update** command:

```
hostname(config)# show auto-update
Server: https://********@209.165.200.224:1742/management.cgi?1276
Certificate will be verified
Poll period: 720 minutes, retry count: 2, retry period: 5 minutes
Timeout: none
Device ID: host name [corporate]
Next poll in 4.93 minutes
Last poll: 11:36:46 PST Tue Nov 13 2004
Last PDM update: 23:36:46 PST Tue Nov 12 2004
```

CHAPTER 24

Monitoring the Firewall Services Module

This chapter describes how to configure logging and SNMP for the FWSM. It also describes the contents of syslog messages and the syslog message format.

This chapter does not provide comprehensive information about all monitoring, logging, and SNMP commands and options. For detailed descriptions and additional commands, see the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

This chapter includes the following sections:

Configuring and Managing Syslog Messages

This section describes the logging functionality and configuration. It also describes the syslog message format, options, and variables. This section includes the following topics:

Logging Overview

The FWSM supports the generation of an audit trail of syslog messages that describe its activities (for example, what kinds of network traffic has been allowed and denied) and enables you to configure system logging.

All syslog messages have a default severity level. You can reassign a message to a new severity level, if necessary. When you choose a severity level, logging messages from that level and lower levels are generated. Messages from a higher level are not included. The higher the severity level, the more messages are included. For more information about logging and syslog messages, see *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module System Log Messages*.

The FWSM syslog messages provide you with information for monitoring and troubleshooting the FWSM. Using the logging feature, you can do the following:

- Specify which syslog messages should be logged.
- Disable or change the severity level of a syslog message.
- Specify the severity level of a syslog message by color.
- Display a brief description of the syslog message as a tooltip.
- Specify explanations and recommended actions for a syslog message.
- Specify one or more locations to which syslog messages should be sent, including an internal buffer, one or more syslog servers, an SNMP management station, specified e-mail addresses, or Telnet and SSH sessions.
- Configure and manage syslog messages in groups, such as by severity level or class of message.
- Specify what happens to the contents of the internal buffer when the buffer becomes full: overwrite the buffer, send the buffer contents to an FTP server, or save the contents to internal flash memory.
- Send all syslog messages, or subsets of syslog messages, to any or all output locations.
- Filter which syslog messages are sent to which locations by the severity of the syslog message, the class of the syslog message, or by creating a custom log message list.

Security Contexts and Logging

Each security context includes its own logging configuration and generates its own messages. If you log in to the system or admin context, and then change to another context, messages you view in your session are only those that are related to the current context.

Syslog messages that are generated in the system execution space, including failover messages, are viewed in the admin context along with messages generated in the admin context. You cannot configure logging or view any logging information in the system execution space.

You can configure the FWSM to include the context name with each message, which helps you differentiate context messages that are sent to a single syslog server. This feature also helps you to determine which messages are from the admin context and which are from the system; messages that originate in the system execution space use a device ID of **system**, and messages that originate in the admin context use the name of the admin context as the device ID. For more information about enabling logging device IDs, see the "Including the Device ID in Syslog Messages" section on page 24-15.

Enabling and Disabling Logging

This section describes how to enable and disable logging on the FWSM. It includes the following topics:

Enabling Logging to All Configured Output Destinations

The following command enables logging; however, you must also specify at least one output destination so that you can view or save the logged messages. If you do not specify an output destination, the FWSM does not save syslog messages that are generated when events occur.

For more information about configuring log output destinations, see the "Configuring Log Output Destinations" section on page 24-3.

To enable logging, enter the following command:

```
hostname(config)# logging enable
```

Disabling Logging to All Configured Output Destinations

To disable all logging to all configured log output destinations, enter the following command:

```
hostname(config)# no logging enable
```

Viewing the Log Configuration

To view the running log configuration, enter the following command:

```
hostname(config)# show logging
```

The following is sample output of the **show logging** command:

```
Syslog logging: enabled
    Facility: 16
    Timestamp logging: disabled
    Standby logging: disabled
    Deny Conn when Queue Full: disabled
    Console logging: disabled
    Monitor logging: disabled
    Buffer logging: disabled
    Trap logging: level errors, facility 16, 3607 messages logged
        Logging to infrastructure 10.1.2.3
    History logging: disabled
    Device ID: 'inside' interface IP address "10.1.1.1"
    Mail logging: disabled
    ASDM logging: disabled
```

Configuring Log Output Destinations

This section describes how to specify where the FWSM should save or send the log messages it generates. To view syslog messages generated by the FWSM, you must specify a log output destination. If you enable logging without specifying a log output destination, the FWSM generates messages but does not save them to a location from which you can view them.

This section includes the following topics:

Sending Syslog Messages to a Syslog Server

This section describes how to configure the FWSM to send syslog messages to a syslog server.

Configuring the FWSM to send syslog messages to a syslog server enables you to archive syslog messages, limited only by the available disk space on the server, and it enables you to manipulate log data after it is saved. For example, you could specify actions to be executed when certain types of syslog messages are logged, extract data from the log and save the records to another file for reporting, or track statistics using a site-specific script.

The syslog server must run a program (known as a server) called syslogd. UNIX provides a syslog server as part of its operating system. For Windows 95 and Windows 98, obtain a syslogd server from another vendor.

Note To start logging to a syslog server you define in this procedure, be sure to enable logging for all output locations. See the "Enabling Logging to All Configured Output Destinations" section on page 24-2. To disable logging, see the "Disabling Logging to All Configured Output Destinations" section on page 24-3.

To configure the FWSM to send syslog messages to a syslog server, perform the following steps:

Step 1 To designate a syslog server to receive the syslog messages, enter the following command:

```
hostname(config)# logging host interface_name ip_address [tcp[/port] | udp[/port]]
[format emblem]
```

Where the **format emblem** keyword enables EMBLEM format logging for the syslog server (UDP only).

The *interface_name* argument specifies the interface through which you access the syslog server.

The *ip_address* argument specifies the IP address of the syslog server.

The **tcp**[*/port*] or **udp**[*/port*] argument specifies that the FWSM should use TCP or UDP to send syslog messages to the syslog server. The default protocol is UDP. You can configure the FWSM to send data to a syslog server using either UDP or TCP, but not both. If you specify TCP, the FWSM discovers when the syslog server fails and discontinues sending syslog messages. If you specify UDP, the FWSM continues to send syslog messages regardless of whether the syslog server is operational. The *port* argument specifies the port that the syslog server listens to for syslog messages. Valid port values are 1025 through 65535, for either protocol. The default UDP port is 514. The default TCP port is 1470.

For example:

```
hostname(config)# logging host dmz1 192.168.1.5
```

If you want to designate more than one syslog server as an output destination, enter a new command for each syslog server.

Step 2 To specify which syslog messages should be sent to the syslog server, enter the following command:

```
hostname(config)# logging trap {severity_level | message_list}
```

Where the *severity_level* argument specifies the severity levels of messages to be sent to the syslog server. You can specify the severity level number (0 through 7) or name. For severity level names, see the "Severity Levels" section on page 24-19. For example, if you set the severity level to 3, then the FWSM sends syslog messages for severity levels 3, 2, 1, and 0.

The *message_list* argument specifies a customized message list that identifies the syslog messages to send to the syslog server. For information about creating custom message lists, see the "Filtering Syslog Messages with Custom Message Lists" section on page 24-13.

The following example specifies that the FWSM should send to the syslog server all syslog messages with a severity level of 3 (errors) and higher. The FWSM will send messages with the severity level of 3, 2, and 1.

```
hostname(config)# logging trap errors
```

Step 3 (Optional) If needed, set the logging facility to a value other than its default of 20 by entering the following command:

```
hostname(config)# logging facility number
```

Most UNIX systems expect the syslog messages to arrive at facility 20.

```
hostname(config)# logging
```

Step 4 (Optional) To continue to pass traffic when the TCP syslog server is down, enter the following command:

hostname(config)# **logging permit-hostdown**

Where the **permit-hostdown** keyword allows new network access sessions for a TCP-based syslog server.

Sending Syslog Messages to an E-mail Address

You can configure the FWSM to send some or all syslog messages to an e-mail address. When sent by e-mail, a syslog message appears in the subject line of the e-mail message. For this reason, we recommend configuring this option to notify administrators of syslog messages with high severity levels, such as critical, alert, and emergency.

Note To start logging to an e-mail address you define in this procedure, be sure to enable logging for all output locations. See the "Enabling Logging to All Configured Output Destinations" section on page 24-2. To disable logging, see the "Disabling Logging to All Configured Output Destinations" section on page 24-3.

To designate an e-mail address as an output destination, perform the following steps:

Step 1 To specify the syslog messages to be sent to one or more e-mail addresses, enter the following command:

```
hostname(config)# logging mail {severity_level | message_list}
```

Where the *severity_level* argument specifies the severity levels of messages to be sent to the e-mail address. You can specify the severity level number (0 through 7) or name. For severity level names, see the "Severity Levels" section on page 24-19. For example, if you set the severity level to 3, then the FWSM sends syslog messages for severity levels 3, 2, 1, and 0.

The *message_list* argument specifies a customized message list that identifies the syslog messages to send to the e-mail address. For information about creating custom message lists, see the "Filtering Syslog Messages with Custom Message Lists" section on page 24-13.

The following example uses a *message_list* with the name "high-priority," previously set up with the **logging list** command:

```
hostname(config)# logging mail high-priority
```

Step 2 To specify the source e-mail address to be used when sending syslog messages to an e-mail address, enter the following command:

```
hostname(config)# logging
from-address email_address
```

For example:

```
hostname(config)# logging from-address xxx-001@example.com
```

Step 3 Specify the recipient e-mail address to be used when sending syslog messages to an e-mail destination. You can configure up to five recipient addresses. You must enter each recipient separately.

To specify a recipient address, enter the following command:

```
hostname(config)# logging recipient-address e-mail_address [severity_level]
```

If a severity level is not specified, the default severity level is used (error condition, severity level 3).

For example:

```
hostname(config)# logging recipient-address admin@example.com
```

Step 4 To specify the SMTP server to be used when sending syslog messages to an e-mail destination, enter the following command:

```
hostname(config)# smtp-server ip_address
```

For example:

```
hostname(config)# smtp-server 10.1.1.1
```

Sending Syslog Messages to ASDM

You can configure the FWSM to send syslog messages to ASDM. The FWSM sets aside a buffer area for syslog messages waiting to be sent to ASDM and saves messages in the buffer as they occur. The ASDM log buffer is a different buffer than the internal log buffer. For information about the internal log buffer, see the "Sending Syslog Messages to the Log Buffer" section on page 24-8.

When the ASDM log buffer is full, the FWSM deletes the oldest syslog message to make room in the buffer for new syslog messages. To control the number of syslog messages retained in the ASDM log buffer, you can change the size of the buffer.

This section includes the following topics:

Configuring Logging for ASDM

Note To start logging to ASDM as defined in this procedure, be sure to enable logging for all output locations. See the "Enabling Logging to All Configured Output Destinations" section on page 24-2. To disable logging, see the "Disabling Logging to All Configured Output Destinations" section on page 24-3.

To specify ASDM as an output destination, perform the following steps:

Step 1 To specify which syslog messages should go to ASDM, enter the following command:

```
hostname(config)# logging asdm {severity_level | message_list}
```

where the *severity_level* argument specifies the severity levels of messages to be sent to ASDM. You can specify the severity level number (0 through 7) or name. For severity level names, see the "Severity Levels" section on page 24-19. For example, if you set the level to 3, then the FWSM sends syslog messages for severity levels 3, 2, 1, and 0.

The *message_list* argument specifies a customized message list that identifies the syslog messages to send to ASDM. For information about creating custom message lists, see the "Filtering Syslog Messages with Custom Message Lists" section on page 24-13.

The following example shows how to enable logging and send syslog messages of severity levels 0, 1, and 2 to the ASDM log buffer:

```
hostname(config)# logging asdm 2
```

Step 2 To specify the number of syslog messages retained in the ASDM log buffer, enter the following command:

```
hostname(config)# logging asdm-buffer-size num_of_msgs
```

where *num_of_msgs* specifies the number of syslog messages that the FWSM retains in the ASDM log buffer.

The following example shows how to set the ASDM log buffer size to 200 syslog messages:

```
hostname(config)# logging asdm-buffer-size 200
```

Clearing the ASDM Log Buffer

To erase the current contents of the ASDM log buffer, enter the following command:

```
hostname(config)# clear logging asdm
```

Sending Syslog Messages to a Switch Session, Telnet Session, or SSH Session

When you log in to the FWSM from the switch, you are connected using a Telnet session. Therefore, you configure logging to a switch session the same way as you configure logging to a Telnet or SSH session.

Viewing syslog messages in a Telnet or SSH session requires two steps:

1. Specify which messages should be sent to a Telnet or SSH session.
2. View syslog messages in the current session.

This section includes the following topics:

Configuring Logging for Telnet and SSH Sessions

Note To start logging to a Telnet or SSH session as defined in this procedure, be sure to enable logging for all output locations. See the "Enabling Logging to All Configured Output Destinations" section on page 24-2. To disable logging, see the "Disabling Logging to All Configured Output Destinations" section on page 24-3.

To specify which messages should be sent to a Telnet or SSH session, enter the following command:

```
hostname(config)# logging monitor {severity_level | message_list}
```

Where the *severity_level* argument specifies the severity levels of messages to be sent to the session. You can specify the severity level number (0 through 7) or name. For severity level names, see the "Severity Levels" section on page 24-19. For example, if you set the severity level to 3, then the FWSM sends syslog messages for severity levels 3, 2, 1, and 0.

The *message_list* argument specifies a customized message list that identifies the syslog messages to send to the session. For information about creating custom message lists, see the "Filtering Syslog Messages with Custom Message Lists" section on page 24-13.

Viewing Syslog Messages in the Current Session

To enable logging in the current session, perform the following steps:

Step 1 After you log in to the FWSM, enable logging for the current session by entering the following command:

```
hostname# terminal monitor
```

This command enables logging only for the current session. If you log out, and then log in again, you need to reenter this command.

Step 2 To disable logging for the current session, enter the following command:

```
hostname(config)# terminal no monitor
```

Sending Syslog Messages to the Log Buffer

If configured as an output destination, the log buffer serves as a temporary storage location for syslog messages. New messages are appended to the end of the listing. When the buffer is full, (that is, when the buffer wraps), old messages are overwritten as new messages are generated, unless you configure the FWSM to save the full buffer to another location.

This section includes the following topics:

Enabling the Log Buffer as an Output Destination

Note To start logging to the buffer as defined in this procedure, be sure to enable logging for all output locations. See the "Enabling Logging to All Configured Output Destinations" section on page 24-2. To disable logging, see the "Disabling Logging to All Configured Output Destinations" section on page 24-3.

To enable the log buffer as a log output destination, enter the following command:

```
hostname(config)# logging buffered {severity_level | message_list}
```

Where the *severity_level* argument specifies the severity levels of messages to be sent to the buffer. You can specify the severity level number (0 through 7) or name. For severity level names, see the "Severity Levels" section on page 24-19. For example, if you set the severity level to 3, then the FWSM sends syslog messages for severity levels 3, 2, 1, and 0.

The *message_list* argument specifies a customized message list that identifies the syslog messages to send to the buffer. For information about creating custom message lists, see the "Filtering Syslog Messages with Custom Message Lists" section on page 24-13.

For example, to specify that messages with severity levels 1 and 2 should be saved in the log buffer, enter one of the following commands:

```
hostname(config)# logging buffered critical
```

or

```
hostname(config)# logging buffered level 2
```

For the *message_list* option, specify the name of a message list containing criteria for selecting messages to be saved in the log buffer.

```
hostname(config)# logging buffered notif-list
```

Viewing the Log Buffer

To view the log buffer, enter the following command:

```
hostname(config)# show logging
```

Changing the Log Buffer Size

By default, the log buffer size is 4 KB. To change the size of the log buffer, enter the following command:

```
hostname(config)# logging buffer-size bytes
```

Where the *bytes* argument sets the amount of memory used for the log buffer, in bytes. For example, if you specify 8192, the FWSM uses 8 KB of memory for the log buffer.

The following example specifies that the FWSM uses 16 KB of memory for the log buffer:

```
hostname(config)# logging buffer-size 16384
```

Automatically Saving the Full Log Buffer to Flash Memory

Unless configured otherwise, the FWSM sends messages to the log buffer on a continuing basis, overwriting old messages when the buffer is full. If you want to keep a history of syslog messages, you can configure the FWSM to send the buffer contents to another output location each time the buffer fills. Buffer contents can be saved either to internal flash memory or to an FTP server.

When saving the buffer content to another location, the FWSM creates log files with names that use a default time-stamp format, as follows:

```
LOG-YYYY-MM-DD-HHMMSS.TXT
```

Where *YYYY* is the year, *MM* is the month, *DD* is the day of the month, and *HHMMSS* is the time in hours, minutes, and seconds.

While the FWSM writes the log buffer contents to internal flash memory or an FTP server, it continues saving new messages to the log buffer.

To specify that messages in the log buffer should be saved to internal flash memory each time the buffer wraps, enter the following command:

```
hostname(config)# logging flash-bufferwrap
```

Automatically Saving the Full Log Buffer to an FTP Server

For more information about saving the buffer, see the "Saving the Current Contents of the Log Buffer to Internal Flash Memory" section.

To specify that messages in the log buffer should be saved to an FTP server each time the buffer wraps, perform the following steps:

Step 1 To enable the FWSM to send the log buffer contents to an FTP server each time the buffer wraps, enter the following command:

```
hostname(config)# logging ftp-bufferwrap
```

Step 2 To identify the FTP server, enter the following command:

```
hostname(config)# logging ftp-server server path username password
```

where the *server* argument specifies the IP address of the external FTP server.

The *path* argument specifies the directory path on the FTP server where the log buffer data is to be saved. This path is relative to the FTP root directory.

The *username* argument specifies a username that is valid for logging in to the FTP server.

The *password* argument specifies the password for the username specified.

For example:

```
hostname(config)# logging ftp-server 10.1.1.1 /syslogs logsupervisor 1luvMy10gs
```

Saving the Current Contents of the Log Buffer to Internal Flash Memory

At any time, you can save the contents of the buffer to internal flash memory. To save the current contents of the log buffer to internal flash memory, enter the following command:

```
hostname(config)# logging savelog [savefile]
```

For example, the following command saves the contents of the log buffer to internal flash memory using the filename, latest-logfile.txt:

```
hostname(config)# logging savelog latest-logfile.txt
```

Clearing the Contents of the Log Buffer

To delete the contents of the log buffer, enter the following command:

```
hostname(config)# clear logging buffer
```

Filtering Syslog Messages

This section describes how to specify which syslog messages should go to output destinations, and includes the following topics:

Message Filtering Overview

You can filter generated syslog messages so that only certain syslog messages are sent to a particular output destination. For example, you could configure the FWSM to send all syslog messages to one output destination and also to send a subset of those syslog messages to a different output destination.

Specifically, you can configure the FWSM so that syslog messages are directed to an output destination according to the following criteria:

- Syslog message ID number
- Syslog message severity level
- Syslog message class (equivalent to a functional area of the FWSM)

You customize these criteria by creating a message list that you can specify when you set the output destination in the "Configuring Log Output Destinations" section on page 24-3.

Alternatively, you can configure the FWSM to send a particular message class to each type of output destination independently of the message list.

For example, you could configure the FWSM to send to the internal log buffer all syslog messages with severity levels of 1, 2 and 3, send all syslog messages in the "ha" class to a particular syslog server, or create a list of messages that you name "high-priority" that are sent to an e-mail address to notify system administrators of a possible problem.

Filtering Syslog Messages by Class

The syslog message class provides a method of categorizing syslog messages by type, equivalent to a feature or function of the FWSM. For example, the "auth" class denotes user authentication.

This section includes the following topics:

Message Class Overview

With logging classes, you can specify an output location for an entire category of syslog messages with a single command.

You can use syslog message classes in two ways:

- Issue the **logging class** command to specify an output location for an entire category of syslog messages.
- Create a message list using the **logging list** command that specifies the message class. For instructions, see the "Filtering Syslog Messages with Custom Message Lists" section on page 24-13.

All syslog messages in a particular class share the same initial three digits in their syslog message ID numbers. For example, all syslog message IDs that begin with the digits 400 are associated with the ids class. Syslog messages associated with the IDS feature range from 400400 to 400415.

Sending All Messages in a Class to a Specified Output Destination

When you configure all messages in a class to go to a type of output destination, this configuration overrides the configuration in the specific output destination command. For example, if you specify that messages at severity level 7 should go to the log buffer, and you also specify that ha class messages at severity level 3 should go to the buffer, then the latter configuration takes precedence.

To configure the FWSM to send an entire syslog message class to a configured output destination, enter the following command:

```
hostname(config)# logging class message_class {buffered | history | mail | monitor | trap}
[severity_level]
```

Where the *message_class* argument specifies a class of syslog messages to be sent to the specified output destination. See Table 24-1 for a list of syslog message classes.

The **buffered**, **history**, **mail**, **monitor**, and **trap** keywords specify the output destination to which syslog messages in this class should be sent. The **history** keyword enables SNMP logging. The **monitor** keyword enables Telnet and SSH logging. The **trap** keyword enables syslog server logging. Select one destination per command-line entry. If you want to specify that a class should go to more than one destination, enter a new command for each output destination.

The *severity_level* argument further restricts the syslog messages to be sent to the output destination by specifying a severity level. For more information about message severity levels, see the "Severity Levels" section on page 24-19.

The following example specifies that all syslog messages related to the class ha (high availability, also known as failover) with a severity level of 1 (alerts) should be sent to the internal logging buffer.

```
hostname(config)# logging class ha buffered alerts
```

Table 24-1 lists the syslog message classes and the ranges of syslog message IDs associated with each class.

Table 24-1 Syslog Message Classes and Associated Message ID Numbers

Class	Definition	Syslog Message ID Numbers
auth	User Authentication	109, 113
bridge	Transparent Firewall	110, 220
ca	PKI Certification Authority	717
config	Command interface	111, 112, 208, 308
e-mail	E-mail Proxy	719
ha	Failover (High Availability)	101, 102, 103, 104, 210, 311, 709
ip	IP Stack	209, 215, 313, 317, 408
np	Network Processor	319
ospf	OSPF Routing	318, 409, 503, 613
rip	RIP Routing	107, 312
rm	Resource Manager	321

Table 24-1 Syslog Message Classes and Associated Message ID Numbers (continued)

Class	Definition	Syslog Message ID Numbers
session	User Session	106, 108, 201, 202, 204, 302, 303, 304, 305, 314, 405, 406, 407, 500, 502, 607, 608, 609, 616, 620, 703, 710
snmp	SNMP	212
sys	System	199, 211, 214, 216, 306, 307, 315, 414, 604, 605, 606, 610, 612, 614, 615, 701, 711

Filtering Syslog Messages with Custom Message Lists

Creating a custom message list is a flexible way to exercise fine control over which syslog messages are sent to which output destination. In a custom syslog message list, you specify groups of syslog messages using any or all of the following criteria: severity level, message IDs, ranges of syslog message IDs, or message class.

For example, you can use message lists to:

- Select syslog messages with severity levels of 1 and 2 and send them to one or more e-mail addresses.
- Select all syslog messages associated with a message class (such as “ha”) and save them to the internal buffer.

A message list can include multiple criteria for selecting messages. However, you must add each message selection criterion with a new command entry. You can create a message list containing overlapping message selection criteria. If two criteria in a message list select the same message, the message is logged only once.

To create a customized list that the FWSM can use to select messages to be saved in the log buffer, perform the following steps:

Step 1 Create a message list containing criteria for selecting messages by entering the following command:

```
hostname(config)# logging list name {level level [class message_class] |
message start_id[-end_id]}
```

Where the *name* argument specifies the name of the list. Do not use the names of severity levels as the name of a syslog message list. Prohibited names include “emergency,” “alert,” “critical,” “error,” “warning,” “notification,” “informational,” and “debugging.” Similarly, do not use the first three characters of these words at the beginning of a filename. For example, do not use a filename that starts with the characters “err.”

The **level** *level* argument specifies the severity level. You can specify the severity level number (0 through 7) or name. For severity level names, see the “Severity Levels” section on page 24-19. For example, if you set the severity level to 3, then the FWSM sends syslog messages for severity levels 3, 2, 1, and 0.

The **class** *message_class* argument specifies a particular message class. For a list of class names, see Table 24-1 on page 24-12.

The **message** *start_id*[**-***end_id*] argument specifies an individual syslog message ID number or a range of numbers.

The following example creates a message list named notif-list that specifies messages with a severity level of 3 or higher should be saved in the log buffer:

```
hostname(config)# logging list notif-list level 3
```

Step 2 (Optional) If you want to add more criteria for message selection to the list, enter the same command as in the previous step specifying the name of the existing message list and the additional criterion. Enter a new command for each criterion you want to add to the list.

The following example adds criteria to the message list: a range of message ID numbers, and the message class ha (high availability or failover).

```
hostname(config)# logging list notif-list 104024-105999
hostname(config)# logging list notif-list level critical
hostname(config)# logging list notif-list level warning class ha
```

The preceding example states that syslog messages that match the criteria specified will be sent to the output destination. The specified criteria for syslog messages to be included in the list are:

- Syslog message IDs that fall in the range of 104024 to 105999
- All syslog messages with the level of critical or higher (emergency, alert, or critical)
- All ha class syslog messages with a severity level of warning or higher (emergency, alert, critical, error, or warning)

A syslog message is logged if it satisfies any of these conditions. If a syslog message satisfies more than one of the conditions, the message is logged only once.

Customizing the Log Configuration

This section describes other options for fine tuning the logging configuration. It includes the following topics:

Configuring the Logging Queue

The FWSM has a fixed number of blocks in memory that can be allocated for buffering syslog messages while they are waiting to be sent to the configured output destination. The number of blocks required depends on the length of the syslog message queue and the number of syslog servers specified.

To specify the number of syslog messages the FWSM can hold in its queue before sending them to the configured output destination, enter the following command:

```
hostname(config)# logging queue message_count
```

where the *message_count* variable specifies the number of syslog messages that can remain in the syslog message queue while awaiting processing. The default is 512 syslog messages. A setting of 0 (zero) indicates unlimited syslog messages, that is, the queue size is limited only by block memory availability.

To view the queue and queue statistics, enter the following command:

```
hostname(config)# show logging queue
```

Including the Date and Time in Syslog Messages

To specify that syslog messages should include the date and time that the syslog messages was generated, enter the following command:

```
hostname(config)# logging timestamp
```

Including the Device ID in Syslog Messages

To configure the FWSM to include a device ID in non-EMBLEM-format syslog messages, enter the following command:

```
hostname(config)# logging device-id {context-name | hostname | ipaddress interface_name |
string text}
```

You can specify only one type of device ID for the syslog messages.

The **context-name** keyword indicates that the name of the current context should be used as the device ID (applies to multiple context mode only). If you enable the logging device ID for the admin context in multiple context mode, messages that originate in the system execution space use a device ID of **system**, and messages that originate in the admin context use the name of the admin context as the device ID.

The **hostname** keyword specifies that the hostname of the FWSM should be used as the device ID.

The **ipaddress** *interface_name* argument specifies that the IP address of the interface specified as *interface_name* should be used as the device ID. If you use the **ipaddress** keyword, the device ID becomes the specified FWSM interface IP address, regardless of the interface from which the syslog message is sent. This keyword provides a single, consistent device ID for all syslog messages that are sent from the device.

The **string** *text* argument specifies that the text string should be used as the device ID. The string can contain up to16 characters. You cannot use blank spaces or any of the following characters:

- & (ampersand)
- ' (single quote)
- " (double quote)
- < (less than)
- > (greater than)
- ? (question mark)

Note If enabled, the device ID does not appear in EMBLEM-formatted syslog messages or SNMP traps.

The following example enables the logging device ID for the security appliance:

```
hostname(config)# logging device-id hostname
```

The following example enables the logging device ID for a security context on the security appliance:

```
hostname(config)# logging device-id context-name
```

Generating Syslog Messages in EMBLEM Format

To use the EMBLEM format for syslog messages sent to a syslog server over UDP, specify the **format emblem** option when you configure the syslog server as an output destination by entering the following command:

```
hostname(config)# logging host interface_name ip_address {tcp[/port] | udp[/port]}
[format emblem]
```

Where the *interface_name* and *IP_address* specify the syslog server to receive the syslog messages, **tcp**[*/port*] and **udp**[*/port*] indicate the protocol and port that should be used, and **format emblem** enables EMBLEM formatting for messages sent to the syslog server.

The security appliance can send syslog messages using either the UDP or TCP protocol; however, you can enable the EMBLEM format only for messages sent over UDP. The default protocol and port are UDP and 514.

For example:

```
hostname(config)# logging host interface_1 122.243.006.123 udp format emblem
```

To use the EMBLEM format for syslog messages sent to destinations other than a syslog server, enter the following command:

```
hostname(config)# logging emblem
```

For more information about syslog servers, see the "Sending Syslog Messages to a Syslog Server" section on page 24-4.

Disabling a Syslog Message

To prevent the security appliance from generating a particular syslog message, enter the following command:

```
hostname(config)# no logging message message_number
```

For example:

```
hostname(config)# no logging message 113019
```

To reenable a disabled syslog message, enter the following command:

```
hostname(config)# logging message message_number
```

For example:

```
hostname(config)# logging message 113019
```

To see a list of disabled syslog messages, enter the following command:

```
hostname(config)# show logging message
```

To reenable logging of all disabled syslog messages, enter the following command:

```
hostname(config)# clear config logging disabled
```

Changing the Severity Level of a Syslog Message

To specify the logging level of a syslog message, enter the following command:

```
hostname(config)# logging message message_ID level severity_level
```

The following example modifies the severity level of syslog message 113019 from 4 (warnings) to 5 (notifications):

```
hostname(config)# logging message 113019 level 5
```

To reset the logging level of a syslog message to its default level, enter the following command:

```
hostname(config)# no logging message message_ID level current_severity_level
```

The following example modifies the severity level of syslog message 113019 to its default value of 4 (warnings):

```
hostname(config)# no logging message 113019 level 5
```

To see the severity level of a specific message, enter the following command:

```
hostname(config)# show logging message message_ID
```

To see a list of syslog messages with modified severity levels, enter the following command:

```
hostname(config)# show logging message
```

To reset the severity level of all modified syslog messages back to their defaults, enter the following command:

```
hostname(config)# clear configure logging level
```

The series of commands in the following example illustrate the use of the **logging message** command to control both whether a syslog message is enabled and the severity level of the syslog message:

```
hostname(config)# show logging message 403503
syslog 403503: default-level errors (enabled)

hostname(config)# logging message 403503 level 1
hostname(config)# show logging message 403503
syslog 403503: default-level errors, current-level alerts (enabled)

hostname(config)# no logging message 403503
hostname(config)# show logging message 403503
syslog 403503: default-level errors, current-level alerts (disabled)

hostname(config)# logging message 403503
hostname(config)# show logging message 403503
syslog 403503: default-level errors, current-level alerts (enabled)

hostname(config)# no logging message 403503 level 3
hostname(config)# show logging message 403503
syslog 403503: default-level errors (enabled)
```

Changing the Amount of Internal Flash Memory Available for Syslog Messages

You can have the FWSM save the contents of the log buffer to internal flash memory in two ways:

- Configure logging so that the contents of the log buffer are saved to internal flash memory each time the buffer wraps
- Enter a command instructing the FWSM to save the current contents of the log buffer to internal flash memory immediately

By default, the FWSM can use up to 1 MB of internal flash memory for log data. The default minimum amount of internal flash memory that must be free for the FWSM to save log data is 3 MB.

If a log file being saved to internal flash memory would cause the amount of free internal flash memory to fall below the configured minimum limit, the FWSM deletes the oldest log files to ensure that the minimum amount of memory remains free after saving the new log file. If there are no files to delete or if, after all old files are deleted, free memory would still be below the limit, the FWSM fails to save the new log file.

To modify the settings for the amount of internal flash memory available for syslog messages, perform the following steps:

Step 1 To specify the maximum amount of internal flash memory available for saving log files, enter the following command:

```
hostname(config)# logging flash-maximum-allocation kbytes
```

Where *kbytes* specifies the maximum amount of internal flash memory, in kilobytes, that can be used for saving log files.

The following example sets the maximum amount of internal flash memory that can be used for log files to approximately 1.2 MB:

```
hostname(config)# logging flash-maximum-allocation 1200
```

Step 2 To specify the minimum amount of internal flash memory that must be free for the FWSM to save a log file, enter the following command:

```
hostname(config)# logging flash-minimum-free kbytes
```

Where *kbytes* specifies the minimum amount of internal flash memory, in kilobytes, that must be available before the FWSM saves a new log file.

The following example specifies that the minimum amount of free internal flash memory must be 4000 KB before the FWSM can save a new log file:

```
hostname(config)# logging flash-minimum-free 4000
```

Understanding Syslog Messages

This section describes the contents of syslog messages generated by the security appliance. It includes the following topics:

Syslog Message Format

Syslog messages begin with a percent sign (%) and are structured as follows:

```
%FWSM Level Message_number: Message_text
```

Field descriptions are as follows:

FWSM	Identifies the syslog message facility code for messages generated by the security appliance. This value is always FWSM.
Level	Specifies 1 through 7. The level reflects the severity of the condition described by the syslog message. The lower the number, the more severe the condition. For more information, see Table 24-2.
Message_number	A unique six-digit number that identifies the syslog message.
Message_text	A text string describing the condition. This portion of the syslog message sometimes includes IP addresses, port numbers, or usernames. For a list of variable fields and their descriptions, see *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module System Log Messages*.

Severity Levels

Table 24-2 lists the syslog message severity levels.

Table 24-2 Syslog Message Severity Levels

Level Number	Level Keyword	Description
0	emergency	System unusable.
1	alert	Immediate action needed.
2	critical	Critical condition.
3	error	Error condition.
4	warning	Warning condition.
5	notification	Normal but significant condition.
6	informational	Informational message only.
7	debugging	Appears during debugging only.

Note The security appliance does not generate syslog messages with a severity level of 0 (emergency). This level is provided in the **logging** command for compatibility with the UNIX system log feature, but is not used by the security appliance.

Configuring SNMP

This section describes how to configure SNMP, but does not provide comprehensive information about all SNMP MIBs and traps. For detailed MIB and event notification information, see Appendix D, "Mapping MIBs to CLI Commands."

It includes the following topics:

- SNMP Overview, page 24-20
- Enabling SNMP, page 24-31

SNMP Overview

The FWSM provides support for network monitoring using SNMP V1 and V2c. The FWSM supports traps and SNMP read access, but does not support SNMP write access.

You can configure the FWSM to send traps (event notifications) to a network management station (NMS), or you can use the NMS to browse the MIBs on the FWSM. MIBs are a collection of definitions, and the FWSM maintains a database of values for each definition. Browsing a MIB entails issuing an SNMP get request from the NMS. Use CiscoWorks for Windows or any other SNMP V1 or V2C, MIB-II-compliant browser to receive SNMP traps and browse a MIB.

Table 24-3 lists supported MIBs and traps for the FWSM and, in multiple mode, for each context. You can download Cisco MIBs from the following website.

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

After you download the MIBs, compile them for your NMS.

Note Limit the frequency of using SNMP to obtain data, because it might degrade performance. In addition, to collect resource usage data efficiently, schedule polling on a per-context basis.

Table 24-3 SNMP MIB and Trap Support

MIB and Trap	Description
CISCO-CRYPTO-ACCELERATOR-MIB	The FWSM supports browsing of the MIB.
• CISCO-ENTITY-MIB • CISCO-ENTITY-ALARM-MIB • CISCO-ENTITY-FRU-CONTROL-MIB • CISCO-ENTITY-REDUNDANCY-MIB	The FWSM supports browsing of the following groups and tables: • entLogicalTable • entPhysicalTable The FWSM sends the following traps: • alarm-asserted • alarm-cleared • config-change • fru-insert • fru-remove • redun-switchover

Table 24-3 SNMP MIB and Trap Support (continued)

MIB and Trap	Description

CISCO-IP-PROTOCOL-FILTER-MIB	The FWSM supports browsing of the following tables:
	• cippfIpProfileTable
	• cippfIpFilterExtTable
	• cippfIpFilterStatsTable
	• cippfIpFilterTable
	The following example shows how to retrieve entries displayed from the **show access-list** command through SNMP operations on the cippfIpfilterTable and cippfIpfilterStatsTable objects.
	`!`
	`interface Vlan50`
	`nameif inside`
	`security-level 100`
	`ip address 50.0.0.2 255.0.0.0`
	`!`
	`interface Vlan60`
	`nameif outside`
	`security-level 0`
	`ip address 60.0.0.2 255.0.0.0`
	`!`
	`snmp-server host outside 60.0.0.1 community public version 2c udp-port 161`
	`!`
	
	`hostname# show access-list`
	`access-list aaa line 1 extended permit tcp any any eq www (hitcnt=0) 0xe0998155`
	
	`snmpwalk 60.0.0.2 -c public -v 2c 1.3.6.1.4.1.9.9.278 returns as`
	
	`SNMPv2-SMI::enterprises.9.9.278.1.1.1.1.2.3.97.97.97 = INTEGER: 2`
	`<<<< 2 means extended access-list`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.2.1.2.1.1 = STRING: "aaa"`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.2.1.2.2.1 = STRING: "aaa"`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.2.1.3.1.1 = INTEGER: 1`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.2.1.3.2.1 = INTEGER: 1`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.3.3.97.97.97.1 = INTEGER: 2`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.4.3.97.97.97.1 = INTEGER: 1`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.5.3.97.97.97.1 =`
	`Hex-STRING: 00 00 00 00  <-- denotes src network`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.6.3.97.97.97.1 =`
	`Hex-STRING: 00 00 00 00  <-- denotes src network mask`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.7.3.97.97.97.1 =`
	`Hex-STRING: 00 00 00 00  <-- denotes dest network`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.8.3.97.97.97.1 =`
	`Hex-STRING: 00 00 00 00  <-- denotes dest network mask`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.9.3.97.97.97.1 = INTEGER: 6  <-- 6 stands for tcp protocol number`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.10.3.97.97.97.1 =`
	`Gauge32: 0  <-0 means any port`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.11.3.97.97.97.1 =`
	`Gauge32: 0  <-0 means any port.`
	`SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.12.3.97.97.97.1 =`
	`Gauge32: 80 <- www translates to 80`

Table 24-3 SNMP MIB and Trap Support (continued)

MIB and Trap	Description
CISCO-IP-PROTOCOL-FILTER-MIB (Continued)	SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.13.3.97.97.97.1 = Gauge32: 0 <- 0 means any port. SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.16.3.97.97.97.1 = INTEGER: 2 <- 2 means log for ACL is disabled. SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.17.3.97.97.97.1 = INTEGER: 1 <- 1 means ACL log enabled. SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.22.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.23.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.24.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.25.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.26.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.27.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.4.1.2.3.97.97.97.1 = INTEGER: 0 SNMPv2-SMI::enterprises.9.9.278.1.1.4.1.3.3.97.97.97.1 = Gauge32: 0 SNMPv2-SMI::enterprises.9.9.278.1.2.1.1.1.3.97.97.97.1 = Counter64: 0 <<<< 0 is current ACL hit counter for ACL 'aaa' where “3.97.97.97” denotes the access-list name in ASCII characters. The access-list name "aaa" translates to 97.97.97, where “97” is the ASCII equivalent of the character “a.” The “3” denotes the number of characters in the ASCII list name. The following example shows an unexpanded access-list with a network object-group, which can be retrieved through SNMP operations. The hit counter for individual access-lists is aggregated and displayed in the SNMP OID "cipppfIpFilterHits." ! interface Vlan50 nameif inside security-level 100 ip address 50.0.0.2 255.0.0.0 ! interface Vlan60 nameif outside security-level 0 ip address 60.0.0.2 255.0.0.0 ! object-group network src-network network-object 50.1.1.1 255.255.255.255 network-object 50.1.1.2 255.255.255.255 network-object 50.1.1.3 255.255.255.255 object-group network dest-network network-object 60.1.1.1 255.255.255.255 network-object 60.1.1.2 255.255.255.255 network-object 60.1.1.3 255.255.255.255 access-list aaa extended permit tcp object-group src-network object-group dest-network ! snmp-server host outside 60.0.0.1 community public version 2c udp-port 161 ! hostname(config)# **show access-list**

Table 24-3 SNMP MIB and Trap Support (continued)

MIB and Trap	Description
CISCO-IP-PROTOCOL-FILTER-MIB (Continued)	access-list mode auto-commit access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096) alert-interval 300 access-list aaa; 9 elements access-list aaa line 1 extended permit tcp object-group src-network object-group dest-network 0x705bc913 <---- only exposed access-list aaa line 1 extended permit tcp host 50.1.1.1 host 60.1.1.1 (hitcnt=0) 0xcb224dc0 <---- not exposed access-list aaa line 1 extended permit tcp host 50.1.1.1 host 60.1.1.2 (hitcnt=0) 0x324aa638 <---- not exposed access-list aaa line 1 extended permit tcp host 50.1.1.1 host 60.1.1.3 (hitcnt=0) 0xca52e993 <---- not exposed access-list aaa line 1 extended permit tcp host 50.1.1.2 host 60.1.1.1 (hitcnt=0) 0xa45db454 <---- not exposed access-list aaa line 1 extended permit tcp host 50.1.1.2 host 60.1.1.2 (hitcnt=0) 0xd69df47f <---- not exposed access-list aaa line 1 extended permit tcp host 50.1.1.2 host 60.1.1.3 (hitcnt=0) 0xb06956a6 <---- not exposed access-list aaa line 1 extended permit tcp host 50.1.1.3 host 60.1.1.1 (hitcnt=0) 0xcd7aeba4 <---- not exposed access-list aaa line 1 extended permit tcp host 50.1.1.3 host 60.1.1.2 (hitcnt=0) 0x3210272d <---- not exposed access-list aaa line 1 extended permit tcp host 50.1.1.3 host 60.1.1.3 (hitcnt=0) 0xa2b03187 <---- not exposed snmpwalk 60.0.0.2 -c public -v 2c 1.3.6.1.4.1.9.9.278 SNMPv2-SMI::enterprises.9.9.278.1.1.1.1.2.3.97.97.97 = INTEGER: 2 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.3.3.97.97.97.1 = INTEGER: 2 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.4.3.97.97.97.1 = INTEGER: 1 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.5.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.6.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.7.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.8.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.9.3.97.97.97.1 = INTEGER: 6 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.10.3.97.97.97.1 = Gauge32: 0 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.11.3.97.97.97.1 = Gauge32: 0 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.12.3.97.97.97.1 = Gauge32: 0 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.13.3.97.97.97.1 = Gauge32: 0 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.16.3.97.97.97.1 = INTEGER: 2 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.17.3.97.97.97.1 = INTEGER: 1 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.22.3.97.97.97.1 = STRING: "src-network" <--- source network object group name SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.23.3.97.97.97.1 = STRING: "dest-network" <-- destination network object-group name.. SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.24.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.25.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.26.3.97.97.97.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.27.3.97.97.97.1 = ""

Table 24-3 SNMP MIB and Trap Support (continued)

MIB and Trap	Description
CISCO-IP-PROTOCOL-FILTER-MIB (Continued)	SNMPv2-SMI::enterprises.9.9.278.1.1.4.1.2.3.97.97.97.1 = INTEGER: 0 SNMPv2-SMI::enterprises.9.9.278.1.1.4.1.3.3.97.97.97.1 = Gauge32: 0 SNMPv2-SMI::enterprises.9.9.278.1.2.1.1.1.3.97.97.97.1 = Counter64: 0 <-- aggregated ACL hit counter The following example shows access-list entries displayed in the **show ipv6 access-list** command can be retrieved and displayed through SNMP operations. interface Vlan50 nameif inside security-level 100 ip address 50.0.0.2 255.0.0.0 ipv6 address 2000:400:3:1::100/64 ! interface Vlan60 nameif outside security-level 0 ip address 60.0.0.2 255.0.0.0 ipv6 address 2001:400:3:1::100/64 ! ! ipv6 access-list allow_ipv6 permit tcp any any eq www ! access-group allow_ipv6 in interface inside access-group allow_ipv6 in interface outside ! snmp-server host outside 60.0.0.1 community public version 2c udp-port 161 ! FWSM# show ipv6 access-list ipv6 access-list allow_ipv6; 1 elements ipv6 access-list allow_ipv6 line 1 permit tcp any any eq www (hitcnt=0) 0xfabbda56 snmpwalk 60.0.0.2 -c public -v 2c 1.3.6.1.4.1.9.9.278 returns as SNMPv2-SMI::enterprises.9.9.278.1.1.1.1.2.10.97.108.108.111.119.95.105.112.118.54 = INTEGER: 3 SNMPv2-SMI::enterprises.9.9.278.1.1.2.1.2.1.3 = STRING: "allow_ipv6" SNMPv2-SMI::enterprises.9.9.278.1.1.2.1.2.2.3 = STRING: "allow_ipv6" SNMPv2-SMI::enterprises.9.9.278.1.1.2.1.3.1.3 = INTEGER: 1 SNMPv2-SMI::enterprises.9.9.278.1.1.2.1.3.2.3 = INTEGER: 1 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.3.10.97.108.108.111.119.95.105.112.118.54.1 = INTEGER: 2 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.4.10.97.108.108.111.119.95.105.112.118.54.1 = INTEGER: 2 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.5.10.97.108.108.111.119.95.105.112.118.54.1 = Hex-STRING: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.6.10.97.108.108.111.119.95.105.112.118.54.1 = Hex-STRING: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 95.105.112.118.54.1 = Gauge32: 0

Table 24-3 SNMP MIB and Trap Support (continued)

MIB and Trap	Description
CISCO-IP-PROTOCOL-FILTER-MIB (Continued)	SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.7.10.97.108.108.111.119.95.105.112.118.54.1 = Hex-STRING: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.8.10.97.108.108.111.119.95.105.112.118.54.1 = Hex-STRING: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.9.10.97.108.108.111.119.95.105.112.118.54.1 = INTEGER: 6 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.10.10.97.108.108.111.119. SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.11.10.97.108.108.111.119.95.105.112.118.54.1 = Gauge32: 0 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.12.10.97.108.108.111.119.95.105.112.118.54.1 = Gauge32: 80 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.13.10.97.108.108.111.119.95.105.112.118.54.1 = Gauge32: 0 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.16.10.97.108.108.111.119.95.105.112.118.54.1 = INTEGER: 2 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.17.10.97.108.108.111.119.95.105.112.118.54.1 = INTEGER: 1 SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.22.10.97.108.108.111.119.95.105.112.118.54.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.23.10.97.108.108.111.119.95.105.112.118.54.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.24.10.97.108.108.111.119.95.105.112.118.54.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.25.10.97.108.108.111.119.95.105.112.118.54.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.26.10.97.108.108.111.119.95.105.112.118.54.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.3.1.27.10.97.108.108.111.119.95.105.112.118.54.1 = "" SNMPv2-SMI::enterprises.9.9.278.1.1.4.1.2.10.97.108.108.111.119.95.105.112.118.54.1 = INTEGER: 0 SNMPv2-SMI::enterprises.9.9.278.1.1.4.1.3.10.97.108.108.111.119.95.105.112.118.54.1 = Gauge32: 0 SNMPv2-SMI::enterprises.9.9.278.1.2.1.1.1.10.97.108.108.111.119.95.105.112.118.54.1 = Counter64: 0 **Note** You cannot perform an SNMP query for either type of access-list. You cannot perform an SNMP query for access-list entries expanded because of the use of an object-group. You can only perform an SNMP query for unexpanded access-lists using an object-group. You can only perform an SNMP query for an aggregated access-list hit counter for an access-list using an object-group. You cannot perform an SNMP query for the hit counter for access-list entries expanded because of an object-group in an access-list. You cannot perform an SNMP query for access-list names configured with more than 112 characters.

Table 24-3 SNMP MIB and Trap Support (continued)

MIB and Trap	Description
CISCO-FIREWALL-MIB	The FWSM supports browsing of the MIB. The FWSM supports browsing of the following group: • cfwSystem The information in cfwSystem.cfwStatus, which relates to failover status, pertains to the entire device and not just a single context. The FWSM supports browsing of the following table: • cfwConnectionStatTable
CISCO-IPSEC-FLOW-MONITOR-MIB	The FWSM supports browsing of the MIB. The FWSM sends the following traps: • start • stop
CISCO-L4L7-RESOURCE-LIMIT-MIB	The FWSM supports browsing of the MIB. The FWSM supports browsing of the following traps: • limit-reached • rate-limit-reached The FWSM supports browsing of the following tables: • ciscoL4L7ResourceLimitTable • ciscoL4L7ResourceRateLimitTable
CISCO-MEMORY-POOL-MIB	The FWSM supports browsing of the following table: • ciscoMemoryPoolTable—The memory usage described in this table applies only to the security appliance general-purpose processor, and not to the network processors.
CISCO-NAT-EXT-MIB	The FWSM supports browsing of the MIB.
CISCO-PROCESS-MIB	The FWSM supports browsing of the MIB. The FWSM supports browsing of the following table: • cpmCPUTotalTable The FWSM sends the following trap: • rising threshold
CISCO-REMOTE-ACCESS-MONITOR-MIB	The FWSM supports browsing of the MIB. The FWSM sends the following trap: • session-threshold-exceeded
CISCO-SYSLOG-MIB	The FWSM sends the following trap: • clogMessageGenerated You cannot browse this MIB.

Table 24-3 SNMP MIB and Trap Support (continued)

MIB and Trap	Description
CISCO-UNIFIED-FIREWALL-MIB	The FWSM supports browsing of the MIB. The FWSM supports browsing of the following group: • cufwUrlFilterGlobals—This group provides global URL filtering statistics.
IF-MIB	The FWSM supports browsing of the following tables: • ifTable • ifXTable

Table 24-3 SNMP MIB and Trap Support (continued)

MIB and Trap	Description

IP-FORWARD-MIB	The FWSM supports browsing of the following table: inetCidrRouteTable.
	
	The following example shows how entries displayed from the **show route** command can be retrieved through SNMP operations.
	
	`!`
	`interface Vlan50`
	`  nameif inside`
	`  security-level 100`
	`  ip address 50.0.0.2 255.0.0.0`
	`!`
	`interface Vlan60`
	`  nameif outside`
	`  security-level 0`
	`  ip address 60.0.0.2 255.0.0.0`
	`!`
	`snmp-server host outside 60.0.0.1 community public version 2c`
	`udp-port 161`
	`!`
	
	`hostname# show route`
	`    50.0.0.0 255.0.0.0 is directly connected, inside`
	`    60.0.0.0 255.0.0.0 is directly connected, outside`
	
	An SNMP request from the inetCidrRouteTable returns:
	
	`snmpwalk 60.0.0.2 -c public -v 2c 1.3.6.1.2.1.4.24.7 returns`
	
	`IP-MIB::ip.24.7.1.7.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: 1`
	`<---- ifindex`
	`IP-MIB::ip.24.7.1.7.1.4.60.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: 2`
	`<---- Inindex`
	`IP-MIB::ip.24.7.1.8.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: 3`
	`<---- refer local`
	`IP-MIB::ip.24.7.1.8.1.4.60.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: 3`
	`<---- refer local`
	`IP-MIB::ip.24.7.1.9.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: 2`
	`<---- 2 means local or connected route`
	`IP-MIB::ip.24.7.1.9.1.4.60.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: 2`
	`<---- 2 means local or connected route`
	`IP-MIB::ip.24.7.1.10.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = Gauge32: 0`
	`IP-MIB::ip.24.7.1.10.1.4.60.0.0.0.8.0.1.4.0.0.0.0 = Gauge32: 0`
	`IP-MIB::ip.24.7.1.11.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = Gauge32: 0`
	`IP-MIB::ip.24.7.1.11.1.4.60.0.0.0.8.0.1.4.0.0.0.0 = Gauge32: 0`
	`IP-MIB::ip.24.7.1.12.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: 0`
	`<--- primary metric 0 for connected route`
	`IP-MIB::ip.24.7.1.12.1.4.60.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: 0`
	`<--- primary metric 0 for connected route`
	`IP-MIB::ip.24.7.1.13.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: -1`
	`IP-MIB::ip.24.7.1.13.1.4.60.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: -1`
	`IP-MIB::ip.24.7.1.14.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: -1`
	`IP-MIB::ip.24.7.1.14.1.4.60.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: -1`
	`IP-MIB::ip.24.7.1.15.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: -1`
	`IP-MIB::ip.24.7.1.15.1.4.60.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: -1`
	`IP-MIB::ip.24.7.1.16.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: -1`
	`IP-MIB::ip.24.7.1.16.1.4.60.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: -1`
	`IP-MIB::ip.24.7.1.17.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: 1`
	`<----- 1 means route is active`
	`IP-MIB::ip.24.7.1.17.1.4.60.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: 1`
	`<----- 1 means route is active`

Table 24-3 SNMP MIB and Trap Support (continued)

MIB and Trap	Description

IP-FORWARD-MIB (Continued)	For an SNMP request to retrieve the SNMP OID "inetCidrRouteIfIndex" from the inetCidrRouteTable, enter the following:
	
	`snmpget 60.0.0.2 -c public -v 2c ip.24.7.1.7.1.4.50.0.0.0.8.0.1.4.0.0.0.0 returns as`
	
	`IP-MIB::ip.24.7.1.7.1.4.50.0.0.0.8.0.1.4.0.0.0.0 = INTEGER: 1`
	
	Note You cannot perform an SNMP query for IPv6 route entries.
	
	Up to a three-minute delay may occur between route entries displayed in the **show route** command, and you can perform an SNMP query for this entry.

Table 24-3 SNMP MIB and Trap Support (continued)

MIB and Trap	Description

IP-MIB	The FWSM supports browsing of the following table: ipNetToPhysicalTable
	
	The following examples show how entries displayed through the **show arp** command can be retrieved through SNMP operations.
	
	`interface Vlan50`
	` nameif inside`
	` security-level 100`
	` ip address 50.0.0.2 255.0.0.0`
	`!`
	`interface Vlan60`
	` nameif outside`
	` security-level 0`
	` ip address 60.0.0.2 255.0.0.0`
	`!`
	`snmp-server host outside 60.0.0.1 community public version 2c udp-port 161`
	`!`
	
	`hostname# show arp`
	`inside 50.0.0.1 0004.23b3.9dea`
	`outside 60.0.0.1 000e.0c4e.f6cc`
	
	For an SNMP request from the ipNetToPhysicalTable, enter the following:
	
	`snmpwalk 60.0.0.2 -c public -v 2c IP-MIB::ip.35 returns`
	
	`IP-MIB::ip.35.1.4.1.1.4.50.0.0.1 = Hex-STRING: 00 04 23 B3 9D EA`
	`IP-MIB::ip.35.1.4.2.1.4.60.0.0.1 = Hex-STRING: 00 0E 0C 4E F6 CC`
	
	For an SNMP request for a specific IP address from the ipNetToPhysicalTable, enter the following:
	
	`snmpwalk 60.0.0.2 -c public -v 2c IP-MIB::ip.35.1.4.1.1.4.50.0.0.1 returns`
	
	`IP-MIB::ip.35.1.4.1.1.4.50.0.0.1 = Hex-STRING: 00 04 23 B3 9D EA`
	
	The ipNetToPhysicalTable object is indexed by ipNetToPhysicalIfIndex, ipNetToPhysicalNetAddressType, and ipNetToPhysicalNetAddress, in which ipNetToPhysicalIfIndex will be the VLAN interface number. The ipNetToPhysicalNetAddress object is the IP address for which the MAC entry is to be retrieved. Only the ipNetToPhysicalPhysAddress object is populated from ipNetToPhysicalTable to retrieve the MAC address for the indexed IP address.
	
	Note Up to a three-minute delay may occur between ARP entries displayed in the **show arp** command, and you can perform an SNMP query for this entry.

MIB-II	The FWSM supports browsing of the following group and table: • system

Table 24-3 SNMP MIB and Trap Support (continued)

MIB and Trap	Description
NAT-MIB	The FWSM supports browsing of the MIB. The FWSM sends the following trap: • packet-discard The FWSM supports browsing of the following tables: • natAddrBindTable • natAddrPortBindTable
RFC1213-MIB	The FWSM supports browsing of the following table: • ip.ipAddrTable
SNMP core traps	The FWSM sends the following SNMP core traps: • authentication—An SNMP request fails because the NMS did not authenticate with the correct community string. • linkup—An interface has transitioned to the "up" state. • linkdown—An interface is down, for example, if you removed the **nameif** command. • coldstart—The FWSM is running after a reload.
SNMPv2-MIB	The FWSM supports browsing of the following: • snmp
TCP-MIB	The FWSM supports browsing of the following table: • tcpConnectionTable
UDP-MIB	The FWSM supports browsing of the following table: • udpEndpointTable

Enabling SNMP

This section describes how to enable SNMP on the FWSM. The SNMP agent that runs on the FWSM performs two functions:

- Replies to SNMP requests from NMSs.
- Sends traps (event notifications) to NMSs.

To enable the SNMP agent and identify an NMS that can connect to the FWSM, perform the following steps:

Step 1 To ensure that the SNMP server on the FWSM is enabled, enter the following command:

```
hostname(config)# snmp-server enable
```

The SNMP server is enabled by default.

Step 2 To identify the IP address of the NMS that can connect to the FWSM, enter the following command:

```
hostname(config)# snmp-server host interface_name ip_address [trap | poll]
[community text] [version {1 | 2c}] [udp-port port]
```

Where the *interface_name* argument specifies the interface through which you access the NMS.

The *ip_address* argument specifies the IP address of the NMS.

Specify **trap** or **poll** if you want to limit the NMS to receiving traps only or browsing (polling) only. By default, the NMS can use both functions.

To change the port number, use the **udp-port** keyword.

Step 3 To specify the community string, enter the following command:

```
hostname(config)# snmp-server community key
```

The SNMP community string is a shared secret between the FWSM and the NMS. The key is a case-sensitive value up to 32 characters in length. Spaces are not permitted.

Step 4 (Optional) To set the SNMP server location or contact information, enter the following command:

```
hostname(config)# snmp-server {contact | location} text
```

Where *text* defines the SNMP server location or contact information.

Step 5 To enable the FWSM to send traps to the NMS, enter the following command:

```
hostname(config)# snmp-server enable traps [all | syslog | snmp [trap] [...] |
cpu threshold [trap] | entity [trap] [...] | ipsec [trap] [...] | nat [trap] |
remote-access [trap] | resource [trap]]
```

Enter this command for each feature type to enable individual traps or sets of traps, or enter the **all** keyword to enable all traps.

The default configuration has all SNMP traps enabled (**snmp-server enable traps snmp authentication linkup linkdown coldstart**). You can disable these traps using the **no** form of this command with the **snmp** keyword. However, the **clear configure snmp-server** command restores the default enabling of SNMP traps.

If you enter this command and do not specify a trap type, then the default is **syslog**. (The default **snmp** traps continue to be enabled along with the **syslog** trap.)

Traps for **snmp** include:

- **authentication**
- **linkup**
- **linkdown**
- **coldstart**

Traps for **entity** include:

- **config-change**
- **fru-insert**
- **fru-remove**
- **redun-switchover**
- **alarm-asserted**
- **alarm-cleared**

Traps for **ipsec** include:

- **start**
- **stop**

Traps for **nat** include:

- **packet-discard**

Traps for **remote-access** include:

- **session-threshold-exceeded**

Traps for **resource** include:

- **limit-reached**
- **rate-limit-reached**

Traps for **cpu threshold** include:

- **rising**

To receive cpu threshold rising traps, the cpu threshold rising and monitoring values must be specified by entering the following command:

```
hostname(config)# cpu threshold rising threshold_value monitoring level
```

Step 6 To enable syslog messages to be sent as traps to the NMS, enter the following command:

```
hostname(config)# logging history level
```

You must also enable **syslog** traps using the preceding **snmp-server enable traps** command.

Step 7 To enable logging and generate syslog messages, which can then be sent to an NMS, enter the following command:

```
hostname(config)# logging enable
```

The following example sets the FWSM to receive requests from host 192.168.3.2 on the inside interface.

```
hostname(config)# snmp-server host inside 192.168.3.2
hostname(config)# snmp-server location building 42
hostname(config)# snmp-server contact Pat lee
hostname(config)# snmp-server community ohwhatakeyisthee
```

CHAPTER 25

Troubleshooting the Firewall Services Module

This chapter describes how to troubleshoot the FWSM, and includes the following sections:

Testing Your Configuration

This section describes how to test connectivity for the single mode FWSM or for each security context. The following steps describe how to ping the FWSM interfaces, and how to allow hosts on one interface to ping through to hosts on another interface.

We recommend that you only enable pinging and debug messages during troubleshooting. When you are done testing the FWSM, follow the steps in the "Disabling the Test Configuration" section on page 25-5.

This section includes:

Enabling ICMP Debug Messages and System Log Messages

Debug messages and system log messages can help you troubleshoot why your pings are not successful. The FWSM only shows ICMP debug messages for pings to the FWSM interfaces, and not for pings through the FWSM to other hosts. To enable debugging and system log messages, perform the following steps:

Step 1 To show ICMP packet information for pings to the FWSM interfaces, enter the following command:

```
hostname(config)# debug icmp trace
```

Step 2 To set system log messages to be sent to Telnet or SSH sessions, enter the following command:

```
hostname(config)# logging monitor debug
```

You can alternately use **logging buffer debug** to send messages to a buffer, and then view them later using the **show logging** command.

Step 3 To send the system log messages to your Telnet or SSH session, enter the following command:

```
hostname(config)# terminal monitor
```

Step 4 To enable system log messages, enter the following command:

```
hostname(config)# logging enable
```

The following example shows a successful ping from an external host (209.165.201.2) to the FWSM outside interface (209.165.201.1):

```
hostname(config)# debug icmp trace
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 512) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 768) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 768) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 1024) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 1024) 209.165.201.1 > 209.165.201.2
```

The preceding example shows the ICMP packet length (32 bytes), the ICMP packet identifier (1), and the ICMP sequence number (the ICMP sequence number starts at 0 and is incremented each time a request is sent).

Pinging FWSM Interfaces

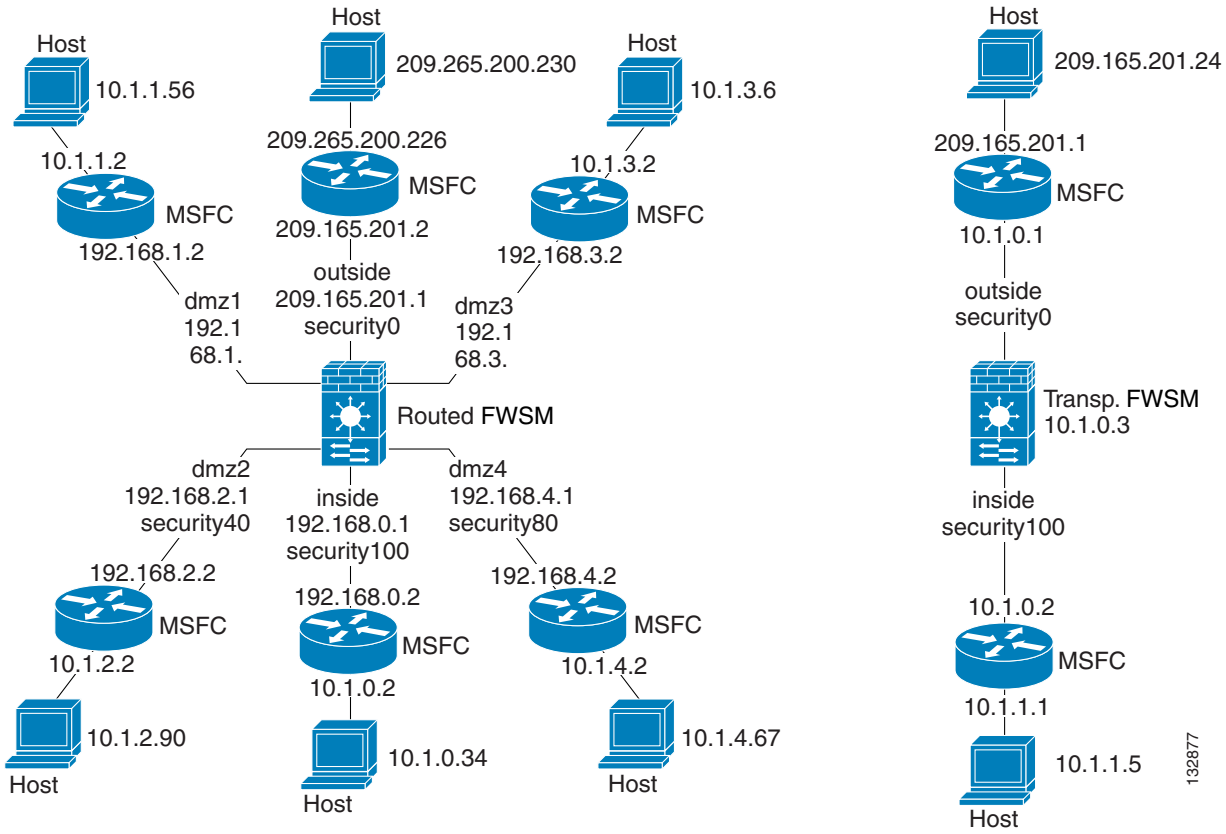
To test that the FWSM interfaces are up and running and that the FWSM and connected routers are routing correctly, you can ping the FWSM interfaces.

Note You can ping only the closest interface. Pinging the far interface is not supported.

To ping the FWSM interfaces, perform the following steps:

Step 1 Create a sketch of your single mode FWSM or security context showing the interface names, security levels, and IP addresses. The sketch should also include any directly connected routers, and a host on the other side of the router from which you will ping the FWSM. You will use this information for this procedure as well as the procedure in the "Pinging Through the FWSM" section on page 25-4. For example:

Figure 25-1 Network Sketch with Interfaces, Routers, and Hosts

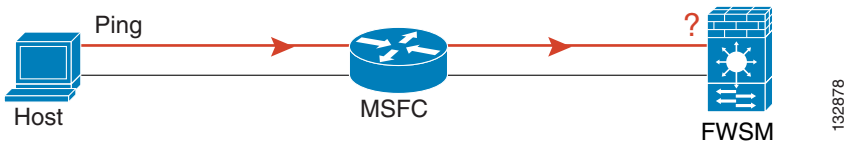


Step 2 Ping each FWSM interface from the *directly connected* routers. For transparent mode, ping the management IP address.

This test ensures that the FWSM interfaces are active and that the interface configuration is correct.

A ping might fail if the FWSM interface is not active, the interface configuration is incorrect, or if a switch between the FWSM and router is down (see Figure 25-2). In this case, no debug messages or system log messages appear on the FWSM, because the packet never reaches it.

Figure 25-2 Ping Failure at FWSM Interface

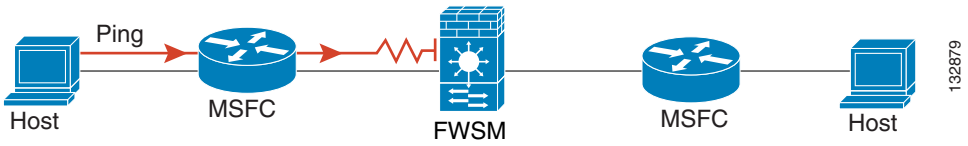


If the ping reaches the FWSM, and the FWSM responds, you see debug messages like the following:

```
ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
```

If the ping reply does not return to the router, then you might have a switch loop or redundant IP addresses (see Figure 25-3).

Figure 25-3 Ping Failure Because of IP Addressing Problems



Step 3 Ping each FWSM interface from a remote host. For transparent mode, ping the management IP address.

This test checks that the directly connected router can route the packet between the host and the FWSM, and that the FWSM can correctly route the packet back to the host.

A ping might fail if the FWSM does not have a route back to the host through the intermediate router (see Figure 25-4). In this case, the debug messages show that the ping was successful, but you see system log message 110001 indicating a routing failure.

Figure 25-4 Ping Failure Because the FWSM has no Route

Pinging Through the FWSM

After you successfully ping the FWSM interfaces, you should make sure traffic can pass successfully through the FWSM. For routed mode, this test shows that NAT is working correctly, if configured. For transparent mode, which does not use NAT, this test confirms that the FWSM is operating correctly; if the ping fails in transparent mode, contact Cisco TAC.

To ping between hosts on different interfaces, perform the following steps:

Step 1 To add an access list allowing ICMP from any source host, enter the following command:

```
hostname(config)# access-list ICMPACL extended permit icmp any any
```

By default, when hosts access a lower security interface, all traffic is allowed through. However, to access a higher security interface, you need the preceding access list.

Step 2 To assign the access list to each source interface, enter the following command:

```
hostname(config)# access-group ICMPACL in interface interface_name
```

Repeat this command for each source interface.

Step 3 To enable the ICMP inspection engine, so ICMP responses are allowed back to the source host, enter the following commands:

```
hostname(config)# class-map ICMP-CLASS
hostname(config-cmap)# match access-list ICMPACL
hostname(config-cmap)# policy-map ICMP-POLICY
hostname(config-pmap)# class ICMP-CLASS
hostname(config-pmap-c)# inspect icmp
hostname(config-pmap-c)# service-policy ICMP-POLICY global
```

Alternatively, you can also apply the ICMPACL access list to the destination interface to allow ICMP traffic back through the FWSM.

Step 4 Ping from the host or router through the source interface to another host or router on another interface.

Repeat this step for as many interface pairs as you want to check.

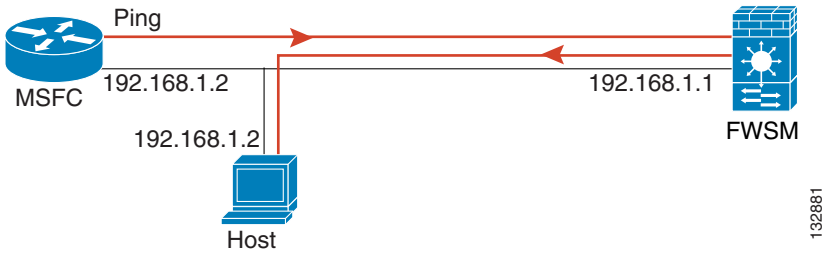
If the ping succeeds, you see a system log message confirming the address translation for routed mode (305009 or 305011) and that an ICMP connection was established (302020). You can also enter the **show xlate** and **show conns** commands to view this information.

If the ping fails for transparent mode, contact Cisco TAC.

For routed mode, the ping might fail because NAT is not configured correctly (see Figure 25-5). This is more likely if you enable NAT control. In this case, you see a system log message showing that the NAT translation failed (305005 or 305006). If the ping is from an outside host to an inside host, and you do not have a static translation (which is required with NAT control), you see message 106010: deny inbound icmp.

Note The FWSM only shows ICMP debug messages for pings to the FWSM interfaces, and not for pings through the FWSM to other hosts.

Figure 25-5 Ping Failure Because the FWSM is not Translating Addresses



Disabling the Test Configuration

After you complete your testing, disable the test configuration that allows ICMP to and through the FWSM and that prints debug messages. If you leave this configuration in place, it can pose a serious security risk. Debug messages also slow the FWSM performance.

To disable the test configuration, perform the following steps:

Step 1 To disable ICMP debug messages, enter the following command:

```
hostname(config)# no debug icmp trace
```

Step 2 To disable logging, if desired, enter the following command:

```
hostname(config)# no logging on
```

Step 3 To remove the ICMPACL access list, and also delete the related **access-group** commands, enter the following command:

```
hostname(config)# no access-list ICMPACL
```

Step 4 (Optional) To disable the ICMP inspection engine, enter the following command:

```
hostname(config)# no service-policy ICMP-POLICY
```

Reloading the FWSM

In multiple mode, you can only reload from the system execution space. To reload the FWSM, enter the following command:

```
hostname# reload
```

Performing Password Recovery

If you forget passwords, or you create a lockout situation because of AAA settings, the following sections describe how to recover:

- Clearing the Application Partition Passwords and AAA Settings, page 25-6
- Resetting the Maintenance Partition Passwords, page 25-7

Clearing the Application Partition Passwords and AAA Settings

If you forget the login and enable passwords, or you create a lockout situation because of AAA settings, you can reset the passwords and portions of AAA configuration to the default values. You must log in to the maintenance partition to perform this procedure:

Step 1 Set the application boot partition by entering the following command at the switch prompt:

```
Router# set boot device cf:n [mod_num]
```

The default boot partition for the module is cf:4. The maintenance partition is cf:1. Later in this procedure, you specify the boot partition for which you want to clear passwords.

Step 2 To boot the FWSM in to the maintenance partition, enter the following command:

```
Router# hw-module module mod_num reset cf:1
```

Step 3 To session in to the FWSM, enter the following command:

```
Router# session slot mod_num processor 1
```

Step 4 To log in to the maintenance partition as root, enter the following command:

```
Login: root
```

Step 5 Enter the password at the prompt:

```
Password: password
```

By default, the password is “cisco.”

Step 6 To clear the login and enable passwords, as well as the **aaa authentication console** and **aaa authorization command** commands, enter the following command:

```
root@localhost# clear passwd cf:{4 | 5}
```

Specify the boot partition for which you want to clear passwords. By default, the FWSM boots from **cf:4**. See Step 1 for more information about viewing the boot partition.

Step 7 Follow the screen prompts, as follows:

```
Do you wish to erase the passwords? [yn] y
The following lines will be removed from the configuration:
        enable password 8Ry2YjIyt7RRXU24 encrypted
        passwd 2KFQnbNIdI.2KYOU encrypted
Do you want to remove the commands listed above from the configuration?
[yn] y
Passwords and aaa commands have been erased.
```

Resetting the Maintenance Partition Passwords

If you forget the passwords for the maintenance partition, you can reset them to the default values. You must be logged in to the application partition. In multiple mode, you can only reset the passwords from the system execution space.

To reset the maintenance passwords, enter the following command:

```
hostname# clear mp-passwd
```

After you have reset the password, you can log in to the FWSM using the default values.

When you are logged into the FWSM, reboot it by entering the **reload** or **reboot** command.

Reset the FWSM to boot from the maintenance partition by entering the **hw-module module** *mod_num* **reset cf:1** command.

For more information, see the "Setting the Default Boot Partition" section on page 2-10 and the "Resetting the Maintenance Partition Passwords" section on page 25-7.

Other Troubleshooting Tools

The FWSM provides other troubleshooting tools to be used in conjunction with Cisco TAC:

Viewing Debug Messages

Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco TAC. Moreover, it is best to use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased debug command processing overhead will affect system use. To enable debug messages, see the **debug** commands in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Capturing Packets

Capturing packets is sometimes useful when troubleshooting connectivity problems or monitoring suspicious activity. This section includes the following topics:

Capture Overview

The FWSM is capable of tracking all IP traffic that flows across it. It is also capable of capturing all the IP traffic that is destined to the FWSM, including all the management traffic (such as SSH and Telnet traffic) to the FWSM.

The FWSM architecture consists of three different sets of processors for packet processing; this architecture poses certain restrictions on the capability of the capture feature. Typically most of the packet forwarding functionality in the FWSM is handled by the two front-end network processors, and packets are sent to the control-plane general-purpose processor only if they need application inspection (see the "Stateful Inspection Overview" section on page 1-9 for more information). The packets are sent to the session management path network processor only if there is a session miss in the accelerated path processor.

Because all the packets that are forwarded or dropped by the FWSM hits the two front-end network processors, the packet capture feature is implemented in these network processors. So all the packets that hit the FWSM can be captured by these front end processors, if an appropriate capture is configured for those traffic interfaces. On the ingress side, the packets are captured the moment the packet hits the FWSM interfaces, and on the egress side the packets are captured just before they are sent out on the wire.

Capture Limitations

The following are some of the limitations of the capture feature. Most of the limitations are due to the distributed nature of the FWSM architecture and due to the hardware accelerators that are being used in the FWSM.

- You cannot configure more than one capture per interface. But you can configure multiple ACEs in the capture access list to have a flexible configuration.
- You can only capture IP traffic. Non-IP packets like ARPs cannot be captured by the capture feature.
- For a shared VLAN:
  - You can only configure one capture for the VLAN; if you configure a capture in multiple contexts on the shared VLAN, then only the last capture that was configured is used.

    If you remove the last-configured (active) capture, no captures become active, even if you previously configured a capture in another context; you must remove and readd the capture to make it active.
  - All traffic that enters the interface to which the capture is attached (and that matches the capture access list) is captured, including traffic to other contexts on the shared VLAN.

    Therefore, if you enable a capture in Context A for a VLAN that is also used by Context B, both Context A and Context B ingress traffic is captured.

For egress traffic, only the traffic of the context with the active capture is captured. The only exception is when you do not enable the ICMP inspection (therefore the ICMP traffic does not have a session in the accelerated path). In this case, both ingress and egress ICMP traffic for all contexts on the shared VLAN is captured.

Configuring a Packet Capture

Configuring a capture typically involves configuring an access list that matches the traffic that needs to be captured. Once an access list that matches the traffic pattern is configured, then you need to define a capture and associate this access list to the capture, along with the interface on which the capture needs to be configured. Note that a capture only works if an access list and an interface are associated with a capture for capturing IPv4 traffic. The access list is not required for IPv6 traffic.

To configure a packet capture for IPv4 traffic, perform the following steps:

Step 1 Configure an extended access list that matches the traffic that needs to be captured according to the "Adding an Extended Access List" section on page 12-6.

For example, the following access list identifies all traffic:

```
hostname(config)# access-list capture extended permit ip any any
```

Step 2 To configure the capture, enter the following command.

```
hostname(config)# capture name access-list acl_name interface interface_name
```

By default configuring a capture creates a linear capture buffer of size 512 KB. You can optionally configure a circular buffer. By default only 68 bytes of the packets are captured in the buffer. You can optionally change this value. See the **capture** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for these and other options.

For example, the following command creates a capture called ip-capture using the capture access lost configured in Step 1 that is applied to the outside interface:

```
hostname(config)# capture ip-capture access-list capture interface outside
```

Step 3 To view the capture, enter the following command:

```
hostname(config)# show capture name
```

You can also copy the capture using the **copy capture** command. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference* for more information.

Step 4 To end the capture but retain the buffer, enter the following command:

```
hostname(config)# no capture name access-list acl_name interface interface_name
```

Step 5 To end the capture and delete the buffer, enter the following command:

```
hostname(config)# no capture name
```

Viewing the Crash Dump

If the FWSM crashes, you can view the crash dump information. We recommend contacting Cisco TAC if you want to interpret the crash dump. See the **show crashdump** command in the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*.

Common Problems

This section describes common problems with the FWSM, and how you might resolve them.

Symptom When you reset the FWSM from the switch CLI, the system always boots in to the maintenance partition.

Possible Cause The default boot partition is set to cf:1.

Recommended Action Change the default boot partition according to the "Setting the Default Boot Partition" section on page 2-10.

Symptom You are unable to log in to the maintenance partition with the same password as the application partition.

Possible Cause The application partition and the maintenance partition have different password databases.

Recommended Action Use the password appropriate for your partition. See the "Changing the Passwords" section on page 7-1 for more information.

Symptom Traffic does not pass through the FWSM.

Possible Cause The VLANs are not configured on the switch or are not assigned to the FWSM.

Recommended Action Configure the VLANs and assign them to the FWSM according to the "Assigning VLANs to the Firewall Services Module" section on page 2-2.

Symptom You cannot configure a VLAN interface within a context.

Possible Cause You did not assign that VLAN to the context.

Recommended Action Assign VLANs to contexts according to the "Configuring a Security Context" section on page 4-27.

Symptom You cannot add more than one switched virtual interface (SVI) to the MSFC.

Possible Cause You did not enable multiple SVIs.

Recommended Action Enable multiple SVIs according to the "Adding Switched Virtual Interfaces to the MSFC" section on page 2-4.

Symptom You cannot make a Telnet connection or SSH to the FWSM interface.

Possible Cause You did not enable Telnet or SSH to the FWSM.

Recommended Action Enable Telnet or SSH to the FWSM according to the "Allowing Telnet Access" section on page 22-1 or the "Allowing SSH Access" section on page 22-2.

Symptom You cannot ping the FWSM interface.

Possible Cause You did not enable ICMP to the FWSM.

Recommended Action Enable ICMP to the FWSM according to the "Allowing ICMP to and from the FWSM" section on page 22-9.

Symptom You cannot ping through the FWSM, even though the access list allows it.

Possible Cause You did not enable the ICMP inspection engine or apply access lists on both the source and destination interfaces.

Recommended Action Because ICMP is a connectionless protocol, the FWSM does not automatically allow returning traffic through. In addition to an access list on the source interface, you either need to apply an access list to destination interface to allow replying traffic, or enable the ICMP inspection engine, which treats ICMP connections as stateful connections.

Symptom Traffic does not go through the FWSM from a higher security interface to a lower security interface.

Possible Cause You did not apply an access list to the higher security interface to allow traffic through. Unlike the PIX firewall, the FWSM does not automatically allow traffic to pass between interfaces.

Recommended Action Apply an access list to the source interface to allow traffic through. See the "Adding an Extended Access List" section on page 12-6.

Symptom Traffic does not pass between two interfaces on the same security level.

Possible Cause You did not enable the feature that allows traffic to pass between interfaces on the same security level.

Recommended Action Enable this feature according to the "Allowing Communication Between Interfaces on the Same Security Level" section on page 6-6.

Symptom When the FWSM fails over, the secondary unit does not pass traffic.

Possible Cause You did not assign the same VLANs for both units.

Recommended Action Make sure to assign the same VLANs to both units in the switch configuration.

PART 4

Reference

APPENDIX A

Specifications

This appendix lists the specifications of the FWSM and includes the following sections:

Switch Hardware and Software Compatibility

You can install the FWSM in the Catalyst 6500 series switches or the Cisco 7600 series routers. The configuration of both series is identical, and the series are referred to generically in this guide as the "switch." The switch includes a switch (the supervisor engine) as well as a router (the MSFC 2).

The switch supports Cisco IOS software on both the switch supervisor engine and the integrated MSFC router.

Note The Catalyst operating system software is not supported.

The FWSM does not support a direct connection to a switch WAN port because WAN ports do not use static VLANs. However, the WAN port can connect to the MSFC, which can connect to the FWSM.

The FWSM runs its own operating system.

This section includes the following topics:

Catalyst 6500 Series Requirements

Table A-1 shows the supervisor engine version and software.

Table A-1 Support for FWSM 4.0 on the Catalyst 6500

		FWSM Features:		
	Supervisor Engines[1]	PISA Integration	Route Health Injection	Virtual Switching System
Cisco IOS Software Release				
12.2(18)SXF and higher	720, 32	No	No	No
12.2(18)SXF2 and higher	2, 720, 32	No	No	No
12.2(33)SXI	720-10GE	No	**Yes**	**Yes**
12.2(33)SXI	720	No	**Yes**	No
12.2(33)SXI	32	No	**Yes**	No
12.2(18)ZYA	32-PISA	**Yes**	No	No
Cisco IOS Software Modularity Release				
12.2(18)SXF4	720, 32	No	No	No

1. The FWSM does not support the supervisor 1 or 1A.

Cisco 7600 Series Requirements

Table A-2 shows the supervisor engine version and software.

Table A-2 Support for FWSM 4.0 on the Cisco 7600

		FWSM Features:		
	Supervisor Engines[1]	PISA Integration	Route Health Injection	Virtual Switching System
Cisco IOS Software Release				
12.2(33)SRA	720, 32	No	No	No
12.2(33)SRC	720-1GE	No	No	No
12.2(33)SRD	720-1GE	No	No	No

1. The FWSM does not support the supervisor 1 or 1A.

Licensed Features

The FWSM supports the following licensed features:

- Multiple security contexts. The FWSM supports two contexts plus one admin context for a total of three security contexts without a license. For more than three contexts, obtain one of the following licenses:

  - 20
  - 50
  - 100
  - 250
- GTP/GPRS support.
- BGP stub support.

Physical Attributes

Table A-3 lists the physical attributes of the FWSM.

Table A-3 Physical Attributes

Specification	Description
Bandwidth	CEF256 line card with a 6-Gbps path to the Switch Fabric Module (if present) or the 32-Gbps shared bus.
Memory	• 1-GB RAM. • 128-MB Flash memory.
Modules per switch	Maximum four modules per switch. If you are using failover, you can still only have four modules per switch even if two of them are in standby mode.

Feature Limits

Table A-4 lists the feature limits for the FWSM.

Table A-4 Feature Limits

	Context Mode	
Specification	**Single**	**Multiple**
AAA servers (RADIUS and TACACS+)	16	4 per context
Failover interface monitoring	250	250 divided between all contexts
Filtering servers (Websense Enterprise and Sentian by N2H2)	16	4 per context

Table A-4 Feature Limits (continued)

Specification	Context Mode	
	Single	Multiple
Fragmented packets	• If the FWSM receives a fragment set that is originally 8782 Bytes or smaller, then it reassembles the set and transmits it back on the wire, but the fragment size may be different than what was received. • If the FWSM receives a fragment set that is originally 8783 Bytes or larger, then: – If the frame is the first packet in a connection (as in the case of ICMP) then the FWSM reassembles the first 8782 Bytes and pass those on, but the remaining fragments are dropped. – If the frame is *not* the first packet in a connection, then the FWSM reassembles the first 8782 bytes and passes those on, and the remaining fragments are also passed on, but without the reassembly check.	
Jumbo Ethernet packets	8500 Bytes	8500 Bytes
Security contexts	N/A	250 security contexts (depending on your software license).
Syslog servers	16	4 per context Maximum of 16 divided between all contexts
VLAN interfaces		
Routed Mode	256	100 per context The FWSM has an overall limit of 1000 VLAN interfaces divided between all contexts. You can share outside interfaces between contexts, and in some circumstances, you can share inside interfaces.
Transparent Mode	8 pairs	8 pairs per context

Managed System Resources

Table A-5 lists the managed system resources of the FWSM. You can manage these resources per context using the resource manager. See the "Configuring Resource Management" section on page 4-21.

Table A-5 Managed System Resources

Specification	Context Mode	
	Single	Multiple
MAC addresses (transparent firewall mode only)	65,536	65,536 divided between all contexts
Hosts allowed to connect through the FWSM, concurrent	262,144	262,144 divided between all contexts

Table A-5 Managed System Resources (continued)

Specification	Context Mode	
	Single	Multiple
Inspection engine connections, rate	10,000 per second	10,000 per second divided between all contexts
IPSec management connections, concurrent	5	5 per context Maximum of 10 divided between all contexts
ASDM management sessions, concurrent[1]	5	Up to 5 per context Maximum of 80 divided between all contexts
NAT translations (xlates), concurrent	262,144	262,144 divided between all contexts
SSH management connections, concurrent[2]	5	5 per context Maximum of 100 divided between all contexts
System log messages, rate	30,000 per second for messages sent to the FWSM terminal or buffer 25,000 per second for messages sent to a syslog server	30,000 per second divided between all contexts for messages sent to the FWSM terminal or buffer 25,000 per second divided between all contexts for messages sent to a syslog server
TCP or UDP connections[3] [4] between any two hosts, including connections between one host and multiple other hosts, concurrent and rate	999,900[5] 100,000 per second	999,900 divided between all contexts[5] 100,000 per second divided between all contexts
Telnet management connections, concurrent[2]	5	5 per context Maximum of 100 connections divided between all contexts.

1. ASDM sessions use two HTTPS connections: one for monitoring that is always present, and one for making configuration changes that is present only when you make changes. For example, the system limit of 80 ASDM sessions represents a limit of 160 HTTPS connections.
2. The admin context can use up to 15 Telnet and SSH connections.
3. Embryonic connections are included in the total number of connections. If you configure an embryonic connection limit, then embryonic connections above the limit are not counted.
4. The FWSM might take up to 500 ms to remove a connection that is marked for deletion. Because any traffic on the connection is dropped during this period, you cannot initiate a new connection to the same destination using the same source and destination ports until the connection is deleted. Although most TCP applications do not reuse the same ports in back-to-back connections, RSH might reuse the same ports. If you use RSH or any other application that reuses the same ports in back-to-back connections, the FWSM might drop packets.
5. Because PAT requires a separate translation for each connection, the effective limit of connections using PAT is the translation limit (256 K), not the higher connection limit. To use the connection limit, you need to use NAT, which allows multiple connections using the same translation session.

Fixed System Resources

Table A-6 lists the fixed system resources of the FWSM.

Table A-6 Fixed System Resources

Specification	Context Mode	
	Single	Multiple
AAA connections, rate	80 per second	80 per second divided between all contexts
Downloaded ACEs for network access authorization	3,500	3,500 divided between all contexts
ACL logging flows, concurrent	32,768	32,768 divided between all contexts
Alias statements	512	512 divided between all contexts
ARP table entries, concurrent	65,536	65,536 divided between all contexts
DNS inspections, rate	5000 per second	5000 per second divided between all contexts
Global statements	4204	4204 divided between all contexts
Inspection statements	32	32 per context
NAT statements	2048	2048 divided between all contexts
Packet reassembly, concurrent	30,000	30,000 fragments divided between all contexts
Route table entries, concurrent	32,768	32,768 divided between all contexts
Shun statements	5120	5120 divided between all contexts
Static NAT statements	2048	2048 divided between all contexts
TFTP sessions, concurrent[1]	999,100	999,100 divided between all contexts
URL filtering requests	200 per second causes 50% CPU usage	200 per second causes 50% CPU usage divided between all contexts
User authentication sessions, concurrent	51,200	51,200 divided between all contexts
User authorization sessions, concurrent	153,600 Maximum 15 sessions per user.	153,600 divided between all contexts Maximum 15 sessions per user.

1. In FWSM Version 1.1, the number of TFTP sessions was limited to 1024 sessions.

Rule Limits

The FWSM supports a fixed number of rules for the entire system. This section includes the following topics:

- Default Rule Allocation, page A-7
- Rules in Multiple Context Mode, page A-7
- Reallocating Rules Between Features, page A-8

Default Rule Allocation

Table A-7 lists the default number of rules for each feature type.

Note Some access lists use more memory than others. Depending on the type of access list, the actual limit the system can support will be less than the maximum. See the "Maximum Number of ACEs" section on page 12-6 for more information about ACEs and memory usage.

Table A-7 Default Rule Allocation

	Context Mode	
Specification	**Single**	**Multiple (Maximum per Partition) with 12[1] pools**
AAA Rules	8744	1345
ACEs	100,567	14,801
established commands[2]	624	96
Filter Rules	3747	576
ICMP, Telnet, SSH, and HTTP Rules	2498	384
Policy NAT ACEs[3]	2498	384
Inspect Rules	5621	1537
Total Rules	**124,923**	**19,219**

1. Use the **show resource rule** command to view the default values for partitions other than 12.
2. Each **established** command creates a control and data rule, so this value is doubled in the Total Rules value.
3. This limit is lower than in release 2.3.

Rules in Multiple Context Mode

In multiple context mode with the default of 12 memory partitions, each context supports the maximum number of rules listed in Table A-7; the actual number of rules supported in a context might be more or less, depending on how many contexts you have and how many partitions you configure. See the "About Memory Partitions" section on page 4-12 for information about memory distribution among contexts.

If you reduce the number of partitions, the maximum number of rules is recalculated and might not match the total system number available for 12 partitions. To view the maximum number of rules for partitions, enter the following command in the system execution space:

```
hostname(config)# show resource rule
```

For example, the following is sample output from the **show resource rule** command, and shows the maximum rules as 19219 per partition with 12 partitions (this is an example only, and might differ from the actual number of rules for your system):

```
hostname(config)# show resource rule

             Default  Configured  Absolute
 CLS Rule     Limit      Limit      Max
-----------+---------+----------+---------
 Policy NAT     384        384        833
 ACL          14801      14801      14801
```

```
 Filter          576         576        1152
 Fixup          1537        1537        3074
 Est Ctl          96          96          96
 Est Data         96          96          96
 AAA            1345        1345        2690
 Console         384         384         768
-----------+---------+----------+---------
 Total         19219       19219

Partition Limit - Configured Limit = Available to allocate
      19219     -      19219       =           0
```

Reallocating Rules Between Features

You can reallocate rules from one feature to another feature.

Note In multiple context mode, you can also set the rule allocation per partition, which overrides the global setting in this section. See the "Reallocating Rules Between Features for a Specific Memory Partition" section on page 4-19.

Guidelines

Caution Failure to follow these guidelines might result in dropped access list configuration as well as other anomalies, including ACL tree corruption.

- The target partition and rule allocation settings must be carefully calculated, planned, and preferably tested in a non-production environment prior to making the change to ensure that all existing contexts and rules can be accommodated.
- When failover is used, both FWSMs need to be reloaded at the same time after making partition changes. Reloading both FWSMs causes an outage with no possibility for a zero-downtime reload. At no time should two FWSMs with a mismatched number of partitions or rule limits synchronize over failover.

Detailed Steps

To reallocate rules, perform the following steps:

Step 1 To view the total number of rules available, the default values, current rule allocation, and the absolute maximum number of rules you can allocate per feature, enter the following command:

```
hostname(config)# show resource rule
```

For multiple context mode, enter this command in the system execution space. It shows the number of rules per partition. See the "About Memory Partitions" section on page 4-12 for more information about partitions.

For example, the following is sample output from the **show resource rule** command, and shows the maximum rules as 124923 in single mode (this is an example only, and might differ from the actual number of rules for your system):

```
hostname(config)# show resource rule
```

```
             Default  Configured  Absolute
 CLS Rule     Limit      Limit      Max
-----------+---------+----------+---------
 Policy NAT    2498       2498      10000
 ACL         100567     100567     100567
 Filter        3747       3747       7494
 Fixup         5621       5621      10000
 Est Ctl        624        624        624
 Est Data       624        624        624
 AAA           8744       8744      10000
 Console       2498       2498       4996
-----------+---------+----------+---------
 Total       124923     124923

Partition Limit - Configured Limit = Available to allocate
     124923     -     124923       =           0
```

Step 2 To view the number of rules currently being used so you can plan your reallocation, enter one of the following commands.

- In single mode or within a context, enter the following command:

  ```
  hostname(config)# show np 3 acl count 0
  ```

- In multiple context mode system execution space, enter the following command:

  ```
  hostname(config)# show np 3 acl count partition_number
  ```

For example, the following is sample output from the **show np 3 acl count** command, and shows the number of inspections (Fixup Rule) close to the maximum of 9216. You might choose to reallocate some access list rules (ACL Rule) to inspections.

```
hostname(config)# show np 3 acl count 0

-------------- CLS Rule Current Counts --------------
CLS Filter Rule Count       :             0
CLS Fixup Rule Count        :          9001
CLS Est Ctl Rule Count      :             4
CLS AAA Rule Count          :            15
CLS Est Data Rule Count     :             4
CLS Console Rule Count      :            16
CLS Policy NAT Rule Count   :             0
CLS ACL Rule Count          :         30500
CLS ACL Uncommitted Add     :             0
CLS ACL Uncommitted Del     :             0
...
```

Note The **established** command creates two types of rules, control and data. Both of these types are shown in the display, but you allocate both rules by setting the number of **established** commands; you do not set each rule separately.

Step 3 To reallocate rules between features, enter the following command (in multiple context mode, enter it in the system execution space). If you increase the value for one feature, then you must decrease the value by the same amount for one or more features so the total number of rules does not exceed the system limit. See Step 1 to use the **show resource rule** command for the total number of rules allowed.

```
hostname(config)# resource rule nat {max_policy_nat_rules | current | default | max}
acl {max_ace_rules | current | default | max}
filter {max_filter_rules | current | default | max}
fixup {max_inspect_rules | current | default | max}
```

est {*max_established_rules* | **current** | **default** | **max**}
aaa {*max_aaa_rules* | **current** | **default** | **max**}
console {*max_console_rules* | **current** | **default** | **max**}

In multiple context mode, this command sets the rule allocation *per partition*. You must enter all arguments in this command. This command takes effect immediately.

The **nat** *max_nat_rules* arguments set the maximum number of policy NAT ACEs, between 0 and 10000.

The **acl** *max_nat_rules* arguments set the maximum number of ACEs, between 0 and the system limit. The system limit depends on single or multiple context mode, and how many memory partitions you configured. For single mode, the value is 100567. For multiple mode, see Step 1 to use the **show resource rule** command.

The **filter** *max_nat_rules* arguments set the maximum number of filter rules, between 0 and 6000.

The **fixup** *max_nat_rules* arguments set the maximum number of inspect rules, between 0 and 10000.

The **est** *max_nat_rules* arguments set the maximum number of **established** commands, between 0 and 716. The established command creates two types of rules, control and data. Both of these types are shown in the **show np 3 acl count** and **show resource rules** display, but you set both rules using the **est** keyword, which correlates with the number of **established** commands. Be sure to double the value you enter here when comparing the total number of configured rules with the total number of rules shown in the **show** commands.

The **aaa** *max_nat_rules* arguments set the maximum number of AAA rules, between 0 and 10000.

The **console** *max_nat_rules* arguments set the maximum number of ICMP, Telnet, SSH, and HTTP rules, between 0 and 4000.

The **current** keyword keeps the current value set.

The **default** keyword sets the maximum rules to the default.

The **max** keyword sets the rules to the maximum allowed for the feature. Be sure to set other features lower to accommodate this value.

For example, to reallocate 1000 rules from the single-mode default 74,188 ACEs to inspections (default 4147), enter the following command:

```
hostname(config)# resource rule nat default acl 73188 filter default fixup 5157 est
default aaa default console default
```

In multiple context mode with 12 partitions, to reallocate 100 ACEs (default 10,633) to inspections (default 1417) as well as all but one established rule (default 70) to filter (default 425), enter the following command:

```
hostname(config)# resource rule nat default acl 10533 filter 494 fixup 1517 est 1 aaa
default console default
```

APPENDIX B

Sample Configurations

This appendix illustrates and describes a number of common ways to implement FWSM, and includes the following sections:

Routed Mode Sample Configurations

This section includes the following topics:

Example 1: Multiple Mode Firewall with Outside Access

The following configuration creates three security contexts plus the admin context, each with an inside and an outside interface. The Customer C context includes a DMZ interface where a Websense server for HTTP filtering resides on the service provider premises (see Figure B-1).

Inside hosts can access the Internet through the outside interface using dynamic NAT or PAT, but no outside hosts can access the inside.

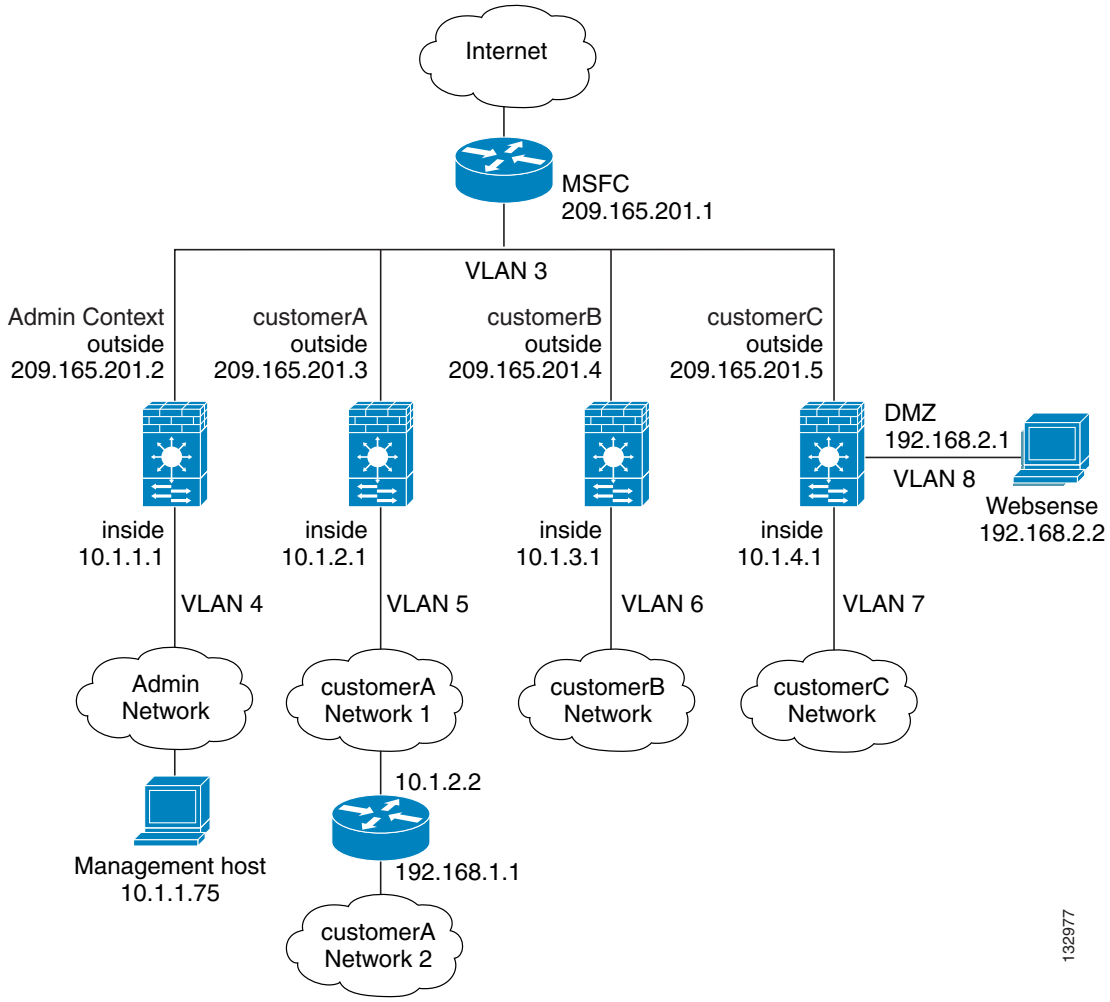
The Customer A context has a second network behind an inside router.

The admin context allows SSH sessions to FWSM from one host.

Each customer context belongs to a class that limits its resources (gold, silver, or bronze).

Although inside IP addresses can be the same across contexts when the interfaces are unique, keeping them unique is easier to manage.

Figure B-1 Example 1



See the following sections for the configurations for this scenario:

- System Configuration (Example 1), page B-2
- Admin Context Configuration (Example 1), page B-3
- Customer A Context Configuration (Example 1), page B-4
- Customer B Context Configuration (Example 1), page B-4
- Customer C Context Configuration (Example 1), page B-5
- Switch Configuration (Example 1), page B-5

System Configuration (Example 1)

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts. The mode and activation key are not stored in the configuration file, even though they endure reboots. If you view the configuration on the FWSM using the **write terminal**, **show startup-config**, or **show running-config** commands, the mode displays after the FWSM Release (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
hostname Farscape
password passw0rd
enable password chr1cht0n
admin-context admin
interface vlan 3
interface vlan 4
interface vlan 5
interface vlan 6
interface vlan 7
interface vlan 8
context admin
    allocate-interface vlan3
    allocate-interface vlan4
    config-url disk://admin.cfg
    member default
context customerA
    description This is the context for customer A
    allocate-interface vlan3
    allocate-interface vlan5
    config-url disk://contexta.cfg
    member gold
context customerB
    description This is the context for customer B
    allocate-interface vlan3
    allocate-interface vlan6
    config-url disk://contextb.cfg
    member silver
context customerC
    description This is the context for customer C
    allocate-interface vlan3
    allocate-interface vlan7-vlan8
    config-url disk://contextc.cfg
    member bronze
class gold
    limit-resource all 7%
    limit-resource rate conns 2000
    limit-resource conns 20000
class silver
    limit-resource all 5%
    limit-resource rate conns 1000
    limit-resource conns 10000
class bronze
    limit-resource all 3%
    limit-resource rate conns 500
    limit-resource conns 5000
```

Admin Context Configuration (Example 1)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

The host at 10.1.1.75 can access the context using SSH, which requires a key to be generated using the **crypto key generate** command. The certificate is saved in Flash memory.

```
interface vlan 3
    nameif outside
    security-level 0
    ip address 209.165.201.2 255.255.255.224
interface vlan 4
    nameif inside
    security-level 100
    ip address 10.1.1.1 255.255.255.0
```

```
passwd secret1969
enable password h1andl0
route outside 0 0 209.165.201.1 1
ssh 10.1.1.75 255.255.255.255 inside
nat (inside) 1 10.1.1.0 255.255.255.0
! This context uses dynamic NAT for inside users that access the outside
global (outside) 1 209.165.201.10-209.165.201.29
! The host at 10.1.1.75 has access to the Websense server in Customer C, and
! it needs a static translation for use in Customer C's access list
static (inside,outside) 209.165.201.30 10.1.1.75 netmask 255.255.255.255
access-list INTERNET remark -Allows inside hosts to access the outside for any IP traffic
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
```

Customer A Context Configuration (Example 1)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface vlan 3
    nameif outside
    security-level 0
    ip address 209.165.201.3 255.255.255.224
interface vlan 5
    nameif inside
    security-level 100
    ip address 10.1.2.1 255.255.255.0
passwd hell0!
enable password enter55
route outside 0 0 209.165.201.1 1
! The Customer A context has a second network behind an inside router that requires a
! static route. All other traffic is handled by the default route pointing to the router.
route inside 192.168.1.0 255.255.255.0 10.1.2.2 1
nat (inside) 1 10.1.2.0 255.255.255.0
! This context uses dynamic PAT for inside users that access that outside. The outside
! interface address is used for the PAT address
global (outside) 1 interface
access-list INTERNET remark -Allows inside hosts to access the outside for any IP traffic
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
```

Customer B Context Configuration (Example 1)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface vlan 3
    nameif outside
    security-level 0
    ip address 209.165.201.4 255.255.255.224
interface vlan 6
    nameif inside
    security-level 100
    ip address 10.1.3.1 255.255.255.0
passwd tenac10us
enable password defen$e
route outside 0 0 209.165.201.1 1
nat (inside) 1 10.1.3.0 255.255.255.0
! This context uses dynamic PAT for inside users that access the outside
```

```
global (outside) 1 209.165.201.9 netmask 255.255.255.255
access-list INTERNET remark Inside users only access HTTP and HTTPS servers on the outside
access-list INTERNET extended permit tcp any any eq http
access-list INTERNET extended permit tcp any any eq https
access-group INTERNET in interface inside
```

Customer C Context Configuration (Example 1)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface vlan 3
    nameif outside
    security-level 0
    ip address 209.165.201.5 255.255.255.224
interface vlan 7
    nameif inside
    security-level 100
    ip address 10.1.4.1 255.255.255.0
interface vlan 8
    nameif dmz
    security-level 50
    ip address 192.168.2.1 255.255.255.0
passwd fl0wer
enable password treeh0u$e
route outside 0 0 209.165.201.1 1
url-server (dmz) vendor websense host 192.168.2.2 url-block block 50
url-cache dst 128
filter url http 10.1.4.0 255.255.255.0 0 0
! When inside users access an HTTP server, FWSM consults with a
! Websense server to determine if the traffic is allowed
nat (inside) 1 10.1.4.0 255.255.255.0
! This context uses dynamic NAT for inside users that access the outside
global (outside) 1 209.165.201.9 netmask 255.255.255.255
! A host on the admin context requires access to the Websense server for management using
! pcAnywhere, so the Websense server uses a static translation for its private address
static (dmz,outside) 209.165.201.6 192.168.2.2 netmask 255.255.255.255
access-list INTERNET remark -Allows all inside hosts to access the outside for any IP
access-list INTERNET remark -traffic, but denies them access to the dmz.
access-list INTERNET extended deny ip any 192.168.2.0 255.255.255.0
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
access-list MANAGE remark -Allows the management host to use pcAnywhere on the
access-list MANAGE remark -Websense server
access-list MANAGE extended permit tcp host 209.165.201.30 host 209.165.201.6 eq
pcanywhere-data
access-list MANAGE extended permit udp host 209.165.201.30 host 209.165.201.6 eq
pcanywhere-status
access-group MANAGE in interface outside
access-list WEBSENSE remark -The Websense server needs to access the Websense updater
access-list WEBSENSE remark -server on the outside
access-list WEBSENSE extended permit tcp host 192.168.2.2 any eq http
access-group WEBSENSE in interface dmz
```

Switch Configuration (Example 1)

The following lines in the Cisco IOS switch configuration relate to the FWSM:

```
...
firewall module 8 vlan-group 1
```

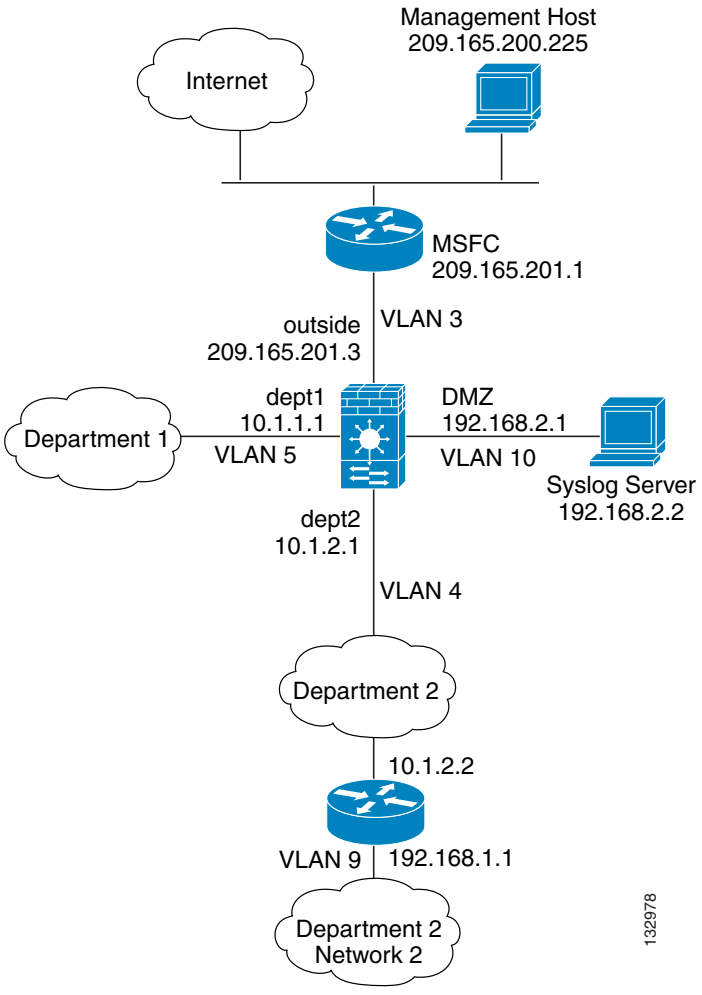
```
firewall vlan-group 1 3-8
interface vlan 3
    ip address 209.165.201.1 255.255.255.224
    no shutdown
...
```

Example 2: Single Mode Firewall Using Same Security Level Example

The following configuration creates three internal interfaces. Two of the interfaces connect to departments that are on the same security level. The DMZ interface hosts a syslog server. The management host on the outside needs access to the Syslog server and the FWSM. To connect to the FWSM, the host uses a VPN connection. FWSM uses RIP on the inside interfaces to learn routes. Because the FWSM does not advertise routes with RIP, the upstream router needs to use static routes for FWSM traffic (see Figure B-2).

The Department networks are allowed to access the Internet and use PAT.

Figure B-2 Example 2



See the following sections for the configurations for this section:

FWSM Configuration (Example 2)

```
interface vlan 3
    nameif outside
    security-level 0
    ip address 209.165.201.3 255.255.255.224
interface vlan 4
    nameif dept2
    security-level 100
    ip address 10.1.2.1 255.255.255.0
interface vlan 5
    nameif dept1
    security-level 100
    ip address 10.1.1.1 255.255.255.0
interface vlan 10
    nameif dmz
    security-level 50
    ip address 192.168.2.1 255.255.255.0
passwd g00fba11
enable password gen1u$
hostname Buster
same-security-traffic permit inter-interface
route outside 0 0 209.165.201.1 1
nat (dept1) 1 10.1.1.0 255.255.255.0
nat (dept2) 1 10.1.2.0 255.255.255.0
! The dept1 and dept2 networks use PAT when accessing the outside
global (outside) 1 209.165.201.9 netmask 255.255.255.255
! Because we perform dynamic NAT on these addresses for outside access, we need to perform
! NAT on them for all other interface access. This identity static statement just
! translates the local address to the same address.
static (dept1,dept2) 10.1.1.0 10.1.1.0 netmask 255.255.255.0
static (dept2,dept1) 10.1.2.0 10.1.2.0 netmask 255.255.255.0
! The syslog server uses a static translation so the outside management host can access
! the server
static (dmz,outside) 209.165.201.5 192.168.2.2 netmask 255.255.255.255
access-list DEPTS remark -Allows all dept1 and dept2 hosts to access the
access-list DEPTS remark -outside for any IP traffic
access-list DEPTS extended permit ip any any
access-group DEPTS in interface dept1
access-group DEPTS in interface dept2
access-list MANAGE remark Allows the management host to access the syslog server
access-list MANAGE extended permit tcp host 209.165.200.225 host 209.165.201.5 eq telnet
access-group MANAGE in interface outside
! Advertises the FWSM IP address as the default gateway for the downstream
! router. FWSM does not advertise a default route to the router.
rip dept2 default version 2 authentication md5 scorpius 1
! Listens for RIP updates from the downstream router. FWSM does not
! listen for RIP updates from the router because a default route to the router is all that
! is required.
rip dept2 passive version 2 authentication md5 scorpius 1
! The client uses a pre-shared key to connect to the FWSM over IPSec. The
! key is the password in the username command following.
isakmp policy 1 authentication pre-share
isakmp policy 1 encryption 3des
isakmp policy 1 group 2
isakmp policy 1 hash sha
isakmp enable outside
crypto ipsec transform-set vpn_client esp-3des esp-sha-hmac
```

```
username admin password passw0rd
crypto ipsec transform-set vpn esp-3des esp-sha-hmac
crypto dynamic-map vpn_client 1 set transform-set vpn
crypto map telnet_tunnel 1 ipsec-isakmp dynamic vpn_client
crypto map telnet_tunnel interface outside
ip local pool client_pool 10.1.1.2
access-list VPN_SPLIT extended permit ip host 209.165.201.3 host 10.1.1.2
telnet 10.1.1.2 255.255.255.255 outside
telnet timeout 30
logging trap 5
! System log messages are sent to the syslog server on the DMZ network
logging host dmz 192.168.2.2
logging enable
```

Switch Configuration (Example 2)

The following lines in the switch configuration relate to the FWSM:

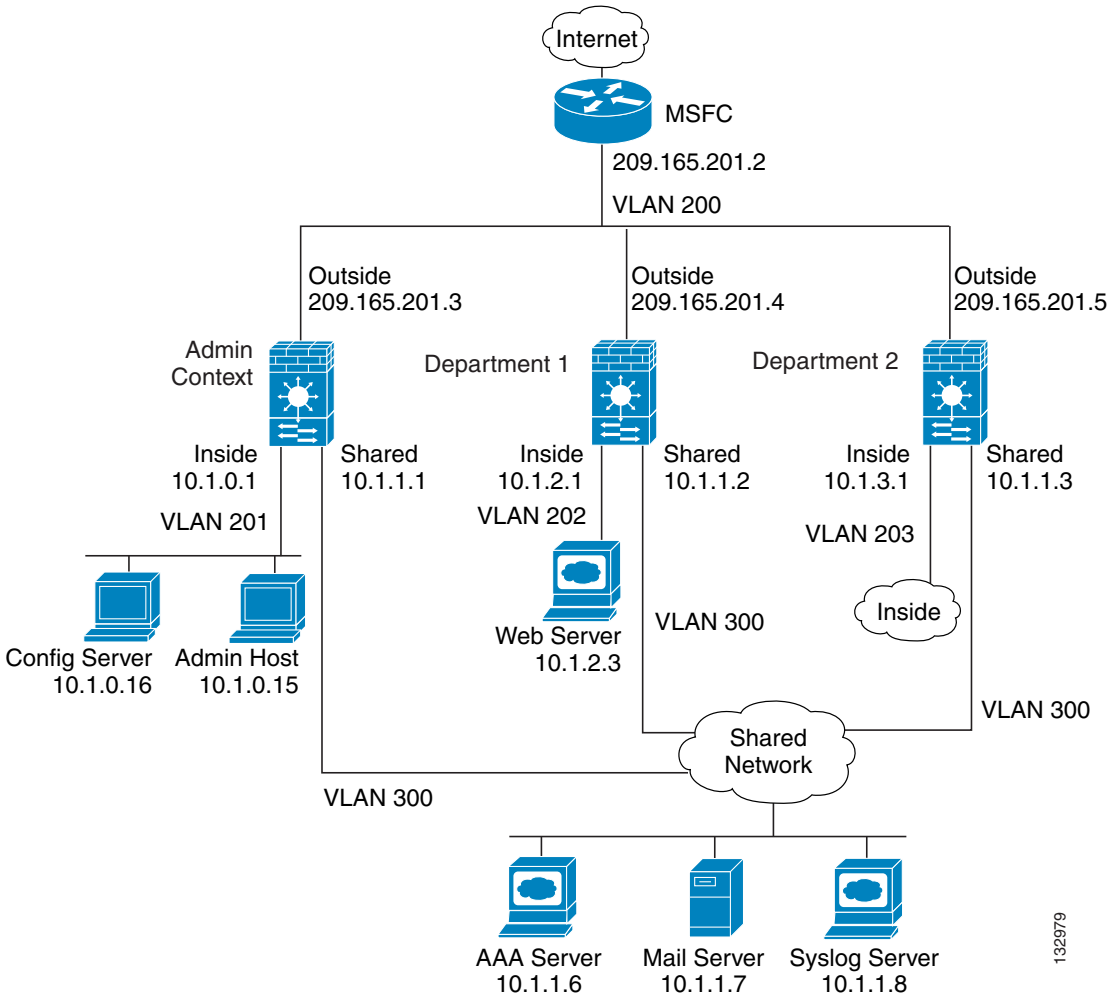
```
interface vlan 3
    ip address 209.165.201.1 255.255.255.224
    no shutdown
...
```

Example 3: Shared Resources for Multiple Contexts Example

The following configuration includes multiple contexts for multiple departments within a company. Each department has its own security context so that each department can have its own security policy. However, the syslog, mail, and AAA servers are shared across all departments. These servers are placed on a shared interface (see Figure B-3).

Department 1 has a web server that outside users who are authenticated by the AAA server can access.

Figure B-3 Example 3



See the following sections for the configurations for this scenario:

- System Configuration (Example 3), page B-9
- Admin Context Configuration (Example 3), page B-10
- Department 1 Context Configuration (Example 3), page B-11
- Department 2 Context Configuration (Example 3), page B-12
- Switch Configuration (Example 3), page B-12

System Configuration (Example 3)

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts using the **activation-key** command. The mode and the activation key are not stored in the configuration file, even though they endure reboots. If you view the configuration on the FWSM using the **write terminal**, **show startup-config**, or **show running-config** commands, the mode displays after the FWSM Release (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
hostname Ubik
password pkd55
enable password deckard69
interface vlan 200
interface vlan 201
interface vlan 202
interface vlan 203
interface vlan 300
admin-context admin
context admin
    allocate-interface vlan200
    allocate-interface vlan201
    allocate-interface vlan300
    config-url disk0://admin.cfg
context department1
    allocate-interface vlan200
    allocate-interface vlan202
    allocate-interface vlan300
    config-url ftp://admin:passw0rd@10.1.0.16/dept1.cfg
context department2
    allocate-interface vlan200
    allocate-interface vlan203
    allocate-interface vlan300
    config-url ftp://admin:passw0rd@10.1.0.16/dept2.cfg
```

Admin Context Configuration (Example 3)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface vlan 200
    nameif outside
    security-level 0
    ip address 209.165.201.3 255.255.255.224
interface vlan 201
    nameif inside
    security-level 100
    ip address 10.1.0.1 255.255.255.0
interface vlan 300
    nameif shared
    security-level 50
    ip address 10.1.1.1 255.255.255.0
passwd v00d00
enable password d011
route outside 0 0 209.165.201.2 1
nat (inside) 1 10.1.0.0 255.255.255.0
! This context uses PAT for inside users that access the outside
global (outside) 1 209.165.201.6 netmask 255.255.255.255
! This context uses PAT for inside users that access the shared network
global (shared) 1 10.1.1.30
! Because this host can access the web server in the Department 1 context, it requires a
! static translation
static (inside,outside) 209.165.201.7 10.1.0.15 netmask 255.255.255.255
! Because this host has management access to the servers on the Shared interface, it
! requires a static translation to be used in an access list
static (inside,shared) 10.1.1.78 10.1.0.15 netmask 255.255.255.255
access-list INTERNET remark -Allows all inside hosts to access the outside
access-list INTERNET remark -and shared network for any IP traffic
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
access-list SHARED remark -Allows only mail traffic from inside to exit shared interface
```

```
access-list SHARED remark -but allows the admin host to access any server.
access-list SHARED extended permit ip host 10.1.1.78 any
access-list SHARED extended permit tcp host 10.1.1.30 host 10.1.1.7 eq smtp
! Note that the translated addresses are used.
access-group SHARED out interface shared
! Allows 10.1.0.15 to access the admin context using Telnet. From the admin context, you
! can access all other contexts.
telnet 10.1.0.15 255.255.255.255 inside
aaa-server AAA-SERVER protocol tacacs+
aaa-server AAA-SERVER (shared) host 10.1.1.6
    key TheUauthKey
    server-port 16
! The host at 10.1.0.15 must authenticate with the AAA server to log in
aaa authentication telnet console AAA-SERVER
logging trap 6
! System log messages are sent to the syslog server on the Shared network
logging host shared 10.1.1.8
logging on
```

Department 1 Context Configuration (Example 3)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface vlan 200
    nameif outside
    security-level 0
    ip address 209.165.201.4 255.255.255.224
interface vlan 202
    nameif inside
    security-level 100
    ip address 10.1.2.1 255.255.255.0
interface vlan 300
    nameif shared
    security-level 50
    ip address 10.1.1.2 255.255.255.0
passwd cugel
enable password rhialto
nat (inside) 1 10.1.2.0 255.255.255.0
! The inside network uses PAT when accessing the outside
global (outside) 1 209.165.201.8 netmask 255.255.255.255
! The inside network uses dynamic NAT when accessing the shared network
global (shared) 1 10.1.1.31-10.1.1.37
! The web server can be accessed from outside and requires a static translation
static (inside,outside) 209.165.201.9 10.1.2.3 netmask 255.255.255.255
access-list INTERNET remark -Allows all inside hosts to access the outside
access-list INTERNET remark -and shared network for any IP traffic
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
access-list WEBSERVER remark -Allows the management host (its translated address) on the
access-list WEBSERVER remark -admin context to access the web server for management
access-list WEBSERVER remark -it can use any IP protocol
access-list WEBSERVER extended permit ip host 209.165.201.7 host 209.165.201.9
access-list WEBSERVER remark -Allows any outside address to access the web server
access-list WEBSERVER extended permit tcp any eq http host 209.165.201.9 eq http
access-group WEBSERVER in interface outside
access-list MAIL remark -Allows only mail traffic from inside to exit out the shared int
! Note that the translated addresses are used.
access-list MAIL extended permit tcp host 10.1.1.31 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.32 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.33 eq smtp host 10.1.1.7 eq smtp
```

```
access-list MAIL extended permit tcp host 10.1.1.34 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.35 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.36 eq smtp host 10.1.1.7 eq smtp
access-list MAIL extended permit tcp host 10.1.1.37 eq smtp host 10.1.1.7 eq smtp
access-group MAIL out interface shared
aaa-server AAA-SERVER protocol tacacs+
aaa-server AAA-SERVER (shared) host 10.1.1.6
    key TheUauthKey
    server-port 16
! All traffic matching the WEBSERVER access list must authenticate with the AAA server
aaa authentication match WEBSERVER outside AAA-SERVER
logging trap 4
! System log messages are sent to the syslog server on the Shared network
logging host shared 10.1.1.8
logging on
```

Department 2 Context Configuration (Example 3)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface vlan 200
    nameif outside
    security-level 0
    ip address 209.165.201.5 255.255.255.224
interface vlan 203
    nameif inside
    security-level 100
    ip address 10.1.3.1 255.255.255.0
interface vlan 300
    nameif shared
    security-level 50
    ip address 10.1.1.3 255.255.255.0
passwd maz1r1an
enable password ly0ne$$e
route outside 0 0 209.165.201.2 1
nat (inside) 1 10.1.3.0 255.255.255.0
! The inside network uses PAT when accessing the outside
global (outside) 1 209.165.201.10 netmask 255.255.255.255
! The inside network uses PAT when accessing the shared network
global (shared) 1 10.1.1.38
access-list INTERNET remark -Allows all inside hosts to access the outside
access-list INTERNET remark -and shared network for any IP traffic
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
access-list MAIL remark -Allows only mail traffic from inside to exit out the shared int
access-list MAIL extended permit tcp host 10.1.1.38 host 10.1.1.7 eq smtp
! Note that the translated PAT address is used.
access-group MAIL out interface shared
logging trap 3
! System log messages are sent to the syslog server on the Shared network
logging host shared 10.1.1.8
logging on
```

Switch Configuration (Example 3)

The following lines in the Cisco IOS switch configuration relate to the FWSM:

```
...
firewall module 6 vlan-group 1
```

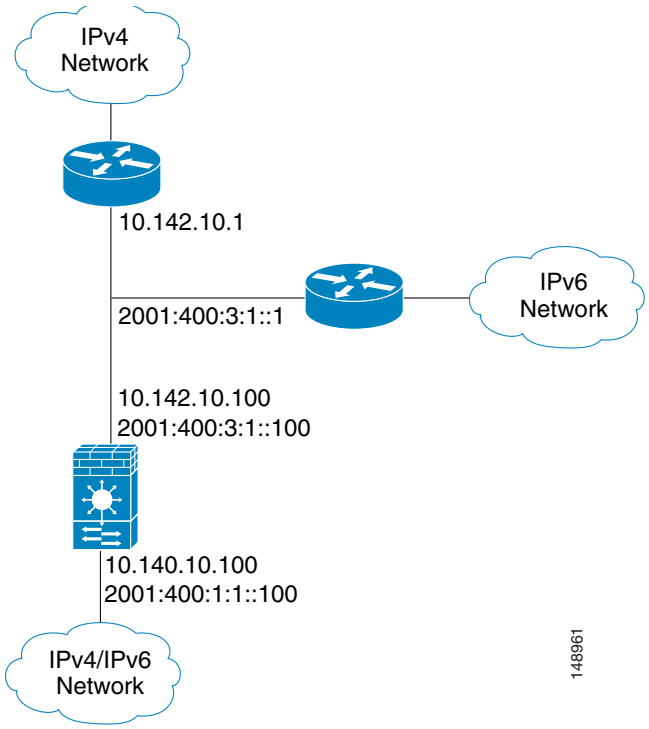
```
firewall vlan-group 1 200-203,300
interface vlan 200
    ip address 209.165.201.2 255.255.255.224
    no shutdown
...
```

Example 4: IPv6 Configuration Example

The following configuration (see Figure B-4) shows several features of IPv6 configured on the FWSM:

- Each interface is configured with both IPv6 and IPv4 addresses.
- The IPv6 default route is set with the **ipv6 route** command.
- An IPv6 access list is applied to the outside interface.

Figure B-4 Example 4: IPv4 and IPv6 Dual Stack Configuration




```
password pkd
enable password happy
hostname ubik
interface vlan 100
    nameif outside
    security-level 0
    ip address 10.142.10.100 255.255.255.0
    ipv6 address 2001:400:3:1::100/64
    ipv6 nd suppress-ra
interface vlan 101
    nameif inside
    security-level 100
    ip address 10.140.10.100 255.255.255.0
    ipv6 address 2001:400:1:1::100/64
```

```
route outside 0.0.0.0 0.0.0.0 10.142.10.1 1
access-list INTERNET remark -Allows all inside IPv4 hosts to access the outside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
ipv6 route outside ::/0 2001:400:3:1::1
ipv6 access-list IPV6INTERNET permit ip any any
access-group IPV6INTERNET in interface inside
ipv6 access-list OUTACL permit icmp6 2001:400:2:1::/64 2001:400:1:1::/64
ipv6 access-list OUTACL permit tcp 2001:400:2:1::/64 2001:400:1:1::/64 eq telnet
ipv6 access-list OUTACL permit tcp 2001:400:2:1::/64 2001:400:1:1::/64 eq ftp
ipv6 access-list OUTACL permit tcp 2001:400:2:1::/64 2001:400:1:1::/64 eq www
access-group OUTACL in interface outside
```

Transparent Mode Sample Configurations

This section includes the following topics:

Example 5: Multiple Mode, Transparent Firewall with Outside Access Example

The following configuration creates three security contexts plus the admin context. Each context allows OSPF traffic to pass between the inside and outside routers (see Figure B-5).

Also, DHCP packets can pass through the transparent firewall, because the transparent firewall does not support the DHCP relay feature.

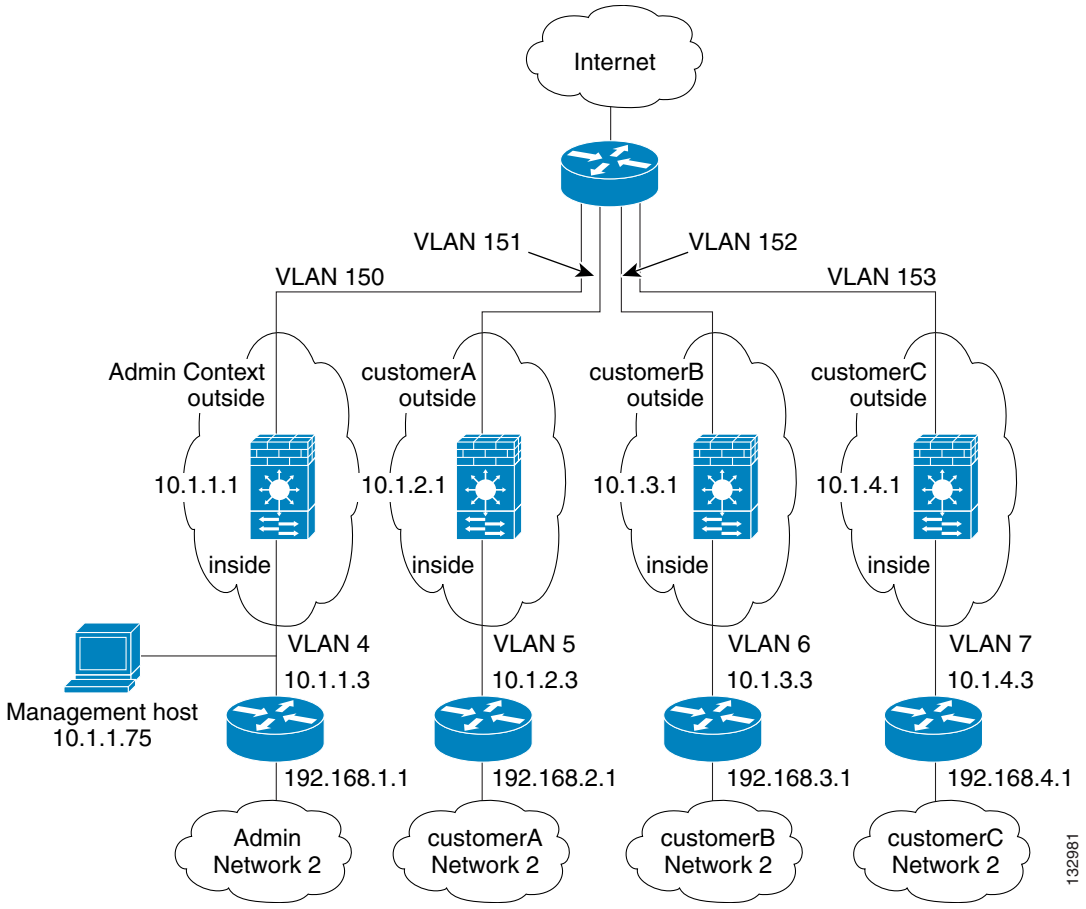
Inside hosts can access the Internet through the outside, but no outside hosts can access the inside.

The admin context allows SSH sessions to the FWSM from one host. It also uses ARP inspection to prevent IP spoofing of the upstream and downstream routers.

Each customer context belongs to a class that limits its resources (gold, silver, or bronze).

Although inside IP addresses can be the same across contexts, keeping them unique is easier to manage.

Figure B-5 Example 5



See the following sections for the configurations for this scenario:

- System Configuration (Example 5), page B-15
- Admin Context Configuration (Example 5), page B-16
- Customer A Context Configuration (Example 5), page B-17
- Customer B Context Configuration (Example 5), page B-17
- Customer C Context Configuration (Example 5), page B-18

System Configuration (Example 5)

You must first enable multiple context mode using the **mode multiple** command. The mode is not stored in the configuration file, even though it endures reboots. If you view the configuration on FWSM using the **write terminal**, **show startup-config**, or **show running-config** commands, the mode displays after the FWSM Release (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
hostname Farscape
password passw0rd
enable password chr1cht0n
interface vlan 4
interface vlan 5
interface vlan 6
```

```
interface vlan 7
interface vlan 150
interface vlan 151
interface vlan 152
interface vlan 153
admin-context admin
context admin
    allocate-interface vlan150
    allocate-interface vlan4
    config-url disk://admin.cfg
    member default
context customerA
    description This is the context for customer A
    allocate-interface vlan151
    allocate-interface vlan5
    config-url disk://contexta.cfg
    member gold
context customerB
    description This is the context for customer B
    allocate-interface vlan152
    allocate-interface vlan6
    config-url disk://contextb.cfg
    member silver
context customerC
    description This is the context for customer C
    allocate-interface vlan153
    allocate-interface vlan7
    config-url disk://contextc.cfg
    member bronze
class gold
    limit-resource all 7%
    limit-resource rate conns 2000
    limit-resource conns 20000
class silver
    limit-resource all 5%
    limit-resource rate conns 1000
    limit-resource conns 10000
class bronze
    limit-resource all 3%
    limit-resource rate conns 500
    limit-resource conns 5000
```

Admin Context Configuration (Example 5)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

The host at 10.1.1.75 can access the context using SSH, which requires a key pair to be generated using the **crypto key generate** command.

```
firewall transparent
passwd secret1969
enable password h1andl0
interface vlan 150
    nameif outside
    security-level 0
    bridge-group 1
interface vlan 4
    nameif inside
    security-level 100
    bridge-group 1
interface bvi 1
```

```
    ip address 10.1.1.1 255.255.255.0
route outside 0 0 10.1.1.2 1
ssh 10.1.1.75 255.255.255.255 inside
arp outside 10.1.1.2 0009.7cbe.2100
arp inside 10.1.1.3 0009.7cbe.1000
arp-inspection inside enable flood
arp-inspection outside enable flood
access-list INTERNET remark -Allows all inside hosts to access the outside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
access-list RETURN remark -Allows OSPF back
access-list RETURN extended permit 89 any any
access-list RETURN remark -Allows DHCP back
access-list RETURN extended permit udp any any eq 68
access-group RETURN in interface outside
```

Customer A Context Configuration (Example 5)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
firewall transparent
passwd hell0!
enable password enter55
interface vlan 151
    nameif outside
    security-level 0
    bridge-group 45
interface vlan 5
    nameif inside
    security-level 100
    bridge-group 45
interface bvi 45
    ip address 10.1.2.1 255.255.255.0
route outside 0 0 10.1.2.2 1
access-list INTERNET remark -Allows all inside hosts to access the outside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
access-list RETURN remark -Allows OSPF back
access-list RETURN extended permit 89 any any
access-list RETURN remark -Allows DHCP back
access-list RETURN extended permit udp any any eq 68
access-group RETURN in interface outside
```

Customer B Context Configuration (Example 5)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
firewall transparent
passwd tenac10us
enable password defen$e
interface vlan 152
    nameif outside
    security-level 0
    bridge-group 1
interface vlan 6
    nameif inside
    security-level 100
```

```
    bridge-group 1
interface bvi 1
    ip address 10.1.3.1 255.255.255.0
route outside 0 0 10.1.3.2 1
access-list INTERNET remark -Allows all inside hosts to access the outside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
access-list RETURN remark -Allows OSPF back
access-list RETURN extended permit 89 any any
access-list RETURN remark -Allows DHCP back
access-list RETURN extended permit udp any any eq 68
access-group RETURN in interface outside
```

Customer C Context Configuration (Example 5)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
firewall transparent
passwd fl0wer
enable password treeh0u$e
interface vlan 153
    nameif outside
    security-level 0
    bridge-group 100
interface vlan 7
    nameif inside
    security-level 100
    bridge-group 100
interface bvi 100
    ip address 10.1.4.1 255.255.255.0
route outside 0 0 10.1.4.2 1
access-list INTERNET remark -Allows all inside hosts to access the outside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
access-list RETURN remark -Allows OSPF back
access-list RETURN extended permit 89 any any
access-list RETURN remark -Allows DHCP back
access-list RETURN extended permit udp any any eq 68
access-group RETURN in interface outside
```

Failover Example Configurations

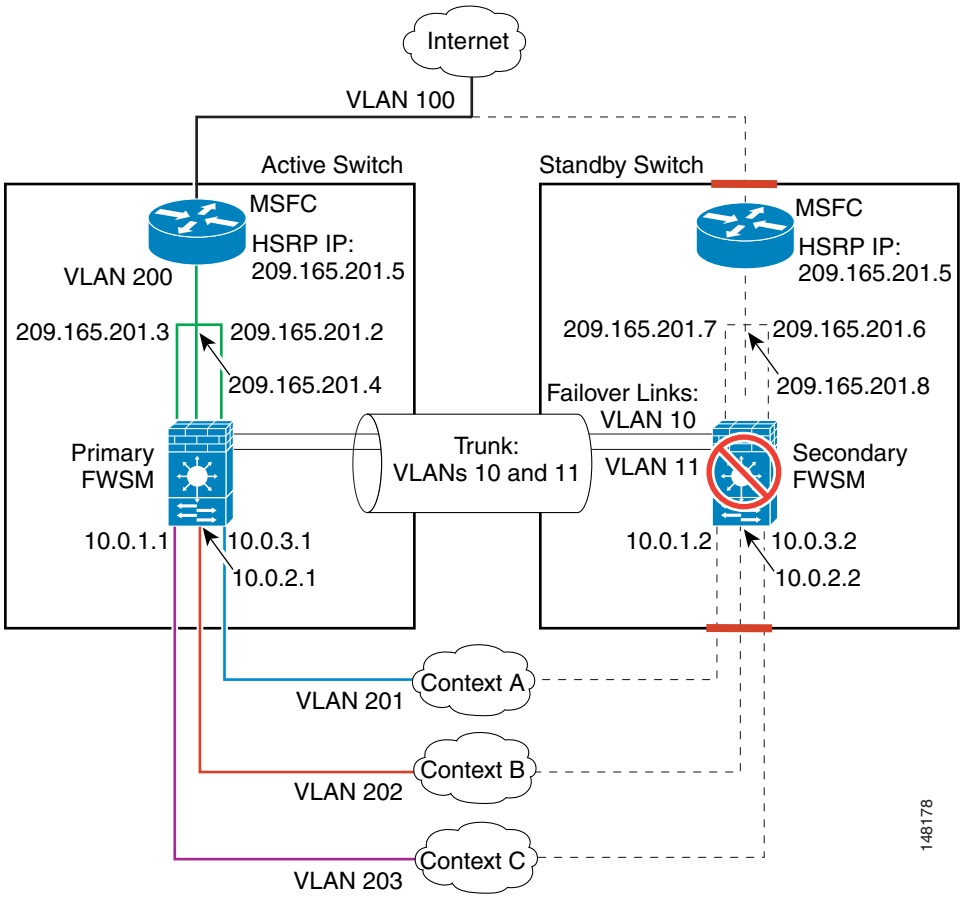
This section includes the following topics:

Example 6: Routed Mode Failover

The following configuration shows a multiple context mode FWSM with each context in routed mode in one switch, and another FWSM in a second switch acting as a backup (see Figure B-6). Each context (A, B, and C) monitors the inside interface, and context A, which is the admin context, also monitors the outside interface. Because the outside interface is shared among all contexts, monitoring in one context benefits all contexts.

The primary FWSM is configured with the **failover preempt** command, which causes it to become the active unit upon boot, even if the secondary unit is in the active state. The secondary FWSM is also in multiple context mode, and has the same software release.

Figure B-6 Example 6



See the following sections for the configurations for this scenario:

- Primary FWSM Configuration (Example 6), page B-19
- Secondary FWSM System Configuration (Example 6), page B-22
- Switch Configuration (Example 6), page B-22

Primary FWSM Configuration (Example 6)

The following sections include the configuration for the primary FWSM:

System Configuration (Primary Unit—Example 6)

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts using the **activation-key** command. The mode and the activation key are not stored in the configuration file, even though they do endure reboots. If you view the configuration on the FWSM using the **write terminal**, **show startup**, or **show running** commands, the mode displays after the FWSM Release (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
hostname primary
enable password farscape
password crichton
!The vlan 10 and 11 interfaces are created when you enter the failover lan interface and
failover link commands.
interface vlan 10
    description LAN Failover interface
interface vlan 11
    description STATE Failover interface
interface vlan 200
interface vlan 201
interface vlan 202
interface vlan 203
failover lan interface faillink vlan 10
failover link statelink vlan 11
failover lan unit primary
failover preempt 5
failover interface ip faillink 192.168.253.1 255.255.255.252 standby 192.168.253.2
failover interface ip statelink 192.168.253.5 255.255.255.252 standby 192.168.253.6
failover interface-policy 50%
failover replication http
failover
admin-context contexta
context contexta
    allocate-interface vlan200
    allocate-interface vlan201
    config-url disk://contexta.cfg
context contextb
    allocate-interface vlan200
    allocate-interface vlan202
    config-url ftp://admin:passw0rd@10.0.3.16/contextb.cfg
context contextc
    allocate-interface vlan200
    allocate-interface vlan203
    config-url ftp://admin:passw0rd@10.0.3.16/contextc.cfg
```

Context A Configuration (Primary Unit—Example 6)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface vlan 200
    nameif outside
    security-level 0
    ip address 209.165.201.2 255.255.255.224 standby 209.165.201.6
```

```
interface vlan 201
    nameif inside
    security-level 100
    ip address 10.0.3.1 255.255.255.0 standby 10.0.3.2
passwd secret1969
enable password h1andl0
monitor-interface inside
monitor-interface outside
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
global (outside) 1 209.165.201.10 netmask 255.255.255.224
! This context uses dynamic PAT for inside users that access the outside
route outside 0 0 209.165.201.5 1
telnet 10.0.3.75 255.255.255.255 inside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
! Allows all inside hosts to access the outside for any IP traffic
```

Context B Configuration (Primary Unit—Example 6)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface vlan 200
    nameif outside
    security-level 0
    ip address 209.165.201.4 255.255.255.224 standby 209.165.201.8
interface vlan 202
    nameif inside
    security-level 100
    ip address 10.0.2.1 255.255.255.0 standby 10.0.2.2
passwd secret1978
enable password 7samura1
monitor-interface inside
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
global (outside) 1 209.165.201.11 netmask 255.255.255.224
! This context uses dynamic PAT for inside users that access the outside
route outside 0 0 209.165.201.5 1
telnet 10.0.2.14 255.255.255.255 inside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
! Allows all inside hosts to access the outside for any IP traffic
```

Context C Configuration (Primary Unit—Example 6)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface vlan 200
    nameif outside
    security-level 0
    ip address 209.165.201.3 255.255.255.224 standby 209.165.201.7
interface vlan 203
    nameif inside
    security-level 100
    ip address 10.0.1.1 255.255.255.0 standby 10.0.1.2
passwd secret0997
enable password strayd0g
monitor-interface inside
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
global (outside) 1 209.165.201.12 netmask 255.255.255.224
! This context uses dynamic PAT for inside users that access the outside
```

```
route outside 0 0 209.165.201.5 1
telnet 10.0.1.65 255.255.255.255 inside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
! Allows all inside hosts to access the outside for any IP traffic
```

Secondary FWSM System Configuration (Example 6)

You do not need to configure any contexts, just the following minimal configuration for the system.

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts using the **activation-key** command. The mode and the activation key are not stored in the configuration file, even though they do endure reboots. If you view the configuration on the FWSM using the **write terminal**, **show startup**, or **show running** commands, the mode displays after the FWSM Release line (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
failover lan interface faillink vlan 10
failover interface ip faillink 192.168.253.1 255.255.255.252 standby 192.168.253.2
failover lan unit secondary
failover
```

Switch Configuration (Example 6)

The following lines in the Cisco IOS switch configuration on both switches relate to the FWSM. For information about configuring redundancy for the switch, see the switch documentation.

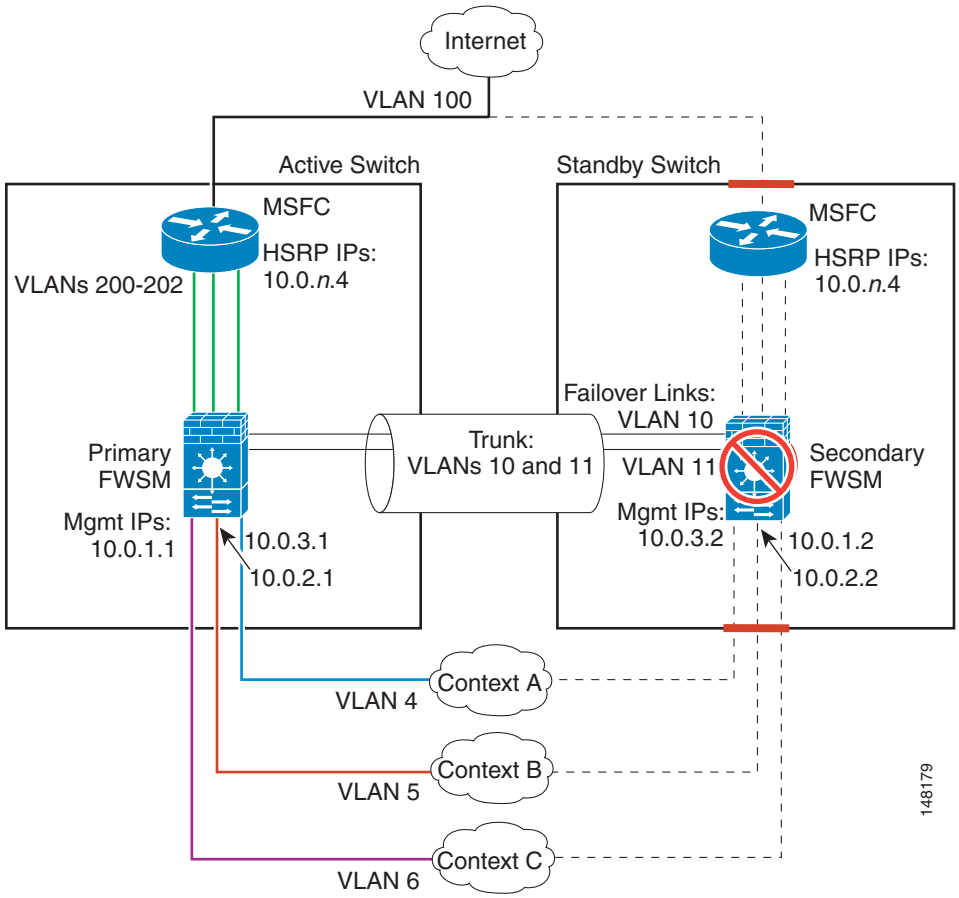
```
...
firewall module 1 vlan-group 1
firewall vlan-group 1 10,11,200-203
interface vlan 200
    ip address 209.165.201.1 255.255.255.224
    standby 200 ip 209.165.201.5
    standby 200 priority 110
    standby 200 preempt
    standby 200 timers 5 15
    standby 200 authentication Secret
    no shutdown
interface range gigabitethernet 2/1-3
    channel-group 2 mode on
    switchport trunk encapsulation dot1q
    no shutdown
...
```

Example 7: Transparent Mode Failover

The following configuration shows a multiple context mode FWSM with transparent mode contexts in one switch, and another FWSM in a second switch acting as a backup (see Figure B-6). Each context (A, B, and C) monitors the inside interface and outside interface.

The secondary FWSM is also in multiple context mode, and has the same software release.

Figure B-7 Example 7



See the following sections for the configurations for this scenario:

- Primary FWSM Configuration (Example 7), page B-23
- Secondary FWSM System Configuration (Example 7), page B-26
- Switch Configuration (Example 7), page B-26

Primary FWSM Configuration (Example 7)

The following sections include the configuration for the primary FWSM:

- System Configuration (Primary Unit—Example 7), page B-23
- Context A Configuration (Primary Unit—Example 7), page B-24
- Context B Configuration (Primary Unit—Example 7), page B-25
- Context C Configuration (Primary Unit—Example 7), page B-25

System Configuration (Primary Unit—Example 7)

You must first enable multiple context mode using the **mode multiple** command. Then enter the activation key to allow more than two contexts using the **activation-key** command. The mode and the activation key are not stored in the configuration file, even though they do endure reboots. If you view

the configuration on the FWSM using the **write terminal**, **show startup**, or **show running** commands, the mode displays after the FWSM Release (blank means single mode, "<system>" means you are in multiple mode in the system configuration, and <context> means you are in multiple mode in a context).

```
hostname primary
enable password farscape
password crichton
interface vlan 4
interface vlan 5
interface vlan 6
!The vlan 10 and 11 interfaces are created when you enter the failover lan interface and
failover link commands.
interface vlan 10
    description LAN Failover interface
interface vlan 11
    description STATE Failover interface
interface vlan 200
interface vlan 201
interface vlan 202
failover lan interface faillink vlan 10
failover link statelink vlan 11
failover lan unit primary
failover interface ip faillink 192.168.253.1 255.255.255.252 standby 192.168.253.2
failover interface ip statelink 192.168.253.5 255.255.255.252 standby 192.168.253.6
failover interface-policy 1
failover replication http
failover
admin-context contexta
context contexta
    allocate-interface vlan200
    allocate-interface vlan4
    config-url disk://contexta.cfg
context contextb
    allocate-interface vlan201
    allocate-interface vlan5
    config-url ftp://admin:passw0rd@10.0.3.16/contextb.cfg
context contextc
    allocate-interface vlan202
    allocate-interface vlan6
    config-url ftp://admin:passw0rd@10.0.3.16/contextc.cfg
```

Context A Configuration (Primary Unit—Example 7)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
firewall transparent
passwd secret1969
enable password h1andl0
interface vlan 200
    nameif outside
    security-level 0
    bridge-group 56
interface vlan 4
    nameif inside
    security-level 100
    bridge-group 56
interface bvi 56
    ip address 10.0.3.1 255.255.255.0 standby 10.0.3.2
monitor-interface inside
monitor-interface outside
route outside 0 0 10.0.3.4 1
```

```
telnet 10.0.3.75 255.255.255.255 inside
access-list INTERNET remark -Allows all inside hosts to access the outside for
access-list INTERNET remark -any IP traffic
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
access-list BPDU ethertype permit bpdu
access-group BPDU in interface inside
access-group BPDU in interface outside
```

Context B Configuration (Primary Unit—Example 7)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
firewall transparent
passwd secret1978
enable password 7samura1
interface vlan 201
    nameif outside
    security-level 0
    bridge-group 2
interface vlan 5
    nameif inside
    security-level 100
    bridge-group 2
interface bvi 2
ip address inside 10.0.2.1 255.255.255.0 standby 10.0.2.2
monitor-interface inside
monitor-interface outside
route outside 0 0 10.0.2.4 1
telnet 10.0.2.14 255.255.255.255 inside
access-list INTERNET remark -Allows all inside hosts to access the outside for
access-list INTERNET remark -any IP traffic
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
access-list BPDU ethertype permit bpdu
access-group BPDU in interface inside
access-group BPDU in interface outside
```

Context C Configuration (Primary Unit—Example 7)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
firewall transparent
passwd secret0997
enable password strayd0g
interface vlan 202
    nameif outside
    security-level 0
    bridge-group 1
interface vlan 6
    nameif inside
    security-level 100
    bridge-group 1
interface bvi 1
    ip address inside 10.0.1.1 255.255.255.0 standby 10.0.1.2
monitor-interface inside
monitor-interface outside
route outside 0 0 10.0.1.4 1
telnet 10.0.1.65 255.255.255.255 inside
```

```
access-list INTERNET remark -Allows all inside hosts to access the outside for
access-list INTERNET remark -any IP traffic
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
access-list BPDU ethertype permit bpdu
access-group BPDU in interface inside
access-group BPDU in interface outside
```

Secondary FWSM System Configuration (Example 7)

You do not need to configure any contexts, just the following minimal configuration for the system.

```
    failover lan interface faillink vlan 10
failover interface ip faillink 192.168.253.1 255.255.255.252 standby 192.168.253.2
failover lan unit secondary
failover
```

Switch Configuration (Example 7)

The following lines in the Cisco IOS switch configuration on both switches relate to the FWSM. For information about configuring redundancy for the switch, see the switch documentation.

```
...
firewall multiple-vlan-interfaces
firewall module 1 vlan-group 1
firewall vlan-group 1 4-6,10,11,200-202
interface vlan 200
    ip address 10.0.3.3 255.255.255.0
    standby 200 ip 10.0.1.4
    standby 200 priority 110
    standby 200 preempt
    standby 200 timers 5 15
    standby 200 authentication Secret
    no shutdown
interface vlan 201
    ip address 10.0.2.3 255.255.255.0
    standby 200 ip 10.0.2.4
    standby 200 priority 110
    standby 200 preempt
    standby 200 timers 5 15
    standby 200 authentication Secret
    no shutdown
interface vlan 202
    ip address 10.0.1.3 255.255.255.0
    standby 200 ip 10.0.3.4
    standby 200 priority 110
    standby 200 preempt
    standby 200 timers 5 15
    standby 200 authentication Secret
    no shutdown
interface range gigabitethernet 2/1-3
    channel-group 2 mode on
    switchport trunk encapsulation dot1q
    no shutdown
...
```

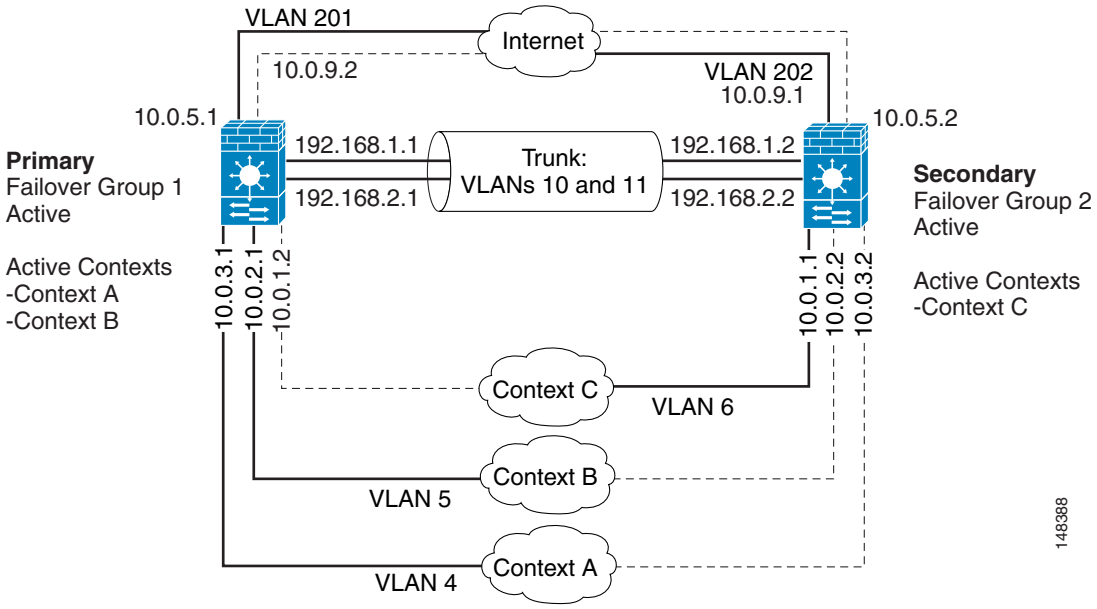
Example 8: Active/Active Failover with Asymmetric Routing Support

The following example shows how to configure Active/Active failover. In this example there are three contexts: Context A (the admin context), Context B, and Context C.

- The failover groups are configured with the **preempt** command.
- The admin context only has one interface.

Figure B-8 shows the network diagram for the example.

Figure B-8 Active/Active Failover Configuration



Prerequisites

Both units must be in multiple context mode. Use the **mode multiple** command to switch the primary and secondary FWSMs to multiple context mode. You must enter the **mode multiple** command on both the primary and secondary unit to change modes; the **mode multiple** command is not replicated to the secondary unit even in existing Active/Standby failover configurations.

Both FWSMs must be licensed for the same number of security contexts.

Primary FWSM Configuration (Example 8)

The following sections include the configuration for the primary FWSM:

- System Context Configuration (Primary FWSM—Example 8), page B-28
- Context A Configuration (Primary FWSM—Example 8), page B-28
- Context B Configuration (Primary FWSM—Example 8), page B-29
- Context C Configuration (Primary FWSM—Example 8), page B-29

System Context Configuration (Primary FWSM—Example 8)

The failover groups and the failover and Stateful Failover VLANs are configured in the system context.

```
hostname cisco-primary
enable password farscape
password crichton
interface vlan 4
interface vlan 5
interface vlan 6
!The vlan 10 and 11 interfaces are created when you enter the failover lan interface and
failover link commands.
interface vlan 10
    description LAN Failover interface
interface vlan 11
    description STATE Failover interface
interface vlan 201
interface vlan 202
failover
failover lan unit primary
failover lan interface faillink vlan 10
failover key MySecretKey
failover link statelink vlan 11
failover interface ip faillink 192.168.1.1 255.255.255.0 standby 192.168.1.2
failover interface ip statelink 192.168.2.1 255.255.255.0 standby 192.168.2.2
failover group 1
    preempt
    replication http
    interface-policy 50%
failover group 2
    secondary
    preempt
    replication http
    interface-policy 50%
admin-context contexta
context contexta
    description administrative context
    allocate-interface vlan4
    config-url disk://contexta.cfg
    join-failover-group 1
context contextb
    allocate-interface vlan201
    allocate-interface vlan5
    config-url ftp://admin:passw0rd@10.0.3.16/contextb.cfg
    join-failover-group 1
context contextc
    allocate-interface vlan202
    allocate-interface vlan6
    config-url ftp://admin:passw0rd@10.0.3.16/contextc.cfg
    join-failover-group 2
```

Context A Configuration (Primary FWSM—Example 8)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

Context A is the admin context. In this example the admin context contains only one interface, the inside interface, for administrative access. Because the context contains only one interface, you cannot use Telnet to access the FWSM through the interface. Telnet access is not permitted to the lowest security level interface in a context, and because Context A has only one interface, it is the lowest level interface by default. Instead, you must define an SSH connection to manage the FWSM through this interface.

```
interface vlan 4
    nameif mgmt
    security-level 5
    ip address 10.0.3.1 255.255.255.0 standby 10.0.3.2
passwd secret1969
enable password h1andl0
monitor-interface inside
crypto key generate rsa modulus 1024
ssh 10.0.3.0 255.255.255.0 inside
ssh version 2
```

Context B Configuration (Primary FWSM—Example 8)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface vlan 201
    nameif outside
    security-level 0
    ip address 10.0.5.1 255.255.255.0 standby 10.0.5.2
    asr-group 1
interface vlan 5
    nameif inside
    security-level 100
    ip address 10.0.2.1 255.255.255.0 standby 10.0.2.2
passwd secret1978
enable password 7samura1
monitor-interface inside
monitor-interface outside
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
global (outside) 1 10.0.5.1 netmask 255.255.255.0
! This context uses dynamic PAT for inside users that access the outside
route outside 0 0 10.0.5.5 1
telnet 10.0.2.14 255.255.255.255 inside
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
! Allows all inside hosts to access the outside for any IP traffic
```

Context C Configuration (Primary FWSM—Example 8)

To change to a context configuration, enter the **changeto context** *name* command. To change back to the system, enter **changeto system**.

```
interface vlan 202
    nameif outside
    security-level 0
    ip address 10.0.9.1 255.255.255.224 standby 10.0.9.2
    asr-group 1
interface vlan 6
    nameif inside
    security-level 100
    ip address 10.0.1.1 255.255.255.0 standby 10.0.1.2
passwd secret0997
enable password strayd0g
monitor-interface inside
monitor-interface outside
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
global (outside) 1 10.0.9.1 netmask 255.255.255.0
! This context uses dynamic PAT for inside users that access the outside
route outside 0 0 10.0.9.5 1
telnet 10.0.1.65 255.255.255.255 inside
```

```
access-list INTERNET extended permit ip any any
access-group INTERNET in interface inside
! Allows all inside hosts to access the outside for any IP traffic
```

The Secondary FWSM Configuration (Example 8)

You only need to configure the secondary FWSM to recognize the failover link. The secondary FWSM obtains the context configurations from the primary FWSM upon booting or when **failover** is first enabled. The **preempt** commands in the failover group configurations cause the failover groups to become active on their designated unit after the configurations have been synchronized and the preempt delay has passed.

Note that you must configure the **failover key** command on the secondary FWSM so that it can receive the configuration from the primary FWSM.

```
failover
failover lan unit secondary
failover lan interface faillink vlan 10
failover key MySecretKey
failover interface ip faillink 192.168.253.1 255.255.255.252 standby 192.168.253.2
```

When you enable failover with the **failover** command, the secondary FWSM obtains the configuration from the primary FWSM.

Switch Configuration (Example 8)

The following lines in the Cisco IOS switch configuration on both switches relate to the FWSM. For information about configuring redundancy for the switch, see the switch documentation.

```
...
firewall multiple-vlan-interfaces
firewall module 1 vlan-group 1
firewall vlan-group 1 4-6,10,11,201,202
interface vlan 201
    ip address 10.0.5.3 255.255.255.0
    standby 200 ip 10.0.5.4
    standby 200 priority 110
    standby 200 preempt
    standby 200 timers 5 15
    standby 200 authentication Secret
    no shutdown
interface vlan 202
    ip address 10.0.9.3 255.255.255.0
    standby 200 ip 10.0.9.4
    standby 200 priority 110
    standby 200 preempt
    standby 200 timers 5 15
    standby 200 authentication Secret
    no shutdown
interface range gigabitethernet 2/1-3
    channel-group 2 mode on
    switchport trunk encapsulation dot1q
    no shutdown
...
```

APPENDIX C

Using the Command-Line Interface

This appendix describes how to use the CLI on the FWSM, and includes the following sections:

Note The CLI uses similar syntax and other conventions to the Cisco IOS CLI, but the FWSM operating system is not a version of Cisco IOS software. Do not assume that a Cisco IOS CLI command works with or has the same function on the FWSM.

Firewall Mode and Security Context Mode

The FWSM runs in a combination of the following modes:

- Transparent firewall or routed firewall mode

  The firewall mode determines if the FWSM runs as a Layer 2 or Layer 3 firewall.

- Multiple context or single context mode

  The security context mode determines if the FWSM runs as a single device or as multiple security contexts, which act like virtual devices.

Some commands are only available in certain modes.

Command Modes and Prompts

The FWSM CLI includes command modes. Some commands can only be entered in certain modes. For example, to enter commands that show sensitive information, you need to enter a password and enter a more privileged mode. Then, to ensure that configuration changes are not entered accidentally, you have to enter a configuration mode. All lower commands can be entered in higher modes, for example, you can enter a privileged EXEC command in global configuration mode.

When you are in the system configuration or in single context mode, the prompt begins with the hostname:

```
hostname
```

When you are within a context, the prompt begins with the hostname followed by the context name:

```
hostname/context
```

The prompt changes depending on the access mode:

- User EXEC mode

  User EXEC mode lets you see minimum FWSM settings. The user EXEC mode prompt appears as follows when you first access the FWSM:

  ```
  hostname>

  hostname/context>
  ```

- Privileged EXEC mode

  Privileged EXEC mode lets you see all current settings up to your privilege level. Any user EXEC mode command will work in privileged EXEC mode. Enter the **enable** command in user EXEC mode, which requires a password, to start privileged EXEC mode. The prompt includes the number sign (#):

  ```
  hostname#

  hostname/context#
  ```

- Global configuration mode

  Global configuration mode lets you change the FWSM configuration. All user EXEC, privileged EXEC, and global configuration commands are available in this mode. Enter the **configure terminal** command in privileged EXEC mode to start global configuration mode. The prompt changes to the following:

  ```
  hostname(config)#

  hostname/context(config)#
  ```

- Command-specific configuration modes

  From global configuration mode, some commands enter a command-specific configuration mode. All user EXEC, privileged EXEC, global configuration, and command-specific configuration commands are available in this mode. For example, the **interface** command enters interface configuration mode. The prompt changes to the following:

  ```
  hostname(config-if)#

  hostname/context(config-if)#
  ```

Syntax Formatting

Command syntax descriptions use the following conventions:

Table C-1 Syntax Conventions

Convention	Description
bold	Bold text indicates commands and keywords that you enter literally as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical bar indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Abbreviating Commands

You can abbreviate most commands down to the fewest unique characters for a command; for example, you can enter **`wr t`** to view the configuration instead of entering the full command **`write terminal`**, or you can enter **`en`** to start privileged mode and **`conf t`** to start configuration mode. In addition, you can enter **`0`** to represent **`0.0.0.0`**.

Command-Line Editing

The FWSM uses the same command-line editing conventions as Cisco IOS software. You can view all previously entered commands with the **show history** command or individually with the up arrow or **^p** command. Once you have examined a previously entered command, you can move forward in the list with the down arrow or **^n** command. When you reach a command you wish to reuse, you can edit it or press the **Enter** key to start it. You can also delete the word to the left of the cursor with **^w**, or erase the line with **^u**.

The FWSM permits up to 512 characters in a command; additional characters are ignored.

Command Completion

To complete a command or keyword after entering a partial string, press the **Tab** key. The FWSM only completes the command or keyword if the partial string matches only one command or keyword. For example, if you enter **s** and press the **Tab** key, the FWSM does not complete the command because it matches more than one command. However, if you enter **dis**, the **Tab** key completes the command **disable**.

Command Help

Help information is available from the command line by entering the following commands:

- **help** *command_name*

  Shows help for the specific command.

- **help ?**

  Shows commands for which there is help.

- *command_name* **?**

  Shows a list of arguments available.

- *string***?** (no space)

  Lists the possible commands that start with the string.

- **?** and **+?**

  Lists all commands available. If you enter **?**, the FWSM shows only commands available for the current mode. To show all commands available, including those for lower modes, enter **+?**.

Note If you want to include a question mark (?) in a command string, you must press **Ctrl-V** before typing the question mark so you do not inadvertently invoke CLI help.

Filtering show Command Output

You can use the vertical bar (|) with any **show** command and include a filter option and filtering expression. The filtering is performed by matching each output line with a regular expression, similar to Cisco IOS software. By selecting different filter options you can include or exclude all output that matches the expression. You can also display all output beginning with the line that matches the expression.

The syntax for using filtering options with the **show** command is as follows:

```
hostname# show command | {include | exclude | begin | grep [-v]} regexp
```

In this command string, the first vertical bar (|) is the operator and must be included in the command. This operator directs the output of the **show** command to the filter. In the syntax diagram, the other vertical bars (|) indicate alternative options and are not part of the command.

The **include** option includes all output lines that match the regular expression. The **grep** option without **-v** has the same effect. The **exclude** option excludes all output lines that match the regular expression. The **grep** option with **-v** has the same effect. The **begin** option shows all the output lines starting with the line that matches the regular expression.

Replace *regexp* with any Cisco IOS regular expression. See The regular expression is not enclosed in quotes or double-quotes, so be careful with trailing white spaces, which will be taken as part of the regular expression.

When creating regular expressions, you can use any letter or number that you want to match. In addition, certain keyboard characters have special meaning when used in regular expressions. Table C-2 lists the keyboard characters that have special meaning.

Table C-2 Using Special Characters in Regular Expressions

Character Type	Character	Special Meaning
period	.	Matches any single character, including white space.
asterisk	*	Matches 0 or more sequences of the pattern.
plus sign	+	Matches 1 or more sequences of the pattern.
question mark	?[1]	Matches 0 or 1 occurrences of the pattern.
caret	^	Matches the beginning of the input string.
dollar sign	$	Matches the end of the input string.
underscore	_	Matches a comma (,), left brace ({), right brace (}), left parenthesis, right parenthesis, the beginning of the input string, the end of the input string, or a space.
brackets	[]	Designates a range of single-character patterns.
hyphen	-	Separates the end points of a range.

1. Precede the question mark with **Ctrl-V** to prevent the question mark from being interpreted as a help command.

To use these special characters as single-character patterns, remove the special meaning by preceding each character with a backslash (\).

Command Output Paging

On commands such as **help** or**?**, **show**, **show xlate**, or other commands that provide long listings, you can determine whether the information displays a screen and pauses, or lets the command run to completion. The **pager** command lets you choose the number of lines to display before the More prompt appears.

When paging is enabled, the following prompt appears:

```
<--- More --->
```

The More prompt uses syntax similar to the UNIX **more** command:

- To view another screen, press the Space bar.
- To view the next line, press the **Enter** key.
- To return to the command line, press the **q** key.

Adding Comments

You can precede a line with a colon (:) to create a comment. However, the comment only appears in the command history buffer and not in the configuration. Therefore, you can view the comment with the **show history** command or by pressing an arrow key to retrieve a previous command, but because the comment is not in the configuration, the **write terminal** command does not display it.

Text Configuration Files

This section describes how to format a text configuration file that you can download to the FWSM, and includes the following topics:

How Commands Correspond with Lines in the Text File

The text configuration file includes lines that correspond with the commands described in this guide.

In examples, commands are preceded by a CLI prompt. The prompt in the following example is "hostname(config)#":

```
hostname(config)# context a
```

In the text configuration file you are not prompted to enter commands, so the prompt is omitted:

```
context a
```

Command-Specific Configuration Mode Commands

Command-specific configuration mode commands appear indented under the main command when entered at the command line. Your text file lines do not need to be indented, as long as the commands appear directly following the main command. For example, the following unindented text is read the same as indented text:

```
interface gigabitethernet0
nameif inside
interface gigabitethernet1
    nameif outside
```

Automatic Text Entries

When you download a configuration to the FWSM, the FWSM inserts some lines automatically. For example, the FWSM inserts lines for default settings or for the time the configuration was modified. You do not need to enter these automatic entries when you create your text file.

Line Order

For the most part, commands can be in any order in the file. However, some lines, such as ACEs, are processed in the order they appear, and the order can affect the function of the access list. Other commands might also have order requirements. For example, you must enter the **nameif** command for an interface first because many subsequent commands use the name of the interface. Also, commands in a command-specific configuration mode must directly follow the main command.

Commands Not Included in the Text Configuration

Some commands do not insert lines in the configuration. For example, a runtime command such as **show running-config** does not have a corresponding line in the text file.

Passwords

The login, enable, and user passwords are automatically encrypted before they are stored in the configuration. For example, the encrypted form of the password “cisco” might look like jMorNbK0514fadBh. You can copy the configuration passwords to another FWSM in their encrypted form, but you cannot unencrypt the passwords yourself.

If you enter an unencrypted password in a text file, the FWSM does not automatically encrypt them when you copy the configuration to the FWSM. The FWSM only encrypts them when you save the running configuration from the command line using the **copy running-config startup-config** or **write memory** command.

Multiple Security Context Files

For multiple security contexts, the entire configuration consists of multiple parts:

- The security context configurations
- The system configuration, which identifies basic settings for the FWSM, including a list of contexts
- The admin context, which provides network interfaces for the system configuration

  The system configuration does not include any interfaces or network settings for itself. Rather, when the system needs to access network resources (such as downloading the contexts from the server), it uses a context that is designated as the admin context.

Each context is similar to a single context mode configuration. The system configuration differs from a context configuration in that the system configuration includes system-only commands (such as a list of all contexts), while other typical commands are not present (such as many interface parameters).

APPENDIX D

Mapping MIBs to CLI Commands

Table D-1 lists MIB details, MIB objects, and the equivalent CLI commands.

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands

MIB Details	MIB Objects	CLI Field or Description
CISCO-CRYPTO-ACCELERATOR-MIB	—	show crypto accelerator statistics
1.3.6.1.4.1.9.9.467.1.1.1.	ccaSupportsHwCrypto	See this MIB for an explanation of the objects.
1.3.6.1.4.1.9.9.467.1.1.2.	ccaSupportsModularHwCrypto	
1.3.6.1.4.1.9.9.467.1.1.3.	ccaMaxAccelerators	
1.3.6.1.4.1.9.9.467.1.1.4.	ccaMaxCryptoThroughput	
1.3.6.1.4.1.9.9.467.1.1.5.	ccaMaxCryptoConnections	
1.3.6.1.4.1.9.9.467.1.2.1.1.	ccaGlobalNumActiveAccelerators	
1.3.6.1.4.1.9.9.467.1.2.1.2.	ccaGlobalNumNonOperAccelerators	
1.3.6.1.4.1.9.9.467.1.2.1.3.	ccaGlobalInOctets	
1.3.6.1.4.1.9.9.467.1.2.1.4.	ccaGlobalOutOctets	
1.3.6.1.4.1.9.9.467.1.2.1.5.	ccaGlobalInPkts	
1.3.6.1.4.1.9.9.467.1.2.1.6.	ccaGlobalOutPkts	
1.3.6.1.4.1.9.9.467.1.2.1.7.	ccaGlobalOutErrPkts	
1.3.6.1.4.1.9.9.467.1.2.2	ccaAcceleratorTable	
—	**Index** • ccaAcclIndex	Information and statistics about each crypto accelerator card
1.3.6.1.4.1.9.9.467.1.2.2.1.2.	ccaAcclEntPhysicalIndex	—
1.3.6.1.4.1.9.9.467.1.2.2.1.3.	ccaAcclStatus	—
1.3.6.1.4.1.9.9.467.1.2.2.1.4.	ccaAcclType	—
1.3.6.1.4.1.9.9.467.1.2.2.1.5.	ccaAcclVersion	—
1.3.6.1.4.1.9.9.467.1.2.2.1.6.	ccaAcclSlot	—
1.3.6.1.4.1.9.9.467.1.2.2.1.7.	ccaAcclActiveTime	—
1.3.6.1.4.1.9.9.467.1.2.2.1.8.	ccaAcclInPkts	—
1.3.6.1.4.1.9.9.467.1.2.2.1.9.	ccaAcclOutPkts	—

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.467.1.2.2.1.10.	ccaAcclOutBadPkts	—
1.3.6.1.4.1.9.9.467.1.2.2.1.11.	ccaAcclInOctets	—
1.3.6.1.4.1.9.9.467.1.2.2.1.12.	ccaAcclOutOctets	—
1.3.6.1.4.1.9.9.467.1.2.2.1.13.	ccaAcclHashOutboundPkts	—
1.3.6.1.4.1.9.9.467.1.2.2.1.14.	ccaAcclHashOutboundOctets	—
1.3.6.1.4.1.9.9.467.1.2.2.1.15.	ccaAcclHashInboundPkts	—
1.3.6.1.4.1.9.9.467.1.2.2.1.16.	ccaAcclHashInboundOctets	—
1.3.6.1.4.1.9.9.467.1.2.2.1.17.	ccaAcclEncryptPkts	—
1.3.6.1.4.1.9.9.467.1.2.2.1.18.	ccaAcclEncryptOctets	—
1.3.6.1.4.1.9.9.467.1.2.2.1.19.	ccaAcclDecryptPkts	—
1.3.6.1.4.1.9.9.467.1.2.2.1.20.	ccaAcclDecryptOctets	—
1.3.6.1.4.1.9.9.467.1.2.2.1.21.	ccaAcclTransformsTotal	—
1.3.6.1.4.1.9.9.467.1.2.2.1.22.	ccaAcclDropsPkts	—
1.3.6.1.4.1.9.9.467.1.2.2.1.23.	ccaAcclRandRequests	—
1.3.6.1.4.1.9.9.467.1.2.2.1.24.	ccaAcclRandReqFails	—
1.3.6.1.4.1.9.9.467.1.2.2.1.25.	ccaAcclDHKeysGenerated	—
1.3.6.1.4.1.9.9.467.1.2.2.1.26.	ccaAcclDHDerivedSecretKeys	—
1.3.6.1.4.1.9.9.467.1.2.2.1.27.	ccaAcclRSAKeysGenerated	—
1.3.6.1.4.1.9.9.467.1.2.2.1.28.	ccaAcclRSASignings	—
1.3.6.1.4.1.9.9.467.1.2.2.1.29.	ccaAcclRSAVerifications	—
1.3.6.1.4.1.9.9.467.1.2.2.1.30.	ccaAcclRSAEncryptPkts	—
1.3.6.1.4.1.9.9.467.1.2.2.1.31.	ccaAcclRSAEncryptOctets	—
1.3.6.1.4.1.9.9.467.1.2.2.1.32.	ccaAcclRSADecryptPkts	—
1.3.6.1.4.1.9.9.467.1.2.2.1.33.	ccaAcclRSADecryptOctets	—
1.3.6.1.4.1.9.9.467.1.2.2.1.34.	ccaAcclDSAKeysGenerated	—
1.3.6.1.4.1.9.9.467.1.2.2.1.35.	ccaAcclDSASignings	—
1.3.6.1.4.1.9.9.467.1.2.2.1.36.	ccaAcclDSAVerifications	—
1.3.6.1.4.1.9.9.467.1.2.2.1.37.	ccaAcclOutboundSSLRecords	—
1.3.6.1.4.1.9.9.467.1.2.2.1.38.	ccaAcclInboundSSLRecords	—
1.3.6.1.4.1.9.9.467.1.2.3.1	ccaProtocolStatsTable	Crypto accelerator statistics according to security protocols
—	**Index** • ccaProtId	—
1.3.6.1.4.1.9.9.467.1.2.3.1.1.2.	ccaProtPktEncryptsReqs	—
1.3.6.1.4.1.9.9.467.1.2.3.1.1.3.	ccaProtPktDecryptsReqs	—
1.3.6.1.4.1.9.9.467.1.2.3.1.1.4.	ccaProtHmacCalcReqs	—

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.467.1.2.3.1.1.5.	ccaProtSaCreateReqs	—
1.3.6.1.4.1.9.9.467.1.2.3.1.1.6.	ccaProtSaRekeyReqs	—
1.3.6.1.4.1.9.9.467.1.2.3.1.1.7.	ccaProtSaDeleteReqs	—
1.3.6.1.4.1.9.9.467.1.2.3.1.1.8.	ccaProtPktEncapReqs	—
1.3.6.1.4.1.9.9.467.1.2.3.1.1.9.	ccaProtPktDecapReqs	—
1.3.6.1.4.1.9.9.467.1.2.3.1.1.10.	ccaProtNextPhaseKeyAllocReqs	—
1.3.6.1.4.1.9.9.467.1.2.3.1.1.11.	ccaProtRndGenReqs	—
1.3.6.1.4.1.9.9.467.1.2.3.1.1.12.	ccaProtFailedReqs	—
CISCO-ENTITY-ALARM-MIB	—	—
1.3.6.1.4.1.9.9.138.2.0.1	ceAlarmAsserted	A physical entity asserts an alarm.
CISCO-ENTITY-REDUNDANCY-MIB	—	—
1.3.6.1.4.1.9.9.480.0.2	ceRedunEventSwitchover	The status of the standby unit changes to active.
CISCO-FIREWALL-MIB	—	—
1.3.6.1.4.1.9.9.147.1.1.1.1.	cfwBasicEventsTableLastRow	Not applicable (placeholder only)
1.3.6.1.4.1.9.9.147.1.1.2.1.	cfwNetEventsTableLastRow	Not applicable (placeholder only)
1.3.6.1.4.1.9.9.147.1.2.1.1	cfwHardwareStatusTable	—
—	**Index** • cfwHardwareType	 Primary unit/Secondary unit
1.3.6.1.4.1.9.9.147.1.2.1.1.1.2.	cfwHardwareInformation	Description of the resource
1.3.6.1.4.1.9.9.147.1.2.1.1.1.3.	cfwHardwareStatusValue	Current status of the resource. Applies only to a failover pair. Does not apply to a standalone blade.
1.3.6.1.4.1.9.9.147.1.2.1.1.1.4.	cfwHardwareStatusDetail	Description of the resource status
1.3.6.1.4.1.9.9.147.1.2.2.1	cfwBufferStatsTable	—
—	**Index** • cfwBufferStatSize • cfwBufferStatType	 Buffer size Buffer statistics type
1.3.6.1.4.1.9.9.147.1.2.2.1.1.3.	cfwBufferStatInformation	Description of the buffer statistics
1.3.6.1.4.1.9.9.147.1.2.2.1.1.4.	cfwBufferStatValue	Buffer statistics value
1.3.6.1.4.1.9.9.147.1.2.2.2.	cfwConnectionStatTable	—
—	**Index** • cfwConnectionStatService • cfwConnectionStatType	 protoIP currentInUse/high
1.3.6.1.4.1.9.9.147.1.2.2.2.1.3.	cfwConnectionStatDescription	Description of the connection statistics
1.3.6.1.4.1.9.9.147.1.2.2.2.1.4.	cfwConnectionStatCount	Not applicable (placeholder only)

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.147.1.2.2.2.1.5.	cfwConnectionStatValue	Connection statistics value
CISCO-IP-FORWARD-MIB	inetCidrRouteTable	show route
—	**Index** • inetCidrRouteDestType • inetCidrRouteDest • inetCidrRoutePolicy • inetCidrRouteNextHopType • inetCidrRouteNextHop	—
1.3.6.1.2.1.4.24.7.1.1	inetCidrRouteDestType	Route destination address type
1.3.6.1.2.1.4.24.7.1.2	inetCidrRouteDest	Route destination
1.3.6.1.2.1.4.24.7.1.3	inetCidrRoutePfxLen	Route mask
1.3.6.1.2.1.4.24.7.1.4	inetCidrRoutePolicy	Route policy, if any
1.3.6.1.2.1.4.24.7.1.5	inetCidrRouteNextHopType	Route next hop type (gateway)
1.3.6.1.2.1.4.24.7.1.6	inetCidrRouteNextHop	Route next hop
1.3.6.1.2.1.4.24.7.1.7	inetCidrRouteIfIndex	Interface through which routed
1.3.6.1.2.1.4.24.7.1.9	inetCidrRouteProto	Routed protocol
1.3.6.1.2.1.4.24.7.1.10	inetCidrRouteAge	Route age
1.3.6.1.2.1.4.24.7.1.12	inetCidrRouteMetric1	Route metric
1.3.6.1.2.1.4.24.7.1.17	inetCidrRouteStatus	Route status
CISCO-IP-PROTOCOL-FILTER-MIB	cippfIpFilterTable	show run access-list
—	**Index** • cippfIpProfileName • cippfIpFilterIndex	—
1.3.6.1.4.1.9.9.278.1.1.1.1.1	cippfIpProfileName	ACL name
1.3.6.1.4.1.9.9.278.1.1.3.1.1	cippfIpFilterIndex	ACE line number
1.3.6.1.4.1.9.9.278.1.1.3.1.3	cippfIpFilterAction	Permit/Deny
1.3.6.1.4.1.9.9.278.1.1.3.1.4	cippfIpFilterAddressType	Either IPv4 or IPv6
1.3.6.1.4.1.9.9.278.1.1.3.1.5	cippfIpFilterSrcAddress	Source IP address
1.3.6.1.4.1.9.9.278.1.1.3.1.6	cippfIpFilterSrcMask	Source IP mask
1.3.6.1.4.1.9.9.278.1.1.3.1.7	cippfIpFilterDestAddress	Destination IP address
1.3.6.1.4.1.9.9.278.1.1.3.1.8	cippfIpFilterDestMask	Destination IP mask
1.3.6.1.4.1.9.9.278.1.1.3.1.9	cippfFilterProtocol	Protocol (IP/TCP/UDP/ICMP)
1.3.6.1.4.1.9.9.278.1.1.3.1.10	cippfIpFilterSrcPortLow	Source port (low)
1.3.6.1.4.1.9.9.278.1.1.3.1.11	cippfIpFilterSrcPortHigh	Source port (high)
1.3.6.1.4.1.9.9.278.1.1.3.1.12	cippfIpFilterDestPortLow	Destination port (low)
1.3.6.1.4.1.9.9.278.1.1.3.1.13	cippfIpFilterDestPortHigh	Destination port (high)

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.278.1.1.3.1.16	cippfIpFilterLogEnabled	Log is enabled or disabled.
1.3.6.1.4.1.9.9.278.1.1.3.1.17	cippfIpFilterStatus	ACL is active or inactive.
1.3.6.1.4.1.9.9.278.1.1.3.1.22	cippfIpFilterSrcIPGroupName	Source network object group name
1.3.6.1.4.1.9.9.278.1.1.3.1.23	cippfIpFilterDstIPGroupName	Destination network object group name
1.3.6.1.4.1.9.9.278.1.1.3.1.24	cippfIpFilterProtocolGroupName	Protocol object group name
1.3.6.1.4.1.9.9.278.1.1.3.1.25	cippfIpFilterSrcServiceGroupName	Source service object group name
1.3.6.1.4.1.9.9.278.1.1.3.1.26	cippfIpFilterDstServiceGroupName	Destination service object group name
1.3.6.1.4.1.9.9.278.1.1.3.1.27	cippfIpFilterICMPGroupName	ICMP object group
1.3.6.1.4.1.9.9.278.1.1.4.1.1	cippfIpFilterExtDescription	ACL entry
—	cippfIpFilterExtTables	SNMP filter tables
1.3.6.1.4.1.9.9.278.1.1.4.1.2	cippfIpFilterLogLevel	Log level
1.3.6.1.4.1.9.9.278.1.1.4.1.3	cippfIpFilterLogInterval	Log interval
—	cippfIpFilterStatsTable	show access-list *acl-name*
—	**Index** • cippfIpProfileName • cippfIpFilterIndex	—
1.3.6.1.4.1.9.9.278.1.1.1.1.1	cippfIpProfileName	ACL name
1.3.6.1.4.1.9.9.278.1.1.3.1.1	cippfIpFilterIndex	ACE line number within the ACL
1.3.6.1.4.1.9.9.278.1.2.1.1.1	cippfIpFilterHits	ACE hit-count
CISCO-IP-MIB	ipNetToPhysicalTable	show arp
—	**Index** • ipNetToPhysicalIndex • ipNetToPhysicalNetAddressType • ipNetToPhysicalNetAddress	—
	ipNetToPhysicalIndex	Interfce number for the ARP entry
	ipNetToPhysicalNetAddressType	IP address type for the ARP entry
	ipNetToPhysicalNetAddress	IP address for the ARP entry
	ipNetToPhysicalPhysAddress	MAC address for the IP address
CISCO-IPSEC-FLOW-MONITOR-MIB	—	show ipsec stats

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.171.1.1.1.	cipSecMibLevel	See this MIB for an explanation of the objects.
1.3.6.1.4.1.9.9.171.1.2.1.1.	cikeGlobalActiveTunnels	
1.3.6.1.4.1.9.9.171.1.2.1.2.	cikeGlobalPreviousTunnels	
1.3.6.1.4.1.9.9.171.1.2.1.3.	cikeGlobalInOctets	
1.3.6.1.4.1.9.9.171.1.2.1.4.	cikeGlobalInPkts	
1.3.6.1.4.1.9.9.171.1.2.1.5.	cikeGlobalInDropPkts	
1.3.6.1.4.1.9.9.171.1.2.1.6.	cikeGlobalInNotifys	
1.3.6.1.4.1.9.9.171.1.2.1.7.	cikeGlobalInP2Exchgs	
1.3.6.1.4.1.9.9.171.1.2.1.8.	cikeGlobalInP2ExchgInvalids	
1.3.6.1.4.1.9.9.171.1.2.1.9.	cikeGlobalInP2ExchgRejects	
1.3.6.1.4.1.9.9.171.1.2.1.10.	cikeGlobalInP2SaDelRequests	
1.3.6.1.4.1.9.9.171.1.2.1.11.	cikeGlobalOutOctets	
1.3.6.1.4.1.9.9.171.1.2.1.12.	cikeGlobalOutPkts	
1.3.6.1.4.1.9.9.171.1.2.1.13.	cikeGlobalOutDropPkts	
1.3.6.1.4.1.9.9.171.1.2.1.14.	cikeGlobalOutNotifys	
1.3.6.1.4.1.9.9.171.1.2.1.15.	cikeGlobalOutP2Exchgs	
1.3.6.1.4.1.9.9.171.1.2.1.16.	cikeGlobalOutP2ExchgInvalids	
1.3.6.1.4.1.9.9.171.1.2.1.17.	cikeGlobalOutP2ExchgRejects	
1.3.6.1.4.1.9.9.171.1.2.1.18.	cikeGlobalOutP2SaDelRequests	
1.3.6.1.4.1.9.9.171.1.2.1.19.	cikeGlobalInitTunnels	
1.3.6.1.4.1.9.9.171.1.2.1.20.	cikeGlobalInitTunnelFails	
1.3.6.1.4.1.9.9.171.1.2.1.21.	cikeGlobalRespTunnelFails	
1.3.6.1.4.1.9.9.171.1.2.1.22.	cikeGlobalSysCapFails	
1.3.6.1.4.1.9.9.171.1.2.1.23.	cikeGlobalAuthFails	
1.3.6.1.4.1.9.9.171.1.2.1.24.	cikeGlobalDecryptFails	
1.3.6.1.4.1.9.9.171.1.2.1.25.	cikeGlobalHashValidFails	
1.3.6.1.4.1.9.9.171.1.2.1.26.	cikeGlobalNoSaFails	
1.3.6.1.4.1.9.9.171.1.2.2.1.6.	cikePeerLocalAddr	
1.3.6.1.4.1.9.9.171.1.2.2.1.7.	cikePeerRemoteAddr	
1.3.6.1.4.1.9.9.171.1.2.2.1.8.	cikePeerActiveTime	
1.3.6.1.4.1.9.9.171.1.2.2.1.9.	cikePeerActiveTunnelIndex	
1.3.6.1.4.1.9.9.171.1.2.3.1.2.	cikeTunLocalType	
1.3.6.1.4.1.9.9.171.1.2.3.1.3.	cikeTunLocalValue	

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.171.1.2.3.1.4.	cikeTunLocalAddr	See this MIB for an explanation of the objects.
1.3.6.1.4.1.9.9.171.1.2.3.1.5.	cikeTunLocalName	
1.3.6.1.4.1.9.9.171.1.2.3.1.6.	cikeTunRemoteType	
1.3.6.1.4.1.9.9.171.1.2.3.1.7.	cikeTunRemoteValue	
1.3.6.1.4.1.9.9.171.1.2.3.1.8.	cikeTunRemoteAddr	
1.3.6.1.4.1.9.9.171.1.2.3.1.9.	cikeTunRemoteName	
1.3.6.1.4.1.9.9.171.1.2.3.1.10.	cikeTunNegoMode	
1.3.6.1.4.1.9.9.171.1.2.3.1.11.	cikeTunDiffHellmanGrp	
1.3.6.1.4.1.9.9.171.1.2.3.1.12.	cikeTunEncryptAlgo	
1.3.6.1.4.1.9.9.171.1.2.3.1.13.	cikeTunHashAlgo	
1.3.6.1.4.1.9.9.171.1.2.3.1.14.	cikeTunAuthMethod	
1.3.6.1.4.1.9.9.171.1.2.3.1.15.	cikeTunLifeTime	
1.3.6.1.4.1.9.9.171.1.2.3.1.16.	cikeTunActiveTime	
1.3.6.1.4.1.9.9.171.1.2.3.1.17.	cikeTunSaRefreshThreshold	
1.3.6.1.4.1.9.9.171.1.2.3.1.18.	cikeTunTotalRefreshes	
1.3.6.1.4.1.9.9.171.1.2.3.1.19.	cikeTunInOctets	
1.3.6.1.4.1.9.9.171.1.2.3.1.20.	cikeTunInPkts	
1.3.6.1.4.1.9.9.171.1.2.3.1.21.	cikeTunInDropPkts	
1.3.6.1.4.1.9.9.171.1.2.3.1.22.	cikeTunInNotifys	
1.3.6.1.4.1.9.9.171.1.2.3.1.23.	cikeTunInP2Exchgs	
1.3.6.1.4.1.9.9.171.1.2.3.1.24.	cikeTunInP2ExchgInvalids	
1.3.6.1.4.1.9.9.171.1.2.3.1.25.	cikeTunInP2ExchgRejects	
1.3.6.1.4.1.9.9.171.1.2.3.1.26.	cikeTunInP2SaDelRequests	
1.3.6.1.4.1.9.9.171.1.2.3.1.27.	cikeTunOutOctets	
1.3.6.1.4.1.9.9.171.1.2.3.1.28.	cikeTunOutPkts	
1.3.6.1.4.1.9.9.171.1.2.3.1.29.	cikeTunOutDropPkts	
1.3.6.1.4.1.9.9.171.1.2.3.1.30.	cikeTunOutNotifys	
1.3.6.1.4.1.9.9.171.1.2.3.1.31.	cikeTunOutP2Exchgs	
1.3.6.1.4.1.9.9.171.1.2.3.1.32.	cikeTunOutP2ExchgInvalids	
1.3.6.1.4.1.9.9.171.1.2.3.1.33.	cikeTunOutP2ExchgRejects	
1.3.6.1.4.1.9.9.171.1.2.3.1.34.	cikeTunOutP2SaDelRequests	
1.3.6.1.4.1.9.9.171.1.2.3.1.35.	cikeTunStatus	
1.3.6.1.4.1.9.9.171.1.2.4.1.7.	cikePeerCorrIpSecTunIndex	
1.3.6.1.4.1.9.9.171.1.3.1.1.	cipSecGlobalActiveTunnels	
1.3.6.1.4.1.9.9.171.1.3.1.2.	cipSecGlobalPreviousTunnels	
1.3.6.1.4.1.9.9.171.1.3.1.3.	cipSecGlobalInOctets	

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.171.1.3.1.4.	cipSecGlobalHcInOctets	See this MIB for an explanation of the objects.
1.3.6.1.4.1.9.9.171.1.3.1.5.	cipSecGlobalInOctWraps	
1.3.6.1.4.1.9.9.171.1.3.1.6.	cipSecGlobalInDecompOctets	
1.3.6.1.4.1.9.9.171.1.3.1.7.	cipSecGlobalHcInDecompOctets	
1.3.6.1.4.1.9.9.171.1.3.1.8.	cipSecGlobalInDecompOctWraps	
1.3.6.1.4.1.9.9.171.1.3.1.9.	cipSecGlobalInPkts	
1.3.6.1.4.1.9.9.171.1.3.1.10.	cipSecGlobalInDrops	
1.3.6.1.4.1.9.9.171.1.3.1.11.	cipSecGlobalInReplayDrops	
1.3.6.1.4.1.9.9.171.1.3.1.12.	cipSecGlobalInAuths	
1.3.6.1.4.1.9.9.171.1.3.1.13.	cipSecGlobalInAuthFails	
1.3.6.1.4.1.9.9.171.1.3.1.14.	cipSecGlobalInDecrypts	
1.3.6.1.4.1.9.9.171.1.3.1.15.	cipSecGlobalInDecryptFails	
1.3.6.1.4.1.9.9.171.1.3.1.16.	cipSecGlobalOutOctets	
1.3.6.1.4.1.9.9.171.1.3.1.17.	cipSecGlobalHcOutOctets	
1.3.6.1.4.1.9.9.171.1.3.1.18.	cipSecGlobalOutOctWraps	
1.3.6.1.4.1.9.9.171.1.3.1.19.	cipSecGlobalOutUncompOctets	
1.3.6.1.4.1.9.9.171.1.3.1.20.	cipSecGlobalHcOutUncompOctets	
1.3.6.1.4.1.9.9.171.1.3.1.21.	cipSecGlobalOutUncompOctWraps	
1.3.6.1.4.1.9.9.171.1.3.1.22.	cipSecGlobalOutPkts	
1.3.6.1.4.1.9.9.171.1.3.1.23.	cipSecGlobalOutDrops	
1.3.6.1.4.1.9.9.171.1.3.1.24.	cipSecGlobalOutAuths	
1.3.6.1.4.1.9.9.171.1.3.1.25.	cipSecGlobalOutAuthFails	
1.3.6.1.4.1.9.9.171.1.3.1.26.	cipSecGlobalOutEncrypts	
1.3.6.1.4.1.9.9.171.1.3.1.27.	cipSecGlobalOutEncryptFails	
1.3.6.1.4.1.9.9.171.1.3.1.28.	cipSecGlobalProtocolUseFails	
1.3.6.1.4.1.9.9.171.1.3.1.29.	cipSecGlobalNoSaFails	
1.3.6.1.4.1.9.9.171.1.3.1.30.	cipSecGlobalSysCapFails	
1.3.6.1.4.1.9.9.171.1.3.2.1.2.	cipSecTunIkeTunnelIndex	
1.3.6.1.4.1.9.9.171.1.3.2.1.3.	cipSecTunIkeTunnelAlive	
1.3.6.1.4.1.9.9.171.1.3.2.1.4.	cipSecTunLocalAddr	
1.3.6.1.4.1.9.9.171.1.3.2.1.5.	cipSecTunRemoteAddr	
1.3.6.1.4.1.9.9.171.1.3.2.1.6.	cipSecTunKeyType	
1.3.6.1.4.1.9.9.171.1.3.2.1.7.	cipSecTunEncapMode	
1.3.6.1.4.1.9.9.171.1.3.2.1.8.	cipSecTunLifeSize	
1.3.6.1.4.1.9.9.171.1.3.2.1.9.	cipSecTunLifeTime	
1.3.6.1.4.1.9.9.171.1.3.2.1.10.	cipSecTunActiveTime	

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.171.1.3.2.1.11.	cipSecTunSaLifeSizeThreshold	See this MIB for an explanation of the objects.
1.3.6.1.4.1.9.9.171.1.3.2.1.12.	cipSecTunSaLifeTimeThreshold	
1.3.6.1.4.1.9.9.171.1.3.2.1.13.	cipSecTunTotalRefreshes	
1.3.6.1.4.1.9.9.171.1.3.2.1.14.	cipSecTunExpiredSaInstances	
1.3.6.1.4.1.9.9.171.1.3.2.1.15.	cipSecTunCurrentSaInstances	
1.3.6.1.4.1.9.9.171.1.3.2.1.16.	cipSecTunInSaDiffHellmanGrp	
1.3.6.1.4.1.9.9.171.1.3.2.1.17.	cipSecTunInSaEncryptAlgo	
1.3.6.1.4.1.9.9.171.1.3.2.1.18.	cipSecTunInSaAhAuthAlgo	
1.3.6.1.4.1.9.9.171.1.3.2.1.19.	cipSecTunInSaEspAuthAlgo	
1.3.6.1.4.1.9.9.171.1.3.2.1.20.	cipSecTunInSaDecompAlgo	
1.3.6.1.4.1.9.9.171.1.3.2.1.21.	cipSecTunOutSaDiffHellmanGrp	
1.3.6.1.4.1.9.9.171.1.3.2.1.22.	cipSecTunOutSaEncryptAlgo	
1.3.6.1.4.1.9.9.171.1.3.2.1.23.	cipSecTunOutSaAhAuthAlgo	
1.3.6.1.4.1.9.9.171.1.3.2.1.24.	cipSecTunOutSaEspAuthAlgo	
1.3.6.1.4.1.9.9.171.1.3.2.1.25.	cipSecTunOutSaCompAlgo	
1.3.6.1.4.1.9.9.171.1.3.2.1.26.	cipSecTunInOctets	
1.3.6.1.4.1.9.9.171.1.3.2.1.27.	cipSecTunHcInOctets	
1.3.6.1.4.1.9.9.171.1.3.2.1.28.	cipSecTunInOctWraps	
1.3.6.1.4.1.9.9.171.1.3.2.1.29.	cipSecTunInDecompOctets	
1.3.6.1.4.1.9.9.171.1.3.2.1.30.	cipSecTunHcInDecompOctets	
1.3.6.1.4.1.9.9.171.1.3.2.1.31.	cipSecTunInDecompOctWraps	
1.3.6.1.4.1.9.9.171.1.3.2.1.32.	cipSecTunInPkts	
1.3.6.1.4.1.9.9.171.1.3.2.1.33.	cipSecTunInDropPkts	
1.3.6.1.4.1.9.9.171.1.3.2.1.34.	cipSecTunInReplayDropPkts	
1.3.6.1.4.1.9.9.171.1.3.2.1.35.	cipSecTunInAuths	
1.3.6.1.4.1.9.9.171.1.3.2.1.36.	cipSecTunInAuthFails	
1.3.6.1.4.1.9.9.171.1.3.2.1.37.	cipSecTunInDecrypts	
1.3.6.1.4.1.9.9.171.1.3.2.1.38.	cipSecTunInDecryptFails	
1.3.6.1.4.1.9.9.171.1.3.2.1.39.	cipSecTunOutOctets	
1.3.6.1.4.1.9.9.171.1.3.2.1.40.	cipSecTunHcOutOctets	
1.3.6.1.4.1.9.9.171.1.3.2.1.41.	cipSecTunOutOctWraps	
1.3.6.1.4.1.9.9.171.1.3.2.1.42.	cipSecTunOutUncompOctets	
1.3.6.1.4.1.9.9.171.1.3.2.1.43.	cipSecTunHcOutUncompOctets	
1.3.6.1.4.1.9.9.171.1.3.2.1.44.	cipSecTunOutUncompOctWraps	
1.3.6.1.4.1.9.9.171.1.3.2.1.45.	cipSecTunOutPkts	
1.3.6.1.4.1.9.9.171.1.3.2.1.46.	cipSecTunOutDropPkts	

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.171.1.3.2.1.47.	cipSecTunOutAuths	See this MIB for an explanation of the objects.
1.3.6.1.4.1.9.9.171.1.3.2.1.48.	cipSecTunOutAuthFails	
1.3.6.1.4.1.9.9.171.1.3.2.1.49.	cipSecTunOutEncrypts	
1.3.6.1.4.1.9.9.171.1.3.2.1.50.	cipSecTunOutEncryptFails	
1.3.6.1.4.1.9.9.171.1.3.2.1.51.	cipSecTunStatus	
1.3.6.1.4.1.9.9.171.1.3.3.1.2.	cipSecEndPtLocalName	
1.3.6.1.4.1.9.9.171.1.3.3.1.3.	cipSecEndPtLocalType	
1.3.6.1.4.1.9.9.171.1.3.3.1.4.	cipSecEndPtLocalAddr1	
1.3.6.1.4.1.9.9.171.1.3.3.1.5.	cipSecEndPtLocalAddr2	
1.3.6.1.4.1.9.9.171.1.3.3.1.6.	cipSecEndPtLocalProtocol	
1.3.6.1.4.1.9.9.171.1.3.3.1.7.	cipSecEndPtLocalPort	
1.3.6.1.4.1.9.9.171.1.3.3.1.8.	cipSecEndPtRemoteName	
1.3.6.1.4.1.9.9.171.1.3.3.1.9.	cipSecEndPtRemoteType	
1.3.6.1.4.1.9.9.171.1.3.3.1.10.	cipSecEndPtRemoteAddr1	
1.3.6.1.4.1.9.9.171.1.3.3.1.11.	cipSecEndPtRemoteAddr2	
1.3.6.1.4.1.9.9.171.1.3.3.1.12.	cipSecEndPtRemoteProtocol	
1.3.6.1.4.1.9.9.171.1.3.3.1.13.	cipSecEndPtRemotePort	
1.3.6.1.4.1.9.9.171.1.3.4.1.2.	cipSecSpiDirection	
1.3.6.1.4.1.9.9.171.1.3.4.1.3.	cipSecSpiValue	
1.3.6.1.4.1.9.9.171.1.3.4.1.4.	cipSecSpiProtocol	
1.3.6.1.4.1.9.9.171.1.3.4.1.5.	cipSecSpiStatus	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.2.	cipSecTunHistTermReason	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.3.	cipSecTunHistActiveIndex	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.4.	cipSecTunHistIkeTunnelIndex	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.5.	cipSecTunHistLocalAddr	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.6.	cipSecTunHistRemoteAddr	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.7.	cipSecTunHistKeyType	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.8.	cipSecTunHistEncapMode	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.9.	cipSecTunHistLifeSize	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.10.	cipSecTunHistLifeTime	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.11.	cipSecTunHistStartTime	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.12.	cipSecTunHistActiveTime	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.13.	cipSecTunHistTotalRefreshes	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.14.	cipSecTunHistTotalSas	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.15.	cipSecTunHistInSaDiffHellmanGrp	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.16.	cipSecTunHistInSaEncryptAlgo	

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.171.1.4.3.1.1.17.	cipSecTunHistInSaAhAuthAlgo	See this MIB for an explanation of the objects.
1.3.6.1.4.1.9.9.171.1.4.3.1.1.18.	cipSecTunHistInSaEspAuthAlgo	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.19.	cipSecTunHistInSaDecompAlgo	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.20.	cipSecTunHistOutSaDiffHellmanGrp	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.21.	cipSecTunHistOutSaEncryptAlgo	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.22.	cipSecTunHistOutSaAhAuthAlgo	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.23.	cipSecTunHistOutSaEspAuthAlgo	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.24.	cipSecTunHistOutSaCompAlgo	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.25.	cipSecTunHistInOctets	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.26.	cipSecTunHistHcInOctets	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.27.	cipSecTunHistInOctWraps	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.28.	cipSecTunHistInDecompOctets	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.29.	cipSecTunHistHcInDecompOctets	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.30.	cipSecTunHistInDecompOctWraps	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.31.	cipSecTunHistInPkts	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.32.	cipSecTunHistInDropPkts	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.33.	cipSecTunHistInReplayDropPkts	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.34.	cipSecTunHistInAuths	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.35.	cipSecTunHistInAuthFails	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.36.	cipSecTunHistInDecrypts	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.37.	cipSecTunHistInDecryptFails	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.38.	cipSecTunHistOutOctets	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.39.	cipSecTunHistHcOutOctets	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.40.	cipSecTunHistOutOctWraps	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.41.	cipSecTunHistOutUncompOctets	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.42.	cipSecTunHistHcOutUncompOctets	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.43.	cipSecTunHistOutUncompOctWraps	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.44.	cipSecTunHistOutPkts	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.45.	cipSecTunHistOutDropPkts	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.46.	cipSecTunHistOutAuths	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.47.	cipSecTunHistOutAuthFails	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.48.	cipSecTunHistOutEncrypts	
1.3.6.1.4.1.9.9.171.1.4.3.1.1.49.	cipSecTunHistOutEncryptFails	
1.3.6.1.4.1.9.9.171.1.5.1.1.1.	cipSecFailTableSize	
1.3.6.1.4.1.9.9.171.1.5.2.1.1.2.	cikeFailReason	
1.3.6.1.4.1.9.9.171.1.5.2.1.1.3.	cikeFailTime	

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.171.1.5.2.1.1.4.	cikeFailLocalType	See this MIB for an explanation of the objects.
1.3.6.1.4.1.9.9.171.1.5.2.1.1.5.	cikeFailLocalValue	
1.3.6.1.4.1.9.9.171.1.5.2.1.1.6.	cikeFailRemoteType	
1.3.6.1.4.1.9.9.171.1.5.2.1.1.7.	cikeFailRemoteValue	
1.3.6.1.4.1.9.9.171.1.5.2.1.1.8.	cikeFailLocalAddr	
1.3.6.1.4.1.9.9.171.1.5.2.1.1.9.	cikeFailRemoteAddr	
CISCO-L4L7MODULE-RESOURCE-LIMIT-MIB	ciscoL4L7ResourceLimitTable	show resource usage context *ctx-name* Obtains the resource usage statistics of the absolute resource types.
1.3.6.1.4.1.9.9.480.1.1.1.1.1	**Index** • crlResourceClassName • crlResourceLimitType	—
1.3.6.1.4.1.9.9.480.0.1	crlResourceLimitReached	The configured resource limit has been reached.
1.3.6.1.4.1.9.9.480.0.2	crlResourceRateLimitReached	The configured resource rate limit has been reached.
1.3.6.1.4.1.9.9.480.1.1.2.1.2	crlResourceLimitType	Resource type (Conns/Xlates/Hosts/SSH/Telnet /ASDM/IPSec/MAC Address)
1.3.6.1.4.1.9.9.480.1.1.2.1.3	crlResourceLimitValueType	Absolute or percentage
1.3.6.1.4.1.9.9.480.1.1.2.1.4	crlResourceLimitMin	Always set to zero. Not applicable to FWSM.
1.3.6.1.4.1.9.9.480.1.1.2.1.5	crlResourceLimitMax	Configured limit value
1.3.6.1.4.1.9.9.480.1.1.2.1.8	crlResourceLimitCurrentUsage	Current resource usage
1.3.6.1.4.1.9.9.480.1.1.2.1.9	crlResourceLimitPeakUsage	Peak resource usage
1.3.6.1.4.1.9.9.480.1.1.2.1.10	crlResourceLimitDeniedCount	Denied count
—	ciscoL4L7ResourceRateLimitTable	show resource usage context *ctx-name* Obtains the resource usage statistics of the rate resource types.
1.3.6.1.4.1.9.9.480.1.1.1.1.1	**Index** • crlResourceClassName • crlRateLimitResourceType	—
1.3.6.1.4.1.9.9.480.1.1.4.1.1	crlRateLimitResourceType	Resource type (Conns/Fixups/Syslogs)
1.3.6.1.4.1.9.9.480.1.1.4.1.2	crlRateLimitMin	Always set to zero. Not applicable to FWSM.
1.3.6.1.4.1.9.9.480.1.1.4.1.3	crlRateLimitMax	Configured rate limit value
1.3.6.1.4.1.9.9.480.1.1.4.1.6	crlRateLimitCurrentUsage	Current resource usage

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.480.1.1.4.1.7	crlRateLimitPeakUsage	Peak resource usage
1.3.6.1.4.1.9.9.480.1.1.4.1.8	crlRateLimitDeniedCount	Denied count
CISCO-MEMORY-POOL-MIB	—	show memory
1.3.6.1.4.1.9.9.48.1.1	ciscoMemoryPoolTable	—
—	**Index** • ciscoMemoryPoolType	—
1.3.6.1.4.1.9.9.48.1.1.1.2.	ciscoMemoryPoolName	Memory pool name
1.3.6.1.4.1.9.9.48.1.1.1.3.	ciscoMemoryPoolAlternate	Memory pool alternate
1.3.6.1.4.1.9.9.48.1.1.1.4.	ciscoMemoryPoolValid	Is this pool a valid one?
1.3.6.1.4.1.9.9.48.1.1.1.5.	ciscoMemoryPoolUsed	Memory used in bytes
1.3.6.1.4.1.9.9.48.1.1.1.6.	ciscoMemoryPoolFree	Free memory in bytes
1.3.6.1.4.1.9.9.48.1.1.1.7.	ciscoMemoryPoolLargestFree	Largest free memory in bytes
CISCO-NAT-EXT-MIB	cneAddTranslationStatsTable	show xlate count
—	**Index** • entPhysicalIndex=1	—
1.3.6.1.4.1.9.9.xxx.1.1.1	cneAddrTranslationNumActive	Currently active xlate in use
1.3.6.1.4.1.9.9.xxx.1.1.2	cneAddrTranslationNumPeak	Max in use xlate at any time
—	—	show perfmon detail
1.3.6.1.4.1.9.9.xxx.1.1.3	cneAddrTranslation1min	xlate for one minute
1.3.6.1.4.1.9.9.xxx.1.1.4	cneAddrTranslation5min	xlate for five minutes
CISCO-PROCESS-MIB	—	show cpu
—	cpmCPUTotalTable **Index** • cpmCPUTotalIndex	Always set to one.
1.3.6.1.4.1.9.9.109.1.1.1.1.2.	cpmCPUTotalPhysicalIndex	entPhysicalIndex (always set to zero).
1.3.6.1.4.1.9.9.109.1.1.1.1.3.	cpmCPUTotal5sec	CPU utilization for five seconds
1.3.6.1.4.1.9.9.109.1.1.1.1.4.	cpmCPUTotal1min	CPU utilization for one minute
1.3.6.1.4.1.9.9.109.1.1.1.1.5.	cpmCPUTotal5min	CPU utilization for five minutes
1.3.6.1.4.1.9.9.109.1.1.1.1.6.	cpmCPUTotal5secRev	CPU utilization for five seconds
1.3.6.1.4.1.9.9.109.1.1.1.1.7.	cpmCPUTotal1minRev	CPU utilization for one minute
1.3.6.1.4.1.9.9.109.1.1.1.1.8.	cpmCPUTotal5minRev	CPU utilization for five minutes
1.3.6.1.4.1.9.9.109.1.1.1.1.9.	cpmCPUMonInterval	Monitoring interval (always set to five seconds).
1.3.6.1.4.1.9.9.109.1.1.1.1.10.	cpmCPUTotalMonIntervalValue	CPU utilization for five seconds
1.3.6.1.4.1.9.9.109.1.1.1.1.11.	cpmCPUInterruptMonIntervalValue	Not applicable (placeholder only)
1.3.6.1.4.1.9.9.109.1.2.1.1.1.	cpmProcessPID	Not applicable (placeholder only)

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.109.1.2.1.1.5.	cpmProcessTimeCreated	Not applicable (placeholder only)
1.3.6.1.4.1.9.9.109.1.2.3.1.5.	cpmProcExtUtil5SecRev	Not applicable (placeholder only)
1.3.6.1.4.1.9.9.109.1.2.4.1.2.	cpmCPURisingThresholdValue	CPU rising threshold value
1.3.6.1.4.1.9.9.109.1.2.4.1.3.	cpmCPURisingThresholdPeriod	CPU rising threshold monitoring period
CISCO-REMOTE-ACCESS-MONITOR-MIB	—	See this MIB for an explanation of the objects.
1.3.6.1.4.1.9.9.392.1.1.1.	crasMaxSessionsSupportable	
1.3.6.1.4.1.9.9.392.1.1.2.	crasMaxUsersSupportable	
1.3.6.1.4.1.9.9.392.1.1.3.	crasMaxGroupsSupportable	
1.3.6.1.4.1.9.9.392.1.1.4.	crasNumCryptoAccelerators	
1.3.6.1.4.1.9.9.392.1.3.1.	crasNumSessions	
1.3.6.1.4.1.9.9.392.1.3.2.	crasNumPrevSessions	
1.3.6.1.4.1.9.9.392.1.3.3.	crasNumUsers	
1.3.6.1.4.1.9.9.392.1.3.4.	crasNumGroups	
1.3.6.1.4.1.9.9.392.1.3.5.	crasGlobalInPkts	
1.3.6.1.4.1.9.9.392.1.3.6.	crasGlobalOutPkts	
1.3.6.1.4.1.9.9.392.1.3.7.	crasGlobalInOctets	
1.3.6.1.4.1.9.9.392.1.3.8.	crasGlobalInDecompOctets	
1.3.6.1.4.1.9.9.392.1.3.9.	crasGlobalOutOctets	
1.3.6.1.4.1.9.9.392.1.3.10.	crasGlobalOutUncompOctets	
1.3.6.1.4.1.9.9.392.1.3.11.	crasGlobalInDropPkts	
1.3.6.1.4.1.9.9.392.1.3.12.	crasGlobalOutDropPkts	
1.3.6.1.4.1.9.9.392.1.3.21.1.2.	crasGroup	
1.3.6.1.4.1.9.9.392.1.3.21.1.4.	crasAuthenMethod	
1.3.6.1.4.1.9.9.392.1.3.21.1.5.	crasAuthorMethod	
1.3.6.1.4.1.9.9.392.1.3.21.1.6.	crasSessionDuration	
1.3.6.1.4.1.9.9.392.1.3.21.1.7.	crasLocalAddressType	
1.3.6.1.4.1.9.9.392.1.3.21.1.8.	crasLocalAddress	
1.3.6.1.4.1.9.9.392.1.3.21.1.9.	crasISPAddressType	
1.3.6.1.4.1.9.9.392.1.3.21.1.10.	crasISPAddress	
1.3.6.1.4.1.9.9.392.1.3.21.1.11.	crasSessionProtocol	
1.3.6.1.4.1.9.9.392.1.3.21.1.12.	crasProtocolElement	
1.3.6.1.4.1.9.9.392.1.3.21.1.13.	crasSessionEncryptionAlgo	
1.3.6.1.4.1.9.9.392.1.3.21.1.14.	crasSessionPktAuthenAlgo	
1.3.6.1.4.1.9.9.392.1.3.21.1.15.	crasSessionCompressionAlgo	
1.3.6.1.4.1.9.9.392.1.3.21.1.16.	crasHeartbeatInterval	

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.392.1.3.21.1.17.	crasClientVendorString	See this MIB for an explanation of the objects.
1.3.6.1.4.1.9.9.392.1.3.21.1.18.	crasClientVersionString	
1.3.6.1.4.1.9.9.392.1.3.21.1.19.	crasClientOSVendorString	
1.3.6.1.4.1.9.9.392.1.3.21.1.20.	crasClientOSVersionString	
1.3.6.1.4.1.9.9.392.1.3.21.1.21.	crasPrimWINSServerAddrType	
1.3.6.1.4.1.9.9.392.1.3.21.1.22.	crasPrimWINSServer	
1.3.6.1.4.1.9.9.392.1.3.21.1.23.	crasSecWINSServerAddrType	
1.3.6.1.4.1.9.9.392.1.3.21.1.24.	crasSecWINSServer	
1.3.6.1.4.1.9.9.392.1.3.21.1.25.	crasPrimDNSServerAddrType	
1.3.6.1.4.1.9.9.392.1.3.21.1.26.	crasPrimDNSServer	
1.3.6.1.4.1.9.9.392.1.3.21.1.27.	crasSecDNSServerAddrType	
1.3.6.1.4.1.9.9.392.1.3.21.1.28.	crasSecDNSServer	
1.3.6.1.4.1.9.9.392.1.3.21.1.29.	crasDHCPServerAddrType	
1.3.6.1.4.1.9.9.392.1.3.21.1.30.	crasDHCPServer	
1.3.6.1.4.1.9.9.392.1.3.21.1.31.	crasSessionInPkts	
1.3.6.1.4.1.9.9.392.1.3.21.1.32.	crasSessionOutPkts	
1.3.6.1.4.1.9.9.392.1.3.21.1.33.	crasSessionInDropPkts	
1.3.6.1.4.1.9.9.392.1.3.21.1.34.	crasSessionOutDropPkts	
1.3.6.1.4.1.9.9.392.1.3.21.1.35.	crasSessionInOctets	
1.3.6.1.4.1.9.9.392.1.3.21.1.36.	crasSessionOutOctets	
1.3.6.1.4.1.9.9.392.1.3.21.1.37.	crasSessionState	
1.3.6.1.4.1.9.9.392.1.3.22.1.2.	crasActGrNumUsers	
1.3.6.1.4.1.9.9.392.1.3.22.1.3.	crasActGrpInPkts	
1.3.6.1.4.1.9.9.392.1.3.22.1.4.	crasActGrpOutPkts	
1.3.6.1.4.1.9.9.392.1.3.22.1.5.	crasActGrpInDropPkts	
1.3.6.1.4.1.9.9.392.1.3.22.1.6.	crasActGrpOutDropPkts	
1.3.6.1.4.1.9.9.392.1.3.22.1.7.	crasActGrpInOctets	
1.3.6.1.4.1.9.9.392.1.3.22.1.8.	crasActGrpOutOctets	
1.3.6.1.4.1.9.9.392.1.6.1.	crasThrMaxSessions	
1.3.6.1.4.1.9.9.392.1.6.2.	crasThrMaxFailedAuths	
1.3.6.1.4.1.9.9.392.1.6.3.	crasThrMaxThroughput	
CISCO-SYSLOG-MIB	—	—
1.3.6.1.4.1.9.9.41.1.1.6.	clogOriginIDType	Origin identification type
1.3.6.1.4.1.9.9.41.1.1.7.	clogOriginID	Origin identification string
CISCO-UNIFIED-FIREWALL-MIB	—	show perfmon detail

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.491.1.1.1.6.	cufwConnGlobalNumActive	Total no. of active connections (TCP + UDP)
1.3.6.1.4.1.9.9.491.1.1.1.10.	cufwConnGlobalConnSetupRate1	Rate of connections for one minute
1.3.6.1.4.1.9.9.491.1.1.1.11.	cufwConnGlobalConnSetupRate5	Rate of connections for five minutes
1.3.6.1.4.1.9.9.491.1.1.4.1.1.9.6	cufwConnSetupRate1	Rate of UDP connections for one minute
1.3.6.1.4.1.9.9.491.1.1.4.1.1.9.7		Rate of TCP connections for one minute
1.3.6.1.4.1.9.9.491.1.1.4.1.1.10.6	cufwConnSetupRate5	Rate of UDP connections for five minutes
1.3.6.1.4.1.9.9.491.1.1.4.1.1.10.7		Rate of TCP connections for five minutes
1.3.6.1.4.1.9.9.491.1.3.1.2.	cufwUrlfRequestsNumProcessed	show url-server statistics
—	—	URL filtering global statistics
1.3.6.1.4.1.9.9.491.1.3.1.3.	cufwUrlfRequestsProcRate1	
1.3.6.1.4.1.9.9.491.1.3.1.4.	cufwUrlfRequestsProcRate5	
1.3.6.1.4.1.9.9.491.1.3.1.5.	cufwUrlfRequestsNumAllowed	
1.3.6.1.4.1.9.9.491.1.3.1.6.	cufwUrlfRequestsNumDenied	
1.3.6.1.4.1.9.9.491.1.3.1.7.	cufwUrlfRequestsDeniedRate1	
1.3.6.1.4.1.9.9.491.1.3.1.8.	cufwUrlfRequestsDeniedRate5	
1.3.6.1.4.1.9.9.491.1.3.1.9.	cufwUrlfRequestsNumCacheAllowed	
1.3.6.1.4.1.9.9.491.1.3.1.10.	cufwUrlfRequestsNumCacheDenied	
1.3.6.1.4.1.9.9.491.1.3.1.13.	cufwUrlfRequestsNumResDropped	
1.3.6.1.4.1.9.9.491.1.3.1.14.	cufwUrlfRequestsResDropRate1	
1.3.6.1.4.1.9.9.491.1.3.1.15.	cufwUrlfRequestsResDropRate5	
1.3.6.1.4.1.9.9.491.1.3.1.16.	cufwUrlfNumServerTimeouts	
1.3.6.1.4.1.9.9.491.1.3.1.17.	cufwUrlfNumServerRetries	
—	**Unsupported Objects** • cufwUrlfFunctionEnabled • cufwUrlfAllowModeReqNumAllowed • cufwUrlfAllowModeReqNumDenied • cufwUrlfResponsesNumLate • cufwUrlfUrlAccRespsNumResDropped	—
—	cufwUrlServerTable	Per URL server statistics

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.4.1.9.9.491.1.3.3.1.1.1. 1.3.6.1.4.1.9.9.491.1.3.3.1.1.2. 1.3.6.1.4.1.9.9.491.1.3.3.1.1.3	**Index** • cufwUrlfServerAddrType • cufwUrlfServerAddress • cufwUrlfServerPort	—
1.3.6.1.4.1.9.9.491.1.3.3.1.1.4.	cufwUrlfServerVendor	—
1.3.6.1.4.1.9.9.491.1.3.3.1.1.5.	cufwUrlfServerStatus	—
1.3.6.1.4.1.9.9.491.1.3.3.1.1.6.	cufwUrlfServerReqsNumProcessed	—
1.3.6.1.4.1.9.9.491.1.3.3.1.1.7.	cufwUrlfServerReqsNumAllowed	—
1.3.6.1.4.1.9.9.491.1.3.3.1.1.8.	cufwUrlfServerReqsNumDenied	—
1.3.6.1.4.1.9.9.491.1.3.3.1.1.9.	cufwUrlfServerNumTimeouts	—
1.3.6.1.4.1.9.9.491.1.3.3.1.1.10.	cufwUrlfServerNumRetries	—
1.3.6.1.4.1.9.9.491.1.3.3.1.1.11.	cufwUrlfServerRespsNumReceived	—
1.3.6.1.4.1.9.9.491.1.3.3.1.1.13.	cufwUrlfServerAvgRespTime1	—
1.3.6.1.4.1.9.9.491.1.3.3.1.1.14.	cufwUrlfServerAvgRespTime5	—
ENTITY-MIB (1.3.6.1.2.1.47)	—	—
1.3.6.1.2.1.47.1.1.1	entPhysicalTable	Information about a physical entity
—	**Index** • entPhysicalIndex	—
1.3.6.1.2.1.47.1.1.1.1.2.	entPhysicalDescr	—
1.3.6.1.2.1.47.1.1.1.1.3.	entPhysicalVendorType	—
1.3.6.1.2.1.47.1.1.1.1.4.	entPhysicalContainedIn	—
1.3.6.1.2.1.47.1.1.1.1.5.	entPhysicalClass	—
1.3.6.1.2.1.47.1.1.1.1.6.	entPhysicalParentRelPos	—
1.3.6.1.2.1.47.1.1.1.1.7.	entPhysicalName	—
1.3.6.1.2.1.47.1.1.1.1.8.	entPhysicalHardwareRev	—
1.3.6.1.2.1.47.1.1.1.1.9.	entPhysicalFirmwareRev	—
1.3.6.1.2.1.47.1.1.1.1.10.	entPhysicalSoftwareRev	—
1.3.6.1.2.1.47.1.1.1.1.11.	entPhysicalSerialNum	—
1.3.6.1.2.1.47.1.1.1.1.12.	entPhysicalMfgName	—
1.3.6.1.2.1.47.1.1.1.1.13.	entPhysicalModelName	—
1.3.6.1.2.1.47.1.1.1.1.14.	entPhysicalAlias	—
1.3.6.1.2.1.47.1.1.1.1.15.	entPhysicalAssetID	—
1.3.6.1.2.1.47.1.1.1.1.16.	entPhysicalIsFRU	—
1.3.6.1.2.1.47.1.2.1	entLogicalTable	Information about a logical entity
—	**Index** • entLogicalIndex	—

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.2.1.47.1.2.1.1.2.	entLogicalDescr	—
1.3.6.1.2.1.47.1.2.1.1.3.	entLogicalType	—
1.3.6.1.2.1.47.1.2.1.1.4.	entLogicalCommunity	—
1.3.6.1.2.1.47.1.2.1.1.5.	entLogicalTAddress	—
1.3.6.1.2.1.47.1.2.1.1.6.	entLogicalTDomain	—
1.3.6.1.2.1.47.1.2.1.1.7.	entLogicalContextEngineID	—
1.3.6.1.2.1.47.1.2.1.1.8.	entLogicalContextName	—
1.3.6.1.2.1.47.1.3.1.1.1.	entLPPhysicalIndex	—
1.3.6.1.2.1.47.1.3.2.1.2.	entAliasMappingIdentifier	—
1.3.6.1.2.1.47.1.3.3.1.1.	entPhysicalChildIndex	—
1.3.6.1.2.1.47.1.4.1.	entLastChangeTime	—
INTERFACES-MIB (1.3.6.1.2.1.2)	—	show interface
1.3.6.1.2.1.2.1.	ifNumber	Number of interfaces in the system
1.3.6.1.2.1.2.2	ifTable	—
—	**Index** • ifIndex	—
1.3.6.1.2.1.2.2.1.1.	ifIndex	Interface index
1.3.6.1.2.1.2.2.1.2.	ifDescr	Interface description
1.3.6.1.2.1.2.2.1.3.	ifType	Interface type
1.3.6.1.2.1.2.2.1.4.	ifMtu	MTU of the interface
1.3.6.1.2.1.2.2.1.5.	ifSpeed	Speed of the interface
1.3.6.1.2.1.2.2.1.6.	ifPhysAddress	MAC address of the interface
1.3.6.1.2.1.2.2.1.7.	ifAdminStatus	Admin status
1.3.6.1.2.1.2.2.1.8.	ifOperStatus	Operational status
1.3.6.1.2.1.2.2.1.9.	ifLastChange	Last changed time
1.3.6.1.2.1.2.2.1.10.	ifInOctets	Total octets received
1.3.6.1.2.1.2.2.1.11.	ifInUcastPkts	Total unicast packets received
1.3.6.1.2.1.2.2.1.12.	ifInNUcastPkts	Total nonunicast packets received
1.3.6.1.2.1.2.2.1.13.	ifInDiscards	Total inbound packets discarded
1.3.6.1.2.1.2.2.1.14.	ifInErrors	No. of erroneous packets received
1.3.6.1.2.1.2.2.1.16.	ifOutOctets	Total octets sent out
1.3.6.1.2.1.2.2.1.17.	ifOutUcastPkts	Total unicast packets sent out
1.3.6.1.2.1.2.2.1.18.	ifOutNUcastPkts	Total nonunicast packets sent out
1.3.6.1.2.1.2.2.1.19.	ifOutDiscards	Total outbound packets discarded
1.3.6.1.2.1.2.2.1.20.	ifOutErrors	No. of erroneous packets
1.3.6.1.2.1.2.2.1.21.	ifOutQLen	Output packet queue length

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.2.1.2.2.1.22.	ifSpecific	Specific value
1.3.6.1.2.1.31.1	ifXTable	—
—	**Index** • ifIndex	—
1.3.6.1.2.1.31.1.1.1.1.	ifName	Name of the interface
1.3.6.1.2.1.31.1.1.1.2.	ifInMulticastPkts	Total inbound multicast packets
1.3.6.1.2.1.31.1.1.1.3.	ifInBroadcastPkts	Total inbound broadcast packets
1.3.6.1.2.1.31.1.1.1.4.	ifOutMulticastPkts	Total outbound multicast packets
1.3.6.1.2.1.31.1.1.1.5.	ifOutBroadcastPkts	Total outbound broadcast packets
1.3.6.1.2.1.31.1.1.1.6.	ifHCInOctets	Total octets received
1.3.6.1.2.1.31.1.1.1.7.	ifHCInUcastPkts	Total unicast packets received
1.3.6.1.2.1.31.1.1.1.8.	ifHCInMulticastPkts	Total multicast packets received
1.3.6.1.2.1.31.1.1.1.9.	ifHCInBroadcastPkts	Total broadcast packets received
1.3.6.1.2.1.31.1.1.1.10.	ifHCOutOctets	Total octets sent out
1.3.6.1.2.1.31.1.1.1.11.	ifHCOutUcastPkts	Total unicast packets sent out
1.3.6.1.2.1.31.1.1.1.12.	ifHCOutMulticastPkts	Total multicast packets sent out
1.3.6.1.2.1.31.1.1.1.13.	ifHCOutBroadcastPkts	Total broadcast packets sent out
1.3.6.1.2.1.31.1.1.1.14.	ifLinkUpDownTrapEnable	Link up/down trap enabled
1.3.6.1.2.1.31.1.1.1.15.	ifHighSpeed	Interface speed
1.3.6.1.2.1.31.1.1.1.1	ifPromiscuousMode	Is the interface in promiscuous mode?
1.3.6.1.2.1.31.1.1.1.17.	ifConnectorPresent	Does the interface have a physical connector?
1.3.6.1.2.1.31.1.1.1.18.	ifAlias	Alias name of the interface
1.3.6.1.2.1.31.1.1.1.19.	ifCounterDiscontinuityTime	Discontinuity time for interface counters
IP-MIB(1.3.6.1.2.1.4)	—	—
1.3.6.1.2.1.4.1.	ipForwarding	Is IP forwarding enabled?
1.3.6.1.2.1.4.20	ipAddrTable	—
—	**Index** • ipAdEntAddr	—
1.3.6.1.2.1.4.20.1.1.	ipAdEntAddr	IP address
1.3.6.1.2.1.4.20.1.2.	ipAdEntIfIndex	Interface index
1.3.6.1.2.1.4.20.1.3.	ipAdEntNetMask	Subnet mask
1.3.6.1.2.1.4.20.1.4.	ipAdEntBcastAddr	Broadcast address
1.3.6.1.2.1.4.20.1.5.	ipAdEntReasmMaxSize	Max reassembly packet size
NAT-MIB	NatAddressBindTable	show xlate state static detail

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
—	**Index** • ifIndex • natAddrBindLocalAddrType • natAddrBindLocalAddr	—
1.3.6.1.2.1.123.1.6.1.1.	natAddrBindLocalAddrType	ipv4 or ipv6
1.3.6.1.2.1.123.1.6.1.2.	natAddrBindLocalAddr	*local_addr*
1.3.6.1.2.1.123.1.6.1.3.	natAddrBindGlobalAddrType	ipv4 or ipv6
1.3.6.1.2.1.123.1.6.1.4.	natAddrBindGlobalAddr	*global_addr*
1.3.6.1.2.1.123.1.6.1.9.	natAddrBindSessions	No. of conns using this xlate
1.3.6.1.2.1.123.0.1	natPackcketDiscard	Usage of the NAT/PAT xlate has reached its system limit.
—	**Unsupported Objects** • natAddrBindId • natAddrBindTranslationEntity • natAddrBindMapIndex • natAddrBindType • natAddrBindInTranslates • natAddrBindOutTranslates	—
—	NatAddressPortBindTable	show xlate state portmap detail
—	**Index** • ifIndex • natAddrPortBindLocalAddrType • natAddrPortBindLocalAddr • natAddrPortBindLocalPort • natAddrPortBindProtocol	—
1.3.6.1.2.1.123.1.8.1.1.	natAddrPortBindLocalAddrType	ipv4 or ipv6
1.3.6.1.2.1.123.1.8.1.1.2.	natAddrPortBindLocalAddr	*local_addr*
1.3.6.1.2.1.123.1.8.1.1.3.	natAddrPortBindLocalPort	*local_port*
1.3.6.1.2.1.123.1.8.1.1.4.	natAddrPortBindProtocol	TCP/UDP/IP
1.3.6.1.2.1.123.1.8.1.1.5.	natAddrPortBindGlobalAddrType	ipv4 or ipv6
1.3.6.1.2.1.123.1.8.1.1.6.	natAddrPortBindGlobalAddr	*global_addr*
1.3.6.1.2.1.123.1.8.1.1.7.	natAddrPortBindGlobalPort	*global_port*
1.3.6.1.2.1.123.1.8.1.1.12.	natAddrPortBindSessions	No. of conns using this xlate

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
—	**Unsupported Objects** • natAddrPortBindId • natAddrPortBindTranslationEntity • natAddrPortBindMapIndex • natAddrPortBindType • natAddrPortBindInTranslates • natAddrPortBindOutTranslates	—
SNMP-MIB (1.3.6.1.2.1.11)	—	show snmp-server statistics
1.3.6.1.2.1.11.1.	snmpInPkts	Total incoming packets received
1.3.6.1.2.1.11.2.	snmpOutPkts	Total SNMP packets sent out
1.3.6.1.2.1.11.3.	snmpInBadVersions	Bad SNMP version errors
1.3.6.1.2.1.11.4.	snmpInBadCommunityNames	Unknown community name
1.3.6.1.2.1.11.5.	snmpInBadCommunityUses	Illegal operation for the community name
1.3.6.1.2.1.11.6.	snmpInASNParseErrs	Encoding errors
1.3.6.1.2.1.11.8.	snmpInTooBigs	Too big errors
1.3.6.1.2.1.11.9.	snmpInNoSuchNames	No such name errors
1.3.6.1.2.1.11.10.	snmpInBadValues	Bad values errors
1.3.6.1.2.1.11.11.	snmpInReadOnlys	Read-only packets
1.3.6.1.2.1.11.12.	snmpInGenErrs	General errors
1.3.6.1.2.1.11.13.	snmpInTotalReqVars	Total variables queried
1.3.6.1.2.1.11.14.	snmpInTotalSetVars	Total variables modified
1.3.6.1.2.1.11.15.	snmpInGetRequests	Total Get requests received
1.3.6.1.2.1.11.16.	snmpInGetNexts	Total GetNext requests received
1.3.6.1.2.1.11.17.	snmpInSetRequests	Total Set requests received
1.3.6.1.2.1.11.18.	snmpInGetResponses	Total Get responses received
1.3.6.1.2.1.11.19.	snmpInTraps	Total traps received
1.3.6.1.2.1.11.20.	snmpOutTooBigs	Too big errors
1.3.6.1.2.1.11.21.	snmpOutNoSuchNames	No such name errors
1.3.6.1.2.1.11.22.	snmpOutBadValues	Bad values errors
1.3.6.1.2.1.11.24.	snmpOutGenErrs	General errors
1.3.6.1.2.1.11.25.	snmpOutGetRequests	Total Get requests generated
1.3.6.1.2.1.11.26.	snmpOutGetNexts	Total GetNext requests generated
1.3.6.1.2.1.11.27.	snmpOutSetRequests	Total Set requests generated
1.3.6.1.2.1.11.28.	snmpOutGetResponses	Total GetNext responses generated
1.3.6.1.2.1.11.29.	snmpOutTraps	Total traps generated

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
1.3.6.1.2.1.11.30.	snmpEnableAuthenTraps	Is the authentication-failure trap enabled?
1.3.6.1.2.1.11.31.	snmpSilentDrops	No. of packets silently dropped
1.3.6.1.2.1.11.32.	snmpProxyDrops	No. of packets dropped by proxy
SYSTEM-MIB (1.3.6.1.2.1.1)	—	—
1.3.6.1.2.1.1.1.	sysDescr	System description
1.3.6.1.2.1.1.2.	sysObjectID	System OID
1.3.6.1.2.1.1.3.	sysUpTime	System uptime
1.3.6.1.2.1.1.4.	sysContact	Contact person for the system
1.3.6.1.2.1.1.5.	sysName	Name of the system
1.3.6.1.2.1.1.6.	sysLocation	Location of the system
1.3.6.1.2.1.1.7.	sysServices	Services offered by the system
TCP-MIB	tcpConnectionTable	show conn protocol tcp
—	**Index** • tcpConnectionLocalAddressType • tcpConnectionLocalAddress • tcpConnectionLocalPort • tcpConnectionRemAddressType • tcpConnectionRemAddress • tcpConnectionRemPort	—
1.3.6.1.2.1.6.19.1.1.	tcpConnectionLocalAddressType	ipv4 or ipv6
1.3.6.1.2.1.6.19.1.2.	tcpConnectionLocalAddress	*local_addr*
1.3.6.1.2.1.6.19.1.3.	tcpConnectionLocalPort	*local_port*
1.3.6.1.2.1.6.19.1.4.	tcpConnectionRemAddressType	ipv4 or ipv6
1.3.6.1.2.1.6.19.1.5.	tcpConnectionRemAddress	*foreign_addr*
1.3.6.1.2.1.6.19.1.6.	tcpConnectionRemPort	*foreign_port*
1.3.6.1.2.1.6.19.1.8	tcpConnectionProcess	Placeholder; always one.
—	**Unsupported Object** • tcpConnectionState	—
UDP-MIB	udpEndpointTable	show conn protocol udp

Table D-1 Cross-Reference of MIB Details and MIB Objects to CLI Commands (continued)

MIB Details	MIB Objects	CLI Field or Description
—	**Index** • udpEndpointLocalAddressType • udpEndpointLocalAddress • udpEndpointLocalPort • udpEndpointRemoteAddressType • udpEndpointRemoteAddress • udpEndpointRemotePort • udpEndpointInstance	—
1.3.6.1.2.1.7.7.1.1.	udpEndpointLocalAddressType	ipv4 or ipv6
1.3.6.1.2.1.7.7.1.2.	udpEndpointLocalAddress	*local_addr*
1.3.6.1.2.1.7.7.1.3.	udpEndpointLocalPort	*local_port*
1.3.6.1.2.1.7.7.1.4.	udpEndpointRemoteAddressType	ipv4 or ipv6
1.3.6.1.2.1.7.7.1.5.	udpEndpointRemoteAddress	*foreign_addr*
1.3.6.1.2.1.7.7.1.6.	udpEndpointRemotePort	*foreign_port*
1.3.6.1.2.1.7.7.1.7.	udpEndpointInstance	Always set to one. Not applicable to FWSM.
1.3.6.1.2.1.7.7.1.8.	udpEndpointProcess	Placeholder; always one.

APPENDIX E

Addresses, Protocols, and Ports

This appendix provides a quick reference for IP addresses, protocols, and applications. This appendix includes the following sections:

IPv4 Addresses and Subnet Masks

This section describes how to use IPv4 addresses with FWSM. An IPv4 address is a 32-bit number written in dotted-decimal notation: four 8-bit fields (octets) converted from binary to decimal numbers, separated by dots. The first part of an IP address identifies the network on which the host resides, while the second part identifies the particular host on the given network. The network number field is called the network prefix. All hosts on a given network share the same network prefix but must have a unique host number. In classful IP, the class of the address determines the boundary between the network prefix and the host number.

This section includes the following topics:

Classes

IP host addresses are divided into three different address classes: Class A, Class B, and Class C. Each class fixes the boundary between the network prefix and the host number at a different point within the 32-bit address. Class D addresses are reserved for multicast IP.

- Class A addresses (1.xxx.xxx.xxx through 126.xxx.xxx.xxx) use only the first octet as the network prefix.
- Class B addresses (128.0.xxx.xxx through 191.255.xxx.xxx) use the first two octets as the network prefix.
- Class C addresses (192.0.0.xxx through 223.255.255.xxx) use the first three octets as the network prefix.

Because Class A addresses have 16,777,214 host addresses, and Class B addresses 65,534 hosts, you can use subnet masking to break these huge networks into smaller subnets.

Private Networks

If you need large numbers of addresses on your network, and they do not need to be routed on the Internet, you can use private IP addresses that the Internet Assigned Numbers Authority (IANA) recommends (see RFC 1918). The following address ranges are designated as private networks that should not be advertised:

- 10.0.0.0 through 10.255.255.255
- 172.16.0.0 through 172.31.255.255
- 192.168.0.0 through 192.168.255.255

Subnet Masks

A subnet mask lets you convert a single Class A, B, or C network into multiple networks. With a subnet mask, you can create an extended network prefix that adds bits from the host number to the network prefix. For example, a Class C network prefix always consists of the first three octets of the IP address. But a Class C extended network prefix uses part of the fourth octet as well.

Subnet masking is easy to understand if you use binary notation instead of dotted decimal. The bits in the subnet mask have a one-to-one correspondence with the Internet address:

- The bits are set to 1 if the corresponding bit in the IP address is part of the extended network prefix.
- The bits are set to 0 if the bit is part of the host number.

Example 1: If you have the Class B address 129.10.0.0 and you want to use the entire third octet as part of the extended network prefix instead of the host number, you must specify a subnet mask of 11111111.11111111.11111111.00000000. This subnet mask converts the Class B address into the equivalent of a Class C address, where the host number consists of the last octet only.

Example 2: If you want to use only part of the third octet for the extended network prefix, then you must specify a subnet mask like 11111111.11111111.11111000.00000000, which uses only 5 bits of the third octet for the extended network prefix.

You can write a subnet mask as a dotted-decimal mask or as a */bits* (“slash *bits*”) mask. In Example 1, for a dotted-decimal mask, you convert each binary octet into a decimal number: 255.255.255.0. For a */bits* mask, you add the number of 1s: /24. In Example 2, the decimal number is 255.255.248.0 and the /bits is /21.

You can also supernet multiple Class C networks into a larger network by using part of the third octet for the extended network prefix. For example, 192.168.0.0/20.

This section includes the following topics:

Determining the Subnet Mask

To determine the subnet mask based on how many hosts you want, see Table E-1.

Table E-1 Hosts, Bits, and Dotted-Decimal Masks

Hosts[1]	/Bits Mask	Dotted-Decimal Mask
16,777,216	/8	255.0.0.0 Class A Network
65,536	/16	255.255.0.0 Class B Network
32,768	/17	255.255.128.0
16,384	/18	255.255.192.0
8192	/19	255.255.224.0
4096	/20	255.255.240.0
2048	/21	255.255.248.0
1024	/22	255.255.252.0
512	/23	255.255.254.0
256	/24	255.255.255.0 Class C Network
128	/25	255.255.255.128
64	/26	255.255.255.192
32	/27	255.255.255.224
16	/28	255.255.255.240
8	/29	255.255.255.248
4	/30	255.255.255.252
Do not use	/31	255.255.255.254
1	/32	255.255.255.255 Single Host Address

1. The first and last number of a subnet are reserved, except for /32, which identifies a single host.

Determining the Address to Use with the Subnet Mask

The following sections describe how to determine the network address to use with a subnet mask for a Class C-size and a Class B-size network. This section includes the following topics:

Class C-Size Network Address

For a network between 2 and 254 hosts, the fourth octet falls on a multiple of the number of host addresses, starting with 0. For example, the 8-host subnets (/29) of 192.168.0.x are as follows:

Subnet with Mask /29 (255.255.255.248)	Address Range[1]
192.168.0.0	192.168.0.0 to 192.168.0.7
192.168.0.8	192.168.0.8 to 192.168.0.15
192.168.0.16	192.168.0.16 to 192.168.0.31
…	…
192.168.0.248	192.168.0.248 to 192.168.0.255

1. The first and last address of a subnet are reserved. In the first subnet example, you cannot use 192.168.0.0 or 192.168.0.7.

Class B-Size Network Address

To determine the network address to use with the subnet mask for a network with between 254 and 65,534 hosts, you need to determine the value of the third octet for each possible extended network prefix. For example, you might want to subnet an address like 10.1.*x*.0, where the first two octets are fixed because they are used in the extended network prefix, and the fourth octet is 0 because all bits are used for the host number.

To determine the value of the third octet, perform the following steps:

Step 1 Calculate how many subnets you can make from the network by dividing 65,536 (the total number of addresses using the third and fourth octet) by the number of host addresses you want.

For example, 65,536 divided by 4096 hosts equals 16.

Therefore, there are 16 subnets of 4096 addresses each in a Class B-size network.

Step 2 Determine the multiple of the third octet value by dividing 256 (the number of values for the third octet) by the number of subnets:

In this example, 256/16 = 16.

The third octet falls on a multiple of 16, starting with 0.

Therefore, the 16 subnets of the network 10.1 are as follows:

Subnet with Mask /20 (255.255.240.0)	Address Range[1]
10.1.0.0	10.1.0.0 to 10.1.15.255
10.1.16.0	10.1.16.0 to 10.1.31.255
10.1.32.0	10.1.32.0 to 10.1.47.255
…	…
10.1.240.0	10.1.240.0 to 10.1.255.255

1. The first and last address of a subnet are reserved. In the first subnet example, you cannot use 10.1.0.0 or 10.1.15.255.

IPv6 Addresses

IPv6 is the next generation of the Internet Protocol after IPv4. It provides an expanded address space, a simplified header format, improved support for extensions and options, flow labeling capability, and authentication and privacy capabilities. IPv6 is described in RFC 2460. The IPv6 addressing architecture is described in RFC 3513.

This section describes the IPv6 address format and architecture and includes the following topics:

Note This section describes the IPv6 address format, the types, and prefixes. For information about configuring FWSM to use IPv6, see Chapter 10, "Configuring IPv6."

IPv6 Address Format

IPv6 addresses are represented as a series of eight 16-bit hexadecimal fields separated by colons (:) in the format: x:x:x:x:x:x:x:x. The following are two examples of IPv6 addresses:

- 2001:0DB8:7654:3210:FEDC:BA98:7654:3210
- 2001:0DB8:0000:0000:0008:0800:200C:417A

Note The hexadecimal letters in IPv6 addresses are not case-sensitive.

It is not necessary to include the leading zeros in an individual field of the address. But each field must contain at least one digit. So the example address 2001:0DB8:0000:0000:0008:0800:200C:417A can be shortened to 2001:0DB8:0:0:8:800:200C:417A by removing the leading zeros from the third through sixth fields from the left. The fields that contained all zeros (the third and fourth fields from the left) were shortened to a single zero. The fifth field from the left had the three leading zeros removed, leaving a single 8 in that field, and the sixth field from the left had the one leading zero removed, leaving 800 in that field.

It is common for IPv6 addresses to contain several consecutive hexadecimal fields of zeros. You can use two colons (::) to compress consecutive fields of zeros at the beginning, middle, or end of an IPv6 address (the colons represent the successive hexadecimal fields of zeros). Table E-2 shows several examples of address compression for different types of IPv6 address.

Table E-2 IPv6 Address Compression Examples

Address Type	Standard Form	Compressed Form
Unicast	2001:0DB8:0:0:0:BA98:0:3210	2001:0DB8::BA98:0:3210
Multicast	FF01:0:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

Note Two colons (::) can be used only once in an IPv6 address to represent successive fields of zeros.

An alternative form of the IPv6 format is often used when dealing with an environment that contains both IPv4 and IPv6 addresses. This alternative has the format x:x:x:x:x:x:y.y.y.y, where x represent the hexadecimal values for the six high-order parts of the IPv6 address and y represent decimal values for the 32-bit IPv4 part of the address (which takes the place of the remaining two 16-bit parts of the IPv6 address). For example, the IPv4 address 192.168.1.1 could be represented as the IPv6 address 0:0:0:0:0:0:FFFF:192.168.1.1, or ::FFFF:192.168.1.1.

IPv6 Address Types

The following are the three main types of IPv6 addresses:

- **Unicast**—A unicast address is an identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address. An interface may have more than one unicast address assigned to it.
- **Multicast**—A multicast address is an identifier for a set of interfaces. A packet sent to a multicast address is delivered to all addresses identified by that address.
- **Anycast**—An anycast address is an identifier for a set of interfaces. Unlike a multicast address, a packet sent to an anycast address is only delivered to the "nearest" interface, as determined by the measure of distances for the routing protocol.

Note There are no broadcast addresses in IPv6. Multicast addresses provide the broadcast functionality.

This section includes the following topics:

Unicast Addresses

This section describes IPv6 unicast addresses. Unicast addresses identify an interface on a network node.

This section includes the following topics:

Global Address

The general format of an IPv6 global unicast address is a global routing prefix followed by a subnet ID followed by an interface ID. The global routing prefix can be any prefix not reserved by another IPv6 address type (see IPv6 Address Prefixes, page E-10, for information about the IPv6 address type prefixes).

All global unicast addresses, other than those that start with binary 000, have a 64-bit interface ID in the Modified EUI-64 format. See Interface Identifiers, page E-8, for more information about the Modified EUI-64 format for interface identifiers.

Global unicast address that start with the binary 000 do not have any constraints on the size or structure of the interface ID portion of the address. One example of this type of address is an IPv6 address with an embedded IPv4 address (see IPv4-Compatible IPv6 Addresses, page E-7).

Site-Local Address

Site-local addresses are used for addressing within a site. They can be use to address an entire site without using a globally unique prefix. Site-local addresses have the prefix FEC0::/10, followed by a 54-bit subnet ID, and end with a 64-bit interface ID in the modified EUI-64 format.

Site-local Routers do not forward any packets that have a site-local address for a source or destination outside of the site. Therefore, site-local addresses can be considered private addresses.

Link-Local Address

All interfaces are required to have at least one link-local address. You can configure multiple IPv6 addresses per interfaces, but only one link-local address.

A link-local address is an IPv6 unicast address that can be automatically configured on any interface using the link-local prefix FE80::/10 and the interface identifier in modified EUI-64 format. Link-local addresses are used in the neighbor discovery protocol and the stateless autoconfiguration process. Nodes with a link-local address can communicate; they do not need a site-local or globally unique address to communicate.

Routers do not forward any packets that have a link-local address for a source or destination. Therefore, link-local addresses can be considered private addresses.

IPv4-Compatible IPv6 Addresses

There are two types of IPv6 addresses that can contain IPv4 addresses.

The first type is the “IPv4-compatibly IPv6 address.” The IPv6 transition mechanisms include a technique for hosts and routers to dynamically tunnel IPv6 packets over IPv4 routing infrastructure. IPv6 nodes that use this technique are assigned special IPv6 unicast addresses that carry a global IPv4 address in the low-order 32 bits. This type of address is termed an “IPv4-compatible IPv6 address” and has the format ::y.y.y.y, where y.y.y.y is an IPv4 unicast address.

Note The IPv4 address used in the “IPv4-compatible IPv6 address” must be a globally-unique IPv4 unicast address.

The second type of IPv6 address which holds an embedded IPv4 address is called the “IPv4-mapped IPv6 address.” This address type is used to represent the addresses of IPv4 nodes as IPv6 addresses. This type of address has the format ::FFFF:y.y.y.y, where y.y.y.y is an IPv4 unicast address.

Unspecified Address

The unspecified address, 0:0:0:0:0:0:0:0, indicates the absence of an IPv6 address. For example, a newly initialized node on an IPv6 network may use the unspecified address as the source address in its packets until it receives its IPv6 address.

Note The IPv6 unspecified address cannot be assigned to an interface. The unspecified IPv6 addresses must not be used as destination addresses in IPv6 packets or the IPv6 routing header.

Loopback Address

The loopback address, 0:0:0:0:0:0:0:1, may be used by a node to send an IPv6 packet to itself. The loopback address in IPv6 functions the same as the loopback address in IPv4 (127.0.0.1).

Note The IPv6 loopback address cannot be assigned to a physical interface. A packet that has the IPv6 loopback address as its source or destination address must remain within the node that created the packet. IPv6 routers do not forward packets that have the IPv6 loopback address as their source or destination address.

Interface Identifiers

Interface identifiers in IPv6 unicast addresses are used to identify the interfaces on a link. They need to be unique within a subnet prefix. In many cases, the interface identifier is derived from the interface link-layer address. The same interface identifier may be used on multiple interfaces of a single node, as long as those interfaces are attached to different subnets.

For all unicast addresses, except those that start with the binary 000, the interface identifier is required to be 64 bits long and to be constructed in the Modified EUI-64 format. The Modified EUI-64 format is created from the 48-bit MAC address by inverting the universal/local bit in the address and by inserting the hexadecimal number FFFE between the upper three bytes and lower three bytes of the of the MAC address.

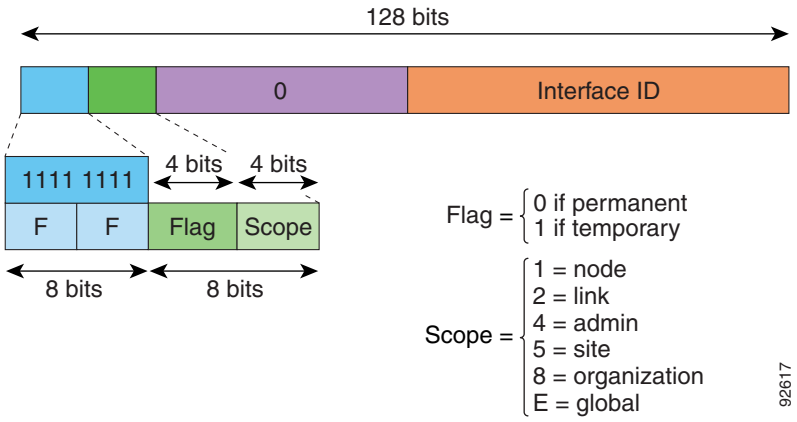
For example, and interface with the MAC address of 00E0.b601.3B7A would have a 64-bit interface ID of 02E0:B6FF:FE01:3B7A.

Multicast Address

An IPv6 multicast address is an identifier for a group of interfaces, typically on different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. An interface may belong to any number of multicast groups.

An IPv6 multicast address has a prefix of FF00::/8 (1111 1111). The octet following the prefix defines the type and scope of the multicast address. A permanently assigned (“well known”) multicast address has a flag parameter equal to 0; a temporary (“transient”) multicast address has a flag parameter equal to 1. A multicast address that has the scope of a node, link, site, or organization, or a global scope has a scope parameter of 1, 2, 5, 8, or E, respectively. For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link scope. Figure E-1 shows the format of the IPv6 multicast address.

Figure E-1 IPv6 Multicast Address Format



IPv6 nodes (hosts and routers) are required to join the following multicast groups:

- The All Nodes multicast addresses:
  - FF01:: (interface-local)
  - FF02:: (link-local)
- The Solicited-Node Address for each IPv6 unicast and anycast address on the node: FF02:0:0:0:0:1:FFXX:XXXX/104, where XX:XXXX is the low-order 24-bits of the unicast or anycast address.

Note Solicited-Node addresses are used in Neighbor Solicitation messages.

IPv6 routers are required to join the following multicast groups:

- FF01::2 (interface-local)
- FF02::2 (link-local)
- FF05::2 (site-local)

Multicast address should not be used as source addresses in IPv6 packets.

Note There are no broadcast addresses in IPv6. IPv6 multicast addresses are used instead of broadcast addresses.

Anycast Address

The IPv6 anycast address is a unicast address that is assigned to more than one interface (typically belonging to different nodes). A packet that is routed to an anycast address is routed to the nearest interface having that address, the nearness being determined by the routing protocol in effect.

Anycast addresses are allocated from the unicast address space. An anycast address is simply a unicast address that has been assigned to more than one interface, and the interfaces must be configured to recognize the address as an anycast address.

The following restrictions apply to anycast addresses:

- An anycast address cannot be used as the source address for an IPv6 packet.
- An anycast address cannot be assigned to an IPv6 host; it can only be assigned to an IPv6 router.

Note Anycast addresses are not supported on FWSM.

Required Addresses

IPv6 hosts must, at a minimum, be configured with the following addresses (either automatically or manually):

- A link-local address for each interface.
- The loopback address.
- The All-Nodes multicast addresses
- A Solicited-Node multicast address for each unicast or anycast address.

IPv6 routers must, at a minimum, be configured with the following addresses (either automatically or manually):

- The required host addresses.
- The Subnet-Router anycast addresses for all interfaces for which it is configured to act as a router.
- The All-Routers multicast addresses.

IPv6 Address Prefixes

An IPv6 address prefix, in the format ipv6-prefix/prefix-length, can be used to represent bit-wise contiguous blocks of the entire address space. The IPv6-prefix must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example, 2001:0DB8:8086:6502::/32 is a valid IPv6 prefix.

The IPv6 prefix identifies the type of IPv6 address. Table E-3 shows the prefixes for each IPv6 address type.

Table E-3 IPv6 Address Type Prefixes

Address Type	Binary Prefix	IPv6 Notation
Unspecified	000...0 (128 bits)	::/128
Loopback	000...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local (unicast)	1111111010	FE80::/10
Site-Local (unicast)	1111111111	FEC0::/10
Global (unicast)	All other addresses.	
Anycast	Taken from the unicast address space.	

Protocols and Applications

Table E-4 lists the protocol literal values and port numbers; either can be entered in FWSM commands.

Table E-4 Protocol Literal Values

Literal	Value	Description
ah	51	Authentication Header for IPv6, RFC 1826.
eigrp	88	Enhanced Interior Gateway Routing Protocol.
esp	50	Encapsulated Security Payload for IPv6, RFC 1827.
gre	47	Generic Routing Encapsulation.
icmp	1	Internet Control Message Protocol, RFC 792.
icmp6	58	Internet Control Message Protocol for IPv6, RFC 2463.
igmp	2	Internet Group Management Protocol, RFC 1112.
igrp	9	Interior Gateway Routing Protocol.
ip	0	Internet Protocol.
ipinip	4	IP-in-IP encapsulation.
ipsec	50	IP Security. Entering the ipsec protocol literal is equivalent to entering the esp protocol literal.
nos	94	Network Operating System (Novell's NetWare).
ospf	89	Open Shortest Path First routing protocol, RFC 1247.
pcp	108	Payload Compression Protocol.
pim	103	Protocol Independent Multicast.
pptp	47	Point-to-Point Tunneling Protocol. Entering the pptp protocol literal is equivalent to entering the gre protocol literal.
snp	109	Sitara Networks Protocol.
tcp	6	Transmission Control Protocol, RFC 793.
udp	17	User Datagram Protocol, RFC 768.

Protocol numbers can be viewed online at the IANA website:

http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

TCP and UDP Ports

Table E-5 lists the literal values and port numbers; either can be entered in FWSM commands. See the following caveats:

- FWSM uses port 1521 for SQL*Net. This is the default port used by Oracle for SQL*Net. This value, however, does not agree with IANA port assignments.
- FWSM listens for RADIUS on ports 1645 and 1646. If your RADIUS server uses the standard ports 1812 and 1813, you can configure FWSM to listen to those ports using the **authentication-port** and **accounting-port** commands.

- To assign a port for DNS access, use the **domain** literal value, not **dns**. If you use **dns**, FWSM assumes you meant to use the **dnsix** literal value.

Port numbers can be viewed online at the IANA website:

http://www.iana.org/assignments/port-numbers

Table E-5 Port Literal Values

Literal	TCP or UDP?	Value	Description
aol	TCP	5190	America Online
bgp	TCP	179	Border Gateway Protocol, RFC 1163
biff	UDP	512	Used by mail system to notify users that new mail is received
bootpc	UDP	68	Bootstrap Protocol Client
bootps	UDP	67	Bootstrap Protocol Server
chargen	TCP	19	Character Generator
citrix-ica	TCP	1494	Citrix Independent Computing Architecture (ICA) protocol
cmd	TCP	514	Similar to **exec** except that **cmd** has automatic authentication
ctiqbe	TCP	2748	Computer Telephony Interface Quick Buffer Encoding
daytime	TCP	13	Day time, RFC 867
discard	TCP, UDP	9	Discard
domain	TCP, UDP	53	DNS
dnsix	UDP	195	DNSIX Session Management Module Audit Redirector
echo	TCP, UDP	7	Echo
exec	TCP	512	Remote process execution
finger	TCP	79	Finger
ftp	TCP	21	File Transfer Protocol (control port)
ftp-data	TCP	20	File Transfer Protocol (data port)
gopher	TCP	70	Gopher
https	TCP	443	HTTP over SSL
h323	TCP	1720	H.323 call signalling
hostname	TCP	101	NIC Host Name Server
ident	TCP	113	Ident authentication service
imap4	TCP	143	Internet Message Access Protocol, version 4
irc	TCP	194	Internet Relay Chat protocol
isakmp	UDP	500	Internet Security Association and Key Management Protocol
kerberos	TCP, UDP	750	Kerberos

Table E-5 Port Literal Values (continued)

Literal	TCP or UDP?	Value	Description
klogin	TCP	543	KLOGIN
kshell	TCP	544	Korn Shell
ldap	TCP	389	Lightweight Directory Access Protocol
ldaps	TCP	636	Lightweight Directory Access Protocol (SSL)
lpd	TCP	515	Line Printer Daemon - printer spooler
login	TCP	513	Remote login
lotusnotes	TCP	1352	IBM Lotus Notes
mobile-ip	UDP	434	MobileIP-Agent
nameserver	UDP	42	Host Name Server
netbios-ns	UDP	137	NetBIOS Name Service
netbios-dgm	UDP	138	NetBIOS Datagram Service
netbios-ssn	TCP	139	NetBIOS Session Service
nntp	TCP	119	Network News Transfer Protocol
ntp	UDP	123	Network Time Protocol
pcanywhere-status	UDP	5632	pcAnywhere status
pcanywhere-data	TCP	5631	pcAnywhere data
pim-auto-rp	TCP, UDP	496	Protocol Independent Multicast, reverse path flooding, dense mode
pop2	TCP	109	Post Office Protocol - Version 2
pop3	TCP	110	Post Office Protocol - Version 3
pptp	TCP	1723	Point-to-Point Tunneling Protocol
radius	UDP	1645	Remote Authentication Dial-In User Service
radius-acct	UDP	1646	Remote Authentication Dial-In User Service (accounting)
rip	UDP	520	Routing Information Protocol
secureid-udp	UDP	5510	SecureID over UDP
smtp	TCP	25	Simple Mail Transport Protocol
snmp	UDP	161	Simple Network Management Protocol
snmptrap	UDP	162	Simple Network Management Protocol - Trap
sqlnet	TCP	1521	Structured Query Language Network
ssh	TCP	22	Secure Shell
sunrpc (rpc)	TCP, UDP	111	Sun Remote Procedure Call
syslog	UDP	514	System Log
tacacs	TCP, UDP	49	Terminal Access Controller Access Control System Plus
talk	TCP, UDP	517	Talk
telnet	TCP	23	RFC 854 Telnet

Table E-5 Port Literal Values (continued)

Literal	TCP or UDP?	Value	Description
tftp	UDP	69	Trivial File Transfer Protocol
time	UDP	37	Time
uucp	TCP	540	UNIX-to-UNIX Copy Program
who	UDP	513	Who
whois	TCP	43	Who Is
www	TCP	80	World Wide Web
xdmcp	UDP	177	X Display Manager Control Protocol

Local Ports and Protocols

Table E-6 lists the protocols, TCP ports, and UDP ports that FWSM may open to process traffic destined to FWSM. Unless you enable the features and services listed in Table E-6, FWSM does *not* open any local protocols or any TCP or UDP ports. You must configure a feature or service for FWSM to open the default listening protocol or port. In many cases you can configure ports other than the default port when you enable a feature or service.

Table E-6 Protocols and Ports Opened by Features and Services

Feature or Service	Protocol	Port Number	Comments
DHCP	UDP	67,68	—
Failover Control	108	N/A	—
HTTP	TCP	80	—
HTTPS	TCP	443	—
ICMP	1	N/A	—
IGMP	2	N/A	Protocol only open on destination IP address 224.0.0.1
ISAKMP/IKE	UDP	500	Configurable.
IPSec (ESP)	50	N/A	—
NTP	UDP	123	—
OSPF	89	N/A	Protocol only open on destination IP address 224.0.0.5 and 224.0.0.6
PIM	103	N/A	Protocol only open on destination IP address 224.0.0.13
RIP	UDP	520	—
RIPv2	UDP	520	Port only open on destination IP address 224.0.0.9
SNMP	UDP	161	Configurable.
SSH	TCP	22	—

Table E-6 Protocols and Ports Opened by Features and Services (continued)

Feature or Service	Protocol	Port Number	Comments
Stateful Update	105	N/A	—
Telnet	TCP	23	—

ICMP Types

Table E-7 lists the ICMP type numbers and names that you can enter in FWSM commands:

Table E-7 ICMP Types

ICMP Number	ICMP Name
0	echo-reply
3	unreachable
4	source-quench
5	redirect
6	alternate-address
8	echo
9	router-advertisement
10	router-solicitation
11	time-exceeded
12	parameter-problem
13	timestamp-request
14	timestamp-reply
15	information-request
16	information-reply
17	mask-request
18	mask-reply
31	conversion-error
32	mobile-redirect

GLOSSARY

A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X

Numerics

3DES See DES.

A

AAA Authentication, authorization, and accounting. See also TACACS+ and RADIUS.

ABR Area Border Router. In OSPF, a router with interfaces in multiple areas.

ASBR Autonomous System Boundary Router

ACE Access Control Entry. Information entered into the configuration that lets you specify what type of traffic to permit or deny on an interface. By default, traffic that is not explicitly permitted is denied.

Access Modes The FWSM CLI uses several command modes. The commands available in each mode vary. See also user EXEC mode, privileged EXEC mode, global configuration mode, command-specific configuration mode.

ACL access control list. A collection of ACEs. An ACL lets you specify what type of traffic to allow on an interface. By default, traffic that is not explicitly permitted is denied. ACLs are usually applied to the interface which is the source of inbound traffic. See also rule, outbound ACL.

ActiveX A set of object-oriented programming technologies and tools used to create mobile or portable programs. An ActiveX program is roughly equivalent to a Java applet.

Address Resolution Protocol See ARP.

address translation The translation of a network address and/or port to another network address/or port. See also IP address, interface PAT, NAT, PAT, Static PAT, xlate.

AES Advanced Encryption Standard. A symmetric block cipher that can encrypt and decrypt information. The AES algorithm is capable of using cryptographic keys of 128, 192 and 256 bits to encrypt and decrypt data in blocks of 128 bits. See also DES.

AH Authentication Header. An IP protocol (type 51) that can ensure data integrity, authentication, and replay detection. AH is embedded in the data to be protected (a full IP datagram, for example). AH can be used either by itself or with ESP. This is an older IPSec protocol that is less important in most networks than ESP. AH provides authentication services but does not provide encryption services. It is provided to ensure compatibility with IPSec peers that do not support ESP, which provides both authentication and encryption. See also encryption and VPN. Refer to the RFC 2402.

A record address "A" stands for address, and refers to name-to-address mapped records in DNS.

ARP Address Resolution Protocol. A low-level TCP/IP protocol that maps a hardware address, or MAC address, to an IP address. An example hardware address is 00:00:a6:00:01:ba. The first three groups of characters (00:00:a6) identify the manufacturer; the rest of the characters (00:01:ba) identify the system card. ARP is defined in RFC 826.

ASA Adaptive Security Algorithm. Used by the FWSM to perform inspections. ASA allows one-way (inside to outside) connections without an explicit configuration for each internal system and application. See also inspection engine.

ASA adaptive security appliance.

ASDM Adaptive Security Device Manager. An application for managing and configuring a single FWSM.

asymmetric encryption Also called public key systems, asymmetric encryption allows anyone to obtain access to the public key of anyone else. Once the public key is accessed, one can send an encrypted message to that person using the public key. See also encryption, public key.

authentication Cryptographic protocols and services that verify the identity of users and the integrity of data. One of the functions of the IPSec framework. Authentication establishes the integrity of datastream and ensures that it is not tampered with in transit. It also provides confirmation about the origin of the datastream. See also AAA, encryption, and VPN.

B

BGP Border Gateway Protocol. BGP performs interdomain routing in TCP/IP networks. BGP is an Exterior Gateway Protocol, which means that it performs routing between multiple autonomous systems or domains and exchanges routing and access information with other BGP systems. The FWSM does not support BGP. See also EGP.

BLT stream Bandwidth Limited Traffic stream. Stream or flow of packets whose bandwidth is constrained.

BOOTP Bootstrap Protocol. Lets diskless workstations boot over the network as is described in RFC 951 and RFC 1542.

BPDU Bridge Protocol Data Unit. Spanning-Tree Protocol hello packet that is sent out at configurable intervals to exchange information among bridges in the network. Protocol data unit is the OSI term for packet.

C

CA Certificate Authority, Certification Authority. A third-party entity that is responsible for issuing and revoking certificates. Each device with the public key of the CA can authenticate a device that has a certificate issued by the CA. The term CA also refers to software that provides CA services. See also certificate, CRL, public key, RA.

cache A temporary repository of information accumulated from previous task executions that can be reused, decreasing the time required to perform the tasks.

CBC Cipher Block Chaining. A cryptographic technique that increases the encryption strength of an algorithm. CBC requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet.

certificate A signed cryptographic object that contains the identity of a user or device and the public key of the CA that issued the certificate. Certificates have an expiration date and may also be placed on a CRL if known to be compromised. Certificates also establish non-repudiation for IKE negotiation, which means that you can prove to a third party that IKE negotiation was completed with a specific peer.

CHAP Challenge Handshake Authentication Protocol.

CLI command-line interface. The primary interface for entering configuration and monitoring commands to the FWSM.

client/server computing Distributed computing (processing) network systems in which transaction responsibilities are divided into two parts: client (front end) and server (back end). Also called distributed computing. See also RPC.

command-specific configuration mode From global configuration mode, some commands enter a command-specific configuration mode. All user EXEC, privileged EXEC, global configuration, and command-specific configuration commands are available in this mode. See also global configuration mode, privileged EXEC mode, user EXEC mode.

configuration, config, config file A file on the FWSM that represents the equivalent of settings, preferences, and properties administered by ASDM or the CLI.

cookie A cookie is a object stored by a browser. Cookies contain information, such as user preferences, to persistent storage.

CPU Central Processing Unit. Main processor.

CRC cyclical redundancy check. Error-checking technique in which the frame recipient calculates a remainder by dividing frame contents by a prime binary divisor and compares the calculated remainder to a value stored in the frame by the sending node.

CRL Certificate Revocation List. A digitally signed message that lists all of the current but revoked certificates listed by a given CA. This is analogous to a book of stolen charge card numbers that allow stores to reject bad credit cards. When certificates are revoked, they are added to a CRL. When you implement authentication using certificates, you can choose to use CRLs or not. Using CRLs lets you easily revoke certificates before they expire, but the CRL is generally only maintained by the CA or an RA. If you are using CRLs and the connection to the CA or RA is not available when authentication is requested, the authentication request will fail. See also CA, certificate, public key, RA.

CRV Call Reference Value. Used by H.225.0 to distinguish call legs signalled between two entities.

cryptography Encryption, authentication, integrity, keys and other services used for secure communication over networks. See also VPN and IPSec.

crypto map A data structure with a unique name and sequence number that is used for configuring VPNs on the FWSM. A crypto map selects data flows that need security processing and defines the policy for these flows and the crypto peer that traffic needs to go to. A crypto map is applied to an interface. Crypto maps contain the ACLs, encryption standards, peers, and other parameters necessary to specify security policies for VPNs using IKE and IPSec. See also VPN.

CTIQBE Computer Telephony Interface Quick Buffer Encoding. A protocol used in IP telephony between the Cisco CallManager and CTI TAPI and JTAPI applications. CTIQBE is used by the TAPI/JTAPI protocol inspection module and supports NAT, PAT, and bi-directional NAT. This enables Cisco IP SoftPhone and other Cisco TAPI/JTAPI applications to communicate with Cisco CallManager for call setup and voice traffic across the FWSM.

cut-through proxy Enables the FWSM to provide faster traffic flow after user authentication. The cut-through proxy challenges a user initially at the application layer. After the security appliance authenticates the user, it shifts the session flow and all traffic flows directly and quickly between the source and destination while maintaining session state information.

D

data confidentiality Describes any method that manipulates data so that no attacker can read it. This is commonly achieved by data encryption and keys that are only available to the parties involved in the communication.

data integrity Describes mechanisms that, through the use of encryption based on secret key or public key algorithms, allow the recipient of a piece of protected data to verify that the data has not been modified in transit.

data origin authentication A security service where the receiver can verify that protected data could have originated only from the sender. This service requires a data integrity service plus a key distribution mechanism, where a secret key is shared only between the sender and receiver.

decryption Application of a specific algorithm or cipher to encrypted data so as to render the data comprehensible to those who are authorized to see the information. See also encryption.

DES Data encryption standard. DES was published in 1977 by the National Bureau of Standards and is a secret key encryption scheme based on the Lucifer algorithm from IBM. Cisco uses DES in classic crypto (40-bit and 56-bit key lengths), IPSec crypto (56-bit key), and 3DES (triple DES), which performs encryption three times using a 56-bit key. 3DES is more secure than DES but requires more processing for encryption and decryption. See also AES, ESP.

DHCP Dynamic Host Configuration Protocol. Provides a mechanism for allocating IP addresses to hosts dynamically, so that addresses can be reused when hosts no longer need them and so that mobile computers, such as laptops, receive an IP address applicable to the LAN to which it is connected.

Diffie-Hellman A public key cryptography protocol that allows two parties to establish a shared secret over insecure communications channels. Diffie-Hellman is used within IKE to establish session keys. Diffie-Hellman is a component of Oakley key exchange.

Diffie-Hellman Group 1, Group 2, Group 5, Group 7 Diffie-Hellman refers to a type of public key cryptography using asymmetric encryption based on large prime numbers to establish both Phase 1 and Phase 2 SAs. Group 1 provides a smaller prime number than Group 2 but may be the only version supported by some IPSec peers. Diffe-Hellman Group 5 uses a 1536-bit prime number, is the most secure, and is recommended for use with AES. Group 7 has an elliptical curve field size of 163 bits and is for use with the Movian VPN client, but works with any peer that supports Group 7 (ECC). See also VPN and encryption.

digital certificate See certificate.

DMZ See interface.

DN Distinguished Name. Global, authoritative name of an entry in the OSI Directory (X.500).

DNS Domain Name System (or Service). An Internet service that translates domain names into IP addresses.

DoS Denial of Service. A type of network attack in which the goal is to render a network service unavailable.

DSL digital subscriber line. Public network technology that delivers high bandwidth over conventional copper wiring at limited distances. DSL is provisioned via modem pairs, with one modem located at a central office and the other at the customer site. Because most DSL technologies do not use the whole bandwidth of the twisted pair, there is room remaining for a voice channel.

DSP digital signal processor. A DSP segments a voice signal into frames and stores them in voice packets.

DSS Digital Signature Standard. A digital signature algorithm designed by The US National Institute of Standards and Technology and based on public-key cryptography. DSS does not do user datagram encryption. DSS is a component in classic crypto, as well as the Redcreek IPSec card, but not in IPSec implemented in Cisco IOS software.

Dynamic NAT See NAT and address translation.

Dynamic PAT Dynamic Port Address Translation. Dynamic PAT lets multiple outbound sessions appear to originate from a single IP address. With PAT enabled, the FWSM chooses a unique port number from the PAT IP address for each outbound translation slot (xlate). This feature is valuable when an ISP cannot allocate enough unique IP addresses for your outbound connections. The global pool addresses always come first, before a PAT address is used. See also NAT, Static PAT, and xlate.

E

ECHO See Ping, ICMP. See also inspection engine.

EGP Exterior Gateway Protocol. Replaced by BGP. The FWSM does not support EGP. See also BGP.

EIGRP Enhanced Interior Gateway Routing Protocol. The FWSM does not support EIGRP.

EMBLEM Enterprise Management BaseLine Embedded Manageability. A syslog format designed to be consistent with the Cisco IOS system log format and is more compatible with CiscoWorks management applications.

encryption Application of a specific algorithm or cipher to data so as to render the data incomprehensible to those unauthorized to see the information. See also decryption.

EPM Endpoint Mapper.

ESMTP Extended SMTP. Extended version of SMTP that includes additional functionality, such as delivery notification and session delivery. ESMTP is described in RFC 1869, SMTP Service Extensions.

ESP Encapsulating Security Payload. An IPSec protocol, ESP provides authentication and encryption services for establishing a secure tunnel over an insecure network. For more information, refer to RFCs 2406 and 1827.

F

failover, failover mode Failover lets you configure two FWSMs so that one will take over operation if the other one fails. The FWSM supports two failover configurations, Active/Active failover and Active/Standby failover. Each failover configuration has its own method for determining and performing failover. With Active/Active failover, both units can pass network traffic. This lets you configure load balancing on your network. Active/Active failover is only available on units running in multiple context mode. With Active/Standby failover, only one unit passes traffic while the other unit waits in a standby state. Active/Standby failover is available on units running in either single or multiple context mode.

Fixup See inspection engine.

Flash, Flash memory A nonvolatile storage device used to store the configuration file when the FWSM is powered down.

FQDN/IP Fully qualified domain name/IP address. IPSec parameter that identifies peers that are security gateways.

FragGuard Provides IP fragment protection and performs full reassembly of all ICMP error messages and virtual reassembly of the remaining IP fragments that are routed through the FWSM.

FTP File Transfer Protocol. Part of the TCP/IP protocol stack, used for transferring files between hosts.

G

GGSN gateway GPRS support node. A wireless gateway that allows mobile cell phone users to access the public data network or specified private IP networks.

global configuration mode Global configuration mode lets you to change the FWSM configuration. All user EXEC, privileged EXEC, and global configuration commands are available in this mode. See also user EXEC mode, privileged EXEC mode, command-specific configuration mode.

GMT Greenwich Mean Time. Replaced by UTC (Coordinated Universal Time) in 1967 as the world time standard.

GPRS general packet radio service. A service defined and standardized by the European Telecommunication Standards Institute. GPRS is an IP-packet-based extension of GSM networks and provides mobile, wireless, data communications

GRE Generic Routing Encapsulation described in RFCs 1701 and 1702. GRE is a tunneling protocol that can encapsulate a wide variety of protocol packet types inside IP tunnels, creating a virtual point-to-point link to routers at remote points over an IP network. By connecting multiprotocol subnetworks in a single-protocol backbone environment, IP tunneling using GRE allows network expansion across a single protocol backbone environment.

GSM Global System for Mobile Communication. A digital, mobile, radio standard developed for mobile, wireless, voice communications.

GSN Global Seamless Network.

GTP GPRS tunneling protocol. GTP handles the flow of user packet data and signaling information between the SGSN and GGSN in a GPRS network. GTP is defined on both the Gn and Gp interfaces of a GPRS network.

GUP Gatekeeper Update Protocol.

H

H.225 A protocol used for TCP signalling in applications such as video conferencing. See also H.323 and inspection engine.

H.225.0 An ITU standard that governs H.225.0 session establishment and packetization. H.225.0 actually describes several different protocols: RAS, use of Q.931, and use of RTP.

H.245 An ITU standard that governs H.245 endpoint control.

H.320 Suite of ITU-T standard specifications for video conferencing over circuit-switched media, such as ISDN, fractional T-1, and switched-56 lines. Extensions of ITU-T standard H.320 enable video conferencing over LANs and other packet-switched networks, as well as video over the Internet.

H.323 Allows dissimilar communication devices to communicate with each other by using a standardized communication protocol. H.323 defines a common set of CODECs, call setup and negotiating procedures, and basic data transport methods.

H.323 RAS Registration, admission, and status signaling protocol. Enables devices to perform registration, admissions, bandwidth changes, and status and disengage procedures between VoIP gateway and the gatekeeper.

H.450.2 Call transfer supplementary service for H.323.

H.450.3 Call diversion supplementary service for H.323.

Hash, Hash Algorithm A hash algorithm is a one way function that operates on a message of arbitrary length to create a fixed-length message digest used by cryptographic services to ensure its data integrity. MD5 has a smaller digest and is considered to be slightly faster than SHA-1. Cisco uses both SHA-1 and MD5 hashes within our implementation of the IPSec framework. See also encryption, HMAC, and VPN.

headend A firewall, concentrator, or other host that serves as the entry point into a private network for VPN client connections over the public network. See also ISP and VPN.

HMAC A mechanism for message authentication using cryptographic hashes such as SHA-1 and MD5.

host The name for any device on a TCP/IP network that has an IP address. See also network and node.

host/network An IP address and netmask used with other information to identify a single host or network subnet for FWSM configuration, such as an address translation (xlate) or ACE.

HSRP Hot Standby Routing Protocol. A Cisco-proprietary protocol, HSRP is a routing protocol that provides backup to a router in the event of failure.

HTTP Hypertext Transfer Protocol. A protocol used by browsers and web servers to transfer files. When a user views a web page, the browser can use HTTP to request and receive the files used by the web page. HTTP transmissions are not encrypted.

HTTPS HTTP over SSL. An SSL-encrypted version of HTTP.

I

IANA Internet Assigned Number Authority. Assigns all port and protocol numbers for use on the Internet.

ICMP Internet Control Message Protocol. Network-layer Internet protocol that reports errors and provides other information relevant to IP packet processing.

IETF The Internet Engineering Task Force. A technical standards organization that develops RFC documents defining protocols for the Internet.

IGMP Internet Group Management Protocol. IGMP is a protocol used by IPv4 systems to report IP multicast memberships to neighboring multicast routers.

IKE Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPSec) that require keys. Before any IPSec traffic can be passed, each FWSM must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service. IKE is a hybrid protocol that uses part Oakley and part of another protocol suite called SKEME inside ISAKMP framework. This is the protocol formerly known as ISAKMP/Oakley, and is defined in RFC 2409.

IKE Extended Authentication IKE Extended Authenticate (Xauth) is implemented per the IETF draft-ietf-ipsec-isakmp-xauth-04.txt ("extended authentication" draft). This protocol provides the capability of authenticating a user within IKE using TACACS+ or RADIUS.

IKE Mode Configuration IKE Mode Configuration is implemented per the IETF draft-ietf-ipsec-isakmp-mode-cfg-04.txt. IKE Mode Configuration provides a method for a security gateway to download an IP address (and other network level configuration) to the VPN client as part of an IKE negotiation.

ILS Internet Locator Service. ILS is based on LDAP and is ILSv2 compliant. ILS was developed by Microsoft for use with its NetMeeting, SiteServer, and Active Directory products.

IMAP Internet Message Access Protocol. Method of accessing e-mail or bulletin board messages kept on a mail server that can be shared. IMAP permits client e-mail applications to access remote message stores as if they were local without actually transferring the message.

implicit rule An access rule automatically created by the FWSM based on default rules or as a result of user-defined rules.

IMSI International Mobile Subscriber Identity. One of two components of a GTP tunnel ID, the other being the NSAPI. See also NSAPI.

inside The first interface, usually port 1, that connects your internal, "trusted" network protected by the FWSM. See also interface, interface names.

inspection engine The FWSM inspects certain application-level protocols to identify the location of embedded addressing information in traffic. This allows NAT to translate these embedded addresses and to update any checksum or other fields that are affected by the translation. Because many protocols open secondary TCP or UDP ports, each application inspection engine also monitors sessions to determine the port numbers for secondary channels. The initial session on a well-known port is used to negotiate dynamically assigned port numbers. The application inspection engine monitors these sessions, identifies the dynamic port assignments, and permits data exchange on these ports for the duration of the specific session. Some of the protocols that the FWSM can inspect are CTIQBE, FTP, H.323, HTTP, MGCP, SMTP, and SNMP.

interface The physical connection between a particular network and a FWSM.

interface ip_address The IP address of a FWSM network interface. Each interface IP address must be unique. Two or more interfaces must not be given the same IP address or IP addresses that are on the same IP network.

interface names Human readable name assigned to a FWSM network interface. The inside interface default name is "inside" and the outside interface default name is "outside." Any perimeter interface default names are "intf*n*", such as intf2 for the first perimeter interface, intf3 for the second perimeter interface, and so on to the last interface. The numbers in the intf string corresponds to the position of the interface card in the FWSM. You can use the default names or, if you are an experienced user, give each interface a more meaningful name. See also inside, intfn, outside.

intf*n* Any interface, usually beginning with port 2, that connects to a subset network of your design that you can custom name and configure.

interface PAT The use of PAT where the PAT IP address is also the IP address of the outside interface. See Dynamic PAT, Static PAT.

Internet The global network that uses IP. Not a LAN. See also intranet.

intranet Intranetwork. A LAN that uses IP. See also network and Internet.

IP Internet Protocol. IP protocols are the most popular nonproprietary protocols because they can be used to communicate across any set of interconnected networks and are equally well suited for LAN and WAN communications.

IPS Intrusion Prevention System. An in-line, deep-packet inspection-based solution that helps mitigate a wide range of network attacks.

IP address An IP protocol address. A FWSM interface ip_address. IP version 4 addresses are 32 bits in length. This address space is used to designate the network number, optional subnetwork number, and a host number. The 32 bits are grouped into four octets (8 binary bits), represented by 4 decimal numbers separated by periods, or dots. The meaning of each of the four octets is determined by their use in a particular network.

IP pool A range of local IP addresses specified by a name, and a range with a starting IP address and an ending address. IP Pools are used by DHCP and VPNs to assign local IP addresses to clients on the inside interface.

IPSec IP Security. A framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer. IPSec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPSec. IPSec can protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

IPSec Phase 1 The first phase of negotiating IPSec, includes the key exchange and the ISAKMP portions of IPSec.

IPSec Phase 2 The second phase of negotiating IPSec. Phase two determines the type of encryption rules used for payload, the source and destination that will be used for encryption, the definition of interesting traffic according to access lists, and the IPSec peer. IPSec is applied to the interface in Phase 2.

IPSec transform set A transform set specifies the IPSec protocol, encryption algorithm, and hash algorithm to use on traffic matching the IPSec policy. A transform describes a security protocol (AH or ESP) with its corresponding algorithms. The IPSec protocol used in almost all transform sets is ESP with the DES algorithm and HMAC-SHA for authentication.

ISAKMP Internet Security Association and Key Management Protocol. A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association. See IKE.

ISP Internet service provider. An organization that provides connection to the Internet via their services, such as modem dial in over telephone voice lines or DSL.

J

JTAPI Java Telephony Application Programming Interface. A Java-based API supporting telephony functions. See also TAPI.

K

key A data object used for encryption, decryption, or authentication.

Kerberos A strong network authentication protocol for client-server applications that uses secret-key cryptography. Kerberos is one of the SASL mechanisms available for security appliance authentication to an LDAP server.

L

LAN Local area network. A network residing in one location, such as a single building or campus. See also Internet, intranet, and network.

layer, layers Networking models implement layers with which different protocols are associated. The most common networking model is the OSI model, which consists of the following 7 layers, in order: physical, data link, network, transport, session, presentation, and application.

LCN Logical channel number.

LDAP Lightweight Directory Access Protocol. LDAP provides management and browser applications with access to X.500 directories.

LDP Label Distribution Protocol.

LLA link-local address.

M

mask A 32-bit mask that shows how an Internet address is divided into network, subnet, and host parts. The mask has ones in the bit positions to be used for the network and subnet parts, and zeros for the host part. The mask should contain at least the standard network portion, and the subnet field should be contiguous with the network portion.

MCR See multicast.

MC router Multicast (MC) routers route multicast data transmissions to the hosts on each LAN in an internetwork that are registered to receive specific multimedia or other broadcasts. See also multicast.

MD5 Message Digest 5. A one-way hashing algorithm that produces a 128-bit hash. Both MD5 and SHA-1 are variations on MD4 and are designed to strengthen the security of the MD4 hashing algorithm. SHA-1 is more secure than MD4 and MD5. Cisco uses hashes for authentication within the IPSec framework. Also used for message authentication in SNMP v.2. MD5 verifies the integrity of the communication, authenticates the origin, and checks for timeliness. MD5 has a smaller digest and is considered to be slightly faster than SHA-1.

MDI Media dependent interface.

MDIX Media dependent interface crossover.

Message Digest A message digest is created by a hash algorithm, such as MD5 or SHA-1, that is used for ensuring message integrity.

MSFC Multilayer Switch Feature Card. The Multilayer Switch Feature Card is a router card installed in the Catalyst 6500 switch or Cisco 7600 router.

MGCP Media Gateway Control Protocol. Media Gateway Control Protocol is a protocol for the control of VoIP calls by external call-control elements known as media gateway controllers or call agents. MGCP merges the IPDC and SGCP protocols.

Mode See Access Modes.

Mode Config See IKE Mode Configuration.

Modular Policy Framework Modular Policy Framework. A means of configuring FWSM features in a manner to similar to Cisco IOS software Modular QoS CLI.

MS mobile station. Refers generically to any mobile device, such as a mobile handset or computer, that is used to access network services. GPRS networks support three classes of MS, which describe the type of operation supported within the GPRS and the GSM mobile wireless networks. For example, a Class A MS supports simultaneous operation of GPRS and GSM services.

MS-CHAP Microsoft CHAP.

MTU Maximum transmission unit, the maximum number of bytes in a packet that can flow efficiently across the network with best response time. For Ethernet, the default MTU is 1500 bytes, but each network can have different values, with serial connections having the smallest values. The MTU is described in RFC 1191.

multicast Multicast refers to a network addressing method in which the source transmits a packet to multiple destinations, a multicast group, simultaneously. See also PIM, SMR.

N

N2H2 A third-party, policy-oriented filtering application that works with the FWSM to control user web access. N2H2 can filter HTTP requests based on destination host name, destination IP address, and username and password. The N2H2 corporation was acquired by Secure Computing in October, 2003.

NAT Network Address Translation. Mechanism for reducing the need for globally unique IP addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet by translating those addresses into a globally routable address space.

NBAR Network-Based Application Recognition.

NEM Network Extension Mode. Lets VPN hardware clients present a single, routable network to the remote private network over the VPN tunnel.

NetBIOS Network Basic Input/Output System. A Microsoft protocol that supports Windows host name registration, session management, and data transfer. The FWSM supports NetBIOS by performing NAT of the packets for NBNS UDP port 137 and NBDS UDP port 138.

netmask See mask.

network In the context of FWSM configuration, a network is a group of computing devices that share part of an IP address space and not a single host. A network consists of multiple nodes or hosts. See also host, Internet, intranet, IP, LAN, and node.

NMS network management system. System responsible for managing at least part of a network. An NMS is generally a reasonably powerful and well-equipped computer, such as an engineering workstation. NMSs communicate with agents to help keep track of network statistics and resources.

node Devices such as routers and printers that would not normally be called hosts. See also host, network.

nonvolatile storage, memory Storage or memory that, unlike RAM, retains its contents without power. Data in a nonvolatile storage device survives a power-off, power-on cycle or reboot.

NP Network processor.

NSAPI Network service access point identifier. One of two components of a GTP tunnel ID, the other component being the IMSI. See also IMSI.

NSSA Not-so-stubby-area. An OSPF feature described by RFC 1587. NSSA was first introduced in Cisco IOS software release 11.2. It is a non-proprietary extension of the existing stub area feature that allows the injection of external routes in a limited fashion into the stub area.

NTLM NT Lan Manager. A Microsoft Windows challenge-response authentication method.

NTP Network time protocol.

O

Oakley A key exchange protocol that defines how to acquire authenticated keying material. The basic mechanism for Oakley is the Diffie-Hellman key exchange algorithm. Oakley is defined in RFC 2412.

object grouping Simplifies access control by letting you apply access control statements to groups of network objects, such as protocol, services, hosts, and networks.

OSPF Open Shortest Path First. OSPF is a routing protocol for IP networks. OSPF is a routing protocol widely deployed in large networks because of its efficient use of network bandwidth and its rapid convergence after changes in topology. The FWSM supports OSPF.

OU Organizational Unit. An X.500 directory attribute.

outbound Refers to traffic whose destination is on an interface with lower security than the source interface.

outbound ACL An ACL applied to outbound traffic.

outside The first interface, usually port 0, that connects to other "untrusted" networks outside the FWSM; the Internet. See also interface, interface names, outbound.

P

PAC PPTP Access Concentrator. A device attached to one or more PSTN or ISDN lines capable of PPP operation and of handling the PPTP protocol. The PAC need only implement TCP/IP to pass traffic to one or more PNSs. It may also tunnel non-IP protocols.

PAT See Dynamic PAT, interface PAT, and Static PAT.

Perfmon The FWSM feature that gathers and reports a wide variety of feature statistics, such as connections/second, xlates/second, etc.

PFS perfect forward secrecy. PFS enhances security by using different security key for the IPSec Phase 1 and Phase 2 SAs. Without PFS, the same security key is used to establish SAs in both phases. PFS ensures that a given IPSec SA key was not derived from any other secret (like some other keys). In other words, if someone were to break a key, PFS ensures that the attacker would not be able to derive any other key. If PFS were not enabled, someone could hypothetically break the IKE SA secret key, copy all the IPSec protected data, and then use knowledge of the IKE SA secret to compromise the IPSec SA setup by this IKE SA. With PFS, breaking IKE would not give an attacker immediate access to IPSec. The attacker would have to break each IPSec SA individually.

Phase 1 See IPSec Phase 1.

Phase 2 See IPSec Phase 2.

PIM Protocol Independent Multicast. PIM provides a scalable method for determining the best paths for distributing a specific multicast transmission to a group of hosts. Each host has registered using IGMP to receive the transmission. See also PIM-SM.

PIM-SM Protocol Independent Multicast-Sparse Mode. With PIM-SM, which is the default for Cisco routers, when the source of a multicast transmission begins broadcasting, the traffic is forwarded from one MC router to the next, until the packets reach every registered host. See also PIM.

Ping An ICMP request sent by a host to determine if a second host is accessible.

PIX Private Internet eXchange. The Cisco PIX 500-series FWSMs range from compact, plug-and-play desktop models for small/home offices to carrier-class gigabit models for the most demanding enterprise and service provider environments. Cisco PIX FWSMs provide robust, enterprise-class integrated network security services to create a strong multilayered defense for fast changing network environments.

PKCS12 A standard for the transfer of PKI-related data, such as private keys, certificates, and other data. Devices supporting this standard let administrators maintain a single set of personal identity information.

PNS PPTP Network Server. A PNS is envisioned to operate on general-purpose computing/server platforms. The PNS handles the server side of PPTP. Because PPTP relies completely on TCP/IP and is independent of the interface hardware, the PNS may use any combination of IP interface hardware including LAN and WAN devices.

Policy NAT Lets you identify local traffic for address translation by specifying the source and destination addresses (or ports) in an access list.

POP Post Office Protocol. Protocol that client e-mail applications use to retrieve mail from a mail server.

Pool See IP pool.

Port A field in the packet headers of TCP and UDP protocols that identifies the higher level service which is the source or destination of the packet.

PPP Point-to-Point Protocol. Developed for dial-up ISP access using analog phone lines and modems.

PPTP Point-to-Point Tunneling Protocol. PPTP was introduced by Microsoft to provide secure remote access to Windows networks; however, because it is vulnerable to attack, PPTP is commonly used only when stronger security methods are not available or are not required. PPTP Ports are pptp, 1723/tcp, 1723/udp, and pptp. For more information about PPTP, see RFC 2637. See also PAC, PPTP GRE, PPTP GRE tunnel, PNS, PPTP session, and PPTP TCP.

PPTP GRE Version 1 of GRE for encapsulating PPP traffic.

PPTP GRE tunnel A tunnel defined by a PNS-PAC pair. The tunnel protocol is defined by a modified version of GRE. The tunnel carries PPP datagrams between the PAC and the PNS. Many sessions are multiplexed on a single tunnel. A control connection operating over TCP controls the establishment, release, and maintenance of sessions and of the tunnel itself.

PPTP session PPTP is connection-oriented. The PNS and PAC maintain state for each user that is attached to a PAC. A session is created when end-to-end PPP connection is attempted between a dial user and the PNS. The datagrams related to a session are sent over the tunnel between the PAC and PNS.

PPTP TCP Standard TCP session over which PPTP call control and management information is passed. The control session is logically associated with, but separate from, the sessions being tunneled through a PPTP tunnel.

preshared key A preshared key provides a method of IKE authentication that is suitable for networks with a limited, static number of IPSec peers. This method is limited in scalability because the key must be configured for each pair of IPSec peers. When a new IPSec peer is added to the network, the preshared key must be configured for every IPSec peer with which it communicates. Using certificates and CAs provides a more scalable method of IKE authentication.

primary, primary unit The FWSM normally operating when two units, a primary and secondary, are operating in failover mode.

privileged EXEC mode Privileged EXEC mode lets you to change current settings. Any user EXEC mode command will work in privileged EXEC mode. See also command-specific configuration mode, global configuration mode, user EXEC mode.

protocol, protocol literals A standard that defines the exchange of packets between network nodes for communication. Protocols work together in layers. Protocols are specified in a FWSM configuration as part of defining a security policy by their literal values or port numbers. Possible FWSM protocol literal values are ahp, eigrp, esp, gre, icmp, igmp, igrp, ip, ipinip, ipsec, nos, ospf, pcp, snp, tcp, and udp.

Proxy-ARP Enables the FWSM to reply to an ARP request for IP addresses in the global pool. See also ARP.

public key A public key is one of a pair of keys that are generated by devices involved in public key infrastructure. Data encrypted with a public key can only be decrypted using the associated private key. When a private key is used to produce a digital signature, the receiver can use the public key of the sender to verify that the message was signed by the sender. These characteristics of key pairs provide a scalable and secure method of authentication over an insecure media, such as the Internet.

Q

QoS quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

R

RA Registration Authority. An authorized proxy for a CA. RAs can perform certificate enrollment and can issue CRLs. See also CA, certificate, public key.

RADIUS Remote Authentication Dial-In User Service. RADIUS is a distributed client/server system that secures networks against unauthorized access. RFC 2058 and RFC 2059 define the RADIUS protocol standard. See also AAA and TACACS+.

Refresh Retrieve the running configuration from the FWSM and update the screen. The icon and the button perform the same function.

registration authority See RA.

replay-detection A security service where the receiver can reject old or duplicate packets to defeat replay attacks. Replay attacks rely on the attacker sending out older or duplicate packets to the receiver and the receiver thinking that the bogus traffic is legitimate. Replay-detection is done by using sequence numbers combined with authentication, and is a standard feature of IPSec.

RFC Request for Comments. RFC documents define protocols and standards for communications over the Internet. RFCs are developed and published by IETF.

RIP Routing Information Protocol. Interior gateway protocol (IGP) supplied with UNIX BSD systems. The most common IGP in the Internet. RIP uses hop count as a routing metric.

RLLA Reserved Link Local Address. Multicast addresses range from 224.0.0.0 to 239.255.255.255, however only the range 224.0.1.0 to 239.255.255.255 is available to us. The first part of the multicast address range, 224.0.0.0 to 224.0.0.255, is reserved and referred to as the RLLA. These addresses are unavailable. We can exclude the RLLA range by specifying: 224.0.1.0 to 239.255.255.255. 224.0.0.0 to 239.255.255.255 excluding 224.0.0.0 to 224.0.0.255. This is the same as specifying: 224.0.1.0 to 239.255.255.255.

route, routing The path through a network.

routed firewall mode In routed firewall mode, the FWSM is counted as a router hop in the network. It performs NAT between connected networks and can use OSPF or RIP. See also transparent firewall mode.

RP Rendezvous Point. An RP acts as the meeting place for sources and receivers of multicast data in a PIM multicast environment.

RPC Remote Procedure Call. RPCs are procedure calls that are built or specified by clients and executed on servers, with the results returned over the network to the clients.

RSA A public key cryptographic algorithm (named after its inventors, Rivest, Shamir, and Adelman) with a variable key length. The main weakness of RSA is that it is significantly slow to compute compared to popular secret-key algorithms, such as DES. The Cisco implementation of IKE uses a Diffie-Hellman exchange to get the secret keys. This exchange can be authenticated with RSA (or preshared keys). With the Diffie-Hellman exchange, the DES key never crosses the network (not even in encrypted form), which is not the case with the RSA encrypt and sign technique. RSA is not public domain, and must be licensed from RSA Data Security.

RSH Remote Shell. A protocol that allows a user to execute commands on a remote system without having to log in to the system. For example, RSH can be used to remotely examine the status of a number of access servers without connecting to each communication server, executing the command, and then disconnecting from the communication server.

RTCP RTP Control Protocol. Protocol that monitors the QoS of an IPv6 RTP connection and conveys information about the on-going session. See also RTP.

RTP Real-Time Transport Protocol. Commonly used with IP networks. RTP is designed to provide end-to-end network transport functions for applications transmitting real-time data, such as audio, video, or simulation data, over multicast or unicast network services. RTP provides such services as payload type identification, sequence numbering, timestamping, and delivery monitoring to real-time applications.

RTSP Real Time Streaming Protocol. Enables the controlled delivery of real-time data, such as audio and video. RTSP is designed to work with established protocols, such as RTP and HTTP.

rule Conditional statements added to the FWSM configuration to define security policy for a particular situation. See also ACE, ACL, NAT.

running configuration The configuration currently running in RAM on the FWSM. The configuration that determines the operational characteristics of the FWSM.

S

SA security association. An instance of security policy and keying material applied to a data flow. SAs are established in pairs by IPSec peers during both phases of IPSec. SAs specify the encryption algorithms and other security parameters used to create a secure tunnel. Phase 1 SAs (IKE SAs) establish a secure tunnel for negotiating Phase 2 SAs. Phase 2 SAs (IPSec SAs) establish the secure tunnel used for sending user data. Both IKE and IPSec use SAs, although SAs are independent of one another. IPSec SAs are unidirectional and they are unique in each security protocol. A set of SAs are needed for a protected data pipe, one per direction per protocol. For example, if you have a pipe that supports ESP between peers, one ESP SA is required for each direction. SAs are uniquely identified by destination (IPSec endpoint) address, security protocol (AH or ESP), and Security Parameter Index. IKE negotiates and establishes SAs on behalf of IPSec. A user can also establish IPSec SAs manually. An IKE SA is used by IKE only, and unlike the IPSec SA, it is bidirectional.

SASL Simple Authentication and Security Layer. An Internet standard method for adding authentication support to connection-based protocols. SASL can be used between a security appliance and an LDAP server to secure user authentication.

SCCP Skinny Client Control Protocol. A Cisco-proprietary protocol used between Cisco Call Manager and Cisco VoIP phones.

SCEP Simple Certificate Enrollment Protocol. A method of requesting and receiving (also known as enrolling) certificates from CAs.

SDP Session Definition Protocol. An IETF protocol for the definition of Multimedia Services. SDP messages can be part of SGCP and MGCP messages.

secondary unit The backup FWSM when two are operating in failover mode.

secret key A secret key is a key shared only between the sender and receiver. See key, public key.

security context You can partition a single FWSM into multiple virtual firewalls, known as security contexts. Each context is an independent firewall, with its own security policy, interfaces, and administrators. Multiple contexts are similar to having multiple stand-alone firewalls.

security services See cryptography.

serial transmission A method of data transmission in which the bits of a data character are transmitted sequentially over a single channel.

SGCP Simple Gateway Control Protocol. Controls VoIP gateways by an external call control element (called a call-agent).

SGSN Serving GPRS Support Node. The SGSN ensures mobility management, session management and packet relaying functions.

SHA-1 Secure Hash Algorithm 1. SHA-1 [NIS94c] is a revision to SHA that was published in 1994. SHA is closely modeled after MD4 and produces a 160-bit digest. Because SHA produces a 160-bit digest, it is more resistant to brute-force attacks than 128-bit hashes (such as MD5), but it is slower. Secure Hash Algorithm 1 is a joint creation of the National Institute of Standards and Technology and the National Security Agency. This algorithm, like other hash algorithms, is used to generate a hash value, also known as a message digest, that acts like a CRC used in lower-layer protocols to ensure that message contents are not changed during transmission. SHA-1 is generally considered more secure than MD5.

SIP Session Initiation Protocol. Enables call handling sessions, particularly two-party audio conferences, or "calls." SIP works with SDP for call signaling. SDP specifies the ports for the media stream. Using SIP, the FWSM can support any SIP VoIP gateways and VoIP proxy servers.

site-to-site VPN A site-to-site VPN is established between two IPSec peers that connect remote networks into a single VPN. In this type of VPN, neither IPSec peer is the destination or source of user traffic. Instead, each IPSec peer provides encryption and authentication services for hosts on the LANs connected to each IPSec peer. The hosts on each LAN send and receive data through the secure tunnel established by the pair of IPSec peers.

SKEME A key exchange protocol that defines how to derive authenticated keying material, with rapid key refreshment.

SMR Stub Multicast Routing. SMR allows the FWSM to function as a "stub router." A stub router is a device that acts as an IGMP proxy agent. IGMP is used to dynamically register specific hosts in a multicast group on a particular LAN with a multicast router. Multicast routers route multicast data transmissions to hosts that are registered to receive specific multimedia or other broadcasts. A stub router forwards IGMP messages between hosts and MC routers.

SMTP Simple Mail Transfer Protocol. SMTP is an Internet protocol that supports email services.

SNMP Simple Network Management Protocol. A standard method for managing network devices using data structures called Management Information Bases.

split tunneling Allows a remote VPN client simultaneous encrypted access to a private network and clear unencrypted access to the Internet. If you do not enable split tunneling, all traffic between the VPN client and the FWSM is sent through an IPSec tunnel. All traffic originating from the VPN client is sent to the outside interface through a tunnel, and client access to the Internet from its remote site is denied.

spoofing A type of attack designed to foil network security mechanisms such as filters and access lists. A spoofing attack sends a packet that claims to be from an address from which it was not actually sent.

SQL*Net Structured Query Language Protocol. An Oracle protocol used to communicate between client and server processes.

SSH Secure Shell. An application running on top of a reliable transport layer, such as TCP/IP, that provides strong authentication and encryption capabilities.

SSL Secure Sockets Layer. A protocol that resides between the application layer and TCP/IP to provide transparent encryption of data traffic.

standby unit See secondary unit.

stateful inspection Network protocols maintain certain data, called state information, at each end of a network connection between two hosts. State information is necessary to implement the features of a protocol, such as guaranteed packet delivery, data sequencing, flow control, and transaction or session IDs. Some of the protocol state information is sent in each packet while each protocol is being used. For example, a browser connected to a web server uses HTTP and supporting TCP/IP protocols. Each protocol layer maintains state information in the packets it sends and receives. The FWSM and some other firewalls inspect the state information in each packet to verify that it is current and valid for every protocol it contains. This is called stateful inspection and is designed to create a powerful barrier to certain types of computer security threats.

Static PAT Static Port Address Translation. Static PAT is a static address that also maps a local port to a global port. See also Dynamic PAT, NAT.

subnetmask See mask.

SVC The SSL VPN Client (SVC) is a VPN tunneling technology that gives remote users the benefits of an IPSec VPN client without the need for network administrators to install and configure IPSec VPN clients on remote computers. The SVC uses the SSL encryption that is already present on the remote computer and the WebVPN login and authentication of the security appliance.

SVI Switched virtual interface. An SVI is a VLAN assigned to the MSFC.

T

TACACS+ Terminal Access Controller Access Control System Plus. A client-server protocol that supports AAA services, including command authorization. See also AAA, RADIUS.

TAPI Telephony Application Programming Interface. A programming interface in Microsoft Windows that supports telephony functions.

TCP Transmission Control Protocol. Connection-oriented transport layer protocol that provides reliable full-duplex data transmission.

TCP Intercept With the TCP intercept feature, once the optional embryonic connection limit is reached, and until the embryonic connection count falls below this threshold, every SYN bound for the effected server is intercepted. For each SYN, the FWSM responds on behalf of the server with an empty SYN/ACK segment. The FWSM retains pertinent state information, drops the packet, and waits for the client acknowledgment. If the ACK is received, then a copy of the client SYN segment is sent to the server and the TCP three-way handshake is performed between the FWSM and the server. If this three-way handshake completes, may the connection resume as normal. If the client does not respond during any part of the connection phase, then the FWSM retransmits the necessary segment using exponential back-offs.

TDP Tag Distribution Protocol. TDP is used by tag switching devices to distribute, request, and release tag binding information for multiple network layer protocols in a tag switching network. TDP does not replace routing protocols. Instead, it uses information learned from routing protocols to create tag bindings. TDP is also used to open, monitor, and close TDP sessions and to indicate errors that occur during those sessions. TDP operates over a connection-oriented transport layer protocol with guaranteed sequential delivery (such as TCP). The use of TDP does not preclude the use of other mechanisms to distribute tag binding information, such as piggybacking information on other protocols.

Telnet A terminal emulation protocol for TCP/IP networks such as the Internet. Telnet is a common way to control web servers remotely; however, its security vulnerabilities have led to its replacement by SSH.

TFTP Trivial File Transfer Protocol. TFTP is a simple protocol used to transfer files. It runs on UDP and is explained in depth in RFC 1350.

TLS Transport Layer Security. A future IETF protocol to replace SSL.

traffic policing The traffic policing feature ensures that no traffic exceeds the maximum rate (bits per second) that you configure, thus ensuring that no one traffic flow can take over the entire resource.

transform set See IPSec transform set.

translate, translation See xlate.

transparent firewall mode A mode in which the FWSM is not a router hop. You can use transparent firewall mode to simplify your network configuration or to make the FWSM invisible to attackers. You can also use transparent firewall mode to allow traffic through that would otherwise be blocked in routed firewall mode. See also routed firewall mode.

transport mode An IPSec encryption mode that encrypts only the data portion (payload) of each packet, but leaves the header untouched. Transport mode is less secure than tunnel mode.

TSP TAPI Service Provider. See also TAPI.

tunnel mode An IPSec encryption mode that encrypts both the header and data portion (payload) of each packet. Tunnel mode is more secure than transport mode.

tunnel A method of transporting data in one protocol by encapsulating it in another protocol. Tunneling is used for reasons of incompatibility, implementation simplification, or security. For example, a tunnel lets a remote VPN client have encrypted access to a private network.

Turbo ACL Increases ACL lookup speeds by compiling them into a set of lookup tables. Packet headers are used to access the tables in a small, fixed number of lookups, independent of the existing number of ACL entries.

U

UDP User Datagram Protocol. A connectionless transport layer protocol in the IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, which requires other protocols to handle error processing and retransmission. UDP is defined in RFC 768.

UMTS Universal Mobile Telecommunication System. An extension of GPRS networks that moves toward an all-IP network by delivering broadband information, including commerce and entertainment services, to mobile users via fixed, wireless, and satellite networks

Unicast RPF Unicast Reverse Path Forwarding. Unicast RPF guards against spoofing by ensuring that packets have a source IP address that matches the correct source interface according to the routing table.

URL Uniform Resource Locator. A standardized addressing scheme for accessing hypertext documents and other services using a browser. For example, http://www.cisco.com.

user EXEC mode User EXEC mode lets you to see the FWSM settings. The user EXEC mode prompt appears as follows when you first access the FWSM. See also command-specific configuration mode, global configuration mode, and privileged EXEC mode.

UTC Coordinated Universal Time. The time zone at zero degrees longitude, previously called Greenwich Mean Time (GMT) and Zulu time. UTC replaced GMT in 1967 as the world time standard. UTC is based on an atomic time scale rather than an astronomical time scale.

UTRAN Universal Terrestrial Radio Access Network. Networking protocol used for implementing wireless networks in UMTS. GTP allows multi-protocol packets to be tunneled through a UMTS/GPRS backbone between a GGSN, an SGSN and the UTRAN.

UUIE User-User Information Element. An element of an H.225 packet that identifies the users implicated in the message.

V

VLAN Virtual LAN. A group of devices on one or more LANs that are configured (using management software) so that they can communicate as if they were attached to the same physical network cable, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, they are extremely flexible.

VoIP Voice over IP. VoIP carries normal voice traffic, such as telephone calls and faxes, over an IP-based network. DSP segments the voice signal into frames, which then are coupled in groups of two and stored in voice packets. These voice packets are transported using IP in compliance with ITU-T specification H.323.

VPN Virtual Private Network. A network connection between two peers over the public network that is made private by strict authentication of users and the encryption of all data traffic. You can establish VPNs between clients, such as PCs, or a headend, such as the FWSM.

virtual firewall See security context.

VRRP Virtual Router Redundancy Protocol. VRRP is an election protocol that dynamically assigns responsibility for one or more virtual router(s) to the VRRP router(s) on a LAN, allowing several routers on a multiaccess link to utilize the same virtual IP address.

VSA Vendor-specific attribute. An attribute in a RADIUS packet that is defined by a vendor rather than by RADIUS RFCs. The RADIUS protocol uses IANA-assigned vendor numbers to help identify VSAs. This lets different vendors have VSAs of the same number. The combination of a vendor number and a VSA number makes a VSA unique. For example, the cisco-av-pair VSA is attribute 1 in the set of VSAs related to vendor number 9. Each vendor can define up to 256 VSAs. A RADIUS packet contains any VSAs attribute 26, named Vendor-specific. VSAs are sometimes referred to as subattributes.

W

WAN wide-area network. Data communications network that serves users across a broad geographic area and often uses transmission devices provided by common carriers.

Websense A content filtering solution that manages employee access to the Internet. Websense uses a policy engine and a URL database to control user access to websites.

WEP Wired Equivalent Privacy. A security protocol for wireless LANs, defined in the IEEE 802.11b standard.

WINS Windows Internet Naming Service. A Windows system that determines the IP address associated with a particular network device, also known as "name resolution." WINS uses a distributed database that is automatically updated with the NetBIOS names of network devices currently available and the IP address assigned to each one.WINS provides a distributed database for registering and querying dynamic NetBIOS names to IP address mapping in a routed network environment. It is the best choice for NetBIOS name resolution in such a routed network because it is designed to solve the problems that occur with name resolution in complex networks.

X

X.509 A widely used standard for defining digital certificates. X.509 is actually an ITU recommendation, which means that it has not yet been officially defined or approved for standardized usage.

xauth See IKE Extended Authentication.

xlate An xlate, also referred to as a translation entry, represents the mapping of one IP address to another, or the mapping of one IP address/port pair to another.

INDEX

Symbols

A

B

C

D

E

F

G

H

K

L

M

N

O

P

Q

R

S

T

U

V

W

X