Cisco Firewall Video Mentor

David Hucaby, CCIE No. 4594

Cisco Press

800 East 96th Street

Indianapolis, Indiana 46240 USA

Cisco Firewall Video Mentor

David Hucaby, CCIE No. 4594

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America

First Printing May 2008

ISBN-13: 978-1-58720-198-1

ISBN-10: 1-58720-198-4

Warning and Disclaimer

This book and video product are designed to provide information about configuring Cisco firewalls. Every effort has been made to make this book as complete and accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The author, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Publisher
Paul Boger

Associate Publisher
Dave Dusthimer

Cisco Representative
Anthony Wolfenden

Cisco Press Program Manager
Jeff Brady

Executive Editor
Brett Bartow

Managing Editor
Patrick Kanouse

Development Editor
Andrew Cupp

Project Editor
Jennifer Gallant

Copy Editor
Gayle Johnson

Technical Editor
Mark Macumber

Team Coordinator
Vanessa Evans

Book Designer
Louisa Adair

Composition
Mark Shirar

Proofreader
Water Crest Publishing, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Corporate and Government Sales

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
1-800-382-3419
corpsales@pearsontechgroup.com

For sales outside the United States, please contact:

International Sales
international@pearsoned.com

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members of the professional technical community.

Reader feedback is a natural continuation of this process. If you have any comments about how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through e-mail at feedback@ciscopress.com. Please be sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

About the Author

David Hucaby, CCIE No. 4594, is a network architect for the University of Kentucky, where he works with healthcare networks based on the Cisco Catalyst, ASA, FWSM, and VPN product lines. He was one of the beta reviewers of the ASA 8.0 operating system software. He has B.S. and M.S. degrees in electrical engineering from the University of Kentucky. He is the author of several other books from Cisco Press, including *Cisco ASA, PIX, and FWSM Firewall Handbook* and *CCNP BCMSN Official Exam Certification Guide*. He lives in Kentucky with his wife, Marci, and two daughters.

About the Technical Reviewer

Mark Macumber is a systems engineer in the field sales organization for Cisco. He joined Cisco in 1999, working in the Network Service Provider sales division, working on Internet service provider networks and with telco DSL network designs. Since 2002, he has served in the large enterprise customer space, working through customer designs for campus switching, WAN routing, unified communications, wireless, and security. The Enterprise Security SE team learns and delivers content on Cisco security products such as firewalls, host/network-based intrusion prevention/detection systems, AAA, security information management, network admission control, and SSL/IPsec VPNs.

Dedication

To Marci, Lauren, and Kara, who make my life complete as a husband and a daddy.

Acknowledgments

Mark Macumber, technical reviewer and Cisco SE, continues to amaze me with his breadth of knowledge and his ability to juggle many things at once. I am grateful that he found time to review this project and to make it better.

I'm indebted to Wendell Odom (as always) for generously helping me learn the ropes of creating video labs. I could not have started or finished this project without his encouragement and advice.

Jim Owens has been an encouragement to me, and has provided dependable help every time I've needed it.

I am grateful to Cisco Press for letting me try a project that is different from my usual book writing. I've had fun, but I'm sure I've tried the patience of a variety of people—probably the ones listed next!

Drew Cupp took care of the editing process in what turned out to be a long project. His efforts have not gone unappreciated.

Chris Cleveland offered some good advice early on, which was invaluable to a new Video Mentor author.

Eric Strom traveled to Kentucky to record the live video segments. The fact that there are live video segments at all is a testament to his patience and his skill with audio and video.

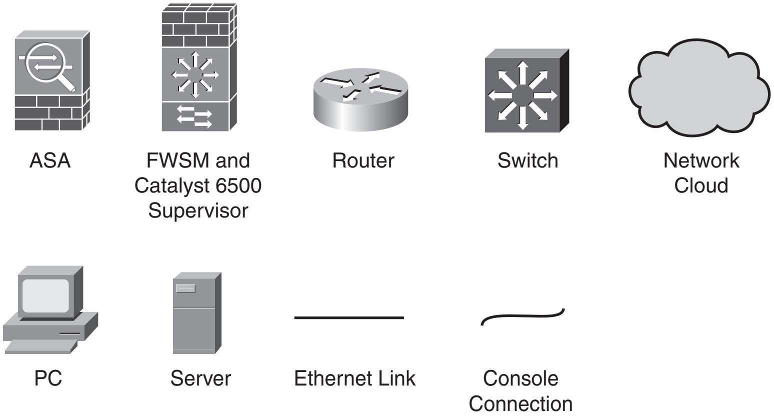
Brett Bartow deserves my thanks for helping organize the project, for suffering through deadlines that I didn't meet, and for patiently (but firmly) keeping me on track to complete the project.

Finally, I am grateful for the support that my family provided. They even took the time to learn some Cisco terminology, just to feel like a part of the project. God is so very good to provide encouragement and endurance in all things—especially in finding a way to use my book projects to spread His own book project to new readers.

Contents at a Glance

Contents

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Bold** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), bold indicates commands that the user enters (such as a **show** command).
- *Italic* indicates arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([{ }]) indicate a required choice within an optional element.

Introduction

The Cisco Firewall Video Mentor supplies 16 instructional videos that cover a variety of firewall configuration tasks. Because firewall features can be complex and tedious to configure, each video presents a scenario that visually demonstrates a feature configuration step by step, along with a running audio commentary.

This product is one of several in the Cisco Press Video Mentor series. The Video Mentor series offers a learning environment that is different from that of printed books, where you can only read about concepts and look at static examples. With the video labs, you can learn about concepts much as you would in a classroom setting, with a live instructor. As well, you can watch configurations and examples unfold, step by step, with explanations along the way.

The Video Mentor covers the firewall features found in the Cisco ASA 5500 family of security appliances, as well as the Cisco Catalyst 6500 Firewall Services Module (FWSM).

Who Should Use the Cisco Firewall Video Mentor?

The Cisco Firewall Video Mentor is intended for people who are involved in firewall installation and administration. Although it is not designed around any specific Cisco course or exam, it can be used to augment self-study books about firewalls and security topics.

Because of the multimedia format, the Video Mentor uses video and audio media to deliver information more effectively than printed material alone—especially for visual learners.

Goals and Methods

The Cisco Firewall Video Mentor shows the author's computer desktop as a firewall is being configured and tested. A running audio commentary accompanies the video so that every activity is explained.

Most of the video labs follow the same format, using these steps as they are appropriate to the lab:

Step 1. The video begins by listing goals or topics for the lab.

Step 2. An overview of specific firewall features is given.

Step 3. A scenario involving a firewall feature is presented, and related command syntax is discussed.

Step 4. A terminal emulator window shows how the firewall feature is configured with the command-line interface, step by step.

Step 5. The configuration is reset, and the same scenario is rebuilt using the Adaptive Security Device Manager (ASDM) management tool.

Cisco Firewall Video Mentor Contents

The Cisco Firewall Video Mentor contains a DVD and a printed booklet. The DVD consists of a series of 16 video labs. The DVD is viewed on a computer screen and is optimized for display in a 1024×768-pixel minimum area.

The booklet contains information that you can use as a reference while watching the video labs. It is not meant to be a standalone tool. The booklet has a section devoted to each of the 16 video labs, containing the figures and configuration information used in the video.

Each booklet section includes the following:

- A list of objectives or topics for the video lab
- A description of the scenario, broken into steps
- The initial configuration entered in the firewall *before* the video lab begins
- The configuration commands that are entered *during* the video lab

The booklet also includes topology figures from the video labs as appropriate.

The booklet is also available in PDF format on the disc. You can switch between displaying the video and the booklet as you work your way through the video labs.

How the Cisco Firewall Video Mentor Is Organized

When the DVD starts, the Cisco Firewall Video Mentor application displays the list of 16 video labs. From the initial menu, you can also view an introductory video that describes the entire product. The video labs are organized as follows:

Lab 1, "Initial Configuration": This lab demonstrates how a new firewall can be configured for the first time. The command-line interface (CLI) is used while the computer is connected to the firewall console.

Lab 2, "Configuring Interfaces": This lab shows how the firewall mode (transparent or routed) is set. Then a variety of firewall interfaces, both physical and logical, are configured.

Lab 3, "Setting Up Routing": In this lab, sources of routing information are configured. Static routes, default routes, standby ISPs, and the OSPF dynamic routing protocol are all demonstrated.

Lab 4, "Firewall Administration over the Network": This lab shows how a firewall can be configured for remote management through Telnet, SSH, and ASDM sessions.

Lab 5, "Using Multiple Security Contexts": This lab demonstrates how a single physical firewall platform can be configured to run multiple instances of virtual firewalls or security contexts.

Lab 6, "Using Failover for High Availability": In this lab, two firewalls are configured as a failover pair. This enables them to operate in a redundant fashion, increasing their availability during a failure.

Lab 7, "Failover in Action": This lab demonstrates several different kinds of failures, triggering the failover operation presented in Lab 6. A "hitless" upgrade is also shown, in which the operating system of each firewall in a failover pair is upgraded without impacting the traffic passing through.

Lab 8, “Setting Up Address Translation and Connection Limits”: This lab shows examples of six different ways to configure address translation on a firewall.

Lab 9, “Setting Up Firewall Rules”: In this lab, security policies are defined through access list configuration. Furthermore, access lists are configured in a more organized, compact fashion with object groups.

Lab 10, “Setting Up a DMZ”: This lab demonstrates how additional interfaces can be added to a firewall, beyond the simple “inside” and “outside” interfaces.

Lab 11, “Setting Up Logging”: In this lab, a firewall is configured to generate and send logging messages to a collection point. After they are collected, the messages can be analyzed, or they can become a record for an audit trail.

Lab 12, “Using MPF to Control Layer 3/4 Connections”: This lab demonstrates how the Modular Policy Framework (MPF) is used to define a policy that sets connection limits on UDP and TCP connections.

Lab 13, “Using MPF to Perform QoS Queuing and Policing”: In this lab, the MPF is used to configure priority queuing policies that handle specific types of traffic more efficiently than other traffic. In addition, policing is used to limit the bandwidth used by certain types of traffic.

Lab 14, “Using MPF to Tune Application Inspection Engines”: This lab shows how a firewall can be configured to change how it inspects traffic related to specific applications.

Lab 15, “Testing Security Policies with Packet Tracer”: This lab demonstrates the Packet Tracer tool and how it can be used to verify a firewall’s configuration. A virtual packet is sent from one interface to another, with a graphical display showing what happens to the packet at each step along the way.

Lab 16, “Capturing Traffic”: In this lab, a firewall is configured to capture traffic for further analysis. Both the CLI and ASDM are used to configure a capture session and to display the packets captured.

Initial Configuration

This Cisco Firewall Video Mentor lab shows you how to interact with a firewall using the command-line interface (CLI) to begin a firewall's initial configuration. Many of the nuances of the CLI are demonstrated.

The objectives of this lab are as follows:

- Connect to the firewall console
- Work with the various CLI modes, and learn how to make configuration changes
- Search through CLI output quickly
- Set the firewall hostname and domain name

Scenario

This lab contains four main steps:

Step 1. Connect to the console of an ASA and to the virtual console of an FWSM to get access to the EXEC mode of the CLI.

Step 2. Explore the EXEC, privileged EXEC, and configuration modes within the CLI. In addition, change the firewall configuration by adding commands to set the EXEC and privileged EXEC mode passwords.

Step 3. Search the running configuration for specific command entries. In addition, learn about various methods of searching through **show** command output.

Step 4. Configure the firewall hostname and domain name.

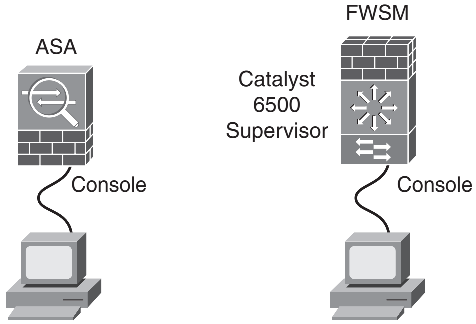
Initial Configurations

The firewalls used in this lab have their default configurations in place, so there are no relevant initial configuration commands.

Video Presentation Reference

Figure 1-1 shows how a PC is connected to the console port of an ASA and then a Catalyst 6500 Supervisor to access an FWSM. A terminal emulator is used on the PC to interact with the firewall's CLI.

Figure 1-1 Lab 1 Topology



Step 1: Connect to the Firewall Console

In this step, a PC is connected directly to an ASA's console port to access the CLI. For the FWSM platform, the PC is connected directly to the Catalyst 6500 Supervisor console port, and then the commands shown in Example 1-1 are entered.

Example 1-1 Accessing the FWSM Console Session

```
Switch# show module
Switch# session slot 3 processor 1
```

Tip

You can obtain PuTTY, a free Telnet and SSH client for Windows platforms, at http://www.putty.org. To connect directly to a firewall console port, you need a terminal emulator that supports asynchronous serial communication. TuTTY, a variation of the PuTTY package that is used in the videos, can be found at http://putty.dwalin.ru. You can also use HyperTerminal Private Edition (http://www.hilgraeve.com/htpe/index.html), a package commonly included as a Windows accessory program.

Step 2: Explore the CLI Modes

In this step, you move from EXEC mode into privileged EXEC mode, and then into configuration mode. The commands to enter privileged EXEC and configuration modes are shown in Examples 1-2 and 1-3, respectively.

Example 1-2 Entering and Exiting Privileged EXEC or "Enable" Mode

```
Firewall> enable
Firewall# exit
```

Example 1-3 Entering and Exiting Configuration Mode

```
Firewall# configure terminal
Firewall(config)# exit
```

Note

You can also press Ctrl-Z to exit configuration mode.

You enter the commands shown in Example 1-4 to configure the EXEC and privileged EXEC passwords, and then to save the running configuration.

Example 1-4 Making a Configuration Change

```
Firewall# configure terminal
Firewall(config)# passwd cisco123
Firewall(config)# enable password enable123
Firewall(config)# exit
Firewall# copy running-config startup-config
```

Note

You can also use the legacy **write memory** command to save the running configuration.

Step 3: Search the CLI Output

In this step, the running configuration is searched for specific commands and their associated keywords. Example 1-5 shows the commands that are entered in this step of the lab.

Example 1-5 Searching the Running Configuration

```
Firewall# show running-config passwd
Firewall# show running-config clock
Firewall# show running-config all clock
```

The search function progresses so that you can search for regular expressions in the text output produced by any **show** command. Example 1-6 lists the commands used to demonstrate output searches in the lab.

Example 1-6 Searching Through show Command Output

```
Firewall# show running-config | begin class-map
Firewall# show running-config | include inspect
Firewall# show interface | include MAC
Firewall# show interface | include (interface|MAC)
```

Table 1-1 lists the syntax of commands that can be used to search output on a firewall.

Table 1-1 Useful Output Search Commands and Their Syntax

Command Syntax	Description
Firewall# **show running-config** *command*	Finds entries for *command* in the running configuration.
Firewall# **show running-config all** *command*	Finds all entries (even defaults) for *command* in the running configuration.
Firewall# **show** *command* | **begin** *regexp*	Begins displaying the output of the **show** command at the first instance of the regular expression.
Firewall# **show** *command* | {**include** | **grep**} *regexp*	Displays only lines from the output of the **show** command that match the regular expression.
Firewall# **show** *command* | {**exclude** | **grep -v**} *regexp*	Displays only lines from the output of the **show** command that *don't* match the regular expression.

Step 4: Set the Firewall Hostname and Domain Name

In this step, the firewall is configured with a hostname and domain name. Example 1-7 shows the commands that are entered.

Example 1-7 Setting the Hostname and Domain Name

```
Firewall(config)# hostname asa1
Firewall(config)# domain-name mycompany.com
```

Configuring Interfaces

This Firewall Video Mentor lab shows you two firewall modes—transparent and routed. In addition, it demonstrates how to configure a variety of firewall interfaces.

The objectives of this lab are as follows:

- Set the firewall mode
- Set interface attributes
- Configure a physical interface
- Configure a redundant interface
- Configure a VLAN interface

Scenario

This lab contains five main steps:

Step 1. Configure the firewall mode according to Layer 2 or Layer 3 operation.

Step 2. Set the interface parameters related to security policies.

Step 3. Configure a physical interface on an ASA platform.

Step 4. Configure a redundant interface on an ASA platform, made up of two other physical member interfaces.

Step 5. Configure logical interfaces on both an ASA and an FWSM platform. On the ASA, a physical interface is configured for trunking, along with its member subinterfaces. The FWSM is configured with logical VLAN interfaces.

Initial Configurations

The firewalls used in this lab have their default configurations in place. There are no relevant initial configuration commands other than the interface names unique to each firewall platform.

Video Presentation Reference

The following sections show the network topology diagrams and commands that are entered in each of the lab steps.

Step 1: Set the Firewall Mode

In this step, the firewall is moved from its default routed mode into transparent mode, and then back into routed mode. Example 2-1 shows the commands that are entered.

Example 2-1 Commands Used to Set and Display the Firewall Mode

```
Firewall# show firewall

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

Step 2: Configure Interface Parameters

This step discusses the commands used to configure three interface parameters as they relate to security policies. Each interface must be configured with a name, a security level, and an IP address, using the commands shown in Example 2-2.

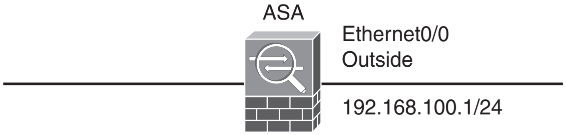
Example 2-2 Commands Used to Configure Interface Parameters

```
Firewall(config)# interface if-name
Firewall(config-if)# nameif name
Firewall(config-if)# security-level level
Firewall(config-if)# ip address ip-address
Firewall(config-if)# exit
```

Step 3: Configure a Physical Interface

In this step, an ASA physical interface is configured according to the network diagram shown in Figure 2-1. The configuration commands used in this step of the lab are shown in Example 2-3.

Figure 2-1 Lab 2 Physical Interface Topology



Example 2-3 Configuring a Physical Interface

```
Firewall(config)# interface ethernet0/0
Firewall(config-if)# nameif outside
Firewall(config-if)# security-level 0
Firewall(config-if)# ip address 192.168.100.1 255.255.255.0
Firewall(config-if)# no shutdown
Firewall(config-if)# exit
```

Table 2-1 lists the syntax of additional commands that can be used to configure a physical interface.

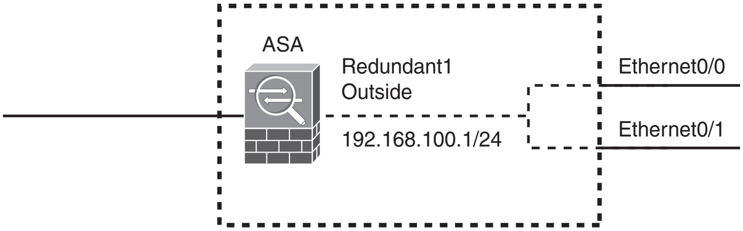
Table 2-1 Useful Interface Configuration Commands

Command Syntax	Description
Firewall(config)# **interface** *if-name*	Identifies a physical interface.
Firewall(config-if)# **speed** {**10** | **100** | **1000** | **auto**}	Sets the interface speed (the default is **auto**).
Firewall(config-if)# **duplex** {**half** | **full** | **auto**}	Sets the interface duplex mode (the default is **auto**).
Firewall(config-if)# **media-type** {**rj45** | **sfp**}	Selects the media type for interfaces that support either RJ-45 copper or SFP modules.
Firewall(config-if)# **mac-address** *mac-address*	Sets the interface MAC address to override the default burned-in address.

Step 4: Configure a Redundant Interface

In this step, a logical redundant interface is configured from two member physical interfaces, as shown in Figure 2-2. Example 2-4 shows the commands that are entered.

Figure 2-2 Lab 2 Redundant Interface Topology



Example 2-4 Configuring a Redundant Interface

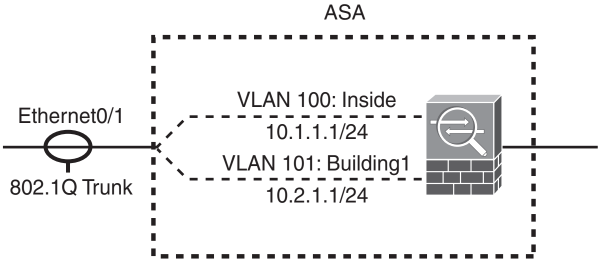
```
Firewall(config)# interface redundant 1
Firewall(config-if)# member-interface ethernet0/0
Firewall(config-if)# member-interface ethernet0/1
Firewall(config-if)# nameif outside
Firewall(config-if)# security-level 0
Firewall(config-if)# ip address 192.168.100.100 255.255.255.0
Firewall(config-if)# no shutdown
Firewall(config-if)# exit
```

Step 5: Configure a Logical VLAN Interface

In this step, two types of logical interfaces are configured. First, a physical interface on an ASA platform is configured as a trunk. Each VLAN carried over the trunk is configured as a logical subinterface.

Figure 2-3 shows a diagram of the ASA topology, and Example 2-5 shows the commands used to configure the logical interfaces.

Figure 2-3 Lab 2 ASA Trunk Interface Topology

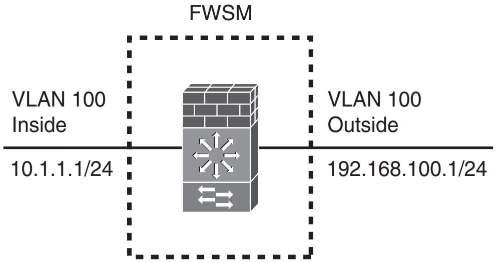


Example 2-5 Configuring an ASA Trunk Interface

```
Firewall(config)# interface ethernet0/1
Firewall(config-if)# no shutdown
Firewall(config-if)# exit
!
Firewall(config)# interface ethernet0/1.1
Firewall(config-if)# vlan 100
Firewall(config-if)# nameif inside
Firewall(config-if)# security-level 100
Firewall(config-if)# ip address 10.1.1.1 255.255.255.0
Firewall(config-if)# no shutdown
Firewall(config-if)# exit
!
Firewall(config)# interface ethernet0/1.2
Firewall(config-if)# vlan 101
Firewall(config-if)# nameif building1
Firewall(config-if)# security-level 50
Firewall(config-if)# ip address 10.2.1.1 255.255.255.0
Firewall(config-if)# no shutdown
Firewall(config-if)# exit
```

Next, logical VLAN interfaces are configured on an FWSM platform. Figure 2-4 shows a diagram of the FWSM topology. Example 2-6 shows the commands used to configure the Catalyst 6500 Supervisor to pass the VLANs to the FWSM. Example 2-7 shows the commands used to configure the FWSM logical interfaces.

Figure 2-4 Lab 2 FWSM VLAN Interface Topology



Example 2-6 Configuring a Catalyst 6500 Supervisor for FWSM VLANs

```
Switch(config)# firewall vlan-group 1 10,100
Switch(config)# firewall module 3 vlan-group 1
```

Example 2-7 Configuring VLAN Interfaces on an FWSM

```
Firewall(config)# interface vlan 10
Firewall(config-if)# nameif outside
Firewall(config-if)# security-level 0
Firewall(config-if)# ip address 192.168.100.1 255.255.255.0
Firewall(config-if)# no shutdown
Firewall(config-if)# exit
!
Firewall(config)# interface vlan 100
Firewall(config-if)# nameif inside
Firewall(config-if)# security-level 100
Firewall(config-if)# ip address 10.1.1.1 255.255.255.0
Firewall(config-if)# no shutdown
Firewall(config-if)# exit
```

Lab 3

Setting Up Routing

This Cisco Firewall Video Mentor lab demonstrates how to configure routing and use a variety of routing information so that a firewall can learn to communicate with surrounding networks.

The objectives of this lab are as follows:

- Use static routes to manually configure routing information
- Leverage multiple paths to the Internet when possible
- Use dynamic routing protocols so that the firewall can learn routing information from other nearby routers

Scenario

In this lab, you will learn how to display the firewall's routing table to verify its knowledge of surrounding networks. Then you will see how to configure a progression of routing information, in the following four steps:

Step 1. Configure static routes so that the firewall can reach networks that aren't directly connected.

Step 2. Define a static default route.

Step 3. Configure the firewall to use more than one path to the Internet as it monitors reachability.

Step 4. Use dynamic routing protocols so that the firewall can learn up-to-date routing information from other routers.

Initial Configurations

The firewall used in this lab has been preconfigured with the commands shown in Example 3-1.

Example 3-1 Initial Firewall Configuration

```
Firewall(config)# interface ethernet0/0
Firewall(config-if)# nameif outside
Firewall(config-if)# security-level 0
Firewall(config-if)# ip address 192.168.100.10 255.255.255.0
Firewall(config-if)# no shutdown
!
Firewall(config-if)# interface ethernet0/1
Firewall(config-if)# nameif inside
Firewall(config-if)# security-level 100
Firewall(config-if)# ip address 192.168.1.1 255.255.255.0
Firewall(config-if)# no shutdown
```

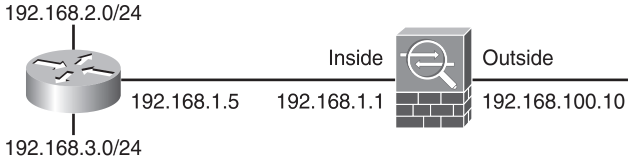
Video Presentation Reference

To display the current contents of a firewall's routing table, you can use the **show route** command.

Step 1: Configure Static Routes

In this step, a firewall is aware of its directly connected networks, but it doesn't know about two other networks that are connected to an inside router. Figure 3-1 shows the network topology.

Figure 3-1 Topology for Static Route Configuration



The inside router does not advertise the existence of the networks. Therefore, the firewall must be configured with static routes using the commands shown in Example 3-2.

Example 3-2 Static Route Configuration

```
Firewall(config)# route inside 192.168.2.0 255.255.255.0 192.168.1.5
Firewall(config)# route inside 192.168.3.0 255.255.255.0 192.168.1.5
```

Table 3-1 lists the syntax of the command that is used to configure a static route.

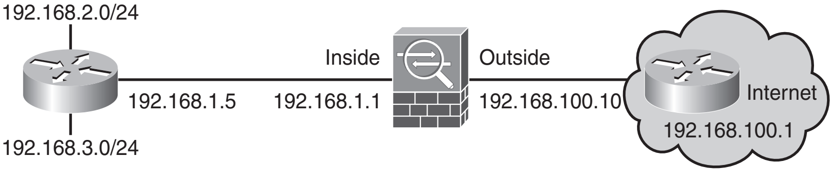
Table 3-1 Static Route Command Syntax

Command Syntax	Description
Firewall(config)# **route** *if-name ip-address netmask gateway-ip* [*distance*]	Defines a static route.

Step 2: Configure a Default Route

In this step, a single static route is configured as a default route, pointing toward the Internet. Figure 3-2 shows a network topology, and Example 3-3 shows the configuration commands that are entered.

Figure 3-2 Topology for Default Route Configuration



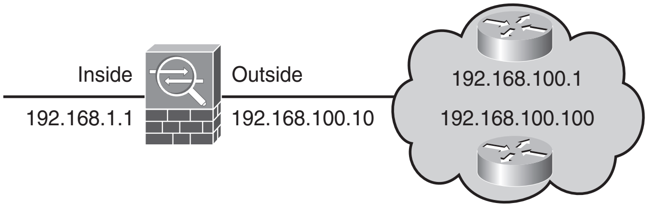
Example 3-3 Default Route Configuration

```
Firewall(config)# route outside 0.0.0.0 0.0.0.0 192.168.100.1 1
```

Step 3: Configure the Standby ISP

In this step, the firewall has two next-hop router choices to reach the Internet. Although two default routes could be configured, the Standby ISP feature is used instead. Figure 3-3 shows a network topology.

Figure 3-3 Topology for Standby ISP Configuration



The router at 192.168.100.1 is used as the primary gateway to reach the Internet. The firewall is configured to monitor the router's reachability; if it becomes unreachable, the firewall will begin using a default static route to gateway 192.168.100.100. Example 3-4 shows the commands that are entered in this step of the lab.

Example 3-4 Standby ISP Configuration

```
Firewall(config)# sla monitor 1
Firewall(config-sla-monitor)# type echo protocol ipIcmpEcho 192.168.100.1
    interface outside
Firewall(config-sla-monitor)# frequency 30
Firewall(config-sla-monitor)# exit
Firewall(config)# sla monitor schedule 1 life forever start-time now
!
Firewall(config)# track 100 rtr 1 reachability
Firewall(config)# route outside 0.0.0.0 0.0.0.0 192.168.100.1 1 track 100
!
Firewall(config)# route outside 0.0.0.0 0.0.0.0 192.168.100.100 2
```

Table 3-2 lists the syntax of commands that can be used to configure Standby ISP and SLA monitoring.

Table 3-2 Useful Standby ISP Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **sla monitor** *sla-id*	Defines an SLA monitor function.
Firewall(config-sla-monitor)# **type echo protocol ipIcmpEcho** *target* **interface** *if-name*	Identifies the monitor target and egress interface.
Firewall(config-sla-monitor)# **frequency** *seconds*	Sets the monitor poll period.
Firewall(config)# **sla monitor schedule** *sla-id* **life forever start-time now**	Sets the monitor process schedule to last "forever" and to start immediately.
Firewall(config)# **track** *track-id* **rtr** *sla-id* **reachability**	Binds an SLA monitor function with a track process.
Firewall(config)# **route** *if-name ip-address netmask gateway_ip* [*distance*] **track** *track-id*	Makes a static route dependent on reachability that is monitored by a track process.
Firewall# **show track**	Displays the current track process definition.

Step 4: Use a Dynamic Routing Protocol

In this step, the firewall is configured to use OSPF so that it can exchange routing information dynamically with other neighboring routers. Although subnets 192.168.4.0/24, 192.168.5.0/24, and 192.168.6.0/24 are not shown in Figure 3-1, the firewall learns about them through an OSPF neighbor. Example 3-5 shows the commands that are entered to configure OSPF.

Example 3-5 Configuring OSPF

```
Firewall(config)# router ospf 1
Firewall(config-router)# network 192.168.1.0 255.255.255.0 area 0
Firewall(config-router)# exit
!
Firewall(config)# interface ethernet0/2
Firewall(config-if)# ospf authentication message-digest
Firewall(config-if)# ospf message-digest 1 md5 mysecret
```

Lab 4

Firewall Administration over the Network

This Cisco Firewall Video Mentor lab shows you how to configure several different methods of remote access so that you can manage your firewall remotely.

The objectives of this lab are as follows:

- Configure remote Telnet session access
- Configure remote Secure Shell (SSH) access
- Configure remote Adaptive Security Device Manager (ASDM) access
- Take a brief tour of ASDM

Scenario

This lab contains four main steps:

Step 1. Set up inbound Telnet access for remote clients.

Step 2. Set up inbound SSH access for remote clients.

Step 3. Set up the firewall's HTTP server, and grant inbound ASDM access for remote clients.

Step 4. Look at an overview of ASDM and how to use it to manage an ASA or FWSM firewall platform.

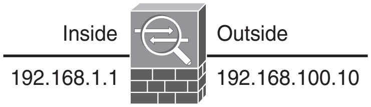
Initial Configurations

Example 4-1 lists the initial configuration commands used for this lab. Only the firewall's interfaces have been configured, as shown in Figure 4-1.

Example 4-1 Initial Firewall Configuration

```
Firewall(config)# interface ethernet0/0
Firewall(config-if)# nameif outside
Firewall(config-if)# security-level 0
Firewall(config-if)# ip address 192.168.100.10 255.255.255.0
Firewall(config-if)# no shutdown
!
Firewall(config-if)# interface ethernet0/1
Firewall(config-if)# nameif inside
Firewall(config-if)# security-level 100
Firewall(config-if)# ip address 192.168.1.1 255.255.255.0
Firewall(config-if)# no shutdown
```

Figure 4-1 Lab 4 Topology



Video Presentation Reference

Refer to the following descriptions of each step in Lab 4.

Step 1: Set Up Telnet Access

In this step, the firewall is configured to allow inbound Telnet access from any host on the inside network. Example 4-2 shows the configuration commands that are entered.

Example 4-2 Configuring Inbound Telnet Access

```
Firewall(config)# telnet 192.168.1.0 255.255.255.0 inside
Firewall(config)# telnet timeout 30
```

Table 4-1 lists the syntax of Telnet configuration commands.

Table 4-1 Useful Telnet Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **telnet** *ip-addr netmask if-name*	Permits inbound Telnet access from IP addresses on the ingress interface.
Firewall(config)# **telnet timeout** *minutes*	Sets the Telnet session idle timeout period.

Step 2: Set Up SSH Access

In this step, inbound SSH access is configured for clients on both the inside and outside networks. The configuration commands shown in Example 4-3 are entered.

Example 4-3 Configuring Inbound SSH Access

```
Firewall(config)# domain-name mycompany.com
Firewall(config)# crypto key generate rsa general-keys modulus 1024
!
Firewall(config)# ssh 192.168.1.0 255.255.255.0 inside
Firewall(config)# ssh 0.0.0.0 0.0.0.0 outside
Firewall(config)# ssh version 2
Firewall(config)# ssh timeout 30
```

Table 4-2 lists the syntax of SSH configuration commands.

Table 4-2 Useful SSH Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **domain-name** *name*	Sets the firewall's domain name, used in RSA key generation.
Firewall(config)# **crypto key generate rsa general-keys modulus** *modulus*	Generates the RSA keys for SSH use.
Firewall(config)# **ssh** *ip-addr netmask if-name*	Permits inbound Telnet access from IP addresses on the ingress interface.
Firewall(config)# **ssh version** {**1** | **2**}	Sets the permitted SSH version.
Firewall(config)# **ssh timeout** *minutes*	Sets the SSH idle session timeout period.

Step 3: Set Up ASDM Access

In this step, the firewall is configured to download and then use a new ASDM image. Then the embedded HTTP server and inbound access are configured on the firewall. This allows remote clients to use ASDM to manage and monitor the firewall. Example 4-4 shows the commands that are entered in this step of the lab.

Example 4-4 Commands Used to Configure ASDM

```
Firewall# show asdm image
Firewall# copy tftp://192.168.100.239/asdm-602.bin flash:/asdm.bin
Firewall# configure terminal
Firewall(config)# asdm image disk0:/asdm.bin
!
Firewall(config)# http 192.168.l.0 255.255.255.0 inside
Firewall(config)# http 0.0.0.0 0.0.0.0 outside
Firewall(config)# http server enable
```

Table 4-3 lists the syntax of commands that can be used to configure ASDM on a firewall.

Table 4-3 Useful ASDM Configuration Commands and Their Syntax

Command Syntax	Description
Firewall# **show asdm image**	Displays the current ASDM image name and location.
Firewall# **copy** *url device*:/*path*	Downloads an ASDM image.
Firewall(config)# **asdm image** *device*:/*path*	Specifies the location and filename of the ASDM image to use for new sessions.
Firewall(config)# **http** *ip-address netmask if-name*	Permits inbound HTTP access to IP addresses on an ingress interface.
Firewall(config)# **http server enable** [*port*]	Enables the firewall's embedded HTTP server on a specific port (the default is TCP 443).

Step 4: View ASDM Session Demonstration

This step demonstrates how an ASDM session is started and how ASDM can be used to manage a firewall. No additional configuration commands are required.

Lab 5

Using Multiple Security Contexts

This Cisco Firewall Video Mentor lab shows you how to configure a firewall so that it operates as several virtual or logical firewalls.

The objectives of this lab are as follows:

- Learn how to enable multiple security context mode
- Create and configure security contexts
- Open administrative sessions to the contexts
- Arrange security contexts within the firewall platform

Scenario

This lab contains five main steps:

Step 1. Enable multiple context mode.

Step 2. Create new security contexts.

Step 3. Administer the contexts through the CLI.

Step 4. Configure the security contexts and their interfaces.

Step 5. Learn context arrangement and how to solve packet-forwarding issues.

Initial Configurations

The firewall is in its default initial configuration at the start of this lab. Therefore, no additional commands are needed.

Video Presentation Reference

Refer to the following descriptions of each step in Lab 5.

Step 1: Enable Multiple Context Mode

In this step, the firewall is configured to operate in multiple context mode, where one or more virtual firewalls can be configured on a single physical firewall platform. Example 5-1 shows the commands that are entered to display the current security context mode and then enable multiple security context mode.

Example 5-1 Configuring Multiple Security Context Mode

```
Firewall# show version
Firewall# configure terminal
Firewall(config)# mode multiple
```

Table 5-1 lists the syntax of multiple context mode configuration commands.

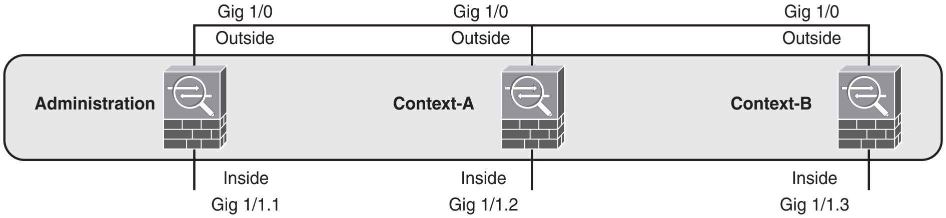
Table 5-1 Useful Multiple Context Mode Configuration Commands and Their Syntax

Command Syntax	Description
Firewall# **show version**	Displays current feature licensing, showing the number of security contexts that are supported.
Firewall# **show mode**	Displays the current operating mode.
Firewall(config)# [**no**] **mode multiple**	Begins operation in multiple security context mode. With the **no** keyword, the firewall returns to single context mode.

Step 2: Create New Security Contexts

In this step, new security contexts are created from within the system execution space. Figure 5-1 shows the network topology of the three new contexts: admin, context-a, and context-b. The new contexts will operate independently, but within the same physical firewall platform. The configuration commands shown in Example 5-2 are entered.

Figure 5-1 Lab 5 Topology



Example 5-2 Configuring Security Contexts

```
!
Firewall(config)# interface gigabitethernet1/1
Firewall(config-if)# no shut
Firewall(config-if)# interface gigabitethernet1/1.1
Firewall(config-if)# vlan 100
Firewall(config-if)# no shut
Firewall(config-if)# interface gigabitethernet1/1.2
Firewall(config-if)# vlan 101
Firewall(config-if)# no shut
Firewall(config-if)# exit
!
Firewall(config)# context  admin
Firewall(config-ctx)# description  Admin context
Firewall(config-ctx)# allocate-interface  gigabitethernet1/0
Firewall(config-ctx)# config-url  flash:/admin.cfg
Firewall(config-ctx)# exit
Firewall(config)# admin-context  admin
Firewall(config)# context  context-a
Firewall(config-ctx)# description  Example context A
Firewall(config-ctx)# allocate-interface  gigabitethernet1/0  intf0
Firewall(config-ctx)# allocate-interface  gigabitethernet1/1.1  intf1
Firewall(config-ctx)# config-url  flash:/context-a.cfg
Firewall(config-ctx)# exit
Firewall(config)# context  context-b
Firewall(config-ctx)# description  Example context B
Firewall(config-ctx)# allocate-interface  gigabitethernet1/0  intf0
Firewall(config-ctx)# allocate-interface   gigabitethernet1/1.2  intf1
Firewall(config-ctx)# config-url  flash:/context-b.cfg
Firewall(config-ctx)# exit
```

Table 5-2 lists the syntax of context configuration commands.

Table 5-2 Useful Context Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **context** *name*	Defines a security context.
Firewall(config-ctx)# **description** *text*	Adds a descriptive text string.
Firewall(config-ctx)# **allocate-interface** *if-name* [*map-name*] [**visible** | **invisible**]	Allocates or maps a firewall interface to the security context.
Firewall(config-ctx)# **config-url** *url*	Identifies the startup-config file for the security context.

Step 3: Administer Contexts Through the CLI

This step demonstrates how you can connect to the firewall platform with a CLI session and then move around to each security context. Example 5-3 shows the commands that are entered in this step of the lab.

Example 5-3 Commands Used to Administer Security Contexts

```
Firewall# changeto context admin
Firewall/admin# changeto context context-a
Firewall/context-a# changeto context context-b
Firewall/context-b# changeto context system
Firewall#
```

Table 5-3 lists the syntax of commands that can be used to administer contexts on a firewall.

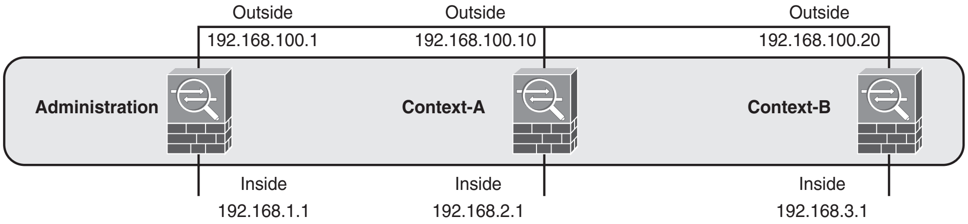
Table 5-3 Useful Context Administration Commands and Their Syntax

Command Syntax	Description
Firewall# **changeto context** *name*	Moves to a CLI session with security context *name*.
Firewall# **changeto system**	Moves to a CLI session with the system execution space.

Step 4: Configure Security Contexts and Their Interfaces

This step shows you how to configure the contexts and their interfaces, according to the network topology shown in Figure 5-2. The commands shown in Example 5-4 are entered during the configuration.

Figure 5-2 Lab 5 Context Interface Topology



Example 5-4 Commands Used to Configure Security Contexts and Their Interfaces

```
Firewall# changeto  context  admin
Firewall/admin# configure  terminal
Firewall/admin(config)# interface  gigabitethernet1/0
Firewall/admin(config-if)# nameif  outside
Firewall/admin(config-if)# ip address 192.168.100.1  255.255.255.0
Firewall/admin(config-if)# no shutdown
Firewall/admin(config-if)# exit
Firewall/admin(config)# exit

Firewall/admin# changeto  context  context-a
Firewall/context-a# configure  terminal
Firewall/context-a(config)# interface  intf0
Firewall/context-a(config-if)# nameif  outside
Firewall/context-a(config-if)# ip address 192.168.100.10  255.255.255.0
Firewall/context-a(config-if)# no shutdown
Firewall/context-a(config-if)# exit
Firewall/context-a(config)# interface  intf1
Firewall/context-a(config-if)# nameif  inside
Firewall/context-a(config-if)# ip address  192.168.2.1  255.255.255.0
Firewall/context-a(config-if)# no shutdown
Firewall/context-a(config-if)# exit
Firewall/context-a(config)# exit

Firewall/context-a# changeto  context  context-b
Firewall/context-b# configure  terminal
Firewall/context-b(config)# interface  intf0
Firewall/context-b(config-if)# nameif  outside
Firewall/context-b(config-if)# ip address 192.168.100.20  255.255.255.0
Firewall/context-b(config-if)# no shutdown
Firewall/context-b(config-if)# exit
Firewall/context-b(config)# interface  intf1
Firewall/context-b(config-if)# nameif  inside
Firewall/context-b(config-if)# ip address  192.168.3.1  255.255.255.0
Firewall/context-b(config-if)# no shutdown
Firewall/context-b(config-if)# exit
Firewall/context-b(config)# exit
```

Step 5: Learn Context Arrangement

This step demonstrates many of the topologies that can be created when security contexts share interfaces. The classifier function is explained, along with some pitfalls to keep in mind. Finally, the **mac-address auto** command is discussed so that each context interface can be assigned a unique MAC address.

Lab 6

Using Failover for High Availability

This Cisco Firewall Video Mentor lab shows you how to configure a pair of firewalls in failover mode so that they offer higher availability than a single firewall.

The objectives of this lab are as follows:

- Learn how active-standby failover works
- Learn how active-active failover works
- Learn how to tune failover operation

Scenario

This lab contains one scenario, in which a pair of ASA devices are configured to operate as a failover pair. Active-active failover is configured in multiple context mode in a series of eight steps:

Step 1. Identify the failover role.

Step 2. Set up LAN-based failover.

Step 3. Set up stateful failover.

Step 4. Tune the unit failover threshold.

Step 5. Set up the active-active failover groups.

Step 6. Define context interface addresses.

Step 7. Assign contexts to the two failover groups.

Step 8. Bootstrap the secondary firewall unit for failover.

Initial Configurations

The primary firewall begins with the multiple context configuration that resulted from Lab 5. The initial configuration commands for the system execution space are shown in Example 6-1.

Example 6-1 Initial ASA System Execution Space Configuration

```
hostname asa1
domain-name mycompany.com
enable password iE9elCMOvCJAfUw3 encrypted
!
interface Ethernet0/0
 description admin inside
!
interface Ethernet0/1
 shutdown
!
interface Ethernet0/2
 shutdown
!
interface Ethernet0/3
 shutdown
!
interface Management0/0
 shutdown
!
interface GigabitEthernet1/0
 description outside - all contexts
!
interface GigabitEthernet1/1
!
interface GigabitEthernet1/1.1
 description context-a inside
 vlan 100
!
interface GigabitEthernet1/1.2
 description context-b inside
 vlan 101
!
interface GigabitEthernet1/2
 shutdown
!
interface GigabitEthernet1/3
 shutdown
!
boot system disk0:/asa801-18-k8.bin
asdm image disk0:/asdm-523.bin
!
admin-context admin
context admin
  description Admin context
```

Example 6-1 Initial ASA System Execution Space Configuration

```
  allocate-interface Ethernet0/0
  allocate-interface GigabitEthernet1/0
  config-url disk0:/admin.cfg
!
context context-a
  description Example context A
  allocate-interface GigabitEthernet1/0 intf0
  allocate-interface GigabitEthernet1/1.1 intf1
  config-url disk0:/context-a.cfg
!
context context-b
  description Example context B
  allocate-interface GigabitEthernet1/0 intf0
  allocate-interface GigabitEthernet1/1.2 intf1
  config-url disk0:/context-b.cfg
```

The initial configuration commands for the “admin” context are shown in Example 6-2.

Example 6-2 Initial ASA “admin” Context Configuration

```
hostname admin
domain-name mycompany.com
enable password iE9elCMOvCJAfUw3 encrypted
!
interface Ethernet0/0
 nameif inside
 security-level 100
 ip address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet1/0
 nameif outside
 security-level 0
 ip address 192.168.100.1 255.255.255.0
!
passwd l1L6nJyCpFrdy9oK encrypted
!
route outside 0.0.0.0 0.0.0.0 192.168.100.3 1
!
http server enable
http 0.0.0.0 0.0.0.0 outside
telnet timeout 5
ssh 0.0.0.0 0.0.0.0 outside
ssh timeout 30
ssh version 2
```

The initial configuration commands for the “context-a” context are shown in Example 6-3.

Example 6-3 Initial ASA “context-a” Context Configuration

```
hostname context-a
domain-name mycompany.com
enable password iE9elCMOvCJAfUw3 encrypted
!
interface intf0
 nameif outside
 security-level 0
 ip address 192.168.100.10 255.255.255.0
!
interface intf1
 nameif inside
 security-level 100
 ip address 192.168.2.1 255.255.255.0
!
passwd l1L6nJyCpFrdy9oK encrypted
!
route outside 0.0.0.0 0.0.0.0 192.168.100.3 1
!
http server enable
http 0.0.0.0 0.0.0.0 outside
telnet timeout 5
ssh 0.0.0.0 0.0.0.0 outside
ssh timeout 30
ssh version 2
```

The initial configuration commands for the “context-b” context are shown in Example 6-4.

Example 6-4 Initial ASA “context-b” Context Configuration

```
hostname context-b
domain-name mycompany.com
enable password iE9elCMOvCJAfUw3 encrypted
!
interface intf0
 nameif outside
 security-level 0
 ip address 192.168.100.20 255.255.255.0
!
interface intf1
 nameif inside
 security-level 100
```

Example 6-4 Initial ASA "context-b" Context Configuration

```
 ip address 192.168.3.1 255.255.255.0
!
passwd l1L6nJyCpFrdy9oK encrypted
!
http server enable
http 0.0.0.0 0.0.0.0 outside
telnet timeout 5
ssh 0.0.0.0 0.0.0.0 outside
ssh timeout 30
ssh version 2
```

Video Presentation Reference

Refer to the following descriptions of each step in Lab 6.

Step 1: Identify the Failover Role

In this step, the "asa1" firewall is configured to identify itself as the primary failover unit. Example 6-5 shows the configuration commands that are entered.

Example 6-5 Configuring the Firewall as the Primary Failover Unit

```
asa1(config)# failover lan unit primary
asa1(config)# failover preempt 30
```

Table 6-1 lists the syntax of the primary failover unit configuration commands.

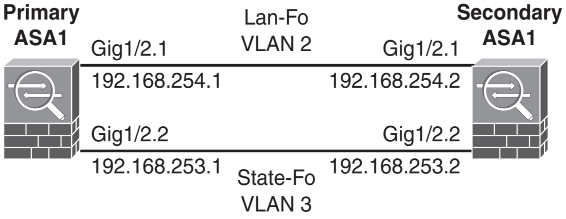
Table 6-1 Failover Role Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **failover lan unit** {**primary** | **secondary**}	Identifies the current firewall as either the primary or secondary failover unit.
Firewall(config)# **failover preempt** *seconds*	Forces the failover unit to assume the active role after a delay of *seconds* by preempting the other unit.

Step 2: Set Up LAN-Based Failover

In this step, LAN-based failover is configured on the primary unit so that it can begin to learn about a failover peer, as well as advertise itself to that peer. The LAN-based failover link is shown in Figure 6-1.

Figure 6-1 Failover Link Network Diagram



The configuration commands shown in Example 6-6 are entered to prepare the LAN-based failover interface and set up the LAN-based failover operation.

Example 6-6 Configuring LAN-Based Failover

```
asa1(config)# interface GigabitEthernet1/2
asa1(config-if)# no shutdown
asa1(config-if)# exit
asa1(config)# interface GigabitEthernet1/2.1
asa1(config-if)# description LAN Failover Interface
asa1(config-if)# vlan 2
asa1(config-if)# no shutdown
asa1(config-if)# exit
!
asa1(config)# failover lan interface lan-fo gig1/2.1
asa1(config)# failover interface ip lan-fo 192.168.254.1 255.255.255.0 standby
    192.168.254.2
asa1(config)# failover key MyBigSecret
```

Table 6-2 lists the syntax of LAN-based failover configuration commands.

Table 6-2 Useful LAN-Based Failover Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **failover lan interface** *if-name type mod/num*	Defines the LAN-based failover interface.
Firewall(config)# **failover interface ip** *if-name ip-address mask* **standby** *standby-address*	Assigns an IP address to the LAN-based failover interface and identifies the secondary unit's interface.
Firewall(config)# **failover key** *string*	Defines a key string that will be used to encrypt the LAN-based failover traffic between failover units.

Step 3: Set Up Stateful Failover

This step demonstrates how you can configure stateful failover so that the two failover units can synchronize information about connection state, address translation, ARP table entries, and so on. The stateful failover link is shown in Figure 6-1. Example 6-7 shows the commands that are entered in this step of the lab.

Example 6-7 Commands Used to Configure Stateful Failover

```
asa1(config)# interface GigabitEthernet1/2.2
asa1(config-if)# description Stateful Failover Interface
asa1(config-if)# vlan 3
asa1(config-if)# no shutdown
asa1(config-if)# exit
!
asa1(config)# failover link state-fo gig1/2.2
asa1(config)# failover interface ip state-fo 192.168.253.1 255.255.255.0
    standby 192.168.253.2
asa1(config)# failover replication http
```

Table 6-3 lists the syntax of stateful failover configuration commands.

Table 6-3 Useful Stateful Failover Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **failover link** *if-name type mod/num*	Sets aside and names the interface for stateful failover use.
Firewall(config)# **failover interface ip** *if-name ip-address mask* **standby** *standby-address*	Assigns an IP address to the stateful failover interface and identifies the secondary unit's stateful failover interface.
Firewall(config)# **failover replication http**	Replicates HTTP connection state information to the standby unit.

Step 4: Tune the Unit Failover Threshold

This step shows you how to tune the failover operation so that the firewall advertises itself to its peer at regular intervals. The firewall also expects to hear from its peer at the same regular intervals. The commands shown in Example 6-8 are entered during the configuration.

Example 6-8 Commands Used to Tune Unit Failover Operation

```
asa1(config)# failover polltime unit msec 200 holdtime msec 800
```

Table 6-4 lists the syntax of the unit poll time failover configuration command.

Table 6-4 Unit Failover Polling Command and Syntax

Command Syntax	Description
Firewall(config)# **failover polltime unit** [**msec**] *time* [**holdtime** [**msec**] *holdtime*]	Sets the unit "hello" message interval and holdtime.

Step 5: Set Up the Active-Active Failover Groups

This step shows you how to configure the two failover groups for active-active failover operation. The commands shown in Example 6-9 are entered during the configuration.

Example 6-9 Commands Used to Configure Failover Groups

```
asa1(config)# failover group 1
asa1(config-fover-group)# primary
asa1(config-fover-group)# preempt
asa1(config-fover-group)# polltime interface msec 500 holdtime 5
asa1(config-fover-group)# interface-policy 1
asa1(config-fover-group)# exit
```

Table 6-5 lists the failover group configuration command syntax.

Table 6-5 Useful Failover Group Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **failover group** {**1** | **2**}	Selects the failover group to configure.
Firewall(config-fover-group)# {**primary** | **secondary**}	Assigns the failover group to the primary or secondary failover unit, where normally it will be active.
Firewall(config-fover-group)# **preempt**	The assigned failover peer can preempt the active role for the failover group.
Firewall(config-fover-group)# **polltime interface** [**msec**] *time* [**holdtime** *holdtime*]	Sets the interface "hello" message interval and holdtime for the failover group.
Firewall(config-fover-group)# **interface-policy** *num*[**%**]	Sets the interface failure threshold that will trigger a failover for the failover group. Entering *num* signifies the number of interfaces. Entering *num%* (with a percent sign) signifies a percentage of the total number of interfaces.

Step 6: Define Context Interface Addresses

This step shows you how to configure IP addresses on context interfaces so that the context can participate in the failover operation. The commands shown in Example 6-10 are entered during the configuration.

Example 6-10 Commands Used to Configure Context Interfaces for Failover

```
asa1# changeto context admin
asa1/admin# config term
asa1/admin(config)# interface gig1/0
asa1/admin(config-if)# ip address  192.168.100.1 255.255.255.0 standby
    192.168.100.2
asa1/admin(config-if)# interface e0/0
asa1/admin(config-if)# ip address 192.168.1.1 255.255.255.0 standby 192.168.1.2
asa1/admin(config-if)# exit
asa1/admin(config)# exit
asa1/admin#

asa1/admin# changeto context context-a
asa1/context-a# config term
asa1/context-a(config)# interface intf0
asa1/context-a(config-if)# ip address 192.168.100.10 255.255.255.0 standby
    192.168.100.11
asa1/context-a(config-if)# interface intf1
asa1/context-a(config-if)# ip address 192.168.2.1 255.255.255.0 standby
    192.168.2.2
asa1/context-a(config-if)# exit
asa1/context-a(config)# monitor-interface inside
asa1/context-a(config)# exit
asa1/context-a#

asa1/context-a# changeto context context-b
asa1/context-b(config)# interface intf0
asa1/context-b(config-if)# ip address 192.168.100.20 255.255.255.0 standby
    192.168.100.21
asa1/context-b(config-if)# interface intf1
asa1/context-b(config-if)# ip  address 192.168.3.1 255.255.255.0 standby
    192.168.3.2
asa1/context-b(config-if)# exit
asa1/context-b(config)# monitor-interface inside
asa1/context-b(config)# exit
asa1/context-b#
```

Table 6-6 lists the context interface configuration command syntax for failover operation.

Table 6-6 Useful Context Interface Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **interface** *type mod/num*	Identifies the interface to be configured.
Firewall(config-if)# **ip address** *ip-address mask* **standby** *standby-address*	Defines the active unit's IP address and subnet mask, as well as the standby unit's IP address.
Firewall(config)# **monitor-interface** *if-name*	Enables failover monitoring on the interface.

Step 7: Assign Contexts to the Two Failover Groups

This step shows you how to assign security contexts to failover groups so that the contexts can be distributed arbitrarily between the two failover units. Failover operation is then enabled on the primary unit. The commands shown in Example 6-11 are entered in the system execution space during the configuration.

Example 6-11 Commands Used to Assign Contexts to Failover Groups

```
asa1/context-b# changeto system

asa1# config term
asa1(config)# context admin
asa1(config-ctx)# join-failover-group 1
asa1(config-ctx)# exit
asa1(config)# context context-a
asa1(config-ctx)# join-failover-group 1
asa1(config-ctx)# exit
asa1(config)# context context-b
asa1(config-ctx)# join-failover-group 2
asa1(config-ctx)# exit
!
asa1(config)# failover
```

Table 6-7 lists the command syntax for the context assignment configuration commands.

Table 6-7 Useful Context Assignment Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config-ctx)# **join-failover-group** {**1** | **2**}	Assigns the context to failover group 1 or 2.
Firewall(config)# **failover**	Enables failover operation on the firewall platform.

Step 8: Bootstrap the Secondary Firewall Unit

This step shows you how to enter enough failover configuration commands into the secondary failover unit to bootstrap its failover operation. From that point on, the secondary unit forms a failover peer relationship with the primary unit and synchronizes the rest of its configuration automatically.

The commands shown in Example 6-12 are entered in the secondary unit's system execution space during the configuration.

Example 6-12 Commands Used to Bootstrap the Secondary Failover Unit

```
asa2(config)# interface GigabitEthernet1/2
asa2(config-if)# no shutdown
asa2(config-if)# exit
!
asa2(config)# interface GigabitEthernet1/2.1
asa2(config-if)# description LAN Failover Interface
asa2(config-if)# vlan 2
!
asa2(config-if)# interface GigabitEthernet1/2.2
asa2(config-if)# description STATE Failover Interface
asa2(config-if)# vlan 3
asa2(config-if)# exit
!
asa2(config)# failover lan unit secondary
asa2(config)# failover lan interface lan-fo gig1/2.1
asa2(config)# failover interface ip lan-fo 192.168.254.1 255.255.255.0 standby
    192.168.254.2
asa2(config)# failover key MyBigSecret
asa2(config)# failover
```

Monitoring Failover Operation

You can use the **show failover** and **show failover group** {**1** | **2**} commands on either failover peer unit to monitor the current failover status.

Lab 7

Failover in Action

This Cisco Firewall Video Mentor lab demonstrates several different conditions that cause a failure in a firewall interface or unit. This, in turn, triggers the failover operation.

This lab also works through the "hitless" upgrade process, where you can upgrade the operating system image in each of the active-active failover units—without impacting the traffic passing through the firewall pair.

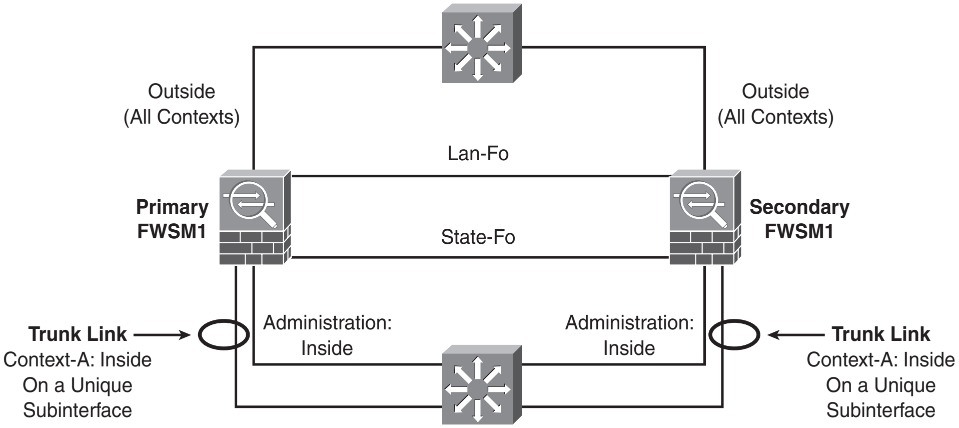
The objectives of this lab are as follows:

- Observe a physical interface failure
- Observe a logical interface failure
- Observe a failover unit failure
- Observe a hitless upgrade

Scenario

This lab contains several failover demonstrations, using the two firewalls from Lab 6 configured as a failover pair. A network diagram of the failover pair is shown in Figure 7-1.

Figure 7-1 Network Diagram for Lab 7 Scenarios



- Scenario 1: Force the link state on the "admin" context inside interface to go down. This causes a failure on a physical firewall interface, which triggers a failover operation.
- Scenario 2: Remove the VLAN used on the "context-a" context inside interface while the interface stays up. This causes a failure on a logical interface, which indirectly triggers a failover operation.

- Scenario 3: Reload the primary firewall unit suddenly, as if it experiences a power cycle. This causes the failure of an entire firewall unit, which triggers a failover operation.
- Scenario 4: Manually control the failover operation so that the code image can be upgraded on each firewall unit. The upgrades occur while live connections are being handled by the failover pair, such that no traffic is impacted or lost.

Initial Configurations

The failover pair of ASA devices configured in Lab 6 is used for the interface and unit failover demonstrations. No additional configuration commands are necessary to perform the first three scenarios.

The final scenario involves active-active failover and a hitless upgrade on two running firewalls. The FWSM platform is used. The initial configurations are listed in the Scenario 4 section.

Video Presentation Reference

Refer to the following descriptions of each scenario presented in Lab 7.

Scenario 1: Physical Interface Failure

In this scenario, the interface polltime remains configured at 500 ms, with a holdtime of 5 seconds. The switch interface connected to the primary unit's "admin" context inside interface (physical interface Ethernet0/0) is shut down, causing the link status to go down.

No additional configuration is necessary on the failover pair.

Scenario 2: Logical Interface Failure

In this scenario, the interface polltime remains configured at 500 ms, with a holdtime of 5 seconds. The inside interface of the "context-a" context is mapped to VLAN 100, which is carried over a trunk link from an upstream switch to each firewall in the failover pair.

On the switch connected to the primary failover unit, VLAN 100 is removed from the trunk. This simulates a failure on a logical interface, where the two failover units can no longer communicate with each other on the context interface.

No additional configuration is necessary on the failover pair.

Scenario 3: Failover Unit Failure

In this scenario, the unit polltime remains configured at 200 ms, with a holdtime of 800 ms. You reload the entire primary failover unit by entering the **reload** command from the system execution space.

This simulates a failover unit failure, where the secondary unit can no longer detect the primary unit.

No additional configuration is necessary on the failover pair.

Scenario 4: Hitless Code Upgrade

In this scenario, the two failover units are configured for active-active failover operation. This scenario is unique because it uses two FWSMs as a failover pair.

The primary FWSM unit begins with the multiple context configuration that resulted from Lab 6. The initial Catalyst 6500 configuration commands related to FWSM operation are shown in Example 7-1. In this case, the primary FWSM is contained in module 3, and the secondary FWSM in slot 4.

Example 7-1 Initial Catalyst 6500 Supervisor Configuration

```
vlan 2
 name lan-fo
!
vlan 3
 name stateful-fo
!
vlan 10
 name FWSM-outside
!
vlan 100
 name FWSM-inside
!
vlan 101
 name context-a-inside
!
vlan 102
 name context-b-insidefirewall vlan-group 1  2,3,10,100-103
!
firewall module 3 vlan-group 1
firewall module 4 vlan-group 1
```

The initial configuration commands for the primary FWSM system execution space are shown in Example 7-2.

Example 7-2 Initial FWSM System Execution Space Configuration

```
hostname fwsm1
domain-name mycompany.com
enable password iE9elCMOvCJAfUw3 encrypted
passwd l1L6nJyCpFrdy9oK encrypted
!
interface Vlan2
 description LAN Failover Interface
!
interface Vlan3
 description STATE Failover Interface
!
interface Vlan10
!
interface Vlan100
!
interface Vlan101
!
interface Vlan102
!
failover
failover lan unit primary
failover lan interface lan-fo Vlan2
failover polltime unit msec 500 holdtime 3
failover key *****
failover replication http
failover link state-fo Vlan3
failover interface ip lan-fo 192.168.254.1 255.255.255.0 standby 192.168.254.2
failover interface ip state-fo 192.168.253.1 255.255.255.0 standby 192.168.253.2
failover group 1
  preempt
  polltime interface 3
failover group 2
  secondary
  preempt
  polltime interface 3
!
admin-context admin
context admin
  description Admin context
  allocate-interface Vlan10
  allocate-interface Vlan100
  config-url disk:/admin.cfg
  join-failover-group 1
!
```

Example 7-2 Initial FWSM System Execution Space Configuration

```
context context-a
  description Example context A
  allocate-interface Vlan10 intf0
  allocate-interface Vlan101 intf1
  config-url disk:/context-a.cfg
  join-failover-group 1
!

context context-b
  description Example context B
  allocate-interface Vlan102 intf1
  allocate-interface Vlan103 intf0
  config-url disk:/context-b.cfg
  join-failover-group 2
```

The initial configuration commands for the primary FWSM “admin” context are shown in Example 7-3.

Example 7-3 Initial FWSM “admin” Context Configuration

```
hostname admin
passwd l1L6nJyCpFrdy9oK encrypted
enable password iE9elCMOvCJAfUw3 encrypted
!
interface Vlan10
 nameif outside
 security-level 0
 ip address 192.168.100.1 255.255.255.0 standby 192.168.100.2
!
interface Vlan100
 nameif inside
 security-level 100
 ip address 192.168.1.1 255.255.255.0 standby 192.168.1.2
!
monitor-interface outside
route outside 0.0.0.0 0.0.0.0 192.168.100.3 1
http 0.0.0.0 0.0.0.0 outside
http server enable
ssh 0.0.0.0 0.0.0.0 outside
ssh version 2
!
tftp-server outside 192.168.100.239 /
```

The initial configuration commands for the primary FWSM “context-a” context are shown in Example 7-4.

Example 7-4 Initial FWSM “context-a” Context Configuration

```
hostname context-a
domain-name mycompany.com
passwd l1L6nJyCpFrdy9oK encrypted
enable password iE9elCMOvCJAfUw3 encrypted
!
interface intf0
 nameif outside
 security-level 0
 ip address 192.168.100.10 255.255.255.0 standby 192.168.100.11
!
interface intf1
 nameif inside
 security-level 100
 ip address 192.168.2.1 255.255.255.0 standby 192.168.2.2
!
access-list acl_outside extended permit ip 172.21.4.0 255.255.254.0 host
    192.168.100.100
access-list acl_inside extended permit ip 192.168.2.0 255.255.255.0 any
access-list acl_in extended permit ip 192.168.2.0 255.255.255.0 any
!
static (inside,outside) 192.168.100.100 192.168.2.100 netmask 255.255.255.255
!
access-group acl_outside in interface outside
access-group acl_in in interface inside
route outside 0.0.0.0 0.0.0.0 192.168.100.3 1
!
http server enable
http 0.0.0.0 0.0.0.0 outside
ssh 0.0.0.0 0.0.0.0 outside
ssh timeout 30
ssh version 2
```

The initial configuration commands for the primary FWSM “context-b” context are shown in Example 7-5.

Example 7-5 Initial FWSM "context-b" Context Configuration

```
hostname context-b
domain-name mycompany.com
passwd l1L6nJyCpFrdy9oK encrypted
enable password iE9elCMOvCJAfUw3 encrypted
!
interface intf1
 nameif inside
 security-level 100
 ip address 192.168.3.1 255.255.255.0 standby 192.168.3.2
!
interface intf0
 nameif outside
 security-level 0
 ip address 192.168.100.20 255.255.255.0 standby 192.168.100.21
!
http 0.0.0.0 0.0.0.0 outside
http server enable
ssh 0.0.0.0 0.0.0.0 outside
ssh timeout 30
ssh version 2
```

Finally, the initial configuration for the system execution space and all contexts are identical on the secondary FWSM unit—except for the failover configuration in the system execution space. The secondary unit's failover configuration commands are shown in Example 7-6.

Example 7-6 Initial Secondary FWSM Failover Configuration

```
failover
failover lan unit secondary
failover lan interface lan-fo Vlan2
failover polltime unit msec 500 holdtime 3
failover key *****
failover replication http
failover link state-fo Vlan3
failover interface ip lan-fo 192.168.254.1 255.255.255.0 standby 192.168.254.2
failover interface ip state-fo 192.168.253.1 255.255.255.0 standby 192.168.253.2
failover group 1
  preempt
  polltime interface 3
failover group 2
  secondary
  preempt
  polltime interface 3
```

The primary unit is active for the “admin” and “context-a” contexts, and the secondary unit is active for the “context-b” context. The code image running on each failover unit is upgraded from 3.2(1) to 3.2(2) individually, as part of a hitless upgrade process.

The following steps demonstrate the hitless upgrade. All commands are entered in the system execution space on the primary unit.

Step 1. Download the new code image to each failover unit.

The code image is downloaded into FWSM flash with the following command:

```
Firewall# copy tftp: flash:image
```

Currently the FWSM platform supports only one code image in flash memory. Therefore, you don’t have to configure the specific image location and filename, as the ASA platform requires.

Step 2. Force the primary unit to be active in all contexts.

The following command causes the primary unit to immediately take the active role in all contexts:

```
Firewall# failover active
```

Step 3. Reload the secondary unit:

```
Firewall# failover reload-standby
```

When you force the secondary unit to reload, it automatically picks up the new code image.

Step 4. Swap failover roles.

As soon as the secondary unit finishes reloading and the failover operation has stabilized, the following command is used to push the active role to the secondary unit:

```
Firewall# no failover active
```

The primary unit immediately tells the secondary unit to take over the active role in all contexts, while the primary unit assumes the standby role.

Step 5. Reload the primary unit:

```
Firewall# reload
```

When you force the primary unit to reload, it automatically picks up the new code image. In the meantime, the secondary unit handles all firewall operations.

Step 6. Resume the original failover roles.

As soon as the primary unit has finished reloading and the failover operation has stabilized, the unit needs to take over the active role in the “admin” and “context-a” contexts—returning to the role it had before the hitless upgrade began. The following command can be used to accomplish this:

```
Firewall# failover active group 1
```

However, because the primary unit has been configured to preempt the active role for failover group 1, it automatically assumes the active role as soon as it reloads.

Lab 8

Setting Up Address Translation and Connection Limits

This Cisco Firewall Video Mentor lab shows you how to configure address translation so that hosts on one firewall interface will appear with different IP addresses on another interface. Limits on UDP and TCP connections are also configured.

The objectives of this lab are as follows:

- Configure at least one example of each type of address translation
- Configure connection limits for UDP and TCP protocols

Scenario

This lab contains several scenarios, presented in the following steps:

Step 1. Configure static NAT.

Step 2. Configure policy NAT.

Step 3. Configure identity NAT.

Step 4. Configure NAT exemption.

Step 5. Configure dynamic NAT and PAT.

Step 6. Configure UDP and TCP connection limits.

Initial Configurations

The firewall begins with a simple configuration used in previous labs. Although the lab configurations take place on the “context-a” security context, they could just as easily be configured on a firewall running in single context mode. The initial configuration commands for the firewall are shown in Example 8-1.

Example 8-1 Initial Firewall Configuration

```
hostname context-a
!
interface intf0
 nameif outside
 security-level 0
 ip address 192.168.100.1 255.255.255.0 standby 192.168.100.2
!
interface intf1
 nameif inside
 security-level 100
 ip address 192.168.2.1 255.255.255.0 standby 192.168.2.2
```

Video Presentation Reference

Refer to the following descriptions of each step in Lab 8.

Step 1: Configure Static NAT

In this step, static NAT is used to translate virtual outside address 192.168.100.100 to the real inside host at 192.168.2.100. In addition, the following two services are translated:

- TCP port 80 at the virtual outside address 192.168.100.101 is translated into the HTTP service running on the real inside host 192.168.2.90 TCP port 80.
- TCP port 25 at the virtual outside address 192.168.100.101 is translated into the SMTP service running on the real inside host 192.168.2.91 TCP port 25.

Example 8-2 shows the configuration commands that are entered.

Example 8-2 Configuring Static NAT

```
Firewall(config)# static (inside,outside) 192.168.100.100 192.168.2.100 netmask
    255.255.255.255
Firewall(config)# static (inside,outside) tcp 192.168.100.101 80 192.168.2.90 80
    netmask 255.255.255.255
Firewall(config)# static (inside,outside) tcp 192.168.100.101 25 192.168.2.91 25
    netmask 255.255.255.255
```

Table 8-1 lists the syntax of the static NAT commands.

Table 8-1 Static NAT Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **static** (*real-if,mapped-if*) {*mapped-ip* | **interface**} *real-ip* [**netmask** *mask*] [**dns**]	Defines a static translation between the mapped and real IP addresses, across the real and mapped interfaces.
Firewall(config)# **static** (*real-if,mapped-if*) {**tcp** | **udp**} {*mapped-ip* | **interface**} *mapped-port real-ip real-port* [**netmask** *mask*] [**dns**]	Defines a static translation between the mapped and real IP addresses, protocols, and port numbers, across the real and mapped interfaces and port numbers.

Step 2: Configure Policy NAT

In this step, policy NAT is used to translate the inside host at 192.168.2.102 to two different outside addresses, depending on the IP addresses that the inside host is communicating with. The **static** command is used to set up the policy NAT for connections initiating in both the inbound and outbound directions.

The configuration commands shown in Example 8-3 are entered to prepare the following policy NAT scenarios:

- Translate inside host 192.168.2.102 to outside address 192.168.100.102 when it communicates with any host in the 172.16.0.0/16 subnet.
- Translate inside host 192.168.2.102 to outside address 192.168.100.202 when it communicates with any host in the 172.17.0.0/16 subnet.

Example 8-3 Configuring Policy NAT with the static Command

```
Firewall(config)# access-list policynat1 permit ip host 192.168.2.102 172.16.0.0
    255.255.0.0
Firewall(config)# static (inside,outside) 192.168.100.102 access-list policynat1
Firewall(config)# access-list  policynat2  permit ip host 192.168.2.102
    172.17.0.0 255.255.0.0
Firewall(config)# static (inside,outside) 192.168.100.202 access-list policynat2
```

In addition, a policy NAT is configured to translate inside host 192.168.2.102 to outside address 192.168.100.203 when it communicates with hosts in the 172.18.0.0/16 subnet. The **global** and **nat** commands are used to set up policy NAT for only the outbound direction. The commands shown in Example 8-4 are entered to configure the policy NAT scenario.

Example 8-4 Configuring Policy NAT with the global and nat Commands

```
Firewall(config)# access-list policynat3 permit  ip  host  192.168.2.102
    172.18.0.0 255.255.0.0
Firewall(config)# global (outside) 1  192.168.100.203
Firewall(config)# nat (inside) 1 access-list policynat3
```

Table 8-2 lists the syntax of policy NAT configuration commands.

Table 8-2 Useful Policy NAT Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **access-list** *acl-name* **permit ip** *real-ip real-mask foreign-ip foreign-mask*	Defines the condition that will permit a policy NAT translation.
Firewall(config)# **static (***real-if*,*mapped-if***)** *mapped-ip* **access-list** *acl-name* [**dns**]	Applies the access list to a static NAT to make it conditional.
Firewall(config)# **global (***mapped-if***)** *nat-id global-ip*[-*global-ip*] [**netmask** *global-mask*]	Defines an outside address or range of addresses to be used with a **nat** command for policy NAT.
Firewall(config)# **nat (***real-if***)** *nat-id* **access-list** *acl-name* [**dns**] [**outside**]	Defines the inside interface and an access list that will trigger policy NAT when paired with a **global** command.

Step 3: Configure Identity NAT

This step demonstrates how you can configure identity NAT so that one or more inside host addresses are seen on the outside of the firewall unchanged. In other words, no NAT takes place for the addresses. Example 8-5 shows the command that is entered in this step of the lab.

Example 8-5 Command Used to Configure Identity NAT

```
Firewall(config)# nat (inside) 0 192.168.2.128 255.255.255.128
```

Table 8-3 lists the syntax of the identity NAT configuration command.

Table 8-3 Useful Identity NAT Command and Its Syntax

Command Syntax	Description
Firewall(config)# **nat (***real-if***)** **0** *real-ip real-mask* [**dns**]	Defines the inside (real) IP addresses that will not be translated.

Step 4: Configure NAT Exemption

This step shows you how to configure NAT exemption so that inside addresses in the 192.168.3.0/24 and 192.168.4.0/24 subnets pass through the firewall unchanged, without translation. The commands shown in Example 8-6 are entered during the configuration.

Example 8-6 Commands Used to Configure NAT Exemption

```
Firewall(config)# access-list natexempt extended permit ip 192.168.3.0
    255.255.255.0 any
Firewall(config)# access-list natexempt extended permit ip 192.168.4.0
    255.255.255.0 any
Firewall(config)# nat (inside) 0 access-list natexempt
```

Table 8-4 lists the syntax of NAT exemption configuration commands.

Table 8-4 Useful NAT Exemption Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **access-list** *acl-name* **permit ip** *real-ip real-mask foreign-ip foreign-mask*	Permits the inside source addresses and their destination addresses that will be exempted from NAT.
Firewall(config)# **nat** (*real-if*) **0 access-list** *acl-name* [**dns**] [**outside**]	Applies the access list to NAT exemption, using the special nat-id 0.

Step 5: Configure Dynamic NAT and PAT

This step shows you how to configure dynamic NAT and PAT so that inside host addresses in the 192.168.2.0/24 subnet will be translated under the following three conditions:

- Use dynamic NAT to translate an inside address to the next available outside address in the pool 192.168.100.20 to 192.168.100.30.
- As soon as all the dynamic NAT addresses are in use, use dynamic PAT to translate an inside address to outside address 192.168.100.31 or 192.168.100.32. The inside port number will be translated to the next available port number associated with the outside address.
- As soon as all the previously described translation slots are in use, use dynamic PAT to translate an inside address to the address used by the firewall's outside interface. The inside port number will be translated to the next available port number associated with the outside interface address.

The commands shown in Example 8-7 are entered during the configuration.

Example 8-7 Commands Used to Configure Dynamic NAT and PAT

```
Firewall(config)# global (outside) 3 192.168.100.20-192.168.100.30 netmask
    255.255.255.0
Firewall(config)# nat (inside) 3 192.168.2.0 255.255.255.0
Firewall(config)# global (outside) 3 192.168.100.31
Firewall(config)# global (outside) 3 192.168.100.32
Firewall(config)# global (outside) 3 interface
```

Table 8-5 lists the failover group configuration command syntax.

Table 8-5 Useful Dynamic NAT and PAT Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **global** (*mapped-if*) *nat-id* *global-ip*[-*global-ip*] [**netmask** *global-mask*]	Defines a range of outside addresses to be used in a dynamic NAT pool.
Firewall(config)# **global** (*mapped-if*) *nat-id global-ip*	Defines an outside address to be used for dynamic PAT.
Firewall(config)# **global** (*mapped-if*) *nat-id* **interface**	Defines an interface address to be used for dynamic PAT.
Firewall(config)# **nat** (*real-if*) *nat-id* *real-ip mask* [**dns**] [**outside**]	Identifies the inside interface and addresses that will be translated with dynamic NAT or PAT.

Step 6: Set UDP and TCP Connection Limits

This step shows you how to add keywords to the **static** and **nat** commands to set limits on UDP and TCP connections taking place through address translations. The commands shown in Example 8-8 are entered during the configuration. The previous **nat (inside)** configuration must be removed before the connection limit keywords can be added as the **nat** command is reentered.

Example 8-8 Commands Used to Configure Connection Limits

```
Firewall(config)# no nat (inside) 3 192.168.2.0 255.255.255.0
Firewall(config)# nat (inside) 3 192.168.2.0 255.255.255.0 tcp 200 50 udp 100
```

Table 8-6 lists the connection limit configuration command syntax.

Table 8-6 Useful Connection Limit Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **static** (*real-if,mapped-if*) *mapped-ip* **access-list** *acl-name* [**dns**] [**norandomseq**] [**tcp** *max-conns* [*emb-limit*]] [**udp** *udp-max-conns*]	Defines optional TCP and UDP connection limits for address translations configured with the **static** command.
Firewall(config)# **nat** (*real-if*) *nat-id real-ip* *mask* [**dns**] [**outside**] [**norandomseq**] [**tcp** *max-conns* [*emb-limit*]] [**udp** *udp-max-conns*]	Defines optional TCP and UDP connection limits for address translations configured with the **nat** command.
Firewall(config)# **timeout** [**conn** *hh:mm:ss*] [**udp** *hh:mm:ss*]	Defines the TCP and UDP connection idle times.

Lab 9

Setting Up Firewall Rules

This Cisco Firewall Video Mentor lab shows you how to configure and apply access lists to implement security policies on a firewall. Object groups are also used to group common parameters and simplify the access list structure.

The objectives of this lab are as follows:

- Configure an access list and apply it to a firewall interface
- Configure object groups and use them in an access list

Scenario

This lab contains several scenarios, presented in the following steps:

Step 1. Configure an access list to filter outbound connections initiated on the inside interface.

Step 2. Configure an access list to filter inbound connections initiated on the outside interface.

Step 3. Configure a network object group that contains a list of IP addresses.

Step 4. Configure an enhanced service object group that contains a list of protocols and port numbers.

Step 5. Reconfigure the outside interface access list to use the network and service object groups.

Initial Configurations

The firewall begins with a simple configuration used in previous labs. Although the lab configurations take place on the "context-a" security context, they could just as easily be configured on a firewall running in single context mode. The initial configuration commands for the firewall are shown in Example 9-1.

Example 9-1 Initial Firewall Configuration

```
hostname context-a
!
interface intf0
 nameif outside
 security-level 0
 ip address 192.168.100.10 255.255.255.0 standby 192.168.100.11
!
interface intf1
 nameif inside
 security-level 100
 ip address 192.168.2.1 255.255.255.0 standby 192.168.2.2
```

Video Presentation Reference

Refer to the following descriptions of each step in Lab 9.

Step 1: Configure an Access List for the Inside Interface

In this step, an access list named *acl_inside* is configured to filter traffic that is coming into the inside interface. This filters connections that are initiated in the *outbound* direction, from the inside interface toward the outside interface.

IP traffic from any inside host address in the 192.168.2.0/24 subnet is permitted to enter the inside interface. Example 9-2 shows the configuration commands that are entered.

Example 9-2 Configuring an Access List for the Inside Interface

```
Firewall(config)# access-list acl_inside extended permit ip 192.168.2.0
    255.255.255.0 any
!
Firewall(config)# access-group acl_inside in interface inside
```

Table 9-1 lists the syntax of the access list configuration commands.

Table 9-1 Access List Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **access-list** *acl-id* [**line** *line-num*] {**permit** | **deny**} *protocol source-addr source-mask* [*operator sport*] *destination-addr destination-mask* [*operator dport*] [**log** [[**disable** | **default**] | [*level*]] [**interval** *secs*]]	Defines an access control entry that is added to an access list.
Firewall(config)# **access-group** *acl-id* {**in** | **out**} **interface** *if_name*	Applies the access list to the interface in the specified direction.

Step 2: Configure an Access List for the Outside Interface

In this step, an access list named *acl_outside* is configured to filter traffic that is coming into the outside interface. This filters connections that are initiated in the *inbound* direction, from the outside interface toward the inside interface.

The configuration commands shown in Example 9-3 are entered to build the following access list policies:

- Permit inbound connections from any outside address to destination address 192.168.100.100 and 192.168.100.101, destined for TCP ports 80 and 443.
- Permit inbound VPN connections from any outside address to destination address 192.168.100.199. Only GRE and ESP protocols will be permitted, in addition to destination port UDP 500 (ISAKMP/IKE), UDP ports 10000 and 10001, UDP port 4500, and TCP port 10000.

Example 9-3 Configuring an Access List for the Outside Interface

```
Firewall(config)# access-list acl_outside permit tcp any host 192.168.100.100
    eq www
Firewall(config)# access-list acl_outside permit tcp any host 192.168.100.100
    eq https
Firewall(config)# access-list acl_outside permit tcp any host 192.168.100.101
    eq www
Firewall(config)# access-list acl_outside permit tcp any host 192.168.100.101
    eq https
!
Firewall(config)# access-list acl_outside permit esp any host 192.168.100.199
Firewall(config)# access-list acl_outside permit gre any host 192.168.100.199
Firewall(config)# access-list acl_outside permit udp any host 192.168.100.199
    eq isakmp
Firewall(config)# access-list acl_outside permit udp any host 192.168.100.199
    range 10000 10001
Firewall(config)# access-list acl_outside permit udp any host 192.168.100.199
    eq 4500
Firewall(config)# access-list acl_outside permit tcp any host 192.168.100.199
    eq 10000
!
Firewall(config)# access-group  acl_outside in interface outside
```

Step 3: Configure a Network Object Group

This step demonstrates how you can configure a network object group that contains a list of IP addresses that will have similar security policies applied. Object group *server_group1* contains two host addresses: 192.168.100.100 and 192.168.100.101. Example 9-4 shows the commands that are entered in this step of the lab. The object group is applied to the access list in Step 4.

Example 9-4 Configuring a Network Object Group

```
Firewall(config)# object-group network server_group1
Firewall(config-network)# network-object host 192.168.100.100
Firewall(config-network)# network-object host 192.168.100.101
Firewall(config)# exit
```

Table 9-2 lists the syntax of the network object group configuration commands.

Table 9-2 Useful Network Object Group Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **object-group network** *name*	Defines the network object group.
Firewall(config-network)# **description** *text*	Adds a descriptive text label to the object group.
Firewall(config-network)# **group-object** *name*	References a different network object group, if needed.
Firewall(config-network)# **network-object** {*ip-address mask* | **host** *ip-address*}	Adds an IP subnet or a single IP host address to the object group.

Step 4: Configure an Enhanced Service Object Group

This step shows you how to configure an enhanced object group that contains protocols and port numbers that will be applied in similar security policies. Object group *service_group1* contains TCP ports 80 and 443, or **www** and **https**, respectively.

The commands shown in Example 9-5 are entered during the configuration.

Example 9-5 Commands Used to Configure the *service_group1* Enhanced Service Object Group

```
Firewall(config)# object-group service service_group1
Firewall(config-service)# service-object tcp http
Firewall(config-service)# service-object tcp https
Firewall(config-service)# exit
```

A second object group, *service_vpn*, is configured so that it contains the ESP and GRE protocols, as well as the following destination port numbers: UDP port 500, UDP ports 10000 and 10001, UDP port 4500, and TCP port 10000. Example 9-6 shows the commands that are entered during the configuration.

Example 9-6 Commands Used to Configure the *service_vpn* Enhanced Service Object Group

```
Firewall(config)# object-group service service_vpn
Firewall(config-service)# service-object esp
Firewall(config-service)# service-object gre
Firewall(config-service)# service-object udp eq isakmp
Firewall(config-service)# service-object udp range 10000 10001
Firewall(config-service)# service-object udp eq 4500
Firewall(config-service)# service-object tcp eq 10000
Firewall(config-service)# exit
```

Table 9-3 lists the syntax of the service object group configuration commands.

Table 9-3 Useful Enhanced Service Object Group Configuration Commands and Their Syntax

Command Syntax	Description
Firewall(config)# **object-group service** *name*	Defines the enhanced service object group.
Firewall(config-service)# **description** *text*	Adds a descriptive text label to the object group.
Firewall(config-service)# **group-object** *name*	References a different service object group, if needed.
Firewall(config-service)# **service-object** [**source** [*src-op*] *src-port*] [*dest-op*] *dest-port*	Adds port operators and port numbers to the object group.

Step 5: Configure an Access List to Use the Object Groups

This step shows you how to configure the access list *acl_outside* so that it references the network and service object groups created in Steps 3 and 4 of this lab.

First, the *acl_outside* access list configuration is cleared to erase any previous configuration from Step 2 of this lab. Then the commands shown in Example 9-7 are entered to reconfigure the access list so that it references the object groups. The end result is identical to the access list that was entered in Step 2, although the access list configuration is simplified.

Example 9-7 Commands Used to Reconfigure the *acl_outside* Access List

```
Firewall(config)# clear configure access-list acl_outside
!
Firewall(config)# access-list acl_outside permit object-group service_vpn any
    host 192.168.100.199
Firewall(config)# access-list acl_outside permit object-group service_group1 any
    object-group server_group1
!
Firewall(config)# access-group acl_outside in interface outside
```

Lab 10

Setting Up a DMZ

This Cisco Firewall Video Mentor lab shows you how to add a demilitarized zone (DMZ) interface to a firewall, whereas previous labs dealt with only inside and outside interfaces.

The objectives of this lab are to configure address translation and access lists as security policies for the following scenarios:

- Connections initiated from a higher security level interface toward a lower one
- Connections initiated from a lower security level interface toward a higher one

Scenario

This lab contains several scenarios, presented in the following steps:

Step 1. Consider connections initiated from the inside interface toward the DMZ, and configure the firewall accordingly.

Step 2. Consider connections initiated from the DMZ interface toward the outside, and configure the firewall accordingly.

Step 3. Consider connections initiated from the outside interface toward the DMZ, and configure the firewall accordingly.

Step 4. Consider connections initiated from the DMZ interface toward the inside, and configure the firewall accordingly.

Step 5. Double-check the DMZ access list for conflicting entries.

Initial Configurations

The firewall begins with a simple configuration used in previous labs. Although the lab configurations take place on the "context-a" security context, they could just as easily be configured on a firewall running in single context mode.

The firewall is configured with an inside and outside interface, with address translation and access lists configured for inside-to-outside connections, as well as outside-to-inside connections. The initial configuration commands for the firewall are shown in Example 10-1.

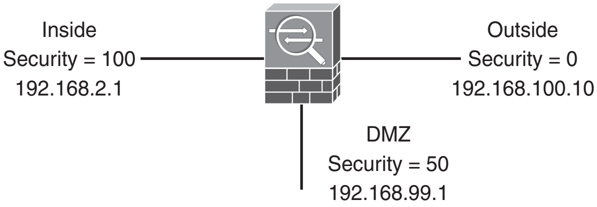
Example 10-1 Initial Firewall Configuration

```
hostname context-a
!
interface intf0
 nameif outside
 security-level 0
 ip address 192.168.100.10 255.255.255.0 standby 192.168.100.11
!
interface intf1
 nameif inside
 security-level 100
 ip address 192.168.2.1 255.255.255.0 standby 192.168.2.2
!
nat (inside) 1 192.168.2.0 255.255.255.0
global (outside) 1 interface outside
access-list acl_inside extended permit ip 192.168.2.0 255.255.255.0 any
access-group acl_inside in interface inside
!
static (inside,outside) 192.168.100.100 192.168.2.100 netmask 255.255.255.255
access-list acl_outside extended permit tcp any host 192.168.100.100 eq www
access-list acl_outside extended permit tcp any host 192.168.100.100 eq https
access-group acl_outside in interface outside
```

Video Presentation Reference

Refer to the following descriptions of each step in Lab 10. A DMZ interface is added according to the network diagram shown in Figure 10-1, with the configuration shown in Example 10-2.

Figure 10-1 Adding a DMZ Interface to a Firewall



Example 10-2 Initial DMZ Interface Configuration

```
interface intf2
   nameif dmz
   security-level 50
   ip address 192.168.99.1 255.255.255.0
   no shutdown
```

Step 1: Consider Connections from the Inside Toward the DMZ

In this step, address translation is configured across the inside and DMZ interfaces. Because the access list *acl_inside* was previously configured to permit all traffic from the inside subnet toward any destination address, no changes need to be made for traffic destined for the DMZ.

Example 10-3 shows the configuration command that is entered. The firewall's initial configuration includes a definition for **nat (inside) 3**, so only the corresponding **global** command is needed.

Example 10-3 Configuring Security Policies for Inside-to-DMZ Connections

```
Firewall(config)# global (dmz) 3 interface dmz
```

Step 2: Consider Connections from the DMZ Toward the Outside

In this step, address translation is configured across the DMZ and outside interfaces. A series of **global** commands are already present in the firewall's initial configuration, requiring only a corresponding **nat** command to be added.

The access list *acl_dmz* is created to permit any IP traffic from the DMZ subnet 192.168.99.0/24 to any destination address on the outside. Example 10-4 shows the configuration commands that are entered in this step.

Example 10-4 Configuring Security Policies for DMZ-to-Outside Connections

```
Firewall(config)# nat (dmz) 3 192.168.99.0 255.255.255.0
Firewall(config)# access-list acl_dmz extended permit ip 192.168.99.0
255.255.255.0 any
Firewall(config)# access-group acl_dmz in interface dmz
```

Step 3: Consider Connections from the Outside Toward the DMZ

In this step, address translation is configured across the outside and DMZ interfaces with a **static** command. DMZ address 192.168.99.10 is mapped to outside address 192.168.100.110.

In addition, the access list *acl_outside* is amended to include rules that permit inbound connections from any outside address to the mapped address 192.168.100.110 with destination ports TCP 80 and 443. Example 10-5 shows the configuration commands that are entered.

Example 10-5 Configuring Security Policies for Outside-to-DMZ Connections

```
Firewall(config)# static (dmz,outside) 192.168.100.110 192.168.99.10 netmask
    255.255.255.255
!
Firewall(config)# access-list acl_outside extended permit tcp any host
    192.168.100.110 eq www
Firewall(config)# access-list acl_outside extended permit tcp any host
    192.168.100.110 eq https
```

Step 4: Consider Connections from the DMZ Toward the Inside

This step shows you how to configure address translation across the DMZ and inside interfaces using a **static** command. Inside server address 192.168.2.99 is mapped to DMZ address 192.168.99.99.

Rules are added to the *acl_dmz* access list to permit inbound connections from DMZ server 192.168.99.10 to the mapped address 192.168.99.99. Only connections to destination ports TCP 1433 and FTP are permitted.

The commands shown in Example 10-6 are entered during the configuration.

Example 10-6 Configuring Security Policies for DMZ-to-Inside Connections

```
Firewall(config)# static (dmz,inside) 192.168.2.99 192.168.99.99 netmask
    255.255.255.255
!
Firewall(config)# access-list acl_dmz extended permit tcp host 192.168.99.10
    host 192.168.99.99 eq 1433
Firewall(config)# access-list acl_dmz extended permit tcp host 192.168.99.10
    host 192.168.99.99 eq ftp
```

Step 5: Review the DMZ Access List for Conflicting Entries

Access list entries were added to the *acl_dmz* access list on the DMZ interface in Steps 2 and 4. In this step, the access list is examined to make sure that it doesn’t contain conflicting or interfering entries.

In fact, the access list entries are somewhat out of order. Any traffic is permitted from the DMZ interface to any destination address, whether on the inside or outside interface. To correct this, the access list is first cleared from the configuration, and then it is reentered with the entries in the appropriate order. The commands to accomplish this are shown in Example 10-7.

Example 10-7 Commands Used to Clear and Reconfigure the *acl_outside* Access List

```
Firewall(config)# clear configure access-list acl_dmz
!
Firewall(config)# access-list acl_dmz extended permit tcp host 192.168.99.10
    host 192.168.99.99 eq 1433
Firewall(config)# access-list acl_dmz extended permit tcp host 192.168.99.10
    host 192.168.99.99 eq ftp
Firewall(config)# access-list acl_dmz extended deny ip 192.168.99.0
    255.255.255.0 192.168.2.0 255.255.255.0
Firewall(config)# access-list acl_dmz extended permit ip 192.168.99.0
    255.255.255.0 any
!
Firewall(config)# access-group acl_dmz in interface dmz
```

Setting Up Logging

This Cisco Firewall Video Mentor lab shows you how to configure a firewall to generate logging messages. Logging messages can be sent to a variety of destinations for archiving or examination.

The objectives of this lab are as follows:

- Configure the firewall to send logging messages to its internal buffer
- Configure the firewall to send logging messages to ASDM sessions
- Configure the firewall to send logging messages to an external Syslog server
- Fine-tune firewall logging

Scenario

This lab contains several scenarios, presented in the following steps:

Step 1. Send firewall logging messages to the internal logging buffer.

Step 2. Send logging messages to ASDM sessions.

Step 3. Send logging messages to one or more syslog servers for collection.

Initial Configurations

The firewall begins with a simple configuration used in previous labs. Although the lab configurations take place on the “context-a” security context, they could just as easily be configured on a firewall running in single context mode.

The firewall is configured with an inside and outside interface, with address translation and access lists configured for inside-to-outside connections, as well as outside-to-inside connections. The initial configuration commands for the firewall are shown in Example 11-1.

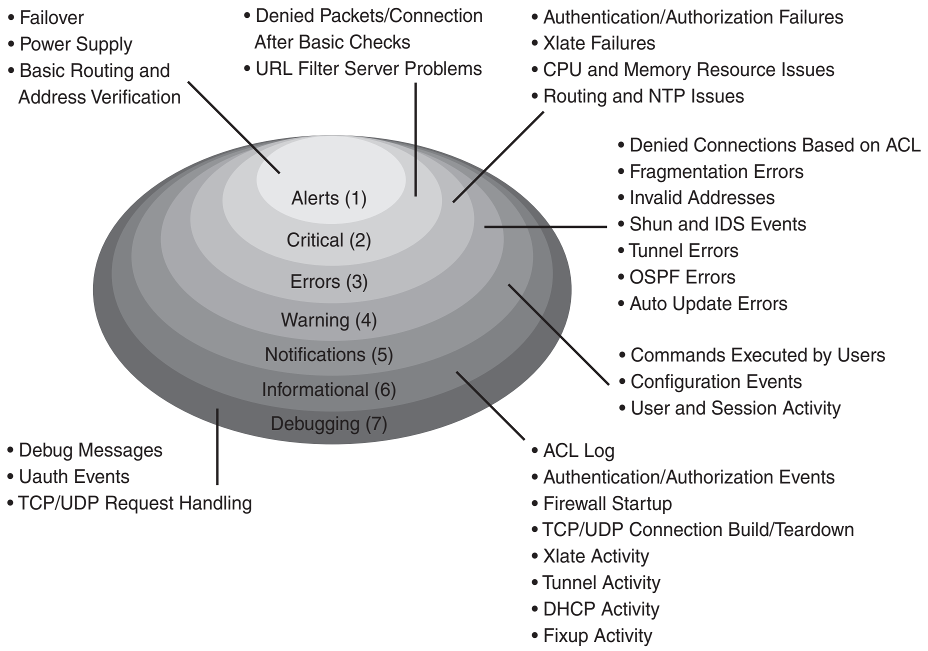
Example 11-1 Initial Firewall Configuration

```
hostname context-a
!
interface intf0
 nameif outside
 security-level 0
 ip address 192.168.100.10 255.255.255.0 standby 192.168.100.11
!
interface intf1
 nameif inside
 security-level 100
 ip address 192.168.2.1 255.255.255.0 standby 192.168.2.2
!
nat (inside) 1 192.168.2.0 255.255.255.0
global (outside) 1 interface outside
access-list acl_inside extended permit ip 192.168.2.0 255.255.255.0 any
access-group acl_inside in interface inside
!
static (inside,outside) 192.168.100.100 192.168.2.100 netmask 255.255.255.255
access-list acl_outside extended permit tcp any host 192.168.100.100 eq www
access-list acl_outside extended permit tcp any host 192.168.100.100 eq https
access-group acl_outside in interface outside
```

Video Presentation Reference

Refer to the following descriptions of each step in Lab 11. Figure 11-1 shows logging severity levels 1 through 7, along with a basic list of firewall activities that are reported at each level.

Figure 11-1 Logging Message Severity Levels



Step 1: Send Logging Messages to the Internal Buffer

In this step, the firewall is configured to send logging messages to its internal logging buffer. Example 11-2 shows the configuration commands that are entered.

Example 11-2 Logging to the Internal Logging Buffer

```
Firewall(config)# logging list test level informational class session
Firewall(config)# logging list test message 106100
Firewall(config)# logging buffered test
```

Step 2: Send Logging Messages to an ASDM Session

In this step, ASDM is used to configure the firewall to send logging messages to the ASDM session, as long as it is active and accepting logs. To configure ASDM logging, click the **Enable ASDM Logging** button on the ASDM home screen. Then choose **Configuration > Device Management > Logging** and click **Logging Filters** to configure the logging destination and source information.

To view logs as they are received, choose **Monitoring > Logging > Real Time Log Viewer**.

Step 3: Send Logging Messages to a Syslog Server

In this step, the firewall is configured to send logging messages to a syslog server located on the outside network. Example 11-3 shows the configuration commands that are entered.

Example 11-3 Logging to a Syslog Server

```
Firewall(config)# logging trap informational
Firewall(config)# logging host outside 172.21.4.172
```

Lab 12

Using MPF to Control Layer 3/4 Connections

This Cisco Firewall Video Mentor lab shows you how to use the Modular Policy Framework (MPF) to set connection limits at OSI Layers 3 and 4.

The objectives of this lab are as follows:

- Configure the firewall to classify interesting traffic that will be acted upon
- Configure a policy map that defines a specific action to take on matched traffic
- Apply the policy map to one or more firewall interfaces

Scenario

This lab steps through configuring the MPF to implement a new security policy. The policy is built in stages, with each stage presented in a different scenario:

Step 1. Classify interesting traffic with a class map.

Step 2. Define a policy with a policy map.

Step 3. Apply the policy to one or more firewall interfaces.

Step 4. Monitor the policy and its usage.

Initial Configurations

The firewall begins with a simple configuration used in previous labs. Although the lab configurations take place on the “context-a” security context, they could just as easily be configured on a firewall running in single context mode.

The firewall is configured with an inside and outside interface, with address translation and access lists configured for inside-to-outside connections, as well as outside-to-inside connections. The initial configuration commands for the firewall are shown in Example 12-1.

Example 12-1   Initial Firewall Configuration

```
hostname context-a
!
interface intf0
 nameif outside
 security-level 0
 ip address 192.168.100.10 255.255.255.0 standby 192.168.100.11
!
interface intf1
 nameif inside
 security-level 100
 ip address 192.168.2.1 255.255.255.0 standby 192.168.2.2
!
nat (inside) 1 192.168.2.0 255.255.255.0
global (outside) 1 interface outside
access-list acl_inside extended permit ip 192.168.2.0 255.255.255.0 any
access-group acl_inside in interface inside
!
static (inside,outside) 192.168.100.100 192.168.2.100 netmask 255.255.255.255
access-list acl_outside extended permit tcp any host 192.168.100.100 eq www
access-list acl_outside extended permit tcp any host 192.168.100.100 eq https
access-group acl_outside in interface outside
```

Video Presentation Reference

A Cisco firewall begins with the default MPF configuration shown in Example 12-2.

Example 12-2   Default MPF Configuration

```
class-map inspection_default
   match default-inspection-traffic
class-map class-default
   match any
policy-map global_policy
   class inspection_default
   inspect dns preset_dns_map
   inspect ftp
   inspect h2323 h225 _default_h323_map
   inspect h323 ras _default_h323_map
   inspect netbios
   inspect rsh
   inspect skinny
   inspect esmtp _default_esmtp_map
```

Example 12-2 Default MPF Configuration

```
   inspect sqlnet
   inspect sunrpc
   inspect tftp
   inspect sip
   inspect xdmcp
   class class-default
service-policy global_policy global
```

Refer to the following descriptions of each step in Lab 12.

Step 1: Classify Interesting Traffic with a Class Map

In this step, a class map is used to match and identify traffic that the policy will act upon. The class map is configured to match against an access list, classifying traffic from the 192.168.2.0/24 subnet to the 192.168.19.0/24 subnet over TCP ports 23 and 22. The configuration commands are shown in Example 12-3.

Example 12-3 Classifying Interesting Traffic

```
Firewall(config)# access-list acl_outboundsessions extended permit tcp
    192.168.2.0 255.255.255.0 192.168.19.0 255.255.255.0 eq telnet
Firewall(config)# access-list acl_outboundsessions extended permit tcp
    192.168.2.0 255.255.255.0 192.168.19.0 255.255.255.0 eq ssh
Firewall(config)# class-map  outboundsessions
Firewall(config-cmap)# description lengthy telnet or SSH sessions
Firewall(config-cmap)# match access-list acl_outboundsessions
Firewall(config-cmap)# exit
```

Step 2: Define a Policy with a Policy Map

In this step, a policy map is configured to define a matching condition and an action—constituting a security policy. The class map named *outboundsessions* is referenced to match “interesting” traffic. A TCP connection idle timeout of 12 hours is applied to the matched traffic. In addition, the Dead Connection Detection (DCD) feature is used to detect connections that are still alive; in that case, they are not closed even if they have been idle too long. Example 12-4 shows the configuration commands that are used.

Example 12-4 Defining a Policy

```
Firewall(config)# policy-map outbound
Firewall(config-pmap)# class outboundsessions
Firewall(config-pmap-c)# set connection timeout tcp 12:00:00 reset dcd
Firewall(config-pmap-c)# exit
Firewall(config-pmap)# exit
```

Step 3: Apply the Policy

In this step, the policy map named *outbound* is applied to the firewall's inside interface as part of a service policy. Example 12-5 shows the configuration command that is entered.

Example 12-5 Applying a Policy Map to a Firewall Interface

```
Firewall(config)# service-policy outbound interface inside
```

Step 4: Monitor the Policy

Finally, the firewall policy is monitored using the commands shown in Example 12-6.

Example 12-6 Monitoring a Service Policy

```
Firewall# show service-policy
Firewall# show service-policy set connection
```

Using MPF to Perform QoS Queuing and Policing

This Cisco Firewall Video Mentor lab shows you how to use the Modular Policy Framework (MPF) to control how the firewall forwards packets after they have been inspected and classified. The firewall can specify priority handling to deliver certain traffic ahead of other packets. As well, it can limit the bandwidth of specific traffic according to a policer configuration.

The objectives of this lab are as follows:

- Use the MPF to configure a policy to place specific traffic in a priority queue
- Use the MPF to configure a traffic policer to control bandwidth used by matched traffic

Scenario

This lab shows you how to configure security policies within the MPF to accomplish the following steps:

Step 1. Classify specific traffic to be placed in a priority queue.

Step 2. Implement a traffic policer to limit bandwidth for a class of traffic.

Initial Configurations

The firewall begins with a simple configuration used in previous labs, applied to the “context-a” security context in multiple security context mode. However, priority queuing is not available in multiple context mode. In this lab, the same configuration is applied to the firewall running in single context mode.

The firewall is configured with an inside and outside interface, with address translation and access lists configured for inside-to-outside connections, as well as outside-to-inside connections. The MPF configuration added in Lab 12 is also present here.

The initial configuration commands for the firewall are shown in Example 13-1.

Example 13-1 Initial Firewall Configuration

```
hostname context-a
!
interface intf0
 nameif outside
 security-level 0
 ip address 192.168.100.10 255.255.255.0 standby 192.168.100.11
!
interface intf1
 nameif inside
 security-level 100
 ip address 192.168.2.1 255.255.255.0 standby 192.168.2.2
!
nat (inside) 1 192.168.2.0 255.255.255.0
global (outside) 1 interface outside
access-list acl_inside extended permit ip 192.168.2.0 255.255.255.0 any
access-group acl_inside in interface inside
!
static (inside,outside) 192.168.100.100 192.168.2.100 netmask 255.255.255.255
access-list acl_outside extended permit tcp any host 192.168.100.100 eq www
access-list acl_outside extended permit tcp any host 192.168.100.100 eq https
access-group acl_outside in interface outside
!
access-list acl_outboundsessions extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq telnet
access-list acl_outboundsessions extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq ssh
!
class-map  outboundsessions
   description lengthy telnet or SSH sessions
   match access-list acl_outboundsessions
policy-map outbound
   class outboundsessions
   set connection timeout tcp 12:00:00 reset dcd
!
service-policy outbound interface inside
```

Video Presentation Reference

Refer to the following descriptions of each step in Lab 13.

Step 1: Send Traffic to the Priority Queue

In this step, the priority queue is enabled on the firewall's outside interface. A class map is used to identify packets that have a DSCP value of EF already set in the ToS byte of the IP header. The matched traffic is placed in the priority queue as it is forwarded toward the outside interface.

The configuration commands are shown in Example 13-2.

Example 13-2 Using a Priority Queue

```
Firewall(config)# priority-queue outside
Firewall(config-priority-queue)# queue-limit 2048
Firewall(config-priority-queue)# exit
Firewall(config)# class-map  voice_bearer
Firewall(config-cmap)# description LLQ policy
Firewall(config-cmap)# match dscp ef
Firewall(config-cmap)# exit
Firewall(config)# policy-map outside_policies
Firewall(config-pmap)# description Policies for the outside interface
Firewall(config-pmap)# class voice_bearer
Firewall(config-pmap-c)# priority
Firewall(config-pmap-c)# exit
Firewall(config-pmap)# exit
Firewall(config)# service-policy outside_policies interface outside
```

Step 2: Use a Policer to Limit Bandwidth

In this step, a class map named **test-class** is configured to match against packets coming from outside host 128.163.111.7 inbound toward any address on the inside network.

The matched traffic is applied to a bandwidth policer in the policy map named **outsidepolicies** so that it is limited to using no more than 100,000 bits per second on the outside interface. The matched traffic is allowed to burst up to 3000 bytes over the conform rate limit. Example 13-3 shows the configuration commands that are used.

Example 13-3 Defining a Policer

```
Firewall(config)# access-list acl_test extended permit ip host 128.163.111.7 any
!
Firewall(config)# class-map test-class
Firewall(config-cmap)# match access-list acl_test
Firewall(config-cmap)# exit
!
Firewall(config)# policy-map outsidepolicies
Firewall(config-pmap)# class test-class
Firewall(config-pmap-c)# police input conform_rate 100000 3000 conform-action
    transmit exceed-action drop
Firewall(config-pmap-c)# exit
Firewall(config-pmap)# exit
```

Lab 14

Using MPF to Tune Application Inspection Engines

This Cisco Firewall Video Mentor lab shows you how to use the Modular Policy Framework (MPF) to change how packets are inspected when they are applied to an application inspection engine.

The objectives of this lab are as follows:

- Examine the default application inspection engines
- Enable and disable an application inspection engine
- Configure an inspection engine policy map
- Configure an inspection engine regular expression (regex) match

Scenario

This lab shows you how to configure security policies within the MPF to accomplish the following steps:

Step 1. Enable or disable an application inspection engine.

Step 2. Change the port where an inspection engine listens.

Step 3. Tune an inspection engine with a special-purpose inspection policy map.

Step 4. Configure an inspection engine to match against a regular expression through the use of a regex match.

Initial Configurations

The firewall begins with a simple configuration used in previous labs, as shown in Example 14-1.

The firewall is configured with an inside and outside interface, with address translation and access lists configured for inside-to-outside connections, as well as outside-to-inside connections. The MPF configurations from Labs 12 and 13 are present here.

Example 14-1 Initial Firewall Configuration

```
hostname context-a
!
interface intf0
 nameif outside
 security-level 0
 ip address 192.168.100.10 255.255.255.0 standby 192.168.100.11
!
interface intf1
 nameif inside
 security-level 100
 ip address 192.168.2.1 255.255.255.0 standby 192.168.2.2
!
nat (inside) 1 192.168.2.0 255.255.255.0
global (outside) 1 interface outside
access-list acl_inside extended permit ip 192.168.2.0 255.255.255.0 any
access-group acl_inside in interface inside
!
static (inside,outside) 192.168.100.100 192.168.2.100 netmask 255.255.255.255
access-list acl_outside extended permit tcp any host 192.168.100.100 eq www
access-list acl_outside extended permit tcp any host 192.168.100.100 eq https
access-group acl_outside in interface outside
!
access-list acl_outboundsessions extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq telnet
access-list acl_outboundsessions extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq ssh
!
class-map  outboundsessions
   description lengthy telnet or SSH sessions
   match access-list acl_outboundsessions
policy-map outbound
   class outboundsessions
   set connection timeout tcp 12:00:00 reset dcd
!
service-policy outbound interface inside
```

Video Presentation Reference

Refer to the following descriptions of each step in Lab 14.

Step 1: Enable an Inspection Engine

In this step, the **global_policy** policy map is edited so that the default inspection engines also include ICMP and ICMP error inspection. In addition, the SIP inspection engine (enabled by default) is disabled.

The configuration commands are shown in Example 14-2.

Example 14-2 Enabling and Disabling Inspection Engines

```
Firewall(config)# policy-map global_policy
Firewall(config-pmap)# class inspection_default
Firewall(config-pmap-c)# inspect icmp
Firewall(config-pmap-c)# inspect icmp error
Firewall(config-pmap-c)# no inspect sip
Firewall(config-pmap-c)# exit
Firewall(config-pmap)# exit
```

Step 2: Change the Inspection Engine Listening Port

By default, every application inspection engine listens to one or more default port numbers to identify traffic to be inspected. In this step, packets using destination port TCP 8080 are sent to the HTTP inspection engine, along with packets using destination port TCP 80.

A class map named **http_8080** is configured to match against destination port TCP 8080, and class map **http_80** matches against TCP port 80. Two new policies are added to the **outside_policies** policy map to funnel matched traffic into the HTTP inspection engine. Example 14-3 shows the configuration commands that are entered.

Example 14-3 Changing the Default Application Inspection Engine Port

```
Firewall(config)# class-map http_8080
Firewall(config-cmap)# match port tcp eq 8080
Firewall(config-cmap)# exit
!
Firewall(config)# class-map http_80
Firewall(config-cmap)# match port tcp eq 80
Firewall(config-cmap)# exit
!
Firewall(config)# policy-map outside_policies
Firewall(config-pmap)# class http_8080
Firewall(config-pmap-c)# inspect http
Firewall(config-pmap-c)# exit
Firewall(config-pmap)# class http_80
Firewall(config-pmap-c)# inspect http
Firewall(config-pmap-c)# exit
Firewall(config-pmap)# exit
```

Step 3: Use an Inspection Policy Map

In this step, the Instant Messenger (IM) inspection engine is configured to use an inspection policy map, as shown by the configuration commands listed in Example 14-4. Through the inspection policy map, IM traffic can be matched according to specific behavior, and an action can be applied to it.

An IM inspection policy map named **IMpolicy** is configured to match against requests to perform a file transfer. File transfers are allowed to occur by default, but the policy map is configured to log the file transfer requests. In previous labs, a service policy was configured to apply the **outside_policies** policy map to the outside interface. Therefore, there is no need to enter the **service-policy outside_policies interface outside** command again in this lab.

Example 14-4 Configuring an IM Inspection Engine Policy Map

```
Firewall(config)# policy-map type inspect im IMpolicy
Firewall(config-pmap)# match service file-transfer
Firewall(config-pmap)# log
Firewall(config-pmap)# exit
Firewall(config)# exit
Firewall(config-pmap)# policy-map outside_policies
Firewall(config-pmap)# class class-default
Firewall(config-pmap-c)# inspect im IMpolicy
Firewall(config-pmap-c)# exit
Firewall(config-pmap)# exit
```

Step 4: Use an Inspection Regex Match

In this step, the IM inspection engine is further configured to inspect the filenames found in file transfer requests. Filenames ending with .exe or .zip are not allowed to be transferred, so those requests are simply reset. All other file transfers are permitted, according to the configuration in Step 3 of this lab.

The configuration commands shown in Example 14-5 are entered in this step of the lab.

Example 14-5 Using an Inspection Regex Match with the IM Inspection Engine

```
Firewall(config)# regex exe .*\.exe
Firewall(config)# regex zip .*\.zip
Firewall(config)# class-map type regex match-any IMregex
Firewall(config-cmap)# match regex  exe
Firewall(config-cmap)# match regex  zip
Firewall(config-cmap)# exit
!
Firewall(config)# policy-map type inspect im IMpolicy
Firewall(config-pmap)# match filename regex class IMregex
Firewall(config-pmap)# reset
Firewall(config-pmap)# exit
Firewall(config)# exit
```

To test the regex operation, enter the following command:

```
Firewall# test regex "install.exe" ".*\.exe"
```

Lab 15

Testing Security Policies with Packet Tracer

This Cisco Firewall Video Mentor lab shows you how to use the Packet Tracer tool within ASDM to model and test packet movement through the firewall. After you define a test packet type, as well as source and destination addresses, Packet Tracer simulates sending the packet from one interface to another. Each security feature within the firewall is shown graphically, along with the results of the packet's movement within the feature.

Scenario

This lab shows you how to use Packet Tracer to model security policy behavior in the following steps:

Step 1. Send an outbound ICMP echo packet using a static address translation.

Step 2. Send an outbound ICMP echo packet using a dynamic address translation.

Step 3. Send an inbound ICMP echo reply packet.

Step 4. Send an outbound TCP packet through an application inspection engine.

Initial Configurations

The firewall begins with a configuration that evolved in previous labs, as shown in Example 15-1.

The firewall is configured with an inside and outside interface, with address translation and access lists configured for inside-to-outside connections, as well as outside-to-inside connections.

Example 15-1 Initial Firewall Configuration

```
hostname asa1
domain-name mycompany.com
enable password iE9elCMOvCJAfUw3 encrypted
names
name 192.168.100.199 vpn_gateway
http server enable
http 0.0.0.0 0.0.0.0 outside
!
interface GigabitEthernet1/0
 description outside - all contexts
 nameif outside
 security-level 0
 ip address 192.168.100.10 255.255.255.0
```

continues

Example 15-1 Initial Firewall Configuration *continued*

```
!
interface GigabitEthernet1/1
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet1/1.1
 description context-a inside
 vlan 100
 nameif inside
 security-level 100
 ip address 192.168.2.1 255.255.255.0
!
global (outside) 1 192.168.100.203
global (outside) 3 192.168.100.20-192.168.100.30 netmask 255.255.255.0
global (outside) 3 192.168.100.31
global (outside) 3 192.168.100.32
global (outside) 3 interface
!
nat (inside) 0 access-list natexempt
nat (inside) 0 192.168.2.128 255.255.255.128
nat (inside) 1 access-list policynat3
nat (inside) 3 192.168.2.0 255.255.255.0 tcp 200 50  udp 100
!
static (inside,outside) tcp 192.168.100.101 www 192.168.2.90 www netmask
    255.255.255.255
static (inside,outside) tcp 192.168.100.101 smtp 192.168.2.91 smtp netmask
    255.255.255.255
static (inside,outside) 192.168.100.100 192.168.2.100 netmask 255.255.255.255
static (inside,outside) 192.168.100.102  access-list policynat1
static (inside,outside) 192.168.100.202  access-list policynat2
static (inside,outside) 192.168.100.99 192.168.2.99 netmask 255.255.255.255
!
object-group network server_group1
 description servers
 network-object host 192.168.100.100
 network-object host 192.168.100.101
!
object-group service service_group1
 description server ports
 service-object tcp eq www
 service-object tcp eq https
!
object-group service service_vpn
```

Example 15-1 Initial Firewall Configuration *continued*

```
 description vpn gateway services
 service-object gre
 service-object esp
 service-object tcp eq 10000
 service-object udp range 10000 10001
 service-object udp eq 4500
 service-object udp eq isakmp
!
access-list inside_mpc extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq telnet
access-list inside_mpc extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq ssh log disable
access-list outside_mpc extended permit ip host 128.163.111.7 any
access-list acl_inside extended permit ip 192.168.2.0 255.255.255.0 any
access-list acl_outside extended permit object-group service_group1 any
    object-group server_group1
access-list acl_outside extended permit object-group service_vpn any host
    vpn_gateway log disable
access-list acl_outside extended permit tcp any host 192.168.100.110 eq www
access-list acl_outside extended permit tcp any host 192.168.100.110 eq https
access-list policynat1 extended permit ip host 192.168.2.102 172.16.0.0
    255.255.0.0
access-list policynat2 extended permit ip host 192.168.2.102 172.17.0.0
    255.255.0.0
access-list policynat3 extended permit ip host 192.168.2.102 172.18.0.0
    255.255.0.0
access-list natexempt extended permit ip 192.168.3.0 255.255.255.0 any
access-list natexempt extended permit ip 192.168.4.0 255.255.255.0 any
!
access-list acl_outboundsessions extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq telnet
access-list acl_outboundsessions extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq ssh
!
access-group acl_outside in interface outside
access-group acl_inside in interface inside
!
route outside 0.0.0.0 0.0.0.0 192.168.100.3 1
!
class-map http_8080
 match port tcp eq 8080
class-map type regex match-any IMregex
 match regex zip
```

continues

Example 15-1 Initial Firewall Configuration *continued*

```
 match regex exe
class-map http_80
 match port tcp eq www
class-map outboundsessions
 match access-list inside_mpc
class-map test-class
 match access-list outside_mpc
class-map inspection_default
 match default-inspection-traffic
class-map voice_bearer
 match dscp ef
class-map default
!
!
policy-map global_policy
 class inspection_default
  inspect dns migrated_dns_map_1
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect xdmcp
  inspect icmp
  inspect icmp error
policy-map outside_policies
 class voice_bearer
  priority
 class test-class
  police input 100000 3000
 class http_80
  inspect http
 class default
  inspect im IMpolicy
 class http_8080
  inspect http
 class class-default
policy-map inside-policy
```

Example 15-1 Initial Firewall Configuration *continued*

```
 description Control outbound sessions
 class outboundsessions
  set connection timeout tcp 0:05:00 reset dcd 0:15:00 5
!
service-policy global_policy global
service-policy outside_policies interface outside
service-policy inside-policy interface inside
```

Video Presentation Reference

Refer to the following descriptions for information about each of the packet types that are sent through Packet Tracer in Lab 15.

Step 1: Send an Outbound ICMP Packet Through a Static Address Translation

In this step, Packet Tracer is configured to send an ICMP echo packet from the inside toward the outside so that a static address translation is used. The following packet parameters are given:

- Ingress interface: **inside**
- Source address: **192.168.2.99**
- Destination address: **192.168.100.250**
- Packet type: **ICMP echo**, **code 0**, **ID 1024**

The resulting packet is forwarded on the outside interface. Packet Tracer shows how the address translation occurs.

Step 2: Send an Outbound ICMP Packet Through a Dynamic Address Translation

In this step, Packet Tracer is configured to send an ICMP echo packet from the inside toward the outside so that a dynamic address translation is used. The following packet parameters are given:

- Ingress interface: **inside**
- Source address: **192.168.2.50**
- Destination address: **192.168.100.250**
- Packet type: **ICMP echo**, **code 0**, **ID 1024**

The resulting packet is forwarded on the outside interface. Packet Tracer shows how the dynamic address translation is triggered.

Step 3: Send an Inbound ICMP Packet

In this step, Packet Tracer is configured to send an ICMP echo reply packet from the outside interface toward the inside. The following packet parameters are used:

- Ingress interface: **inside**
- Source address: **192.168.2.50**
- Destination address: **192.168.100.99**
- Packet type: **ICMP echo-reply**, **code 0**, **ID 1024**

The resulting packet arrives on the outside interface, but it is dropped after being denied by an access list entry. Packet Tracer shows where the packet is dropped and explains why.

Step 4: Send an Outbound TCP Packet Through an Application Inspection Engine

In this step, Packet Tracer is configured to send a TCP packet from the inside interface toward the outside. TCP destination port 8080 is used to model the firewall's MPF inspection engine configuration. The following packet parameters are used:

- Ingress interface: **inside**
- Source address: **192.168.2.50**
- Destination address: **198.133.219.25**
- Packet type: **TCP**, source port **43100**, destination port **8080**

The resulting packet arrives on the outside interface, where it would be forwarded to the destination address. Packet Tracer verifies that the packet is sent to the HTTP application inspection engine, just as the firewall is configured to do.

Lab 16

Capturing Traffic

This Cisco Firewall Video Mentor lab shows you how to configure a firewall to capture packets on one or more of its interfaces. Packets can be captured into a memory buffer and displayed or copied to an external host for examination.

The objectives of this lab are as follows:

- Configure a packet-capture session using the CLI
- Use the ASDM Packet Capture Wizard to capture packets

Scenario

This lab shows you how to configure packet capture in the following steps:

Step 1. Use the CLI to capture packets as raw data.

Step 2. Use the CLI to capture traffic that the firewall dropped.

Step 3. Copy the capture buffer to an external host.

Step 4. Use the ASDM Packet Capture wizard to capture traffic.

Initial Configurations

The firewall begins with a configuration that evolved in previous labs, as shown in Example 16-1.

The firewall is configured with an inside and outside interface, with address translation and access lists configured for inside-to-outside connections, as well as outside-to-inside connections.

Example 16-1 Initial Firewall Configuration

```
hostname asa1
domain-name mycompany.com
enable password iE9elCMOvCJAfUw3 encrypted
names
name 192.168.100.199 vpn_gateway
http server enable
http 0.0.0.0 0.0.0.0 outside
!
interface GigabitEthernet1/0
 description outside - all contexts
 nameif outside
 security-level 0
```

continues

Example 16-1 Initial Firewall Configuration *continued*

```
 ip address 192.168.100.10 255.255.255.0
!
interface GigabitEthernet1/1
 no nameif
 no security-level
 no ip address
!
interface GigabitEthernet1/1.1
 description context-a inside
 vlan 100
 nameif inside
 security-level 100
 ip address 192.168.2.1 255.255.255.0
!
global (outside) 1 192.168.100.203
global (outside) 3 192.168.100.20-192.168.100.30 netmask 255.255.255.0
global (outside) 3 192.168.100.31
global (outside) 3 192.168.100.32
global (outside) 3 interface
!
nat (inside) 0 access-list natexempt
nat (inside) 0 192.168.2.128 255.255.255.128
nat (inside) 1 access-list policynat3
nat (inside) 3 192.168.2.0 255.255.255.0 tcp 200 50  udp 100
!
static (inside,outside) tcp 192.168.100.101 www 192.168.2.90 www netmask
    255.255.255.255
static (inside,outside) tcp 192.168.100.101 smtp 192.168.2.91 smtp netmask
    255.255.255.255
static (inside,outside) 192.168.100.100 192.168.2.100 netmask 255.255.255.255
static (inside,outside) 192.168.100.102  access-list policynat1
static (inside,outside) 192.168.100.202  access-list policynat2
static (inside,outside) 192.168.100.99 192.168.2.99 netmask 255.255.255.255
!
object-group network server_group1
 description servers
 network-object host 192.168.100.100
 network-object host 192.168.100.101
!
object-group service service_group1
 description server ports
 service-object tcp eq www
 service-object tcp eq https
```

Example 16-1 Initial Firewall Configuration *continued*

```
!
object-group service service_vpn
 description vpn gateway services
 service-object gre
 service-object esp
 service-object tcp eq 10000
 service-object udp range 10000 10001
 service-object udp eq 4500
 service-object udp eq isakmp
!
access-list inside_mpc extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq telnet
access-list inside_mpc extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq ssh log disable
access-list outside_mpc extended permit ip host 128.163.111.7 any
access-list acl_inside extended permit ip 192.168.2.0 255.255.255.0 any
access-list acl_outside extended permit object-group service_group1
    any object-group server_group1
access-list acl_outside extended permit object-group service_vpn any host
    vpn_gateway log disable
access-list acl_outside extended permit tcp any host 192.168.100.110 eq www
access-list acl_outside extended permit tcp any host 192.168.100.110 eq https
access-list policynat1 extended permit ip host 192.168.2.102 172.16.0.0
    255.255.0.0
access-list policynat2 extended permit ip host 192.168.2.102 172.17.0.0
    255.255.0.0
access-list policynat3 extended permit ip host 192.168.2.102 172.18.0.0
    255.255.0.0
access-list natexempt extended permit ip 192.168.3.0 255.255.255.0 any
access-list natexempt extended permit ip 192.168.4.0 255.255.255.0 any
!
access-list acl_outboundsessions extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq telnet
access-list acl_outboundsessions extended permit tcp 192.168.2.0 255.255.255.0
    192.168.19.0 255.255.255.0 eq ssh
!
access-group acl_outside in interface outside
access-group acl_inside in interface inside
!
route outside 0.0.0.0 0.0.0.0 192.168.100.3 1
!
class-map http_8080
 match port tcp eq 8080
```

continues

Example 16-1 Initial Firewall Configuration *continued*

```
class-map type regex match-any IMregex
 match regex zip
 match regex exe
class-map http_80
 match port tcp eq www
class-map outboundsessions
 match access-list inside_mpc
class-map test-class
 match access-list outside_mpc
class-map inspection_default
 match default-inspection-traffic
class-map voice_bearer
 match dscp ef
class-map default
!
!
policy-map global_policy
 class inspection_default
  inspect dns migrated_dns_map_1
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect xdmcp
  inspect icmp
  inspect icmp error
policy-map outside_policies
 class voice_bearer
  priority
 class test-class
  police input 100000 3000
 class http_80
  inspect http
 class default
  inspect im IMpolicy
 class http_8080
  inspect http
```

Example 16-1 Initial Firewall Configuration *continued*

```
 class class-default
policy-map inside-policy
 description Control outbound sessions
 class outboundsessions
  set connection timeout tcp 0:05:00 reset dcd 0:15:00 5
!
service-policy global_policy global
service-policy outside_policies interface outside
service-policy inside-policy interface inside
```

Video Presentation Reference

Refer to the following descriptions for information about each of the packet captures that are configured in Lab 16.

Step 1: Configure a Raw Data Capture Through the CLI

In this step, the CLI is used to configure a packet capture. Traffic going to or from the inside host 192.168.2.39 is captured, but only on the inside interface. Example 16-2 shows the configuration commands that are entered.

Example 16-2 Configuring a Packet-Capture Session with the CLI

```
Firewall# configure terminal
Firewall(config)# access-list cap_2.39 extended permit ip host 192.168.2.39 any
Firewall(config)# access-list cap_2.39 extended permit ip any host 192.168.2.39
Firewall(config)# exit

Firewall# capture test access-list cap_2.39 interface inside
```

The capture buffer is displayed with the **show capture** *name* command.

Step 2: Configure an ASP Drop Capture Through the CLI

In this step, the CLI is used to configure a capture session that collects only packets that have been dropped by an access list security policy. The ASP drop reason **acl-drop** is used. Example 16-3 shows the command used to configure the capture session.

Example 16-3 Capturing Dropped Packets

```
Firewall# capture drop type asp-drop acl-drop
```

Step 3: Copy the Capture Buffer to an External Host

In this step, the buffer from an active capture session is copied to an external location. Example 16-4 shows the command used to copy the capture buffer to a TFTP server at 172.21.4.174.

Example 16-4 Copying the Capture Buffer to a TFTP Server

```
Firewall# copy capture:test tftp://172.21.4.174
```

Next, the capture buffer is copied in the PCAP format so that it can be imported into a network protocol analyzer. The command used is shown in Example 16-5.

Example 16-5 Copying the Capture Buffer in PCAP Format

```
Firewall# copy /pcap capture:test tftp://172.21.4.174
```

To retrieve the capture buffer in PCAP format through a web browser, the following URL is used:

https://192.168.100.10/capture/test/pcap

Step 4: Configure a Capture Session with ASDM

In this step, a capture session is created using the Packet Capture Wizard in ASDM. The following parameters are used:

- Ingress interface: **inside**
- Source host: **192.168.2.39 255.255.255.255**
- Destination host: **0.0.0.0 0.0.0.0**
- Protocol: **IP**
- Egress interface: **outside**
- Packet size: **1522** (the default)
- Buffer size: **524288** (the default)