Cisco ASA Series Firewall CLI Configuration Guide

Software Version 9.1

For the ASA 5505, ASA 5510, ASA 5520, ASA 5540, ASA 5550, ASA 5512-X, ASA 5515-X, ASA 5525-X, ASA 5545-X, ASA 5555-X, ASA 5580, ASA 5585-X, and the ASA Services Module

Released: December 3, 2012
Updated: March 31, 2014

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide.
Addresses, phone numbers, and fax numbers
are listed on the Cisco website at
www.cisco.com/go/offices.

Text Part Number: N/A, Online only

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

Cisco ASA Series Firewall CLI Configuration Guide

CONTENTS

PART 5

Configuring Unified Communications

CHAPTER 14

CHAPTER 15

About This Guide

This preface introduces *Cisco ASA Series Firewall CLI Configuration Guide* and includes the following sections:

Document Objectives

The purpose of this guide is to help you configure the firewall features for ASA using the command-line interface. This guide does not cover every feature, but describes only the most common configuration scenarios.

You can also configure and monitor the ASA by using ASDM, a web-based GUI application. ASDM includes configuration wizards to guide you through some common configuration scenarios, and online help for less common scenarios.

This guide applies to the Cisco ASA series. Throughout this guide, the term “ASA” applies generically to supported models, unless specified otherwise.

Related Documentation

For more information, see *Navigating the Cisco ASA Series Documentation* at http://www.cisco.com/go/asadocs.

Conventions

This document uses the following conventions:

Convention	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.

italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`courier` font	Terminal sessions and information the system displays appear in `courier` font.
`courier bold` font	Commands and keywords and user-entered text appear in **`bold courier`** font.
`courier italic` font	Arguments for which you supply values are in *`courier italic`* font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means *reader take note*.

Tip Means *the following information will help you solve a problem*.

Caution Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation*.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. The RSS feeds are a free service.

PART 1

Configuring Service Policies Using the Modular Policy Framework

CHAPTER 1

Configuring a Service Policy Using the Modular Policy Framework

Service policies using Modular Policy Framework provide a consistent and flexible way to configure ASA features. For example, you can use a service policy to create a timeout configuration that is specific to a particular TCP application, as opposed to one that applies to all TCP applications. A service policy consists of multiple actionsapplied to an interface or applied globally.

This chapter includes the following sections:

Information About Service Policies

This section describes how service policies work and includes the following topics:

Supported Features

Table 1-1 lists the features supported by Modular Policy Framework.

Table 1-1 Modular Policy Framework

Feature	For Through Traffic?	For Management Traffic?	See:
Application inspection (multiple types)	**All** except RADIUS accounting	**RADIUS accounting** only	• Chapter 9, "Getting Started with Application Layer Protocol Inspection." • Chapter 10, "Configuring Inspection of Basic Internet Protocols." • Chapter 11, "Configuring Inspection for Voice and Video Protocols." • Chapter 12, "Configuring Inspection of Database and Directory Protocols." • Chapter 13, "Configuring Inspection for Management Application Protocols." • Chapter 25, "Configuring the ASA for Cisco Cloud Web Security."
ASA CSC	**Yes**	No	Chapter 32, "Configuring the ASA CSC Module."
ASA IPS	**Yes**	No	Chapter 31, "Configuring the ASA IPS Module."
ASA CX	**Yes**	No	Chapter 30, "Configuring the ASA CX Module."
NetFlow Secure Event Logging filtering	**Yes**	**Yes**	See the general operations configuration guide.
QoS input and output policing	**Yes**	No	Chapter 23, "Configuring QoS."
QoS standard priority queue	**Yes**	No	Chapter 23, "Configuring QoS."
QoS traffic shaping, hierarchical priority queue	**Yes**	**Yes**	Chapter 23, "Configuring QoS."
TCP and UDP connection limits and timeouts, and TCP sequence number randomization	**Yes**	**Yes**	Chapter 22, "Configuring Connection Settings."
TCP normalization	**Yes**	No	Chapter 22, "Configuring Connection Settings."
TCP state bypass	**Yes**	No	Chapter 22, "Configuring Connection Settings."
User statistics for Identity Firewall	**Yes**	**Yes**	See the **user-statistics** command in the command reference.

Feature Directionality

Actions are applied to traffic bidirectionally or unidirectionally depending on the feature. For features that are applied bidirectionally, all traffic that enters or exits the interface to which you apply the policy map is affected if the traffic matches the class map for both directions.

Note When you use a global policy, all features are unidirectional; features that are normally bidirectional when applied to a single interface only apply to the ingress of each interface when applied globally. Because the policy is applied to all interfaces, the policy will be applied in both directions so bidirectionality in this case is redundant.

For features that are applied unidirectionally, for example QoS priority queue, only traffic that enters (or exits, depending on the feature) the interface to which you apply the policy map is affected. See Table 1-2 for the directionality of each feature.

Table 1-2 Feature Directionality

Feature	Single Interface Direction	Global Direction
Application inspection (multiple types)	Bidirectional	Ingress
ASA CSC	Bidirectional	Ingress
ASA CX	Bidirectional	Ingress
ASA CX authentication proxy	Ingress	Ingress
ASA IPS	Bidirectional	Ingress
NetFlow Secure Event Logging filtering	N/A	Ingress
QoS input policing	Ingress	Ingress
QoS output policing	Egress	Egress
QoS standard priority queue	Egress	Egress
QoS traffic shaping, hierarchical priority queue	Egress	Egress
TCP and UDP connection limits and timeouts, and TCP sequence number randomization	Bidirectional	Ingress
TCP normalization	Bidirectional	Ingress
TCP state bypass	Bidirectional	Ingress
User statistics for Identity Firewall	Bidirectional	Ingress

Feature Matching Within a Service Policy

See the following information for how a packet matches class maps in a policy map for a given interface:

1. A packet can match only one class map in the policy map for each feature type.
2. When the packet matches a class map for a feature type, the ASA does not attempt to match it to any subsequent class maps for that feature type.
3. If the packet matches a subsequent class map for a different feature type, however, then the ASA also applies the actions for the subsequent class map, if supported. See the "Incompatibility of Certain Feature Actions" section on page 1-5 for more information about unsupported combinations.

   Note Application inspection includes multiple inspection types, and most are mutually exclusive. For inspections that can be combined, each inspection is considered to be a separate feature.

For example, if a packet matches a class map for connection limits, and also matches a class map for an application inspection, then both actions are applied.

If a packet matches a class map for HTTP inspection, but also matches another class map that includes HTTP inspection, then the second class map actions are not applied.

If a packet matches a class map for HTTP inspection, but also matches another class map that includes FTP inspection, then the second class map actions are not applied because HTTP and FTP inspections cannpt be combined.

If a packet matches a class map for HTTP inspection, but also matches another class map that includes IPv6 inspection, then both actions are applied because the IPv6 inspection can be combined with any other type of inspection.

Order in Which Multiple Feature Actions are Applied

The order in which different types of actions in a policy map are performed is independent of the order in which the actions appear in the policy map.

Note NetFlow Secure Event Logging filtering and User statistics for Identity Firewall are order-independent.

Actions are performed in the following order:

1. QoS input policing
2. TCP normalization, TCP and UDP connection limits and timeouts, TCP sequence number randomization, and TCP state bypass.

   Note When a the ASA performs a proxy service (such as AAA or CSC) or it modifies the TCP payload (such as FTP inspection), the TCP normalizer acts in dual mode, where it is applied before and after the proxy or payload modifying service.

3. ASA CSC
4. Application inspections that can be combined with other inspections:
   a. IPv6
   b. IP options
   c. WAAS
5. Application inspections that cannot be combined with other inspections. See the "Incompatibility of Certain Feature Actions" section on page 1-5 for more information.
6. ASA IPS
7. ASA CX
8. QoS output policing
9. QoS standard priority queue
10. QoS traffic shaping, hierarchical priority queue

Incompatibility of Certain Feature Actions

Some features are not compatible with each other for the same traffic. The following list may not include all incompatibilities; for information about compatibility of each feature, see the chapter or section for your feature:

- You cannot configure QoS priority queueing and QoS policing for the same set of traffic.
- Most inspections should not be combined with another inspection, so the ASA only applies one inspection if you configure multiple inspections for the same traffic. HTTP inspection can be combined with the Cloud Web Security inspection. Other exceptions are listed in the "Order in Which Multiple Feature Actions are Applied" section on page 1-4.
- You cannot configure traffic to be sent to multiple modules, such as the ASA CX and ASA IPS.
- HTTP inspection is not compatible with the ASA CX.
- The ASA CX is not compatible with Cloud Web Security.

Note The **match default-inspection-traffic** command, which is used in the default global policy, is a special CLI shortcut to match the default ports for all inspections. When used in a policy map, this class map ensures that the correct inspection is applied to each packet, based on the destination port of the traffic. For example, when UDP traffic for port 69 reaches the ASA, then the ASA applies the TFTP inspection; when TCP traffic for port 21 arrives, then the ASA applies the FTP inspection. So in this case only, you can configure multiple inspections for the same class map. Normally, the ASA does not use the port number to determine which inspection to apply, thus giving you the flexibility to apply inspections to non-standard ports, for example.

This traffic class does not include the default ports for Cloud Web Security inspection (80 and 443).

An example of a misconfiguration is if you configure multiple inspections in the same policy map and do not use the default-inspection-traffic shortcut. In Example 1-1, traffic destined to port 21 is mistakenly configured for both FTP and HTTP inspection. In Example 1-2, traffic destined to port 80 is mistakenly configured for both FTP and HTTP inspection. In both cases of misconfiguration examples, only the FTP inspection is applied, because FTP comes before HTTP in the order of inspections applied.

Example 1-1 Misconfiguration for FTP packets: HTTP Inspection Also Configured

```
class-map ftp
   match port tcp eq 21
class-map http
   match port tcp eq 21 [it should be 80]
policy-map test
   class ftp
     inspect ftp
   class http
     inspect http
```

Example 1-2 Misconfiguration for HTTP packets: FTP Inspection Also Configured

```
class-map ftp
   match port tcp eq 80 [it should be 21]
class-map http
   match port tcp eq 80
policy-map test
   class http
     inspect http
```

```
class ftp
  inspect ftp
```

Feature Matching for Multiple Service Policies

For TCP and UDP traffic (and ICMP when you enable stateful ICMP inspection), service policies operate on traffic flows, and not just individual packets. If traffic is part of an existing connection that matches a feature in a policy on one interface, that traffic flow cannot also match the same feature in a policy on another interface; only the first policy is used.

For example, if HTTP traffic matches a policy on the inside interface to inspect HTTP traffic, and you have a separate policy on the outside interface for HTTP inspection, then that traffic is not also inspected on the egress of the outside interface. Similarly, the return traffic for that connection will not be inspected by the ingress policy of the outside interface, nor by the egress policy of the inside interface.

For traffic that is not treated as a flow, for example ICMP when you do not enable stateful ICMP inspection, returning traffic can match a different policy map on the returning interface. For example, if you configure IPS on the inside and outside interfaces, but the inside policy uses virtual sensor 1 while the outside policy uses virtual sensor 2, then a non-stateful Ping will match virtual sensor 1 outbound, but will match virtual sensor 2 inbound.

Licensing Requirements for Service Policies

Model	License Requirement
All models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Supports IPv6 for the following features:

- Application inspection for DNS, FTP, HTTP, ICMP, ScanSafe, SIP, SMTP, IPsec-pass-thru, and IPv6.
- ASA IPS
- ASA CX
- NetFlow Secure Event Logging filtering
- TCP and UDP connection limits and timeouts, TCP sequence number randomization

- TCP normalization
- TCP state bypass
- User statistics for Identity Firewall

Class Map Guidelines

The maximum number of class mapsof all types is 255 in single mode or per context in multiple mode. Class maps include the following types:

- Layer 3/4 class maps (for through traffic and management traffic).
- Inspection class maps
- Regular expression class maps
- **match** commands used directly underneath an inspection policy map

This limit also includes default class maps of all types, limiting user-configured class mapsto approximately 235. See the "Default Class Maps" section on page 1-9.

Policy Map Guidelines

See the following guidelines for using policy maps:

- You can only assign one policy map per interface. (However you can create up to 64 policy maps in the configuration.)
- You can apply the same policy map to multiple interfaces.
- You can identify up to 63 Layer 3/4 class maps in a Layer 3/4 policy map.
- For each class map, you can assign multiple actions from one or more feature types, if supported. See the "Incompatibility of Certain Feature Actions" section on page 1-5.

Service Policy Guidelines

- Interface service policies take precedence over the global service policy for a given feature. For example, if you have a global policy with FTP inspection, and an interface policy with TCP normalization, then both FTP inspection and TCP normalization are applied to the interface. However, if you have a global policy with FTP inspection, and an interface policy with FTP inspection, then only the interface policy FTP inspection is applied to that interface.
- You can only apply one global policy. For example, you cannot create a global policy that includes feature set 1, and a separate global policy that includes feature set 2. All features must be included in a single policy.
- When you make service policy changes to the configuration, all *new* connections use the new service policy. Existing connections continue to use the policy that was configured at the time of the connection establishment. **show** command output will not include data about the old connections.

  For example, if you remove a QoS service policy from an interface, then re-add a modified version, then the **show service-policy** command only displays QoS counters associated with new connections that match the new service policy; existing connections on the old policy no longer show in the command output.

  To ensure that all connections use the new policy, you need to disconnect the current connections so they can reconnect using the new policy. See the **clear conn** or **clear local-host** commands.

Default Settings

The following topics describe the default settings for Modular Policy Framework:

- Default Configuration, page 1-8
- Default Class Maps, page 1-9

Default Configuration

By default, the configuration includes a policy that matches all default application inspection traffic and applies certain inspections to the traffic on all interfaces (a global policy). Not all inspections are enabled by default. You can only apply one global policy, so if you want to alter the global policy, you need to either edit the default policy or disable it and apply a new one. (An interface policy overrides the global policy for a particular feature.)

The default policy includes the following application inspections:

- DNS
- FTP
- H323 (H225)
- H323 (RAS)
- RSH
- RTSP
- ESMTP
- SQLnet
- Skinny (SCCP)
- SunRPC
- XDMCP
- SIP
- NetBios
- TFTP
- IP Options

The default policy configuration includes the following commands:

```
class-map inspection_default
 match default-inspection-traffic
policy-map type inspect dns preset_dns_map
 parameters
    message-length maximum client auto
    message-length maximum 512
    dns-guard
    protocol-enforcement
    nat-rewrite
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225 _default_h323_map
  inspect h323 ras _default_h323_map
  inspect ip-options _default_ip_options_map
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp _default_esmtp_map
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
service-policy global_policy global
```

Note See the "Incompatibility of Certain Feature Actions" section on page 1-5 for more information about the special **match default-inspection-traffic** command used in the default class map.

Default Class Maps

The configuration includes a default Layer 3/4 class map that the ASA uses in the default global policy called default-inspection-traffic; it matches the default inspection traffic. This class, which is used in the default global policy, is a special shortcut to match the default ports for all inspections. When used in a policy, this class ensures that the correct inspection is applied to each packet, based on the destination port of the traffic. For example, when UDP traffic for port 69 reaches the ASA, then the ASA applies the TFTP inspection; when TCP traffic for port 21 arrives, then the ASA applies the FTP inspection. So in this case only, you can configure multiple inspections for the same class map. Normally, the ASA does not use the port number to determine which inspection to apply, thus giving you the flexibility to apply inspections to non-standard ports, for example.

```
class-map inspection_default
 match default-inspection-traffic
```

Another class map that exists in the default configuration is called class-default, and it matches all traffic. This class map appears at the end of all Layer 3/4 policy maps and essentially tells the ASA to not perform any actions on all other traffic. You can use the class-default class if desired, rather than making your own **match any** class map. In fact, some features are only available for class-default, such as QoS traffic shaping.

```
class-map class-default
 match any
```

Task Flows for Configuring Service Policies

This section includes the following topics:

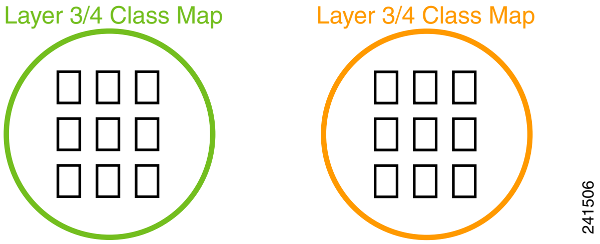
- Task Flow for Using the Modular Policy Framework, page 1-9
- Task Flow for Configuring Hierarchical Policy Maps for QoS Traffic Shaping, page 1-11

Task Flow for Using the Modular Policy Framework

To configure Modular Policy Framework, perform the following steps:

Step 1 Identify the traffic—Identify the traffic on which you want to perform Modular Policy Framework actions by creating Layer 3/4 class maps.

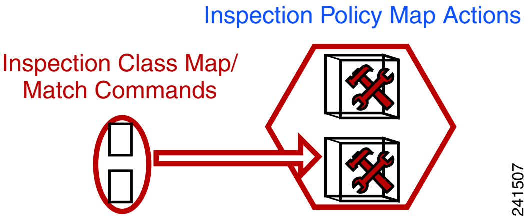
For example, you might want to perform actions on all traffic that passes through the ASA; or you might only want to perform certain actions on traffic from 10.1.1.0/24 to any destination address.



See the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12.

Step 2 Perform additional actions on some inspection traffic—If one of the actions you want to perform is application inspection, and you want to perform additional actions on some inspection traffic, then create an inspection policy map. The inspection policy map identifies the traffic and specifies what to do with it.

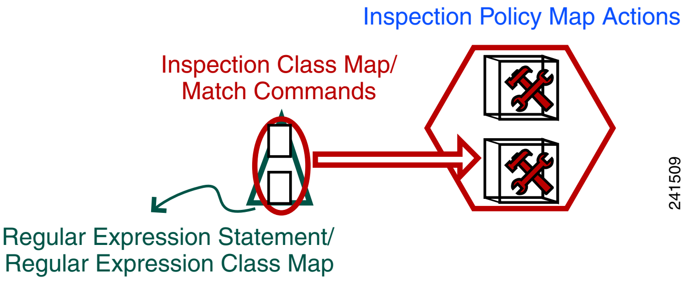
For example, you might want to drop all HTTP requests with a body length greater than 1000 bytes.



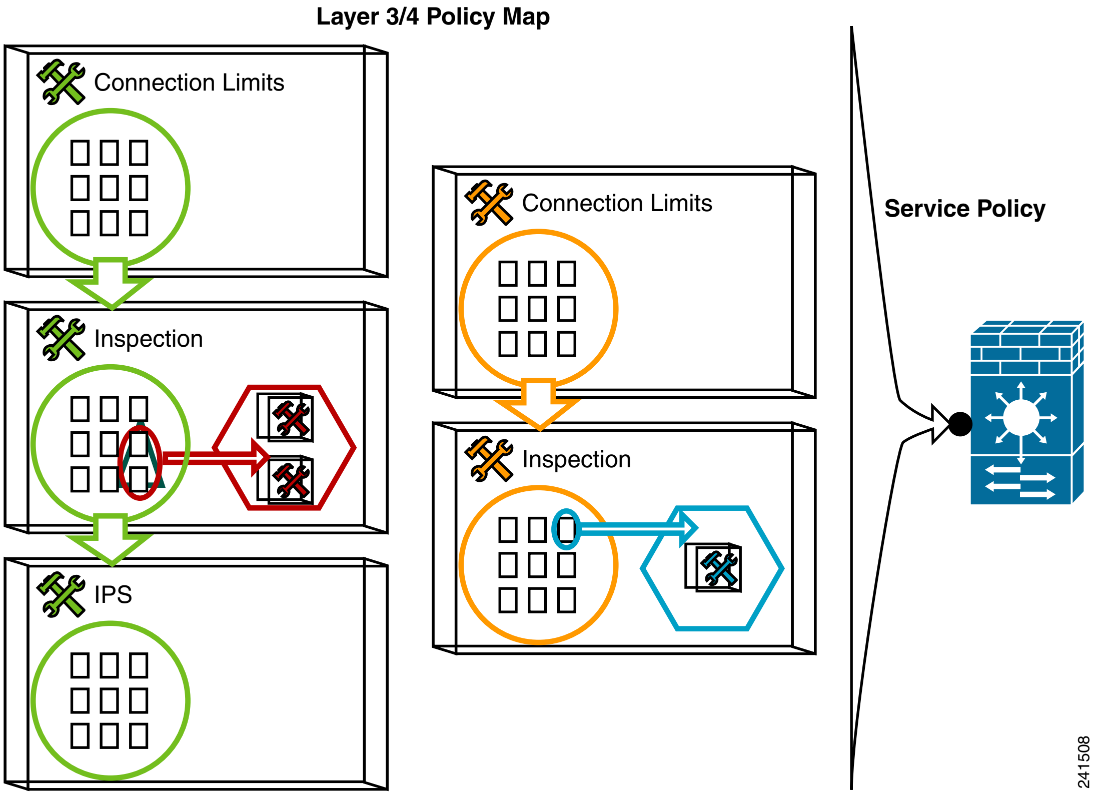
You can create a self-contained inspection policy map that identifies the traffic directly with **match** commands, or you can create an inspection class map for reuse or for more complicated matching. See the "Defining Actions in an Inspection Policy Map" section on page 2-4 and the "Identifying Traffic in an Inspection Class Map" section on page 2-5.

Step 3 Create a regular expression—If you want to match text with a regular expression within inspected packets, you can create a regular expression or a group of regular expressions (a regular expression class map). Then, when you define the traffic to match for the inspection policy map, you can call on an existing regular expression.

For example, you might want to drop all HTTP requests with a URL including the text "example.com."


Inspection Policy Map Actions
Inspection Class Map/
Match Commands
Regular Expression Statement/
Regular Expression Class Map
241509

Step 4 Define the actions you want to perform and determine on which interfaces you want to apply the policy map—Define the actions you want to perform on each Layer 3/4 class map by creating a Layer 3/4 policy map. Then, determine on which interfaces you want to apply the policy map using a service policy.



See the "Defining Actions (Layer 3/4 Policy Map)" section on page 1-15 and the "Applying Actions to an Interface (Service Policy)" section on page 1-17.

Task Flow for Configuring Hierarchical Policy Maps for QoS Traffic Shaping

If you enable QoS traffic shaping for a class map, then you can optionally enable priority queueing for a subset of shaped traffic. To do so, you need to create a policy map for the priority queueing, and then within the traffic shaping policy map, you can call the priority class map. Only the traffic shaping class map is applied to an interface.

See Chapter 23, "Information About QoS," for more information about this feature.

Hierarchical policy maps are only supported for traffic shaping and priority queueing.

To implement a hierarchical policy map, perform the following steps:

Step 1 Identify the prioritized traffic according to the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12.

You can create multiple class maps to be used in the hierarchical policy map.

Step 2 Create a policy map according to the "Defining Actions (Layer 3/4 Policy Map)" section on page 1-15, and identify the sole action for each class map as **priority**.

Step 3 Create a separate policy map according to the "Defining Actions (Layer 3/4 Policy Map)" section on page 1-15, and identify the **shape** action for the **class-default** class map.

Traffic shaping can only be applied the to **class-default** class map.

Step 4 For the same class map, identify the priority policy map that you created in Step 2 using the **service-policy** *priority_policy_map* command.

Step 5 Apply the shaping policy map to the interface accrding to "Applying Actions to an Interface (Service Policy)" section on page 1-17.

Identifying Traffic (Layer 3/4 Class Maps)

A Layer 3/4 class map identifies Layer 3 and 4 traffic to which you want to apply actions. You can create multiple Layer 3/4 class maps for each Layer 3/4 policy map.

This section includes the following topics:

- Creating a Layer 3/4 Class Map for Through Traffic, page 1-12
- Creating a Layer 3/4 Class Map for Management Traffic, page 1-14

Creating a Layer 3/4 Class Map for Through Traffic

A Layer 3/4 class map matches traffic based on protocols, ports, IP addresses and other Layer 3 or 4 attributes.

Detailed Steps

	Command	Purpose

Step 1	**`class-map`** *`class_map_name`*
	Example:
	`ciscoasa(config)# class-map all_udp`
		Creates a Layer 3/4 class map, where *class_map_name* is a string up to 40 characters in length. The name "class-default" is reserved. All types of class maps use the same name space, so you cannot reuse a name already used by another type of class map. The CLI enters class-map configuration mode.

Step 2	(Optional)
	`description` *`string`*
	Example:
	`hostname(config-cmap)# description All UDP traffic`
		Adds a description to the class map.

Step 3	Match traffic using one of the following:	Unless otherwise specified, you can include only one **match** command in the class map.

	`match any`
	Example:
	`hostname(config-cmap)# match any`
		Matches all traffic.

Command	Purpose
match access-list *access_list_name* **Example:** `hostname(config-cmap)# match access-list udp`	Matches traffic specified by an extended ACL. If the ASA is operating in transparent firewall mode, you can use an EtherType ACL.

match port {**tcp** | **udp**} {**eq** *port_num* | range *port_num port_num*}
Example:
`hostname(config-cmap)# match tcp eq 80`
	Matches TCP or UDP destination ports, either a single port or a contiguous range of ports.
	Tip For applications that use multiple, non-contiguous ports, use the **match access-list** command and define an ACE to match each port.

match default-inspection-traffic **Example:** `hostname(config-cmap)# match default-inspection-traffic`	Matches default traffic for inspection: the default TCP and UDP ports used by all applications that the ASA can inspect. This command, which is used in the default global policy, is a special CLI shortcut that when used in a policy map, ensures that the correct inspection is applied to each packet, based on the destination port of the traffic. For example, when UDP traffic for port 69 reaches the ASA, then the ASA applies the TFTP inspection; when TCP traffic for port 21 arrives, then the ASA applies the FTP inspection. So in this case only, you can configure multiple inspections for the same class map (with the exception of WAAS inspection, which can be configured with other inspections. See the "Incompatibility of Certain Feature Actions" section on page 1-5 for more information about combining actions). Normally, the ASA does not use the port number to determine the inspection applied, thus giving you the flexibility to apply inspections to non-standard ports, for example. See the "Default Settings and NAT Limitations" section on page 9-4 for a list of default ports. Not all applications whose ports are included in the **match default-inspection-traffic** command are enabled by default in the policy map. You can specify a **match access-list** command along with the **match default-inspection-traffic** command to narrow the matched traffic. Because the **match default-inspection-traffic** command specifies the ports and protocols to match, any ports and protocols in the ACL are ignored. **Tip** We suggest that you only inspect traffic on ports on which you expect application traffic; if you inspect all traffic, for example using **match any**, the ASA performance can be impacted.
match dscp *value1* [*value2*] [...] [*value8*] **Example:** `hostname(config-cmap)# match dscp af43 cs1 ef`	Matches DSCP value in an IP header, up to eight DSCP values.

Command	Purpose
match precedence *value1* [*value2*] [*value3*] [*value4*] **Example:** `hostname(config-cmap)# match precedence 1 4`	Matches up to four precedence values, represented by the TOS byte in the IP header, where *value1* through *value4* can be 0 to 7, corresponding to the possible precedences.
match rtp *starting_port range* **Example:** `hostname(config-cmap)# match rtp 4004 100`	Matches RTP traffic, where the *starting_port* specifies an even-numbered UDP destination port between 2000 and 65534. The *range* specifies the number of additional UDP ports to match above the *starting_port*, between 0 and 16383.

match tunnel-group *name*
(Optional)
match flow ip destination-address
Example:
`hostname(config-cmap)# match tunnel-group group1`
`hostname(config-cmap)# match flow ip destination-address`
	Matches VPN tunnel group traffic to which you want to apply QoS.
	You can also specify one other **match** command to refine the traffic match. You can specify any of the preceding commands, except for the **match any**, **match access-list**, or **match default-inspection-traffic** commands. Or you can also enter the **match flow ip destination-address** command to match flows in the tunnel group going to each IP address.

Examples

The following is an example for the **class-map** command:

```
ciscoasa(config)# access-list udp permit udp any any
ciscoasa(config)# access-list tcp permit tcp any any
ciscoasa(config)# access-list host_foo permit ip any 10.1.1.1 255.255.255.255

ciscoasa(config)# class-map all_udp
ciscoasa(config-cmap)# description "This class-map matches all UDP traffic"
ciscoasa(config-cmap)# match access-list udp

ciscoasa(config-cmap)# class-map all_tcp
ciscoasa(config-cmap)# description "This class-map matches all TCP traffic"
ciscoasa(config-cmap)# match access-list tcp

ciscoasa(config-cmap)# class-map all_http
ciscoasa(config-cmap)# description "This class-map matches all HTTP traffic"
ciscoasa(config-cmap)# match port tcp eq http

ciscoasa(config-cmap)# class-map to_server
ciscoasa(config-cmap)# description "This class-map matches all traffic to server 10.1.1.1"
ciscoasa(config-cmap)# match access-list host_foo
```

Creating a Layer 3/4 Class Map for Management Traffic

For management traffic to the ASA, you might want to perform actions specific to this kind of traffic. You can specify a management class map that can match an ACL or TCP or UDP ports. The types of actions available for a management class map in the policy map are specialized for management traffic. See the "Supported Features" section on page 1-2.

Detailed Steps

	Command	Purpose

Step 1	**class-map type management** *class_map_name*
	Example:
	`ciscoasa(config)# class-map type management all_mgmt`
		Creates a management class map, where *class_map_name* is a string up to 40 characters in length. The name “class-default” is reserved. All types of class maps use the same name space, so you cannot reuse a name already used by another type of class map. The CLI enters class-map configuration mode.

Step 2	(Optional) **description** *string* **Example:** `hostname(config-cmap)# description All management traffic`	Adds a description to the class map.
Step 3	Match traffic using one of the following:	Unless otherwise specified, you can include only one **match** command in the class map.
	match access-list *access_list_name* **Example:** `hostname(config-cmap)# match access-list udp`	Matches traffic specified by an extended ACL. If the ASA is operating in transparent firewall mode, you can use an EtherType ACL.

	match port {**tcp** | **udp**} {**eq** *port_num* | range *port_num port_num*}
	Example:
	`hostname(config-cmap)# match tcp eq 80`
		Matches TCP or UDP destination ports, either a single port or a contiguous range of ports.
		Tip For applications that use multiple, non-contiguous ports, use the **match access-list** command and define an ACE to match each port.

Defining Actions (Layer 3/4 Policy Map)

This section describes how to associate actions with Layer 3/4 class maps by creating a Layer 3/4 policy map.

Restrictions

The maximum number of policy maps is 64, but you can only apply one policy map per interface.

Detailed Steps

	Command	Purpose
Step 1	`policy-map` *policy_map_name* **Example:** `ciscoasa(config)# policy-map global_policy`	Adds the policy map. The *policy_map_name* argument is the name of the policy map up to 40 characters in length. All types of policy maps use the same name space, so you cannot reuse a name already used by another type of policy map. The CLI enters policy-map configuration mode.
Step 2	(Optional) `class` *class_map_name* **Example:** `hostname(config-pmap)# description global policy map`	Specifies a previously configured Layer 3/4 class map, where the *class_map_name* is the name of the class map. See the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12 to add a class map. **Note** If there is no **match default-inspection-traffic** command in a class map, then at most one **inspect** command is allowed to be configured under the class. For QoS, you can configure a hierarchical policy map for the traffic shaping and priority queue features. See the "Task Flow for Configuring Hierarchical Policy Maps for QoS Traffic Shaping" section on page 1-11 for more information.
Step 3	Specify one or more actions for this class map.	See the "Supported Features" section on page 1-2.
Step 4	Repeat Step 2 and Step 3 for each class map you want to include in this policy map.	

Examples

The following is an example of a **policy-map** command for connection policy. It limits the number of connections allowed to the web server 10.1.1.1:

```
ciscoasa(config)# access-list http-server permit tcp any host 10.1.1.1
ciscoasa(config)# class-map http-server
ciscoasa(config-cmap)# match access-list http-server

ciscoasa(config)# policy-map global-policy
ciscoasa(config-pmap)# description This policy map defines a policy concerning connection
to http server.
ciscoasa(config-pmap)# class http-server
ciscoasa(config-pmap-c)# set connection conn-max 256
```

The following example shows how multi-match works in a policy map:

```
ciscoasa(config)# class-map inspection_default
ciscoasa(config-cmap)# match default-inspection-traffic
ciscoasa(config)# class-map http_traffic
ciscoasa(config-cmap)# match port tcp eq 80

ciscoasa(config)# policy-map outside_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect http http_map
ciscoasa(config-pmap-c)# inspect sip
ciscoasa(config-pmap)# class http_traffic
ciscoasa(config-pmap-c)# set connection timeout idle 0:10:0
```

The following example shows how traffic matches the first available class map, and will not match any subsequent class maps that specify actions in the same feature domain:

```
ciscoasa(config)# class-map telnet_traffic
ciscoasa(config-cmap)# match port tcp eq 23
ciscoasa(config)# class-map ftp_traffic
ciscoasa(config-cmap)# match port tcp eq 21
ciscoasa(config)# class-map tcp_traffic
ciscoasa(config-cmap)# match port tcp range 1 65535
ciscoasa(config)# class-map udp_traffic
ciscoasa(config-cmap)# match port udp range 0 65535
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class telnet_traffic
ciscoasa(config-pmap-c)# set connection timeout idle 0:0:0
ciscoasa(config-pmap-c)# set connection conn-max 100
ciscoasa(config-pmap)# class ftp_traffic
ciscoasa(config-pmap-c)# set connection timeout idle 0:5:0
ciscoasa(config-pmap-c)# set connection conn-max 50
ciscoasa(config-pmap)# class tcp_traffic
ciscoasa(config-pmap-c)# set connection timeout idle 2:0:0
ciscoasa(config-pmap-c)# set connection conn-max 2000
```

When a Telnet connection is initiated, it matches **class telnet_traffic**. Similarly, if an FTP connection is initiated, it matches **class ftp_traffic**. For any TCP connection other than Telnet and FTP, it will match **class tcp_traffic**. Even though a Telnet or FTP connection can match **class tcp_traffic**, the ASA does not make this match because they previously matched other classes.

Applying Actions to an Interface (Service Policy)

To activate the Layer 3/4 policy map, create a service policy that applies it to one or more interfaces or that applies it globally to all interfaces.

Restrictions

You can only apply one global policy, so if you want to alter the global policy, you need to either edit the default policy or disable it and apply a new one. By default, the configuration includes a global policy that matches all default application inspection traffic and applies inspection to the traffic globally. The default service policy includes the following command:

```
service-policy global_policy global
```

Detailed Steps

Command	Purpose
service-policy *policy_map_name* **interface** *interface_name* [**fail-close**] **Example:** `ciscoasa(config)# service-policy inbound_policy interface outside`	Creates a service policy by associating a policy map with an interface. Specify the **fail-close** option to generate a syslog (767001) for IPv6 traffic that is dropped by application inspections that do not support IPv6 traffic. By default, syslogs are not generated. For a list of inspections that support IPv6, see the "IPv6 Guidelines" section on page 1-6.
service-policy *policy_map_name* **global** [**fail-close**] **Example:** `ciscoasa(config)# service-policy inbound_policy global`	Creates a service policy that applies to all interfaces that do not have a specific policy. Specify the **fail-close** option to generate a syslog (767001) for IPv6 traffic that is dropped by application inspections that do not support IPv6 traffic. By default, syslogs are not generated. For a list of inspections that support IPv6, see the "IPv6 Guidelines" section on page 1-6.

Examples

For example, the following command enables the inbound_policy policy map on the outside interface:

```
ciscoasa(config)# service-policy inbound_policy interface outside
```

The following commands disable the default global policy, and enables a new one called new_global_policy on all other ASA interfaces:

```
ciscoasa(config)# no service-policy global_policy global
ciscoasa(config)# service-policy new_global_policy global
```

Monitoring Modular Policy Framework

To monitor Modular Policy Framework, enter the following command:

Command	Purpose
show service-policy	Displays the service policy statistics.

Configuration Examples for Modular Policy Framework

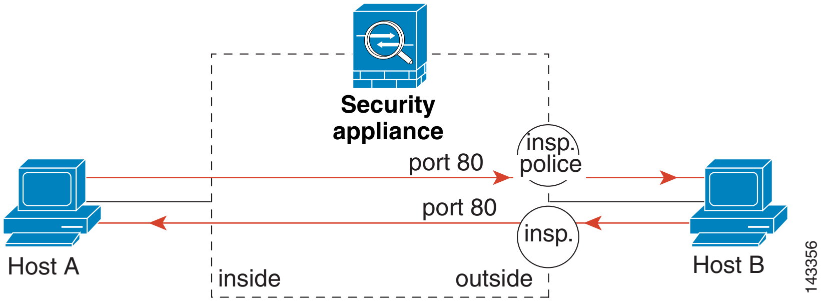
This section includes several Modular Policy Framework examples and includes the following topics:

- Applying Inspection and QoS Policing to HTTP Traffic, page 1-19
- Applying Inspection to HTTP Traffic Globally, page 1-19
- Applying Inspection and Connection Limits to HTTP Traffic to Specific Servers, page 1-20
- Applying Inspection to HTTP Traffic with NAT, page 1-21

Applying Inspection and QoS Policing to HTTP Traffic

In this example (see Figure 1-1), any HTTP connection (TCP traffic on port 80) that enters or exits the ASA through the outside interface is classified for HTTP inspection. Any HTTP traffic that exits the outside interface is classified for policing.

Figure 1-1 HTTP Inspection and QoS Policing



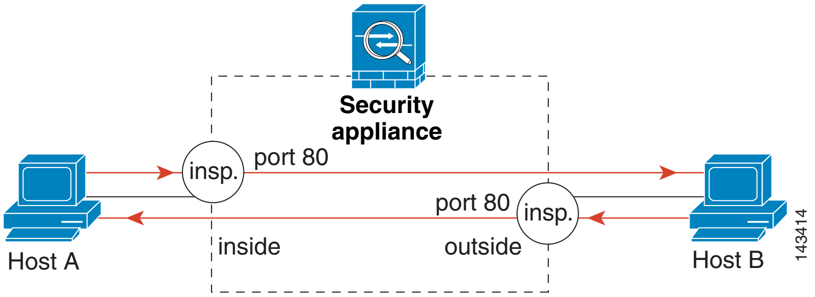
See the following commands for this example:

```
ciscoasa(config)# class-map http_traffic
ciscoasa(config-cmap)# match port tcp eq 80

ciscoasa(config)# policy-map http_traffic_policy
ciscoasa(config-pmap)# class http_traffic
ciscoasa(config-pmap-c)# inspect http
ciscoasa(config-pmap-c)# police output 250000
ciscoasa(config)# service-policy http_traffic_policy interface outside
```

Applying Inspection to HTTP Traffic Globally

In this example (see Figure 1-2), any HTTP connection (TCP traffic on port 80) that enters the ASA through any interface is classified for HTTP inspection. Because the policy is a global policy, inspection occurs only as the traffic enters each interface.

Figure 1-2 Global HTTP Inspection



See the following commands for this example:

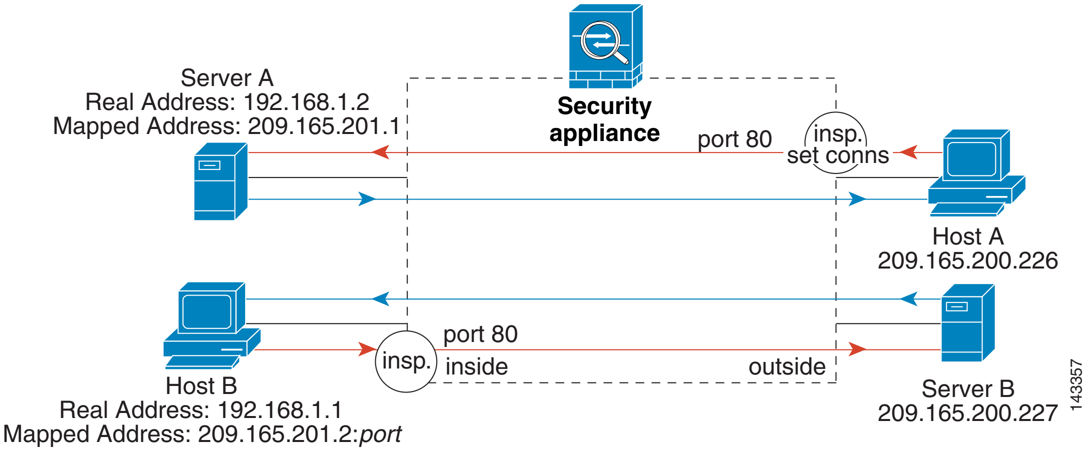
```
ciscoasa(config)# class-map http_traffic
ciscoasa(config-cmap)# match port tcp eq 80
ciscoasa(config)# policy-map http_traffic_policy
ciscoasa(config-pmap)# class http_traffic
ciscoasa(config-pmap-c)# inspect http
ciscoasa(config)# service-policy http_traffic_policy global
```

Applying Inspection and Connection Limits to HTTP Traffic to Specific Servers

In this example (see Figure 1-3), any HTTP connection destined for Server A (TCP traffic on port 80) that enters the ASA through the outside interface is classified for HTTP inspection and maximum connection limits. Connections initiated from Server A to Host A does not match the ACL in the class map, so it is not affected.

Any HTTP connection destined for Server B that enters the ASA through the inside interface is classified for HTTP inspection. Connections initiated from Server B to Host B does not match the ACL in the class map, so it is not affected.

Figure 1-3 HTTP Inspection and Connection Limits to Specific Servers



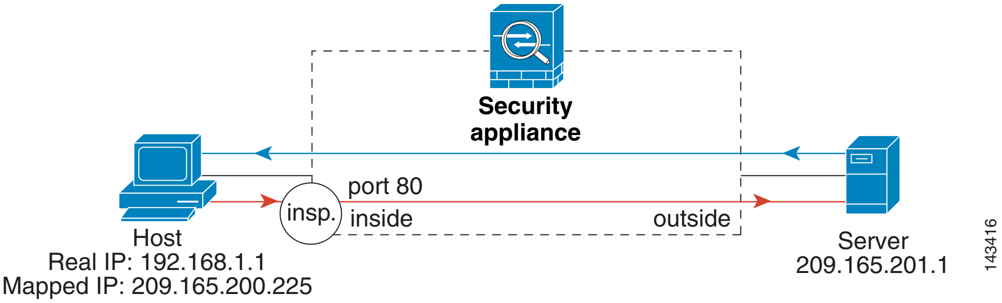
See the following commands for this example:

```
ciscoasa(config)# object network obj-192.168.1.2
ciscoasa(config-network-object)# host 192.168.1.2
ciscoasa(config-network-object)# nat (inside,outside) static 209.165.201.1
ciscoasa(config)# object network obj-192.168.1.0
ciscoasa(config-network-object)# subnet 192.168.1.0 255.255.255.0
ciscoasa(config-network-object)# nat (inside,outside) dynamic 209.165.201.2
ciscoasa(config)# access-list serverA extended permit tcp any host 209.165.201.1 eq 80
ciscoasa(config)# access-list ServerB extended permit tcp any host 209.165.200.227 eq 80

ciscoasa(config)# class-map http_serverA
ciscoasa(config-cmap)# match access-list serverA
ciscoasa(config)# class-map http_serverB
ciscoasa(config-cmap)# match access-list serverB

ciscoasa(config)# policy-map policy_serverA
ciscoasa(config-pmap)# class http_serverA
ciscoasa(config-pmap-c)# inspect http
ciscoasa(config-pmap-c)# set connection conn-max 100
ciscoasa(config)# policy-map policy_serverB
ciscoasa(config-pmap)# class http_serverB
ciscoasa(config-pmap-c)# inspect http
ciscoasa(config)# service-policy policy_serverB interface inside
ciscoasa(config)# service-policy policy_serverA interface outside
```

Applying Inspection to HTTP Traffic with NAT

In this example, the Host on the inside network has two addresses: one is the real IP address 192.168.1.1, and the other is a mapped IP address used on the outside network, 209.165.200.225. You must use the real IP address in the ACL in the class map. If you applied it to the outside interface, you would also use the real address.

Figure 1-4 HTTP Inspection with NAT



See the following commands for this example:

```
ciscoasa(config)# object network obj-192.168.1.1
ciscoasa(config-network-object)# host 192.168.1.1
ciscoasa(config-network-object)# nat (VM1,outside) static 209.165.200.225

ciscoasa(config)# access-list http_client extended permit tcp host 192.168.1.1 any eq 80

ciscoasa(config)# class-map http_client
ciscoasa(config-cmap)# match access-list http_client

ciscoasa(config)# policy-map http_client
ciscoasa(config-pmap)# class http_client
ciscoasa(config-pmap-c)# inspect http

ciscoasa(config)# service-policy http_client interface inside
```

Feature History for Service Policies

Table 1-3 lists the release history for this feature.

Table 1-3 Feature History for Service Policies

Feature Name	Releases	Feature Information
Modular Policy Framework	7.0(1)	Modular Policy Framework was introduced.
Management class map for use with RADIUS accounting traffic	7.2(1)	The management class map was introduced for use with RADIUS accounting traffic. The following commands were introduced: **class-map type management**, and **inspect radius-accounting**.
Inspection policy maps	7.2(1)	The inspection policy map was introduced. The following command was introduced: **class-map type inspect**.
Regular expressions and policy maps	7.2(1)	Regular expressions and policy maps were introduced to be used under inspection policy maps. The following commands were introduced: **class-map type regex**, **regex**, **match regex**.
Match any for inspection policy maps	8.0(2)	The **match any** keyword was introduced for use with inspection policy maps: traffic can match one or more criteria to match the class map. Formerly, only **match all** was available.

Configuring Special Actions for Application Inspections (Inspection Policy Map)

Modular Policy Framework lets you configure special actions for many application inspections. When you enable an inspection engine in the Layer 3/4 policy map, you can also optionally enable actions as defined in an *inspection policy map*. When the inspection policy map matches traffic within the Layer 3/4 class map for which you have defined an inspection action, then that subset of traffic will be acted upon as specified (for example, dropped or rate-limited).

This chapter includes the following sections:

Information About Inspection Policy Maps

See the "Configuring Application Layer Protocol Inspection" section on page 9-7 for a list of applications that support inspection policy maps.

An inspection policy map consists of one or more of the following elements. The exact options available for an inspection policy map depends on the application.

- Traffic matching command—You can define a traffic matching command directly in the inspection policy map to match application traffic to criteria specific to the application, such as a URL string, for which you then enable actions.
  - Some traffic matching commands can specify regular expressions to match text inside a packet. Be sure to create and test the regular expressions before you configure the policy map, either singly or grouped together in a regular expression class map.
- Inspection class map—An inspection class map includes multiple traffic matching commands. You then identify the class map in the policy map and enable actions for the class map as a whole. The difference between creating a class map and defining the traffic match directly in the inspection

policy map is that you can create more complex match criteria and you can reuse class maps. However, you cannot set different actions for different matches. **Note:** Not all inspections support inspection class maps.

- Parameters—Parameters affect the behavior of the inspection engine.

Guidelines and Limitations

- HTTP inspection policy maps—If you modify an in-use HTTP inspection policy map (**policy-map type inspect http**), you must remove and reapply the **inspect http** *map* action for the changes to take effect. For example, if you modify the "http-map" inspection policy map, you must remove and readd the **inspect http http-map** command from the layer 3/4 policy:

```
ciscoasa(config)# policy-map test
ciscoasa(config-pmap)# class http
ciscoasa(config-pmap-c)# no inspect http http-map
ciscoasa(config-pmap-c)# inspect http http-map
```

- All inspection policy maps—If you want to exchange an in-use inspection policy map for a different map name, you must remove the **inspect** *protocol map* command, and readd it with the new map. For example:

```
ciscoasa(config)# policy-map test
ciscoasa(config-pmap)# class sip
ciscoasa(config-pmap-c)# no inspect sip sip-map1
ciscoasa(config-pmap-c)# inspect sip sip-map2
```

- You can specify multiple **class** or **match** commands in the inspection policy map.

  If a packet matches multiple different **match** or **class** commands, then the order in which the ASA applies the actions is determined by internal ASA rules, and not by the order they are added to the inspection policy map. The internal rules are determined by the application type and the logical progression of parsing a packet, and are not user-configurable. For example for HTTP traffic, parsing a Request Method field precedes parsing the Header Host Length field; an action for the Request Method field occurs before the action for the Header Host Length field. For example, the following match commands can be entered in any order, but the **match request method get** command is matched first.

```
match request header host length gt 100
  reset
match request method get
  log
```

  If an action drops a packet, then no further actions are performed in the inspection policy map. For example, if the first action is to reset the connection, then it will never match any further **match** or **class** commands. If the first action is to log the packet, then a second action, such as resetting the connection, can occur.

  If a packet matches multiple **match** or **class** commands that are the same, then they are matched in the order they appear in the policy map. For example, for a packet with the header length of 1001, it will match the first command below, and be logged, and then will match the second command and be reset. If you reverse the order of the two **match** commands, then the packet will be dropped and the connection reset before it can match the second **match** command; it will never be logged.

```
match request header length gt 100
  log
match request header length gt 1000
  reset
```

A class map is determined to be the same type as another class map or **match** command based on the lowest priority **match** command in the class map (the priority is based on the internal rules). If a class map has the same type of lowest priority **match** command as another class map, then the class maps are matched according to the order they are added to the policy map. If the lowest priority match for each class map is different, then the class map with the higher priority **match** command is matched first. For example, the following three class maps contain two types of **match** commands: **match request-cmd** (higher priority) and **match filename** (lower priority). The ftp3 class map includes both commands, but it is ranked according to the lowest priority command, **match filename**. The ftp1 class map includes the highest priority command, so it is matched first, regardless of the order in the policy map. The ftp3 class map is ranked as being of the same priority as the ftp2 class map, which also contains the **match filename** command. They are matched according to the order in the policy map: ftp3 and then ftp2.

```
class-map type inspect ftp match-all ftp1
  match request-cmd get
class-map type inspect ftp match-all ftp2
  match filename regex abc
class-map type inspect ftp match-all ftp3
  match request-cmd get
  match filename regex abc

policy-map type inspect ftp ftp
  class ftp3
    log
  class ftp2
    log
  class ftp1
    log
```

Default Inspection Policy Maps

DNS inspection is enabled by default, using the preset_dns_map inspection class map:

- The maximum DNS message length is 512 bytes.
- The maximum client DNS message length is automatically set to match the Resource Record.
- DNS Guard is enabled, so the ASA tears down the DNS session associated with a DNS query as soon as the DNS reply is forwarded by the ASA. The ASA also monitors the message exchange to ensure that the ID of the DNS reply matches the ID of the DNS query.
- Translation of the DNS record based on the NAT configuration is enabled.
- Protocol enforcement is enabled, which enables DNS message format check, including domain name length of no more than 255 characters, label length of 63 characters, compression, and looped pointer check.

See the following default commands:

```
policy-map type inspect dns preset_dns_map
 parameters
    message-length maximum client auto
    message-length maximum 512
    dns-guard
    protocol-enforcement
    nat-rewrite
```

Note There are other default inspection policy maps such as **_default_esmtp_map**. For example, **inspect esmtp** implicitly uses the policy map "_default_esmtp_map." All the default policy maps can be shown by using the **show running-config all policy-map** command.

Defining Actions in an Inspection Policy Map

When you enable an inspection engine in the Layer 3/4 policy map, you can also optionally enable actions as defined in an inspection policy map.

Detailed Steps

	Command	Purpose
Step 1	(Optional) Create an inspection class map.	See the "Identifying Traffic in an Inspection Class Map" section on page 2-5. Alternatively, you can identify the traffic directly within the policy map.
Step 2	(Optional) Create a regular expression.	For policy map types that support regular expressions, see the general operations configuration guide.

Step 3	**`policy-map type inspect`** *`application policy_map_name`*
	Example:
	`ciscoasa(config)# policy-map type inspect http http_policy`
		Creates the inspection policy map. See the "Configuring Application Layer Protocol Inspection" section on page 9-7 for a list of applications that support inspection policy maps.
		The *policy_map_name* argument is the name of the policy map up to 40 characters in length. All types of policy maps use the same name space, so you cannot reuse a name already used by another type of policy map. The CLI enters policy-map configuration mode.

Step 4	Specify the traffic on which you want to perform actions using one of the following methods:	

	`class` *`class_map_name`*
	Example:
	`ciscoasa(config-pmap)# class http_traffic`
	`ciscoasa(config-pmap-c)#`
		Specifies the inspection class map that you created in the "Identifying Traffic in an Inspection Class Map" section on page 2-5.
		Not all applications support inspection class maps.

	Specify traffic directly in the policy map using one of the **match** commands described for each application in the inspection chapter.
	Example:
	`ciscoasa(config-pmap)# match req-resp content-type mismatch`
	`ciscoasa(config-pmap-c)#`
		If you use a **match not** command, then any traffic that matches the criterion in the **match not** command does not have the action applied.
		For policy map types that support regular expressions, see the general operations configuration guide.

	Command	Purpose
Step 5	*action* **Example:** `ciscoasa(config-pmap-c)# drop-connection log`	Specifies the action you want to perform on the matching traffic. Actions vary depending on the inspection and match type. Common actions include: **drop**, **log**, and **drop-connection**. For the actions available for each match, see the appropriate inspection chapter.

Step 6	**parameters**
	Example:
	`ciscoasa(config-pmap)# parameters`
	`ciscoasa(config-pmap-p)#`
		Configures parameters that affect the inspection engine. The CLI enters parameters configuration mode. For the parameters available for each application, see the appropriate inspection chapter.

Examples

The following is an example of an HTTP inspection policy map and the related class maps. This policy map is activated by the Layer 3/4 policy map, which is enabled by the service policy.

```
ciscoasa(config)# regex url_example example\.com
ciscoasa(config)# regex url_example2 example2\.com
ciscoasa(config)# class-map type regex match-any URLs
ciscoasa(config-cmap)# match regex url_example
ciscoasa(config-cmap)# match regex url_example2

ciscoasa(config-cmap)# class-map type inspect http match-all http-traffic
ciscoasa(config-cmap)# match req-resp content-type mismatch
ciscoasa(config-cmap)# match request body length gt 1000
ciscoasa(config-cmap)# match not request uri regex class URLs

ciscoasa(config-cmap)# policy-map type inspect http http-map1
ciscoasa(config-pmap)# class http-traffic
ciscoasa(config-pmap-c)# drop-connection log
ciscoasa(config-pmap-c)# match req-resp content-type mismatch
ciscoasa(config-pmap-c)# reset log
ciscoasa(config-pmap-c)# parameters
ciscoasa(config-pmap-p)# protocol-violation action log

ciscoasa(config-pmap-p)# policy-map test
ciscoasa(config-pmap)# class test (a Layer 3/4 class map not shown)
ciscoasa(config-pmap-c)# inspect http http-map1

ciscoasa(config-pmap-c)# service-policy test interface outside
```

Identifying Traffic in an Inspection Class Map

This type of class map allows you to match criteria that is specific to an application. For example, for DNS traffic, you can match the domain name in a DNS query.

A class map groups multiple traffic matches (in a match-all class map), or lets you match any of a list of matches (in a match-any class map). The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you group multiple match commands, and you can reuse class maps. For the traffic that you identify in this class map, you can specify actions such as dropping, resetting, and/or logging the connection in the inspection policy map. If you want to perform different actions on different types of traffic, you should identify the traffic directly in the policy map.

Restrictions

Not all applications support inspection class maps. See the CLI help for **class-map type inspect** for a list of supported applications.

Detailed Steps

	Command	Purpose
Step 1	(Optional) Create a regular expression.	See the general operations configuration guide.

Step 2	**class-map type inspect** *application* [**match-all** | **match-any**] *class_map_name*
	Example:
	`ciscoasa(config)# class-map type inspect http http_traffic`
	`ciscoasa(config-cmap)#`
		Creates an inspection class map, where the *application* is the application you want to inspect. For supported applications, see the CLI help for a list of supported applications or see Chapter 9, "Getting Started with Application Layer Protocol Inspection."
		The *class_map_name* argument is the name of the class map up to 40 characters in length.
		The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map.
		The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria.
		The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

Step 3	(Optional) **description** *string* **Example:** `hostname(config-cmap)# description All UDP traffic`	Adds a description to the class map.
Step 4	Define the traffic to include in the class by entering one or more **match** commands available for your application.	To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map. To see the **match** commands available for each application, see the appropriate inspection chapter.

Examples

The following example creates an HTTP class map that must match all criteria:

```
ciscoasa(config-cmap)# class-map type inspect http match-all http-traffic
ciscoasa(config-cmap)# match req-resp content-type mismatch
ciscoasa(config-cmap)# match request body length gt 1000
ciscoasa(config-cmap)# match not request uri regex class URLs
```

The following example creates an HTTP class map that can match any of the criteria:

```
ciscoasa(config-cmap)# class-map type inspect http match-any monitor-http
ciscoasa(config-cmap)# match request method get
ciscoasa(config-cmap)# match request method put
ciscoasa(config-cmap)# match request method post
```

Where to Go Next

To use an inspection policy, see Chapter 1, "Configuring a Service Policy Using the Modular Policy Framework."

Feature History for Inspection Policy Maps

Table 2-1 lists the release history for this feature.

Table 2-1 Feature History for Service Policies

Feature Name	Releases	Feature Information
Inspection policy maps	7.2(1)	The inspection policy map was introduced. The following command was introduced: **class-map type inspect**.
Regular expressions and policy maps	7.2(1)	Regular expressions and policy maps were introduced to be used under inspection policy maps. The following commands were introduced: **class-map type regex**, **regex**, **match regex**.
Match any for inspection policy maps	8.0(2)	The **match any** keyword was introduced for use with inspection policy maps: traffic can match one or more criteria to match the class map. Formerly, only **match all** was available.

PART 2

Configuring Network Address Translation

CHAPTER 3

Information About NAT

This chapter provides an overview of how Network Address Translation (NAT) works on the ASA. This chapter includes the following sections:

Note To start configuring NAT, see Chapter 4, "Configuring Network Object NAT," or Chapter 5, "Configuring Twice NAT."

Why Use NAT?

Each computer and device within an IP network is assigned a unique IP address that identifies the host. Because of a shortage of public IPv4 addresses, most of these IP addresses are private, not routable anywhere outside of the private company network. RFC 1918 defines the private IP addresses you can use internally that should not be advertised:

- 10.0.0.0 through 10.255.255.255
- 172.16.0.0 through 172.31.255.255
- 192.168.0.0 through 192.168.255.255

One of the main functions of NAT is to enable private IP networks to connect to the Internet. NAT replaces a private IP address with a public IP address, translating the private addresses in the internal private network into legal, routable addresses that can be used on the public Internet. In this way, NAT conserves public addresses because it can be configured to advertise at a minimum only one public address for the entire network to the outside world.

Other functions of NAT include:

- Security—Keeping internal IP addresses hidden discourages direct attacks.
- IP routing solutions—Overlapping IP addresses are not a problem when you use NAT.
- Flexibility—You can change internal IP addressing schemes without affecting the public addresses available externally; for example, for a server accessible to the Internet, you can maintain a fixed IP address for Internet use, but internally, you can change the server address.
- Translating between IPv4 and IPv6 (Routed mode only) —If you want to connect an IPv6 network to an IPv4 network, NAT lets you translate between the two types of addresses.

Note NAT is not required. If you do not configure NAT for a given set of traffic, that traffic will not be translated, but will have all of the security policies applied as normal.

NAT Terminology

This document uses the following terminology:

- Real address/host/network/interface—The real address is the address that is defined on the host, before it is translated. In a typical NAT scenario where you want to translate the inside network when it accesses the outside, the inside network would be the "real" network. Note that you can translate any network connected to the ASA, not just an inside network, Therefore if you configure NAT to translate outside addresses, "real" can refer to the outside network when it accesses the inside network.
- Mapped address/host/network/interface—The mapped address is the address that the real address is translated to. In a typical NAT scenario where you want to translate the inside network when it accesses the outside, the outside network would be the "mapped" network.

Note During address translation, IP addresses residing on the ASA's interfaces are not translated.

- Bidirectional initiation—Static NAT allows connections to be initiated *bidirectionally*, meaning both to the host and from the host.
- Source and destination NAT—For any given packet, both the source and destination IP addresses are compared to the NAT rules, and one or both can be translated/untranslated. For static NAT, the rule is bidirectional, so be aware that "source" and "destination" are used in commands and descriptions throughout this guide even though a given connection might originate at the "destination" address.

NAT Types

NAT Types Overview

You can implement NAT using the following methods:

- Static NAT—A consistent mapping between a real and mapped IP address. Allows bidirectional traffic initiation. See the "Static NAT" section on page 3-3.
- Dynamic NAT—A group of real IP addresses are mapped to a (usually smaller) group of mapped IP addresses, on a first come, first served basis. Only the real host can initiate traffic. See the "Dynamic NAT" section on page 3-7.
- Dynamic Port Address Translation (PAT)—A group of real IP addresses are mapped to a single IP address using a unique source port of that IP address. See the "Dynamic PAT" section on page 3-8.
- Identity NAT—A real address is statically translated to itself, essentially bypassing NAT. You might want to configure NAT this way when you want to translate a large group of addresses, but then want to exempt a smaller subset of addresses. See the "Identity NAT" section on page 3-10.

Static NAT

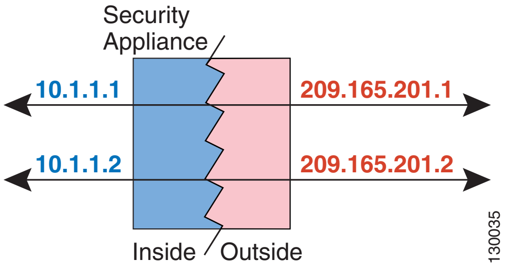
This section describes static NAT and includes the following topics:

Information About Static NAT

Static NAT creates a fixed translation of a real address to a mapped address. Because the mapped address is the same for each consecutive connection, static NAT allows bidirectional connection initiation, both to and from the host (if an access rule exists that allows it). With dynamic NAT and PAT, on the other hand, each host uses a different address or port for each subsequent translation, so bidirectional initiation is not supported.

Figure 3-1 shows a typical static NAT scenario. The translation is always active so both real and remote hosts can initiate connections.

Figure 3-1 Static NAT



Note You can disable bidirectionality if desired.

Information About Static NAT with Port Translation

Static NAT with port translation lets you specify a real and mapped protocol (TCP or UDP) and port.

This section includes the following topics:

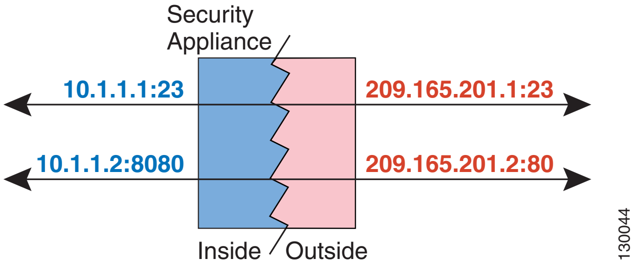
- Information About Static NAT with Port Address Translation, page 3-4
- Static NAT with Identity Port Translation, page 3-5
- Static NAT with Port Translation for Non-Standard Ports, page 3-5
- Static Interface NAT with Port Translation, page 3-5

Information About Static NAT with Port Address Translation

When you specify the port with static NAT, you can choose to map the port and/or the IP address to the same value or to a different value.

Figure 3-2 shows a typical static NAT with port translation scenario showing both a port that is mapped to itself and a port that is mapped to a different value; the IP address is mapped to a different value in both cases. The translation is always active so both translated and remote hosts can initiate connections.

Figure 3-2 Typical Static NAT with Port Translation Scenario



Note For applications that require application inspection for secondary channels (for example, FTP and VoIP), the ASA automatically translates the secondary ports.

Static NAT with Identity Port Translation

The following static NAT with port translation example provides a single address for remote users to access FTP, HTTP, and SMTP. These servers are actually different devices on the real network, but for each server, you can specify static NAT with port translation rules that use the same mapped IP address, but different ports.

Static NAT with Port Translation for Non-Standard Ports

You can also use static NAT with port translation to translate a well-known port to a non-standard port or vice versa. For example, if inside web servers use port 8080, you can allow outside users to connect to port 80, and then undo translation to the original port 8080. Similarly, to provide extra security, you can tell web users to connect to non-standard port 6785, and then undo translation to port 80.

Static Interface NAT with Port Translation

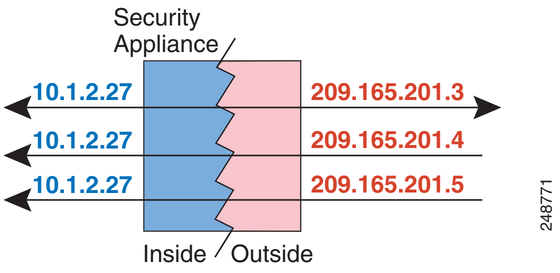
You can configure static NAT to map a real address to an interface address/port combination. For example, if you want to redirect Telnet access for the ASA outside interface to an inside host, then you can map the inside host IP address/port 23 to the ASA interface address/port 23. (Note that although Telnet to the ASA is not allowed to the lowest security interface, static NAT with interface port translation redirects the Telnet session instead of denying it).

Information About One-to-Many Static NAT

Typically, you configure static NAT with a one-to-one mapping. However, in some cases, you might want to configure a single real address to several mapped addresses (one-to-many). When you configure one-to-many static NAT, when the real host initiates traffic, it always uses the first mapped address. However, for traffic initiated to the host, you can initiate traffic to any of the mapped addresses, and they will be untranslated to the single real address.

Figure 3-3 shows a typical one-to-many static NAT scenario. Because initiation by the real host always uses the first mapped address, the translation of real host IP/1st mapped IP is technically the only bidirectional translation.

Figure 3-3 One-to-Many Static NAT



For example, you have a load balancer at 10.1.2.27. Depending on the URL requested, it redirects traffic to the correct web server.

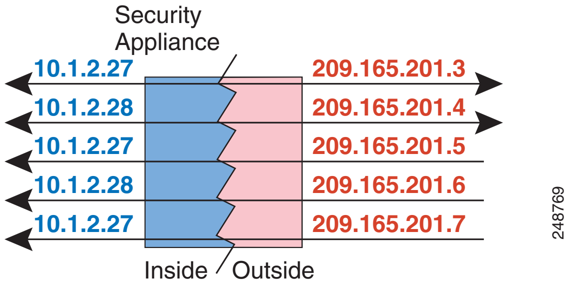
Information About Other Mapping Scenarios (Not Recommended)

The ASA has the flexibility to allow any kind of static mapping scenario: one-to-one, one-to-many, but also few-to-many, many-to-few, and many-to-one mappings. We recommend using only one-to-one or one-to-many mappings. These other mapping options might result in unintended consequences.

Functionally, few-to-many is the same as one-to-many; but because the configuration is more complicated and the actual mappings may not be obvious at a glance, we recommend creating a one-to-many configuration for each real address that requires it. For example, for a few-to-many scenario, the few real addresses are mapped to the many mapped addresses in order (A to 1, B to 2, C to 3). When all real addresses are mapped, the next mapped address is mapped to the first real address, and so on until all mapped addresses are mapped (A to 4, B to 5, C to 6). This results in multiple mapped addresses for each real address. Just like a one-to-many configuration, only the first mappings are bidirectional; subsequent mappings allow traffic to be initiated *to* the real host, but all traffic *from* the real host uses only the first mapped address for the source.

Figure 3-4 shows a typical few-to-many static NAT scenario.

Figure 3-4 Few-to-Many Static NAT

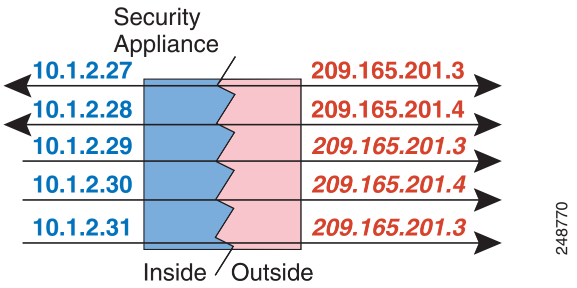


For a many-to-few or many-to-one configuration, where you have more real addresses than mapped addresses, you run out of mapped addresses before you run out of real addresses. Only the mappings between the lowest real IP addresses and the mapped pool result in bidirectional initiation. The remaining higher real addresses can initiate traffic, but traffic cannot be initiated to them (returning traffic for a connection is directed to the correct real address because of the unique 5-tuple (source IP, destination IP, source port, destination port, protocol) for the connection).

Note Many-to-few or many-to-one NAT is not PAT. If two real hosts use the same source port number and go to the same outside server and the same TCP destination port, and both hosts are translated to the same IP address, then both connections will be reset because of an address conflict (the 5-tuple is not unique).

Figure 3-5 shows a typical many-to-few static NAT scenario.

Figure 3-5 Many-to-Few Static NAT



Instead of using a static rule this way, we suggest that you create a one-to-one rule for the traffic that needs bidirectional initiation, and then create a dynamic rule for the rest of your addresses.

Dynamic NAT

This section describes dynamic NAT and includes the following topics:

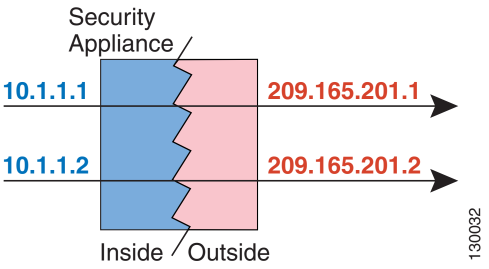
- Information About Dynamic NAT, page 3-7
- Dynamic NAT Disadvantages and Advantages, page 3-8

Information About Dynamic NAT

Dynamic NAT translates a group of real addresses to a pool of mapped addresses that are routable on the destination network. The mapped pool typically includes fewer addresses than the real group. When a host you want to translate accesses the destination network, the ASA assigns the host an IP address from the mapped pool. The translation is created only when the real host initiates the connection. The translation is in place only for the duration of the connection, and a given user does not keep the same IP address after the translation times out. Users on the destination network, therefore, cannot initiate a reliable connection to a host that uses dynamic NAT, even if the connection is allowed by an access rule.

Figure 3-6 shows a typical dynamic NAT scenario. Only real hosts can create a NAT session, and responding traffic is allowed back.

Figure 3-6 Dynamic NAT



Note For the duration of the translation, a remote host can initiate a connection to the translated host if an access rule allows it. Because the address is unpredictable, a connection to the host is unlikely. Nevertheless, in this case you can rely on the security of the access rule.

Dynamic NAT Disadvantages and Advantages

Dynamic NAT has these disadvantages:

- If the mapped pool has fewer addresses than the real group, you could run out of addresses if the amount of traffic is more than expected.

  Use PAT or a PAT fallback method if this event occurs often because PAT provides over 64,000 translations using ports of a single address.
- You have to use a large number of routable addresses in the mapped pool, and routable addresses may not be available in large quantities.

The advantage of dynamic NAT is that some protocols cannot use PAT. PAT does not work with the following:

- IP protocols that do not have a port to overload, such as GRE version 0.
- Some multimedia applications that have a data stream on one port, the control path on another port, and are not open standard.

See the "Default Settings and NAT Limitations" section on page 9-4 for more information about NAT and PAT support.

Dynamic PAT

This section describes dynamic PAT and includes the following topics:

- Information About Dynamic PAT, page 3-8
- Per-Session PAT vs. Multi-Session PAT, page 3-9
- Dynamic PAT Disadvantages and Advantages, page 3-9

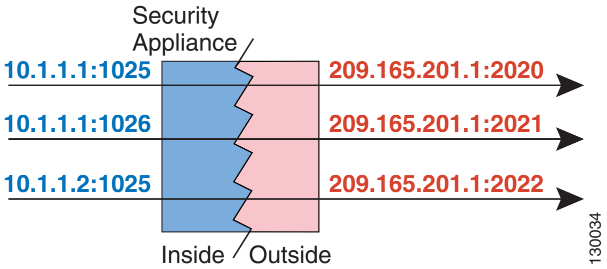
Information About Dynamic PAT

Dynamic PAT translates multiple real addresses to a single mapped IP address by translating the real address and source port to the mapped address and a unique port. If available, the real source port number is used for the mapped port. However, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool that can be used. If you have a lot of traffic that uses the lower port ranges, you can specify a flat range of ports to be used instead of the three unequal-sized tiers.

Each connection requires a separate translation session because the source port differs for each connection. For example, 10.1.1.1:1025 requires a separate translation from 10.1.1.1:1026.

Figure 3-7 shows a typical dynamic PAT scenario. Only real hosts can create a NAT session, and responding traffic is allowed back. The mapped address is the same for each translation, but the port is dynamically assigned.

Figure 3-7 Dynamic PAT



After the connection expires, the port translation also expires. For multi-session PAT, the PAT timeout is used, 30 seconds by default. For per-session PAT, the xlate is immediately removed. Users on the destination network cannot reliably initiate a connection to a host that uses PAT (even if the connection is allowed by an access rule).

Note For the duration of the translation, a remote host can initiate a connection to the translated host if an access rule allows it. Because the port address (both real and mapped) is unpredictable, a connection to the host is unlikely. Nevertheless, in this case you can rely on the security of the access rule.

Per-Session PAT vs. Multi-Session PAT

The per-session PAT feature improves the scalability of PAT and, for clustering, allows each member unit to own PAT connections; multi-session PAT connections have to be forwarded to and owned by the master unit. At the end of a per-session PAT session, the ASA sends a reset and immediately removes the xlate. This reset causes the end node to immediately release the connection, avoiding the TIME_WAIT state. Multi-session PAT, on the other hand, uses the PAT timeout, by default 30 seconds. For “hit-and-run” traffic, such as HTTP or HTTPS, the per-session feature can dramatically increase the connection rate supported by one address. Without the per-session feature, the maximum connection rate for one address for an IP protocol is approximately 2000 per second. With the per-session feature, the connection rate for one address for an IP protocol is 65535/*average-lifetime*.

By default, all TCP traffic and UDP DNS traffic use a per-session PAT xlate. For traffic that can benefit from multi-session PAT, such as H.323, SIP, or Skinny, you can disable per-session PAT be creating a per-session deny rule. See the “Configuring Per-Session PAT Rules” section on page 4-16.

Dynamic PAT Disadvantages and Advantages

Dynamic PAT lets you use a single mapped address, thus conserving routable addresses. You can even use the ASA interface IP address as the PAT address.

Dynamic PAT does not work with some multimedia applications that have a data stream that is different from the control path. See the “Default Settings and NAT Limitations” section on page 9-4 for more information about NAT and PAT support.

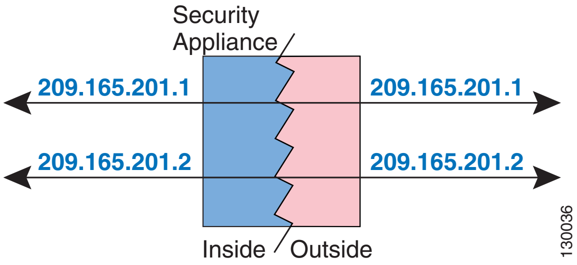
Dynamic PAT may also create a large number of connections appearing to come from a single IP address, and servers might interpret the traffic as a DoS attack. You can configure a PAT pool of addresses and use a round-robin assignment of PAT addresses to mitigate this situation.

Identity NAT

You might have a NAT configuration in which you need to translate an IP address to itself. For example, if you create a broad rule that applies NAT to every network, but want to exclude one network from NAT, you can create a static NAT rule to translate an address to itself. Identity NAT is necessary for remote access VPN, where you need to exempt the client traffic from NAT.

Figure 3-8 shows a typical identity NAT scenario.

Figure 3-8 Identity NAT



NAT in Routed and Transparent Mode

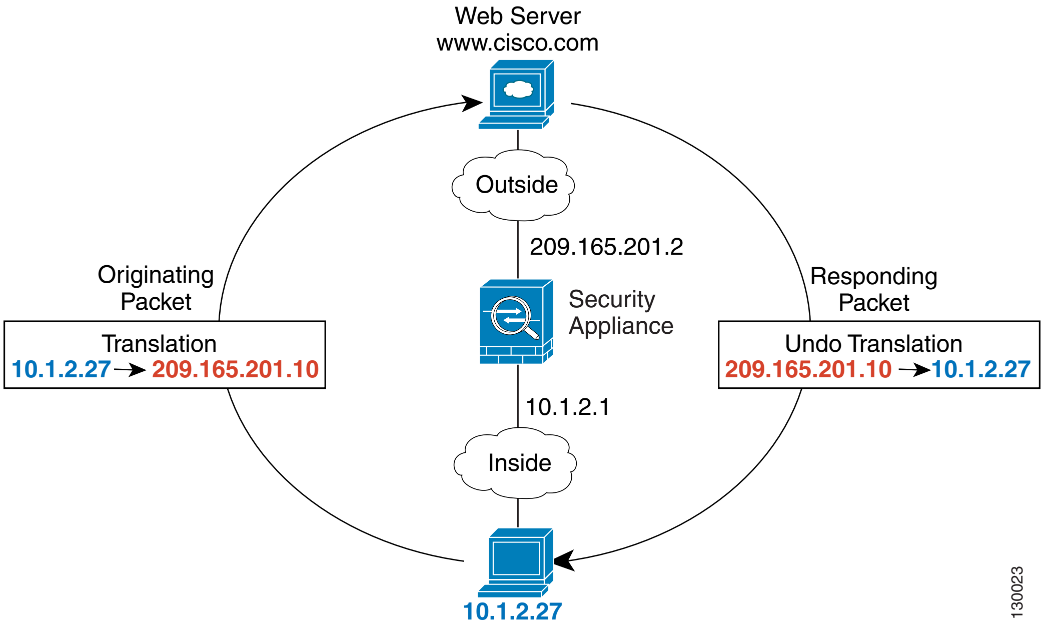
You can configure NAT in both routed and transparent firewall mode. This section describes typical usage for each firewall mode and includes the following topics:

- NAT in Routed Mode, page 3-11
- NAT in Transparent Mode, page 3-11

NAT in Routed Mode

Figure 3-9 shows a typical NAT example in routed mode, with a private network on the inside.

Figure 3-9 NAT Example: Routed Mode



1. When the inside host at 10.1.2.27 sends a packet to a web server, the real source address of the packet, 10.1.2.27, is changed to a mapped address, 209.165.201.10.
2. When the server responds, it sends the response to the mapped address, 209.165.201.10, and the ASA receives the packet because the ASA performs proxy ARP to claim the packet.
3. The ASA then changes the translation of the mapped address, 209.165.201.10, back to the real address, 10.1.2.27, before sending it to the host.

NAT in Transparent Mode

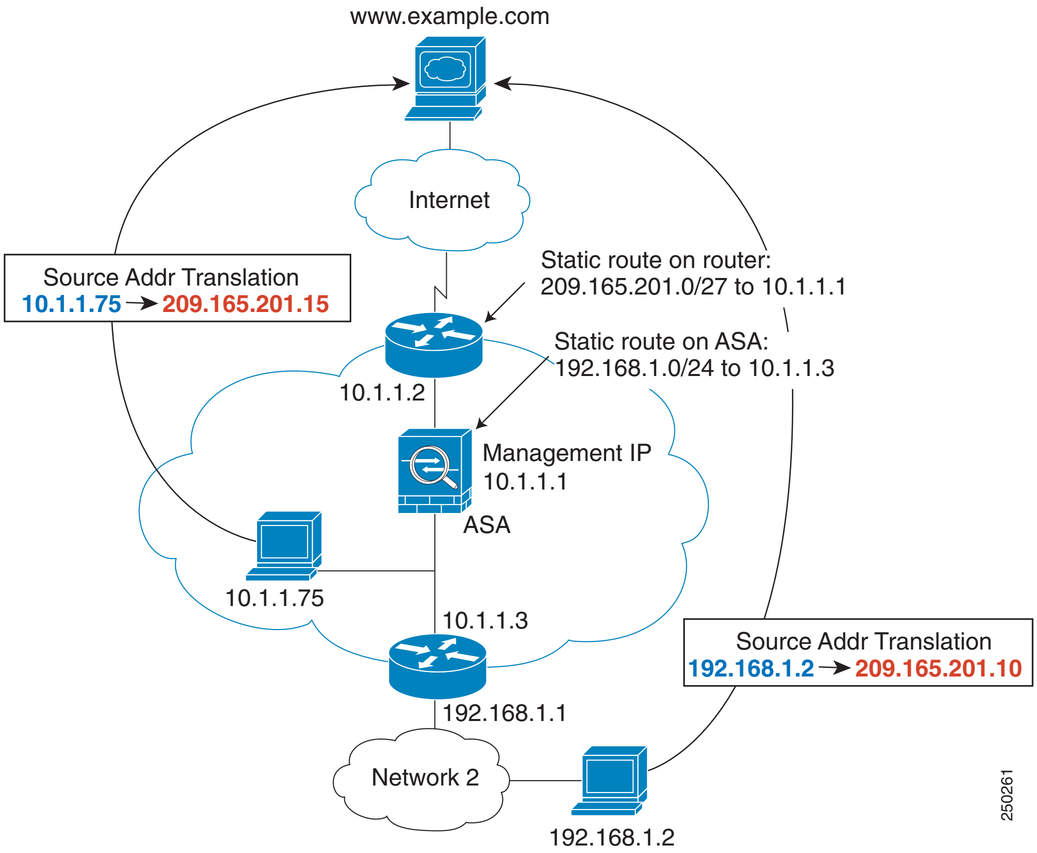
Using NAT in transparent mode eliminates the need for the upstream or downstream routers to perform NAT for their networks.

NAT in transparent mode has the following requirements and limitations:

- Because the transparent firewall does not have any interface IP addresses, you cannot use interface PAT.
- ARP inspection is not supported. Moreover, if for some reason a host on one side of the ASA sends an ARP request to a host on the other side of the ASA, and the initiating host real address is mapped to a different address on the same subnet, then the real address remains visible in the ARP request.
- Translating between IPv4 and IPv6 networks is not supported. Translating between two IPv6 networks, or between two IPv4 networks is supported.

Figure 3-10 shows a typical NAT scenario in transparent mode, with the same network on the inside and outside interfaces. The transparent firewall in this scenario is performing the NAT service so that the upstream router does not have to perform NAT.

Figure 3-10 NAT Example: Transparent Mode



1. When the inside host at 10.1.1.75 sends a packet to a web server, the real source address of the packet, 10.1.1.75, is changed to a mapped address, 209.165.201.15.
2. When the server responds, it sends the response to the mapped address, 209.165.201.15, and the ASA receives the packet because the upstream router includes this mapped network in a static route directed to the ASA management IP address. See the "Mapped Addresses and Routing" section on page 3-20 for more information about required routes.
3. The ASA then undoes the translation of the mapped address, 209.165.201.15, back to the real address, 10.1.1.1.75. Because the real address is directly-connected, the ASA sends it directly to the host.
4. For host 192.168.1.2, the same process occurs, except for returning traffic, the ASA looks up the route in its routing table and sends the packet to the downstream router at 10.1.1.3 based on the ASA static route for 192.168.1.0/24. See the "Transparent Mode Routing Requirements for Remote Networks" section on page 3-21 for more information about required routes.

NAT and IPv6

You can use NAT to translate between IPv6 networks, and also to translate between IPv4 and IPv6 networks (routed mode only). We recommend the following best practices:

- NAT66 (IPv6-to-IPv6)—We recommend using static NAT. Although you can use dynamic NAT or PAT, IPv6 addresses are in such large supply, you do not have to use dynamic NAT. If you do not want to allow returning traffic, you can make the static NAT rule unidirectional (twice NAT only).
- NAT46 (IPv4-to-IPv6)—We recommend using static NAT. Because the IPv6 address space is so much larger than the IPv4 address space, you can easily accommodate a static translation. If you do not want to allow returning traffic, you can make the static NAT rule unidirectional (twice NAT only). When translating to an IPv6 subnet (/96 or lower), the resulting mapped address is by default an IPv4-embedded IPv6 address, where the 32-bits of the IPv4 address is embedded after the IPv6 prefix. For example, if the IPv6 prefix is a /96 prefix, then the IPv4 address is appended in the last 32-bits of the address. For example, if you map 192.168.1.0/24 to 201b::0/96, then 192.168.1.4 will be mapped to 201b::0.192.168.1.4 (shown with mixed notation). If the prefix is smaller, such as /64, then the IPv4 address is appended after the prefix, and a suffix of 0s is appended after the IPv4 address. You can also optionally translate the addresses net-tonet, where the first IPv4 address maps to the first IPv6 address, the second to the second, and so on.
- NAT64 (IPv6-to-IPv4)—You may not have enough IPv4 addresses to accommodate the number of IPv6 addresses. We recommend using a dynamic PAT pool to provide a large number of IPv4 translations.

For specific implementation guidelines and limitations, see the configuration chapters.

How NAT is Implemented

The ASA can implement address translation in two ways: *network object NAT* and *twice NAT*. This section includes the following topics:

- Main Differences Between Network Object NAT and Twice NAT, page 3-13
- Information About Network Object NAT, page 3-14
- Information About Twice NAT, page 3-14

Main Differences Between Network Object NAT and Twice NAT

The main differences between these two NAT types are:

- How you define the real address.
  - Network object NAT—You define NAT as a parameter for a network object. A network object names an IP host, range, or subnet so you can then use the object in configuration instead of the actual IP addresses. The network object IP address serves as the real address. This method lets you easily add NAT to network objects that might already be used in other parts of your configuration.
  - Twice NAT—You identify a network object or network object group for both the real and mapped addresses. In this case, NAT is not a parameter of the network object; the network object or group is a parameter of the NAT configuration. The ability to use a network object *group* for the real address means that twice NAT is more scalable.

- How source and destination NAT is implemented.
  - Network object NAT— Each rule can apply to either the source or destination of a packet. So two rules might be used, one for the source IP address, and one for the destination IP address. These two rules cannot be tied together to enforce a specific translation for a source/destination combination.
  - Twice NAT—A single rule translates both the source and destination. A matching packet only matches the one rule, and further rules are not checked. Even if you do not configure the optional destination address for twice NAT, a matching packet still only matches one twice NAT rule. The source and destination are tied together, so you can enforce different translations depending on the source/destination combination. For example, sourceA/destinationA can have a different translation than sourceA/destinationB.
- Order of NAT Rules.
  - Network object NAT—Automatically ordered in the NAT table.
  - Twice NAT—Manually ordered in the NAT table (before or after network object NAT rules).

  See the "NAT Rule Order" section on page 3-18 for more information.

We recommend using network object NAT unless you need the extra features that twice NAT provides. Network object NAT is easier to configure, and might be more reliable for applications such as Voice over IP (VoIP). (For VoIP, because twice NAT is applicable only between two objects, you might see a failure in the translation of indirect addresses that do not belong to either of the objects.)

Information About Network Object NAT

All NAT rules that are configured as a parameter of a network object are considered to be *network object NAT* rules. Network object NAT is a quick and easy way to configure NAT for a network object, which can be a single IP address, a range of addresses, or a subnet.

After you configure the network object, you can then identify the mapped address for that object, either as an inline address or as another network object or network object group.

When a packet enters the ASA, both the source and destination IP addresses are checked against the network object NAT rules. The source and destination address in the packet can be translated by separate rules if separate matches are made. These rules are not tied to each other; different combinations of rules can be used depending on the traffic.

Because the rules are never paired, you cannot specify that sourceA/destinationA should have a different translation than sourceA/destinationB. Use twice NAT for that kind of functionality (twice NAT lets you identify the source and destination address in a single rule).

To start configuring network object NAT, see Chapter 4, "Configuring Network Object NAT."

Information About Twice NAT

Twice NAT lets you identify both the source and destination address in a single rule. Specifying both the source and destination addresses lets you specify that sourceA/destinationA can have a different translation than sourceA/destinationB.

The destination address is optional. If you specify the destination address, you can either map it to itself (identity NAT), or you can map it to a different address. The destination mapping is always a static mapping.

Twice NAT also lets you use service objects for static NAT with port translation; network object NAT only accepts inline definition.

To start configuring twice NAT, see Chapter 5, “Configuring Twice NAT.”

Figure 3-11 shows a host on the 10.1.2.0/24 network accessing two different servers. When the host accesses the server at 209.165.201.11, the real address is translated to 209.165.202.129. When the host accesses the server at 209.165.200.225, the real address is translated to 209.165.202.130.

Figure 3-11 Twice NAT with Different Destination Addresses

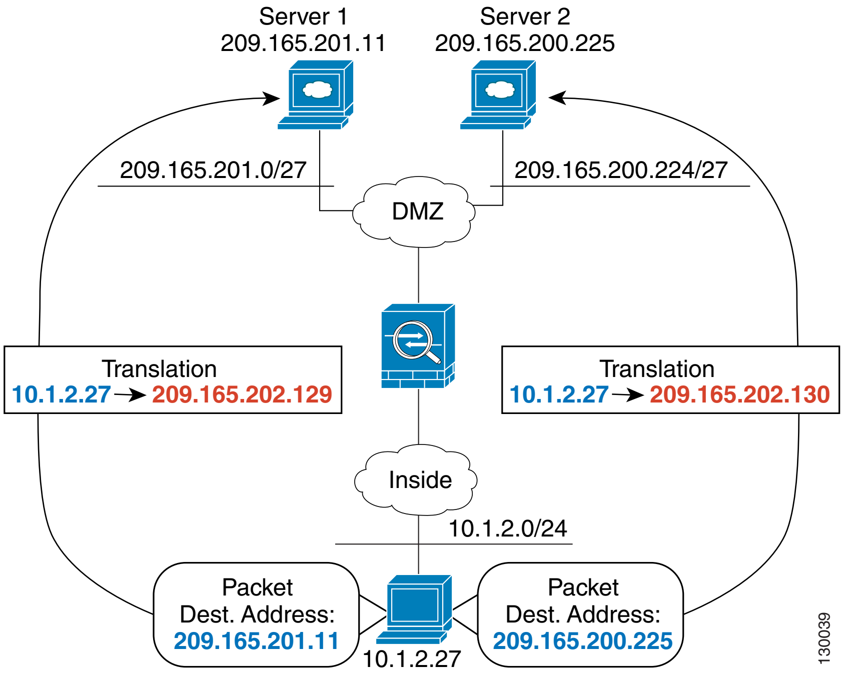


Figure 3-12 shows the use of source and destination ports. The host on the 10.1.2.0/24 network accesses a single host for both web services and Telnet services. When the host accesses the server for web services, the real address is translated to 209.165.202.129. When the host accesses the same server for Telnet services, the real address is translated to 209.165.202.130.

Figure 3-12 Twice NAT with Different Destination Ports

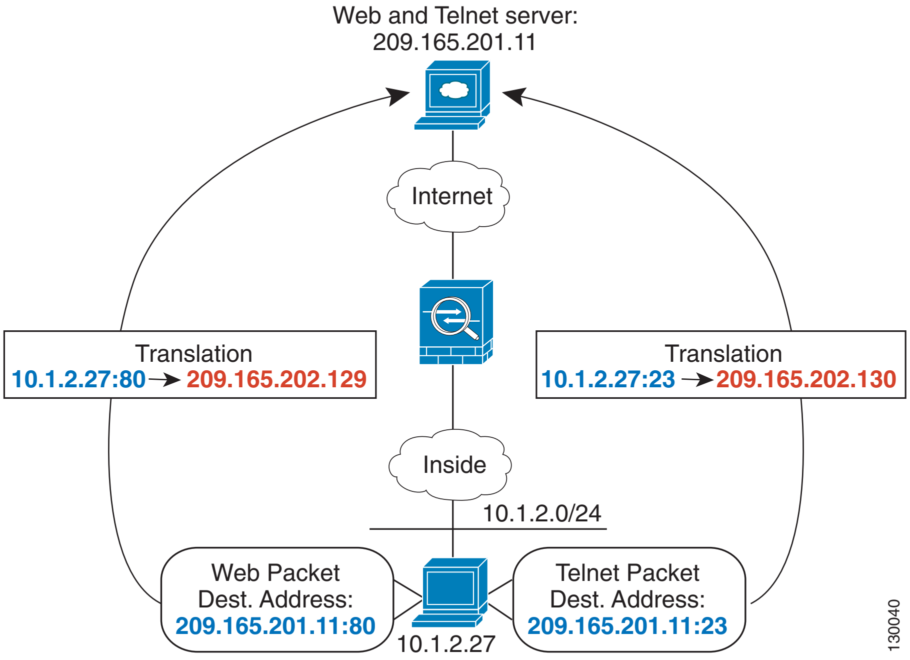
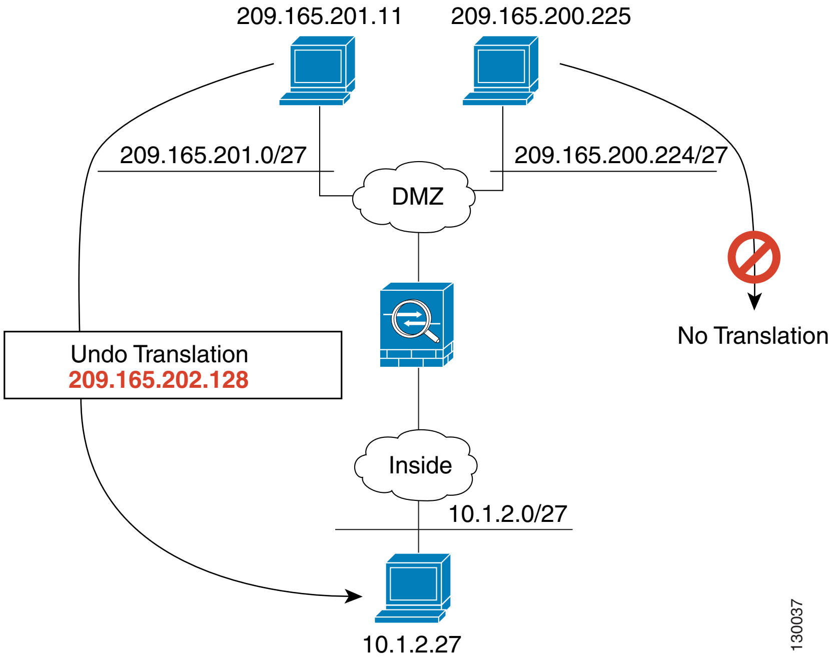


Figure 3-13 shows a remote host connecting to a mapped host. The mapped host has a twice static NAT translation that translates the real address only for traffic to and from the 209.165.201.0/27 network. A translation does not exist for the 209.165.200.224/27 network, so the translated host cannot connect to that network, nor can a host on that network connect to the translated host.

Figure 3-13 Twice Static NAT with Destination Address Translation



NAT Rule Order

Network object NAT rules and twice NAT rules are stored in a single table that is divided into three sections. Section 1 rules are applied first, then section 2, and finally section 3, until a match is found. For example, if a match is found in section 1, sections 2 and 3 are not evaluated. Table 3-1 shows the order of rules within each section.

Table 3-1 NAT Rule Table

Table Section	Rule Type	Order of Rules within the Section
Section 1	Twice NAT	Applied on a first match basis, in the order they appear in the configuration. Because the first match is applied, you must ensure that specific rules come before more general rules, or the specific rules might not be applied as desired. By default, twice NAT rules are added to section 1. **Note** If you configure EasyVPN remote, the ASA dynamically adds invisible NAT rules to the end of this section. Be sure that you do not configure a twice NAT rule in this section that might match your VPN traffic, instead of matching the invisible rule. If VPN does not work due to NAT failure, consider adding twice NAT rules to section 3 instead.
Section 2	Network object NAT	If a match in section 1 is not found, section 2 rules are applied in the following order, as automatically determined by the ASA: 1. Static rules. 2. Dynamic rules. Within each rule type, the following ordering guidelines are used: a. Quantity of real IP addresses—From smallest to largest. For example, an object with one address will be assessed before an object with 10 addresses. b. For quantities that are the same, then the IP address number is used, from lowest to highest. For example, 10.1.1.0 is assessed before 11.1.1.0. c. If the same IP address is used, then the name of the network object is used, in alphabetical order. For example, abracadabra is assessed before catwoman.
Section 3	Twice NAT	If a match is still not found, section 3 rules are applied on a first match basis, in the order they appear in the configuration. This section should contain your most general rules. You must also ensure that any specific rules in this section come before general rules that would otherwise apply. You can specify whether to add a twice NAT rule to section 3 when you add the rule.

For section 2 rules, for example, you have the following IP addresses defined within network objects:

192.168.1.0/24 (static)

192.168.1.0/24 (dynamic)

10.1.1.0/24 (static)

192.168.1.1/32 (static)

172.16.1.0/24 (dynamic) (object def)

172.16.1.0/24 (dynamic) (object abc)

The resultant ordering would be:

192.168.1.1/32 (static)

10.1.1.0/24 (static)

192.168.1.0/24 (static)

172.16.1.0/24 (dynamic) (object abc)

172.16.1.0/24 (dynamic) (object def)

192.168.1.0/24 (dynamic)

NAT Interfaces

You can configure a NAT rule to apply to any interface (in other words, all interfaces), or you can identify specific real and mapped interfaces. You can also specify any interface for the real address, and a specific interface for the mapped address, or vice versa.

For example, you might want to specify any interface for the real address and specify the outside interface for the mapped address if you use the same private addresses on multiple interfaces, and you want to translate them all to the same global pool when accessing the outside.

Note For transparent mode, you must choose specific source and destination interfaces.

Routing NAT Packets

The ASA needs to be the destination for any packets sent to the mapped address. The ASA also needs to determine the egress interface for any packets it receives destined for mapped addresses. This section describes how the ASA handles accepting and delivering packets with NAT, and includes the following topics:

- Mapped Addresses and Routing, page 3-20
- Transparent Mode Routing Requirements for Remote Networks, page 3-21
- Determining the Egress Interface, page 3-22

Mapped Addresses and Routing

When you translate the real address to a mapped address, the mapped address you choose determines how to configure routing, if necessary, for the mapped address.

See additional guidelines about mapped IP addresses in Chapter 4, “Configuring Network Object NAT,” and Chapter 5, “Configuring Twice NAT.”

See the following mapped address types:

- Addresses on the same network as the mapped interface.

  If you use addresses on the same network as the mapped interface, the ASA uses proxy ARP to answer any ARP requests for the mapped addresses, thus intercepting traffic destined for a mapped address. This solution simplifies routing because the ASA does not have to be the gateway for any additional networks. This solution is ideal if the outside network contains an adequate number of free addresses, a consideration if you are using a 1:1 translation like dynamic NAT or static NAT. Dynamic PAT greatly extends the number of translations you can use with a small number of addresses, so even if the available addresses on the outside network is small, this method can be used. For PAT, you can even use the IP address of the mapped interface.

  Note If you configure the mapped interface to be any interface, and you specify a mapped address on the same network as one of the mapped interfaces, then if an ARP request for that mapped address comes in on a *different* interface, then you need to manually configure an ARP entry for that network on the ingress interface, specifying its MAC address (see the **arp** command). Typically, if you specify any interface for the mapped interface, then you use a unique network for the mapped addresses, so this situation would not occur.

- Addresses on a unique network.

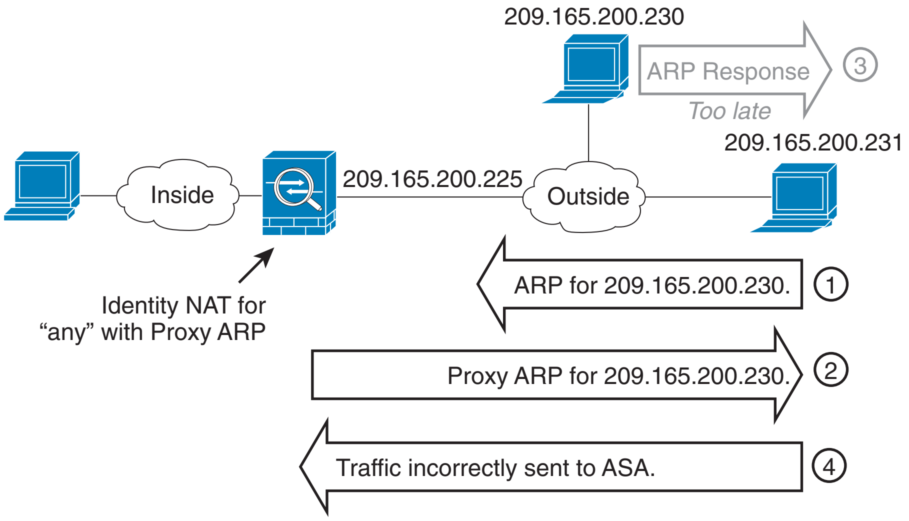
  If you need more addresses than are available on the mapped interface network, you can identify addresses on a different subnet. The upstream router needs a static route for the mapped addresses that points to the ASA. Alternatively for routed mode, you can configure a static route on the ASA for the mapped addresses, and then redistribute the route using your routing protocol. For transparent mode, if the real host is directly-connected, configure the static route on the upstream router to point to the ASA: specify the bridge group IP address. For remote hosts in transparent mode, in the static route on the upstream router, you can alternatively specify the downstream router IP address.

- The same address as the real address (identity NAT).

  The default behavior for identity NAT has proxy ARP enabled, matching other static NAT rules. You can disable proxy ARP if desired. **Note**: You can also disable proxy ARP for regular static NAT if desired, in which case you need to be sure to have proper routes on the upstream router.

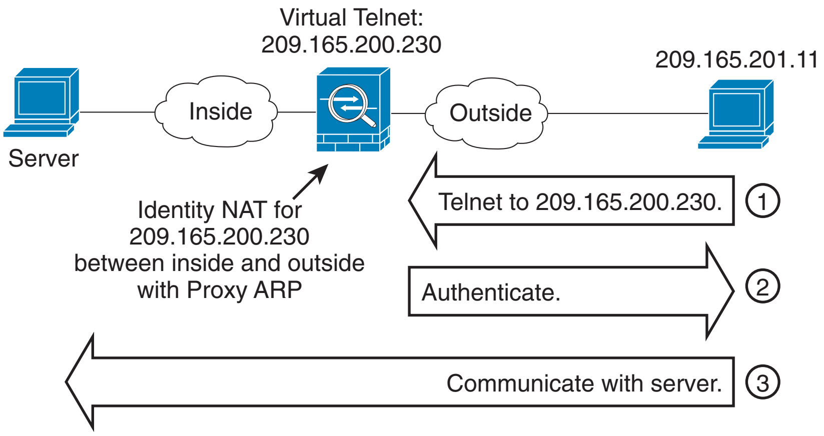
  Normally for identity NAT, proxy ARP is not required, and in some cases can cause connectivity issues. For example, if you configure a broad identity NAT rule for “any” IP address, then leaving proxy ARP enabled can cause problems for hosts on the network directly-connected to the mapped interface. In this case, when a host on the mapped network wants to communicate with another host on the same network, then the address in the ARP request matches the NAT rule (which matches “any” address). The ASA will then proxy ARP for the address, even though the packet is not actually destined for the ASA. (Note that this problem occurs even if you have a twice NAT rule; although the NAT rule must match both the source and destination addresses, the proxy ARP decision is made only on the “source” address). If the ASA ARP response is received before the actual host ARP response, then traffic will be mistakenly sent to the ASA (see Figure 3-14).

Figure 3-14 Proxy ARP Problems with Identity NAT



In rare cases, you need proxy ARP for identity NAT; for example for virtual Telnet. When using AAA for network access, a host needs to authenticate with the ASA using a service like Telnet before any other traffic can pass. You can configure a virtual Telnet server on the ASA to provide the necessary login. When accessing the virtual Telnet address from the outside, you must configure an identity NAT rule for the address specifically for the proxy ARP functionality. Due to internal processes for virtual Telnet, proxy ARP lets the ASA keep traffic destined for the virtual Telnet address rather than send the traffic out the source interface according to the NAT rule. (See Figure 3-15).

Figure 3-15 Proxy ARP and Virtual Telnet



Transparent Mode Routing Requirements for Remote Networks

When you use NAT in transparent mode, some types of traffic require static routes. See the general operations configuration guide for more information.

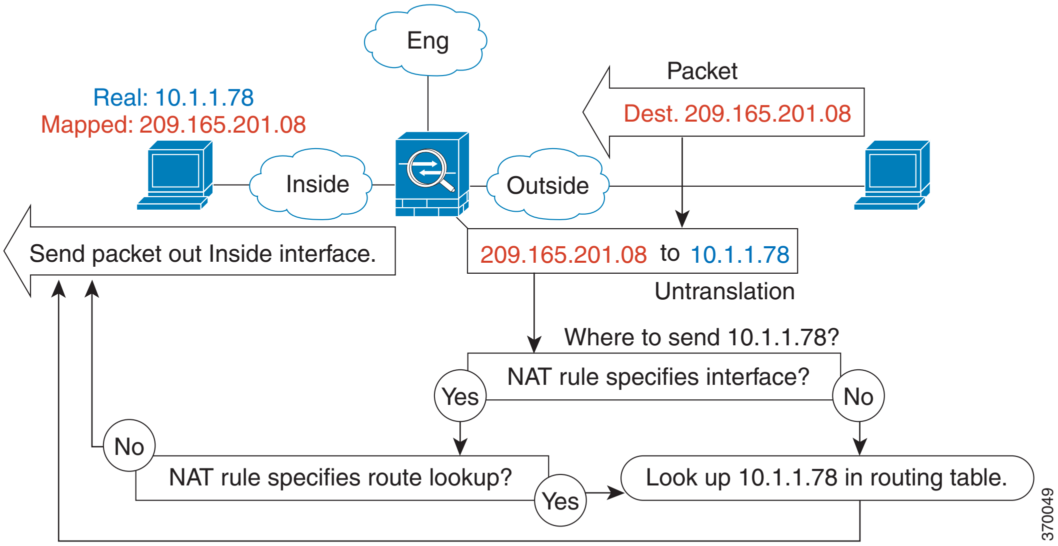
Determining the Egress Interface

When the ASA receives traffic for a mapped address, the ASA unstranslates the destination address according to the NAT rule, and then it sends the packet on to the real address. The ASA determines the egress interface for the packet in the following ways:

- Transparent mode—The ASA determines the egress interface for the real address by using the NAT rule; you must specify the source and destination interfaces as part of the NAT rule.
- Routed mode—The ASA determines the egress interface in one of the following ways:
  - You configure the interface in the NAT rule—The ASA uses the NAT rule to determine the egress interface. However, you have the option to always use a route lookup instead. In certain scenarios, a route lookup override is required; for example, see the "NAT and VPN Management Access" section on page 3-26.
  - You do not configure the interface in the NAT rule—The ASA uses a route lookup to determine the egress interface.

Figure 3-16 shows the egress interface selection method in routed mode. In almost all cases, a route lookup is equivalent to the NAT rule interface, but in some configurations, the two methods might differ.

Figure 3-16 Routed Mode Egress Interface Selection


Eng
Packet
Real: 10.1.1.78
Mapped: 209.165.201.08
Dest. 209.165.201.08
Inside
Outside
Send packet out Inside interface.
209.165.201.08 to 10.1.1.78
Untranslation
Where to send 10.1.1.78?
NAT rule specifies interface?
Yes
No
No
NAT rule specifies route lookup?
Yes
Look up 10.1.1.78 in routing table.
370049

NAT for VPN

- NAT and Remote Access VPN, page 3-23
- NAT and Site-to-Site VPN, page 3-24
- NAT and VPN Management Access, page 3-26
- Troubleshooting NAT and VPN, page 3-28

NAT and Remote Access VPN

Figure 3-17 shows both an inside server (10.1.1.6) and a VPN client (209.165.201.10) accessing the Internet. Unless you configure split tunnelling for the VPN client (where only specified traffic goes through the VPN tunnel), then Internet-bound VPN traffic must also go through the ASA. When the VPN traffic enters the ASA, the ASA decrypts the packet; the resulting packet includes the VPN client local address (10.3.3.10) as the source. For both inside and VPN client local networks, you need a public IP address provided by NAT to access the Internet. The below example uses interface PAT rules. To allow the VPN traffic to exit the same interface it entered, you also need to enable intra-interface communication (AKA "hairpin" networking).

Figure 3-17 Interface PAT for Internet-Bound VPN Traffic (Intra-Interface)

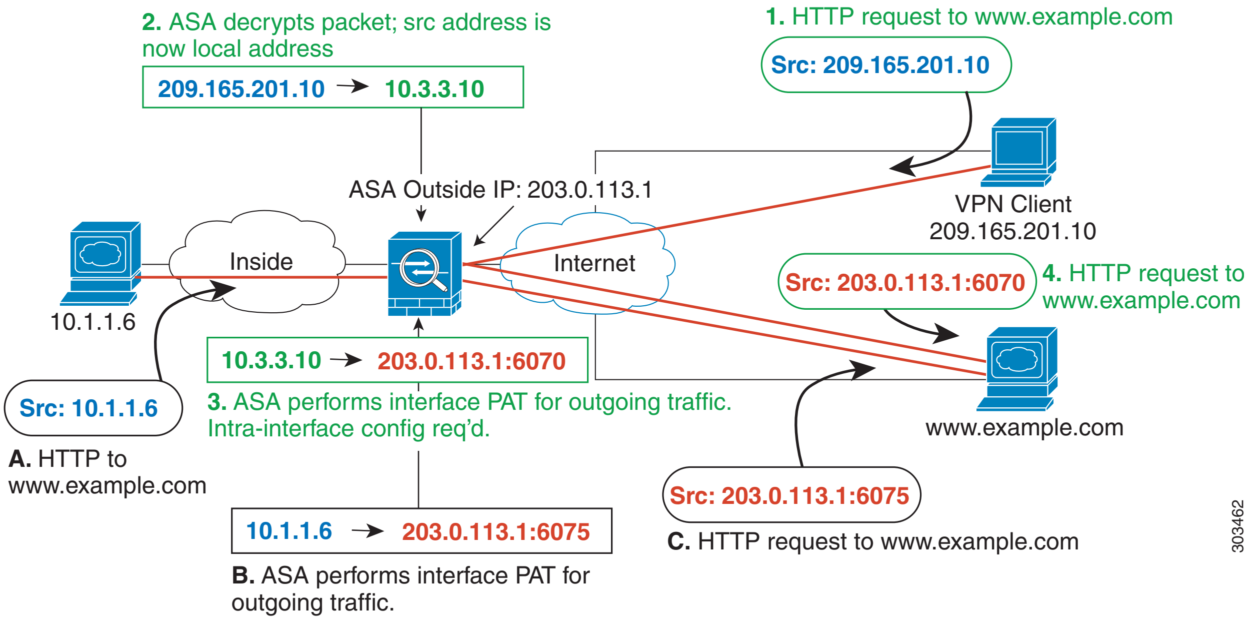
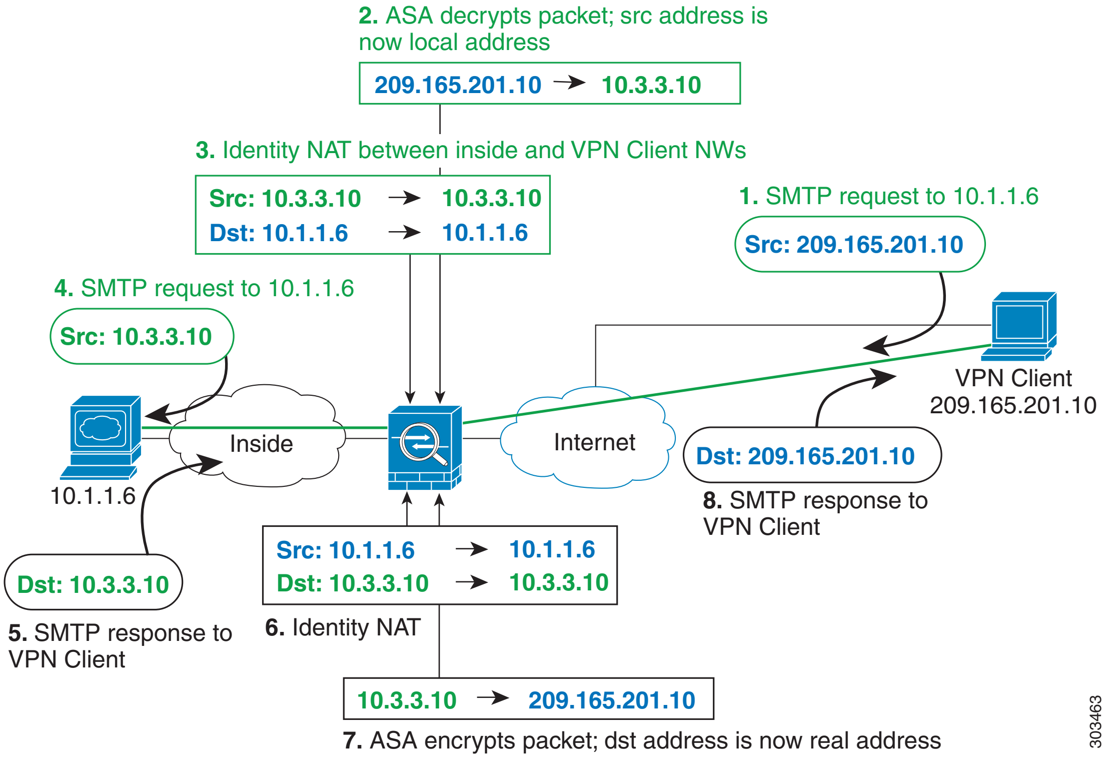

2. ASA decrypts packet; src address is now local address
209.165.201.10 → 10.3.3.10
1. HTTP request to www.example.com
Src: 209.165.201.10
ASA Outside IP: 203.0.113.1
VPN Client
209.165.201.10
Inside
Internet
10.1.1.6
Src: 203.0.113.1:6070
4. HTTP request to www.example.com
10.3.3.10 → 203.0.113.1:6070
Src: 10.1.1.6
3. ASA performs interface PAT for outgoing traffic. Intra-interface config req'd.
www.example.com
A. HTTP to www.example.com
Src: 203.0.113.1:6075
10.1.1.6 → 203.0.113.1:6075
C. HTTP request to www.example.com
B. ASA performs interface PAT for outgoing traffic.
303462

Figure 3-18 shows a VPN client that wants to access an inside mail server. Because the ASA expects traffic between the inside network and any outside network to match the interface PAT rule you set up for Internet access, traffic from the VPN client (10.3.3.10) to the SMTP server (10.1.1.6) will be dropped due to a reverse path failure: traffic from 10.3.3.10 to 10.1.1.6 does not match a NAT rule, but returning traffic from 10.1.1.6 to 10.3.3.10 *should* match the interface PAT rule for outgoing traffic. Because forward and reverse flows do not match, the ASA drops the packet when it is received. To avoid this failure, you need to exempt the inside-to-VPN client traffic from the interface PAT rule by using an identity NAT rule between those networks. Identity NAT simply translates an address to the same address.

Figure 3-18 Identity NAT for VPN Clients


2. ASA decrypts packet; src address is now local address
209.165.201.10 → 10.3.3.10
3. Identity NAT between inside and VPN Client NWs
Src: 10.3.3.10 → 10.3.3.10
Dst: 10.1.1.6 → 10.1.1.6
1. SMTP request to 10.1.1.6
Src: 209.165.201.10
4. SMTP request to 10.1.1.6
Src: 10.3.3.10
VPN Client
209.165.201.10
Inside
Internet
10.1.1.6
Dst: 209.165.201.10
8. SMTP response to VPN Client
Src: 10.1.1.6 → 10.1.1.6
Dst: 10.3.3.10 → 10.3.3.10
Dst: 10.3.3.10
5. SMTP response to VPN Client
6. Identity NAT
10.3.3.10 → 209.165.201.10
7. ASA encrypts packet; dst address is now real address
303463

See the following sample NAT configuration for the above network:

```
! Enable hairpin for non-split-tunneled VPN client traffic:
same-security-traffic permit intra-interface

! Identify local VPN network, & perform object interface PAT when going to Internet:
object network vpn_local
    subnet 10.3.3.0 255.255.255.0
    nat (outside,outside) dynamic interface

! Identify inside network, & perform object interface PAT when going to Internet:
object network inside_nw
    subnet 10.1.1.0 255.255.255.0
    nat (inside,outside) dynamic interface

! Use twice NAT to pass traffic between the inside network and the VPN client without
! address translation (identity NAT):
nat (inside,outside) source static inside_nw inside_nw destination static vpn_local
vpn_local
```

NAT and Site-to-Site VPN

Figure 3-19 shows a site-to-site tunnel connecting the Boulder and San Jose offices. For traffic that you want to go to the Internet (for example from 10.1.1.6 in Boulder to www.example.com), you need a public IP address provided by NAT to access the Internet. The below example uses interface PAT rules. However, for traffic that you want to go over the VPN tunnel (for example from 10.1.1.6 in Boulder to 10.2.2.78 in San Jose), you do not want to perform NAT; you need to exempt that traffic by creating an identity NAT rule. Identity NAT simply translates an address to the same address.

Figure 3-19 Interface PAT and Identity NAT for Site-to-Site VPN

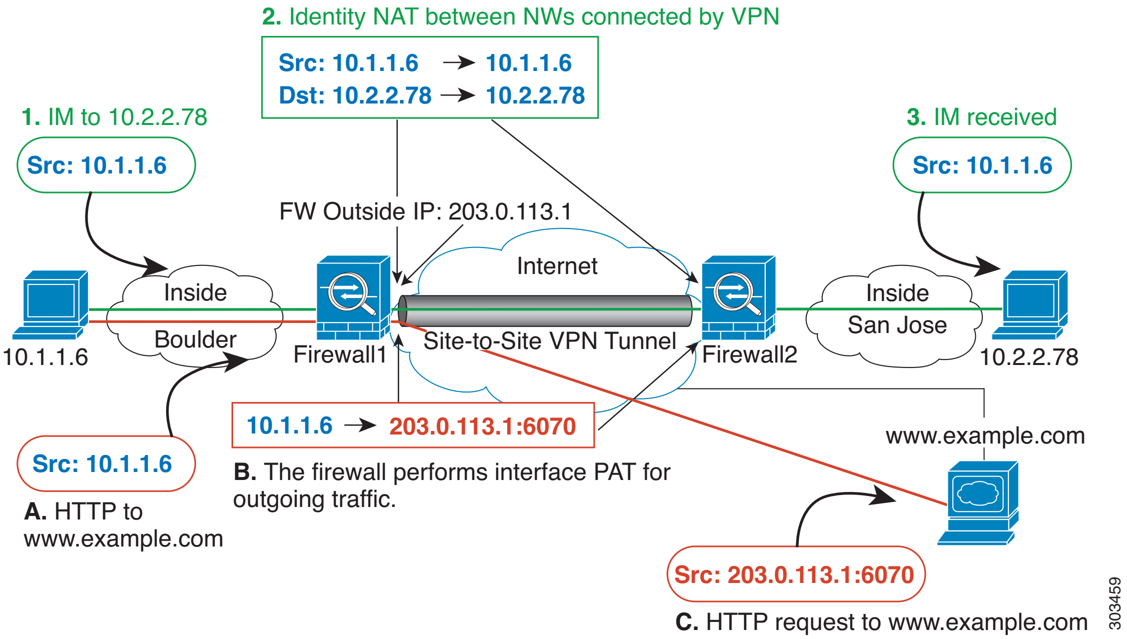
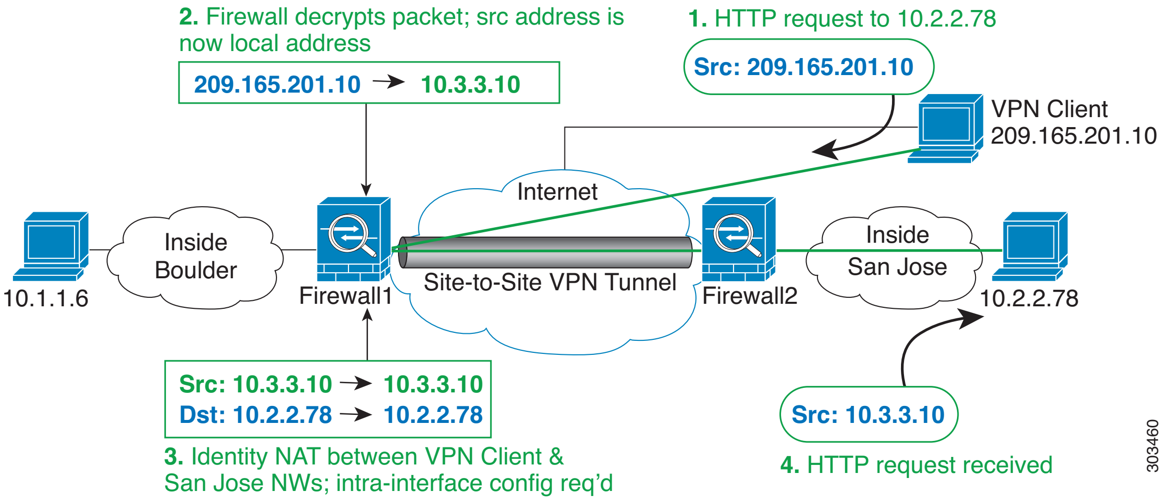


Figure 3-20 shows a VPN client connected to ASA1 (Boulder), with a Telnet request for a server (10.2.2.78) accessible over a site-to-site tunnel between ASA1 and ASA2 (San Jose). Because this is a hairpin connection, you need to enable intra-interface communication, which is also required for non-split-tunneled Internet-bound traffic from the VPN client. You also need to configure identity NAT between the VPN client and the Boulder & San Jose networks, just as you would between any networks connected by VPN to exempt this traffic from outbound NAT rules.

Figure 3-20 VPN Client Access to Site-to-Site VPN


2. Firewall decrypts packet; src address is now local address
209.165.201.10 → 10.3.3.10
1. HTTP request to 10.2.2.78
Src: 209.165.201.10
VPN Client
209.165.201.10
Internet
Inside
Boulder
10.1.1.6
Firewall1
Site-to-Site VPN Tunnel
Firewall2
Inside
San Jose
10.2.2.78
Src: 10.3.3.10 → 10.3.3.10
Dst: 10.2.2.78 → 10.2.2.78
3. Identity NAT between VPN Client & San Jose NWs; intra-interface config req'd
Src: 10.3.3.10
4. HTTP request received
303460

See the following sample NAT configuration for ASA1 (Boulder):

```
! Enable hairpin for VPN client traffic:
same-security-traffic permit intra-interface

! Identify local VPN network, & perform object interface PAT when going to Internet:
```

```
object network vpn_local
    subnet 10.3.3.0 255.255.255.0
    nat (outside,outside) dynamic interface

! Identify inside Boulder network, & perform object interface PAT when going to Internet:
object network boulder_inside
    subnet 10.1.1.0 255.255.255.0
    nat (inside,outside) dynamic interface

! Identify inside San Jose network for use in twice NAT rule:
object network sanjose_inside
    subnet 10.2.2.0 255.255.255.0

! Use twice NAT to pass traffic between the Boulder network and the VPN client without
! address translation (identity NAT):
nat (inside,outside) source static boulder_inside boulder_inside destination static
vpn_local vpn_local

! Use twice NAT to pass traffic between the Boulder network and San Jose without
! address translation (identity NAT):
nat (inside,outside) source static boulder_inside boulder_inside destination static
sanjose_inside sanjose_inside

! Use twice NAT to pass traffic between the VPN client and San Jose without
! address translation (identity NAT):
nat (outside,outside) source static vpn_local vpn_local destination static sanjose_inside
sanjose_inside
```

See the following sample NAT configuration for ASA2 (San Jose):

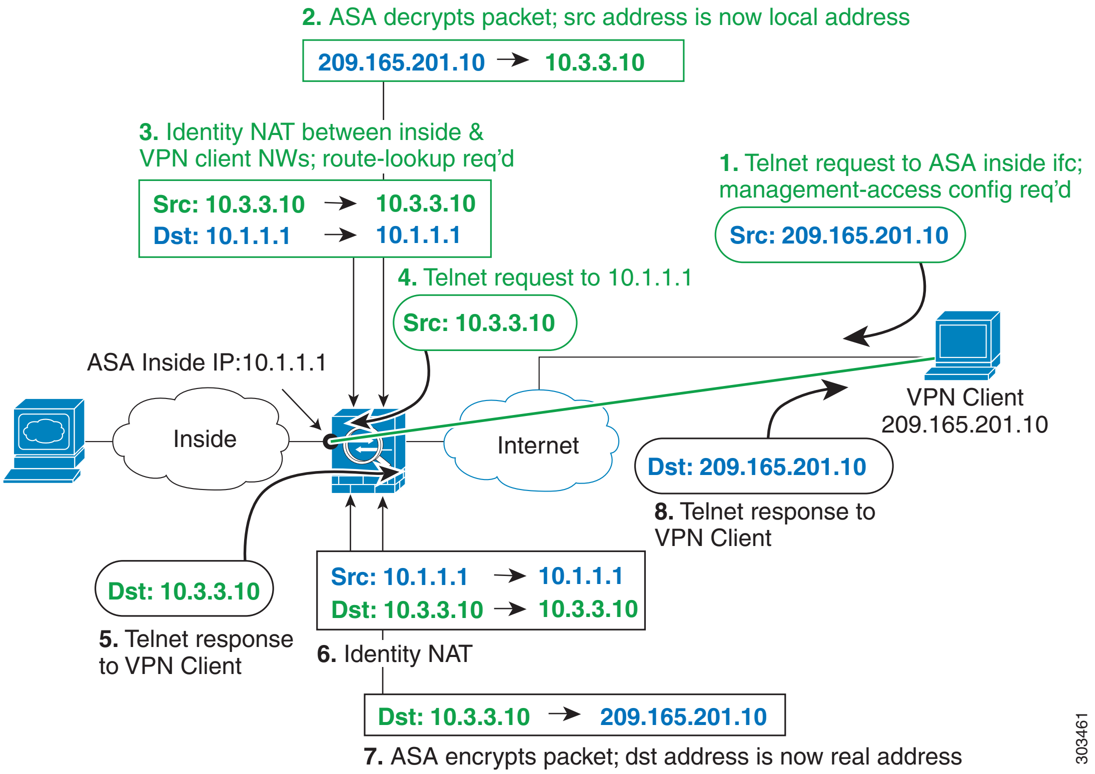
```
! Identify inside San Jose network, & perform object interface PAT when going to Internet:
object network sanjose_inside
    subnet 10.2.2.0 255.255.255.0
    nat (inside,outside) dynamic interface

! Identify inside Boulder network for use in twice NAT rule:
object network boulder_inside
    subnet 10.1.1.0 255.255.255.0

! Identify local VPN network for use in twice NAT rule:
object network vpn_local
    subnet 10.3.3.0 255.255.255.0

! Use twice NAT to pass traffic between the San Jose network and Boulder without
! address translation (identity NAT):
nat (inside,outside) source static sanjose_inside sanjose_inside destination static
boulder_inside boulder_inside

! Use twice NAT to pass traffic between the San Jose network and the VPN client without
! address translation (identity NAT):
nat (inside,outside) source static sanjose_inside sanjose_inside destination static
vpn_local vpn_local
```

NAT and VPN Management Access

When using VPN, you can allow management access to an interface other than the one from which you entered the ASA (see the **management-access** command). For example, if you enter the ASA from the outside interface, the management-access feature lets you connect to the inside interface using ASDM, SSH, Telnet, or SNMP; or you can ping the inside interface.

Figure 3-21 shows a VPN client Telnetting to the ASA inside interface. When you use a management-access interface, and you configure identity NAT according to the "NAT and Remote Access VPN" or "NAT and Site-to-Site VPN" section, you must configure NAT with the route lookup option. Without route lookup, the ASA sends traffic out the interface specified in the NAT command, regardless of what the routing table says; in the below example, the egress interface is the inside interface. You do not want the ASA to send the management traffic out to the inside network; it will never return to the inside interface IP address. The route lookup option lets the ASA send the traffic directly to the inside interface IP address instead of to the inside network. For traffic from the VPN client to a host on the inside network, the route lookup option will still result in the correct egress interface (inside), so normal traffic flow is not affected. See the "Determining the Egress Interface" section on page 3-22 for more information about the route lookup option.

Figure 3-21 VPN Management Access


2. ASA decrypts packet; src address is now local address
209.165.201.10 → 10.3.3.10
3. Identity NAT between inside & VPN client NWs; route-lookup req'd
Src: 10.3.3.10 → 10.3.3.10
Dst: 10.1.1.1 → 10.1.1.1
1. Telnet request to ASA inside ifc; management-access config req'd
Src: 209.165.201.10
4. Telnet request to 10.1.1.1
Src: 10.3.3.10
ASA Inside IP:10.1.1.1
Inside
Internet
VPN Client
209.165.201.10
Dst: 209.165.201.10
8. Telnet response to VPN Client
Dst: 10.3.3.10
5. Telnet response to VPN Client
Src: 10.1.1.1 → 10.1.1.1
Dst: 10.3.3.10 → 10.3.3.10
6. Identity NAT
Dst: 10.3.3.10 → 209.165.201.10
7. ASA encrypts packet; dst address is now real address
303461

See the following sample NAT configuration for the above network:

```
! Enable hairpin for non-split-tunneled VPN client traffic:
same-security-traffic permit intra-interface

! Enable management access on inside ifc:
management-access inside

! Identify local VPN network, & perform object interface PAT when going to Internet:
object network vpn_local
    subnet 10.3.3.0 255.255.255.0
    nat (outside,outside) dynamic interface

! Identify inside network, & perform object interface PAT when going to Internet:
object network inside_nw
    subnet 10.1.1.0 255.255.255.0
    nat (inside,outside) dynamic interface
```

```
! Use twice NAT to pass traffic between the inside network and the VPN client without
! address translation (identity NAT), w/route-lookup:
nat (outside,inside) source static vpn_local vpn_local destination static inside_nw
inside_nw route-lookup
```

Troubleshooting NAT and VPN

See the following monitoring tools for troubleshooting NAT issues with VPN:

- Packet tracer—When used correctly, a packet tracer shows which NAT rules a packet is hitting.
- **show nat detail**—Shows hit counts and untranslated traffic for a given NAT rule.
- **show conn all**—Lets you see active connections including to and from the box traffic.

To familiarize yourself with a non-working configuration vs. a working configuration, you can perform the following steps:

1. Configure VPN without identity NAT.
2. Enter **show nat detail** and **show conn all**.
3. Add the identity NAT configuration.

- Repeat **show nat detail** and **show conn all**.

DNS and NAT

You might need to configure the ASA to modify DNS replies by replacing the address in the reply with an address that matches the NAT configuration. You can configure DNS modification when you configure each translation rule.

This feature rewrites the address in DNS queries and replies that match a NAT rule (for example, the A record for IPv4, the AAAA record for IPv6, or the PTR record for reverse DNS queries). For DNS replies traversing from a mapped interface to any other interface, the record is rewritten from the mapped value to the real value. Inversely, for DNS replies traversing from any interface to a mapped interface, the record is rewritten from the real value to the mapped value.

Note DNS rewrite is not applicable for PAT because multiple PAT rules are applicable for each A-record, and the PAT rule to use is ambiguous.

Note If you configure a twice NAT rule, you cannot configure DNS modification if you specify the source address as well as the destination address. These kinds of rules can potentially have a different translation for a single address when going to A vs. B. Therefore, the ASA cannot accurately match the IP address inside the DNS reply to the correct twice NAT rule; the DNS reply does not contain information about which source/destination address combination was in the packet that prompted the DNS request.

Note This feature requires DNS application inspection to be enabled, which it is by default. See the "DNS Inspection" section on page 10-1 for more information.

Figure 3-22 shows a DNS server that is accessible from the outside interface. A server, ftp.cisco.com, is on the inside interface. You configure the ASA to statically translate the ftp.cisco.com real address (10.1.3.14) to a mapped address (209.165.201.10) that is visible on the outside network. In this case, you want to enable DNS reply modification on this static rule so that inside users who have access to ftp.cisco.com using the real address receive the real address from the DNS server, and not the mapped address. When an inside host sends a DNS request for the address of ftp.cisco.com, the DNS server replies with the mapped address (209.165.201.10). The ASA refers to the static rule for the inside server and translates the address inside the DNS reply to 10.1.3.14. If you do not enable DNS reply modification, then the inside host attempts to send traffic to 209.165.201.10 instead of accessing ftp.cisco.com directly.

Figure 3-22 DNS Reply Modification, DNS Server on Outside

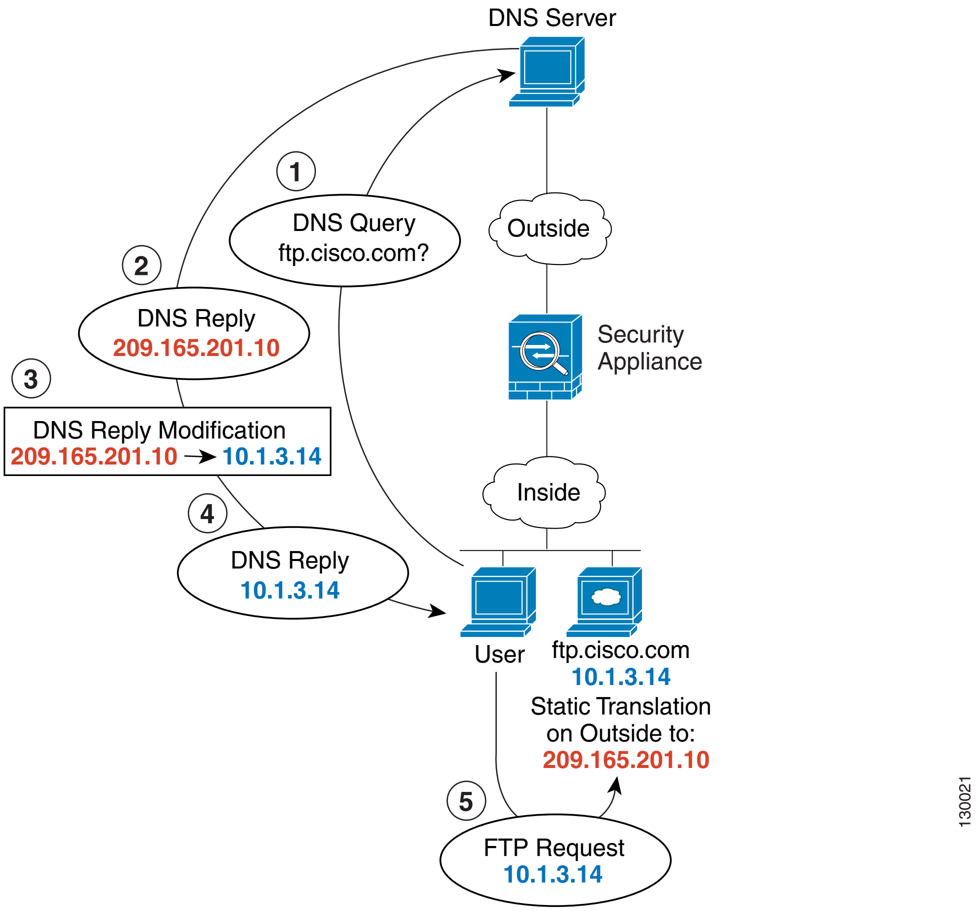


Figure 3-23 shows a user on the inside network requesting the IP address for ftp.cisco.com, which is on the DMZ network, from an outside DNS server. The DNS server replies with the mapped address (209.165.201.10) according to the static rule between outside and DMZ even though the user is not on the DMZ network. The ASA translates the address inside the DNS reply to 10.1.3.14. If the user needs to access ftp.cisco.com using the real address, then no further configuration is required. If there is also a static rule between the inside and DMZ, then you also need to enable DNS reply modification on this rule. The DNS reply will then be modified two times.In this case, the ASA again translates the address inside the DNS reply to 192.168.1.10 according to the static rule between inside and DMZ.

Figure 3-23 DNS Reply Modification, DNS Server, Host, and Server on Separate Networks

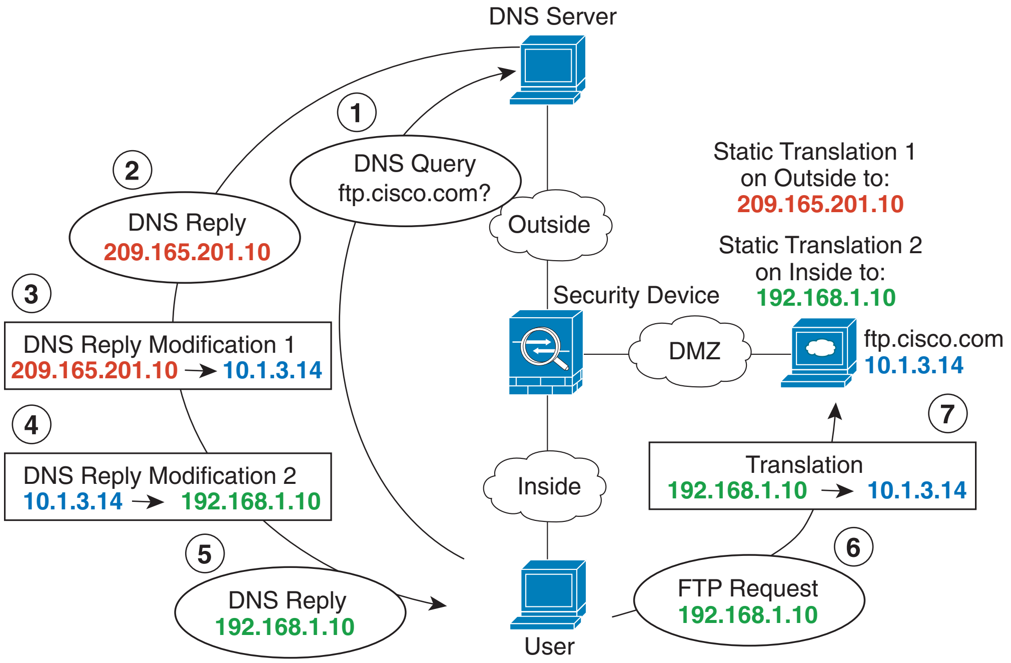


Figure 3-24 shows an FTP server and DNS server on the outside. The ASA has a static translation for the outside server. In this case, when an inside user requests the address for ftp.cisco.com from the DNS server, the DNS server responds with the real address, 209.165.20.10. Because you want inside users to use the mapped address for ftp.cisco.com (10.1.2.56) you need to configure DNS reply modification for the static translation.

Figure 3-24 DNS Reply Modification, DNS Server on Host Network

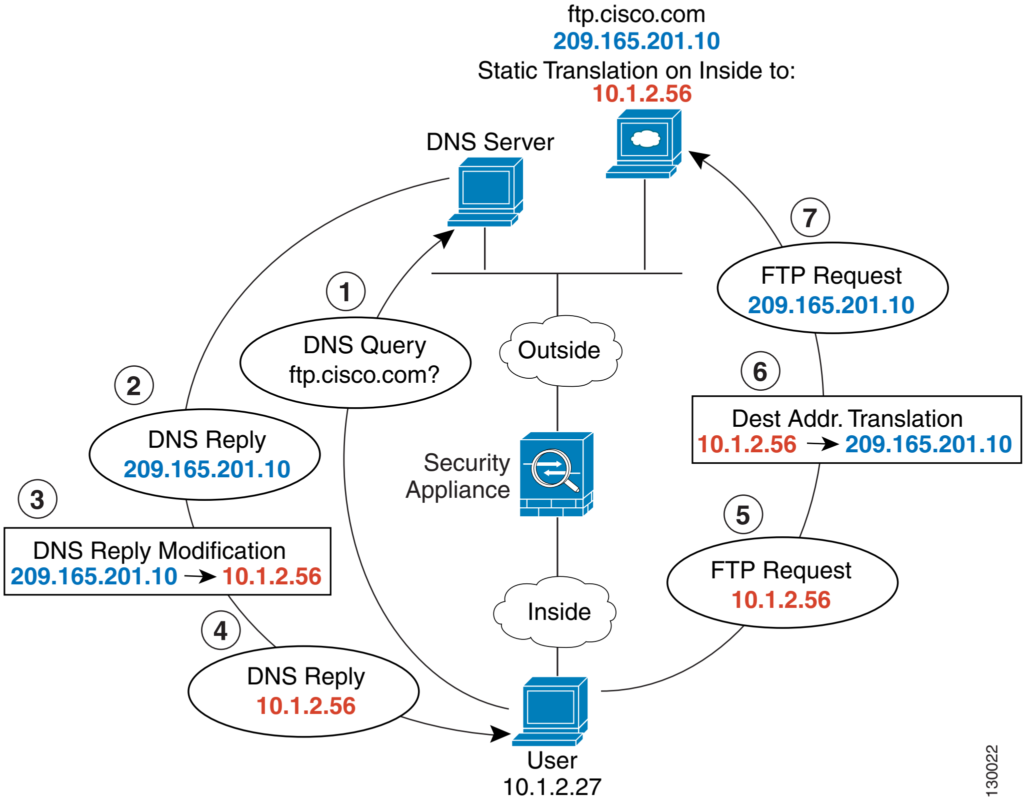


Figure 3-24 shows an FTP server and DNS server on the outside IPv4 network. The ASA has a static translation for the outside server. In this case, when an inside IPv6 user requests the address for ftp.cisco.com from the DNS server, the DNS server responds with the real address, 209.165.200.225.

Because you want inside users to use the mapped address for ftp.cisco.com (2001:DB8::D1A5:C8E1) you need to configure DNS reply modification for the static translation. This example also includes a static NAT translation for the DNS server, and a PAT rule for the inside IPv6 hosts.

Figure 3-25 DNS64 Reply Modification Using Outside NAT

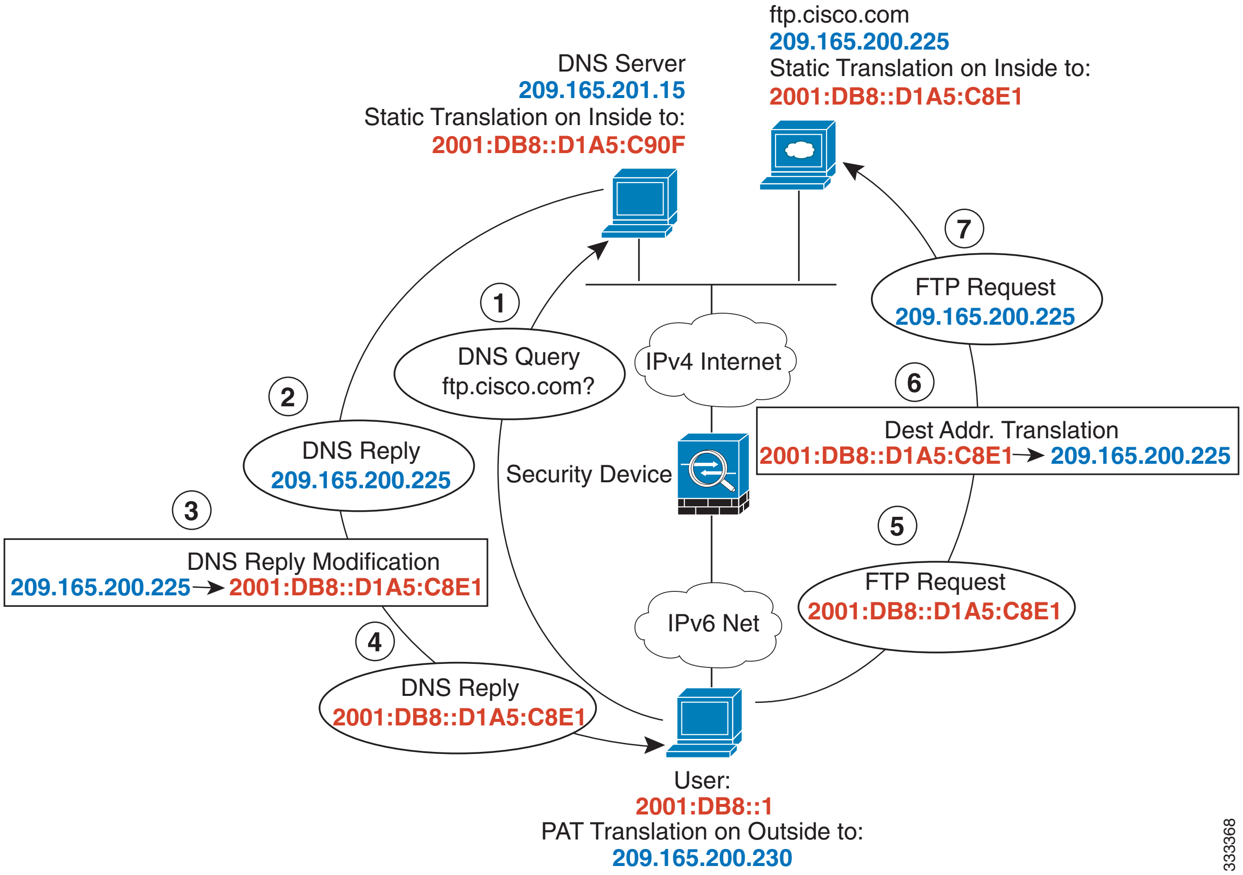
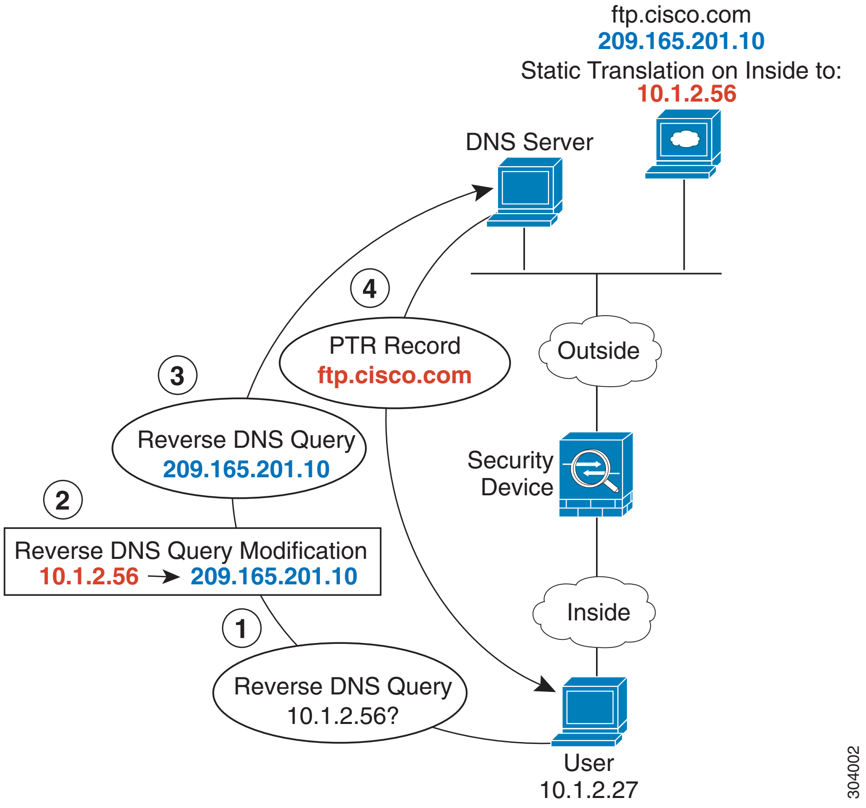


Figure 3-26 shows an FTP server and DNS server on the outside. The ASA has a static translation for the outside server. In this case, when an inside user performs a reverse DNS lookup for 10.1.2.56, the ASA modifies the reverse DNS query with the real address, and the DNS server responds with the server name, ftp.cisco.com.

Figure 3-26 PTR Modification, DNS Server on Host Network



Where to Go Next

To configure network object NAT, see Chapter 4, "Configuring Network Object NAT."

To configure twice NAT, see Chapter 5, "Configuring Twice NAT."

CHAPTER 4

Configuring Network Object NAT

All NAT rules that are configured as a parameter of a network object are considered to be *network object NAT* rules. Network object NAT is a quick and easy way to configure NAT for a single IP address, a range of addresses, or a subnet. After you configure the network object, you can then identify the mapped address for that object.

This chapter describes how to configure network object NAT, and it includes the following sections:

Note For detailed information about how NAT works, see Chapter 3, "Information About NAT."

Information About Network Object NAT

When a packet enters the ASA, both the source and destination IP addresses are checked against the network object NAT rules. The source and destination address in the packet can be translated by separate rules if separate matches are made. These rules are not tied to each other; different combinations of rules can be used depending on the traffic.

Because the rules are never paired, you cannot specify that a source address should be translated to A when going to destination X, but be translated to B when going to destination Y. Use twice NAT for that kind of functionality (twice NAT lets you identify the source and destination address in a single rule).

For detailed information about the differences between twice NAT and network object NAT, see the "How NAT is Implemented" section on page 3-13.

Network object NAT rules are added to section 2 of the NAT rules table. For more information about NAT ordering, see the "NAT Rule Order" section on page 3-18.

Licensing Requirements for Network Object NAT

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Prerequisites for Network Object NAT

Depending on the configuration, you can configure the mapped address inline if desired or you can create a separate network object or network object group for the mapped address (the **object network** or **object-group network** command). Network object groups are particularly useful for creating a mapped address pool with discontinous IP address ranges or multiple hosts or subnets. To create a network object or group, see the general operations configuration guide.

For specific guidelines for objects and groups, see the configuration section for the NAT type you want to configure. See also the "Guidelines and Limitations" section.

Guidelines and Limitations

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

- Supported in routed and transparent firewall mode.
- In transparent mode, you must specify the real and mapped interfaces; you cannot use **any**.
- In transparent mode, you cannot configure interface PAT, because the transparent mode interfaces do not have IP addresses. You also cannot use the management IP address as a mapped address.
- In transparent mode, translating between IPv4 and IPv6 networks is not supported. Translating between two IPv6 networks, or between two IPv4 networks is supported.

IPv6 Guidelines

- Supports IPv6. See also the "NAT and IPv6" section on page 3-13.
- For routed mode, you can also translate between IPv4 and IPv6.
- For transparent mode, translating between IPv4 and IPv6 networks is not supported. Translating between two IPv6 networks, or between two IPv4 networks is supported.
- For transparent mode, a PAT pool is not supported for IPv6.
- For static NAT, you can specify an IPv6 subnet up to /64. Larger subnets are not supported.
- When using FTP with NAT46, when an IPv4 FTP client connects to an IPv6 FTP server, the client must use either the extended passive mode (EPSV) or extended port mode (EPRT); PASV and PORT commands are not supported with IPv6.

Additional Guidelines

- You can only define a single NAT rule for a given object; if you want to configure multiple NAT rules for an object, you need to create multiple objects with different names that specify the same IP address, for example, **object network obj-10.10.10.1-01**, **object network obj-10.10.10.1-02**, and so on.
- If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT configuration is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections that use translations.

Note If you remove a dynamic NAT or PAT rule, and then add a new rule with mapped addresses that overlap the addresses in the removed rule, then the new rule will not be used until all connections associated with the removed rule time out or are cleared using the **clear xlate** command. This safeguard ensures that the same address is not assigned to multiple hosts.

- Objects and object groups used in NAT cannot be undefined; they must include IP addresses.
- You cannot use an object group with both IPv4 and IPv6 addresses; the object group must include only one type of address.
- You can use the same mapped object or group in multiple NAT rules.
- The mapped IP address pool cannot include:
  - The mapped interface IP address. If you specify **any** interface for the rule, then all interface IP addresses are disallowed. For interface PAT (routed mode only), use the **interface** keyword instead of the IP address.
  - (Transparent mode) The management IP address.
  - (Dynamic NAT) The standby interface IP address when VPN is enabled.
  - Existing VPN pool addresses.
- For application inspection limitations with NAT or PAT, see the "Default Settings and NAT Limitations" section on page 9-4 in Chapter 9, "Getting Started with Application Layer Protocol Inspection."

Default Settings

- (Routed mode) The default real and mapped interface is Any, which applies the rule to all interfaces.
- The default behavior for identity NAT has proxy ARP enabled, matching other static NAT rules. You can disable proxy ARP if desired. See the "Routing NAT Packets" section on page 3-19 for more information.
- If you specify an optional interface, then the ASA uses the NAT configuration to determine the egress interface, but you have the option to always use a route lookup instead. See the "Routing NAT Packets" section on page 3-19 for more information.

Configuring Network Object NAT

This section describes how to configure network object NAT and includes the following topics:

- Adding Network Objects for Mapped Addresses, page 4-4
- Configuring Dynamic NAT, page 4-5
- Configuring Dynamic PAT (Hide), page 4-7
- Configuring Static NAT or Static NAT-with-Port-Translation, page 4-11
- Configuring Identity NAT, page 4-14
- Configuring Per-Session PAT Rules, page 4-16

Adding Network Objects for Mapped Addresses

For dynamic NAT, you must use an object or group for the mapped addresses. Other NAT types have the option of using inline addresses, or you can create an object or group according to this section. For more information about configuring a network object or group, see the general operations configuration guide.

Guidelines

- A network object group can contain objects and/or inline addresses of either IPv4 or IPv6 addresses. The group cannot contain both IPv4 and IPv6 addresses; it must contain one type only.
- See the "Guidelines and Limitations" section on page 4-2 for information about disallowed mapped IP addresses.
- Dynamic NAT:
  - You cannot use an inline address; you must configure a network object or group.
  - The object or group cannot contain a subnet; the object must define a range; the group can include hosts and ranges.
  - If a mapped network object contains both ranges and host IP addresses, then the ranges are used for dynamic NAT, and then the host IP addresses are used as a PAT fallback.
- Dynamic PAT (Hide):
  - Instead of using an object, you can optionally configure an inline host address or specify the interface address.
  - If you use an object, the object or group cannot contain a subnet; the object must define a host, or for a PAT pool, a range; the group (for a PAT pool) can include hosts and ranges.
- Static NAT or Static NAT with port translation:
  - Instead of using an object, you can configure an inline address or specify the interface address (for static NAT-with-port-translation).
  - If you use an object, the object or group can contain a host, range, or subnet.
- Identity NAT
  - Instead of using an object, you can configure an inline address.
  - If you use an object, the object must match the real addresses you want to translate.

Detailed Steps

Command	Purpose
object network *obj_name* {**host** *ip_address* | **range** *ip_address_1* *ip_address_2* | **subnet** *subnet_address* *netmask*} **Example:** ciscoasa(config)# object network TEST ciscoasa(config-network-object)# range 10.1.1.1 10.1.1.70	Adds a network object, either IPv4 or IPv6.
object-group network *grp_name* {**network-object** {**object** *net_obj_name* | *subnet_address netmask* | **host** *ip_address*} | **group-object** *grp_obj_name*} **Example:** ciscoasa(config)# object network TEST ciscoasa(config-network-object)# range 10.1.1.1 10.1.1.70 ciscoasa(config)# object network TEST2 ciscoasa(config-network-object)# range 10.1.2.1 10.1.2.70 ciscoasa(config-network-object)# object-group network MAPPED_IPS ciscoasa(config-network)# network-object object TEST ciscoasa(config-network)# network-object object TEST2 ciscoasa(config-network)# network-object host 10.1.2.79	Adds a network object group, either IPv4 or IPv6.

Configuring Dynamic NAT

This section describes how to configure network object NAT for dynamic NAT. For more information, see the "Dynamic NAT" section on page 3-7.

Detailed Steps

	Command	Purpose
Step 1	Create a network object or group for the mapped addresses.	See the "Adding Network Objects for Mapped Addresses" section on page 4-4.
Step 2	**object network** *obj_name* **Example:** ciscoasa(config)# object network my-host-obj1	Configures a network object for which you want to configure NAT, or enters object network configuration mode for an existing network object.

	Command	Purpose
Step 3	{**host** *ip_address* | **subnet** *subnet_address netmask* | **range** *ip_address_1 ip_address_2*} **Example:** ciscoasa(config-network-object)# subnet 10.1.1.0 255.255.255.0	If you are creating a new network object, defines the **real** IP address(es) (either IPv4 or IPv6) that you want to translate.
Step 4	**nat** [**(***real_ifc***,***mapped_ifc***)**] **dynamic** *mapped_obj* [**interface** [**ipv6**]] [**dns**] **Example:** ciscoasa(config-network-object)# nat (inside,outside) dynamic MAPPED_IPS interface	Configures **dynamic NAT** for the object IP addresses. **Note** You can only define a single NAT rule for a given object. See the "Additional Guidelines" section on page 4-3. See the following guidelines: • Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used; you can also specify the keyword **any** for one or both of the interfaces. • Mapped IP address—Specify the mapped IP address as: – An existing network object (see Step 1). – An existing network object group (see Step 1). • Interface PAT fallback—(Optional) The **interface** keyword enables interface PAT fallback. After the mapped IP addresses are used up, then the IP address of the mapped interface is used. If you specify **ipv6**, then the IPv6 address of the interface is used. For this option, you must configure a specific interface for the *mapped_ifc*. (You cannot specify **interface** in transparent mode). • DNS—(Optional) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See the "DNS and NAT" section on page 3-28 for more information.

Examples

The following example configures dynamic NAT that hides 192.168.2.0 network behind a range of outside addresses 10.2.2.1 through 10.2.2.10:

```
ciscoasa(config)# object network my-range-obj
ciscoasa(config-network-object)# range 10.2.2.1 10.2.2.10
ciscoasa(config)# object network my-inside-net
ciscoasa(config-network-object)# subnet 192.168.2.0 255.255.255.0
ciscoasa(config-network-object)# nat (inside,outside) dynamic my-range-obj
```

The following example configures dynamic NAT with dynamic PAT backup. Hosts on inside network 10.76.11.0 are mapped first to the nat-range1 pool (10.10.10.10-10.10.10.20). After all addresses in the nat-range1 pool are allocated, dynamic PAT is performed using the pat-ip1 address (10.10.10.21). In the unlikely event that the PAT translations are also used up, dynamic PAT is performed using the outside interface address.

```
ciscoasa(config)# object network nat-range1
ciscoasa(config-network-object)# range 10.10.10.10 10.10.10.20

ciscoasa(config-network-object)# object network pat-ip1
ciscoasa(config-network-object)# host 10.10.10.21

ciscoasa(config-network-object)# object-group network nat-pat-grp
ciscoasa(config-network-object)# network-object object nat-range1
ciscoasa(config-network-object)# network-object object pat-ip1

ciscoasa(config-network-object)# object network my_net_obj5
ciscoasa(config-network-object)# subnet 10.76.11.0 255.255.255.0
ciscoasa(config-network-object)# nat (inside,outside) dynamic nat-pat-grp interface
```

The following example configures dynamic NAT with dynamic PAT backup to translate IPv6 hosts to IPv4. Hosts on inside network 2001:DB8::/96 are mapped first to the IPv4_NAT_RANGE pool (209.165.201.1 to 209.165.201.30). After all addresses in the IPv4_NAT_RANGE pool are allocated, dynamic PAT is performed using the IPv4_PAT address (209.165.201.31). In the event that the PAT translations are also used up, dynamic PAT is performed using the outside interface address.

```
ciscoasa(config)# object network IPv4_NAT_RANGE
ciscoasa(config-network-object)# range 209.165.201.1 209.165.201.30

ciscoasa(config-network-object)# object network IPv4_PAT
ciscoasa(config-network-object)# host 209.165.201.31

ciscoasa(config-network-object)# object-group network IPv4_GROUP
ciscoasa(config-network-object)# network-object object IPv4_NAT_RANGE
ciscoasa(config-network-object)# network-object object IPv4_PAT

ciscoasa(config-network-object)# object network my_net_obj5
ciscoasa(config-network-object)# subnet 2001:DB8::/96
ciscoasa(config-network-object)# nat (inside,outside) dynamic IPv4_GROUP interface
```

Configuring Dynamic PAT (Hide)

This section describes how to configure network object NAT for dynamic PAT (hide). For more information, see the "Dynamic PAT" section on page 3-8.

Guidelines

For a PAT pool:

- If available, the real source port number is used for the mapped port. However, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool that can be used. (8.4(3) and later, not including 8.5(1) or 8.6(1)) If you have a lot of traffic that uses the lower port ranges, you can now specify a flat range of ports to be used instead of the three unequal-sized tiers: either 1024 to 65535, or 1 to 65535.
- If you use the same PAT pool object in two separate rules, then be sure to specify the same options for each rule. For example, if one rule specifies extended PAT and a flat range, then the other rule must also specify extended PAT and a flat range.

For extended PAT for a PAT pool:

- Many application inspections do not support extended PAT. See the "Default Settings and NAT Limitations" section on page 9-4 in Chapter 9, "Getting Started with Application Layer Protocol Inspection," for a complete list of unsupported inspections.

- If you enable extended PAT for a dynamic PAT rule, then you cannot also use an address in the PAT pool as the PAT address in a separate static NAT-with-port-translation rule. For example, if the PAT pool includes 10.1.1.1, then you cannot create a static NAT-with-port-translation rule using 10.1.1.1 as the PAT address.
- If you use a PAT pool and specify an interface for fallback, you cannot specify extended PAT.
- For VoIP deployments that use ICE or TURN, do not use extended PAT. ICE and TURN rely on the PAT binding to be the same for all destinations.

For round robin for a PAT pool:

- If a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available. **Note**: This "stickiness" does not survive a failover. If the ASA fails over, then subsequent connections from a host may not use the initial IP address.
- Round robin, especially when combined with extended PAT, can consume a large amount of memory. Because NAT pools are created for every mapped protocol/IP address/port range, round robin results in a large number of concurrent NAT pools, which use memory. Extended PAT results in an even larger number of concurrent NAT pools.

Detailed Steps

	Command	Purpose
Step 1	(Optional) Create a network object or group for the mapped addresses.	See the "Adding Network Objects for Mapped Addresses" section on page 4-4.
Step 2	**object network** *obj_name* **Example:** `ciscoasa(config)# object network my-host-obj1`	Configures a network object for which you want to configure NAT, or enters object network configuration mode for an existing network object.

Step 3	{**host** *ip_address* | **subnet** *subnet_address netmask* | **range** *ip_address_1 ip_address_2*}
	Example:
	`ciscoasa(config-network-object)# range 10.1.1.1 10.1.1.90`
		If you are creating a new network object, defines the **real** IP address(es) (either IPv4 or IPv6) that you want to translate.

<table>
<tr><th></th><th>Command</th><th>Purpose</th></tr>
<tr><td>Step 4</td><td>**nat** [(*real_ifc*,*mapped_ifc*)] **dynamic** {*mapped_inline_host_ip* | *mapped_obj* | **pat-pool** *mapped_obj* [**round-robin**] [**extended**] [**flat** [**include-reserve**]] | **interface** [**ipv6**]} [**interface** [**ipv6**]] [**dns**]
Example:
ciscoasa(config-network-object)# nat (any,outside) dynamic interface</td><td>Configures **dynamic PAT** for the object IP addresses. You can only define a single NAT rule for a given object. See the "Additional Guidelines" section on page 4-3.
See the following guidelines:
• Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used; you can also specify the keyword **any** for one or both of the interfaces.
• Mapped IP address—You can specify the mapped IP address as:
– An inline host address.
– An existing network object that is defined as a host address (see Step 1).
– **pat-pool**—An existing network object or group that contains multiple addresses.
– **interface**—(Routed mode only) The IP address of the mapped interface is used as the mapped address. If you specify **ipv6**, then the IPv6 address of the interface is used. For this option, you must configure a specific interface for the *mapped_ifc*. You must use this keyword when you want to use the interface IP address; you cannot enter it inline or as an object.
• For a PAT pool, you can specify one or more of the following options:
– Round robin—The **round-robin** keyword enables round-robin address allocation for a PAT pool. Without round robin, by default all ports for a PAT address will be allocated before the next PAT address is used. The round-robin method assigns an address/port from each PAT address in the pool before returning to use the first address again, and then the second address, and so on.
(continued)</td></tr>
</table>

Command	Purpose
	(continued) – Extended PAT—The **extended** keyword enables extended PAT. Extended PAT uses 65535 ports per *service*, as opposed to per IP address, by including the destination address and port in the translation information. Normally, the destination port and address are not considered when creating PAT translations, so you are limited to 65535 ports per PAT address. For example, with extended PAT, you can create a translation of 10.1.1.1:1027 when going to 192.168.1.7:23 as well as a translation of 10.1.1.1:1027 when going to 192.168.1.7:80. – Flat range—The **flat** keyword enables use of the entire 1024 to 65535 port range when allocating ports. When choosing the mapped port number for a translation, the ASA uses the real source port number if it is available. However, without this option, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 1 to 511, 512 to 1023, and 1024 to 65535. To avoid running out of ports at the low ranges, configure this setting. To use the entire range of 1 to 65535, also specify the **include-reserve** keyword. • Interface PAT fallback—(Optional) The **interface** keyword enables interface PAT fallback when entered after a primary PAT address. After the primary PAT address(es) are used up, then the IP address of the mapped interface is used. If you specify **ipv6**, then the IPv6 address of the interface is used. For this option, you must configure a specific interface for the *mapped_ifc*. (You cannot specify **interface** in transparent mode). • DNS—(Optional) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See the "DNS and NAT" section on page 3-28 for more information.

Examples

The following example configures dynamic PAT that hides the 192.168.2.0 network behind address 10.2.2.2:

```
ciscoasa(config)# object network my-inside-net
ciscoasa(config-network-object)# subnet 192.168.2.0 255.255.255.0
ciscoasa(config-network-object)# nat (inside,outside) dynamic 10.2.2.2
```

The following example configures dynamic PAT that hides the 192.168.2.0 network behind the outside interface address:

```
ciscoasa(config)# object network my-inside-net
ciscoasa(config-network-object)# subnet 192.168.2.0 255.255.255.0
ciscoasa(config-network-object)# nat (inside,outside) dynamic interface
```

The following example configures dynamic PAT with a PAT pool to translate the inside IPv6 network to an outside IPv4 network:

```
ciscoasa(config)# object network IPv4_POOL
ciscoasa(config-network-object)# range 203.0.113.1 203.0.113.254
ciscoasa(config)# object network IPv6_INSIDE
ciscoasa(config-network-object)# subnet 2001:DB8::/96
ciscoasa(config-network-object)# nat (inside,outside) dynamic pat-pool IPv4_POOL
```

Configuring Static NAT or Static NAT-with-Port-Translation

This section describes how to configure a static NAT rule using network object NAT. For more information, see the "Static NAT" section on page 3-3.

Detailed Steps

	Command	Purpose
Step 1	(Optional) Create a network object or group for the mapped addresses.	See the "Adding Network Objects for Mapped Addresses" section on page 4-4.
Step 2	**object network** *obj_name* **Example:** `ciscoasa(config)# object network my-host-obj1`	Configures a network object for which you want to configure NAT, or enters object network configuration mode for an existing network object.

	Command	Purpose

Step 3	{**host** *ip_address* | **subnet** *subnet_address netmask* | **range** *ip_address_1 ip_address_2*}
	Example:
	`ciscoasa(config-network-object)# subnet 10.2.1.0 255.255.255.0`
		If you are creating a new network object, defines the **real** IP address(es) (IPv4 or IPv6) that you want to translate.

	Command	Purpose
Step 4	**nat** [(*real_ifc*,*mapped_ifc*)] **static** {*mapped_inline_ip* | *mapped_obj* | **interface** [**ipv6**]} [**net-to-net**] [**dns** | **service** {**tcp** | **udp**} *real_port mapped_port*] [**no-proxy-arp**] **Example:** `ciscoasa(config-network-object)# nat (inside,outside) static MAPPED_IPS service tcp 80 8080`	Configures **static NAT** for the object IP addresses. You can only define a single NAT rule for a given object. • Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used; you can also specify the keyword **any** for one or both of the interfaces. • Mapped IP Addresses—You can specify the mapped IP address as: – An inline IP address. The netmask or range for the mapped network is the same as that of the real network. For example, if the real network is a host, then this address will be a host address. In the case of a range, then the mapped addresses include the same number of addresses as the real range. For example, if the real address is defined as a range from 10.1.1.1 through 10.1.1.6, and you specify 172.20.1.1 as the mapped address, then the mapped range will include 172.20.1.1 through 172.20.1.6. – An existing network object or group (see Step 1). – **interface**—(Static NAT-with-port-translation only; routed mode) For this option, you must configure a specific interface for the *mapped_ifc*. If you specify **ipv6**, then the IPv6 address of the interface is used. Be sure to also configure the **service** keyword. Typically, you configure the same number of mapped addresses as real addresses for a one-to-one mapping. You can, however, have a mismatched number of addresses. See the "Static NAT" section on page 3-3. • Net-to-net—(Optional) For NAT 46, specify **net-to-net** to translate the first IPv4 address to the first IPv6 address, the second to the second, and so on. Without this option, the IPv4-embedded method is used. For a one-to-one translation, you must use this keyword. • DNS—(Optional) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See the "DNS and NAT" section on page 3-28. This option is not available if you specify the **service** keyword. • Port translation—(Static NAT-with-port-translation only) Specify **tcp** or **udp** and the real and mapped ports. You can enter either a port number or a well-known port name (such as **ftp**). • No Proxy ARP—(Optional) Specify **no-proxy-arp** to disable proxy ARP for incoming packets to the mapped IP addresses. See the "Mapped Addresses and Routing" section on page 3-20 for more information.

Examples

The following example configures static NAT for the real host 10.1.1.1 on the inside to 10.2.2.2 on the outside with DNS rewrite enabled.

```
ciscoasa(config)# object network my-host-obj1
ciscoasa(config-network-object)# host 10.1.1.1
ciscoasa(config-network-object)# nat (inside,outside) static 10.2.2.2 dns
```

The following example configures static NAT for the real host 10.1.1.1 on the inside to 10.2.2.2 on the outside using a mapped object.

```
ciscoasa(config)# object network my-mapped-obj
ciscoasa(config-network-object)# host 10.2.2.2

ciscoasa(config-network-object)# object network my-host-obj1
ciscoasa(config-network-object)# host 10.1.1.1
ciscoasa(config-network-object)# nat (inside,outside) static my-mapped-obj
```

The following example configures static NAT-with-port-translation for 10.1.1.1 at TCP port 21 to the outside interface at port 2121.

```
ciscoasa(config)# object network my-ftp-server
ciscoasa(config-network-object)# host 10.1.1.1
ciscoasa(config-network-object)# nat (inside,outside) static interface service tcp 21 2121
```

The following example maps an inside IPv4 network to an outside IPv6 network.

```
ciscoasa(config)# object network inside_v4_v6
ciscoasa(config-network-object)# subnet 10.1.1.0 255.255.255.0
ciscoasa(config-network-object)# nat (inside,outside) static 2001:DB8::/96
```

The following example maps an inside IPv6 network to an outside IPv6 network.

```
ciscoasa(config)# object network inside_v6
ciscoasa(config-network-object)# subnet 2001:DB8:AAAA::/96
ciscoasa(config-network-object)# nat (inside,outside) static 2001:DB8:BBBB::/96
```

Configuring Identity NAT

This section describes how to configure an identity NAT rule using network object NAT. For more information, see the "Identity NAT" section on page 3-10.

Detailed Steps

	Command	Purpose
Step 1	(Optional) Create a network object for the mapped addresses.	The object must include the same addresses that you want to translate. See the "Adding Network Objects for Mapped Addresses" section on page 4-4.
Step 2	**object network** *obj_name* **Example:** `ciscoasa(config)# object network my-host-obj1`	Configures a network object for which you want to perform identity NAT, or enters object network configuration mode for an existing network object. This network object has a different name from the mapped network object (see Step 1) even though they both contain the same IP addresses.

	Command	Purpose

Step 3	{**host** *ip_address* | **subnet** *subnet_address netmask* | **range** *ip_address_1 ip_address_2*}
	Example:
	`ciscoasa(config-network-object)# subnet 10.1.1.0 255.255.255.0`
		If you are creating a new network object, defines the real IP address(es) (IPv4 or IPv6) to which you want to perform identity NAT. If you configured a network object for the mapped addresses in Step 1, then these addresses must match.

Step 4	**nat** [**(***real_ifc***,***mapped_ifc***)**] **static** {*mapped_inline_ip* | *mapped_obj*} [**no-proxy-arp**] [**route-lookup**] **Example:** `ciscoasa(config-network-object)# nat (inside,outside) static MAPPED_IPS`	Configures **identity NAT** for the object IP addresses. **Note** You can only define a single NAT rule for a given object. See the "Additional Guidelines" section on page 4-3. See the following guidelines: • Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used; you can also specify the keyword **any** for one or both of the interfaces. • Mapped IP addresses—Be sure to configure the same IP address for both the mapped and real address. Use one of the following: – Network object—Including the same IP address as the real object (see Step 1). – Inline IP address—The netmask or range for the mapped network is the same as that of the real network. For example, if the real network is a host, then this address will be a host address. In the case of a range, then the mapped addresses include the same number of addresses as the real range. For example, if the real address is defined as a range from 10.1.1.1 through 10.1.1.6, and you specify 10.1.1.1 as the mapped address, then the mapped range will include 10.1.1.1 through 10.1.1.6. • No Proxy ARP—Specify **no-proxy-arp** to disable proxy ARP for incoming packets to the mapped IP addresses. See the "Mapped Addresses and Routing" section on page 3-20 for more information. • Route lookup—(Routed mode only; interface(s) specified) Specify **route-lookup** to determine the egress interface using a route lookup instead of using the interface specified in the NAT command. See the "Determining the Egress Interface" section on page 3-22 for more information.

Example

The following example maps a host address to itself using an inline mapped address:

```
ciscoasa(config)# object network my-host-obj1
ciscoasa(config-network-object)# host 10.1.1.1
ciscoasa(config-network-object)# nat (inside,outside) static 10.1.1.1
```

The following example maps a host address to itself using a network object:

```
ciscoasa(config)# object network my-host-obj1-identity
ciscoasa(config-network-object)# host 10.1.1.1

ciscoasa(config-network-object)# object network my-host-obj1
ciscoasa(config-network-object)# host 10.1.1.1
ciscoasa(config-network-object)# nat (inside,outside) static my-host-obj1-identity
```

Configuring Per-Session PAT Rules

By default, all TCP PAT traffic and all UDP DNS traffic uses per-session PAT. To use multi-session PAT for traffic, you can configure per-session PAT rules: a permit rule uses per-session PAT, and a deny rule uses multi-session PAT. For more information about per-session vs. multi-session PAT, see the "Per-Session PAT vs. Multi-Session PAT" section on page 3-9.

Defaults

By default, the following rules are installed:

```
xlate per-session permit tcp any4 any4
xlate per-session permit tcp any4 any6
xlate per-session permit tcp any6 any4
xlate per-session permit tcp any6 any6
xlate per-session permit udp any4 any4 eq domain
xlate per-session permit udp any4 any6 eq domain
xlate per-session permit udp any6 any4 eq domain
xlate per-session permit udp any6 any6 eq domain
```

Note You cannot remove these rules, and they always exist after any manually-created rules. Because rules are evaluated in order, you can override the default rules. For example, to completely negate these rules, you could add the following:

```
xlate per-session deny tcp any4 any4
xlate per-session deny tcp any4 any6
xlate per-session deny tcp any6 any4
xlate per-session deny tcp any6 any6
xlate per-session deny udp any4 any4 eq domain
xlate per-session deny udp any4 any6 eq domain
xlate per-session deny udp any6 any4 eq domain
xlate per-session deny udp any6 any6 eq domain
```

Detailed Steps

Command	Purpose
xlate per-session {**permit** | **deny**} {**tcp** | **udp**} *source_ip* [*operator src_port*] *destination_ip operator dest_port* **Example:** `ciscoasa(config)# xlate per-session deny tcp any4 209.165.201.3 eq 1720`	Creates a permit or deny rule. This rule is placed above the default rules, but below any other manually-created rules. Be sure to create your rules in the order you want them applied. For the source and destination IP addresses, you can configure the following: • **host** *ip_address*—Specifies an IPv4 host address. • *ip_address mask*—Specifies an IPv4 network address and subnet mask. • *ipv6-address***/***prefix-length*—Specifies an IPv6 host or network address and prefix. • **any4** and **any6—any4** specifies only IPv4 traffic; and **any6** specifies any6 traffic. The *operator* matches the port numbers used by the source or destination. The permitted operators are as follows: • lt—less than • gt—greater than • eq—equal to • neq—not equal to • range—an inclusive range of values. When you use this operator, specify two port numbers, for example: `range 100 200`

Examples

The following example creates a deny rule for H.323 traffic, so that it uses multi-session PAT:

```
ciscoasa(config)# xlate per-session deny tcp any4 209.165.201.7 eq 1720
ciscoasa(config)# xlate per-session deny udp any4 209.165.201.7 range 1718 1719
```

Monitoring Network Object NAT

To monitor object NAT, enter one of the following commands:

Command	Purpose
show nat	Shows NAT statistics, including hits for each NAT rule.
show nat pool	Shows NAT pool statistics, including the addresses and ports allocated, and how many times they were allocated.

<table>
<tr><th>Command</th><th>Purpose</th></tr>
<tr><td><code>show running-config nat</code></td><td>Shows the NAT configuration.

<b>Note</b> You cannot view the NAT configuration using the <b>show running-config object</b> command. You cannot reference objects or object groups that have not yet been created in <b>nat</b> commands. To avoid forward or circular references in <b>show</b> command output, the <b>show running-config</b> command shows the <b>object</b> command two times: first, where the IP address(es) are defined; and later, where the <b>nat</b> command is defined. This command output guarantees that objects are defined first, then object groups, and finally NAT. For example:<pre>
hostname# show running-config
...
object network obj1
    range 192.168.49.1 192.150.49.100
object network obj2
    object 192.168.49.100
object network network-1
    subnet <network-1>
object network network-2
    subnet <network-2>
object-group network pool
    network-object object obj1
    network-object object obj2
...
object network network-1
    nat (inside,outside) dynamic pool
object network network-2
    nat (inside,outside) dynamic pool
</pre></td></tr>
<tr><td><b><code>show xlate</code></b></td><td>Shows current NAT session information.</td></tr>
</table>

Configuration Examples for Network Object NAT

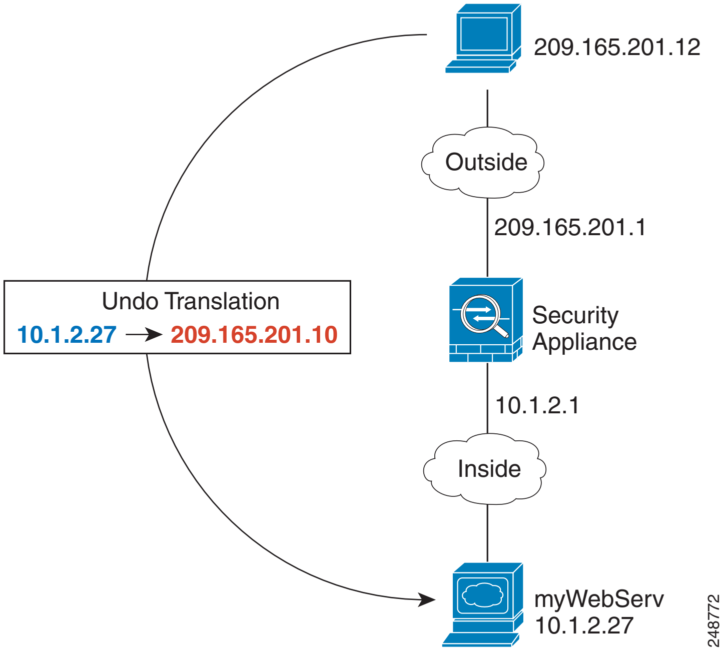
This section includes the following configuration examples:

- Providing Access to an Inside Web Server (Static NAT), page 4-19
- NAT for Inside Hosts (Dynamic NAT) and NAT for an Outside Web Server (Static NAT), page 4-19
- Inside Load Balancer with Multiple Mapped Addresses (Static NAT, One-to-Many), page 4-21
- Single Address for FTP, HTTP, and SMTP (Static NAT-with-Port-Translation), page 4-22
- DNS Server on Mapped Interface, Web Server on Real Interface (Static NAT with DNS Modification), page 4-23
- DNS Server and FTP Server on Mapped Interface, FTP Server is Translated (Static NAT with DNS Modification), page 4-25
- IPv4 DNS Server and FTP Server on Mapped Interface, IPv6 Host on Real Interface (Static NAT64 with DNS64 Modification), page 4-26

Providing Access to an Inside Web Server (Static NAT)

The following example performs static NAT for an inside web server. The real address is on a private network, so a public address is required. Static NAT is necessary so hosts can initiate traffic to the web server at a fixed address. (See Figure 4-1).

Figure 4-1 Static NAT for an Inside Web Server



Step 1 Create a network object for the internal web server:

```
ciscoasa(config)# object network myWebServ
```

Step 2 Define the web server address:

```
ciscoasa(config-network-object)# host 10.1.2.27
```

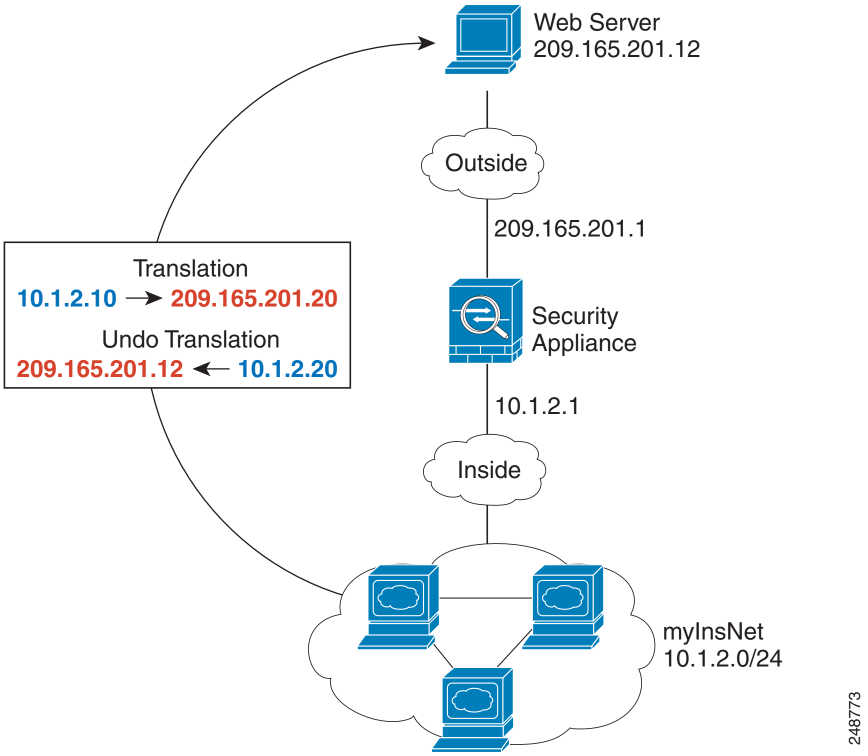
Step 3 Configure static NAT for the object:

```
ciscoasa(config-network-object)# nat (inside,outside) static 209.165.201.10
```

NAT for Inside Hosts (Dynamic NAT) and NAT for an Outside Web Server (Static NAT)

The following example configures dynamic NAT for inside users on a private network when they access the outside. Also, when inside users connect to an outside web server, that web server address is translated to an address that appears to be on the inside network. (See Figure 4-2).

Figure 4-2 Dynamic NAT for Inside, Static NAT for Outside Web Server



Step 1 Create a network object for the dynamic NAT pool to which you want to translate the inside addresses:

```
ciscoasa(config)# object network myNatPool
ciscoasa(config-network-object)# range 209.165.201.20 209.165.201.30
```

Step 2 Create a network object for the inside network:

```
ciscoasa(config)# object network myInsNet
ciscoasa(config-network-object)# subnet 10.1.2.0 255.255.255.0
```

Step 3 Enable dynamic NAT for the inside network:

```
ciscoasa(config-network-object)# nat (inside,outside) dynamic myNatPool
```

Step 4 Create a network object for the outside web server:

```
ciscoasa(config)# object network myWebServ
```

Step 5 Define the web server address:

```
ciscoasa(config-network-object)# host 209.165.201.12
```

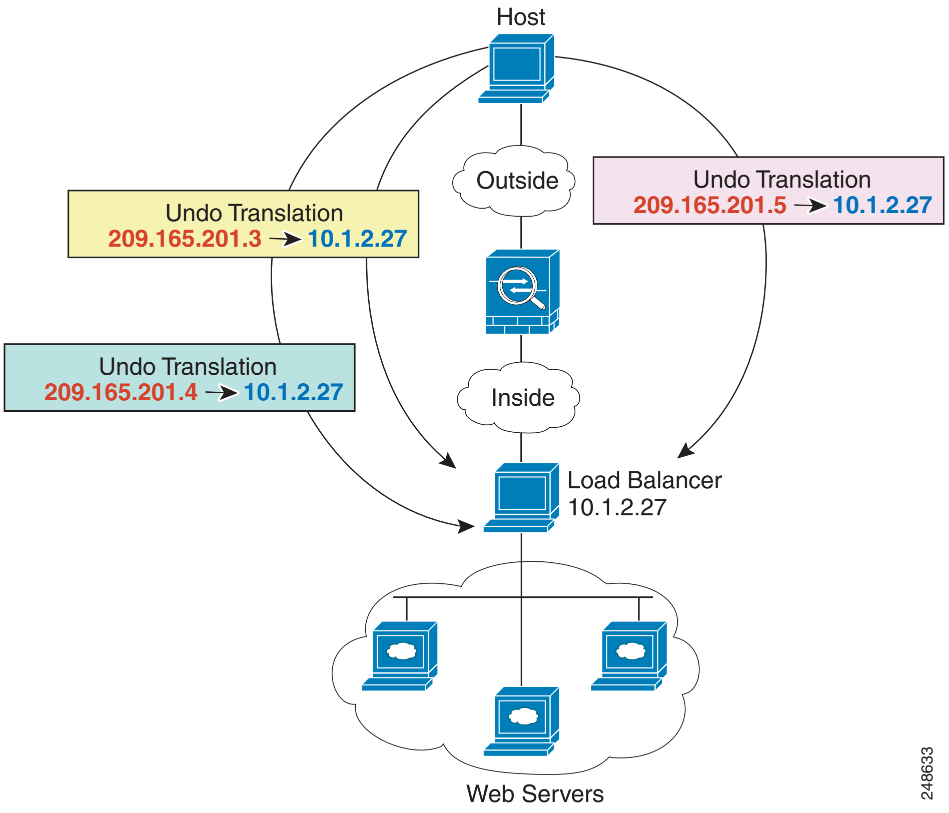
Step 6 Configure static NAT for the web server:

```
ciscoasa(config-network-object)# nat (outside,inside) static 10.1.2.20
```

Inside Load Balancer with Multiple Mapped Addresses (Static NAT, One-to-Many)

The following example shows an inside load balancer that is translated to multiple IP addresses. When an outside host accesses one of the mapped IP addresses, it is untranslated to the single load balancer address. Depending on the URL requested, it redirects traffic to the correct web server. (See Figure 4-3).

Figure 4-3 Static NAT with One-to-Many for an Inside Load Balancer



Step 1 Create a network object for the addresses to which you want to map the load balancer:

```
ciscoasa(config)# object network myPublicIPs
ciscoasa(config-network-object)# range 209.165.201.3 209.265.201.8
```

Step 2 Create a network object for the load balancer:

```
ciscoasa(config)# object network myLBHost
```

Step 3 Define the load balancer address:

```
ciscoasa(config-network-object)# host 10.1.2.27
```

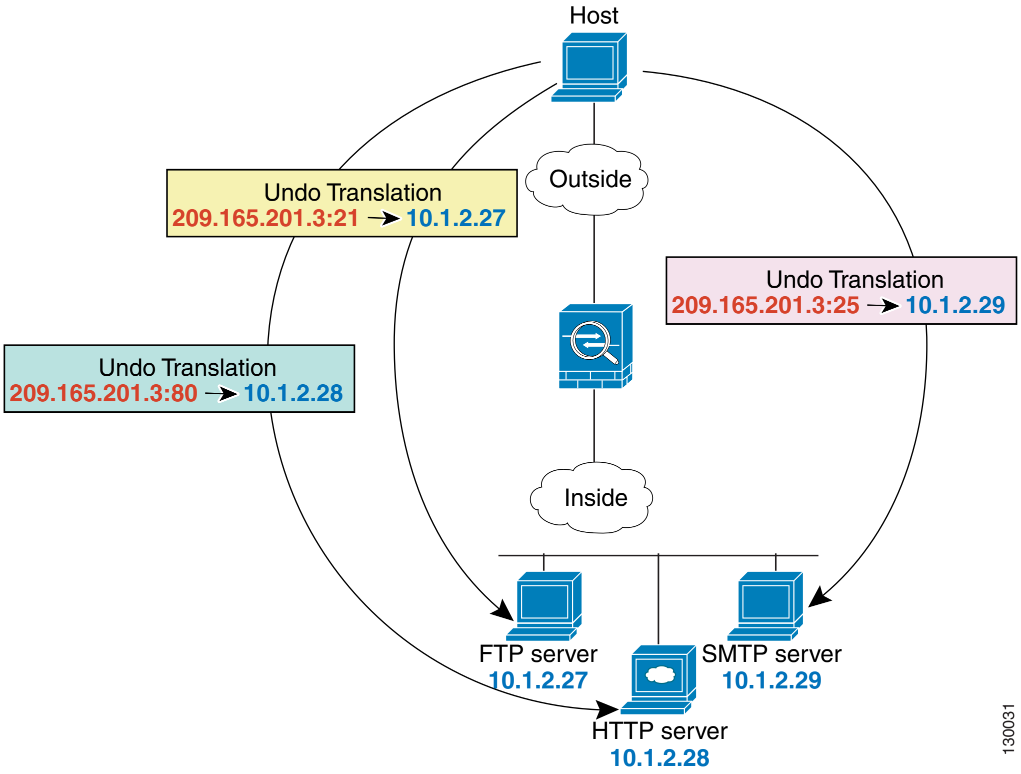
Step 4 Configure static NAT for the load balancer:

```
ciscoasa(config-network-object)# nat (inside,outside) static myPublicIPs
```

Single Address for FTP, HTTP, and SMTP (Static NAT-with-Port-Translation)

The following static NAT-with-port-translation example provides a single address for remote users to access FTP, HTTP, and SMTP. These servers are actually different devices on the real network, but for each server, you can specify static NAT-with-port-translation rules that use the same mapped IP address, but different ports. (See Figure 4-4.)

Figure 4-4 Static NAT-with-Port-Translation



Step 1 Create a network object for the FTP server address:

```
ciscoasa(config)# object network FTP_SERVER
```

Step 2 Define the FTP server address, and configure static NAT with identity port translation for the FTP server:

```
ciscoasa(config-network-object)# host 10.1.2.27
ciscoasa(config-network-object)# nat (inside,outside) static 209.165.201.3 service tcp ftp
ftp
```

Step 3 Create a network object for the HTTP server address:

```
ciscoasa(config)# object network HTTP_SERVER
```

Step 4 Define the HTTP server address, and configure static NAT with identity port translation for the HTTP server:

```
ciscoasa(config-network-object)# host 10.1.2.28
ciscoasa(config-network-object)# nat (inside,outside) static 209.165.201.3 service tcp
http http
```

Step 5 Create a network object for the SMTP server address:

```
ciscoasa(config)# object network SMTP_SERVER
```

Step 6 Define the SMTP server address, and configure static NAT with identity port translation for the SMTP server:

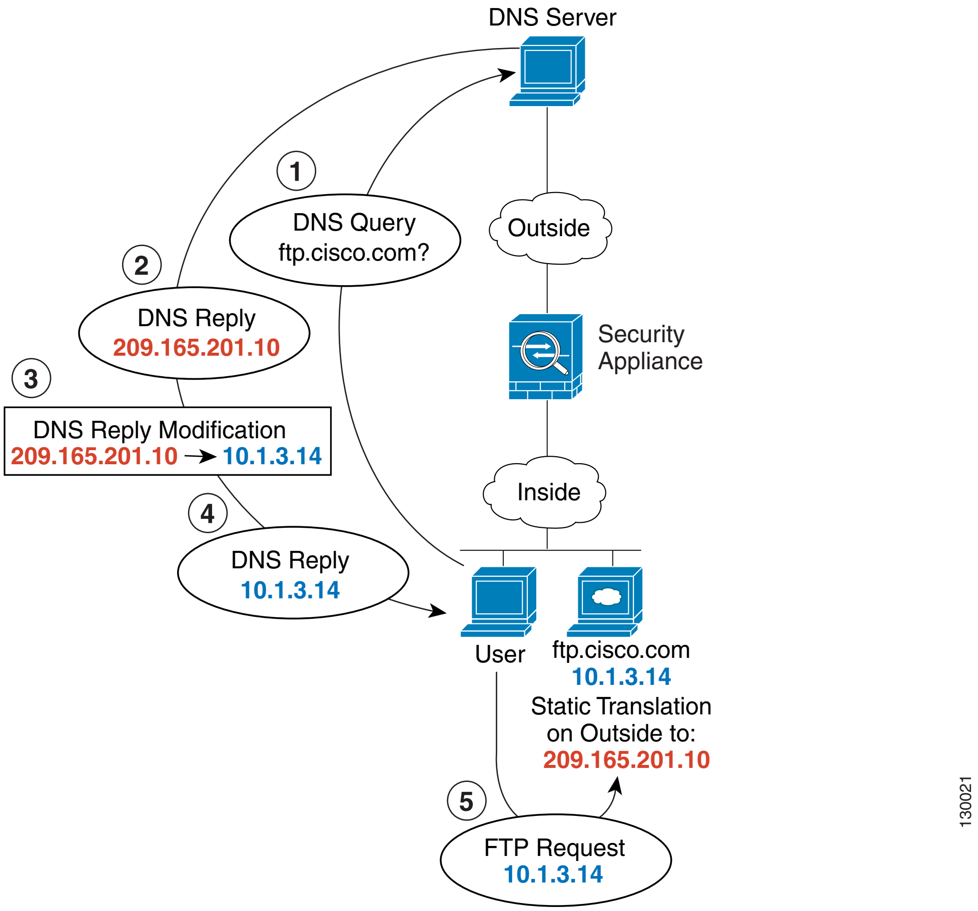
```
ciscoasa(config-network-object)# host 10.1.2.29
ciscoasa(config-network-object)# nat (inside,outside) static 209.165.201.3 service tcp
smtp smtp
```

DNS Server on Mapped Interface, Web Server on Real Interface (Static NAT with DNS Modification)

For example, a DNS server is accessible from the outside interface. A server, ftp.cisco.com, is on the inside interface. You configure the ASA to statically translate the ftp.cisco.com real address (10.1.3.14) to a mapped address (209.165.201.10) that is visible on the outside network. (See Figure 4-5.) In this case, you want to enable DNS reply modification on this static rule so that inside users who have access to ftp.cisco.com using the real address receive the real address from the DNS server, and not the mapped address.

When an inside host sends a DNS request for the address of ftp.cisco.com, the DNS server replies with the mapped address (209.165.201.10). The ASA refers to the static rule for the inside server and translates the address inside the DNS reply to 10.1.3.14. If you do not enable DNS reply modification, then the inside host attempts to send traffic to 209.165.201.10 instead of accessing ftp.cisco.com directly.

Figure 4-5 DNS Reply Modification



Step 1 Create a network object for the FTP server address:

```
ciscoasa(config)# object network FTP_SERVER
```

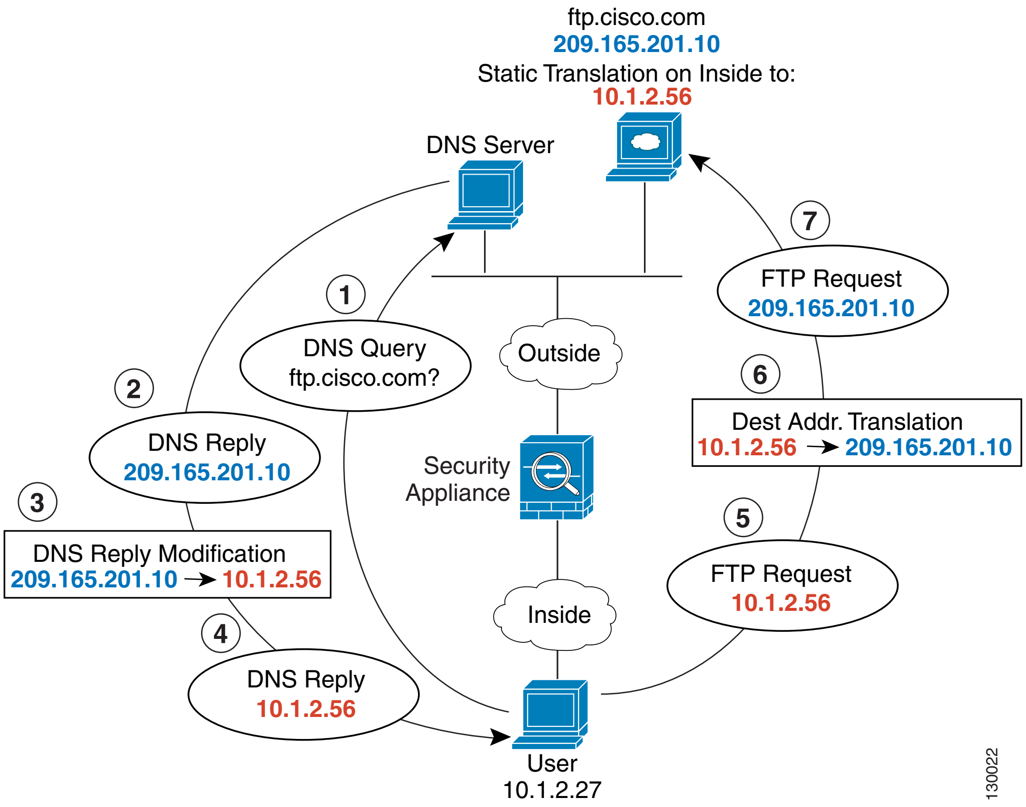
Step 2 Define the FTP server address, and configure static NAT with DNS modification:

```
ciscoasa(config-network-object)# host 10.1.3.14
ciscoasa(config-network-object)# nat (inside,outside) static 209.165.201.10 dns
```

DNS Server and FTP Server on Mapped Interface, FTP Server is Translated (Static NAT with DNS Modification)

Figure 4-6 shows an FTP server and DNS server on the outside. The ASA has a static translation for the outside server. In this case, when an inside user requests the address for ftp.cisco.com from the DNS server, the DNS server responds with the real address, 209.165.201.10. Because you want inside users to use the mapped address for ftp.cisco.com (10.1.2.56) you need to configure DNS reply modification for the static translation.

Figure 4-6 DNS Reply Modification Using Outside NAT



Step 1 Create a network object for the FTP server address:

```
ciscoasa(config)# object network FTP_SERVER
```

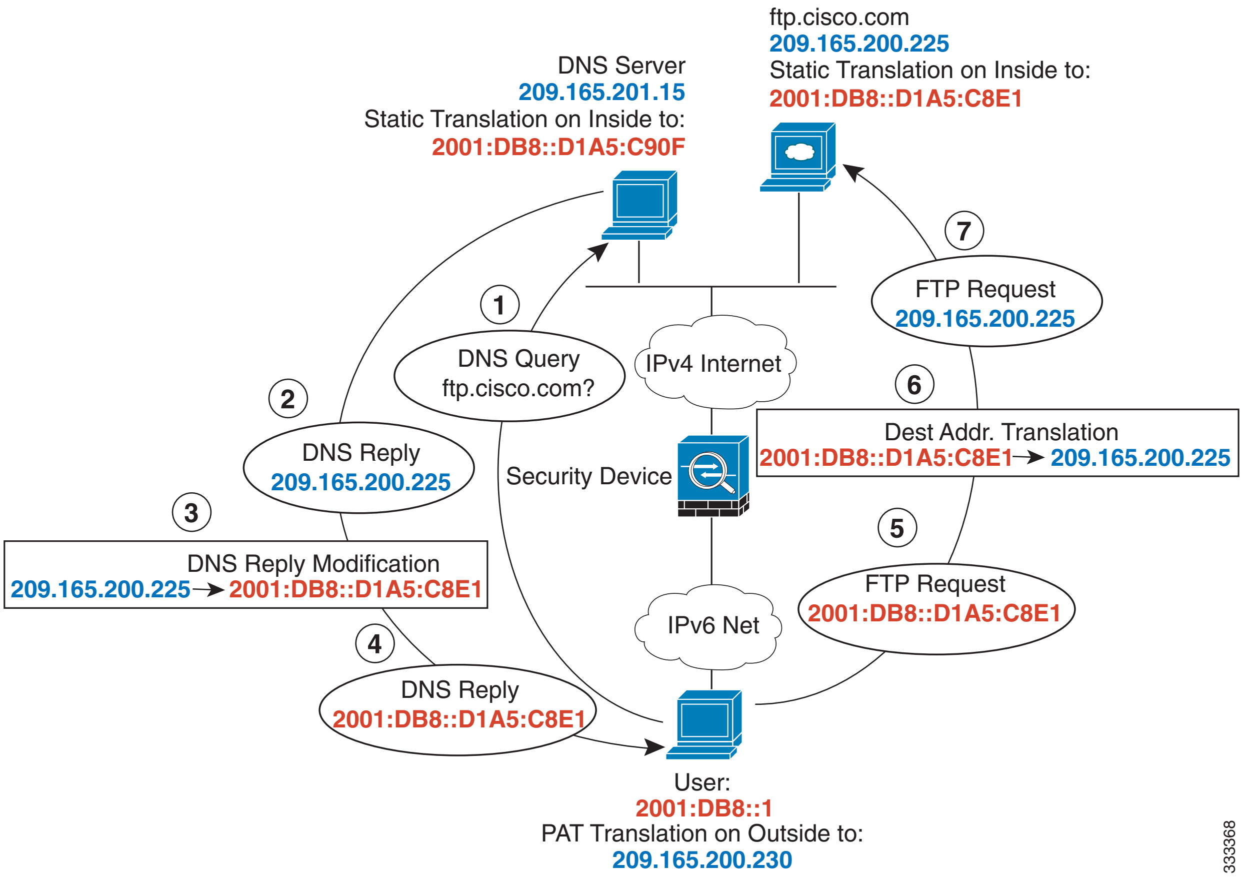
Step 2 Define the FTP server address, and configure static NAT with DNS modification:

```
ciscoasa(config-network-object)# host 209.165.201.10
ciscoasa(config-network-object)# nat (outside,inside) static 10.1.2.56 dns
```

IPv4 DNS Server and FTP Server on Mapped Interface, IPv6 Host on Real Interface (Static NAT64 with DNS64 Modification)

Figure 4-6 shows an FTP server and DNS server on the outside IPv4 network. The ASA has a static translation for the outside server. In this case, when an inside IPv6 user requests the address for ftp.cisco.com from the DNS server, the DNS server responds with the real address, 209.165.200.225. Because you want inside users to use the mapped address for ftp.cisco.com (2001:DB8::D1A5:C8E1) you need to configure DNS reply modification for the static translation. This example also includes a static NAT translation for the DNS server, and a PAT rule for the inside IPv6 hosts.

Figure 4-7 DNS Reply Modification Using Outside NAT



Step 1 Configure static NAT with DNS modification for the FTP server.

a. Create a network object for the FTP server address.

```
ciscoasa(config)# object network FTP_SERVER
```

b. Define the FTP server address, and configure static NAT with DNS modification and, because this is a one-to-one translation, configure the net-to-net method for NAT46.

```
ciscoasa(config-network-object)# host 209.165.200.225
ciscoasa(config-network-object)# nat (outside,inside) static 2001:DB8::D1A5:C8E1/128
net-to-net dns
```

Step 2 Configure NAT for the DNS server.

a. Create a network object for the DNS server address.

```
ciscoasa(config)# object network DNS_SERVER
```

b. Define the DNS server address, and configure static NAT using the net-to-net method.

```
ciscoasa(config-network-object)# host 209.165.201.15
ciscoasa(config-network-object)# nat (outside,inside) static 2001:DB8::D1A5:C90F/128
net-to-net
```

Step 3 Configure an IPv4 PAT pool for translating the inside IPv6 network.

```
ciscoasa(config)# object network IPv4_POOL
ciscoasa(config-network-object)# range 203.0.113.1 203.0.113.254
```

Step 4 Configure PAT for the inside IPv6 network.

a. Create a network object for the inside IPv6 network.

```
ciscoasa(config)# object network IPv6_INSIDE
```

b. Define the IPv6 network address, and configure dynamic NAT using a PAT pool.

```
ciscoasa(config-network-object)# subnet 2001:DB8::/96
ciscoasa(config-network-object)# nat (inside,outside) dynamic pat-pool IPv4_POOL
```

Feature History for Network Object NAT

Table 4-1 lists each feature change and the platform release in which it was implemented.

Table 4-1 Feature History for Network Object NAT

Feature Name	Platform Releases	Feature Information
Network Object NAT	8.3(1)	Configures NAT for a network object IP address(es). We introduced or modified the following commands: **nat** (object network configuration mode), **show nat**, **show xlate**, **show nat pool**.
Identity NAT configurable proxy ARP and route lookup	8.4(2)/8.5(1)	In earlier releases for identity NAT, proxy ARP was disabled, and a route lookup was always used to determine the egress interface. You could not configure these settings. In 8.4(2) and later, the default behavior for identity NAT was changed to match the behavior of other static NAT configurations: proxy ARP is enabled, and the NAT configuration determines the egress interface (if specified) by default. You can leave these settings as is, or you can enable or disable them discretely. Note that you can now also disable proxy ARP for regular static NAT. When upgrading to 8.4(2) from 8.3(1), 8.3(2), and 8.4(1), all identity NAT configurations will now include the **no-proxy-arp** and **route-lookup** keywords, to maintain existing functionality. We modified the following command: **nat static** [**no-proxy-arp**] [**route-lookup**].
PAT pool and round robin address assignment	8.4(2)/8.5(1)	You can now specify a pool of PAT addresses instead of a single address. You can also optionally enable round-robin assignment of PAT addresses instead of first using all ports on a PAT address before using the next address in the pool. These features help prevent a large number of connections from a single PAT address from appearing to be part of a DoS attack and makes configuration of large numbers of PAT addresses easy. We modifed the following command: **nat dynamic** [**pat-pool** *mapped_object* [**round-robin**]].
Round robin PAT pool allocation uses the same IP address for existing hosts	8.4(3)	When using a PAT pool with round robin allocation, if a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available. We did not modify any commands. *This feature is not available in 8.5(1) or 8.6(1).*

Table 4-1 Feature History for Network Object NAT (continued)

Feature Name	Platform Releases	Feature Information
Flat range of PAT ports for a PAT pool	8.4(3)	If available, the real source port number is used for the mapped port. However, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool. If you have a lot of traffic that uses the lower port ranges, when using a PAT pool, you can now specify a flat range of ports to be used instead of the three unequal-sized tiers: either 1024 to 65535, or 1 to 65535. We modifed the following command: **nat dynamic** [**pat-pool** *mapped_object* [**flat** [**include-reserve**]]]. *This feature is not available in 8.5(1) or 8.6(1).*
Extended PAT for a PAT pool	8.4(3)	Each PAT IP address allows up to 65535 ports. If 65535 ports do not provide enough translations, you can now enable extended PAT for a PAT pool. Extended PAT uses 65535 ports per *service*, as opposed to per IP address, by including the destination address and port in the translation information. We modifed the following command: **nat dynamic** [**pat-pool** *mapped_object* [**extended**]]. *This feature is not available in 8.5(1) or 8.6(1).*

Table 4-1 Feature History for Network Object NAT (continued)

Feature Name	Platform Releases	Feature Information
Automatic NAT rules to translate a VPN peer's local IP address back to the peer's real IP address	8.4(3)	In rare situations, you might want to use a VPN peer's real IP address on the inside network instead of an assigned local IP address. Normally with VPN, the peer is given an assigned local IP address to access the inside network. However, you might want to translate the local IP address back to the peer's real public IP address if, for example, your inside servers and network security is based on the peer's real IP address. You can enable this feature on one interface per tunnel group. Object NAT rules are dynamically added and deleted when the VPN session is established or disconnected. You can view the rules using the **show nat** command. **Note** Because of routing issues, we do not recommend using this feature unless you know you need this feature; contact Cisco TAC to confirm feature compatibility with your network. See the following limitations: • Only supports Cisco IPsec and AnyConnect Client. • Return traffic to the public IP addresses must be routed back to the ASA so the NAT policy and VPN policy can be applied. • Does not support load-balancing (because of routing issues). • Does not support roaming (public IP changing). We introduced the following command: **nat-assigned-to-public-ip** *interface* (tunnel-group general-attributes configuration mode).
NAT support for IPv6	9.0(1)	NAT now supports IPv6 traffic, as well as translating between IPv4 and IPv6. Translating between IPv4 and IPv6 is not supported in transparent mode. We modified the following commands: **nat** (object network configuration mode), **show nat**, **show nat pool**, **show xlate**.

Table 4-1 Feature History for Network Object NAT (continued)

Feature Name	Platform Releases	Feature Information
NAT support for reverse DNS lookups	9.0(1)	NAT now supports translation of the DNS PTR record for reverse DNS lookups when using IPv4 NAT, IPv6 NAT, and NAT64 with DNS inspection enabled for the NAT rule.
Per-session PAT	9.0(1)	The per-session PAT feature improves the scalability of PAT and, for clustering, allows each member unit to own PAT connections; multi-session PAT connections have to be forwarded to and owned by the master unit. At the end of a per-session PAT session, the ASA sends a reset and immediately removes the xlate. This reset causes the end node to immediately release the connection, avoiding the TIME_WAIT state. Multi-session PAT, on the other hand, uses the PAT timeout, by default 30 seconds. For "hit-and-run" traffic, such as HTTP or HTTPS, the per-session feature can dramatically increase the connection rate supported by one address. Without the per-session feature, the maximum connection rate for one address for an IP protocol is approximately 2000 per second. With the per-session feature, the connection rate for one address for an IP protocol is 65535/*average-lifetime*. By default, all TCP traffic and UDP DNS traffic use a per-session PAT xlate. For traffic that requires multi-session PAT, such as H.323, SIP, or Skinny, you can disable per-session PAT by creating a per-session deny rule. We introduced the following commands: **xlate per-session**, **show nat pool**.

Configuring Twice NAT

Twice NAT lets you identify both the source and destination address in a single rule. This chapter shows you how to configure twice NAT and includes the following sections:

Note For detailed information about how NAT works, see Chapter 3, “Information About NAT.”

Information About Twice NAT

Twice NAT lets you identify both the source and destination address in a single rule. Specifying both the source and destination addresses lets you specify that a source address should be translated to A when going to destination X, but be translated to B when going to destination Y, for example.

Note For static NAT, the rule is bidirectional, so be aware that “source” and “destination” are used in commands and descriptions throughout this guide even though a given connection might originate at the “destination” address. For example, if you configure static NAT with port address translation, and specify the source address as a Telnet server, and you want all traffic going to that Telnet server to have the port translated from 2323 to 23, then in the command, you must specify the *source* ports to be translated (real: 23, mapped: 2323). You specify the source ports because you specified the Telnet server address as the source address.

The destination address is optional. If you specify the destination address, you can either map it to itself (identity NAT), or you can map it to a different address. The destination mapping is always a static mapping.

Twice NAT also lets you use service objects for static NAT-with-port-translation; network object NAT only accepts inline definition.

For detailed information about the differences between twice NAT and network object NAT, see the "How NAT is Implemented" section on page 3-13.

Twice NAT rules are added to section 1 of the NAT rules table, or if specified, section 3. For more information about NAT ordering, see the "NAT Rule Order" section on page 3-18.

Licensing Requirements for Twice NAT

Model	License Requirement
All models	Base License.

Prerequisites for Twice NAT

- For both the real and mapped addresses, configure network objects or network object groups (the **object network** or **object-group network** command). Network object groups are particularly useful for creating a mapped address pool with discontinuous IP address ranges or multiple hosts or subnets. To create a network object or group, see the general operations configuration guide.
- For static NAT-with-port-translation, configure TCP or UDP service objects (the **object service** command). To create a service object, see the general operations configuration guide.

For specific guidelines for objects and groups, see the configuration section for the NAT type you want to configure. See also the "Guidelines and Limitations" section.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

- Supported in routed and transparent firewall mode.
- In transparent mode, you must specify the real and mapped interfaces; you cannot use **any**.
- In transparent mode, you cannot configure interface PAT, because the transparent mode interfaces do not have IP addresses. You also cannot use the management IP address as a mapped address.
- In transparent mode, translating between IPv4 and IPv6 networks is not supported. Translating between two IPv6 networks, or between two IPv4 networks is supported.

IPv6 Guidelines

- Supports IPv6.

- For routed mode, you can also translate between IPv4 and IPv6.
- For transparent mode, translating between IPv4 and IPv6 networks is not supported. Translating between two IPv6 networks, or between two IPv4 networks is supported.
- For transparent mode, a PAT pool is not supported for IPv6.
- For static NAT, you can specify an IPv6 subnet up to /64. Larger subnets are not supported.
- When using FTP with NAT46, when an IPv4 FTP client connects to an IPv6 FTP server, the client must use either the extended passive mode (EPSV) or extended port mode (EPRT); PASV and PORT commands are not supported with IPv6.

Additional Guidelines

- (This limitation is for 9.1.0 to 9.1.5; this limitation was removed in 9.1.6 and following maintenance releases.) You cannot configure FTP destination port translation when the source IP address is a subnet (or any other application that uses a secondary connection); the FTP data channel establishment does not succeed. For example, the following configuration does not work:

```
object network MyInsNet
 subnet 10.1.2.0 255.255.255.0
object network MapInsNet
 subnet 209.165.202.128 255.255.255.224
object network Server1
 host 209.165.200.225
object network Server1_mapped
 host 10.1.2.67
object service REAL_ftp
 service tcp destination eq ftp
object service MAPPED_ftp
 service tcp destination eq 2021
object network MyOutNet
 subnet 209.165.201.0 255.255.255.224

nat (inside,outside) source static MyInsNet MapInsNet destination static
Server1_mapped Server1 service MAPPED_ftp REAL_ftp
```

- If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT information is used, you can clear the translation table using the **clear xlate** command. However, clearing the translation table disconnects all current connections that use translations.

Note If you remove a dynamic NAT or PAT rule, and then add a new rule with mapped addresses that overlap the addresses in the removed rule, then the new rule will not be used until all connections associated with the removed rule time out or are cleared using the **clear xlate** command. This safeguard ensures that the same address is not assigned to multiple hosts.

- You cannot use an object group with both IPv4 and IPv6 addresses; the object group must include only one type of address.
- When using the **any** keyword in a NAT rule, the definition of "any" traffic (IPv4 vs. IPv6) depends on the rule. Before the ASA performs NAT on a packet, the packet must be IPv6-to-IPv6 or IPv4-to-IPv4; with this prerequisite, the ASA can determine the value of **any** in a NAT rule. For example, if you configure a rule from "any" to an IPv6 server, and that server was mapped from an IPv4 address, then **any** means "any IPv6 traffic." If you configure a rule from "any" to "any," and you map the source to the interface IPv4 address, then **any** means "any IPv4 traffic" because the mapped interface address implies that the destination is also IPv4.
- Objects and object groups used in NAT cannot be undefined; they must include IP addresses.

- You can use the same objects in multiple rules.
- The mapped IP address pool cannot include:
  - The mapped interface IP address. If you specify **any** interface for the rule, then all interface IP addresses are disallowed. For interface PAT (routed mode only), use the **interface** keyword instead of the IP address.
  - (Transparent mode) The management IP address.
  - (Dynamic NAT) The standby interface IP address when VPN is enabled.
  - Existing VPN pool addresses.

Default Settings

- By default, the rule is added to the end of section 1 of the NAT table.
- (Routed mode) The default real and mapped interface is Any, which applies the rule to all interfaces.
- If you specify an optional interface, then the ASA uses the NAT configuration to determine the egress interface, but you have the option to always use a route lookup instead.

Configuring Twice NAT

This section describes how to configure twice NAT. This section includes the following topics:

- Adding Network Objects for Real and Mapped Addresses, page 5-4
- (Optional) Adding Service Objects for Real and Mapped Ports, page 5-6
- Configuring Dynamic NAT, page 5-7
- Configuring Dynamic PAT (Hide), page 5-11
- Configuring Static NAT or Static NAT-with-Port-Translation, page 5-18
- Configuring Identity NAT, page 5-21
- Configuring Per-Session PAT Rules, page 5-24

Adding Network Objects for Real and Mapped Addresses

For each NAT rule, configure up to four network objects or groups for:

- **Source real address**
- **Source mapped address**
- **Destination real address**
- **Destination mapped address**

Objects are required unless you specify the **any** keyword inline to represent all traffic, or for some types of NAT, the **interface** keyword to represent the interface address. For more information about configuring a network object or group, see the general operations configuration guide.

Guidelines

- A network object group can contain objects and/or inline addresses of either IPv4 or IPv6 addresses. The group cannot contain both IPv4 and IPv6 addresses; it must contain one type only.
- See the "Guidelines and Limitations" section on page 5-2 for information about disallowed mapped IP addresses.
- Source Dynamic NAT:
  - You typically configure a larger group of real addresses to be mapped to a smaller group.
  - The mapped object or group cannot contain a subnet; the object must define a range; the group can include hosts and ranges.
  - If a mapped network object contains both ranges and host IP addresses, then the ranges are used for dynamic NAT, and the host IP addresses are used as a PAT fallback.
- Source Dynamic PAT (Hide):
  - The mapped object or group cannot contain a subnet; a network object must define a host, or for a PAT pool, a range; a network object group (for a PAT pool) can include hosts and ranges.
- Source Static NAT or Static NAT with port translation:
  - The mapped object or group can contain a host, range, or subnet.
  - The static mapping is typically one-to-one, so the real addresses have the same quantity as the mapped addresses. You can, however, have different quantities if desired. For more information, see the "Static NAT" section on page 3-3.
- Source Identity NAT
  - The real and mapped objects must match; you can use the same object for both, or you can create separate objects that contain the same IP addresses.
- Destination Static NAT or Static NAT with port translation (the destination translation is always static):
  - Although the main feature of twice NAT is the inclusion of the destination IP address, the destination address is optional. If you do specify the destination address, you can configure static translation for that address or just use identity NAT for it. You might want to configure twice NAT without a destination address to take advantage of some of the other qualities of twice NAT, including the use of network object groups for real addresses, or manually ordering of rules. For more information, see the "Main Differences Between Network Object NAT and Twice NAT" section on page 3-13.
  - For identity NAT, the real and mapped objects must match; you can use the same object for both, or you can create separate objects that contain the same IP addresses.
  - The static mapping is typically one-to-one, so the real addresses have the same quantity as the mapped addresses. You can, however, have different quantities if desired. For more information, see the "Static NAT" section on page 3-3.
  - For static interface NAT with port translation (routed mode only), you can specify the **interface** keyword instead of a network object/group for the mapped address. For more information, see the "Static Interface NAT with Port Translation" section on page 3-5.

Detailed Steps

Command	Purpose

object network *obj_name*
{**host** *ip_address* | **subnet**
subnet_address netmask | **range**
ip_address_1 ip_address_2}
 	
Example:
`ciscoasa(config)# object network MyInsNet`
`ciscoasa(config-network-object)# subnet`
`10.1.1.0 255.255.255.0`
	Adds a network object, either IPv4 or IPv6.

object-group network *grp_name*
{**network-object** {**object** *net_obj_name* |
subnet_address netmask |
host *ip_address*} |
group-object *grp_obj_name*}
 	
Example:
`ciscoasa(config)# object network TEST`
`ciscoasa(config-network-object)# range`
`10.1.1.1 10.1.1.70`
 	
`ciscoasa(config)# object network TEST2`
`ciscoasa(config-network-object)# range`
`10.1.2.1 10.1.2.70`
 	
`ciscoasa(config-network-object)#`
`object-group network MAPPED_IPS`
`ciscoasa(config-network)# network-object`
`object TEST`
`ciscoasa(config-network)# network-object`
`object TEST2`
`ciscoasa(config-network)# network-object`
`host 10.1.2.79`
	Adds a network object group, either IPv4 or IPv6.

(Optional) Adding Service Objects for Real and Mapped Ports

Configure service objects for:

- **Source real port (Static only)** *or* **Destination real port**
- **Source mapped port (Static only)** *or* **Destination mapped port**

For more information about configuring a service object, see the general operations configuration guide.

Guidelines

- NAT only supports TCP or UDP. When translating a port, be sure the protocols in the real and mapped service objects are identical (both TCP or both UDP).
- The "not equal" (**neq**) operator is not supported.
- For identity port translation, you can use the same service object for both the real and mapped ports.
- Source Dynamic NAT—Source Dynamic NAT does not support port translation.

- Source Dynamic PAT (Hide)—Source Dynamic PAT does not support port translation.
- Source Static NAT or Static NAT with port translation—A service object can contain both a source and destination port; however, you should specify *either* the source *or* the destination port for both service objects. You should only specify *both* the source and destination ports if your application uses a fixed source port (such as some DNS servers); but fixed source ports are rare. For example, if you want to translate the port for the source host, then configure the source service.
- Source Identity NAT—A service object can contain both a source and destination port; however, you should specify *either* the source *or* the destination port for both service objects. You should only specify *both* the source and destination ports if your application uses a fixed source port (such as some DNS servers); but fixed source ports are rare. For example, if you want to translate the port for the source host, then configure the source service.
- Destination Static NAT or Static NAT with port translation (the destination translation is always static)—For non-static source NAT, you can only perform port translation on the destination. A service object can contain both a source and destination port, but only the destination port is used in this case. If you specify the source port, it will be ignored.

Detailed Steps

	Command	Purpose

Step 1	**object service** *obj_name*
	service {**tcp** | **udp**} [**source** *operator port*] [**destination** *operator port*]
	
	Example:
	`ciscoasa(config)# object service REAL_SRC_SVC`
	`ciscoasa(config-service-object)# service tcp source eq 80`
	
	`ciscoasa(config)# object service MAPPED_SRC_SVC`
	`ciscoasa(config-service-object)# service tcp source eq 8080`
		Adds a service object.

Configuring Dynamic NAT

This section describes how to configure twice NAT for dynamic NAT. For more information, see the "Dynamic NAT" section on page 3-7.

Detailed Steps

	Command	Purpose
Step 1	Create network objects or groups for the: • Source real addresses • Source mapped addresses • Destination real addresses • Destination mapped addresses	See the "Adding Network Objects for Real and Mapped Addresses" section on page 5-4. If you want to translate all source traffic, you can skip adding an object for the source real addresses, and instead specify the **any** keyword in the **nat** command. If you want to configure destination static interface NAT with port translation only, you can skip adding an object for the destination mapped addresses, and instead specify the **interface** keyword in the **nat** command.
Step 2	(Optional) Create service objects for the: • Destination real ports • Destination mapped ports	See the "(Optional) Adding Service Objects for Real and Mapped Ports" section on page 5-6.

	Command	Purpose

Step 3	**nat** [(*real_ifc*,*mapped_ifc*)]
	[*line* | {**after-auto** [*line*]}]
	source dynamic {*real_obj* | **any**}
	{*mapped_obj* [**interface** [**ipv6**]]}
	[**destination static** {*mapped_obj* |
	interface [**ipv6**]} *real_obj*]
	[**service** *mapped_dest_svc_obj*
	real_dest_svc_obj] [**dns**] [**unidirectional**]
	[**inactive**] [**description** *desc*]
	
	Example:
	`ciscoasa(config)# nat (inside,outside)`
	`source dynamic MyInsNet NAT_POOL`
	`destination static Server1_mapped Server1`
	`service MAPPED_SVC REAL_SVC`
		Configure **dynamic NAT**. See the following guidelines:
		• Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used; you can also specify the keyword **any** for one or both of the interfaces.
		• Section and Line—(Optional) By default, the NAT rule is added to the end of section 1 of the NAT table (see the "NAT Rule Order" section on page 3-18). If you want to add the rule into section 3 instead (after the network object NAT rules), then use the **after-auto** keyword. You can insert a rule anywhere in the applicable section using the *line* argument.
		• Source addresses:
		– Real—Specify a network object, group, or the **any** keyword.
		– Mapped—Specify a different network object or group. You can optionally configure the following fallback method:
		Interface PAT fallback—(Routed mode only) The **interface** keyword enables interface PAT fallback. If you specify **ipv6**, then the IPv6 address of the interface is used. After the mapped IP addresses are used up, then the IP address of the mapped interface is used. For this option, you must configure a specific interface for the *mapped_ifc*.

Command	Purpose
	(Continued) • Destination addresses (Optional): – Mapped—Specify a network object or group, or for static interface NAT with port translation only, specify the **interface** keyword. If you specify **ipv6**, then the IPv6 address of the interface is used. If you specify **interface**, be sure to also configure the **service** keyword. For this option, you must configure a specific interface for the *real_ifc*. See the "Static Interface NAT with Port Translation" section on page 3-5 for more information. – Real—Specify a network object or group. For identity NAT, simply use the same object or group for both the real and mapped addresses. • Destination port—(Optional) Specify the **service** keyword along with the mapped and real service objects. For identity port translation, simply use the same service object for both the real and mapped ports. • DNS—(Optional; for a source-only rule) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). You cannot configure the **dns** keyword if you configure a **destination** address. See the "DNS and NAT" section on page 3-28 for more information. • Unidirectional—(Optional) Specify **unidirectional** so the destination addresses cannot initiate traffic to the source addresses. • Inactive—(Optional) To make this rule inactive without having to remove the command, use the **inactive** keyword. To reactivate it, reenter the whole command without the **inactive** keyword. • Description—Optional) Provide a description up to 200 characters using the **description** keyword.

Examples

The following example configures dynamic NAT for inside network 10.1.1.0/24 when accessing servers on the 209.165.201.1/27 network as well as servers on the 203.0.113.0/24 network:

```
ciscoasa(config)# object network INSIDE_NW
ciscoasa(config-network-object)# subnet 10.1.1.0 255.255.255.0

ciscoasa(config)# object network MAPPED_1
ciscoasa(config-network-object)# range 209.165.200.225 209.165.200.254

ciscoasa(config)# object network MAPPED_2
ciscoasa(config-network-object)# range 209.165.202.129 209.165.200.158

ciscoasa(config)# object network SERVERS_1
ciscoasa(config-network-object)# subnet 209.165.201.0 255.255.255.224

ciscoasa(config)# object network SERVERS_2
ciscoasa(config-network-object)# subnet 203.0.113.0 255.255.255.0

ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW MAPPED_1 destination
static SERVERS_1 SERVERS_1
ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW MAPPED_2 destination
static SERVERS_2 SERVERS_2
```

The following example configures dynamic NAT for an IPv6 inside network 2001:DB8:AAAA::/96 when accessing servers on the IPv4 209.165.201.1/27 network as well as servers on the 203.0.113.0/24 network:

```
ciscoasa(config)# object network INSIDE_NW
ciscoasa(config-network-object)# subnet 2001:DB8:AAAA::/96

ciscoasa(config)# object network MAPPED_1
ciscoasa(config-network-object)# range 209.165.200.225 209.165.200.254

ciscoasa(config)# object network MAPPED_2
ciscoasa(config-network-object)# range 209.165.202.129 209.165.200.158

ciscoasa(config)# object network SERVERS_1
ciscoasa(config-network-object)# subnet 209.165.201.0 255.255.255.224

ciscoasa(config)# object network SERVERS_2
ciscoasa(config-network-object)# subnet 203.0.113.0 255.255.255.0

ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW MAPPED_1 destination
static SERVERS_1 SERVERS_1
ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW MAPPED_2 destination
static SERVERS_2 SERVERS_2
```

Configuring Dynamic PAT (Hide)

This section describes how to configure twice NAT for dynamic PAT (hide). For more information, see the "Dynamic PAT" section on page 3-8.

Guidelines

For a PAT pool:

- If available, the real source port number is used for the mapped port. However, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool that can be used. (8.4(3) and later, not including 8.5(1) or 8.6(1)) If you have a lot of traffic that uses the lower port ranges, you can now specify a flat range of ports to be used instead of the three unequal-sized tiers: either 1024 to 65535, or 1 to 65535.
- If you use the same PAT pool object in two separate rules, then be sure to specify the same options for each rule. For example, if one rule specifies extended PAT and a flat range, then the other rule must also specify extended PAT and a flat range.

For extended PAT for a PAT pool:

- Many application inspections do not support extended PAT. See the "Default Settings and NAT Limitations" section on page 9-4 in Chapter 9, "Getting Started with Application Layer Protocol Inspection," for a complete list of unsupported inspections.
- If you enable extended PAT for a dynamic PAT rule, then you cannot also use an address in the PAT pool as the PAT address in a separate static NAT-with-port-translation rule. For example, if the PAT pool includes 10.1.1.1, then you cannot create a static NAT-with-port-translation rule using 10.1.1.1 as the PAT address.
- If you use a PAT pool and specify an interface for fallback, you cannot specify extended PAT.
- For VoIP deployments that use ICE or TURN, do not use extended PAT. ICE and TURN rely on the PAT binding to be the same for all destinations.

For round robin for a PAT pool:

- If a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available. **Note**: This "stickiness" does not survive a failover. If the ASA fails over, then subsequent connections from a host may not use the initial IP address.
- Round robin, especially when combined with extended PAT, can consume a large amount of memory. Because NAT pools are created for every mapped protocol/IP address/port range, round robin results in a large number of concurrent NAT pools, which use memory. Extended PAT results in an even larger number of concurrent NAT pools.

Detailed Steps

	Command	Purpose
Step 1	Create network objects or groups for the: • Source real addresses • Source mapped addresses • Destination real addresses • Destination mapped addresses	See the "Adding Network Objects for Real and Mapped Addresses" section on page 5-4. If you want to translate all source traffic, you can skip adding an object for the source real addresses, and instead specify the **any** keyword in the **nat** command. If you want to use the interface address as the mapped address, you can skip adding an object for the source mapped addresses, and instead specify the **interface** keyword in the **nat** command. If you want to configure destination static interface NAT with port translation only, you can skip adding an object for the destination mapped addresses, and instead specify the **interface** keyword in the **nat** command.
Step 2	(Optional) Create service objects for the: • Destination real ports • Destination mapped ports	See the "(Optional) Adding Service Objects for Real and Mapped Ports" section on page 5-6.

	Command	Purpose
Step 3	**nat** [(*real_ifc*,*mapped_ifc*)] [*line* | {**after-auto** [*line*]}] **source dynamic** {*real-obj* | **any**} {*mapped_obj* [**interface** [**ipv6**]] | [**pat-pool** *mapped_obj* [**round-robin**] [**extended**] [**flat** [**include-reserve**]] [**interface** [**ipv6**]] | **interface** [**ipv6**]} [**destination static** {*mapped_obj* | **interface** [**ipv6**]} *real_obj*] [**service** *mapped_dest_svc_obj* *real_dest_svc_obj*] [**dns**] [**unidirectional**] [**inactive**] [**description** *desc*] **Example:** `ciscoasa(config)# nat (inside,outside) source dynamic MyInsNet interface destination static Server1 Server1 description Interface PAT for inside addresses when going to server 1`	Configures **dynamic PAT (hide)**. See the following guidelines: • Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used; you can also specify the keyword **any** for one or both of the interfaces. • Section and Line—(Optional) By default, the NAT rule is added to the end of section 1 of the NAT table (see the "NAT Rule Order" section on page 3-18). If you want to add the rule into section 3 instead (after the network object NAT rules), then use the **after-auto** keyword. You can insert a rule anywhere in the applicable section using the *line* argument. • Source addresses: – Real—Specify a network object, group, or the **any** keyword. Use the **any** keyword if you want to translate all traffic from the real interface to the mapped interface. – Mapped—Configure one of the following: - Network object—Specify a network object that contains a host address. - **pat-pool**—Specify the **pat-pool** keyword and a network object or group that contains multiple addresses. - **interface**—(Routed mode only) Specify the **interface** keyword alone to only use interface PAT. If you specify **ipv6**, then the IPv6 address of the interface is used. When specified with a PAT pool or network object, the **interface** keyword enables interface PAT fallback. After the PAT IP addresses are used up, then the IP address of the mapped interface is used. For this option, you must configure a specific interface for the *mapped_ifc*. (continued)

Command	Purpose
	(continued) For a PAT pool, you can specify one or more of the following options: -- Round robin—The **round-robin** keyword enables round-robin address allocation for a PAT pool. Without round robin, by default all ports for a PAT address will be allocated before the next PAT address is used. The round-robin method assigns an address/port from each PAT address in the pool before returning to use the first address again, and then the second address, and so on. -- Extended PAT—The **extended** keyword enables extended PAT. Extended PAT uses 65535 ports per *service*, as opposed to per IP address, by including the destination address and port in the translation information. Normally, the destination port and address are not considered when creating PAT translations, so you are limited to 65535 ports per PAT address. For example, with extended PAT, you can create a translation of 10.1.1.1:1027 when going to 192.168.1.7:23 as well as a translation of 10.1.1.1:1027 when going to 192.168.1.7:80. -- Flat range—The **flat** keyword enables use of the entire 1024 to 65535 port range when allocating ports. When choosing the mapped port number for a translation, the ASA uses the real source port number if it is available. However, without this option, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 1 to 511, 512 to 1023, and 1024 to 65535. To avoid running out of ports at the low ranges, configure this setting. To use the entire range of 1 to 65535, also specify the **include-reserve** keyword. (continued)

Command	Purpose
	(continued) • Destination addresses (Optional): – Mapped—Specify a network object or group, or for static interface NAT with port translation only (routed mode), specify the **interface** keyword. If you specify **ipv6**, then the IPv6 address of the interface is used. If you specify **interface**, be sure to also configure the **service** keyword. For this option, you must configure a specific interface for the *real_ifc*. See the "Static Interface NAT with Port Translation" section on page 3-5 for more information. – Real—Specify a network object or group. For identity NAT, simply use the same object or group for both the real and mapped addresses. • Destination port—(Optional) Specify the **service** keyword along with the real and mapped service objects. For identity port translation, simply use the same service object for both the real and mapped ports. • DNS—(Optional; for a source-only rule) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). You cannot configure the **dns** keyword if you configure a **destination** address. See the "DNS and NAT" section on page 3-28 for more information. • Unidirectional—(Optional) Specify **unidirectional** so the destination addresses cannot initiate traffic to the source addresses. • Inactive—(Optional) To make this rule inactive without having to remove the command, use the **inactive** keyword. To reactivate it, reenter the whole command without the **inactive** keyword. • Description—(Optional) Provide a description up to 200 characters using the **description** keyword.

Examples

The following example configures interface PAT for inside network 192.168.1.0/24 when accessing outside Telnet server 209.165.201.23, and Dynamic PAT using a PAT pool when accessing any server on the 203.0.113.0/24 network.

```
ciscoasa(config)# object network INSIDE_NW
ciscoasa(config-network-object)# subnet 192.168.1.0 255.255.255.0

ciscoasa(config)# object network PAT_POOL
ciscoasa(config-network-object)# range 209.165.200.225 209.165.200.254

ciscoasa(config)# object network TELNET_SVR
ciscoasa(config-network-object)# host 209.165.201.23

ciscoasa(config)# object service TELNET
ciscoasa(config-service-object)# service tcp destination eq 23

ciscoasa(config)# object network SERVERS
ciscoasa(config-network-object)# subnet 203.0.113.0 255.255.255.0

ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW interface destination
static TELNET_SVR TELNET_SVR service TELNET TELNET
ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW pat-pool PAT_POOL
destination static SERVERS SERVERS
```

The following example configures interface PAT for inside network 192.168.1.0/24 when accessing outside IPv6 Telnet server 2001:DB8::23, and Dynamic PAT using a PAT pool when accessing any server on the 2001:DB8:AAAA::/96 network.

```
ciscoasa(config)# object network INSIDE_NW
ciscoasa(config-network-object)# subnet 192.168.1.0 255.255.255.0

ciscoasa(config)# object network PAT_POOL
ciscoasa(config-network-object)# range 2001:DB8:AAAA::1 2001:DB8:AAAA::200

ciscoasa(config)# object network TELNET_SVR
ciscoasa(config-network-object)# host 2001:DB8::23

ciscoasa(config)# object service TELNET
ciscoasa(config-service-object)# service tcp destination eq 23

ciscoasa(config)# object network SERVERS
ciscoasa(config-network-object)# subnet 2001:DB8:AAAA::/96

ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW interface ipv6 destination
static TELNET_SVR TELNET_SVR service TELNET TELNET
ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW pat-pool PAT_POOL
destination static SERVERS SERVERS
```

Configuring Static NAT or Static NAT-with-Port-Translation

This section describes how to configure a static NAT rule using twice NAT. For more information about static NAT, see the "Static NAT" section on page 3-3.

Detailed Steps

	Command	Purpose
Step 1	Create network objects or groups for the: • Source real addresses • Source mapped addresses • Destination real addresses • Destination mapped addresses	See the "Adding Network Objects for Real and Mapped Addresses" section on page 5-4. If you want to configure source static interface NAT with port translation only, you can skip adding an object for the source mapped addresses, and instead specify the **interface** keyword in the **nat** command. If you want to configure destination static interface NAT with port translation only, you can skip adding an object for the destination mapped addresses, and instead specify the **interface** keyword in the **nat** command.
Step 2	(Optional) Create service objects for the: • Source *or* Destination real ports • Source *or* Destination mapped ports	See the "(Optional) Adding Service Objects for Real and Mapped Ports" section on page 5-6.

	Command	Purpose
Step 3	**nat** [(*real_ifc*,*mapped_ifc*)] [*line* | {**after-object** [*line*]}] **source static** *real_ob* [*mapped_obj* | **interface** [**ipv6**]] [**destination static** {*mapped_obj* | **interface** [**ipv6**]} *real_obj*] [**service** *real_src_mapped_dest_svc_obj* *mapped_src_real_dest_svc_obj*][**net-to-net**] [**dns**] [**unidirectional** | **no-proxy-arp**] [**inactive**] [**description** *desc*] **Example:** ciscoasa(config)# nat (inside,dmz) source static MyInsNet MyInsNet_mapped destination static Server1 Server1 service REAL_SRC_SVC MAPPED_SRC_SVC	Configures **static NAT**. See the following guidelines: • Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used; you can also specify the keyword **any** for one or both of the interfaces. • Section and Line—(Optional) By default, the NAT rule is added to the end of section 1 of the NAT table. See the "NAT Rule Order" section on page 3-18 for more information about sections. If you want to add the rule into section 3 instead (after the network object NAT rules), then use the **after-auto** keyword. You can insert a rule anywhere in the applicable section using the *line* argument. • Source addresses: – Real—Specify a network object or group. – Mapped—Specify a different network object or group. For static interface NAT with port translation only, you can specify the **interface** keyword (routed mode only). If you specify **ipv6**, then the IPv6 address of the interface is used. If you specify **interface**, be sure to also configure the **service** keyword (in this case, the service objects should include only the source port). For this option, you must configure a specific interface for the *mapped_ifc*. See the "Static Interface NAT with Port Translation" section on page 3-5 for more information. • Destination addresses (Optional): – Mapped—Specify a network object or group, or for static interface NAT with port translation only, specify the **interface** keyword. If you specify **ipv6**, then the IPv6 address of the interface is used. If you specify **interface**, be sure to also configure the **service** keyword (in this case, the service objects should include only the destination port). For this option, you must configure a specific interface for the *real_ifc*. – Real—Specify a network object or group. For identity NAT, simply use the same object or group for both the real and mapped addresses.

Command	Purpose
	(Continued) • Ports—(Optional) Specify the **service** keyword along with the real and mapped service objects. For source port translation, the objects must specify the source service. The order of the service objects in the command for source port translation is **service** *real_obj mapped_obj*. For destination port translation, the objects must specify the destination service. The order of the service objects for destination port translation is **service** *mapped_obj real_obj*. In the rare case where you specify both the source and destination ports in the object, the first service object contains the real source port/mapped destination port; the second service object contains the mapped source port/real destination port. For identity port translation, simply use the same service object for both the real and mapped ports (source and/or destination ports, depending on your configuration). • Net-to-net—(Optional) For NAT 46, specify **net-to-net** to translate the first IPv4 address to the first IPv6 address, the second to the second, and so on. Without this option, the IPv4-embedded method is used. For a one-to-one translation, you must use this keyword. • DNS—(Optional; for a source-only rule) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). You cannot configure the **dns** keyword if you configure a **destination** address. See the "DNS and NAT" section on page 3-28 for more information. • Unidirectional—(Optional) Specify **unidirectional** so the destination addresses cannot initiate traffic to the source addresses. • No Proxy ARP—(Optional) Specify **no-proxy-arp** to disable proxy ARP for incoming packets to the mapped IP addresses. See the "Mapped Addresses and Routing" section on page 3-20 for more information. • Inactive—(Optional) To make this rule inactive without having to remove the command, use the **inactive** keyword. To reactivate it, reenter the whole command without the **inactive** keyword. • Description—(Optional) Provide a description up to 200 characters using the **description** keyword.

Examples

The following example shows the use of static interface NAT with port translation. Hosts on the outside access an FTP server on the inside by connecting to the outside interface IP address with destination port 65000 through 65004. The traffic is untranslated to the internal FTP server at 192.168.10.100:6500 through :65004. Note that you specify the source port range in the service object (and not the destination port) because you want to translate the source address and port as identified in the command; the destination port is "any." Because static NAT is bidirectional, "source" and "destination" refers primarily to the command keywords; the actual source and destination address and port in a packet depends on which host sent the packet. In this example, connections are originated from outside to inside, so the "source" address and port of the FTP server is actually the destination address and port in the originating packet.

```
ciscoasa(config)# object service FTP_PASV_PORT_RANGE
ciscoasa(config-service-object)# service tcp source range 65000 65004

ciscoasa(config)# object network HOST_FTP_SERVER
ciscoasa(config-network-object)# host 192.168.10.100

ciscoasa(config)# nat (inside,outside) source static HOST_FTP_SERVER interface service
FTP_PASV_PORT_RANGE FTP_PASV_PORT_RANGE
```

The following example shows a static translation of one IPv6 network to another IPv6 when accessing an IPv6 network, and the dynamic PAT translation to an IPv4 PAT pool when accessing the IPv4 network:

```
ciscoasa(config)# object network INSIDE_NW
ciscoasa(config-network-object)# subnet 2001:DB8:AAAA::/96

ciscoasa(config)# object network MAPPED_IPv6_NW
ciscoasa(config-network-object)# subnet 2001:DB8:BBBB::/96

ciscoasa(config)# object network OUTSIDE_IPv6_NW
ciscoasa(config-network-object)# subnet 2001:DB8:CCCC::/96

ciscoasa(config)# object network OUTSIDE_IPv4_NW
ciscoasa(config-network-object)# subnet 10.1.1.0 255.255.255.0

ciscoasa(config)# object network MAPPED_IPv4_POOL
ciscoasa(config-network-object)# range 10.1.2.1 10.1.2.254

ciscoasa(config)# nat (inside,outside) source static INSIDE_NW MAPPED_IPv6_NW destination
static OUTSIDE_IPv6_NW OUTSIDE_IPv6_NW
ciscoasa(config)# nat (inside,outside) source dynamic INSIDE_NW pat-pool MAPPED_IPv4_POOL
destination static OUTSIDE_IPv4_NW OUTSIDE_IPv4_NW
```

Configuring Identity NAT

This section describes how to configure an identity NAT rule using twice NAT. For more information about identity NAT, see the "Identity NAT" section on page 3-10.

Detailed Steps

	Command	Purpose
Step 1	Create network objects or groups for the: • Source real addresses (you will typically use the same object for the source mapped addresses) • Destination real addresses • Destination mapped addresses	See the "Adding Network Objects for Real and Mapped Addresses" section on page 5-4. If you want to perform identity NAT for all addresses, you can skip creating an object for the the source real addresses and instead use the keywords **any any** in the **nat** command. If you want to configure destination static interface NAT with port translation only, you can skip adding an object for the destination mapped addresses, and instead specify the **interface** keyword in the **nat** command.
Step 2	(Optional) Create service objects for the: • Source *or* Destination real ports • Source *or* Destination mapped ports	See the "(Optional) Adding Service Objects for Real and Mapped Ports" section on page 5-6.

	Command	Purpose
Step 3	**nat** [(*real_ifc*,*mapped_ifc*)] [*line* | {**after-object** [*line*]}] **source static** {*nw_obj nw_obj* | **any any**} [**destination static** {*mapped_obj* | **interface** [**ipv6**]} *real_obj*] [**service** *real_src_mapped_dest_svc_obj* *mapped_src_real_dest_svc_obj*] [**no-proxy-arp**] [**route-lookup**] [**inactive**] [**description** *desc*] **Example:** `ciscoasa(config)# nat (inside,outside) source static MyInsNet MyInsNet destination static Server1 Server1`	Configures **identity NAT**. See the following guidelines: • Interfaces—(Required for transparent mode) Specify the real and mapped interfaces. Be sure to include the parentheses in your command. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used; you can also specify the keyword **any** for one or both of the interfaces. • Section and Line—(Optional) By default, the NAT rule is added to the end of section 1 of the NAT table. See the "NAT Rule Order" section on page 3-18 for more information about sections. If you want to add the rule into section 3 instead (after the network object NAT rules), then use the **after-auto** keyword. You can insert a rule anywhere in the applicable section using the *line* argument. • Source addresses—Specify a network object, group, or the **any** keyword for both the real and mapped addresses. • Destination addresses (Optional): – Mapped—Specify a network object or group, or for static interface NAT with port translation only, specify the **interface** keyword (routed mode only).If you specify **ipv6**, then the IPv6 address of the interface is used. If you specify **interface**, be sure to also configure the **service** keyword (in this case, the service objects should include only the destination port). For this option, you must configure a specific interface for the *real_ifc*. See the "Static Interface NAT with Port Translation" section on page 3-5 for more information. – Real—Specify a network object or group. For identity NAT, simply use the same object or group for both the real and mapped addresses. • Port—(Optional) Specify the **service** keyword along with the real and mapped service objects. For source port translation, the objects must specify the source service. The order of the service objects in the command for source port translation is **service** *real_obj mapped_obj*. For destination port translation, the objects must specify the destination service. The order of the service objects for destination port translation is **service** *mapped_obj real_obj*. In the rare case where you specify both the source and destination ports in the object, the first service object contains the real source port/mapped destination port; the second service object contains the mapped source port/real destination port. For identity port translation, simply use the same service object for both the real and mapped ports (source and/or destination ports, depending on your configuration).

Command	Purpose
	(Continued) - No Proxy ARP—(Optional) Specify **no-proxy-arp** to disable proxy ARP for incoming packets to the mapped IP addresses. See the "Mapped Addresses and Routing" section on page 3-20 for more information. - Route lookup—(Optional; routed mode only; interface(s) specified) Specify **route-lookup** to determine the egress interface using a route lookup instead of using the interface specified in the NAT command. See the "Determining the Egress Interface" section on page 3-22 for more information. - Inactive—(Optional) To make this rule inactive without having to remove the command, use the **inactive** keyword. To reactivate it, reenter the whole command without the **inactive** keyword. - Description—(Optional) Provide a description up to 200 characters using the **description** keyword.

Configuring Per-Session PAT Rules

By default, all TCP PAT traffic and all UDP DNS traffic uses per-session PAT. To use multi-session PAT for traffic, you can configure per-session PAT rules: a permit rule uses per-session PAT, and a deny rule uses multi-session PAT. For more information about per-session vs. multi-session PAT, see the "Per-Session PAT vs. Multi-Session PAT" section on page 3-9.

Detailed Steps

To configure a per-session PAT rule, see the "Configuring Per-Session PAT Rules" section on page 4-16.

Monitoring Twice NAT

To monitor twice NAT, enter one of the following commands:

Command	Purpose
`show nat`	Shows NAT statistics, including hits for each NAT rule.
`show nat pool`	Shows NAT pool statistics, including the addresses and ports allocated, and how many times they were allocated.
`show xlate`	Shows current NAT session information.
`show nat divert-table`	All NAT rules build an entry in the NAT divert table. If the NAT divert field is set to ignore=yes NAT on the matching rule, the ASA stops the lookup and does a route lookup based on the destination IP to determine the egress interface. If the NAT divert field is set to ignore=no on the matching rule, walk the NAT table based on the found input_ifc and output_ifc and do the necessary translation. Egress interface will be output_ifc.

Configuration Examples for Twice NAT

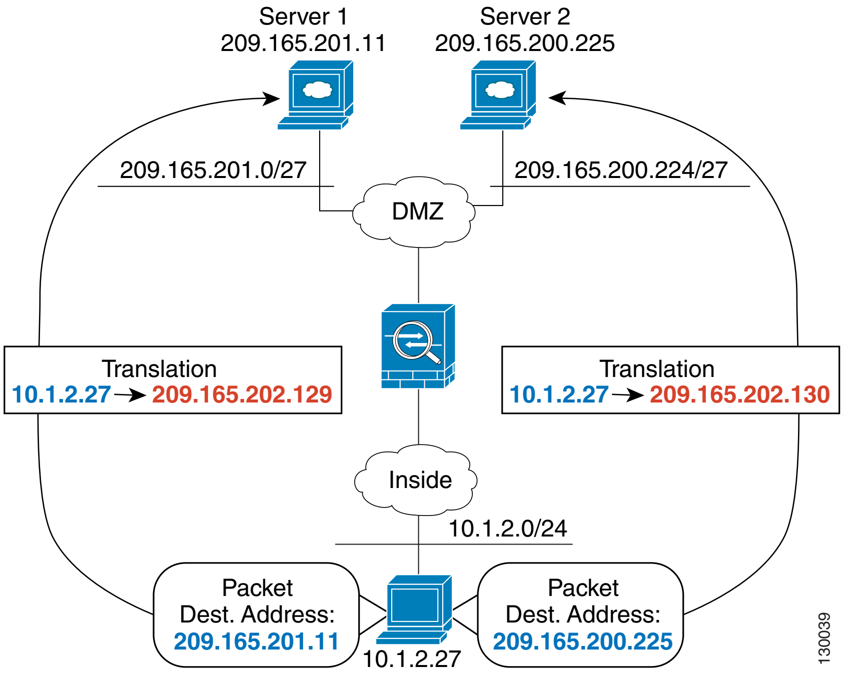
This section includes the following configuration examples:

- Different Translation Depending on the Destination (Dynamic PAT), page 5-25
- Different Translation Depending on the Destination Address and Port (Dynamic PAT), page 5-27

Different Translation Depending on the Destination (Dynamic PAT)

Figure 5-1 shows a host on the 10.1.2.0/24 network accessing two different servers. When the host accesses the server at 209.165.201.11, the real address is translated to 209.165.202.129:*port*. When the host accesses the server at 209.165.200.225, the real address is translated to 209.165.202.130:*port*.

Figure 5-1 Twice NAT with Different Destination Addresses



Step 1 Add a network object for the inside network:

```
ciscoasa(config)# object network myInsideNetwork
ciscoasa(config-network-object)# subnet 10.1.2.0 255.255.255.0
```

Step 2 Add a network object for the DMZ network 1:

```
ciscoasa(config)# object network DMZnetwork1
ciscoasa(config-network-object)# subnet 209.165.201.0 255.255.255.224
```

Step 3 Add a network object for the PAT address:

```
ciscoasa(config)# object network PATaddress1
ciscoasa(config-network-object)# host 209.165.202.129
```

Step 4 Configure the first twice NAT rule:

```
ciscoasa(config)# nat (inside,dmz) source dynamic myInsideNetwork PATaddress1 destination
static DMZnetwork1 DMZnetwork1
```

Because you do not want to translate the destination address, you need to configure identity NAT for it by specifying the same address for the real and mapped destination addresses.

By default, the NAT rule is added to the end of section 1 of the NAT table, See the "Configuring Dynamic PAT (Hide)" section on page 5-11 for more information about specifying the section and line number for the NAT rule.

Step 5 Add a network object for the DMZ network 2:

```
ciscoasa(config)# object network DMZnetwork2
ciscoasa(config-network-object)# subnet 209.165.200.224 255.255.255.224
```

Step 6 Add a network object for the PAT address:

```
ciscoasa(config)# object network PATaddress2
ciscoasa(config-network-object)# host 209.165.202.130
```

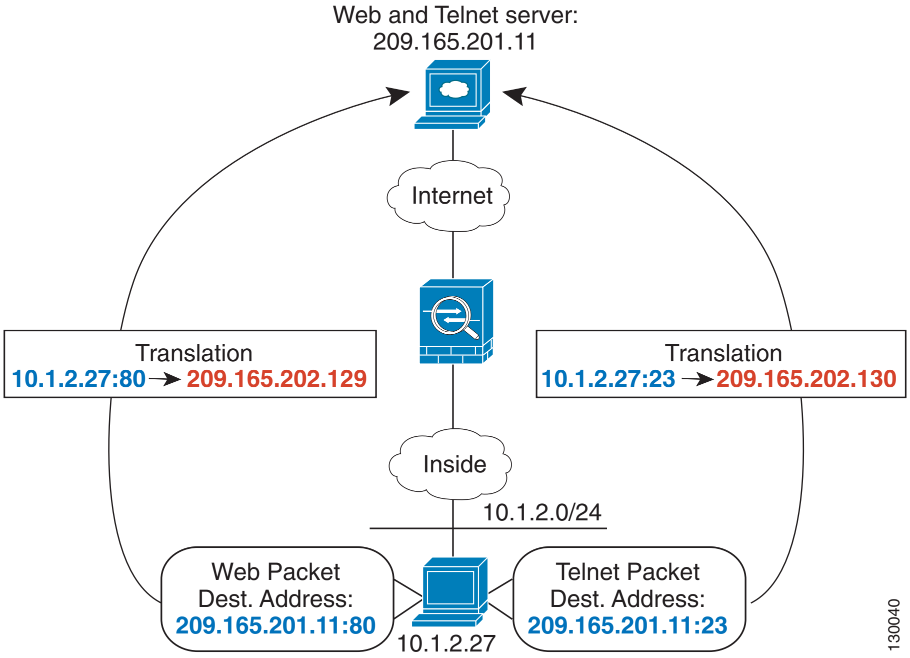
Step 7 Configure the second twice NAT rule:

```
ciscoasa(config)# nat (inside,dmz) source dynamic myInsideNetwork PATaddress2 destination
static DMZnetwork2 DMZnetwork2
```

Different Translation Depending on the Destination Address and Port (Dynamic PAT)

Figure 5-2 shows the use of source and destination ports. The host on the 10.1.2.0/24 network accesses a single host for both web services and Telnet services. When the host accesses the server for Telnet services, the real address is translated to 209.165.202.129:*port*. When the host accesses the same server for web services, the real address is translated to 209.165.202.130:*port*.

Figure 5-2 Twice NAT with Different Destination Ports



Step 1 Add a network object for the inside network:

```
ciscoasa(config)# object network myInsideNetwork
ciscoasa(config-network-object)# subnet 10.1.2.0 255.255.255.0
```

Step 2 Add a network object for the Telnet/Web server:

```
ciscoasa(config)# object network TelnetWebServer
ciscoasa(config-network-object)# host 209.165.201.11
```

Step 3 Add a network object for the PAT address when using Telnet:

```
ciscoasa(config)# object network PATaddress1
ciscoasa(config-network-object)# host 209.165.202.129
```

Step 4 Add a service object for Telnet:

```
ciscoasa(config)# object service TelnetObj
ciscoasa(config-network-object)# service tcp destination eq telnet
```

Step 5 Configure the first twice NAT rule:

```
ciscoasa(config)# nat (inside,outside) source dynamic myInsideNetwork PATaddress1
destination static TelnetWebServer TelnetWebServer service TelnetObj TelnetObj
```

Because you do not want to translate the destination address or port, you need to configure identity NAT for them by specifying the same address for the real and mapped destination addresses, and the same port for the real and mapped service.

By default, the NAT rule is added to the end of section 1 of the NAT table, See the "Configuring Dynamic PAT (Hide)" section on page 5-11 for more information about specifying the section and line number for the NAT rule.

Step 6 Add a network object for the PAT address when using HTTP:

```
ciscoasa(config)# object network PATaddress2
ciscoasa(config-network-object)# host 209.165.202.130
```

Step 7 Add a service object for HTTP:

```
ciscoasa(config)# object service HTTPObj
ciscoasa(config-network-object)# service tcp destination eq http
```

Step 8 Configure the second twice NAT rule:

```
ciscoasa(config)# nat (inside,outside) source dynamic myInsideNetwork PATaddress2
destination static TelnetWebServer TelnetWebServer service HTTPObj HTTPObj
```

Feature History for Twice NAT

Table 5-1 lists each feature change and the platform release in which it was implemented.

Table 5-1 Feature History for Twice NAT

Feature Name	Platform Releases	Feature Information
Twice NAT	8.3(1)	Twice NAT lets you identify both the source and destination address in a single rule. We modified or introduced the following commands: **nat**, **show nat**, **show xlate**, **show nat pool**.
Identity NAT configurable proxy ARP and route lookup	8.4(2)/8.5(1)	In earlier releases for identity NAT, proxy ARP was disabled, and a route lookup was always used to determine the egress interface. You could not configure these settings. In 8.4(2) and later, the default behavior for identity NAT was changed to match the behavior of other static NAT configurations: proxy ARP is enabled, and the NAT configuration determines the egress interface (if specified) by default. You can leave these settings as is, or you can enable or disable them discretely. Note that you can now also disable proxy ARP for regular static NAT. For pre-8.3 configurations, the migration of NAT exempt rules (the **nat 0 access-list** command) to 8.4(2) and later now includes the following keywords to disable proxy ARP and to use a route lookup: **no-proxy-arp** and **route-lookup**. The **unidirectional** keyword that was used for migrating to 8.3(2) and 8.4(1) is no longer used for migration. When upgrading to 8.4(2) from 8.3(1), 8.3(2), and 8.4(1), all identity NAT configurations will now include the **no-proxy-arp** and **route-lookup** keywords, to maintain existing functionality. The **unidirectional** keyword is removed. We modified the following command: **nat source static** [**no-proxy-arp**] [**route-lookup**].
PAT pool and round robin address assignment	8.4(2)/8.5(1)	You can now specify a pool of PAT addresses instead of a single address. You can also optionally enable round-robin assignment of PAT addresses instead of first using all ports on a PAT address before using the next address in the pool. These features help prevent a large number of connections from a single PAT address from appearing to be part of a DoS attack and makes configuration of large numbers of PAT addresses easy. We modified the following command: **nat source dynamic** [**pat-pool** *mapped_object* [**round-robin**]].

Table 5-1 Feature History for Twice NAT (continued)

Feature Name	Platform Releases	Feature Information
Round robin PAT pool allocation uses the same IP address for existing hosts	8.4(3)	When using a PAT pool with round robin allocation, if a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available. We did not modify any commands. *This feature is not available in 8.5(1) or 8.6(1).*
Flat range of PAT ports for a PAT pool	8.4(3)	If available, the real source port number is used for the mapped port. However, if the real port is *not* available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool. If you have a lot of traffic that uses the lower port ranges, when using a PAT pool, you can now specify a flat range of ports to be used instead of the three unequal-sized tiers: either 1024 to 65535, or 1 to 65535. We modified the following command: **nat source dynamic** [**pat-pool** *mapped_object* [**flat** [**include-reserve**]]]. *This feature is not available in 8.5(1) or 8.6(1).*
Extended PAT for a PAT pool	8.4(3)	Each PAT IP address allows up to 65535 ports. If 65535 ports do not provide enough translations, you can now enable extended PAT for a PAT pool. Extended PAT uses 65535 ports per *service*, as opposed to per IP address, by including the destination address and port in the translation information. We modified the following command: **nat source dynamic** [**pat-pool** *mapped_object* [**extended**]]. *This feature is not available in 8.5(1) or 8.6(1).*

Table 5-1 Feature History for Twice NAT (continued)

Feature Name	Platform Releases	Feature Information
Automatic NAT rules to translate a VPN peer's local IP address back to the peer's real IP address	8.4(3)	In rare situations, you might want to use a VPN peer's real IP address on the inside network instead of an assigned local IP address. Normally with VPN, the peer is given an assigned local IP address to access the inside network. However, you might want to translate the local IP address back to the peer's real public IP address if, for example, your inside servers and network security is based on the peer's real IP address. You can enable this feature on one interface per tunnel group. Object NAT rules are dynamically added and deleted when the VPN session is established or disconnected. You can view the rules using the **show nat** command. **Note** Because of routing issues, we do not recommend using this feature unless you know you need this feature; contact Cisco TAC to confirm feature compatibility with your network. See the following limitations: • Only supports Cisco IPsec and AnyConnect Client. • Return traffic to the public IP addresses must be routed back to the ASA so the NAT policy and VPN policy can be applied. • Does not support load-balancing (because of routing issues). • Does not support roaming (public IP changing). We introduced the following command: **nat-assigned-to-public-ip** *interface* (tunnel-group general-attributes configuration mode).
NAT support for IPv6	9.0(1)	NAT now supports IPv6 traffic, as well as translating between IPv4 and IPv6. Translating between IPv4 and IPv6 is not supported in transparent mode. We modified the following commands: **nat** (global configuration mode), **show nat**, **show nat pool**, **show xlate**.

Table 5-1 Feature History for Twice NAT (continued)

Feature Name	Platform Releases	Feature Information
NAT support for reverse DNS lookups	9.0(1)	NAT now supports translation of the DNS PTR record for reverse DNS lookups when using IPv4 NAT, IPv6 NAT, and NAT64 with DNS inspection enabled for the NAT rule.
Per-session PAT	9.0(1)	The per-session PAT feature improves the scalability of PAT and, for clustering, allows each member unit to own PAT connections; multi-session PAT connections have to be forwarded to and owned by the master unit. At the end of a per-session PAT session, the ASA sends a reset and immediately removes the xlate. This reset causes the end node to immediately release the connection, avoiding the TIME_WAIT state. Multi-session PAT, on the other hand, uses the PAT timeout, by default 30 seconds. For "hit-and-run" traffic, such as HTTP or HTTPS, the per-session feature can dramatically increase the connection rate supported by one address. Without the per-session feature, the maximum connection rate for one address for an IP protocol is approximately 2000 per second. With the per-session feature, the connection rate for one address for an IP protocol is 65535/*average-lifetime*. By default, all TCP traffic and UDP DNS traffic use a per-session PAT xlate. For traffic that requires multi-session PAT, such as H.323, SIP, or Skinny, you can disable per-session PAT by creating a per-session deny rule. We introduced the following commands: **xlate per-session**, **show nat pool**.

PART 3

Configuring Access Control

CHAPTER 6

Configuring Access Rules

This chapter describes how to control network access through the ASA using access rules and includes the following sections:

Note You use access rules to control network access in both routed and transparent firewall modes. In transparent mode, you can use both access rules (for Layer 3 traffic) and EtherType rules (for Layer 2 traffic).

To access the ASA interface for management access, you do not also need an access rule allowing the host IP address. You only need to configure management access according to the general operations configuration guide.

Information About Access Rules

You create an access rule by applying an extended or EtherType ACL to an interface or globally for all interfaces.You can use access rules in routed and transparent firewall mode to control IP traffic. An access rule permits or denies traffic based on the protocol, a source and destination IP address or network, and optionally the source and destination ports.

For transparent mode only, an EtherType rule controls network access for non-IP traffic. An EtherType rule permits or denies traffic based on the EtherType.

This section includes the following topics:

- Information About EtherType Rules, page 6-6

General Information About Rules

This section describes information for both access rules and EtherType rules, and it includes the following topics:

- Implicit Permits, page 6-2
- Information About Interface Access Rules and Global Access Rules, page 6-2
- Using Access Rules and EtherType Rules on the Same Interface, page 6-2
- Implicit Deny, page 6-3
- Inbound and Outbound Rules, page 6-3
- Transactional-Commit Model, page 6-4

Implicit Permits

For routed mode, the following types of traffic are allowed through by default:

- Unicast IPv4 traffic from a higher security interface to a lower security interface.
- Unicast IPv6 traffic from a higher security interface to a lower security interface.

For transparent mode, the following types of traffic are allowed through by default:

- Unicast IPv4 traffic from a higher security interface to a lower security interface.
- Unicast IPv6 traffic from a higher security interface to a lower security interface.
- ARPs in both directions.

  Note ARP traffic can be controlled by ARP inspection, but cannot be controlled by an access rule.

- BPDUs in both directions.

For other traffic, you need to use either an extended access rule (IPv4 and IPv6) or an EtherType rule (non-IPv4/IPv6).

Information About Interface Access Rules and Global Access Rules

You can apply an access rule to a specific interface, or you can apply an access rule globally to all interfaces. You can configure global access rules in conjunction with interface access rules, in which case, the specific interface access rules are always processed before the general global access rules.

Note Global access rules apply only to inbound traffic. See the "Inbound and Outbound Rules" section on page 6-3.

Using Access Rules and EtherType Rules on the Same Interface

You can apply one access rule and one EtherType rule to each direction of an interface.

Implicit Deny

ACLs have an implicit deny at the end of the list, so unless you explicitly permit it, traffic cannot pass. For example, if you want to allow all users to access a network through the ASA except for particular addresses, then you need to deny the particular addresses and then permit all others.

For EtherType ACLs, the implicit deny at the end of the ACL does not affect IP traffic or ARPs; for example, if you allow EtherType 8037, the implicit deny at the end of the ACL does not now block any IP traffic that you previously allowed with an extended ACL (or implicitly allowed from a high security interface to a low security interface). However, if you explicitly deny all traffic with an EtherType ACE, then IP and ARP traffic is denied.

If you configure a global access rule, then the implicit deny comes *after* the global rule is processed. See the following order of operations:

1. Interface access rule.
2. Global access rule.
3. Implicit deny.

Inbound and Outbound Rules

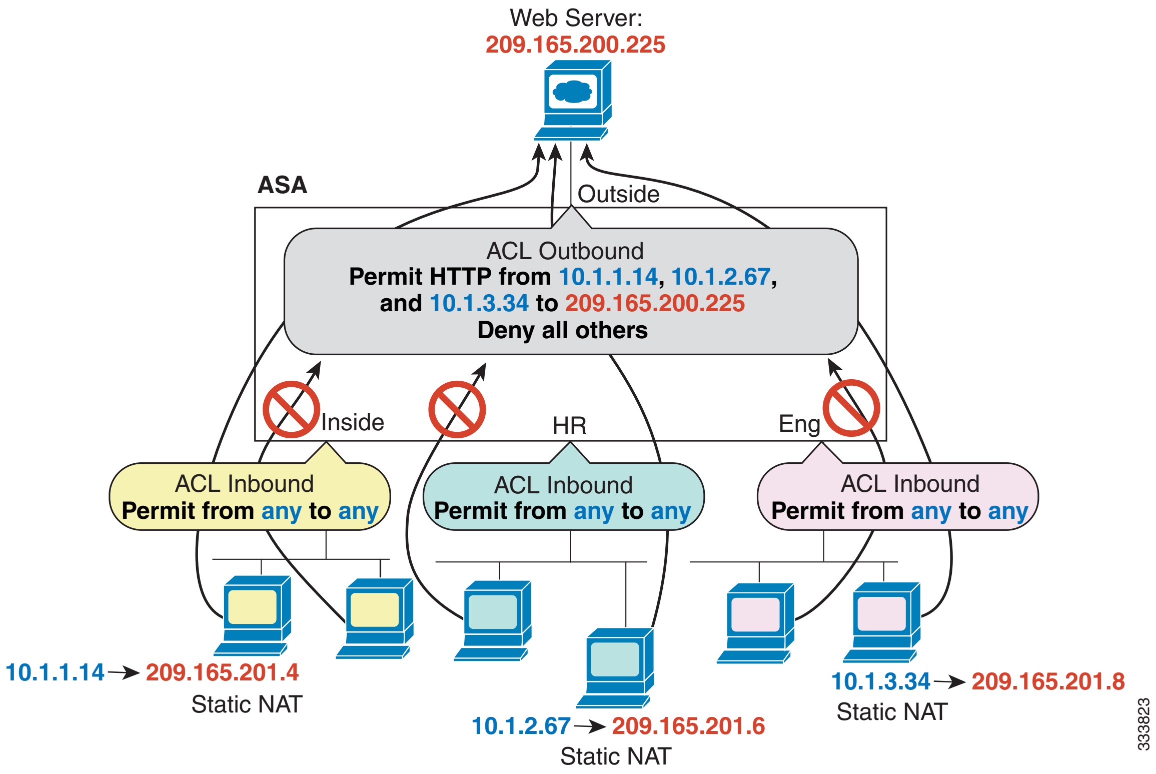
The ASA supports two types of ACLs:

- Inbound—Inbound access rules apply to traffic as it enters an interface. Global access rules are always inbound.
- Outbound—Outbound ACLs apply to traffic as it exits an interface.

Note "Inbound" and "outbound" refer to the application of an ACL on an interface, either to traffic entering the ASA on an interface or traffic exiting the ASA on an interface. These terms do not refer to the movement of traffic from a lower security interface to a higher security interface, commonly known as inbound, or from a higher to lower interface, commonly known as outbound.

An outbound ACL is useful, for example, if you want to allow only certain hosts on the inside networks to access a web server on the outside network. Rather than creating multiple inbound ACLs to restrict access, you can create a single outbound ACL that allows only the specified hosts. (See Figure 6-1.) The outbound ACL prevents any other hosts from reaching the outside network.

Figure 6-1 Outbound ACL



See the following commands for this example:

```
ciscoasa(config)# access-list OUTSIDE extended permit tcp host 10.1.1.14
host 209.165.200.225 eq www
ciscoasa(config)# access-list OUTSIDE extended permit tcp host 10.1.2.67
host 209.165.200.225 eq www
ciscoasa(config)# access-list OUTSIDE extended permit tcp host 10.1.3.34
host 209.165.200.225 eq www
ciscoasa(config)# access-group OUTSIDE out interface outside
```

Transactional-Commit Model

The ASA rule-engine supports a new feature for rule updation called the Transactional-Commit Model. When this feature is enabled, a rule update is applied after the rule compilation is completed; without affecting the rule matching performance. With the legacy model, rule updates take effect immediately but rule matching slows down during the rule compilation period. This feature is useful to prevent potential packet drops during large compilation of rules under high traffic conditions. This feature is also useful to reduce the rule compilation time under two specific patterns of configurations:

- Preventing packet drops while compiling large rules during high traffic rates.
- Reducing rule compilation time while updating a large number of similar rules.

Guidelines and Limitations

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Supports IPv6.

Additional Guidelines and Limitations

Evaluate the following alternatives before using the transactional commit model:

- While using large rules, try to optimize the number of rules by using the Object Group Search setting in Advanced Access Rule Configuration settings.
- Perform an incremental rule update instead of a bulk rule update. If a bulk update is necessary perform the bulk update during the maintenance window, when traffic is low.

Information About Extended Access Rules

This section describes information about extended access rules and includes the following topics:

- Access Rules for Returning Traffic, page 6-5
- Allowing Broadcast and Multicast Traffic through the Transparent Firewall Using Access Rules, page 6-5
- Management Access Rules, page 6-6

Access Rules for Returning Traffic

For TCP and UDP connections for both routed and transparent mode, you do not need an access rule to allow returning traffic because the ASA allows all returning traffic for established, bidirectional connections.

For connectionless protocols such as ICMP, however, the ASA establishes unidirectional sessions, so you either need access rules to allow ICMP in both directions (by applying ACLs to the source and destination interfaces), or you need to enable the ICMP inspection engine. The ICMP inspection engine treats ICMP sessions as bidirectional connections. To control ping, specify **echo-reply** (**0**) (ASA to host) or **echo** (**8**) (host to ASA).

Allowing Broadcast and Multicast Traffic through the Transparent Firewall Using Access Rules

In routed firewall mode, broadcast and multicast traffic is blocked even if you allow it in an access rule, including unsupported dynamic routing protocols and DHCP (unless you configure DHCP relay). Transparent firewall mode can allow any IP traffic through.

Note Because these special types of traffic are connectionless, you need to apply an access rule to both interfaces, so returning traffic is allowed through.

Table 6-1 lists common traffic types that you can allow through the transparent firewall.

Table 6-1 Transparent Firewall Special Traffic

Traffic Type	Protocol or Port	Notes
DHCP	UDP ports 67 and 68	If you enable the DHCP server, then the ASA does not pass DHCP packets.
EIGRP	Protocol 88	—
OSPF	Protocol 89	—
Multicast streams	The UDP ports vary depending on the application.	Multicast streams are always destined to a Class D address (224.0.0.0 to 239.x.x.x).
RIP (v1 or v2)	UDP port 520	—

Management Access Rules

You can configure access rules that control management traffic destined to the ASA. Access control rules for to-the-box management traffic (defined by such commands as **http**, **ssh**, or **telnet**) have higher precedence than an management access rule applied with the **control-plane** option. Therefore, such permitted management traffic will be allowed to come in even if explicitly denied by the to-the-box ACL.

Information About EtherType Rules

This section describes EtherType rules and includes the following topics:

- Supported EtherTypes and Other Traffic, page 6-6
- Access Rules for Returning Traffic, page 6-7
- Allowing MPLS, page 6-7

Supported EtherTypes and Other Traffic

An EtherType rule controls the following:

- EtherType identified by a 16-bit hexadecimal number, including common types IPX and MPLS unicast or multicast.
- Ethernet V2 frames.
- BPDUs, which are permitted by default. BPDUs are SNAP-encapsulated, and the ASA is designed to specifically handle BPDUs.
- Trunk port (Cisco proprietary) BPDUs. Trunk BPDUs have VLAN information inside the payload, so the ASA modifies the payload with the outgoing VLAN if you allow BPDUs.
- IS-IS.

The following types of traffic are not supported:

- 802.3-formatted frames—These frames are not handled by the rule because they use a length field as opposed to a type field.

Access Rules for Returning Traffic

Because EtherTypes are connectionless, you need to apply the rule to both interfaces if you want traffic to pass in both directions.

Allowing MPLS

If you allow MPLS, ensure that Label Distribution Protocol and Tag Distribution Protocol TCP connections are established through the ASA by configuring both MPLS routers connected to the ASA to use the IP address on the ASA interface as the router-id for LDP or TDP sessions. (LDP and TDP allow MPLS routers to negotiate the labels (addresses) used to forward packets.)

On Cisco IOS routers, enter the appropriate command for your protocol, LDP or TDP. The *interface* is the interface connected to the ASA.

```
ciscoasa(config)# mpls ldp router-id interface force
```

Or

```
ciscoasa(config)# tag-switching tdp router-id interface force
```

Licensing Requirements for Access Rules

Model	License Requirement
All models	Base License.

Prerequisites

Before you can create an access rule, create the ACL. See the general operations configuration guide for more information.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall modes.

IPv6 Guidelines

Supports IPv6. The source and destination addresses can include any mix of IPv4 and IPv6 addresses.

Per-User ACL Guidelines

- The per-user ACL uses the value in the **timeout uauth** command, but it can be overridden by the AAA per-user session timeout value.
- If traffic is denied because of a per-user ACL, syslog message 109025 is logged. If traffic is permitted, no syslog message is generated. The **log** option in the per-user ACL has no effect.

Default Settings

See the "Implicit Permits" section on page 6-2.

Configuring Access Rules

To apply an access rule, perform the following steps.

Detailed Steps

<table>
<tr><th>Command</th><th>Purpose</th></tr>
<tr><td>**access-group** *access_list* {{**in** | **out**} **interface** *interface_name* [**per-user-override** | **control-plane**] | **global**}

Example:
ciscoasa(config)# access-group outside_access in interface outside</td><td>Binds an ACL to an interface or applies it globally.
Specify the extended or EtherType ACL name. You can configure one **access-group** command per ACL type per interface. You cannot reference empty ACLs or ACLs that contain only a remark.
For an interface-specific rule:
• The **in** keyword applies the ACL to inbound traffic. The **out** keyword applies the ACL to the outbound traffic.
• Specify the **interface** name.
• The **per-user-override** keyword (for inbound ACLs only) allows dynamic user ACLs that are downloaded for user authorization to override the ACL assigned to the interface. For example, if the interface ACL denies all traffic from 10.0.0.0, but the dynamic ACL permits all traffic from 10.0.0.0, then the dynamic ACL overrides the interface ACL for that user.
By default, VPN remote access traffic is not matched against interface ACLs. However, if you use the **no sysopt connection permit-vpn** command to turn off this bypass, the behavior depends on whether there is a **vpn-filter** applied in the group policy and whether you set the **per-user-override** option:
– No **per-user-override**, no **vpn-filter**—Traffic is matched against the interface ACL.
– No **per-user-override**, **vpn-filter**—Traffic is matched first against the interface ACL, then against the VPN filter.
– **per-user-override**, **vpn-filter**—Traffic is matched against the VPN filter only.
See Per-User ACL Guidelines, page 6-8.
• The **control-plane** keyword specifies if the rule is for to-the-box traffic.
For a global rule, specify the **global** keyword to apply the ACL to the inbound direction of all interfaces.</td></tr>
</table>

Examples

The following example shows how to use the **access-group** command:

```
hostname(config)# access-list outside_access permit tcp any host 209.165.201.3 eq 80
hostname(config)# access-group outside_access interface outside
```

The **access-list** command lets any host access the global address using port 80. The **access-group** command specifies that the **access-list** command applies to traffic entering the outside interface.

Monitoring Access Rules

To monitor network access, enter the following command:

Command	Purpose
show running-config access-group	Displays the current ACL bound to the interfaces.

Configuration Examples for Permitting or Denying Network Access

This section includes typical configuration examples for permitting or denying network access.

The following example adds a network object for inside server 1, performs static NAT for the server, and enables access to from the outside for inside server 1.

```
ciscoasa(config)# object network inside-server1
ciscoasa(config)# host 10.1.1.1
ciscoasa(config)# nat (inside,outside) static 209.165.201.12

ciscoasa(config)# access-list outside_access extended permit tcp any object inside-server1
eq www
ciscoasa(config)# access-group outside_access in interface outside
```

The following example allows all hosts to communicate between the **inside** and **hr** networks but only specific hosts to access the outside network:

```
ciscoasa(config)# access-list ANY extended permit ip any any
ciscoasa(config)# access-list OUT extended permit ip host 209.168.200.3 any
ciscoasa(config)# access-list OUT extended permit ip host 209.168.200.4 any

ciscoasa(config)# access-group ANY in interface inside
ciscoasa(config)# access-group ANY in interface hr
ciscoasa(config)# access-group OUT out interface outside
```

For example, the following sample ACL allows common EtherTypes originating on the inside interface:

```
ciscoasa(config)# access-list ETHER ethertype permit ipx
ciscoasa(config)# access-list ETHER ethertype permit mpls-unicast
ciscoasa(config)# access-group ETHER in interface inside
```

The following example allows some EtherTypes through the ASA, but it denies all others:

```
ciscoasa(config)# access-list ETHER ethertype permit 0x1234
ciscoasa(config)# access-list ETHER ethertype permit mpls-unicast
ciscoasa(config)# access-group ETHER in interface inside
ciscoasa(config)# access-group ETHER in interface outside
```

The following example denies traffic with EtherType 0x1256 but allows all others on both interfaces:

```
ciscoasa(config)# access-list nonIP ethertype deny 1256
ciscoasa(config)# access-list nonIP ethertype permit any
ciscoasa(config)# access-group ETHER in interface inside
ciscoasa(config)# access-group ETHER in interface outside
```

The following example uses object groups to permit specific traffic on the inside interface:

```
!
hostname (config)# object-group service myaclog
hostname (config-service)# service-object tcp source range 2000 3000
hostname (config-service)# service-object tcp source range 3000 3010 destinatio$
hostname (config-service)# service-object ipsec
hostname (config-service)# service-object udp destination range 1002 1006
hostname (config-service)# service-object icmp echo

ciscoasa(config)# access-list outsideacl extended permit object-group myaclog interface
inside any
```

Feature History for Access Rules

Table 6-2 lists each feature change and the platform release in which it was implemented.

Table 6-2 Feature History for Access Rules

Feature Name	Platform Releases	Feature Information
Interface access rules	7.0(1)	Controlling network access through the ASA using ACLs. We introduced the following command: **access-group**.
Global access rules	8.3(1)	Global access rules were introduced. We modified the following command: **access-group.**
Support for Identity Firewall	8.4(2)	You can now use identity firewall users and groups for the source and destination. You can use an identity firewall ACL with access rules, AAA rules, and for VPN authentication. We modified the following commands: **access-list extended**.
EtherType ACL support for IS-IS traffic	8.4(5), 9.1(2)	In transparent firewall mode, the ASA can now pass IS-IS traffic using an EtherType ACL. We modified the following command: **access-list ethertype** {**permit** | **deny**} **is-is**.
Support for TrustSec	9.0(1)	You can now use TrustSec security groups for the source and destination. You can use an identity firewall ACL with access rules. We modified the following commands: **access-list extended**.

Table 6-2 Feature History for Access Rules (continued)

Feature Name	Platform Releases	Feature Information
Unified ACL for IPv4 and IPv6	9.0(1)	ACLs now support IPv4 and IPv6 addresses. You can even specify a mix of IPv4 and IPv6 addresses for the source and destination. The **any** keyword was changed to represent IPv4 and IPv6 traffic. The **any4** and **any6** keywords were added to represent IPv4-only and IPv6-only traffic, respectively. The IPv6-specific ACLs are deprecated. Existing IPv6 ACLs are migrated to extended ACLs. See the release notes for more information about migration. We modified the following commands: **access-list extended**, **access-list webtype**. We removed the following commands: **ipv6 access-list**, **ipv6 access-list webtype**, **ipv6-vpn-filter**
Extended ACLand object enhancement to filter ICMP traffic by ICMP code	9.0(1)	ICMP traffic can now be permitted/denied based on ICMP code. We introduced or modified the following commands: **access-list extended, service-object, service**.
Transactional Commit Model on Rule Engine for Access groups	9.1(5)	When enabled, a rule update is applied after the rule compilation is completed; without affecting the rule matching performance. We introduced the following commands: **asp rule-engine transactional-commit**, **show running-config asp rule-engine transactional-commit**, **clear configure asp rule-engine transactional-commit**.

Configuring AAA Rules for Network Access

This chapter describes how to enable AAA (pronounced “triple A”) for network access.

For information about AAA for management access, see the general operations configuration guide.

This chapter includes the following sections:

AAA Performance

The ASA uses “cut-through proxy” to significantly improve performance compared to a traditional proxy server. The performance of a traditional proxy server suffers because it analyzes every packet at the application layer of the OSI model. The ASA cut-through proxy challenges a user initially at the application layer and then authenticates with standard AAA servers or the local database. After the ASA authenticates the user, it shifts the session flow, and all traffic flows directly and quickly between the source and destination while maintaining session state information.

Licensing Requirements for AAA Rules

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Supports IPv6.

Additional Guidelines

In clustering, this feature is only supported on the master unit.

Configuring Authentication for Network Access

This section includes the following topics:

- Information About Authentication, page 7-2
- Configuring Network Access Authentication, page 7-7
- Enabling Secure Authentication of Web Clients, page 7-10
- Authenticating Directly with the ASA, page 7-11

Information About Authentication

The ASA lets you configure network access authentication using AAA servers. This section includes the following topics:

- One-Time Authentication, page 7-3
- Applications Required to Receive an Authentication Challenge, page 7-3
- ASA Authentication Prompts, page 7-3
- AAA Prompts and Identity Firewall, page 7-4
- AAA Rules as a Backup Authentication Method, page 7-5
- Static PAT and HTTP, page 7-5

One-Time Authentication

A user at a given IP address only needs to authenticate one time for all rules and types, until the authentication session expires. (See the **timeout uauth** command in the command reference for timeout values.) For example, if you configure the ASA to authenticate Telnet and FTP, and a user first successfully authenticates for Telnet, then as long as the authentication session exists, the user does not also have to authenticate for FTP.

Applications Required to Receive an Authentication Challenge

Although you can configure the ASA to require authentication for network access to any protocol or service, users can authenticate directly with HTTP, HTTPS, Telnet, or FTP only. A user must first authenticate with one of these services before the ASA allows other traffic requiring authentication.

The authentication ports that the ASA supports for AAA are fixed as follows:

- Port 21 for FTP
- Port 23 for Telnet
- Port 80 for HTTP
- Port 443 for HTTPS

ASA Authentication Prompts

For Telnet and FTP, the ASA generates an authentication prompt.

For HTTP, the ASA uses basic HTTP authentication by default, and provides an authentication prompt. You can optionally configure the ASA to redirect users to an internal web page where they can enter their username and password (configured with the **aaa authentication listener** command).

For HTTPS, the ASA generates a custom login screen. You can optionally configure the ASA to redirect users to an internal web page where they can enter their username and password (configured with the **aaa authentication listener** command).

Redirection is an improvement over the basic method because it provides an improved user experience during authentication, and an identical user experience for HTTP and HTTPS in both Easy VPN and firewall modes. It also supports authentication directly with the ASA.

You might want to continue to use basic HTTP authentication for the following reasons:

- You do not want the ASA to open listening ports.
- You use NAT on a router and you do not want to create a translation rule for the web page served by the ASA.
- Basic HTTP authentication might work better with your network.

For example non-browser applications, as when a URL is embedded in e-mail, might be more compatible with basic authentication.

After you authenticate correctly, the ASA redirects you to your original destination. If the destination server also has its own authentication, the user enters another username and password. If you use basic HTTP authentication and need to enter another username and password for the destination server, then you need to configure the **virtual http** command.

Note If you use HTTP authentication, by default the username and password are sent from the client to the ASA in clear text; in addition, the username and password are sent on to the destination web server as well. See the "Enabling Secure Authentication of Web Clients" section on page 7-10 for information to secure your credentials.

For FTP, a user has the option of entering the ASA username followed by an at sign (@) and then the FTP username (name1@name2). For the password, the user enters the ASA password followed by an at sign (@) and then the FTP password (password1@password2). For example, enter the following text:

```
name> name1@name2
password> password1@password2
```

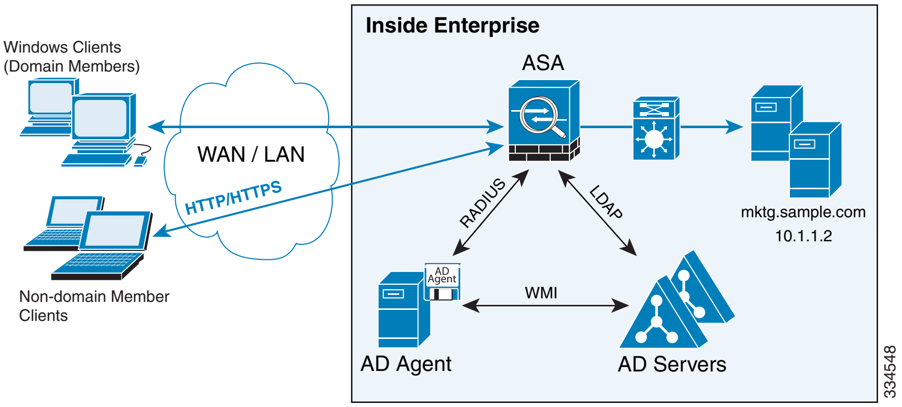
This feature is useful when you have cascaded firewalls that require multiple logins. You can separate several names and passwords by multiple at signs (@).

AAA Prompts and Identity Firewall

In an enterprise, some users log into the network by using other authentication mechanisms, such as authenticating with a web portal (cut-through proxy). For example, users with a Mac and Linux client might log into a web portal (cut-through proxy). Therefore, you must configure the identity firewall to allow these types of authentication in connection with identity-based access policies.

Figure 7-1 shows a deployment to support a cut-through proxy authentication captive portal. Active Directory servers and the AD Agent are installed on the main site LAN. However, the identity firewall is configured to support authentication of clients that are not part of the Active Directory domain.

Figure 7-1 Deployment Supporting Cut-through Proxy Authentication



The ASA designates users logging in through a web portal (cut-through proxy) as belonging to the Active Directory domain with which they authenticated.

The ASA reports users logging in through a web portal (cut-through proxy) to the AD Agent, which distributes the user information to all registered ASA devices. In this case, the identity firewall can associate the users with their Active Directory domain. Specifically, the user identity-IP address mappings of authenticated users are forwarded to all ASA contexts that contain the input interface where packets are received and authenticated.

Users can log in by using HTTP/HTTPS, FTP, Telnet, or SSH. When users log in with these authentication methods, the following guidelines apply:

- For HTTP/HTTPS traffic, an authentication window appears for unauthenticated users.

- For Telnet and FTP traffic, users must log in through the cut-through proxy server and again to the Telnet and FTP servers.
- A user can specify an Active Directory domain while providing login credentials (in the format, domain\username). The ASA automatically selects the associated AAA server group for the specified domain.
- If a user specifies an Active Directory domain while providing login credentials (in the format, domain\username), the ASA parses the domain and uses it to select an authentication server from the AAA servers that have been configured for the identity firewall. Only the username is passed to the AAA server.
- If the backslash (\) delimiter is not found in the login credentials, the ASA does not parse the domain and authentication is conducted with the AAA server that corresponds to the default domain configured for the identity firewall.
- If a default domain or a server group is not configured for that default domain, the ASA rejects the authentication.
- If the domain is not specified, the ASA selects the AAA server group for the default domain that is configured for the identity firewall.

AAA Rules as a Backup Authentication Method

An authentication rule (also known as "cut-through proxy") controls network access based on the user. Because this function is very similar to an access rule plus an identity firewall, AAA rules can now be used as a backup method of authentication if a user AD login expires or a valid user has not yet logged into AD. For example, for any user without a valid login, you can trigger a AAA rule. To ensure that the AAA rule is only triggered for users that do not have valid logins, you can specify special usernames in the extended ACL that are used for the access rule and for the AAA rule: None (users without a valid login) and Any (users with a valid login). In the access rule, configure your policy as usual for users and groups, but then include a rule that permits all None users before deny any any; you must permit these users so they can later trigger a AAA rule. Then, configure a AAA rule that does not match Any users (these users are not subject to the AAA rule, and were handled already by the access rule), but matches all None users only to trigger AAA authentication for these users. After the user has successfully logged in via cut-through proxy, the traffic will flow normally again.

Static PAT and HTTP

For HTTP authentication, the ASA checks real ports when static PAT is configured. If it detects traffic destined for real port 80, regardless of the mapped port, the ASA intercepts the HTTP connection and enforces authentication.

For example, assume that outside TCP port 889 is translated to port 80 and that any relevant ACLs permit the traffic:

```
object network obj-192.168.123.10-01
    host 192.168.123.10
    nat (inside,outside) static 10.48.66.155 service tcp 80 889
```

Then when users try to access 10.48.66.155 on port 889, the ASA intercepts the traffic and enforces HTTP authentication. Users see the HTTP authentication page in their web browsers before the ASA allows HTTP connection to complete.

If the local port is different than port 80, as in the following example:

```
object network obj-192.168.123.10-02
    host 192.168.123.10
    nat (inside,outside) static 10.48.66.155 service tcp 111 889
```

Then users do not see the authentication page. Instead, the ASA sends an error message to the web browser, indicating that the user must be authenticated before using the requested service.

When a mapped address is used for static PAT, it is automatically placed into the dynamic PAT pool.

For instance, this configuration,

```
object network my-ftp-server
    host <real-server>
    nat (inside,outside) static <mapped-server> ftp ftp
```

is equivalent to

```
object network my-ftp-server
    host <real-server>
    nat (inside,outside) static <mapped-server> ftp ftp

object network <internal>
    nat (inside,outside) dynamic <mapped-server>
```

The second line ensures that all PAT bindings are accounted for.This accounting is necessary to avoid connection failure from port collision.

As the the mapped address is placed under dynamic PAT, any additional service that is to be accessed through the mapped address, must also be explicitly configured.

For example, the following is the correct configuration for three services through address 192.150.49.10. Additionally, the SMTP and HTTP services also reside at a host with the same address as the mapped address, 192.150.49.10.

```
object network my-ftp-server
    host <real-server>
    nat (inside,outside) static <mapped-server> ftp ftp

object network my-ftp-server
    host "192.150.49.10"
    nat (inside,outside) static 192.150.49.10 smtp smtp

object network my-ftp-server
    host "192.150.49.10"
    nat (inside,outside) static 192.150.49.10 http http
```

Configuring Network Access Authentication

To configure network access authentication, perform the following steps:

	Command	Purpose
Step 1	**aaa-server** **Example:** `ciscoasa(config)# aaa-server AuthOutbound protocol tacacs+`	Identifies your AAA servers. If you have already identified them, continue to the next step.
Step 2	**access-list** *access_list_name* **extended** {**deny** | **permit**} {**tcp** | **udp**} [*user_argument*] [*security_group_argument*] *source_address_argument* [*port_argument*] [*security_group_argument*] *dest_address_argument* [*port_argument*] **Example:** `ciscoasa(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp`	Creates an ACL that identifies the source addresses and destination addresses of traffic that you want to authenticate. The syntax shown here is just an example. For details, see the general operations configuration guide. If you specify identity firewall arguments in the ACL, then the following keywords in the ACL are specifically relevant to AAA rules. The keywords **user-group any** and **user-group none** can be specified to support cut-through proxy authentication. • **any**—The ACL matches any IP addresses that has already been associated with any users. • **none**—The ACL matches any IP addresses that has not been associated with any IP address.
Step 3	**aaa authentication match** *acl_name interface_name server_group* [**user-identity**] **Example:** `ciscoasa(config)# aaa authentication match MAIL_AUTH inside AuthOutbound`	Configures authentication. The *acl_name* argument is the name of the ACL that you created in Step 2. The *interface_name* argument is the name of the interface specified with the **nameif** command. The *server_group* argument is the AAA server group that you created in Step 1. **Note** You can alternatively use the **aaa authentication include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the command reference for more information. The **user-identity** keyword matches authentication to the identity firewall.

	Command	Purpose
Step 4	**aaa authentication listener http**[**s**] *interface_name* [**port** *portnum*] **redirect** **Example:** `ciscoasa(config)# aaa authentication listener http inside redirect`	(Optional) Enables the redirection method of authentication for HTTP or HTTPS connections. The *interface_name* argument is the interface on which you want to enable listening ports. The **port** *portnum* argument specifies the port number on which the ASA listens; the defaults are 80 (HTTP) and 443 (HTTPS). You can use any port number and retain the same functionality, but be sure your direct authentication users know the port number; redirected traffic is sent to the correct port number automatically, but direct authenticators must specify the port number manually. Enter this command separately for HTTP and for HTTPS.
Step 5	**aaa local authentication attempts max-fail** *number* **Example:** `ciscoasa(config)# aaa local authentication attempts max-fail 7`	(Optional) Uses the local database for network access authentication and limits the number of consecutive failed login attempts that the ASA allows any given user account (with the exception of users with a privilege level of 15. This feature does not affect level 15 users). The *number* argument value is between 1 and 16. **Tip** To clear the lockout status of a specific user or all users, use the **clear aaa local user lockout** command.

Examples

The following example authenticates all inside HTTP traffic and SMTP traffic:

```
ciscoasa(config)# aaa-server AuthOutbound protocol tacacs+
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
ciscoasa(config-aaa-server-host)# key TACPlusUauthKey
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp
ciscoasa(config)# access-list MAIL_AUTH extended permit tcp any any eq www
ciscoasa(config)# aaa authentication match MAIL_AUTH inside AuthOutbound
ciscoasa(config)# aaa authentication listener http inside redirect
```

The following example authenticates Telnet traffic from the outside interface to a particular server (209.165.201.5):

```
ciscoasa(config)# aaa-server AuthInbound protocol tacacs+
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server AuthInbound (inside) host 10.1.1.1
ciscoasa(config-aaa-server-host)# key TACPlusUauthKey
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# access-list TELNET_AUTH extended permit tcp any host 209.165.201.5 eq
telnet
ciscoasa(config)# aaa authentication match TELNET_AUTH outside AuthInbound
```

The following example shows a typical cut-through proxy configuration to allow a user to log in through the ASA. In this example, the following conditions apply:

- The ASA IP address is 192.168.123.10.
- The Active Directory domain controller has the IP address 10.1.2.10.
- The end user client has the IP address 192.168.123.10 and uses HTTPS to log in through a web portal.
- The user is authenticated by the Active Directory domain controller via LDAP.
- The ASA uses the inside interface to connect to the Active Directory domain controller on the corporate network.

```
hostname(config)# access-list AUTH extended permit tcp any 192.168.123.10 255.255.255.0 eq http
hostname(config)# access-list AUTH extended permit tcp any 192.168.123.10 255.255.255.0 eq https
hostname(config)# aaa-server LDAP protocol ldap
hostname(config-aaa-server-group)# aaa-server LDAP (inside) host 10.1.2.10
hostname(config-aaa-server-host)#  ldap-base-dn DC=cisco,DC=com
hostname(config-aaa-server-host)#  ldap-group-base-dn DC=cisco,DC=com
hostname(config-aaa-server-host)#  ldap-scope subtree
hostname(config-aaa-server-host)#  ldap-login-dn cn=kao,OU=Employees,OU=Cisco  Users,DC=cisco,DC=com
hostname(config-aaa-server-host)#  ldap-login-password *****
hostname(config-aaa-server-host)#  ldap-over-ssl enable
hostname(config-aaa-server-host)#  server-type microsoft
hostname(config-aaa-server-host)# aaa authentication match AUTH inside LDAP
hostname(config)#
hostname(config)# http server enable
hostname(config)# http 0.0.0.0 0.0.0.0 inside
hostname(config)#
hostname(config)# auth-prompt prompt Enter Your Authentication
hostname(config)# auth-prompt accept You are Good
hostname(config)# auth-prompt reject Goodbye
```

In this example, the following guidelines apply:

- In **access-list** commands, you should configure permit user NONE rules before entering the **access-list 100 ex deny any any** command to allow unauthenticated incoming users to trigger AAA cut-through proxy.
- In **access-list AUTH** commands, permit user NONE rules specify that only unauthenticated users can trigger AAA cut-through proxy.

```
hostname(config)# access-list listenerAuth extended permit tcp any any
hostname(config)# aaa authentication match listenerAuth inside ldap
hostname(config)# aaa authentication listener http inside port 8888
hostname(config)# access-list 100 ex permit ip user SAMPLE\user1 any any
hostname(config)# access-list 100 ex deny ip user SAMPLE\user2 any any
hostname(config)# access-list 100 ex permit ip user NONE any any
hostname(config)# access-list 100 ex deny any any
hostname(config)# access-group 100 in interface inside
hostname(config)# aaa authenticate match 100 inside user-identity
```

The following example shows how you can use AAA rules plus identity firewall (cut-through proxy) to authenticate successfully:

```
hostname(config)# access-list 100 ex permit ip user CISCO\xyz any any
hostname(config)# access-list 100 ex deny ip user CISCO\abc any any
hostname(config)# access-list 100 ex permit ip user NONE any any
hostname(config)# access-list 100 ex deny any any
hostname(config)# access-group 100 in interface inside
hostname(config)# access-list 200 ex permit user NONE any any
hostname(config)# aaa authenticate match 200 inside user-identity
```

For more information about authentication, see the "Information About Authentication" section on page 7-2.

Enabling Secure Authentication of Web Clients

If you use HTTP authentication, by default the username and password are sent from the client to the ASA in clear text; in addition, the username and password are sent to the destination web server as well.

The ASA provides the following methods for securing HTTP authentication:

- Enable the redirection method of authentication for HTTP—Use the **aaa authentication listener** command with the **redirect** keyword. This method prevents the authentication credentials from continuing to the destination server. See the "ASA Authentication Prompts" section on page 7-3 for more information about the redirection method compared to the basic method.
- Enable virtual HTTP—Use the **virtual http** command to authenticateseparately with the ASA and with the HTTP server. Even if the HTTP server does not need a second authentication, this command achieves the effect of stripping the basic authentication credentials from the HTTP GET request. See the "Authenticating HTTP(S) Connections with a Virtual Server" section on page 7-11 for more information.

  Enable the exchange of usernames and passwords between a web client and the ASA with HTTPS—Use the **aaa authentication secure-http-client** command to enable the exchange of usernames and passwords between a web client and the ASA with HTTPS. This is the only method that protects credentials between the client and the ASA, as well as between the ASA and the destination server. You can use this method alone, or in conjunction with either of the other methods so you can maximize your security.

  After enabling this feature, when a user requires authentication when using HTTP, the ASA redirects the HTTP user to an HTTPS prompt. After you authenticate correctly, the ASA redirects you to the original HTTP URL.

  Secured, web-client authentication has the following limitations:

  – A maximum of 64 concurrent HTTPS authentication sessions are allowed. If all 64 HTTPS authentication processes are running, a new connection requiring authentication will not succeed.

  – When **uauth timeout 0** is configured (the **uauth timeout** is set to 0),HTTPS authentication might not work. If a browser initiates multiple TCP connections to load a web page after HTTPS authentication, the first connection is let through, but the subsequent connections trigger authentication. As a result, users are continuously presented with an authentication page, even if the correct username and password are entered each time. To work around this, set the **uauth timeout** to 1 second with the **timeout uauth 0:0:1** command. However, this workaround opens a 1-second window of opportunity that might allow unauthenticated users to go through the firewall if they are coming from the same source IP address.

Because HTTPS authentication occurs on the SSL port 443, users must not configure an **access-list** command statement to block traffic from the HTTP client to the HTTP server on port 443. Furthermore, if static PAT is configured for web traffic on port 80, it must also be configured for the SSL port.

  – In the following example, the first set of commands configures static PAT for web traffic, and the second set of commands must be added to support the HTTPS authentication configuration:

```
object network obj-10.130.16.10-01
  host 10.130.16.10
  nat (inside,outside) static 10.132.16.200 service tcp 80 80
object network obj-10.130.16.10-02
  host 10.130.16.10
```

```
nat (inside,outside) static 10.132.16.200 service tcp 443 443
```

Authenticating Directly with the ASA

If you do not want to allow HTTP, HTTPS, Telnet, or FTP through the ASA but want to authenticate other types of traffic, you can authenticate with the ASA directly using HTTP, HTTPS, or Telnet.

This section includes the following topics:

- Authenticating HTTP(S) Connections with a Virtual Server, page 7-11
- Authenticating Telnet Connections with a Virtual Server, page 7-12

Authenticating HTTP(S) Connections with a Virtual Server

If you enabled the redirection method of HTTP and HTTPS authentication in the "Configuring Network Access Authentication" section on page 7-7, then you have also automatically enabled direct authentication.

When you use HTTP authentication on the ASA (see the"Configuring Network Access Authentication" section on page 7-7), the ASA uses basic HTTP authentication by default.

To continue to use basic HTTP authentication, and to enable direct authentication for HTTP and HTTPS, enter the following command:

Command	Purpose
aaa authentication listener http[**s**] *interface_name* [**port** *portnum*] **redirect** **Example:** `ciscoasa(config)# aaa authentication listener http inside redirect`	(Optional) Enables the redirection method of authentication for HTTP or HTTPS connections. The *interface_name* argument is the interface on which you want to enable listening ports. The **port** *portnum* argument specifies the port number on which the ASA listens; the defaults are 80 (HTTP) and 443 (HTTPS). You can use any port number and retain the same functionality, but be sure your direct authentication users know the port number; redirected traffic is sent to the correct port number automatically, but direct authenticators must specify the port number manually. Enter this command separately for HTTP and for HTTPS.

If the destination HTTP server requires authentication in addition to the ASA, then to authenticate separately with the ASA (via a AAA server) and with the HTTP server, enter the following command:

Command	Purpose

`virtual http`
Example:
`ciscoasa(config)# virtual http`
	Redirects all HTTP connections that require AAA authentication to the virtual HTTP server on the ASA. The ASA prompts for the AAA server username and password. After the AAA server authenticates the user, the ASA redirects the HTTP connection back to the original server, but it does not include the AAA server username and password. Because the username and password are not included in the HTTP packet, the HTTP server prompts the user separately for the HTTP server username and password.
	For inbound users (from lower security to higher security), you must also include the virtual HTTP address as a destination interface in the ACL applied to the source interface. In addition, you must add a static NAT command for the virtual HTTP IP address, even if NAT is not required. An identity NAT command is typically used (where you translate the address to itself).
	For outbound users, there is an explicit permit for traffic, but if you apply an ACL to an inside interface, be sure to allow access to the virtual HTTP address. A static statement is not required.
	Note Do not set the **timeout uauth** command duration to 0 seconds when using the **virtual http** command, because this setting prevents HTTP connections to the actual web server.
	You can authenticate directly with the ASA at the following URLs when you enable AAA for the interface:
	`http://`*`interface_ip`*`[`**`:`***`port`*`]`**`/netaccess/connstatus.html`**
	`https://`*`interface_ip`*`[`**`:`***`port`*`]`**`/netaccess/connstatus.html`**
	Without virtual HTTP, the same username and password that you used to authenticate with the ASA are sent to the HTTP server; you are not prompted separately for the HTTP server username and password. Assuming the username and password are not the same for the AAA and HTTP servers, then the HTTP authentication fails.

Authenticating Telnet Connections with a Virtual Server

Although you can configure network access authentication for any protocol or service (see the **aaa authentication match** or **aaa authentication include** command), you can authenticate directly with HTTP, Telnet, or FTP only. A user must first authenticate with one of these services before other traffic that requires authentication is allowed through. If you do not want to allow HTTP, Telnet, or FTP traffic through the ASA, but want to authenticate other types of traffic, you can configure virtual Telnet; the user Telnets to a given IP address configured on the ASA, and the ASA issues a Telnet prompt.

To configure a virtual Telnet server, enter the following command:

Command	Purpose
virtual telnet *ip_address* **Example:** `ciscoasa(config)# virtual telnet 209.165.202.129`	Configures a virtual Telnet server. The *ip_address* argument sets the IP address for the virtual Telnet server. Make sure this address is an unused address that is routed to the ASA. You must configure authentication for Telnet access to the virtual Telnet address as well as the other services that you want to authenticate using the **authentication match** or **aaa authentication include** command. When an unauthenticated user connects to the virtual Telnet IP address, the user is challenged for a username and password, and then authenticated by the AAA server. Once authenticated, the user sees the message "Authentication Successful." Then, the user can successfully access other services that require authentication. For inbound users (from lower security to higher security), you must also include the virtual Telnet address as a destination interface in the ACL applied to the source interface. In addition, you must add a static NAT command for the virtual Telnet IP address, even if NAT is not required. An identity NAT command is typically used (where you translate the address to itself). For outbound users, there is an explicit permit for traffic, but if you apply an ACL to an inside interface, be sure to allow access to the virtual Telnet address. A static statement is not required. To log out from the ASA, reconnect to the virtual Telnet IP address; you are then prompted to log out.

Examples

The following example shows how to enable virtual Telnet together with AAA authentication for other services:

```
ciscoasa(config)# virtual telnet 209.165.202.129
ciscoasa(config)# access-list ACL-IN extended permit tcp any host 209.165.200.225 eq smtp
ciscoasa(config)# access-list ACL-IN remark This is the SMTP server on the inside
ciscoasa(config)# access-list ACL-IN extended permit tcp any host 209.165.202.129 eq
telnet
ciscoasa(config)# access-list ACL-IN remark This is the virtual Telnet address
ciscoasa(config)# access-group ACL-IN in interface outside
ciscoasa(config)# network object obj-209.165.202.129-01
ciscoasa(config-network-object)# host 209.165.202.129
ciscoasa(config-network-object)# nat (inside,outside) static 209.165.202.129
ciscoasa(config)# access-list AUTH extended permit tcp any host 209.165.200.225 eq smtp
ciscoasa(config)# access-list AUTH remark This is the SMTP server on the inside
ciscoasa(config)# access-list AUTH extended permit tcp any host 209.165.202.129 eq telnet
ciscoasa(config)# access-list AUTH remark This is the virtual Telnet address
ciscoasa(config)# aaa authentication match AUTH outside tacacs+
```

Configuring Authorization for Network Access

After a user authenticates for a given connection, the ASA can use authorization to further control traffic from the user.

This section includes the following topics:

- Configuring TACACS+ Authorization, page 7-14
- Configuring RADIUS Authorization, page 7-17

Configuring TACACS+ Authorization

You can configure the ASA to perform network access authorization with TACACS+. You identify the traffic to be authorized by specifying ACLs that authorization rules must match. Alternatively, you can identify the traffic directly in authorization rules themselves.

Tip Using ACLs to identify traffic to be authorized can greatly reduced the number of authorization commands that you must enter. This is because each authorization rule that you enter can specify only one source and destination subnet and service, whereas an ACL can include many entries.

Authentication and authorization statements are independent; however, any unauthenticated traffic matched by an authorization rule will be denied. For authorization to succeed:

1. A user must first authenticate with the ASA.

   Because a user at a given IP address only needs to authenticate one time for all rules and types, if the authentication session has not expired, authorization can occur even if the traffic is not matched by an authentication rule.

2. After a user authenticates, the ASA checks the authorization rules for matching traffic.
3. If the traffic matches the authorization rule, the ASA sends the username to the TACACS+ server.
4. The TACACS+ server responds to the ASA with a permit or a deny for that traffic, based on the user profile.
5. The ASA enforces the authorization rule in the response.

See the documentation for your TACACS+ server for information about configuring network access authorizations for a user.

To configure TACACS+ authorization, perform the following steps:

	Command	Purpose
Step 1	**aaa-server** **Example:** `ciscoasa(config)# aaa-server AuthOutbound protocol tacacs+`	Identifies your AAA servers. If you have already identified them, continue to the next step.
Step 2	**access-list** **Example:** `ciscoasa(config)# access-list MAIL_AUTH extended permit tcp any any eq smtp`	Creates an ACL that identifies the source addresses and destination addresses of traffic you want to authenticate. For details, see the general operations configuration guide. The **permit** ACEs mark matching traffic for authentication, while **deny** entries exclude matching traffic from authentication. Be sure to include the destination ports for either HTTP, HTTPS, Telnet, or FTP in the ACL, because the user must authenticate with one of these services before other services are allowed through the ASA.
Step 3	**aaa authentication match** *acl_name interface_name server_group* **Example:** `ciscoasa(config)# aaa authentication match MAIL_AUTH inside AuthOutbound`	Configures authentication. The *acl_name* argument is the name of the ACL that you created in Step 2., The *interface_name* argument is the name of the interface specified with the **nameif** command, and the *server_group* argument is the AAA server group that you created in Step 1. **Note** You can alternatively use the **aaa authentication include** command (which identifies traffic within the command). However, you cannot use both methods in the same configuration. See the command reference for more information.
Step 4	**aaa authentication listener http**[**s**] *interface_name* [**port** *portnum*] **redirect** **Example:** `ciscoasa(config)# aaa authentication listener http inside redirect`	(Optional) Enables the redirection method of authentication for HTTP or HTTPS connections. The *interface_name* argument is the interface on which you want to enable listening ports. The **port** *portnum* argument specifies the port number on which the ASA listens; the defaults are 80 (HTTP) and 443 (HTTPS). You can use any port number and retain the same functionality, but be sure your direct authentication users know the port number; redirected traffic is sent to the correct port number automatically, but direct authenticators must specify the port number manually. Enter this command separately for HTTP and for HTTPS.

	Command	Purpose
Step 5	**aaa local authentication attempts max-fail** *number* **Example:** `ciscoasa(config)# aaa local authentication attempts max-fail 7`	(Optional) Uses the local database for network access authentication and limits the number of consecutive failed login attempts that the ASA allows any given user account (with the exception of users with a privilege level of 15. This feature does not affect level 15 users). The *number* argument value is between 1 and 16. **Tip** To clear the lockout status of a specific user or all users, use the **clear aaa local user lockout** command.
Step 6	**access-list** **Example:** `ciscoasa(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet`	Create an ACL that identifies the source addresses and destination addresses of traffic that you want to authorize. For instructions, see the general operations configuration guide. The **permit** ACEs mark matching traffic for authorization, while **deny** entries exclude matching traffic from authorization. The ACL that you use for authorization matching should include rules that are equal to or a subset of the rules in the ACL used for authentication matching. **Note** If you have configured authentication and want to authorize all the traffic being authenticated, you can use the same ACL that you created for use with the **aaa authentication match** command.
Step 7	**aaa authorization match** *acl_name interface_name server_group* **Example:** `ciscoasa(config)# aaa authentication match TELNET_AUTH inside AuthOutbound`	Enables authorization. The *acl_name* argument is the name of the ACL you created in Step 6, the *interface_name* argument is the name of the interface as specified with the **nameif** command or by default, and the *server_group* argument is the AAA server group that you created when you enabled authentication. **Note** Alternatively, you can use the **aaa authorization include** command (which identifies traffic within the command) but you cannot use both methods in the same configuration. See the command reference for more information.

Examples

The following example authenticates and authorizes inside Telnet traffic. Telnet traffic to servers other than 209.165.201.5 can be authenticated alone, but traffic to 209.165.201.5 requires authorization.

```
ciscoasa(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
ciscoasa(config)# access-list SERVER_AUTH extended permit tcp any host 209.165.201.5 eq
telnet
ciscoasa(config)# aaa-server AuthOutbound protocol tacacs+
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
ciscoasa(config-aaa-server-host)# key TACPlusUauthKey
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
ciscoasa(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
```

Configuring RADIUS Authorization

When authentication succeeds, the RADIUS protocol returns user authorizations in the access-accept message sent by a RADIUS server. For more information about configuring authentication, see the "Configuring Network Access Authentication" section on page 7-7.

When you configure the ASA to authenticate users for network access, you are also implicitly enabling RADIUS authorizations; therefore, this section contains no information about configuring RADIUS authorization on the ASA. It does provide information about how the ASA handles ACL information received from RADIUS servers.

You can configure a RADIUS server to download an ACL to the ASA or an ACL name at the time of authentication. The user is authorized to do only what is permitted in the user-specific ACL.

Note If you have used the **access-group** command to apply ACLs to interfaces, be aware of the following effects of the **per-user-override** keyword on authorization by user-specific ACLs:

- Without the **per-user-override** keyword, traffic for a user session must be permitted by both the interface ACL and the user-specific ACL.
- With the **per-user-override** keyword, the user-specific ACL determines what is permitted.

For more information, see the **access-group** command entry in the command reference.

This section includes the following topics:

- Configuring a RADIUS Server to Send Downloadable Access Control Lists, page 7-17
- Configuring a RADIUS Server to Download Per-User Access Control List Names, page 7-21

Configuring a RADIUS Server to Send Downloadable Access Control Lists

This section describes how to configure Cisco Secure ACS or a third-party RADIUS server and includes the following topics:

- About the Downloadable ACL Feature and Cisco Secure ACS, page 7-17
- Configuring Cisco Secure ACS for Downloadable ACLs, page 7-19
- Configuring Any RADIUS Server for Downloadable ACLs, page 7-20
- Converting Wildcard Netmask Expressions in Downloadable ACLs, page 7-21

About the Downloadable ACL Feature and Cisco Secure ACS

Downloadable ACLs is the most scalable means of using Cisco Secure ACS to provide the appropriate ACLs for each user. It provides the following capabilities:

- Unlimited ACL size—Downloadable ACLs are sent using as many RADIUS packets as required to transport the full ACL from Cisco Secure ACS to the ASA.

- Simplified and centralized management of ACLs—Downloadable ACLs enable you to write a set of ACLs once and apply it to many user or group profiles and distribute it to many ASAs.

This approach is most useful when you have very large ACL sets that you want to apply to more than one Cisco Secure ACS user or group; however, its ability to simplify Cisco Secure ACS user and group management makes it useful for ACLs of any size.

The ASA receives downloadable ACLs from Cisco Secure ACS using the following process:

1. The ASA sends a RADIUS authentication request packet for the user session.
2. If Cisco Secure ACS successfully authenticates the user, Cisco Secure ACS returns a RADIUS access-accept message that includes the internal name of the applicable downloadable ACL. The Cisco IOS cisco-av-pair RADIUS VSA (vendor 9, attribute 1) includes the following attribute-value pair to identify the downloadable ACL set:

   ```
   ACS:CiscoSecure-Defined-ACL=acl-set-name
   ```

   where *acl-set-name* is the internal name of the downloadable ACL, which is a combination of the name assigned to the ACL by the Cisco Secure ACS administrator and the date and time that the ACL was last modified.
3. The ASA examines the name of the downloadable ACL and determines if it has previously received the named downloadable ACL.
   - If the ASA has previously received the named downloadable ACL, communication with Cisco Secure ACS is complete and the ASA applies the ACL to the user session. Because the name of the downloadable ACL includes the date and time that it was last modified, matching the name sent by Cisco Secure ACS to the name of an ACL previously downloaded means that the ASA has the most recent version of the downloadable ACL.
   - If the ASA has not previously received the named downloadable ACL, it may have an out-of-date version of the ACL or it may not have downloaded any version of the ACL. In either case, the ASA issues a RADIUS authentication request using the downloadable ACL name as the username in the RADIUS request and a null password attribute. In a cisco-av-pair RADIUS VSA, the request also includes the following attribute-value pairs:

     ```
     AAA:service=ip-admission
     AAA:event=acl-download
     ```

     In addition, the ASA signs the request with the Message-Authenticator attribute (IETF RADIUS attribute 80).
4. After receipt of a RADIUS authentication request that has a username attribute that includes the name of a downloadable ACL, Cisco Secure ACS authenticates the request by checking the Message-Authenticator attribute. If the Message-Authenticator attribute is missing or incorrect, Cisco Secure ACS ignores the request. The presence of the Message-Authenticator attribute prevents malicious use of a downloadable ACL name to gain unauthorized network access. The Message-Authenticator attribute and its use are defined in RFC 2869, RADIUS Extensions, available at http://www.ietf.org.
5. If the ACL required is less than approximately 4 KB in length, Cisco Secure ACS responds with an access-accept message that includes the ACL. The largest ACL that can fit in a single access-accept message is slightly less than 4 KB, because part of the message must be other required attributes.

   Cisco Secure ACS sends the downloadable ACL in a cisco-av-pair RADIUS VSA. The ACL is formatted as a series of attribute-value pairs that each include an ACE and are numbered serially:

   ```
   ip:inacl#1=ACE-1
   ip:inacl#2=ACE-2
   .
   .
   ```

```
.
ip:inacl#n=ACE-n

ip:inacl#1=permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
```

6. If the ACL required is more than approximately 4 KB in length, Cisco Secure ACS responds with an access-challenge message that includes a portion of the ACL, formatted as described previously, and a State attribute (IETF RADIUS attribute 24), which includes control data used by Cisco Secure ACS to track the progress of the download. Cisco Secure ACS fits as many complete attribute-value pairs into the cisco-av-pair RADIUS VSA as it can without exceeding the maximum RADIUS message size.

   The ASA stores the portion of the ACL received and responds with another access-request message that includes the same attributes as the first request for the downloadable ACL, plus a copy of the State attribute received in the access-challenge message.

   This process repeats until Cisco Secure ACS sends the last of the ACL in an access-accept message.

Configuring Cisco Secure ACS for Downloadable ACLs

You can configure downloadable ACLs on Cisco Secure ACS as a shared profile component and then assign the ACL to a group or to an individual user.

The ACL definition consists of one or more ASA commands that are similar to the extended **access-list** command (see command reference), except without the following prefix:

```
access-list acl_name extended
```

The following example is a downloadable ACL definition on Cisco Secure ACS version 3.3:

```
+--------------------------------------------+
| Shared profile Components                  |
|                                            |
|      Downloadable IP ACLs Content          |
|                                            |
| Name:    acs_ten_acl                       |
|                                            |
|      ACL Definitions                       |
|                                            |
| permit tcp any host 10.0.0.254             |
| permit udp any host 10.0.0.254             |
| permit icmp any host 10.0.0.254            |
| permit tcp any host 10.0.0.253             |
| permit udp any host 10.0.0.253             |
| permit icmp any host 10.0.0.253            |
| permit tcp any host 10.0.0.252             |
| permit udp any host 10.0.0.252             |
| permit icmp any host 10.0.0.252            |
| permit ip any any                          |
+--------------------------------------------+
```

For more information about creating downloadable ACLs and associating them with users, see the user guide for your version of Cisco Secure ACS.

On the ASA, the downloaded ACL has the following name:

```
#ACSACL#-ip-acl_name-number
```

The *acl_name* argument is the name that is defined on Cisco Secure ACS (*acs_ten_acl* in the preceding example), and *number* is a unique version ID generated by Cisco Secure ACS.

The downloaded ACL on the ASA consists of the following lines:

```
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.254
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.254
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.254
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.253
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.253
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.253
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit tcp any host 10.0.0.252
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit udp any host 10.0.0.252
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit icmp any host 10.0.0.252
access-list #ACSACL#-ip-asa-acs_ten_acl-3b5385f7 permit ip any any
```

Configuring Any RADIUS Server for Downloadable ACLs

You can configure any RADIUS server that supports Cisco IOS RADIUS VSAs to send user-specific ACLs to the ASA in a Cisco IOS RADIUS cisco-av-pair VSA (vendor 9, attribute 1).

In the cisco-av-pair VSA, configure one or more ACEs that are similar to the **access-list extended** command (see command reference), except that you replace the following command prefix:

```
access-list acl_name extended
```

with the following text:

```
ip:inacl#nnn=
```

The *nnn* argument is a number in the range from 0 to 999999999 that identifies the order of the command statement to be configured on the ASA. If this parameter is omitted, the sequence value is 0, and the order of the ACEs inside the cisco-av-pair RADIUS VSA is used.

The following example is an ACL definition as it should be configured for a cisco-av-pair VSA on a RADIUS server:

```
ip:inacl#1=permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#99=deny tcp any any
ip:inacl#2=permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
ip:inacl#100=deny udp any any
ip:inacl#3=permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
```

For information about making unique per user the ACLs that are sent in the cisco-av-pair attribute, see the documentation for your RADIUS server.

On the ASA, the downloaded ACL name has the following format:

```
AAA-user-username
```

The *username* argument is the name of the user that is being authenticated.

The downloaded ACL on the ASA consists of the following lines. Notice the order based on the numbers identified on the RADIUS server.

```
access-list  AAA-user-bcham34-79AD4A08 permit tcp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-bcham34-79AD4A08 permit udp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-bcham34-79AD4A08 permit icmp 10.1.0.0 255.0.0.0 10.0.0.0 255.0.0.0
access-list  AAA-user-bcham34-79AD4A08 deny tcp any any
access-list  AAA-user-bcham34-79AD4A08 deny udp any any
```

Downloaded ACLs have two spaces between the word “access-list” and the name. These spaces serve to differentiate a downloaded ACL from a local ACL. In this example, “79AD4A08” is a hash value generated by the ASA to help determine when ACL definitions have changed on the RADIUS server.

Converting Wildcard Netmask Expressions in Downloadable ACLs

If a RADIUS server provides downloadable ACLs to Cisco VPN 3000 series concentrators as well as to the ASA, you may need the ASA to convert wildcard netmask expressions to standard netmask expressions. This is because Cisco VPN 3000 series concentrators support wildcard netmask expressions, but the ASA only supports standard netmask expressions. Configuring the ASA to convert wildcard netmask expressions helps minimize the effects of these differences on how you configure downloadable ACLs on your RADIUS servers. Translation of wildcard netmask expressions means that downloadable ACLs written for Cisco VPN 3000 series concentrators can be used by the ASA without altering the configuration of the downloadable ACLs on the RADIUS server.

You configure ACL netmask conversion on a per-server basis using the **acl-netmask-convert** command, available in the aaa-server configuration mode. For more information about configuring a RADIUS server, see the general operations configuration guide. For more information about the **acl-netmask-convert** command, see the command reference

Configuring a RADIUS Server to Download Per-User Access Control List Names

To download a name for an ACL that you already created on the ASA from the RADIUS server when a user authenticates, configure the IETF RADIUS filter-id attribute (attribute number 11) as follows:

```
filter-id=acl_name
```

Note In Cisco Secure ACS, the values for filter-id attributes are specified in boxes in the HTML interface, omitting **filter-id=** and entering only *acl_name*.

For information about making the filter-id attribute value unique per user, see the documentation for your RADIUS server.

To create an ACL on the ASA, see the general operations configuration guide.

Configuring Accounting for Network Access

The ASA can send accounting information to a RADIUS or TACACS+ server about any TCP or UDP traffic that passes through the ASA. If that traffic is also authenticated, then the AAA server can maintain accounting information by username. If the traffic is not authenticated, the AAA server can maintain accounting information by IP address. Accounting information includes session start and stop times, username, the number of bytes that pass through the ASA for the session, the service used, and the duration of each session.

To configure accounting, perform the following steps:

	Command	Purpose

Step 1	**`access-list`**
	Example:
	`ciscoasa(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet`
		If you want the ASA to provide accounting data per user, you must enable authentication. For more information, see the "Configuring Network Access Authentication" section on page 7-7. If you want the ASA to provide accounting data per IP address, enabling authentication is not necessary.
		Creates an ACL that identifies the source addresses and destination addresses of traffic for which you want accounting data. For instructions, see the general operations configuration guide.
		The permit ACEs mark matching traffic for accounting, while deny entries exclude matching traffic from accounting.
		Note If you have configured authentication and want accounting data for all the traffic being authenticated, you can use the same ACL that you created for use with the **aaa authentication match** command.

Step 2	**`aaa accounting match`** *`acl_name interface_name server_group`*
	Example:
	`ciscoasa(config)# aaa accounting match SERVER_AUTH inside AuthOutbound`
		Enables accounting.
		The *acl_name* argument is the ACL name set in the **access-list** command.
		The *interface_name* argument is the interface name set in the **nameif** command.
		The *server_group* argument is the server group name set in the **aaa-server** command.
		Note Alternatively, you can use the **aaa accounting include** command (which identifies traffic within the command), but you cannot use both methods in the same configuration. See the command reference for more information.

Examples

The following example authenticates, authorizes, and accounts for inside Telnet traffic. Telnet traffic to servers other than 209.165.201.5 can be authenticated alone, but traffic to 209.165.201.5 requires authorization and accounting.

```
ciscoasa(config)# aaa-server AuthOutbound protocol tacacs+
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server AuthOutbound (inside) host 10.1.1.1
ciscoasa(config-aaa-server-host)# key TACPlusUauthKey
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# access-list TELNET_AUTH extended permit tcp any any eq telnet
ciscoasa(config)# access-list SERVER_AUTH extended permit tcp any host 209.165.201.5 eq
telnet
ciscoasa(config)# aaa authentication match TELNET_AUTH inside AuthOutbound
ciscoasa(config)# aaa authorization match SERVER_AUTH inside AuthOutbound
ciscoasa(config)# aaa accounting match SERVER_AUTH inside AuthOutbound
```

AAA provides an extra level of protection and control for user access than using ACLs alone. For example, you can create an ACL allowing all outside users to access Telnet on a server on the DMZ network. If you want only some users to access the server and you might not always know IP addresses of these users, you can enable AAA to allow only authenticated and/or authorized users to connect through the ASA. (The Telnet server enforces authentication, too; the ASA prevents unauthorized users from attempting to access the server.)

Using MAC Addresses to Exempt Traffic from Authentication and Authorization

The ASA can exempt from authentication and authorization any traffic from specific MAC addresses. For example, if the ASA authenticates TCP traffic originating on a particular network, but you want to allow unauthenticated TCP connections from a specific server, you would use a MAC exempt rule to exempt from authentication and authorization any traffic from the server specified by the rule.

This feature is particularly useful to exempt devices such as IP phones that cannot respond to authentication prompts.

To use MAC addresses to exempt traffic from authentication and authorization, perform the following steps:

	Command	Purpose
Step 1	**mac-list** *id* {**deny** | **permit**} *mac macmask* **Example:** `ciscoasa(config)# mac-list abc permit 00a0.c95d.0282 ffff.ffff.ffff`	Configures a MAC list. The *id* argument is the hexadecimal number that you assign to the MAC list. To group a set of MAC addresses, enter the **mac-list** command as many times as needed with the same ID value. Because you can only use one MAC list for AAA exemption, be sure that your MAC list includes all the MAC addresses that you want to exempt. You can create multiple MAC lists, but you can only use one at a time. The order of entries matters, because the packet uses the first entry it matches, instead of a best match scenario. If you have a **permit** entry, and you want to deny an address that is allowed by the **permit** entry, be sure to enter the **deny** entry before the **permit** entry. The *mac* argument specifies the source MAC address in 12-digit hexadecimal form; that is, nnnn.nnnn.nnnn. The *macmask* argument specifies the portion of the MAC address that should be used for matching. For example, ffff.ffff.ffff matches the MAC address exactly. ffff.ffff.0000 matches only the first 8 digits.
Step 2	**aaa mac-exempt match** *id* **Example:** `ciscoasa(config)# aaa mac-exempt match 1`	Exempts traffic for the MAC addresses specified in a particular MAC list. The *id* argument is the string identifying the MAC list that includes the MAC addresses whose traffic is to be exempt from authentication and authorization. You can only enter one instance of the **aaa mac-exempt match** command.

Examples

The following example bypasses authentication for a single MAC address:

```
ciscoasa(config)# mac-list abc permit 00a0.c95d.0282 ffff.ffff.ffff
ciscoasa(config)# aaa mac-exempt match abc
```

The following example bypasses authentication for all Cisco IP Phones, which have the hardware ID 0003.E3:

```
ciscoasa(config)# mac-list acd permit 0003.E300.0000 FFFF.FF00.0000
ciscoasa(config)# aaa mac-exempt match acd
```

The following example bypasses authentication for a a group of MAC addresses except for 00a0.c95d.02b2. Enter the **deny** statement before the **permit** statement, because 00a0.c95d.02b2 matches the **permit** statement as well, and if it is first, the **deny** statement will never be matched.

```
ciscoasa(config)# mac-list 1 deny 00a0.c95d.0282 ffff.ffff.ffff
ciscoasa(config)# mac-list 1 permit 00a0.c95d.0000 ffff.ffff.0000
ciscoasa(config)# aaa mac-exempt match 1
```

Feature History for AAA Rules

Table 7-1 lists each feature change and the platform release in which it was implemented.

Table 7-1 Feature History for AAA Rules

Feature Name	Platform Releases	Feature Information
AAA Rules	7.0(1)	AAA Rules describe how to enable AAA for network access. We introduced the following commands: **aaa authentication match, aaa authentication include | exclude, aaa authentication listener http[s]**, **aaa local authentication attempts max-fail**, **virtual http**, **virtual telnet**, **aaa authentication secure-http-client, aaa authorization match**, **aaa accounting match**, **aaa mac-exempt match**.
Authentication using Cut-Through Proxy	9.0(1)	You can authenticate using AAA rules in conjunction with the Identity Firewall feature. We modified the following command: **aaa authentication match**.

PART 4

Configuring Application Inspection

CHAPTER 9

Getting Started with Application Layer Protocol Inspection

This chapter describes how to configure application layer protocol inspection. Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the ASA to do a deep packet inspection instead of passing the packet through the fast path (see the general operations configuration guide for more information about the fast path). As a result, inspection engines can affect overall throughput. Several common inspection engines are enabled on the ASA by default, but you might need to enable others depending on your network.

This chapter includes the following sections:

Information about Application Layer Protocol Inspection

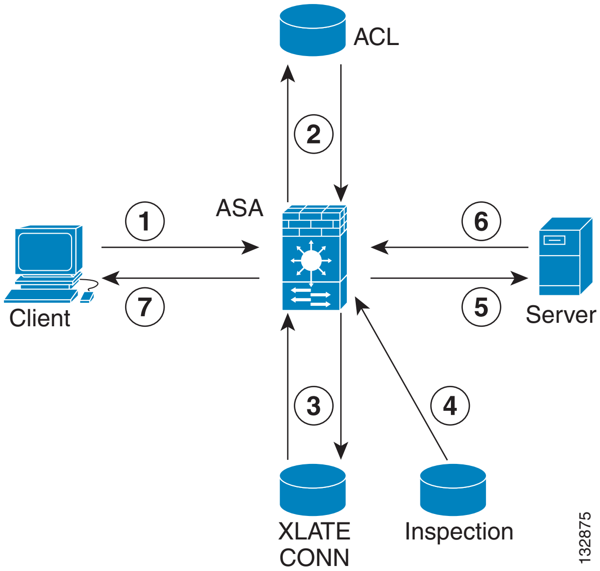
This section includes the following topics:

How Inspection Engines Work

As illustrated in Figure 9-1, the ASA uses three databases for its basic operation:

- ACLs—Used for authentication and authorization of connections based on specific networks, hosts, and services (TCP/UDP port numbers).
- Inspections—Contains a static, predefined set of application-level inspection functions.
- Connections (XLATE and CONN tables)—Maintains state and other information about each established connection. This information is used by the Adaptive Security Algorithm and cut-through proxy to efficiently forward traffic within established sessions.

Figure 9-1 How Inspection Engines Work



In Figure 9-1, operations are numbered in the order they occur, and are described as follows:

1. A TCP SYN packet arrives at the ASA to establish a new connection.
2. The ASA checks the ACL database to determine if the connection is permitted.
3. The ASA creates a new entry in the connection database (XLATE and CONN tables).
4. The ASA checks the Inspections database to determine if the connection requires application-level inspection.
5. After the application inspection engine completes any required operations for the packet, the ASA forwards the packet to the destination system.
6. The destination system responds to the initial request.
7. The ASA receives the reply packet, looks up the connection in the connection database, and forwards the packet because it belongs to an established session.

The default configuration of the ASA includes a set of application inspection entries that associate supported protocols with specific TCP or UDP port numbers and that identify any special handling required.

When to Use Application Protocol Inspection

When a user establishes a connection, the ASA checks the packet against ACLs, creates an address translation, and creates an entry for the session in the fast path, so that further packets can bypass time-consuming checks. However, the fast path relies on predictable port numbers and does not perform address translations inside a packet.

Many protocols open secondary TCP or UDP ports. The initial session on a well-known port is used to negotiate dynamically assigned port numbers.

Other applications embed an IP address in the packet that needs to match the source address that is normally translated when it goes through the ASA.

If you use applications like these, then you need to enable application inspection.

When you enable application inspection for a service that embeds IP addresses, the ASA translates embedded addresses and updates any checksum or other fields that are affected by the translation.

When you enable application inspection for a service that uses dynamically assigned ports, the ASA monitors sessions to identify the dynamic port assignments, and permits data exchange on these ports for the duration of the specific session.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

Failover Guidelines

State information for multimedia sessions that require inspection are not passed over the state link for stateful failover. The exception is GTP, which is replicated over the state link.

IPv6 Guidelines

Supports IPv6 for the following inspections:

- DNS
- FTP
- HTTP
- ICMP
- SIP
- SMTP
- IPsec pass-through
- IPv6

Supports NAT64 for the following inspections:

- DNS
- FTP
- HTTP
- ICMP

Additional Guidelines and Limitations

Some inspection engines do not support PAT, NAT, outside NAT, or NAT between same security interfaces. See "Default Settings and NAT Limitations" for more information about NAT support.

For all the application inspections, the ASA limits the number of simultaneous, active data connections to 200 connections. For example, if an FTP client opens multiple secondary connections, the FTP inspection engine allows only 200 active connections and the 201 connection is dropped and the adaptive security appliance generates a system error message.

Inspected protocols are subject to advanced TCP-state tracking, and the TCP state of these connections is not automatically replicated. While these connections are replicated to the standby unit, there is a best-effort attempt to re-establish a TCP state.

Default Settings and NAT Limitations

By default, the configuration includes a policy that matches all default application inspection traffic and applies inspection to the traffic on all interfaces (a global policy). Default application inspection traffic includes traffic to the default ports for each protocol. You can only apply one global policy, so if you want to alter the global policy, for example, to apply inspection to non-standard ports, or to add inspections that are not enabled by default, you need to either edit the default policy or disable it and apply a new one.

Table 9-1 lists all inspections supported, the default ports used in the default class map, and the inspection engines that are on by default, shown in bold. This table also notes any NAT limitations.

Table 9-1 Supported Application Inspection Engines

Application[1]	Default Port	NAT Limitations	Standards[2]	Comments
CTIQBE	TCP/2748	No extended PAT. No NAT64. (Clustering) No static PAT.	—	—
DCERPC	TCP/135	No NAT64.	—	—
DNS over UDP	UDP/53	No NAT support is available for name resolution through WINS.	RFC 1123	—
FTP	TCP/21	(Clustering) No static PAT.	RFC 959	—
GTP	UDP/3386 UDP/2123	No extended PAT. No NAT64.	—	Requires a special license.
H.323 H.225 and **RAS**	TCP/1720 UDP/1718 UDP (RAS) 1718-1719	No dynamic NAT or PAT. Static PAT may not work. (Clustering) No static PAT. No extended PAT. No per-session PAT. No NAT on same security interfaces. No outside NAT. No NAT64.	ITU-T H.323, H.245, H225.0, Q.931, Q.932	—
HTTP	TCP/80	—	RFC 2616	Beware of MTU limitations stripping ActiveX and Java. If the MTU is too small to allow the Java or ActiveX tag to be included in one packet, stripping may not occur.
ICMP	—	—	—	—

Table 9-1 Supported Application Inspection Engines (continued)

Application[1]	Default Port	NAT Limitations	Standards[2]	Comments
ICMP ERROR	—	—	—	—
ILS (LDAP)	TCP/389	No extended PAT. No NAT64.	—	—
Instant Messaging (IM)	Varies by client	No extended PAT. No NAT64.	RFC 3860	—
IP Options	—	No NAT64.	RFC 791, RFC 2113	—
IPsec Pass Through	UDP/500	No PAT. No NAT64.	—	—
IPv6	—	No NAT64.	RFC 2460	—
MGCP	UDP/2427, 2727	No extended PAT. No NAT64. (Clustering) No static PAT.	RFC 2705bis-05	—
MMP	TCP 5443	No extended PAT. No NAT64.	—	—
NetBIOS Name Server over IP	UDP/137, 138 (Source ports)	No extended PAT. No NAT64.	—	NetBIOS is supported by performing NAT of the packets for NBNS UDP port 137 and NBDS UDP port 138.
PPTP	TCP/1723	No NAT64. (Clustering) No static PAT.	RFC 2637	—
RADIUS Accounting	1646	No NAT64.	RFC 2865	—
RSH	TCP/514	No PAT. No NAT64. (Clustering) No static PAT.	Berkeley UNIX	—
RTSP	TCP/554	No extended PAT. No outside NAT. No NAT64. (Clustering) No static PAT.	RFC 2326, 2327, 1889	No handling for HTTP cloaking.
ScanSafe (Cloud Web Security)	TCP/80 TCP/413	—	—	These ports are not included in the default-inspection-traffic class for the ScanSafe inspection.

Table 9-1 Supported Application Inspection Engines (continued)

Application[1]	Default Port	NAT Limitations	Standards[2]	Comments
SIP	TCP/5060 UDP/5060	No outside NAT. No NAT on same security interfaces. No extended PAT. No per-session PAT. No NAT64. (Clustering) No static PAT.	RFC 2543	—
SKINNY (SCCP)	TCP/2000	No outside NAT. No NAT on same security interfaces. No extended PAT. No per-session PAT. No NAT64. (Clustering) No static PAT.	—	Does not handle TFTP uploaded Cisco IP Phone configurations under certain circumstances.
SMTP and **ESMTP**	TCP/25	No NAT64.	RFC 821, 1123	—
SNMP	UDP/161, 162	No NAT or PAT.	RFC 1155, 1157, 1212, 1213, 1215	v.2 RFC 1902-1908; v.3 RFC 2570-2580.
SQL*Net	TCP/1521	No extended PAT. No NAT64. (Clustering) No static PAT.	—	v.1 and v.2.
Sun RPC over UDP and TCP	UDP/111	No extended PAT. No NAT64.	—	The default rule includes UDP port 111; if you want to enable Sun RPC inspection for TCP port 111, you need to create a new rule that matches TCP port 111 and performs Sun RPC inspection.
TFTP	UDP/69	No NAT64. (Clustering) No static PAT.	RFC 1350	Payload IP addresses are not translated.
WAAS	—	No extended PAT. No NAT64.	—	—
XDCMP	UDP/177	No extended PAT. No NAT64. (Clustering) No static PAT.	—	—

[1] Inspection engines that are enabled by default for the default port are in bold.

[2] The ASA is in compliance with these standards, but it does not enforce compliance on packets being inspected. For example, FTP commands are supposed to be in a particular order, but the ASA does not enforce the order.

The default policy configuration includes the following commands:

```
class-map inspection_default
 match default-inspection-traffic
policy-map type inspect dns preset_dns_map
 parameters
    message-length maximum client auto
    message-length maximum 512
    dns-guard
    protocol-enforcement
    nat-rewrite
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225 _default_h323_map
  inspect h323 ras _default_h323_map
  inspect ip-options _default_ip_options_map
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp _default_esmtp_map
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
```

Configuring Application Layer Protocol Inspection

This feature uses Modular Policy Framework to create a service policy. Service policies provide a consistent and flexible way to configure ASA features. For example, you can use a service policy to create a timeout configuration that is specific to a particular TCP application, as opposed to one that applies to all TCP applications. See Chapter 1, "Configuring a Service Policy Using the Modular Policy Framework," for more information. For some applications, you can perform special actions when you enable inspection. See Chapter 1, "Configuring a Service Policy Using the Modular Policy Framework," for more information.

Inspection is enabled by default for some applications. See the "Default Settings and NAT Limitations" section for more information. Use this section to modify your inspection policy.

Detailed Steps

Step 1 To identify the traffic to which you want to apply inspections, add either a Layer 3/4 class map for through traffic or a Layer 3/4 class map for management traffic. See the "Creating a Layer 3/4 Class Map for Through Traffic" section on page 1-12 and "Creating a Layer 3/4 Class Map for Management Traffic" section on page 1-14 for detailed information. The management Layer 3/4 class map can be used only with the RADIUS accounting inspection.

The default Layer 3/4 class map for through traffic is called "inspection_default." It matches traffic using a special **match** command, **match default-inspection-traffic**, to match the default ports for each application protocol. This traffic class (along with **match any**, which is not typically used for inspection) matches both IPv4 and IPv6 traffic for inspections that support IPv6. See the "Guidelines and Limitations" section on page 9-3 for a list of IPv6-enabled inspections.

You can specify a **match access-list** command along with the **match default-inspection-traffic** command to narrow the matched traffic to specific IP addresses. Because the **match default-inspection-traffic** command specifies the ports to match, any ports in the ACL are ignored.

Tip We suggest that you only inspect traffic on ports on which you expect application traffic; if you inspect all traffic, for example using **match any**, the ASA performance can be impacted.

If you want to match non-standard ports, then create a new class map for the non-standard ports. See the "Default Settings and NAT Limitations" section on page 9-4 for the standard ports for each inspection engine. You can combine multiple class maps in the same policy if desired, so you can create one class map to match certain traffic, and another to match different traffic. However, if traffic matches a class map that contains an inspection command, and then matches another class map that also has an inspection command, only the first matching class is used. For example, SNMP matches the inspection_default class. To enable SNMP inspection, enable SNMP inspection for the default class in Step 5. Do not add another class that matches SNMP.

For example, to limit inspection to traffic from 10.1.1.0 to 192.168.1.0 using the default class map, enter the following commands:

```
ciscoasa(config)# access-list inspect extended permit ip 10.1.1.0 255.255.255.0
192.168.1.0 255.255.255.0
ciscoasa(config)# class-map inspection_default
ciscoasa(config-cmap)# match access-list inspect
```

View the entire class map using the following command:

```
ciscoasa(config-cmap)# show running-config class-map inspection_default
!
class-map inspection_default
 match default-inspection-traffic
 match access-list inspect
!
```

To inspect FTP traffic on port 21 as well as 1056 (a non-standard port), create an ACL that specifies the ports, and assign it to a new class map:

```
ciscoasa(config)# access-list ftp_inspect extended permit tcp any any eq 21
ciscoasa(config)# access-list ftp_inspect extended permit tcp any any eq 1056
ciscoasa(config)# class-map new_inspection
ciscoasa(config-cmap)# match access-list ftp_inspect
```

Step 2 (Optional) Some inspection engines let you control additional parameters when you apply the inspection to the traffic. See the following sections to configure an inspection policy map for your application:

- DCERPC—See the "Configuring a DCERPC Inspection Policy Map for Additional Inspection Control" section on page 13-2
- DNS—See the "(Optional) Configuring a DNS Inspection Policy Map and Class Map" section on page 10-3
- ESMTP—See the "Configuring an ESMTP Inspection Policy Map for Additional Inspection Control" section on page 10-33
- FTP—See the "Configuring an FTP Inspection Policy Map for Additional Inspection Control" section on page 10-12.
- GTP—See the "Configuring a GTP Inspection Policy Map for Additional Inspection Control" section on page 13-4.

- H323—See the "Configuring an H.323 Inspection Policy Map for Additional Inspection Control" section on page 11-6
- HTTP—See the "Configuring an HTTP Inspection Policy Map for Additional Inspection Control" section on page 10-16.
- Instant Messaging—See the "Configuring an Instant Messaging Inspection Policy Map for Additional Inspection Control" section on page 10-21
- IP Options—See the "Configuring an IP Options Inspection Policy Map for Additional Inspection Control" section on page 10-25
- IPsec Pass Through—See the "IPsec Pass Through Inspection" section on page 10-25
- IPv6—See the "(Optional) Configuring an IPv6 Inspection Policy Map" section on page 10-27
- MGCP—See the "Configuring an MGCP Inspection Policy Map for Additional Inspection Control" section on page 11-12.
- NetBIOS—See the "Configuring a NetBIOS Inspection Policy Map for Additional Inspection Control" section on page 10-30
- RADIUS Accounting—See the "Configuring a RADIUS Inspection Policy Map for Additional Inspection Control" section on page 13-9
- RTSP—See the "Configuring an RTSP Inspection Policy Map for Additional Inspection Control" section on page 11-16
- ScanSafe (Cloud Web Security)—See the "Configuring a Service Policy to Send Traffic to Cloud Web Security" section on page 25-10
- SIP—See the "Configuring a SIP Inspection Policy Map for Additional Inspection Control" section on page 11-20
- Skinny—See the "Configuring a Skinny (SCCP) Inspection Policy Map for Additional Inspection Control" section on page 11-26
- SNMP—See the "Configuring an SNMP Inspection Policy Map for Additional Inspection Control" section on page 13-10.

Step 3 To add or edit a Layer 3/4 policy map that sets the actions to take with the class map traffic, enter the following command:

```
ciscoasa(config)# policy-map name
ciscoasa(config-pmap)#
```

The default policy map is called "global_policy." This policy map includes the default inspections listed in the "Default Settings and NAT Limitations" section on page 9-4. If you want to modify the default policy (for example, to add or delete an inspection, or to identify an additional class map for your actions), then enter **global_policy** as the name.

Step 4 To identify the class map from Step 1 to which you want to assign an action, enter the following command:

```
ciscoasa(config-pmap)# class class_map_name
ciscoasa(config-pmap-c)#
```

If you are editing the default policy map, it includes the inspection_default class map. You can edit the actions for this class by entering **inspection_default** as the name. To add an additional class map to this policy map, identify a different name. You can combine multiple class maps in the same policy if desired, so you can create one class map to match certain traffic, and another to match different traffic. However, if traffic matches a class map that contains an inspection command, and then matches another class map that also has an inspection command, only the first matching class is used. For example, SNMP matches the inspection_default class map.To enable SNMP inspection, enable SNMP inspection for the default class in Step 5. Do not add another class that matches SNMP.

Step 5 Enable application inspection by entering the following command:

```
ciscoasa(config-pmap-c)# inspect protocol
```

The *protocol* is one of the following values:

Table 9-2 Protocol Keywords

Keywords	Notes
ctiqbe	—
dcerpc [*map_name*]	If you added a DCERPC inspection policy map according to "Configuring a DCERPC Inspection Policy Map for Additional Inspection Control" section on page 13-2, identify the map name in this command.
dns [*map_name*] [**dynamic-filter-snoop**]	If you added a DNS inspection policy map according to "(Optional) Configuring a DNS Inspection Policy Map and Class Map" section on page 10-3, identify the map name in this command. The default DNS inspection policy map name is "preset_dns_map." The default inspection policy map sets the maximum DNS packet length to 512 bytes. To enable DNS snooping for the Botnet Traffic Filter, enter the **dynamic-filter-snoop** keyword. See the "Enabling DNS Snooping" section on page 26-10 for more information.
esmtp [*map_name*]	If you added an ESMTP inspection policy map according to "Configuring an ESMTP Inspection Policy Map for Additional Inspection Control" section on page 10-33, identify the map name in this command.
ftp [**strict** [*map_name*]]	Use the **strict** keyword to increase the security of protected networks by preventing web browsers from sending embedded commands in FTP requests. See the "Using the strict Option" section on page 10-11 for more information. If you added an FTP inspection policy map according to "Configuring an FTP Inspection Policy Map for Additional Inspection Control" section on page 10-12, identify the map name in this command.
gtp [*map_name*]	If you added a GTP inspection policy map according to the "Configuring a GTP Inspection Policy Map for Additional Inspection Control" section on page 13-4, identify the map name in this command.
h323 h225 [*map_name*]	If you added an H323 inspection policy map according to "Configuring an H.323 Inspection Policy Map for Additional Inspection Control" section on page 11-6, identify the map name in this command.

Table 9-2 Protocol Keywords

Keywords	Notes
h323 ras [*map_name*]	If you added an H323 inspection policy map according to “Configuring an H.323 Inspection Policy Map for Additional Inspection Control” section on page 11-6, identify the map name in this command.
http [*map_name*]	If you added an HTTP inspection policy map according to the “Configuring an HTTP Inspection Policy Map for Additional Inspection Control” section on page 10-16, identify the map name in this command.
icmp	—
icmp error	—
ils	—
im [*map_name*]	If you added an Instant Messaging inspection policy map according to “Configuring an Instant Messaging Inspection Policy Map for Additional Inspection Control” section on page 10-21, identify the map name in this command.
ip-options [*map_name*]	If you added an IP Options inspection policy map according to “Configuring an IP Options Inspection Policy Map for Additional Inspection Control” section on page 10-25, identify the map name in this command.
ipsec-pass-thru [*map_name*]	If you added an IPsec Pass Through inspection policy map according to “IPsec Pass Through Inspection” section on page 10-25, identify the map name in this command.
ipv6 [*map_name*]	If you added an IP Options inspection policy map according to “(Optional) Configuring an IPv6 Inspection Policy Map” section on page 10-27, identify the map name in this command.
mgcp [*map_name*]	If you added an MGCP inspection policy map according to “Configuring an MGCP Inspection Policy Map for Additional Inspection Control” section on page 11-12, identify the map name in this command.
netbios [*map_name*]	If you added a NetBIOS inspection policy map according to “Configuring a NetBIOS Inspection Policy Map for Additional Inspection Control” section on page 10-30, identify the map name in this command.
pptp	—
radius-accounting [*map_name*]	The **radius-accounting** keyword is only available for a management class map. See the “Creating a Layer 3/4 Class Map for Management Traffic” section on page 1-14 for more information about creating a management class map. If you added a RADIUS accounting inspection policy map according to “Configuring a RADIUS Inspection Policy Map for Additional Inspection Control” section on page 13-9, identify the map name in this command.
rsh	—

Table 9-2 Protocol Keywords

Keywords	Notes
rtsp [*map_name*]	If you added a RTSP inspection policy map according to "Configuring an RTSP Inspection Policy Map for Additional Inspection Control" section on page 11-16, identify the map name in this command.
scansafe [*map_name*]	If you added a ScanSafe (Cloud Web Security) inspection policy map according to "Configuring a Service Policy to Send Traffic to Cloud Web Security" section on page 25-10, identify the map name in this command.
sip [*map_name*]	If you added a SIP inspection policy map according to "Configuring a SIP Inspection Policy Map for Additional Inspection Control" section on page 11-20, identify the map name in this command.
skinny [*map_name*]	If you added a Skinny inspection policy map according to "Configuring a Skinny (SCCP) Inspection Policy Map for Additional Inspection Control" section on page 11-26, identify the map name in this command.
snmp [*map_name*]	If you added an SNMP inspection policy map according to "Configuring an SNMP Inspection Policy Map for Additional Inspection Control" section on page 13-10, identify the map name in this command.
sqlnet	—
sunrpc	The default class map includes UDP port 111; if you want to enable Sun RPC inspection for TCP port 111, you need to create a new class map that matches TCP port 111, add the class to the policy, and then apply the **inspect sunrpc** command to that class.
tftp	—
waas	—
xdmcp	—

Step 6 To activate the policy map on one or more interfaces, enter the following command:

```
ciscoasa(config)# service-policy policymap_name {global | interface interface_name}
```

Where **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. By default, the default policy map, "global_policy," is applied globally. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

CHAPTER 10

Configuring Inspection of Basic Internet Protocols

This chapter describes how to configure application layer protocol inspection. Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the ASA to do a deep packet inspection instead of passing the packet through the fast path. As a result, inspection engines can affect overall throughput.

Several common inspection engines are enabled on the ASA by default, but you might need to enable others depending on your network.

This chapter includes the following sections:

DNS Inspection

This section describes DNS application inspection. This section includes the following topics:

- Configuring DNS Inspection, page 10-8
- Monitoring DNS Inspection, page 10-9

Information About DNS Inspection

- General Information About DNS, page 10-2
- DNS Inspection Actions, page 10-2

General Information About DNS

A single connection is created for multiple DNS sessions, as long as they are between the same two hosts, and the sessions have the same 5-tuple (source/destination IP address, source/destination port, and protocol). DNS identification is tracked by app_id, and the idle timer for each app_id runs independently. Because the app_id expires independently, a legitimate DNS response can only pass through the ASA within a limited period of time and there is no resource build-up. However, if you enter the **show conn** command, you will see the idle timer of a DNS connection being reset by a new DNS session. This is due to the nature of the shared DNS connection and is by design.

DNS Inspection Actions

DNS inspection is enabled by default. You can customize DNS inspection to perform many tasks:

- Translate the DNS record based on the NAT configuration. For more information, see the "DNS and NAT" section on page 3-28.
- Enforce message length, domain-name length, and label length.
- Verify the integrity of the domain-name referred to by the pointer if compression pointers are encountered in the DNS message.
- Check to see if a compression pointer loop exists.
- Inspect packets based on the DNS header, type, class and more.

Default Settings for DNS Inspection

DNS inspection is enabled by default, using the preset_dns_map inspection class map:

- The maximum DNS message length is 512 bytes.
- The maximum client DNS message length is automatically set to match the Resource Record.
- DNS Guard is enabled, so the ASA tears down the DNS session associated with a DNS query as soon as the DNS reply is forwarded by the ASA. The ASA also monitors the message exchange to ensure that the ID of the DNS reply matches the ID of the DNS query.
- Translation of the DNS record based on the NAT configuration is enabled.
- Protocol enforcement is enabled, which enables DNS message format check, including domain name length of no more than 255 characters, label length of 63 characters, compression, and looped pointer check.

See the following default DNS inspection commands:

```
class-map inspection_default
 match default-inspection-traffic
```

```
policy-map type inspect dns preset_dns_map
 parameters
    message-length maximum client auto
    message-length maximum 512
    dns-guard
    protocol-enforcement
    nat-rewrite
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
! ...
service-policy global_policy global
```

(Optional) Configuring a DNS Inspection Policy Map and Class Map

To match DNS packets with certain characteristics and perform special actions, create a DNS inspection policy map. You can also configure a DNS inspection class map to group multiple match criteria for reference within the inspection policy map. You can then apply the inspection policy map when you enable DNS inspection.

Prerequisites

If you want to match a DNS message domain name list, then create a regular expression using one of the methods below:

- Create a regular expression (see the general operations configuration guide).
- Create a regular expression class map (see the general operations configuration guide).

Detailed Steps

	Command	Purpose
Step 1	Do one of the following:	
	class-map type inspect dns [**match-all** | **match-any**] *class_map_name* **Example:** `ciscoasa(config)# class-map type inspect dns match-all dns-class-map`	Creates a DNS inspection class map, where *class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match *all* criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. A class map groups multiple traffic matches. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps. The CLI enters class-map configuration mode, where you can enter one or more **match** or **match not** commands. For the traffic that you identify in this class map, you can only specify actions (such as drop) for the entire class. If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

	Command	Purpose
	policy-map type inspect dns *name* **Example:** `ciscoasa(config)# policy-map type inspect dns dns-map`	Creates an inspection policy map in which you want to match traffic directly. You can specify multiple **match** commands in the policy map. For information about the order of **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 2-4.

Step 2	**match** [**not**] **header-flag** [**eq**] {*f_well_known* [*f_well_known...*] | *f_value*}
	For direct match only:
	{**drop** [**log**] | **drop-connection** [**log**]| [**enforce-tsig** {[**drop**] [**log**]}] [**mask** [**log**]] | **log**}
	Example:
	`ciscoasa(config-pmap)# match header-flag AA QR`
	`ciscoasa(config-pmap-c)# mask log`
	`ciscoasa(config-pmap-c)# enforce-tsig log`
		Matches a specific flag or flags that are set in the DNS header, where the *f_well_known* argument is the DNS flag bit. The *f_value* argument is the 16-bit value in hex starting with 0x. The **eq** keyword specifies an exact match (match all); without the **eq** keyword, the packet only needs to match one of the specified headers (match any).
		To specify traffic that should not match, use the **match not** command.
		If you are matching directly in the inspection policy map, specify the action(s) for the match:
		• **drop** [**log**]—Drops the packet. **log** also logs the packet.
		• **drop-connection** [**log**]—Drops the packet and closes the connection. **log** also logs the packet.
		• **enforce-tsig** {[**drop**] [**log**]}—Enforces the TSIG resource record in a message. **drop** drops a packet without the TSIG resource record. **log** also logs the packet.
		• **mask** [**log**]—**M**asks out the matching portion of the packet. **log** also logs the packet.
		• **log**—Logs the packet.

Step 3	**match** [**not**] **dns-type** {**eq** {*t_well_known* | *t_val*}} {**range** *t_val1 t_val2*}
	For direct match only:
	{**drop** [**log**] | **drop-connection** [**log**]| **enforce-tsig** {[**drop**] [**log**]} | **log**}
	Example:
	`ciscoasa(config-pmap)# match dns-type eq aaaa`
	`ciscoasa(config-pmap-c)# enforce-tsig log`
		Matches a DNS type, where the *t_well_known* argument is the DNS flag bit. The *t_val* arguments are arbitrary values in the DNS type field (0-65535). The **range** keyword specifies a range, and the **eq** keyword specifies an exact match.
		To specify traffic that should not match, use the **match not** command.
		If you are matching directly in the inspection policy map, specify the action for the match:
		• **drop** [**log**]—Drops the packet. **log** also logs the packet.
		• **drop-connection** [**log**]—Drops the packet and closes the connection. **log** also logs the packet.
		• **enforce-tsig** {[**drop**] [**log**]}—Enforces the TSIG resource record in a message. **drop** drops a packet without the TSIG resource record. **log** also logs the packet.
		• **log**—Logs the packet.

	Command	Purpose

Step 4	**match** [**not**] **dns-class** {**eq** {**in** | *c_val*}} | **range** *c_val1 c_val2*}
	For direct match only:
	{**drop** [**log**] | **drop-connection** [**log**]| **enforce-tsig** {[**drop**] [**log**]} | **log**}
	Example:
	`ciscoasa(config-pmap)# match dns-class eq in`
	`ciscoasa(config-pmap-c)# log`
		Matches a DNS class, either **in** (for Internet) or *c_val,* an arbitrary value from 0 to 65535 in the DNS class field. The **range** keyword specifies a range, and the **eq** keyword specifies an exact match.
		To specify traffic that should not match, use the **match not** command.
		If you are matching directly in the inspection policy map, specify the action for the match:
		• **drop** [**log**]—Drops the packet. **log** also logs the packet.
		• **drop-connection** [**log**]—Drops the packet and closes the connection. **log** also logs the packet.
		• **enforce-tsig** {[**drop**] [**log**]}—Enforces the TSIG resource record in a message. **drop** drops a packet without the TSIG resource record. **log** also logs the packet.
		• **log**—Logs the packet.

Step 5	**match** {**question** | **resource-record** {**answer** | **authority** | **additional**}}
	For direct match only:
	{**drop** [**log**] | **drop-connection** [**log**]| **enforce-tsig** {[**drop**] [**log**]} | **log**}
	Example:
	`ciscoasa(config-pmap)# match resource-record answer`
	`ciscoasa(config-pmap-c)# drop-connection`
		Matches a DNS question or resource record, where the **question** keyword specifies the question portion of a DNS message. The **resource-record** keyword specifies the resource record portion of a DNS message; the **answer** keyword specifies the Answer RR section; the **authority** keyword specifies the Authority RR section; the **additional** keyword specifies the Additional RR section.
		To specify traffic that should not match, use the **match not** command.
		If you are matching directly in the inspection policy map, specify the action for the match:
		• **drop** [**log**]—Drops the packet. **log** also logs the packet.
		• **drop-connection** [**log**]—Drops the packet and closes the connection. **log** also logs the packet.
		• **enforce-tsig** {[**drop**] [**log**]}—Enforces the TSIG resource record in a message. **drop** drops a packet without the TSIG resource record. **log** also logs the packet.
		• **log**—Logs the packet.

	Command	Purpose

Step 6	**match** [**not**] **domain-name regex** {*regex_id* | **class** *class_id*}
	For direct match only:
	{**drop** [**log**] | **drop-connection** [**log**]| **enforce-tsig** {[**drop**] [**log**]} | **log**}
	Example:
	`ciscoasa(config-pmap)# match domain-name regex regex1`
	`ciscoasa(config-pmap-c)# drop-connection`
		Matches a DNS message domain name list. The *regex_name* argument is a regular expression. The **class** *regex_class_name* is a regular expression class map. See the "Prerequisites" section on page 10-3.
		To specify traffic that should not match, use the **match not** command.
		If you are matching directly in the inspection policy map, specify the action for the match:
		• **drop** [**log**]—Drops the packet. **log** also logs the packet.
		• **drop-connection** [**log**]—Drops the packet and closes the connection. **log** also logs the packet.
		• **enforce-tsig** {[**drop**] [**log**]}—Enforces the TSIG resource record in a message. **drop** drops a packet without the TSIG resource record. **log** also logs the packet.
		• **log**—Logs the packet.

	Command	Purpose

Step 7	(If you are using a DNS inspection class map)
	policy-map type inspect dns *name*
	class *class_map_name*
	{**drop** [**log**] | **drop-connection** [**log**]| **enforce-tsig** {[**drop**] [**log**]} | **mask** [**log**] | **log**}
	Example:
	`ciscoasa(config)# policy-map type inspect dns dns-map`
	`ciscoasa(config-pmap)# class dns-class-map`
	`ciscoasa(config-pmap-c)# drop`
	`ciscoasa(config-pmap-c)# match header-flag eq aa`
	`ciscoasa(config-pmap-c)# drop log`
		Creates an inspection policy map, specifies the DNS inspection class map, and sets the action for the class map:
		• **drop** [**log**]—Drops the packet. **log** also logs the packet.
		• **drop-connection** [**log**]—Drops the packet and closes the connection. **log** also logs the packet.
		• **enforce-tsig** {[**drop**] [**log**]}—Enforces the TSIG resource record in a message. **drop** drops a packet without the TSIG resource record. **log** also logs the packet.
		• **mask** [**log**]—Masks out the matching portion of the packet. **log** also logs the packet.
		• **log**—Logs the packet.
		You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 2-4.

Step 8	**parameters**
	{**dns-guard** | **id-mismatch count** *number* **duration** *seconds* **action log** | **id-randomization** | **message-length maximum** {*length* | **client** {[*length*] [**auto**]} | **server** {[*length*] [**auto**]}} | **nat-rewrite** | **protocol-enforcement** | **tsig enforced action** {[**drop**] [**log**]}}
	Example:
	`ciscoasa(config-pmap)# parameters`
	`ciscoasa(config-pmap-p)# dns-guard`
	`ciscoasa(config-pmap-p)# id-mismatch action log`
	`ciscoasa(config-pmap-p)# message-length maximum 1024`
	`ciscoasa(config-pmap-p)# nat-rewrite`
	`ciscoasa(config-pmap-p)# protocol-enforcement`
		Enters parameters configuration mode so you can set one or more parameters:
		• **dns-guard**—Enables DNS Guard. The ASA tears down the DNS session associated with a DNS query as soon as the DNS reply is forwarded by the ASA. The ASA also monitors the message exchange to ensure that the ID of the DNS reply matches the ID of the DNS query.
		• **id-mismatch count** *number* **duration** *seconds* **action log**—Enables logging for excessive DNS ID mismatches, where the **count** *number* **duration** *seconds* arguments specify the maximum number of mismatch instances per second before a system message log is sent.
		• **id-randomization**—Randomizes the DNS identifier for a DNS query.
		• **message-length maximum** {*length* | **client** {[*length*] [**auto**]} | **server** {[*length*] [**auto**]}}—Sets the maximum DNS message length, from 512 to 65535 bytes. You can also set the maximum length for client or server messages. **auto** sets the maximum length to the value in the Resource Record.
		• **nat-rewrite**—Translates the DNS record based on the NAT configuration.
		• **protocol-enforcement**—Enables DNS message format check, including domain name length of no more than 255 characters, label length of 63 characters, compression, and looped pointer check.
		• **tsig enforced action** {[**drop**] [**log**]}—Requires a TSIG resource record to be present. **drop** drops a non-conforming packet. **log** logs the packet.

Examples

The following example shows a how to define a DNS inspection policy map.

```
regex domain_example “example\.com”
regex domain_foo “foo\.com”

! define the domain names that the server serves
class-map type inspect regex match-any my_domains
    match regex domain_example
    match regex domain_foo

! Define a DNS map for query only
class-map type inspect dns match-all pub_server_map
    match not header-flag QR
    match question
    match not domain-name regex class my_domains

policy-map type inspect dns new_dns_map
    class pub_server_map
        drop log
    match header-flag RD
        mask log
    parameters
        message-length maximum client auto
        message-length maximum 512
        dns-guard
        protocol-enforcement
        nat-rewrite
```

Configuring DNS Inspection

The default ASA configuration includes many default inspections on default ports applied globally on all interfaces. A common method for customizing the inspection configuration is to customize the default global policy. The steps in this section show how to edit the default global policy, but you can alternatively create a new service policy as desired, for example, an interface-specific policy.

Detailed Steps

	Command	Purpose
Step 1	`class-map` *name* **Example:** `ciscoasa(config)# class-map dns_class_map`	Creates a class map to identify the traffic for which you want to apply the inspection. In the default global policy, the inspection_default class map is a special class map that includes default ports for all inspection types (**match default-inspection-traffic**). If you are using this class map in either the default policy or for a new service policy, you can skip this step and the next step.
Step 2	`match` *parameter* **Example:** `ciscoasa(config-cmap)# match access-list dns`	Specifies the traffic in the class map. See the “Identifying Traffic (Layer 3/4 Class Maps)” section on page 1-12 for more information.

	Command	Purpose
Step 3	**policy-map** *name* **Example:** `ciscoasa(config)# policy-map global_policy`	Adds or edits a policy map that sets the actions to take with the class map traffic. In the default configuration, the global_policy policy map is assigned globally to all interfaces. If you want to edit the global_policy, enter global_policy as the policy name.

Step 4	**class** *name*
	Example:
	`ciscoasa(config-pmap)# class inspection_default`
		Identifies the class map created in Step 1.
		To edit the default policy, or to use the special inspection_default class map in a new policy, specify **inspection_default** for the *name*.

Step 5	**inspect dns** [*dns_policy_map*] [**dynamic-filter-snoop**]
	Example:
	`ciscoasa(config-class)# no inspect dns`
	`ciscoasa(config-class)# inspect dns dns-map`
		Configures DNS inspection. Specify the inspection policy map you created in the "(Optional) Configuring a DNS Inspection Policy Map and Class Map" section on page 10-3.
		For information about the Botnet Traffic Filter **dynamic-filter-snoop** keyword, see the "Enabling DNS Snooping" section on page 26-10.
		Note If you are editing the default global policy (or any in-use policy) to use a different DNS inspection policy map from the default preset_dns_map, you must remove the DNS inspection with the **no inspect dns** command, and then re-add it with the new DNS inspection policy map name.

Step 6	**service-policy** *policymap_name* {**global** | **interface** *interface_name*}
	Example:
	`ciscoasa(config)# service-policy global_policy global`
		Activates the policy map on one or more interfaces. **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.
		The default configuration includes a global policy called global_policy. If you are editing that policy, you can skip this step.

Examples

The following example shows a how to use a new inspection policy map in the global default configuration:

```
policy-map global_policy
 class inspection_default
  no inspect dns preset_dns_map
  inspect dns new_dns_map
service-policy global_policy global
```

Monitoring DNS Inspection

To view information about the current DNS connections, enter the following command:

```
ciscoasa# show conn
```

For connections using a DNS server, the source port of the connection may be replaced by the IP address of DNS server in the show conn command output.

A single connection is created for multiple DNS sessions, as long as they are between the same two hosts, and the sessions have the same 5-tuple (source/destination IP address, source/destination port, and protocol). DNS identification is tracked by app_id, and the idle timer for each app_id runs independently.

Because the app_id expires independently, a legitimate DNS response can only pass through the security appliance within a limited period of time and there is no resource build-up. However, when you enter the **show conn** command, you see the idle timer of a DNS connection being reset by a new DNS session. This is due to the nature of the shared DNS connection and is by design.

To display the statistics for DNS application inspection, enter the **show service-policy** command. The following is sample output from the **show service-policy** command:

```
ciscoasa# show service-policy
Interface outside:
  Service-policy: sample_policy
    Class-map: dns_port
      Inspect: dns maximum-length 1500, packet 0, drop 0, reset-drop 0
```

FTP Inspection

This section describes the FTP inspection engine. This section includes the following topics:

- FTP Inspection Overview, page 10-10
- Using the strict Option, page 10-11
- Configuring an FTP Inspection Policy Map for Additional Inspection Control, page 10-12
- Verifying and Monitoring FTP Inspection, page 10-15

FTP Inspection Overview

The FTP application inspection inspects the FTP sessions and performs four tasks:

- Prepares dynamic secondary data connection
- Tracks the FTP command-response sequence
- Generates an audit trail
- Translates the embedded IP address

FTP application inspection prepares secondary channels for FTP data transfer. Ports for these channels are negotiated through PORT or PASV commands. The channels are allocated in response to a file upload, a file download, or a directory listing event.

Note If you disable FTP inspection engines with the **no inspect ftp** command, outbound users can start connections only in passive mode, and all inbound FTP is disabled.

Using the strict Option

Using the **strict** option with the **inspect ftp** command increases the security of protected networks by preventing web browsers from sending embedded commands in FTP requests.

Note To specify FTP commands that are not permitted to pass through the ASA, create an FTP map according to the "Configuring an FTP Inspection Policy Map for Additional Inspection Control" section on page 10-12.

After you enable the **strict** option on an interface, FTP inspection enforces the following behavior:

- An FTP command must be acknowledged before the ASA allows a new command.
- The ASA drops connections that send embedded commands.
- The 227 and PORT commands are checked to ensure they do not appear in an error string.

Caution Using the **strict** option may cause the failure of FTP clients that are not strictly compliant with FTP RFCs.

If the **strict** option is enabled, each FTP command and response sequence is tracked for the following anomalous activity:

- Truncated command—Number of commas in the PORT and PASV reply command is checked to see if it is five. If it is not five, then the PORT command is assumed to be truncated and the TCP connection is closed.
- Incorrect command—Checks the FTP command to see if it ends with <CR><LF> characters, as required by the RFC. If it does not, the connection is closed.
- Size of RETR and STOR commands—These are checked against a fixed constant. If the size is greater, then an error message is logged and the connection is closed.
- Command spoofing—The PORT command should always be sent from the client. The TCP connection is denied if a PORT command is sent from the server.
- Reply spoofing—PASV reply command (227) should always be sent from the server. The TCP connection is denied if a PASV reply command is sent from the client. This prevents the security hole when the user executes "227 xxxxx a1, a2, a3, a4, p1, p2."
- TCP stream editing—The ASA closes the connection if it detects TCP stream editing.
- Invalid port negotiation—The negotiated dynamic port value is checked to see if it is less than 1024. As port numbers in the range from 1 to 1024 are reserved for well-known connections, if the negotiated port falls in this range, then the TCP connection is freed.
- Command pipelining—The number of characters present after the port numbers in the PORT and PASV reply command is cross checked with a constant value of 8. If it is more than 8, then the TCP connection is closed.
- The ASA replaces the FTP server response to the SYST command with a series of Xs. to prevent the server from revealing its system type to FTP clients. To override this default behavior, use the **no mask-syst-reply** command in the FTP map.

Configuring an FTP Inspection Policy Map for Additional Inspection Control

FTP command filtering and security checks are provided using strict FTP inspection for improved security and control. Protocol conformance includes packet length checks, delimiters and packet format checks, command terminator checks, and command validation.

Blocking FTP based on user values is also supported so that it is possible for FTP sites to post files for download, but restrict access to certain users. You can block FTP connections based on file type, server name, and other attributes. System message logs are generated if an FTP connection is denied after inspection.

If you want FTP inspection to allow FTP servers to reveal their system type to FTP clients, and limit the allowed FTP commands, then create and configure an FTP map. You can then apply the FTP map when you enable FTP inspection.

To create an FTP map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the general operations configuration guide. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the general operations configuration guide.

Step 3 (Optional) Create an FTP inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string “example.com,” then any traffic that includes “example.com” does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop, drop-connection, reset, mask, set the rate limit, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
ciscoasa(config)# class-map type inspect ftp [match-all | match-any] class_map_name
ciscoasa(config-cmap)#
```

Where *class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
ciscoasa(config-cmap)# description string
```

c. (Optional) To match a filename for FTP transfer, enter the following command:

```
ciscoasa(config-cmap)# match [not] filename regex [regex_name |
class regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

d. (Optional) To match a file type for FTP transfer, enter the following command:

```
ciscoasa(config-cmap)# match [not] filetype regex [regex_name |
class regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

e. (Optional) To disallow specific FTP commands, use the following command:

```
ciscoasa(config-cmap)# match [not] request-command ftp_command [ftp_command...]
```

Where *ftp_command* with one or more FTP commands that you want to restrict. See Table 10-1 for a list of the FTP commands that you can restrict.

.

Table 10-1 FTP Map request-command deny Options

request-command deny Option	Purpose
appe	Disallows the command that appends to a file.
cdup	Disallows the command that changes to the parent directory of the current working directory.
dele	Disallows the command that deletes a file on the server.
get	Disallows the client command for retrieving a file from the server.
help	Disallows the command that provides help information.
mkd	Disallows the command that makes a directory on the server.
put	Disallows the client command for sending a file to the server.
rmd	Disallows the command that deletes a directory on the server.
rnfr	Disallows the command that specifies rename-from filename.
rnto	Disallows the command that specifies rename-to filename.
site	Disallows the command that are specific to the server system. Usually used for remote administration.
stou	Disallows the command that stores a file using a unique file name.

f. (Optional) To match an FTP server, enter the following command:

```
ciscoasa(config-cmap)# match [not] server regex [regex_name | class regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

g. (Optional) To match an FTP username, enter the following command:

```
ciscoasa(config-cmap)# match [not] username regex [regex_name |
class regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

Step 4 Create an FTP inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect ftp policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the FTP class map that you created in Step 3 by entering the following command:

```
ciscoasa(config-pmap)# class class_map_name
ciscoasa(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
ciscoasa(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the command reference for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 2-4.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

b. To mask the greeting banner from the FTP server, enter the following command:

```
ciscoasa(config-pmap-p)# mask-banner
```

c. To mask the reply to **syst** command, enter the following command:

```
ciscoasa(config-pmap-p)# mask-syst-reply
```

Before submitting a username and password, all FTP users are presented with a greeting banner. By default, this banner includes version information useful to hackers trying to identify weaknesses in a system. The following example shows how to mask this banner:

```
ciscoasa(config)# policy-map type inspect ftp mymap
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# mask-banner

ciscoasa(config)# class-map match-all ftp-traffic
ciscoasa(config-cmap)# match port tcp eq ftp

ciscoasa(config)# policy-map ftp-policy
ciscoasa(config-pmap)# class ftp-traffic
ciscoasa(config-pmap-c)# inspect ftp strict mymap

ciscoasa(config)# service-policy ftp-policy interface inside
```

Verifying and Monitoring FTP Inspection

FTP application inspection generates the following log messages:

- An Audit record 303002 is generated for each file that is retrieved or uploaded.
- The FTP command is checked to see if it is RETR or STOR and the retrieve and store commands are logged.
- The username is obtained by looking up a table providing the IP address.
- The username, source IP address, destination IP address, NAT address, and the file operation are logged.
- Audit record 201005 is generated if the secondary dynamic channel preparation failed due to memory shortage.

In conjunction with NAT, the FTP application inspection translates the IP address within the application payload. This is described in detail in RFC 959.

HTTP Inspection

This section describes the HTTP inspection engine. This section includes the following topics:

- HTTP Inspection Overview, page 10-15
- Configuring an HTTP Inspection Policy Map for Additional Inspection Control, page 10-16

HTTP Inspection Overview

Use the HTTP inspection engine to protect against specific attacks and other threats that are associated with HTTP traffic. HTTP inspection performs several functions:

- Enhanced HTTP inspection
- URL screening through N2H2 or Websense

  See Information About URL Filtering, page 29-6 for information.
- Java and ActiveX filtering

The latter two features are configured in conjunction with the **filter** command. For more information about filtering, see Chapter 29, “Configuring Filtering Services.”

The enhanced HTTP inspection feature, which is also known as an application firewall and is available when you configure an HTTP map (see "Configuring an HTTP Inspection Policy Map for Additional Inspection Control"), can help prevent attackers from using HTTP messages for circumventing network security policy. It verifies the following for all HTTP messages:

- Conformance to RFC 2616
- Use of RFC-defined methods only.
- Compliance with the additional criteria.

Configuring an HTTP Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an HTTP inspection policy map. You can then apply the inspection policy map when you enable HTTP inspection.

Note When you enable HTTP inspection with an inspection policy map, strict HTTP inspection with the action reset and log is enabled by default. You can change the actions performed in response to inspection failure, but you cannot disable strict inspection as long as the inspection policy map remains enabled.

To create an HTTP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the general operations configuration guide. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the general operations configuration guide.

Step 3 (Optional) Create an HTTP inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop, drop-connection, reset, mask, set the rate limit, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
ciscoasa(config)# class-map type inspect http [match-all | match-any] class_map_name
ciscoasa(config-cmap)#
```

Where *class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
ciscoasa(config-cmap)# description string
```

c. (Optional) To match traffic with a content-type field in the HTTP response that does not match the accept field in the corresponding HTTP request message, enter the following command:

```
ciscoasa(config-cmap)# match [not] req-resp content-type mismatch
```

d. (Optional) To match text found in the HTTP request message arguments, enter the following command:

```
ciscoasa(config-cmap)# match [not] request args regex [regex_name | class
regex_class_name]
```

Where the *regex_name* is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

e. (Optional) To match text found in the HTTP request message body or to match traffic that exceeds the maximum HTTP request message body length, enter the following command:

```
ciscoasa(config-cmap)# match [not] request body {regex [regex_name | class
regex_class_name] | length gt max_bytes}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes.

f. (Optional) To match text found in the HTTP request message header, or to restrict the count or length of the header, enter the following command:

```
ciscoasa(config-cmap)# match [not] request header {[field]
[regex [regex_name | class regex_class_name]] |
[length gt max_length_bytes | count gt max_count_bytes]}
```

Where the *field* is the predefined message header keyword. The **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes. The **count gt** *max_count* is the maximum number of header fields.

g. (Optional) To match text found in the HTTP request message method, enter the following command:

```
ciscoasa(config-cmap)# match [not] request method {[method] |
[regex [regex_name | class regex_class_name]]
```

Where the *method* is the predefined message method keyword. The **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

h. (Optional) To match text found in the HTTP request message URI, enter the following command:

```
ciscoasa(config-cmap)# match [not] request uri {regex [regex_name | class
regex_class_name] | length gt max_bytes}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes.

i. Optional) To match text found in the HTTP response message body, or to comment out Java applet and Active X object tags in order to filter them, enter the following command:

```
ciscoasa(config-cmap)# match [not] response body {[active-x] | [java-applet] |
[regex [regex_name | class regex_class_name]] | length gt max_bytes}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes.

j. (Optional) To match text found in the HTTP response message header, or to restrict the count or length of the header, enter the following command:

```
ciscoasa(config-cmap)# match [not] response header {[field]
[regex [regex_name | class regex_class_name]] |
[length gt max_length_bytes | count gt max_count]}
```

Where the *field* is the predefined message header keyword. The **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2. The **length gt** *max_bytes* is the maximum message body length in bytes. The **count gt** *max_count* is the maximum number of header fields.

k. (Optional) To match text found in the HTTP response message status line, enter the following command:

```
ciscoasa(config-cmap)# match [not] response status-line {regex [regex_name | class
regex_class_name]}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

Step 4 Create an HTTP inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect http policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the HTTP class map that you created in Step 3 by entering the following command:

```
ciscoasa(config-pmap)# class class_map_name
ciscoasa(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
ciscoasa(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the command reference for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 2-4.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

b. To check for HTTP protocol violations, enter the following command:

```
ciscoasa(config-pmap-p)# protocol-violation [action [drop-connection | reset | log]]
```

Where the **drop-connection** action closes the connection. The **reset** action closes the connection and sends a TCP reset to the client. The **log** action sends a system log message when this policy map matches traffic.

c. To substitute a string for the server header field, enter the following command:

```
ciscoasa(config-pmap-p)# spoof-server string
```

Where the *string* argument is the string to substitute for the server header field. Note: WebVPN streams are not subject to the **spoof-server** comand.

The following example shows how to define an HTTP inspection policy map that will allow and log any HTTP connection that attempts to access "www\.xyz.com/.*\.asp" or "www\.xyz[0-9][0-9]\.com" with methods "GET" or "PUT." All other URL/Method combinations will be silently allowed.

```
ciscoasa(config)# regex url1 "www\.xyz.com/.*\.asp"
ciscoasa(config)# regex url2 "www\.xyz[0-9][0-9]\.com"
ciscoasa(config)# regex get "GET"
ciscoasa(config)# regex put "PUT"

ciscoasa(config)# class-map type regex match-any url_to_log
ciscoasa(config-cmap)# match regex url1
ciscoasa(config-cmap)# match regex url2
ciscoasa(config-cmap)# exit

ciscoasa(config)# class-map type regex match-any methods_to_log
ciscoasa(config-cmap)# match regex get
ciscoasa(config-cmap)# match regex put
ciscoasa(config-cmap)# exit

ciscoasa(config)# class-map type inspect http http_url_policy
ciscoasa(config-cmap)# match request uri regex class url_to_log
ciscoasa(config-cmap)# match request method regex class methods_to_log
ciscoasa(config-cmap)# exit

ciscoasa(config)# policy-map type inspect http http_policy
ciscoasa(config-pmap)# class http_url_policy
ciscoasa(config-pmap-c)# log
```

ICMP Inspection

The ICMP inspection engine allows ICMP traffic to have a "session" so it can be inspected like TCP and UDP traffic. Without the ICMP inspection engine, we recommend that you do not allow ICMP through the ASA in an ACL. Without stateful inspection, ICMP can be used to attack your network. The ICMP inspection engine ensures that there is only one response for each request, and that the sequence number is correct.

ICMP Error Inspection

When this feature is enabled, the ASA creates translation sessions for intermediate hops that send ICMP error messages, based on the NAT configuration. The ASA overwrites the packet with the translated IP addresses.

When disabled, the ASA does not create translation sessions for intermediate nodes that generate ICMP error messages. ICMP error messages generated by the intermediate nodes between the inside host and the ASA reach the outside host without consuming any additional NAT resource. This is undesirable when an outside host uses the traceroute command to trace the hops to the destination on the inside of the ASA. When the ASA does not translate the intermediate hops, all the intermediate hops appear with the mapped destination IP address.

The ICMP payload is scanned to retrieve the five-tuple from the original packet. Using the retrieved five-tuple, a lookup is performed to determine the original address of the client. The ICMP error inspection engine makes the following changes to the ICMP packet:

- In the IP Header, the mapped IP is changed to the real IP (Destination Address) and the IP checksum is modified.
- In the ICMP Header, the ICMP checksum is modified due to the changes in the ICMP packet.
- In the Payload, the following changes are made:
  - Original packet mapped IP is changed to the real IP
  - Original packet mapped port is changed to the real Port
  - Original packet IP checksum is recalculated

Instant Messaging Inspection

This section describes the IM inspection engine. This section includes the following topics:

- IM Inspection Overview, page 10-20
- Configuring an Instant Messaging Inspection Policy Map for Additional Inspection Control, page 10-21

IM Inspection Overview

The IM inspect engine lets you apply fine grained controls on the IM application to control the network usage and stop leakage of confidential data, propagation of worms, and other threats to the corporate network.

Configuring an Instant Messaging Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an IM inspection policy map. You can then apply the inspection policy map when you enable IM inspection.

To create an IM inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the general operations configuration guide. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the general operations configuration guide.

Step 3 (Optional) Create an IM inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop-connection, reset, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
ciscoasa(config)# class-map type inspect im [match-all | match-any] class_map_name
ciscoasa(config-cmap)#
```

Where *the class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
ciscoasa(config-cmap)# description string
```

Where *the string* is the description of the class map (up to 200 characters).

c. (Optional) To match traffic of a specific IM protocol, such as Yahoo or MSN, enter the following command:

```
ciscoasa(config-cmap)# match [not] protocol {im-yahoo | im-msn}
```

d. (Optional) To match a specific IM service, such as chat, file-transfer, webcam, voice-chat, conference, or games, enter the following command:

```
ciscoasa(config-cmap)# match [not] service {chat | file-transfer | webcam | voice-chat
| conference | games}
```

e. (Optional) To match the source login name of the IM message, enter the following command:

```
ciscoasa(config-cmap)# match [not] login-name regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

f. (Optional) To match the destination login name of the IM message, enter the following command:

```
ciscoasa(config-cmap)# match [not] peer-login-name regex {class class_name |
regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

g. (Optional) To match the source IP address of the IM message, enter the following command:

```
ciscoasa(config-cmap)# match [not] ip-address ip_address ip_address_mask
```

Where the *ip_address* and the *ip_address_mask* is the IP address and netmask of the message source.

h. (Optional) To match the destination IP address of the IM message, enter the following command:

```
ciscoasa(config-cmap)# match [not] peer-ip-address ip_address ip_address_mask
```

Where the *ip_address* and the *ip_address_mask* is the IP address and netmask of the message destination.

i. (Optional) To match the version of the IM message, enter the following command:

```
ciscoasa(config-cmap)# match [not] version regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

j. (Optional) To match the filename of the IM message, enter the following command:

```
ciscoasa(config-cmap)# match [not] filename regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

Note Not supported using MSN IM protocol.

Step 4 Create an IM inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect im policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 6 Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the IM class map that you created in Step 3 by entering the following command:

```
ciscoasa(config-pmap)# class class_map_name
ciscoasa(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 2-4.

Step 7 Specify the action you want to perform on the matching traffic by entering the following command:

```
ciscoasa(config-pmap-c)# {drop-connection | reset | log}
```

Where the **drop-connection** action closes the connection. The **reset** action closes the connection and sends a TCP reset to the client. The **log** action sends a system log message when this policy map matches traffic.

The following example shows how to define an IM inspection policy map.

```
hostname(config)# regex loginname1 "ying\@yahoo.com"
hostname(config)# regex loginname2 "Kevin\@yahoo.com"
hostname(config)# regex loginname3 "rahul\@yahoo.com"
hostname(config)# regex loginname4 "darshant\@yahoo.com"
hostname(config)# regex yahoo_version_regex "1\.0"
hostname(config)# regex gif_files ".*\.gif"
hostname(config)# regex exe_files ".*\.exe"

hostname(config)# class-map type regex match-any yahoo_src_login_name_regex
hostname(config-cmap)# match regex loginname1
hostname(config-cmap)# match regex loginname2

hostname(config)# class-map type regex match-any yahoo_dst_login_name_regex
hostname(config-cmap)# match regex loginname3
hostname(config-cmap)# match regex loginname4

hostname(config)# class-map type inspect im match-any yahoo_file_block_list
hostname(config-cmap)# match filename regex gif_files
hostname(config-cmap)# match filename regex exe_files

hostname(config)# class-map type inspect im match-all yahoo_im_policy
hostname(config-cmap)# match login-name regex class yahoo_src_login_name_regex
hostname(config-cmap)# match peer-login-name regex class yahoo_dst_login_name_regex

hostname(config)# class-map type inspect im match-all yahoo_im_policy2
hostname(config-cmap)# match version regex yahoo_version_regex

hostname(config)# class-map im_inspect_class_map
hostname(config-cmap)# match default-inspection-traffic

hostname(config)# policy-map type inspect im im_policy_all
hostname(config-pmap)# class yahoo_file_block_list
hostname(config-pmap-c)# match service file-transfer
hostname(config-pmap)# class yahoo_im_policy
hostname(config-pmap-c)# drop-connection
hostname(config-pmap)# class yahoo_im_policy2
hostname(config-pmap-c)# reset
hostname(config)# policy-map global_policy_name
hostname(config-pmap)# class im_inspect_class_map
hostname(config-pmap-c)# inspect im im_policy_all
```

IP Options Inspection

This section describes the IP Options inspection engine. This section includes the following topics:

IP Options Inspection Overview

Each IP packet contains an IP header with the Options field. The Options field, commonly referred to as IP Options, provide for control functions that are required in some situations but unnecessary for most common communications. In particular, IP Options include provisions for time stamps, security, and special routing. Use of IP Options is optional, and the field can contain zero, one, or more options.

You can configure IP Options inspection to control which IP packets with specific IP options are allowed through the ASA. Configuring this inspection instructs the ASA to allow a packet to pass or to clear the specified IP options and then allow the packet to pass.

IP Options inspection can check for the following three IP options in a packet:

- End of Options List (EOOL) or IP Option 0—This option, which contains just a single zero byte, appears at the end of all options to mark the end of a list of options. This might not coincide with the end of the header according to the header length.
- No Operation (NOP) or IP Option 1—The Options field in the IP header can contain zero, one, or more options, which makes the total length of the field variable. However, the IP header must be a multiple of 32 bits. If the number of bits of all options is not a multiple of 32 bits, the NOP option is used as “internal padding” to align the options on a 32-bit boundary.
- Router Alert (RTRALT) or IP Option 20—This option notifies transit routers to inspect the contents of the packet even when the packet is not destined for that router. This inspection is valuable when implementing RSVP and similar protocols require relatively complex processing from the routers along the packets delivery path.

Note IP Options inspection is included by default in the global inspection policy. Therefore, the ASA allows RSVP traffic that contains packets with the Router Alert option (option 20) when the ASA is in routed mode.

Dropping RSVP packets containing the Router Alert option can cause problems in VoIP implementations.

When you configure the ASA to clear the Router Alert option from IP headers, the IP header changes in the following ways:

- The Options field is padded so that the field ends on a 32 bit boundary.
- Internet header length (IHL) changes.
- The total length of the packet changes.
- The checksum is recomputed.

If an IP header contains additional options other than EOOL, NOP, or RTRALT, regardless of whether the ASA is configured to allow these options, the ASA will drop the packet.

Configuring an IP Options Inspection Policy Map for Additional Inspection Control

Step 1 To create an IP Options inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect ip-options policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 2 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 3 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

b. To allow or clear packets with the End of Options List (EOOL) option, enter the following command:

```
ciscoasa(config-pmap-p)# eool action {allow | clear}
```

This option, which contains just a single zero byte, appears at the end of all options to mark the end of a list of options. This might not coincide with the end of the header according to the header length.

c. To allow or clear packets with the No Operation (NOP) option, enter the following command:

```
ciscoasa(config-pmap-p)# nop action {allow | clear}
```

The Options field in the IP header can contain zero, one, or more options, which makes the total length of the field variable. However, the IP header must be a multiple of 32 bits. If the number of bits of all options is not a multiple of 32 bits, the NOP option is used as "internal padding" to align the options on a 32-bit boundary.

d. To allowor clear packets with the Router Alert (RTRALT) option, enter the following command:

```
ciscoasa(config-pmap-p)# router-alert action {allow | clear}
```

This option notifies transit routers to inspect the contents of the packet even when the packet is not destined for that router. This inspection is valuable when implementing RSVP and similar protocols require relatively complex processing from the routers along the packets delivery path.

Note Enter the **clear** command to clear the IP option from the packet before allowing the packet through the ASA.

IPsec Pass Through Inspection

This section describes the IPsec Pass Through inspection engine. This section includes the following topics:

- IPsec Pass Through Inspection Overview, page 10-26
- "Example for Defining an IPsec Pass Through Parameter Map" section on page 10-26

IPsec Pass Through Inspection Overview

Internet Protocol Security (IPsec) is a protocol suite for securing IP communications by authenticating and encrypting each IP packet of a data stream. IPsec also includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used to protect data flows between a pair of hosts (for example, computer users or servers), between a pair of security gateways (such as routers or firewalls), or between a security gateway and a host.

IPsec Pass Through application inspection provides convenient traversal of ESP (IP protocol 50) and AH (IP protocol 51) traffic associated with an IKE UDP port 500 connection. It avoids lengthy ACL configuration to permit ESP and AH traffic and also provides security using timeout and max connections.

Specify IPsec Pass Through inspection parameters to identify a specific map to use for defining the parameters for the inspection. Configure a policy map for Specify IPsec Pass Through inspection to access the parameters configuration, which lets you specify the restrictions for ESP or AH traffic. You can set the per client max connections and the idle timeout in parameters configuration.

NAT and non-NAT traffic is permitted. However, PAT is not supported.

Example for Defining an IPsec Pass Through Parameter Map

The following example shows how to use ACLs to identify IKE traffic, define an IPsec Pass Thru parameter map, define a policy, and apply the policy to the outside interface:

```
hostname(config)# access-list ipsecpassthruacl permit udp any any eq 500
hostname(config)# class-map ipsecpassthru-traffic
hostname(config-cmap)# match access-list ipsecpassthruacl
hostname(config)# policy-map type inspect ipsec-pass-thru iptmap
hostname(config-pmap)# parameters
hostname(config-pmap-p)# esp per-client-max 10 timeout 0:11:00
hostname(config-pmap-p)# ah per-client-max 5 timeout 0:06:00
hostname(config)# policy-map inspection_policy
hostname(config-pmap)# class ipsecpassthru-traffic
hostname(config-pmap-c)# inspect ipsec-pass-thru iptmap
hostname(config)# service-policy inspection_policy interface outside
```

IPv6 Inspection

- Information about IPv6 Inspection, page 10-27
- Default Settings for IPv6 Inspection, page 10-27
- (Optional) Configuring an IPv6 Inspection Policy Map, page 10-27
- Configuring IPv6 Inspection, page 10-29

Information about IPv6 Inspection

IPv6 inspection lets you selectively log or drop IPv6 traffic based on the extension header. In addition, IPv6 inspection can check conformance to RFC 2460 for type and order of extension headers in IPv6 packets.

Default Settings for IPv6 Inspection

If you enable IPv6 inspection and do not specify an inspection policy map, then the default IPv6 inspection policy map is used, and the following actions are taken:

- Allows only known IPv6 extension headers
- Enforces the order of IPv6 extension headers as defined in the RFC 2460 specification

If you create an inspection policy map, the above actions are taken by default unless you explicitly disable them.

(Optional) Configuring an IPv6 Inspection Policy Map

To identify extension headers to drop or log, and/or to disable packet verification, create an IPv6 inspection policy map to be used by the service policy.

Detailed Steps

	Command	Purpose
Step 1	`policy-map type inspect ipv6` *name* **Example:** `ciscoasa(config)# policy-map type inspect ipv6 ipv6-map`	Creates an inspection policy map.

Step 2	`match header` *header*
	[`drop` [`log`] | `log`]
	Example:
	`ciscoasa(config-pmap)# match header ah`
	`ciscoasa(config-pmap-c)# drop log`
	`ciscoasa(config-pmap-c)# match header esp`
	`ciscoasa(config-pmap-c)# drop log`
		Specifies the headers you want to match. By default, the packet is logged (**log**); if you want to drop (and optionally also log) the packet, enter the **drop** and optional **log** commands in match configuration mode.
		Re-enter the **match** command and optional **drop** action for each extension you want to match:
		• **ah**—Matches the IPv6 Authentication extension header
		• **count gt** *number*—Specifies the maximum number of IPv6 extension headers, from 0 to 255
		• **destination-option**—Matches the IPv6 destination-option extension header
		• **esp**—Matches the IPv6 Encapsulation Security Payload (ESP) extension header
		• **fragment**—Matches the IPv6 fragment extension header
		• **hop-by-hop**—Matches the IPv6 hop-by-hop extension header
		• **routing-address count gt** *number*—Sets the maximum number of IPv6 routing header type 0 addresses, greater than a number between 0 and 255
		• **routing-type** {**eq** | **range**} *number*—Matches the IPv6 routing header type, from 0 to 255. For a range, separate values by a space, for example, **30 40**.

Step 3	`parameters`
	[`no`] `verify-header` {`order` | `type`}
	Example:
	`ciscoasa(config-pmap)# parameters`
	`ciscoasa(config-pmap-p)# no verify-header order`
	`ciscoasa(config-pmap-p)# no verify-header type`
		Specifies IPv6 parameters. These parameters are enabled by default. To disable them, enter the **no** keyword.
		• [**no**] **verify-header type**—Allows only known IPv6 extension headers
		• [**no**] **verify-header order**—Enforces the order of IPv6 extension headers as defined in the RFC 2460 specification

Examples

The following example creates an inspection policy map that will drop and log all IPv6 packets with the hop-by-hop, destination-option, routing-address, and routing type 0 headers:

```
policy-map type inspect ipv6 ipv6-pm
 parameters
 match header hop-by-hop
```

```
  drop log
match header destination-option
  drop log
match header routing-address count gt 0
  drop log
match header routing-type eq 0
  drop log
```

Configuring IPv6 Inspection

To enable IPv6 inspection, perform the following steps.

Detailed Steps

	Command	Purpose

Step 1	**class-map** *name*
	Example:
	`ciscoasa(config)# class-map ipv6_traffic`
		Creates a class map to identify the traffic for which you want to apply the inspection.

Step 2	**match** *parameter* **Example:** `ciscoasa(config-cmap)# match access-list ipv6`	Specifies the traffic in the class map. See the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12 for more information.
Step 3	**policy-map** *name* **Example:** `ciscoasa(config)# policy-map ipv6_policy`	Adds or edits a policy map that sets the actions to take with the class map traffic.

Step 4	**class** *name*
	Example:
	`ciscoasa(config-pmap)# class ipv6_traffic`
		Identifies the class map created in Step 1

Step 5	**inspect ipv6** [*ipv6_policy_map*] **Example:** `ciscoasa(config-class)# inspect ipv6 ipv6-map`	Configures IPv6 inspection. Specify the inspection policy map you created in the "(Optional) Configuring an IPv6 Inspection Policy Map" section on page 10-27.

Step 6	**service-policy** *policymap_name* {**global** | **interface** *interface_name*}
	Example:
	`ciscoasa(config)# service-policy ipv6_policy outside`
		Activates the policy map on one or more interfaces. **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

Examples

The following example drops all IPv6 traffic with the hop-by-hop, destination-option, routing-address, and routing type 0 headers:

```
policy-map type inspect ipv6 ipv6-pm
 parameters
 match header hop-by-hop
  drop
 match header destination-option
  drop
 match header routing-address count gt 0
  drop
 match header routing-type eq 0
  drop
policy-map global_policy
 class class-default
  inspect ipv6 ipv6-pm
!
service-policy global_policy global
```

NetBIOS Inspection

This section describes the IM inspection engine. This section includes the following topics:

- NetBIOS Inspection Overview, page 10-30
- Configuring a NetBIOS Inspection Policy Map for Additional Inspection Control, page 10-30

NetBIOS Inspection Overview

NetBIOS inspection is enabled by default. The NetBios inspection engine translates IP addresses in the NetBios name service (NBNS) packets according to the ASA NAT configuration.

Configuring a NetBIOS Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create a NETBIOS inspection policy map. You can then apply the inspection policy map when you enable NETBIOS inspection.

To create a NETBIOS inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the general operations configuration guide. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the general operations configuration guide.

Step 3 Create a NetBIOS inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect netbios policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 4 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 5 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the NetBIOS class map that you created in Step 3 by entering the following command:

```
ciscoasa(config-pmap)# class class_map_name
ciscoasa(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
ciscoasa(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the command reference for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 2-4.

Step 6 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

b. To check for NETBIOS protocol violations, enter the following command:

```
ciscoasa(config-pmap-p)# protocol-violation [action [drop-connection / reset / log]]
```

Where the **drop-connection** action closes the connection. The **reset** action closes the connection and sends a TCP reset to the client. The **log** action sends a system log message when this policy map matches traffic.

The following example shows how to define a NETBIOS inspection policy map.

```
ciscoasa(config)# policy-map type inspect netbios netbios_map
ciscoasa(config-pmap)# protocol-violation drop log
ciscoasa(config)# policy-map netbios_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect netbios netbios_map
```

PPTP Inspection

PPTP is a protocol for tunneling PPP traffic. A PPTP session is composed of one TCP channel and usually two PPTP GRE tunnels. The TCP channel is the control channel used for negotiating and managing the PPTP GRE tunnels. The GRE tunnels carries PPP sessions between the two hosts.

When enabled, PPTP application inspection inspects PPTP protocol packets and dynamically creates the GRE connections and xlates necessary to permit PPTP traffic. Only Version 1, as defined in RFC 2637, is supported.

PAT is only performed for the modified version of GRE [RFC 2637] when negotiated over the PPTP TCP control channel. Port Address Translation is *not* performed for the unmodified version of GRE [RFC 1701, RFC 1702].

Specifically, the ASA inspects the PPTP version announcements and the outgoing call request/response sequence. Only PPTP Version 1, as defined in RFC 2637, is inspected. Further inspection on the TCP control channel is disabled if the version announced by either side is not Version 1. In addition, the outgoing-call request and reply sequence are tracked. Connections and xlates are dynamic allocated as necessary to permit subsequent secondary GRE data traffic.

The PPTP inspection engine must be enabled for PPTP traffic to be translated by PAT. Additionally, PAT is only performed for a modified version of GRE (RFC2637) and only if it is negotiated over the PPTP TCP control channel. PAT is not performed for the unmodified version of GRE (RFC 1701 and RFC 1702).

As described in RFC 2637, the PPTP protocol is mainly used for the tunneling of PPP sessions initiated from a modem bank PAC (PPTP Access Concentrator) to the headend PNS (PPTP Network Server). When used this way, the PAC is the remote client and the PNS is the server.

However, when used for VPN by Windows, the interaction is inverted. The PNS is a remote single-user PC that initiates connection to the head-end PAC to gain access to a central network.

SMTP and Extended SMTP Inspection

This section describes the IM inspection engine. This section includes the following topics:

- SMTP and ESMTP Inspection Overview, page 10-32
- Configuring an ESMTP Inspection Policy Map for Additional Inspection Control, page 10-33

SMTP and ESMTP Inspection Overview

ESMTP application inspection provides improved protection against SMTP-based attacks by restricting the types of SMTP commands that can pass through the ASA and by adding monitoring capabilities.

ESMTP is an enhancement to the SMTP protocol and is similar is most respects to SMTP. For convenience, the term SMTP is used in this document to refer to both SMTP and ESMTP. The application inspection process for extended SMTP is similar to SMTP application inspection and includes support for SMTP sessions. Most commands used in an extended SMTP session are the same as those used in an SMTP session but an ESMTP session is considerably faster and offers more options related to reliability and security, such as delivery status notification.

Extended SMTP application inspection adds support for these extended SMTP commands, including AUTH, EHLO, ETRN, HELP, SAML, SEND, SOML, STARTTLS, and VRFY. Along with the support for seven RFC 821 commands (DATA, HELO, MAIL, NOOP, QUIT, RCPT, RSET), the ASA supports a total of fifteen SMTP commands.

Other extended SMTP commands, such as ATRN, ONEX, VERB, CHUNKING, and private extensions and are not supported. Unsupported commands are translated into Xs, which are rejected by the internal server. This results in a message such as "500 Command unknown: 'XXX'." Incomplete commands are discarded.

The ESMTP inspection engine changes the characters in the server SMTP banner to asterisks except for the "2", "0", "0" characters. Carriage return (CR) and linefeed (LF) characters are ignored.

With SMTP inspection enabled, a Telnet session used for interactive SMTP may hang if the following rules are not observed: SMTP commands must be at least four characters in length; must be terminated with carriage return and line feed; and must wait for a response before issuing the next reply.

An SMTP server responds to client requests with numeric reply codes and optional human-readable strings. SMTP application inspection controls and reduces the commands that the user can use as well as the messages that the server returns. SMTP inspection performs three primary tasks:

- Restricts SMTP requests to seven basic SMTP commands and eight extended commands.
- Monitors the SMTP command-response sequence.
- Generates an audit trail—Audit record 108002 is generated when invalid character embedded in the mail address is replaced. For more information, see RFC 821.

SMTP inspection monitors the command and response sequence for the following anomalous signatures:

- Truncated commands.
- Incorrect command termination (not terminated with <CR><LR>).
- The MAIL and RCPT commands specify who are the sender and the receiver of the mail. Mail addresses are scanned for strange characters. The pipeline character (|) is deleted (changed to a blank space) and "<" ,">" are only allowed if they are used to define a mail address (">" must be preceded by "<"). To close the session when the PIPE character is found as a parameter to a MAIL from or RCPT to command, include the **special-character** command in the configuration as part of the inspection parameters (**parameters** command).
- Unexpected transition by the SMTP server.
- For unknown commands, the ASA changes all the characters in the packet to X. In this case, the server generates an error code to the client. Because of the change in the packed, the TCP checksum has to be recalculated or adjusted.
- TCP stream editing.
- Command pipelining.

Configuring an ESMTP Inspection Policy Map for Additional Inspection Control

ESMTP inspection detects attacks, including spam, phising, malformed message attacks, buffer overflow/underflow attacks. It also provides support for application security and protocol conformance, which enforce the sanity of the ESMTP messages as well as detect several attacks, block senders/receivers, and block mail relay.

To specify actions when a message violates a parameter, create an ESMTP inspection policy map. You can then apply the inspection policy map when you enable ESMTP inspection.

To create an ESMTP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the general operations configuration guide. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the general operations configuration guide.

Step 3 Create an ESMTP inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect esmtp policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 4 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 5 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the ESMTP class map that you created in Step 3 by entering the following command:

```
ciscoasa(config-pmap)# class class_map_name
ciscoasa(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
ciscoasa(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the command reference for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 2-4.

Step 6 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

b. To configure a local domain name, enter the following command:

```
ciscoasa(config-pmap-p)# mail-relay domain-name action [drop-connection | log]]
```

Where the **drop-connection** action closes the connection. The **log** action sends a system log message when this policy map matches traffic.

c. To enforce banner obfuscation, enter the following command:

```
ciscoasa(config-pmap-p)# mask-banner
```

The following example shows how to define an ESMTP inspection policy map.

```
ciscoasa(config)# regex user1 "user1@cisco.com"
ciscoasa(config)# regex user2 "user2@cisco.com"
ciscoasa(config)# regex user3 "user3@cisco.com"
ciscoasa(config)# class-map type regex senders_black_list
ciscoasa(config-cmap)# description "Regular expressions to filter out undesired senders"
ciscoasa(config-cmap)# match regex user1
ciscoasa(config-cmap)# match regex user2
ciscoasa(config-cmap)# match regex user3

ciscoasa(config)# policy-map type inspect esmtp advanced_esmtp_map
ciscoasa(config-pmap)# match sender-address regex class senders_black_list
ciscoasa(config-pmap-c)# drop-connection log

ciscoasa(config)# policy-map outside_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect esmtp advanced_esmtp_map

ciscoasa(config)# service-policy outside_policy interface outside
```

TFTP Inspection

TFTP inspection is enabled by default.

TFTP, described in RFC 1350, is a simple protocol to read and write files between a TFTP server and client.

The ASA inspects TFTP traffic and dynamically creates connections and translations, if necessary, to permit file transfer between a TFTP client and server. Specifically, the inspection engine inspects TFTP read request (RRQ), write request (WRQ), and error notification (ERROR).

A dynamic secondary channel and a PAT translation, if necessary, are allocated on a reception of a valid read (RRQ) or write (WRQ) request. This secondary channel is subsequently used by TFTP for file transfer or error notification.

Only the TFTP server can initiate traffic over the secondary channel, and at most one incomplete secondary channel can exist between the TFTP client and server. An error notification from the server closes the secondary channel.

TFTP inspection must be enabled if static PAT is used to redirect TFTP traffic.

Configuring Inspection for Voice and Video Protocols

This chapter describes how to configure application layer protocol inspection. Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the ASA to do a deep packet inspection instead of passing the packet through the fast path. As a result, inspection engines can affect overall throughput.

Several common inspection engines are enabled on the ASA by default, but you might need to enable others depending on your network.

This chapter includes the following sections:

- CTIQBE Inspection, page 11-1
- H.323 Inspection, page 11-3
- MGCP Inspection, page 11-11
- RTSP Inspection, page 11-14
- SIP Inspection, page 11-18
- Skinny (SCCP) Inspection, page 11-24

CTIQBE Inspection

This section describes CTIQBE application inspection. This section includes the following topics:

- CTIQBE Inspection Overview, page 11-1
- Limitations and Restrictions, page 11-2
- Verifying and Monitoring CTIQBE Inspection, page 11-2

CTIQBE Inspection Overview

CTIQBE protocol inspection supports NAT, PAT, and bidirectional NAT. This enables Cisco IP SoftPhone and other Cisco TAPI/JTAPI applications to work successfully with Cisco CallManager for call setup across the ASA.

TAPI and JTAPI are used by many Cisco VoIP applications. CTIQBE is used by Cisco TSP to communicate with Cisco CallManager.

Limitations and Restrictions

The following summarizes limitations that apply when using CTIQBE application inspection:

- CTIQBE application inspection does not support configurations with the **alias** command.
- Stateful failover of CTIQBE calls is not supported.
- Entering the **debug ctiqbe** command may delay message transmission, which may have a performance impact in a real-time environment. When you enable this debugging or logging and Cisco IP SoftPhone seems unable to complete call setup through the ASA, increase the timeout values in the Cisco TSP settings on the system running Cisco IP SoftPhone.

The following summarizes special considerations when using CTIQBE application inspection in specific scenarios:

- If two Cisco IP SoftPhones are registered with different Cisco CallManagers, which are connected to different interfaces of the ASA, calls between these two phones fails.
- When Cisco CallManager is located on the higher security interface compared to Cisco IP SoftPhones, if NAT or outside NAT is required for the Cisco CallManager IP address, the mapping must be static as Cisco IP SoftPhone requires the Cisco CallManager IP address to be specified explicitly in its Cisco TSP configuration on the PC.
- When using PAT or Outside PAT, if the Cisco CallManager IP address is to be translated, its TCP port 2748 must be statically mapped to the same port of the PAT (interface) address for Cisco IP SoftPhone registrations to succeed. The CTIQBE listening port (TCP 2748) is fixed and is not user-configurable on Cisco CallManager, Cisco IP SoftPhone, or Cisco TSP.

Verifying and Monitoring CTIQBE Inspection

The **show ctiqbe** command displays information regarding the CTIQBE sessions established across the ASA. It shows information about the media connections allocated by the CTIQBE inspection engine.

The following is sample output from the **show ctiqbe** command under the following conditions. There is only one active CTIQBE session setup across the ASA. It is established between an internal CTI device (for example, a Cisco IP SoftPhone) at local address 10.0.0.99 and an external Cisco CallManager at 172.29.1.77, where TCP port 2748 is the Cisco CallManager. The heartbeat interval for the session is 120 seconds.

```
ciscoasa# # show ctiqbe

Total: 1
        LOCAL           FOREIGN         STATE   HEARTBEAT
---------------------------------------------------------------
1       10.0.0.99/1117  172.29.1.77/2748        1       120
        ----------------------------------------------
        RTP/RTCP: PAT xlates: mapped to 172.29.1.99(1028 - 1029)
        ----------------------------------------------
        MEDIA: Device ID 27     Call ID 0
               Foreign 172.29.1.99      (1028 - 1029)
               Local   172.29.1.88      (26822 - 26823)
        ----------------------------------------------
```

The CTI device has already registered with the CallManager. The device internal address and RTP listening port is PATed to 172.29.1.99 UDP port 1028. Its RTCP listening port is PATed to UDP 1029.

The line beginning with `RTP/RTCP: PAT xlates:` appears only if an internal CTI device has registered with an external CallManager and the CTI device address and ports are PATed to that external interface. This line does not appear if the CallManager is located on an internal interface, or if the internal CTI device address and ports are translated to the same external interface that is used by the CallManager.

The output indicates a call has been established between this CTI device and another phone at 172.29.1.88. The RTP and RTCP listening ports of the other phone are UDP 26822 and 26823. The other phone locates on the same interface as the CallManager because the ASA does not maintain a CTIQBE session record associated with the second phone and CallManager. The active call leg on the CTI device side can be identified with Device ID 27 and Call ID 0.

The following is sample output from the **show xlate debug** command for these CTIBQE connections:

```
ciscoasa# show xlate debug
3 in use, 3 most used
Flags:  D - DNS, d - dump, I - identity, i - inside, n - no random,
        r - portmap, s - static
TCP PAT from inside:10.0.0.99/1117 to outside:172.29.1.99/1025 flags ri idle 0:00:22
timeout 0:00:30
UDP PAT from inside:10.0.0.99/16908 to outside:172.29.1.99/1028 flags ri idle 0:00:00
timeout 0:04:10
UDP PAT from inside:10.0.0.99/16909 to outside:172.29.1.99/1029 flags ri idle 0:00:23
timeout 0:04:10
```

The **show conn state ctiqbe** command displays the status of CTIQBE connections. In the output, the media connections allocated by the CTIQBE inspection engine are denoted by a ‘C’ flag. The following is sample output from the **show conn state ctiqbe** command:

```
ciscoasa# show conn state ctiqbe
1 in use, 10 most used
ciscoasa# show conn state ctiqbe detail
1 in use, 10 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       B - initial SYN from outside, C - CTIQBE media, D - DNS, d - dump,
       E - outside back connection, F - outside FIN, f - inside FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
       i - incomplete, J - GTP, j - GTP data, k - Skinny media,
       M - SMTP data, m - SIP media, O - outbound data, P - inside back connection,
       q - SQL*Net data, R - outside acknowledged FIN,
       R - UDP RPC, r - inside acknowledged FIN, S - awaiting inside SYN,
       s - awaiting outside SYN, T - SIP, t - SIP transient, U - up
```

H.323 Inspection

This section describes the H.323 application inspection. This section includes the following topics:

- H.323 Inspection Overview, page 11-4
- How H.323 Works, page 11-4
- H.239 Support in H.245 Messages, page 11-5
- Limitations and Restrictions, page 11-5
- Configuring an H.323 Inspection Policy Map for Additional Inspection Control, page 11-6
- Configuring H.323 and H.225 Timeout Values, page 11-9
- Verifying and Monitoring H.323 Inspection, page 11-9

H.323 Inspection Overview

H.323 inspection provides support for H.323 compliant applications such as Cisco CallManager and VocalTec Gatekeeper. H.323 is a suite of protocols defined by the International Telecommunication Union for multimedia conferences over LANs. The ASA supports H.323 through Version 6, including H.323 v3 feature Multiple Calls on One Call Signaling Channel.

With H.323 inspection enabled, the ASA supports multiple calls on the same call signaling channel, a feature introduced with H.323 Version 3. This feature reduces call setup time and reduces the use of ports on the ASA.

The two major functions of H.323 inspection are as follows:

- NAT the necessary embedded IPv4 addresses in the H.225 and H.245 messages. Because H.323 messages are encoded in PER encoding format, the ASA uses an ASN.1 decoder to decode the H.323 messages.
- Dynamically allocate the negotiated H.245 and RTP/RTCP connections.

How H.323 Works

The H.323 collection of protocols collectively may use up to two TCP connection and four to eight UDP connections. FastConnect uses only one TCP connection, and RAS uses a single UDP connection for registration, admissions, and status.

An H.323 client can initially establish a TCP connection to an H.323 server using TCP port 1720 to request Q.931 call setup. As part of the call setup process, the H.323 terminal supplies a port number to the client to use for an H.245 TCP connection. In environments where H.323 gatekeeper is in use, the initial packet is transmitted using UDP.

H.323 inspection monitors the Q.931 TCP connection to determine the H.245 port number. If the H.323 terminals are not using FastConnect, the ASA dynamically allocates the H.245 connection based on the inspection of the H.225 messages.

Note The H.225 connection can also be dynamically allocated when using RAS.

Within each H.245 message, the H.323 endpoints exchange port numbers that are used for subsequent UDP data streams. H.323 inspection inspects the H.245 messages to identify these ports and dynamically creates connections for the media exchange. RTP uses the negotiated port number, while RTCP uses the next higher port number.

The H.323 control channel handles H.225 and H.245 and H.323 RAS. H.323 inspection uses the following ports.

- 1718—Gate Keeper Discovery UDP port
- 1719—RAS UDP port
- 1720—TCP Control Port

You must permit traffic for the well-known H.323 port 1719 for RAS signaling. Additionally, you must permit traffic for the well-known H.323 port 1720 for the H.225 call signaling; however, the H.245 signaling ports are negotiated between the endpoints in the H.225 signaling. When an H.323 gatekeeper is used, the ASA opens an H.225 connection based on inspection of the ACF and RCF nmessages.

After inspecting the H.225 messages, the ASA opens the H.245 channel and then inspects traffic sent over the H.245 channel as well. All H.245 messages passing through the ASA undergo H.245 application inspection, which translates embedded IP addresses and opens the media channels negotiated in H.245 messages.

The H.323 ITU standard requires that a TPKT header, defining the length of the message, precede the H.225 and H.245, before being passed on to the reliable connection. Because the TPKT header does not necessarily need to be sent in the same TCP packet as H.225 and H.245 messages, the ASA must remember the TPKT length to process and decode the messages properly. For each connection, the ASA keeps a record that contains the TPKT length for the next expected message.

If the ASA needs to perform NAT on IP addresses in messages, it changes the checksum, the UUIE length, and the TPKT, if it is included in the TCP packet with the H.225 message. If the TPKT is sent in a separate TCP packet, the ASA proxy ACKs that TPKT and appends a new TPKT to the H.245 message with the new length.

Note The ASA does not support TCP options in the Proxy ACK for the TPKT.

Each UDP connection with a packet going through H.323 inspection is marked as an H.323 connection and times out with the H.323 timeout as configured with the **timeout** command.

Note You can enable call setup between H.323 endpoints when the Gatekeeper is inside the network. The ASA includes options to open pinholes for calls based on the RegistrationRequest/RegistrationConfirm (RRQ/RCF) messages. Because these RRQ/RCF messages are sent to and from the Gatekeeper, the calling endpoint's IP address is unknown and the ASA opens a pinhole through source IP address/port 0/0. By default, this option is disabled. To enable call setup between H.323 endpoint, enter the **ras-rcf-pinholes enable** command during parameter configuration mode while creating an H.323 Inspection policy map. See Configuring an H.323 Inspection Policy Map for Additional Inspection Control, page 11-6.

H.239 Support in H.245 Messages

The ASA sits between two H.323 endpoints. When the two H.323 endpoints set up a telepresentation session so that the endpoints can send and receive a data presentation, such as spreadsheet data, the ASA ensure successful H.239 negotiation between the endpoints.

H.239 is a standar that provides the ability for H.300 series endpoints to open an additional video channel in a single call. In a call, an endpoint (such as a video phone), sends a channel for video and a channel for data presentation. The H.239 negotiation occurs on the H.245 channel.

The ASA opens pinholes for the additional media channel and the media control channel. The endpoints use open logical channel message (OLC) to signal a new channel creation. The message extension is part of H.245 version 13.

The decoding and encoding of of the telepresentation session is enabled by default. H.239 encoding and decoding is preformed by ASN.1 coder.

Limitations and Restrictions

The following are some of the known issues and limitations when using H.323 application inspection:

- Only static NAT is fully supported. Static PAT may not properly translate IP addresses embedded in optional fields within H.323 messages. If you experience this kind of problem, do not use static PAT with H.323.
- Not supported with dynamic NAT or PAT.
- Not supported with extended PAT.
- Not supported with NAT between same-security-level interfaces.
- Not supported with outside NAT.
- Not supported with NAT64.
- When a NetMeeting client registers with an H.323 gatekeeper and tries to call an H.323 gateway that is also registered with the H.323 gatekeeper, the connection is established but no voice is heard in either direction. This problem is unrelated to the ASA.
- If you configure a network static address where the network static address is the same as a third-party netmask and address, then any outbound H.323 connection fails.

Configuring an H.323 Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an H.323 inspection policy map. You can then apply the inspection policy map when you enable H.323 inspection.

To create an H.323 inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the general operations configuration guide. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the general operations configuration guide.

Step 3 (Optional) Create an H.323 inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string “example.com,” then any traffic that includes “example.com” does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop-connection, reset, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
ciscoasa(config)# class-map type inspect h323 [match-all | match-any] class_map_name
ciscoasa(config-cmap)#
```

Where *the class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
ciscoasa(config-cmap)# description string
```

Where *string* is the description of the class map (up to 200 characters).

c. (Optional) To match a called party, enter the following command:

```
ciscoasa(config-cmap)# match [not] called-party regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

d. (Optional) To match a media type, enter the following command:

```
ciscoasa(config-cmap)# match [not] media-type {audio | data | video}
```

Step 4 Create an H.323 inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect h323 policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the H.323 class map that you created in Step 3 by entering the following command:

```
ciscoasa(config-pmap)# class class_map_name
ciscoasa(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
ciscoasa(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the command reference for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 2-4.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

b. To enable call setup betweeen H.323 Endpoings, enter the following command:

```
ciscoasa(config)# ras-rcf-pinholes enable
```

You can enable call setup between H.323 endpoints when the Gatekeeper is inside the network. The ASA includes options to open pinholes for calls based on the RegistrationRequest/RegistrationConfirm (RRQ/RCF) messages. Because these RRQ/RCF messages are sent to and from the Gatekeeper, the calling endpoint's IP address is unknown and the ASA opens a pinhole through source IP address/port 0/0. By default, this option is disabled.

c. To define the H.323 call duration limit, enter the following command:

```
ciscoasa(config-pmap-p)# call-duration-limit time
```

Where *time* is the call duration limit in seconds. Range is from 0:0:0 ti 1163:0;0. A value of 0 means never timeout.

d. To enforce call party number used in call setup, enter the following command:

```
ciscoasa(config-pmap-p)# call-party-number
```

e. To enforce H.245 tunnel blocking, enter the following command:

```
ciscoasa(config-pmap-p)# h245-tunnel-block action {drop-connection | log}
```

f. To define an hsi group and enter hsi group configuration mode, enter the following command:

```
ciscoasa(config-pmap-p)# hsi-group id
```

Where *id* is the hsi group ID. Range is from 0 to 2147483647.

To add an hsi to the hsi group, enter the following command in hsi group configuration mode:

```
ciscoasa(config-h225-map-hsi-grp)# hsi ip_address
```

Where *ip_address* is the host to add. A maximum of five hosts per hsi group are allowed.

To add an endpoint to the hsi group, enter the following command in hsi group configuration mode:

```
ciscoasa(config-h225-map-hsi-grp)# endpoint ip_address if_name
```

Where *ip_address* is the endpoint to add and *if_name* is the interface through which the endpoint is connected to the security appliance. A maximum of ten endpoints per hsi group are allowed.

g. To check RTP packets flowing on the pinholes for protocol conformance, enter the following command:

```
ciscoasa(config-pmap-p)# rtp-conformance [enforce-payloadtype]
```

Where the **enforce-payloadtype** keyword enforces the payload type to be audio or video based on the signaling exchange.

h. To enable state checking validation, enter the following command:

```
ciscoasa(config-pmap-p)# state-checking {h225 | ras}
```

The following example shows how to configure phone number filtering:

```
ciscoasa(config)# regex caller 1 "5551234567"
ciscoasa(config)# regex caller 2 "5552345678"
ciscoasa(config)# regex caller 3 "5553456789"

ciscoasa(config)# class-map type inspect h323 match-all h323_traffic
ciscoasa(config-pmap-c)# match called-party regex caller1
ciscoasa(config-pmap-c)# match calling-party regex caller2

ciscoasa(config)# policy-map type inspect h323 h323_map
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# class h323_traffic
ciscoasa(config-pmap-c)# drop
```

Configuring H.323 and H.225 Timeout Values

To configure the idle time after which an H.225 signalling connection is closed, use the **timeout h225** command. The default for H.225 timeout is one hour.

To configure the idle time after which an H.323 control connection is closed, use the **timeout h323** command. The default is five minutes.

Verifying and Monitoring H.323 Inspection

This section describes how to display information about H.323 sessions. This section includes the following topics:

- Monitoring H.225 Sessions, page 11-9
- Monitoring H.245 Sessions, page 11-10
- Monitoring H.323 RAS Sessions, page 11-10

Monitoring H.225 Sessions

The **show h225** command displays information for H.225 sessions established across the ASA. Along with the **debug h323 h225 event**, **debug h323 h245 event**, and **show local-host** commands, this command is used for troubleshooting H.323 inspection engine issues.

Before entering the **show h225**, **show h245**, or **show h323-ras** commands, we recommend that you configure the **pager** command. If there are a lot of session records and the **pager** command is not configured, it may take a while for the **show** command output to reach its end. If there is an abnormally large number of connections, check that the sessions are timing out based on the default timeout values or the values set by you. If they are not, then there is a problem that needs to be investigated.

The following is sample output from the **show h225** command:

```
ciscoasa# show h225
Total H.323 Calls: 1
1 Concurrent Call(s) for
    Local:   10.130.56.3/1040   Foreign: 172.30.254.203/1720
    1. CRV 9861
    Local:   10.130.56.3/1040   Foreign: 172.30.254.203/1720
```

```
0 Concurrent Call(s) for
    Local:   10.130.56.4/1050   Foreign: 172.30.254.205/1720
```

This output indicates that there is currently 1 active H.323 call going through the ASA between the local endpoint 10.130.56.3 and foreign host 172.30.254.203, and for these particular endpoints, there is 1 concurrent call between them, with a CRV for that call of 9861.

For the local endpoint 10.130.56.4 and foreign host 172.30.254.205, there are 0 concurrent calls. This means that there is no active call between the endpoints even though the H.225 session still exists. This could happen if, at the time of the **show h225** command, the call has already ended but the H.225 session has not yet been deleted. Alternately, it could mean that the two endpoints still have a TCP connection opened between them because they set "maintainConnection" to TRUE, so the session is kept open until they set it to FALSE again, or until the session times out based on the H.225 timeout value in your configuration.

Monitoring H.245 Sessions

The **show h245** command displays information for H.245 sessions established across the ASA by endpoints using slow start. Slow start is when the two endpoints of a call open another TCP control channel for H.245. Fast start is where the H.245 messages are exchanged as part of the H.225 messages on the H.225 control channel.) Along with the **debug h323 h245 event**, **debug h323 h225 event**, and **show local-host** commands, this command is used for troubleshooting H.323 inspection engine issues.

The following is sample output from the **show h245** command:

```
ciscoasa# show h245
Total: 1
        LOCAL           TPKT    FOREIGN         TPKT
1       10.130.56.3/1041        0       172.30.254.203/1245    0
        MEDIA: LCN 258 Foreign 172.30.254.203 RTP 49608 RTCP 49609
                      Local   10.130.56.3 RTP 49608 RTCP 49609
        MEDIA: LCN 259 Foreign 172.30.254.203 RTP 49606 RTCP 49607
                      Local   10.130.56.3 RTP 49606 RTCP 49607
```

There is currently one H.245 control session active across the ASA. The local endpoint is 10.130.56.3, and we are expecting the next packet from this endpoint to have a TPKT header because the TPKT value is 0. The TKTP header is a 4-byte header preceding each H.225/H.245 message. It gives the length of the message, including the 4-byte header. The foreign host endpoint is 172.30.254.203, and we are expecting the next packet from this endpoint to have a TPKT header because the TPKT value is 0.

The media negotiated between these endpoints have an LCN of 258 with the foreign RTP IP address/port pair of 172.30.254.203/49608 and an RTCP IP address/port of 172.30.254.203/49609 with a local RTP IP address/port pair of 10.130.56.3/49608 and an RTCP port of 49609.

The second LCN of 259 has a foreign RTP IP address/port pair of 172.30.254.203/49606 and an RTCP IP address/port pair of 172.30.254.203/49607 with a local RTP IP address/port pair of 10.130.56.3/49606 and RTCP port of 49607.

Monitoring H.323 RAS Sessions

The **show h323-ras** command displays information for H.323 RAS sessions established across the ASA between a gatekeeper and its H.323 endpoint. Along with the **debug h323 ras event** and **show local-host** commands, this command is used for troubleshooting H.323 RAS inspection engine issues.

The **show h323-ras** command displays connection information for troubleshooting H.323 inspection engine issues. The following is sample output from the **show h323-ras** command:

```
ciscoasa# show h323-ras
```

```
Total: 1
        GK                       Caller
        172.30.254.214 10.130.56.14
```

This output shows that there is one active registration between the gatekeeper 172.30.254.214 and its client 10.130.56.14.

MGCP Inspection

This section describes MGCP application inspection. This section includes the following topics:

- MGCP Inspection Overview, page 11-11
- Configuring an MGCP Inspection Policy Map for Additional Inspection Control, page 11-12
- Configuring MGCP Timeout Values, page 11-13
- Verifying and Monitoring MGCP Inspection, page 11-14

MGCP Inspection Overview

MGCP is a master/slave protocol used to control media gateways from external call control elements called media gateway controllers or call agents. A media gateway is typically a network element that provides conversion between the audio signals carried on telephone circuits and data packets carried over the Internet or over other packet networks. Using NAT and PAT with MGCP lets you support a large number of devices on an internal network with a limited set of external (global) addresses. Examples of media gateways are:

- Trunking gateways, that interface between the telephone network and a Voice over IP network. Such gateways typically manage a large number of digital circuits.
- Residential gateways, that provide a traditional analog (RJ11) interface to a Voice over IP network. Examples of residential gateways include cable modem/cable set-top boxes, xDSL devices, broad-band wireless devices.
- Business gateways, that provide a traditional digital PBX interface or an integrated soft PBX interface to a Voice over IP network.

Note To avoid policy failure when upgrading from ASA version 7.1, all layer 7 and layer 3 policies must have distinct names. For instance, a previously configured policy map with the same name as a previously configured MGCP map must be changed before the upgrade.

MGCP messages are transmitted over UDP. A response is sent back to the source address (IP address and UDP port number) of the command, but the response may not arrive from the same address as the command was sent to. This can happen when multiple call agents are being used in a failover configuration and the call agent that received the command has passed control to a backup call agent, which then sends the response.

MGCP endpoints are physical or virtual sources and destinations for data. Media gateways contain endpoints on which the call agent can create, modify and delete connections to establish and control media sessions with other multimedia endpoints. Also, the call agent can instruct the endpoints to detect certain events and generate signals. The endpoints automatically communicate changes in service state to the call agent.

MGCP transactions are composed of a command and a mandatory response. There are eight types of commands:

- CreateConnection
- ModifyConnection
- DeleteConnection
- NotificationRequest
- Notify
- AuditEndpoint
- AuditConnection
- RestartInProgress

The first four commands are sent by the call agent to the gateway. The Notify command is sent by the gateway to the call agent. The gateway may also send a DeleteConnection. The registration of the MGCP gateway with the call agent is achieved by the RestartInProgress command. The AuditEndpoint and the AuditConnection commands are sent by the call agent to the gateway.

All commands are composed of a Command header, optionally followed by a session description. All responses are composed of a Response header, optionally followed by a session description.

- The port on which the gateway receives commands from the call agent. Gateways usually listen to UDP port 2427.
- The port on which the call agent receives commands from the gateway. Call agents usually listen to UDP port 2727.

Note MGCP inspection does not support the use of different IP addresses for MGCP signaling and RTP data. A common and recommended practice is to send RTP data from a resilient IP address, such as a loopback or virtual IP address; however, the ASA requires the RTP data to come from the same address as MGCP signalling.

Configuring an MGCP Inspection Policy Map for Additional Inspection Control

If the network has multiple call agents and gateways for which the ASA has to open pinholes, create an MGCP map. You can then apply the MGCP map when you enable MGCP inspection.

To create an MGCP map, perform the following steps:

Step 1 To create an MGCP inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect mgcp map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 2 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 3 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

b. To configure the call agents, enter the following command for each call agent:

```
ciscoasa(config-pmap-p)# call-agent ip_address group_id
```

Use the **call-agent** command to specify a group of call agents that can manage one or more gateways. The call agent group information is used to open connections for the call agents in the group (other than the one a gateway sends a command to) so that any of the call agents can send the response. call agents with the same *group_id* belong to the same group. A call agent may belong to more than one group. The *group_id* option is a number from 0 to 4294967295. The *ip_address* option specifies the IP address of the call agent.

Note MGCP call agents send AUEP messages to determine if MGCP end points are present. This establishes a flow through the ASA and allows MGCP end points to register with the call agent.

c. To configure the gateways, enter the following command for each gateway:

```
ciscoasa(config-pmap-p)# gateway ip_address group_id
```

Use the **gateway** command to specify which group of call agents are managing a particular gateway. The IP address of the gateway is specified with the *ip_address* option. The *group_id* option is a number from 0 to 4294967295 that must correspond with the *group_id* of the call agents that are managing the gateway. A gateway may only belong to one group.

d. If you want to change the maximum number of commands allowed in the MGCP command queue, enter the following command:

```
ciscoasa(config-pmap-p)# command-queue command_limit
```

The following example shows how to define an MGCP map:

```
ciscoasa(config)# policy-map type inspect mgcp sample_map
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# call-agent 10.10.11.5 101
ciscoasa(config-pmap-p)# call-agent 10.10.11.6 101
ciscoasa(config-pmap-p)# call-agent 10.10.11.7 102
ciscoasa(config-pmap-p)# call-agent 10.10.11.8 102
ciscoasa(config-pmap-p)# gateway 10.10.10.115 101
ciscoasa(config-pmap-p)# gateway 10.10.10.116 102
ciscoasa(config-pmap-p)# gateway 10.10.10.117 102
ciscoasa(config-pmap-p)# command-queue 150
```

Configuring MGCP Timeout Values

The **timeout mgcp command** lets you set the interval for inactivity after which an MGCP media connection is closed. The default is 5 minutes.

The **timeout mgcp-pat** command lets you set the timeout for PAT xlates. Because MGCP does not have a keepalive mechanism, if you use non-Cisco MGCP gateways (call agents), the PAT xlates are torn down after the default timeout interval, which is 30 seconds.

Verifying and Monitoring MGCP Inspection

The **show mgcp commands** command lists the number of MGCP commands in the command queue. The **show mgcp sessions** command lists the number of existing MGCP sessions. The **detail** option includes additional information about each command (or session) in the output. The following is sample output from the **show mgcp commands** command:

```
ciscoasa# show mgcp commands
1 in use, 1 most used, 200 maximum allowed
CRCX, gateway IP: host-pc-2, transaction ID: 2052, idle: 0:00:07
```

The following is sample output from the **show mgcp detail** command.

```
ciscoasa# show mgcp commands detail
1 in use, 1 most used, 200 maximum allowed
CRCX, idle: 0:00:10
        Gateway IP      host-pc-2
        Transaction ID  2052
        Endpoint name   aaln/1
        Call ID         9876543210abcdef
        Connection ID
        Media IP        192.168.5.7
        Media port      6058
```

The following is sample output from the **show mgcp sessions** command.

```
ciscoasa# show mgcp sessions
1 in use, 1 most used
Gateway IP host-pc-2, connection ID 6789af54c9, active 0:00:11
```

The following is sample output from the **show mgcp sessions detail** command.

```
ciscoasa# show mgcp sessions detail
1 in use, 1 most used
Session active 0:00:14
        Gateway IP      host-pc-2
        Call ID         9876543210abcdef
        Connection ID   6789af54c9
        Endpoint name   aaln/1
        Media lcl port  6166
        Media rmt IP    192.168.5.7
        Media rmt port  6058
```

RTSP Inspection

This section describes RTSP application inspection. This section includes the following topics:

- RTSP Inspection Overview, page 11-15
- Using RealPlayer, page 11-15
- Restrictions and Limitations, page 11-15
- Configuring an RTSP Inspection Policy Map for Additional Inspection Control, page 11-16

RTSP Inspection Overview

The RTSP inspection engine lets the ASA pass RTSP packets. RTSP is used by RealAudio, RealNetworks, Apple QuickTime 4, RealPlayer, and Cisco IP/TV connections.

Note For Cisco IP/TV, use RTSP TCP port 554 and TCP 8554.

RTSP applications use the well-known port 554 with TCP (rarely UDP) as a control channel. The ASA only supports TCP, in conformity with RFC 2326. This TCP control channel is used to negotiate the data channels that is used to transmit audio/video traffic, depending on the transport mode that is configured on the client.

The supported RDT transports are: rtp/avp, rtp/avp/udp, x-real-rdt, x-real-rdt/udp, and x-pn-tng/udp.

The ASA parses Setup response messages with a status code of 200. If the response message is travelling inbound, the server is outside relative to the ASA and dynamic channels need to be opened for connections coming inbound from the server. If the response message is outbound, then the ASA does not need to open dynamic channels.

Because RFC 2326 does not require that the client and server ports must be in the SETUP response message, the ASA keeps state and remembers the client ports in the SETUP message. QuickTime places the client ports in the SETUP message and then the server responds with only the server ports.

RTSP inspection does not support PAT or dual-NAT. Also, the ASA cannot recognize HTTP cloaking where RTSP messages are hidden in the HTTP messages.

Using RealPlayer

When using RealPlayer, it is important to properly configure transport mode. For the ASA, add an **access-list** command from the server to the client or vice versa. For RealPlayer, change transport mode by clicking **Options>Preferences>Transport>RTSP Settings**.

If using TCP mode on the RealPlayer, select the **Use TCP to Connect to Server** and **Attempt to use TCP for all content** check boxes. On the ASA, there is no need to configure the inspection engine.

If using UDP mode on the RealPlayer, select the **Use TCP to Connect to Server** and **Attempt to use UDP for static content** check boxes, and for live content not available via Multicast. On the ASA, add an **inspect rtsp** *port* command.

Restrictions and Limitations

The following restrictions apply to the RSTP inspection.

- The ASA does not support multicast RTSP or RTSP messages over UDP.
- The ASA does not have the ability to recognize HTTP cloaking where RTSP messages are hidden in the HTTP messages.
- The ASA cannot perform NAT on RTSP messages because the embedded IP addresses are contained in the SDP files as part of HTTP or RTSP messages. Packets could be fragmented and ASA cannot perform NAT on fragmented packets.
- With Cisco IP/TV, the number of translates the ASA performs on the SDP part of the message is proportional to the number of program listings in the Content Manager (each program listing can have at least six embedded IP addresses).

- You can configure NAT for Apple QuickTime 4 or RealPlayer. Cisco IP/TV only works with NAT if the Viewer and Content Manager are on the outside network and the server is on the inside network.

Configuring an RTSP Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an RTSP inspection policy map. You can then apply the inspection policy map when you enable RTSP inspection.

To create an RTSP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the general operations configuration guide. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the general operations configuration guide.

Step 3 (Optional) Create an RTSP inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop-connection and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
ciscoasa(config)# class-map type inspect rtsp [match-all | match-any] class_map_name
ciscoasa(config-cmap)#
```

Where *class_map_name* is the name of the class map. The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map. The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria. The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
ciscoasa(config-cmap)# description string
```

c. (Optional) To match an RTSP request method, enter the following command:

```
ciscoasa(config-cmap)# match [not] request-method method
```

Where *method* is the type of method to match (announce, describe, get_parameter, options, pause, play, record, redirect, setup, set_parameter, teardown).

d. (Optional) To match URL filtering, enter the following command:

```
ciscoasa(config-cmap)# match [not] url-filter regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

Step 4 To create an RTSP inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect rtsp policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the RTSP class map that you created in Step 3 by entering the following command:

```
ciscoasa(config-pmap)# class class_map_name
ciscoasa(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
ciscoasa(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the command reference for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 2-4.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

b. To restrict usage on reserve port for media negotiation, enter the following command:

```
ciscoasa(config-pmap-p)# reserve-port-protect
```

c. To set the limit on the URL length allowed in the message, enter the following command:

```
ciscoasa(config-pmap-p)# url-length-limit length
```

Where the *length* argument specifies the URL length in bytes (0 to 6000).

The following example shows a how to define an RTSP inspection policy map.

```
ciscoasa(config)# regex badurl1 www.url1.com/rtsp.avi
ciscoasa(config)# regex badurl2 www.url2.com/rtsp.rm
hostname(config)# regex badurl3 www.url3.com/rtsp.asp

ciscoasa(config)# class-map type regex match-any badurl-list
ciscoasa(config-cmap)# match regex badurl1
ciscoasa(config-cmap)# match regex badurl2
ciscoasa(config-cmap)# match regex badurl3

ciscoasa(config)# policy-map type inspect rtsp rtsp-filter-map
ciscoasa(config-pmap)# match url-filter regex class badurl-list
ciscoasa(config-pmap-p)# drop-connection

ciscoasa(config)# class-map rtsp-traffic-class
ciscoasa(config-cmap)# match default-inspection-traffic

ciscoasa(config)# policy-map rtsp-traffic-policy
ciscoasa(config-pmap)# class rtsp-traffic-class
ciscoasa(config-pmap-c)# inspect rtsp rtsp-filter-map

ciscoasa(config)# service-policy rtsp-traffic-policy global
```

SIP Inspection

This section describes SIP application inspection. This section includes the following topics:

- SIP Inspection Overview, page 11-18
- SIP Instant Messaging, page 11-19
- Configuring a SIP Inspection Policy Map for Additional Inspection Control, page 11-20
- Configuring SIP Timeout Values, page 11-24
- Verifying and Monitoring SIP Inspection, page 11-24

SIP Inspection Overview

SIP, as defined by the IETF, enables call handling sessions, particularly two-party audio conferences, or "calls." SIP works with SDP for call signalling. SDP specifies the ports for the media stream. Using SIP, the ASA can support any SIP VoIP gateways and VoIP proxy servers. SIP and SDP are defined in the following RFCs:

- SIP: Session Initiation Protocol, RFC 3261
- SDP: Session Description Protocol, RFC 2327

To support SIP calls through the ASA, signaling messages for the media connection addresses, media ports, and embryonic connections for the media must be inspected, because while the signaling is sent over a well-known destination port (UDP/TCP 5060), the media streams are dynamically allocated. Also, SIP embeds IP addresses in the user-data portion of the IP packet. SIP inspection applies NAT for these embedded IP addresses.

The following limitations and restrictions apply when using PAT with SIP:

- If a remote endpoint tries to register with a SIP proxy on a network protected by the ASA, the registration fails under very specific conditions, as follows:
  - PAT is configured for the remote endpoint.
  - The SIP registrar server is on the outside network.
  - The port is missing in the contact field in the REGISTER message sent by the endpoint to the proxy server.
  - Configuring static PAT is not supported with SIP inspection. If static PAT is configured for the Cisco Unified Communications Manager, SIP inspection cannot rewrite the SIP packet. Configure one-to-one static NAT for the Cisco Unified Communications Manager.
- If a SIP device transmits a packet in which the SDP portion has an IP address in the owner/creator field (o=) that is different than the IP address in the connection field (c=), the IP address in the o= field may not be properly translated. This is due to a limitation in the SIP protocol, which does not provide a port value in the o= field.
- When using PAT, any SIP header field which contains an internal IP address without a port might not be translated and hence the internal IP address will be leaked outside. If you want to avoid this leakage, configure NAT instead of PAT.

SIP Instant Messaging

Instant Messaging refers to the transfer of messages between users in near real-time. SIP supports the Chat feature on Windows XP using Windows Messenger RTC Client version 4.7.0105 only. The MESSAGE/INFO methods and 202 Accept response are used to support IM as defined in the following RFCs:

- Session Initiation Protocol (SIP)-Specific Event Notification, RFC 3265
- Session Initiation Protocol (SIP) Extension for Instant Messaging, RFC 3428

MESSAGE/INFO requests can come in at any time after registration/subscription. For example, two users can be online at any time, but not chat for hours. Therefore, the SIP inspection engine opens pinholes that time out according to the configured SIP timeout value. This value must be configured at least five minutes longer than the subscription duration. The subscription duration is defined in the Contact Expires value and is typically 30 minutes.

Because MESSAGE/INFO requests are typically sent using a dynamically allocated port other than port 5060, they are required to go through the SIP inspection engine.

Note Only the Chat feature is currently supported. Whiteboard, File Transfer, and Application Sharing are not supported. RTC Client 5.0 is not supported.

SIP inspection translates the SIP text-based messages, recalculates the content length for the SDP portion of the message, and recalculates the packet length and checksum. It dynamically opens media connections for ports specified in the SDP portion of the SIP message as address/ports on which the endpoint should listen.

SIP inspection has a database with indices CALL_ID/FROM/TO from the SIP payload. These indices identify the call, the source, and the destination. This database contains the media addresses and media ports found in the SDP media information fields and the media type. There can be multiple media addresses and ports for a session. The ASA opens RTP/RTCP connections between the two endpoints using these media addresses/ports.

The well-known port 5060 must be used on the initial call setup (INVITE) message; however, subsequent messages may not have this port number. The SIP inspection engine opens signaling connection pinholes, and marks these connections as SIP connections. This is done for the messages to reach the SIP application and be translated.

As a call is set up, the SIP session is in the "transient" state until the media address and media port is received from the called endpoint in a Response message indicating the RTP port the called endpoint listens on. If there is a failure to receive the response messages within one minute, the signaling connection is torn down.

Once the final handshake is made, the call state is moved to active and the signaling connection remains until a BYE message is received.

If an inside endpoint initiates a call to an outside endpoint, a media hole is opened to the outside interface to allow RTP/RTCP UDP packets to flow to the inside endpoint media address and media port specified in the INVITE message from the inside endpoint. Unsolicited RTP/RTCP UDP packets to an inside interface does not traverse the ASA, unless the ASA configuration specifically allows it.

Configuring a SIP Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create a SIP inspection policy map. You can then apply the inspection policy map when you enable SIP inspection.

To create a SIP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the general operations configuration guide. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the general operations configuration guide.

Step 3 (Optional) Create a SIP inspection class map by performing the following steps.

A class map groups multiple traffic matches. Traffic must match *all* of the **match** commands to match the class map. You can alternatively identify **match** commands directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that the class map lets you create more complex match criteria, and you can reuse class maps.

To specify traffic that should not match the class map, use the **match not** command. For example, if the **match not** command specifies the string "example.com," then any traffic that includes "example.com" does not match the class map.

For the traffic that you identify in this class map, you can specify actions such as drop-connection, reset, and/or log the connection in the inspection policy map.

If you want to perform different actions for each **match** command, you should identify the traffic directly in the policy map.

a. Create the class map by entering the following command:

```
ciscoasa(config)# class-map type inspect sip [match-all | match-any] class_map_name
ciscoasa(config-cmap)#
```

Where *the class_map_name* is the name of the class map. The match-all keyword is the default, and specifies that traffic must match all criteria to match the class map. The match-any keyword specifies that the traffic matches the class map if it matches at leX(The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

b. (Optional) To add a description to the class map, enter the following command:

```
ciscoasa(config-cmap)# description string
```

Where *string* is the description of the class map (up to 200 characters).

c. (Optional) To match a called party, as specified in the To header, enter the following command:

```
ciscoasa(config-cmap)# match [not] called-party regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

d. (Optional) To match a calling party, as specified in the From header, enter the following command:

```
ciscoasa(config-cmap)# match [not] calling-party regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

e. (Optional) To match a content length in the SIP header, enter the following command:

```
ciscoasa(config-cmap)# match [not] content length gt length
```

Where *length* is the number of bytes the content length is greater than. 0 to 65536.

f. (Optional) To match an SDP content type or regular expression, enter the following command:

```
ciscoasa(config-cmap)# match [not] content type {sdp | regex {class class_name | 
regex_name}}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

g. (Optional) To match a SIP IM subscriber, enter the following command:

```
ciscoasa(config-cmap)# match [not] im-subscriber regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

h. (Optional) To match a SIP via header, enter the following command:

```
ciscoasa(config-cmap)# match [not] message-path regex {class class_name | regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

i. (Optional) To match a SIP request method, enter the following command:

```
ciscoasa(config-cmap)# match [not] request-method method
```

Where *method* is the type of method to match (ack, bye, cancel, info, invite, message, notify, options, prack, refer, register, subscribe, unknown, update).

j. (Optional) To match the requester of a third-party registration, enter the following command:

```
ciscoasa(config-cmap)# match [not] third-party-registration regex {class class_name | 
regex_name}
```

Where the **regex** *regex_name* argument is the regular expression you created in Step 1. The **class** *regex_class_name* is the regular expression class map you created in Step 2.

k. (Optional) To match an URI in the SIP headers, enter the following command:

```
ciscoasa(config-cmap)# match [not] uri {sip | tel} length gt length
```

Where *length* is the number of bytes the URI is greater than. 0 to 65536.

Step 4 Create a SIP inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect sip policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 5 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 6 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the SIP class map that you created in Step 3 by entering the following command:

```
ciscoasa(config-pmap)# class class_map_name
ciscoasa(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
ciscoasa(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the command reference for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 2-4.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

b. To enable or disable instant messaging, enter the following command:

```
ciscoasa(config-pmap-p)# im
```

c. To enable or disable IP address privacy, enter the following command:

```
ciscoasa(config-pmap-p)# ip-address-privacy
```

d. To enable check on Max-forwards header field being 0 (which cannot be 0 before reaching the destination), enter the following command:

```
ciscoasa(config-pmap-p)# max-forwards-validation action {drop | drop-connection |
reset | log} [log]
```

e. To enable check on RTP packets flowing on the pinholes for protocol conformance, enter the following command:

```
ciscoasa(config-pmap-p)# rtp-conformance [enforce-payloadtype]
```

Where the **enforce-payloadtype** keyword enforces the payload type to be audio or video based on the signaling exchange.

f. To identify the Server and User-Agent header fields, which expose the software version of either a server or an endpoint, enter the following command:

```
ciscoasa(config-pmap-p)# software-version action {mask | log} [log]
```

Where the **mask** keyword masks the software version in the SIP messages.

g. To enable state checking validation, enter the following command:

```
ciscoasa(config-pmap-p)# state-checking action {drop | drop-connection | reset | log}
[log]
```

h. To enable strict verification of the header fields in the SIP messages according to RFC 3261, enter the following command:

```
ciscoasa(config-pmap-p)# strict-header-validation action {drop | drop-connection |
reset | log} [log]
```

i. To allow non SIP traffic using the well-known SIP signaling port, enter the following command:

```
ciscoasa(config-pmap-p)# traffic-non-sip
```

j. To identify the non-SIP URIs present in the Alert-Info and Call-Info header fields, enter the following command:

```
ciscoasa(config-pmap-p)# uri-non-sip action {mask | log} [log]
```

The following example shows how to disable instant messaging over SIP:

```
ciscoasa(config)# policy-map type inspect sip mymap
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# no im

ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect sip mymap

ciscoasa(config)# service-policy global_policy global
```

Configuring SIP Timeout Values

The media connections are torn down within two minutes after the connection becomes idle. This is, however, a configurable timeout and can be set for a shorter or longer period of time. To configure the timeout for the SIP control connection, enter the following command:

```
ciscoasa(config)# timeout sip hh:mm:ss
```

This command configures the idle timeout after which a SIP control connection is closed.

To configure the timeout for the SIP media connection, enter the following command:

```
ciscoasa(config)# timeout sip_media hh:mm:ss
```

This command configures the idle timeout after which a SIP media connection is closed.

Verifying and Monitoring SIP Inspection

The **show sip** command assists in troubleshooting SIP inspection engine issues and is described with the **inspect protocol sip udp 5060** command. The **show timeout sip** command displays the timeout value of the designated protocol.

The **show sip** command displays information for SIP sessions established across the ASA. Along with the **debug sip** and **show local-host** commands, this command is used for troubleshooting SIP inspection engine issues.

Note We recommend that you configure the **pager** command before entering the **show sip** command. If there are a lot of SIP session records and the **pager** command is not configured, it takes a while for the **show sip** command output to reach its end.

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

```
ciscoasa# show sip
Total: 2
call-id c3943000-960ca-2e43-228f@10.130.56.44
    state Call init, idle 0:00:01
call-id c3943000-860ca-7e1f-11f7@10.130.56.45
    state Active, idle 0:00:06
```

This sample shows two active SIP sessions on the ASA (as shown in the Total field). Each call-id represents a call.

The first session, with the call-id c3943000-960ca-2e43-228f@10.130.56.44, is in the state Call Init, which means the session is still in call setup. Call setup is not complete until a final response to the call has been received. For instance, the caller has already sent the INVITE, and maybe received a 100 Response, but has not yet seen the 200 OK, so the call setup is not complete yet. Any non-1xx response message is considered a final response. This session has been idle for 1 second.

The second session is in the state Active, in which call setup is complete and the endpoints are exchanging media. This session has been idle for 6 seconds.

Skinny (SCCP) Inspection

This section describes SCCP application inspection. This section includes the following topics:

- SCCP Inspection Overview, page 11-25
- Supporting Cisco IP Phones, page 11-25
- Restrictions and Limitations, page 11-26
- Configuring a Skinny (SCCP) Inspection Policy Map for Additional Inspection Control, page 11-26
- Verifying and Monitoring SIP Inspection, page 11-24

SCCP Inspection Overview

Note For specific information about setting up the Phone Proxy on the ASA, which is part of the Cisco Unified Communications architecture and supports IP phone deployment, see Chapter 16, "Configuring the Cisco Phone Proxy.".

Skinny (SCCP) is a simplified protocol used in VoIP networks. Cisco IP Phones using SCCP can coexist in an H.323 environment. When used with Cisco CallManager, the SCCP client can interoperate with H.323 compliant terminals.

The ASA supports PAT and NAT for SCCP. PAT is necessary if you have more IP phones than global IP addresses for the IP phones to use. By supporting NAT and PAT of SCCP Signaling packets, Skinny application inspection ensures that all SCCP signalling and media packets can traverse the ASA.

Normal traffic between Cisco CallManager and Cisco IP Phones uses SCCP and is handled by SCCP inspection without any special configuration. The ASA also supports DHCP options 150 and 66, which it accomplishes by sending the location of a TFTP server to Cisco IP Phones and other DHCP clients. Cisco IP Phones might also include DHCP option 3 in their requests, which sets the default route. For more information, see the general operations configuration guide.

Note The ASA supports inspection of traffic from Cisco IP Phones running SCCP protocol version 19 and earlier.

Supporting Cisco IP Phones

Note For specific information about setting up the Phone Proxy on the ASA, which is part of the Cisco Unified Communications architecture and supports IP phone deployment, see Chapter 16, "Configuring the Cisco Phone Proxy."

In topologies where Cisco CallManager is located on the higher security interface with respect to the Cisco IP Phones, if NAT is required for the Cisco CallManager IP address, the mapping must be **static** as a Cisco IP Phone requires the Cisco CallManager IP address to be specified explicitly in its configuration. An static identity entry allows the Cisco CallManager on the higher security interface to accept registrations from the Cisco IP Phones.

Cisco IP Phones require access to a TFTP server to download the configuration information they need to connect to the Cisco CallManager server.

When the Cisco IP Phones are on a lower security interface compared to the TFTP server, you must use an ACL to connect to the protected TFTP server on UDP port 69. While you do need a static entry for the TFTP server, this does not have to be an identity static entry. When using NAT, an identity static entry maps to the same IP address. When using PAT, it maps to the same IP address and port.

When the Cisco IP Phones are on a *higher* security interface compared to the TFTP server and Cisco CallManager, no ACL or static entry is required to allow the Cisco IP Phones to initiate the connection.

Restrictions and Limitations

The following are limitations that apply to the current version of PAT and NAT support for SCCP:

- PAT does not work with configurations containing the **alias** command.
- Outside NAT or PAT is *not* supported.

If the address of an internal Cisco CallManager is configured for NAT or PAT to a different IP address or port, registrations for external Cisco IP Phones fail because the ASA currently does not support NAT or PAT for the file content transferred over TFTP. Although the ASA supports NAT of TFTP messages and opens a pinhole for the TFTP file, the ASA cannot translate the Cisco CallManager IP address and port embedded in the Cisco IP Phone configuration files that are transferred by TFTP during phone registration.

Note The ASA supports stateful failover of SCCP calls except for calls that are in the middle of call setup.

Configuring a Skinny (SCCP) Inspection Policy Map for Additional Inspection Control

To specify actions when a message violates a parameter, create an SCCP inspection policy map. You can then apply the inspection policy map when you enable SCCP inspection.

To create an SCCP inspection policy map, perform the following steps:

Step 1 (Optional) Add one or more regular expressions for use in traffic matching commands according to the general operations configuration guide. See the types of text you can match in the **match** commands described in Step 3.

Step 2 (Optional) Create one or more regular expression class maps to group regular expressions according to the general operations configuration guide.

Step 3 Create an SCCP inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect skinny policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 4 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 5 To apply actions to matching traffic, perform the following steps.

a. Specify the traffic on which you want to perform actions using one of the following methods:

- Specify the SCCP class map that you created in Step 3 by entering the following command:

```
ciscoasa(config-pmap)# class class_map_name
ciscoasa(config-pmap-c)#
```

- Specify traffic directly in the policy map using one of the **match** commands described in Step 3. If you use a **match not** command, then any traffic that does not match the criterion in the **match not** command has the action applied.

b. Specify the action you want to perform on the matching traffic by entering the following command:

```
ciscoasa(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

Not all options are available for each **match** or **class** command. See the CLI help or the command reference for the exact options available.

The **drop** keyword drops all packets that match.

The **send-protocol-error** keyword sends a protocol error message.

The **drop-connection** keyword drops the packet and closes the connection.

The **mask** keyword masks out the matching portion of the packet.

The **reset** keyword drops the packet, closes the connection, and sends a TCP reset to the server and/or client.

The **log** keyword, which you can use alone or with one of the other keywords, sends a system log message.

The **rate-limit** *message_rate* argument limits the rate of messages.

Step 6 You can specify multiple **class** or **match** commands in the policy map. For information about the order of **class** and **match** commands, see the "Defining Actions in an Inspection Policy Map" section on page 2-4.To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

b. To enforce registration before calls can be placed, enter the following command:

```
ciscoasa(config-pmap-p)# enforce-registration
```

c. To set the maximum SCCP station message ID allowed, enter the following command:

```
ciscoasa(config-pmap-p)# message-ID max hex_value
```

Where the *hex_value* argument is the station message ID in hex.

d. To check RTP packets flowing on the pinholes for protocol conformance, enter the following command:

```
ciscoasa(config-pmap-p)# rtp-conformance [enforce-payloadtype]
```

Where the **enforce-payloadtype** keyword enforces the payload type to be audio or video based on the signaling exchange.

e. To set the maximum and minimum SCCP prefix length value allowed, enter the following command:

```
ciscoasa(config-pmap-p)# sccp-prefix-len {max | min} value_length
```

Where the *value_length* argument is a maximum or minimum value.

f. To configure the timeout value for signaling and media connections, enter the following command:

```
ciscoasa(config-pmap-p)# timeout
```

The following example shows how to define an SCCP inspection policy map.

```
ciscoasa(config)# policy-map type inspect skinny skinny-map
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# enforce-registration
ciscoasa(config-pmap-p)# match message-id range 200 300
ciscoasa(config-pmap-p)# drop log
ciscoasa(config)# class-map inspection_default
ciscoasa(config-cmap)# match default-inspection-traffic
ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect skinny skinny-map
ciscoasa(config)# service-policy global_policy global
```

Verifying and Monitoring SCCP Inspection

The **show skinny** command assists in troubleshooting SCCP (Skinny) inspection engine issues. The following is sample output from the **show skinny** command under the following conditions. There are two active Skinny sessions set up across the ASA. The first one is established between an internal Cisco IP Phone at local address 10.0.0.11 and an external Cisco CallManager at 172.18.1.33. TCP port 2000 is the CallManager. The second one is established between another internal Cisco IP Phone at local address 10.0.0.22 and the same Cisco CallManager.

```
ciscoasa# show skinny
        LOCAL                   FOREIGN                 STATE
---------------------------------------------------------------
1       10.0.0.11/52238         172.18.1.33/2000             1
  MEDIA 10.0.0.11/22948         172.18.1.22/20798
2       10.0.0.22/52232         172.18.1.33/2000             1
  MEDIA 10.0.0.22/20798         172.18.1.11/22948
```

The output indicates that a call has been established between two internal Cisco IP Phones. The RTP listening ports of the first and second phones are UDP 22948 and 20798 respectively.

The following is sample output from the **show xlate debug** command for these Skinny connections:

```
ciscoasa# show xlate debug
2 in use, 2 most used
Flags:  D - DNS, d - dump, I - identity, i - inside, n - no random,
        r - portmap, s - static
NAT from inside:10.0.0.11 to outside:172.18.1.11 flags si idle 0:00:16 timeout 0:05:00
NAT from inside:10.0.0.22 to outside:172.18.1.22 flags si idle 0:00:14 timeout 0:05:00
```

Configuring Inspection of Database and Directory Protocols

This chapter describes how to configure application layer protocol inspection. Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the ASA to do a deep packet inspection instead of passing the packet through the fast path. As a result, inspection engines can affect overall throughput.

Several common inspection engines are enabled on the ASA by default, but you might need to enable others depending on your network.

This chapter includes the following sections:

ILS Inspection

The ILS inspection engine provides NAT support for Microsoft NetMeeting, SiteServer, and Active Directory products that use LDAP to exchange directory information with an ILS server.

The ASA supports NAT for ILS, which is used to register and locate endpoints in the ILS or SiteServer Directory. PAT cannot be supported because only IP addresses are stored by an LDAP database.

For search responses, when the LDAP server is located outside, NAT should be considered to allow internal peers to communicate locally while registered to external LDAP servers. For such search responses, xlates are searched first, and then DNAT entries to obtain the correct address. If both of these searches fail, then the address is not changed. For sites using NAT 0 (no NAT) and not expecting DNAT interaction, we recommend that the inspection engine be turned off to provide better performance.

Additional configuration may be necessary when the ILS server is located inside the ASA border. This would require a hole for outside clients to access the LDAP server on the specified port, typically TCP 389.

Because ILS traffic only occurs on the secondary UDP channel, the TCP connection is disconnected after the TCP inactivity interval. By default, this interval is 60 minutes and can be adjusted using the **timeout** command.

ILS/LDAP follows a client/server model with sessions handled over a single TCP connection. Depending on the client's actions, several of these sessions may be created.

During connection negotiation time, a BIND PDU is sent from the client to the server. Once a successful BIND RESPONSE from the server is received, other operational messages may be exchanged (such as ADD, DEL, SEARCH, or MODIFY) to perform operations on the ILS Directory. The ADD REQUEST and SEARCH RESPONSE PDUs may contain IP addresses of NetMeeting peers, used by H.323 (SETUP and CONNECT messages) to establish the NetMeeting sessions. Microsoft NetMeeting v2.X and v3.X provides ILS support.

The ILS inspection performs the following operations:

- Decodes the LDAP REQUEST/RESPONSE PDUs using the BER decode functions
- Parses the LDAP packet
- Extracts IP addresses
- Translates IP addresses as necessary
- Encodes the PDU with translated addresses using BER encode functions
- Copies the newly encoded PDU back to the TCP packet
- Performs incremental TCP checksum and sequence number adjustment

ILS inspection has the following limitations:

- Referral requests and responses are not supported
- Users in multiple directories are not unified
- Single users having multiple identities in multiple directories cannot be recognized by NAT

Note Because H225 call signalling traffic only occurs on the secondary UDP channel, the TCP connection is disconnected after the interval specified by the TCP **timeout** command. By default, this interval is set at 60 minutes.

SQL*Net Inspection

SQL*Net inspection is enabled by default.

The SQL*Net protocol consists of different packet types that the ASA handles to make the data stream appear consistent to the Oracle applications on either side of the ASA.

The default port assignment for SQL*Net is 1521. This is the value used by Oracle for SQL*Net, but this value does not agree with IANA port assignments for Structured Query Language (SQL). Use the **class-map** command to apply SQL*Net inspection to a range of port numbers.

Note Disable SQL*Net inspection when SQL data transfer occurs on the same port as the SQL control TCP port 1521. The security appliance acts as a proxy when SQL*Net inspection is enabled and reduces the client window size from 65000 to about 16000 causing data transfer issues.

The ASA translates all addresses and looks in the packets for all embedded ports to open for SQL*Net Version 1.

For SQL*Net Version 2, all DATA or REDIRECT packets that immediately follow REDIRECT packets with a zero data length will be fixed up.

The packets that need fix-up contain embedded host/port addresses in the following format:

```
(ADDRESS=(PROTOCOL=tcp)(DEV=6)(HOST=a.b.c.d)(PORT=a))
```

SQL*Net Version 2 TNSFrame types (Connect, Accept, Refuse, Resend, and Marker) will not be scanned for addresses to NAT nor will inspection open dynamic connections for any embedded ports in the packet.

SQL*Net Version 2 TNSFrames, Redirect, and Data packets will be scanned for ports to open and addresses to NAT, if preceded by a REDIRECT TNSFrame type with a zero data length for the payload. When the Redirect message with data length zero passes through the ASA, a flag will be set in the connection data structure to expect the Data or Redirect message that follows to be translated and ports to be dynamically opened. If one of the TNS frames in the preceding paragraph arrive after the Redirect message, the flag will be reset.

The SQL*Net inspection engine will recalculate the checksum, change IP, TCP lengths, and readjust Sequence Numbers and Acknowledgment Numbers using the delta of the length of the new and old message.

SQL*Net Version 1 is assumed for all other cases. TNSFrame types (Connect, Accept, Refuse, Resend, Marker, Redirect, and Data) and all packets will be scanned for ports and addresses. Addresses will be translated and port connections will be opened.

Sun RPC Inspection

This section describes Sun RPC application inspection. This section includes the following topics:

- Sun RPC Inspection Overview, page 12-3
- Managing Sun RPC Services, page 12-4
- Verifying and Monitoring Sun RPC Inspection, page 12-4

Sun RPC Inspection Overview

The Sun RPC inspection engine enables or disables application inspection for the Sun RPC protocol. Sun RPC is used by NFS and NIS. Sun RPC services can run on any port. When a client attempts to access an Sun RPC service on a server, it must learn the port that service is running on. It does this by querying the port mapper process, usually rpcbind, on the well-known port of 111.

The client sends the Sun RPC program number of the service and the port mapper process responds with the port number of the service. The client sends its Sun RPC queries to the server, specifying the port identified by the port mapper process. When the server replies, the ASA intercepts this packet and opens both embryonic TCP and UDP connections on that port.

The following limitations apply to Sun RPC inspection:

- NAT or PAT of Sun RPC payload information is not supported.
- Sun RPC inspection supports inbound ACLs only. Sun RPC inspection does not support outbound ACLs because the inspection engine uses dynamic ACLs instead of secondary connections. Dynamic ACLs are always added on the ingress direction and not on egress; therefore, this inspection engine does not support outbound ACLs. To view the dynamic ACLs configured for the ASA, use the **show asp table classify domain permit** command. For information about the **show asp table classify domain permit** command, see the CLI configuration guide.

Managing Sun RPC Services

Use the Sun RPC services table to control Sun RPC traffic through the ASA based on established Sun RPC sessions. To create entries in the Sun RPC services table, use the **sunrpc-server** command in global configuration mode:

```
ciscoasa(config)# sunrpc-server interface_name ip_address mask service service_type
protocol {tcp | udp} port[-port] timeout hh:mm:ss
```

You can use this command to specify the timeout after which the pinhole that was opened by Sun RPC application inspection will be closed. For example, to create a timeout of 30 minutes to the Sun RPC server with the IP address 192.168.100.2, enter the following command:

```
ciscoasa(config)# sunrpc-server inside 192.168.100.2 255.255.255.255 service 100003
protocol tcp 111 timeout 00:30:00
```

This command specifies that the pinhole that was opened by Sun RPC application inspection will be closed after 30 minutes. In this example, the Sun RPC server is on the inside interface using TCP port 111. You can also specify UDP, a different port number, or a range of ports. To specify a range of ports, separate the starting and ending port numbers in the range with a hyphen (for example, 111-113).

The service type identifies the mapping between a specific service type and the port number used for the service. To determine the service type, which in this example is 100003, use the **sunrpcinfo** command at the UNIX or Linux command line on the Sun RPC server machine.

To clear the Sun RPC configuration, enter the following command.

```
ciscoasa(config)# clear configure sunrpc-server
```

This removes the configuration performed using the **sunrpc-server** command. The **sunrpc-server** command allows pinholes to be created with a specified timeout.

To clear the active Sun RPC services, enter the following command:

```
ciscoasa(config)# clear sunrpc-server active
```

This clears the pinholes that are opened by Sun RPC application inspection for specific services, such as NFS or NIS.

Verifying and Monitoring Sun RPC Inspection

The sample output in this section is for a Sun RPC server with an IP address of 192.168.100.2 on the inside interface and a Sun RPC client with an IP address of 209.168.200.5 on the outside interface.

To view information about the current Sun RPC connections, enter the **show conn** command. The following is sample output from the **show conn** command:

```
ciscoasa# show conn
15 in use, 21 most used
UDP out 209.165.200.5:800 in 192.168.100.2:2049 idle 0:00:04 flags -
UDP out 209.165.200.5:714 in 192.168.100.2:111 idle 0:00:04 flags -
UDP out 209.165.200.5:712 in 192.168.100.2:647 idle 0:00:05 flags -
UDP out 192.168.100.2:0 in 209.165.200.5:714 idle 0:00:05 flags i
ciscoasa(config)#
```

To display the information about the Sun RPC service table configuration, enter the **show running-config sunrpc-server** command. The following is sample output from the **show running-config sunrpc-server** command:

```
ciscoasa(config)# show running-config sunrpc-server
```

```
sunrpc-server inside 192.168.100.2 255.255.255.255 service 100003 protocol UDP port 111
timeout 0:30:00
sunrpc-server inside 192.168.100.2 255.255.255.255 service 100005 protocol UDP port 111
timeout 0:30:00
```

This output shows that a timeout interval of 30 minutes is configured on UDP port 111 for the Sun RPC server with the IP address 192.168.100.2 on the inside interface.

To display the pinholes open for Sun RPC services, enter the **show sunrpc-server active** command. The following is sample output from **show sunrpc-server active** command:

```
ciscoasa# show sunrpc-server active
LOCAL FOREIGN SERVICE TIMEOUT
-----------------------------------------------
1 209.165.200.5/0 192.168.100.2/2049 100003 0:30:00
2 209.165.200.5/0 192.168.100.2/2049 100003 0:30:00
3 209.165.200.5/0 192.168.100.2/647 100005 0:30:00
4 209.165.200.5/0 192.168.100.2/650 100005 0:30:00
```

The entry in the LOCAL column shows the IP address of the client or server on the inside interface, while the value in the FOREIGN column shows the IP address of the client or server on the outside interface.

To view information about the Sun RPC services running on a Sun RPC server, enter the **rpcinfo -p** command from the Linux or UNIX server command line. The following is sample output from the **rpcinfo -p** command:

```
sunrpcserver:~ # rpcinfo -p
program vers proto port
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 632 status
100024 1 tcp 635 status
100003 2 udp 2049 nfs
100003 3 udp 2049 nfs
100003 2 tcp 2049 nfs
100003 3 tcp 2049 nfs
100021 1 udp 32771 nlockmgr
100021 3 udp 32771 nlockmgr
100021 4 udp 32771 nlockmgr
100021 1 tcp 32852 nlockmgr
100021 3 tcp 32852 nlockmgr
100021 4 tcp 32852 nlockmgr
100005 1 udp 647 mountd
100005 1 tcp 650 mountd
100005 2 udp 647 mountd
100005 2 tcp 650 mountd
100005 3 udp 647 mountd
100005 3 tcp 650 mountd
```

In this output, port 647 corresponds to the mountd daemon running over UDP. The mountd process would more commonly be using port 32780. The mountd process running over TCP uses port 650 in this example.

CHAPTER 13

Configuring Inspection for Management Application Protocols

This chapter describes how to configure application layer protocol inspection. Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the ASA to do a deep packet inspection instead of passing the packet through the fast path. As a result, inspection engines can affect overall throughput.

Several common inspection engines are enabled on the ASA by default, but you might need to enable others depending on your network.

This chapter includes the following sections:

- DCERPC Inspection, page 13-1
- GTP Inspection, page 13-3
- RADIUS Accounting Inspection, page 13-8
- RSH Inspection, page 13-10
- SNMP Inspection, page 13-10
- XDMCP Inspection, page 13-11

DCERPC Inspection

This section describes the DCERPC inspection engine. This section includes the following topics:

- DCERPC Overview, page 13-1
- Configuring a DCERPC Inspection Policy Map for Additional Inspection Control, page 13-2

DCERPC Overview

DCERPC is a protocol widely used by Microsoft distributed client and server applications that allows software clients to execute programs on a server remotely.

This typically involves a client querying a server called the Endpoint Mapper listening on a well known port number for the dynamically allocated network information of a required service. The client then sets up a secondary connection to the server instance providing the service. The security appliance allows the appropriate port number and network address and also applies NAT, if needed, for the secondary connection.

DCERPC inspect maps inspect for native TCP communication between the EPM and client on well known TCP port 135. Map and lookup operations of the EPM are supported for clients. Client and server can be located in any security zone. The embedded server IP address and Port number are received from the applicable EPM response messages. Since a client may attempt multiple connections to the server port returned by EPM, multiple use of pinholes are allowed, which have user configurable timeouts.

Note DCERPC inspection only supports communication between the EPM and clients to open pinholes through theASA. Clients using RPC communication that does not use the EPM is not supported with DCERPC inspection.

Configuring a DCERPC Inspection Policy Map for Additional Inspection Control

To specify additional DCERPC inspection parameters, create a DCERPC inspection policy map. You can then apply the inspection policy map when you enable DCERPC inspection.

To create a DCERPC inspection policy map, perform the following steps:

Step 1 Create a DCERPC inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect dcerpc policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 2 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 3 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

b. To configure the timeout for DCERPC pinholes and override the global system pinhole timeout of two minutes, enter the following command:

```
ciscoasa(config-pmap-p)# timeout pinhole hh:mm:ss
```

Where the *hh:mm:ss* argument is the timeout for pinhole connections. Value is between 0:0:1 and 1193:0:0.

c. To configure options for the endpoint mapper traffic, enter the following command:

```
ciscoasa(config-pmap-p)# endpoint-mapper [epm-service-only] [lookup-operation
[timeout hh:mm:ss]]
```

Where the *hh:mm:ss* argument is the timeout for pinholes generated from the lookup operation. If no timeout is configured for the lookup operation, the timeout pinhole command or the default is used. The **epm-service-only** keyword enforces endpoint mapper service during binding so that only its service traffic is processed. The **lookup-operation** keyword enables the lookup operation of the endpoint mapper service.

The following example shows how to define a DCERPC inspection policy map with the timeout configured for DCERPC pinholes.

```
ciscoasa(config)# policy-map type inspect dcerpc dcerpc_map
ciscoasa(config-pmap)# timeout pinhole 0:10:00

ciscoasa(config)# class-map dcerpc
ciscoasa(config-cmap)# match port tcp eq 135

ciscoasa(config)# policy-map global-policy
ciscoasa(config-pmap)# class dcerpc
ciscoasa(config-pmap-c)# inspect dcerpc dcerpc-map

ciscoasa(config)# service-policy global-policy global
```

GTP Inspection

This section describes the GTP inspection engine. This section includes the following topics:

- GTP Inspection Overview, page 13-3
- Configuring a GTP Inspection Policy Map for Additional Inspection Control, page 13-4
- Verifying and Monitoring GTP Inspection, page 13-7

Note GTP inspection requires a special license. If you enter GTP-related commands on a ASA without the required license, the ASA displays an error message.

GTP Inspection Overview

GPRS provides uninterrupted connectivity for mobile subscribers between GSM networks and corporate networks or the Internet. The GGSN is the interface between the GPRS wireless data network and other networks. The SGSN performs mobility, data session management, and data compression.

The UMTS is the commercial convergence of fixed-line telephony, mobile, Internet and computer technology. UTRAN is the networking protocol used for implementing wireless networks in this system. GTP allows multi-protocol packets to be tunneled through a UMTS/GPRS backbone between a GGSN, an SGSN and the UTRAN.

GTP does not include any inherent security or encryption of user data, but using GTP with the ASA helps protect your network against these risks.

The SGSN is logically connected to a GGSN using GTP. GTP allows multiprotocol packets to be tunneled through the GPRS backbone between GSNs. GTP provides a tunnel control and management protocol that allows the SGSN to provide GPRS network access for a mobile station by creating, modifying, and deleting tunnels. GTP uses a tunneling mechanism to provide a service for carrying user data packets.

Note When using GTP with failover, if a GTP connection is established and the active unit fails before data is transmitted over the tunnel, the GTP data connection (with a “j” flag set) is not replicated to the standby unit. This occurs because the active unit does not replicate embryonic connections to the standby unit.

Configuring a GTP Inspection Policy Map for Additional Inspection Control

If you want to enforce additional parameters on GTP traffic, create and configure a GTP map. If you do not specify a map with the **inspect gtp** command, the ASA uses the default GTP map, which is preconfigured with the following default values:

- **request-queue 200**
- **timeout gsn 0:30:00**
- **timeout pdp-context 0:30:00**
- **timeout request 0:01:00**
- **timeout signaling 0:30:00**
- **timeout tunnel 0:01:00**
- **tunnel-limit 500**

To create and configure a GTP map, perform the following steps. You can then apply the GTP map when you enable GTP inspection according to the "Configuring Application Layer Protocol Inspection" section on page 9-7.

Step 1 Create a GTP inspection policy map, enter the following command:

```
ciscoasa(config)# policy-map type inspect gtp policy_map_name
ciscoasa(config-pmap)#
```

Where the *policy_map_name* is the name of the policy map. The CLI enters policy-map configuration mode.

Step 2 (Optional) To add a description to the policy map, enter the following command:

```
ciscoasa(config-pmap)# description string
```

Step 3 To match an Access Point name, enter the following command:

```
ciscoasa(config-pmap)# match [not] apn regex [regex_name | class regex_class_name]
```

Step 4 To match a message ID, enter the following command:

```
ciscoasa(config-pmap)# match [not] message id [message_id | range lower_range upper_range]
```

Where the *message_id* is an alphanumeric identifier between 1 and 255. The *lower_range* is lower range of message IDs. The *upper_range* is the upper range of message IDs.

Step 5 To match a message length, enter the following command:

```
ciscoasa(config-pmap)# match [not] message length min min_length max max_length
```

Where the *min_length* and *max_length* are both between 1 and 65536. The length specified by this command is the sum of the GTP header and the rest of the message, which is the payload of the UDP packet.

Step 6 To match the version, enter the following command:

```
ciscoasa(config-pmap)# match [not] version [version_id | range lower_range upper_range]
```

Where the *version_id* is between 0and 255. The *lower_range* is lower range of versions. The *upper_range* is the upper range of versions.

Step 7 To configure parameters that affect the inspection engine, perform the following steps:

a. To enter parameters configuration mode, enter the following command:

```
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)#
```

The **mnc** *network_code* argument is a two or three-digit value identifying the network code.

By default, the security appliance does not check for valid MCC/MNC combinations. This command is used for IMSI Prefix filtering. The MCC and MNC in the IMSI of the received packet is compared with the MCC/MNC configured with this command and is dropped if it does not match.

This command must be used to enable IMSI Prefix filtering. You can configure multiple instances to specify permitted MCC and MNC combinations. By default, the ASA does not check the validity of MNC and MCC combinations, so you must verify the validity of the combinations configured. To find more information about MCC and MNC codes, see the ITU E.212 recommendation, *Identification Plan for Land Mobile Stations*.

b. To allow invalid GTP packets or packets that otherwise would fail parsing and be dropped, enter the following command:

```
ciscoasa(config-pmap-p)# permit errors
```

By default, all invalid packets or packets that failed, during parsing, are dropped.

c. To enable support for GSN pooling, use the **permit response** command.

If the ASA performs GTP inspection, by default the ASA drops GTP responses from GSNs that were not specified in the GTP request. This situation occurs when you use load-balancing among a pool of GSNs to provide efficiency and scalability of GPRS.

You can enable support for GSN pooling by using the **permit response** command. This command configures the ASA to allow responses from any of a designated set of GSNs, regardless of the GSN to which a GTP request was sent. You identify the pool of load-balancing GSNs as a network object. Likewise, you identify the SGSN as a network object. If the GSN responding belongs to the same object group as the GSN that the GTP request was sent to and if the SGSN is in a object group that the responding GSN is permitted to send a GTP response to, the ASA permits the response.

d. To create an object to represent the pool of load-balancing GSNs, perform the following steps:

Use the **object-group** command to define a new network object group representing the pool of load-balancing GSNs.

```
ciscoasa(config)# object-group network GSN-pool-name
ciscoasa(config-network)#
```

For example, the following command creates an object group named gsnpool32:

```
ciscoasa(config)# object-group network gsnpool32
ciscoasa(config-network)#
```

e. Use the **network-object** command to specify the load-balancing GSNs. You can do so with one **network-object** command per GSN, using the **host** keyword. You can also using **network-object** command to identify whole networks containing GSNs that perform load balancing.

```
ciscoasa(config-network)# network-object host IP-address
```

For example, the following commands create three network objects representing individual hosts:

```
ciscoasa(config-network)# network-object host 192.168.100.1
ciscoasa(config-network)# network-object host 192.168.100.2
ciscoasa(config-network)# network-object host 192.168.100.3
ciscoasa(config-network)#
```

f. To create an object to represent the SGSN that the load-balancing GSNs are permitted to respond to, perform the following steps:

a. Use the **object-group** command to define a new network object group that will represent the SGSN that sends GTP requests to the GSN pool.

```
ciscoasa(config)# object-group network SGSN-name
ciscoasa(config-network)#
```

For example, the following command creates an object group named sgsn32:

```
ciscoasa(config)# object-group network sgsn32
ciscoasa(config-network)#
```

b. Use the **network-object** command with the **host** keyword to identify the SGSN.

```
ciscoasa(config-network)# network-object host IP-address
```

For example, the following command creates a network objects representing the SGSN:

```
ciscoasa(config-network)# network-object host 192.168.50.100
ciscoasa(config-network)#
```

g. To allow GTP responses from any GSN in the network object representing the GSN pool, defined in c., d, to the network object representing the SGSN, defined in c., f., enter the following commands:

```
ciscoasa(config)# gtp-map map_name
ciscoasa(config-gtp-map)# permit response to-object-group SGSN-name from-object-group
GSN-pool-name
```

For example, the following command permits GTP responses from any host in the object group named gsnpool32 to the host in the object group named sgsn32:

```
ciscoasa(config-gtp-map)# permit response to-object-group sgsn32 from-object-group
gsnpool32
```

The following example shows how to support GSN pooling by defining network objects for the GSN pool and the SGSN. An entire Class C network is defined as the GSN pool but you can identify multiple individual IP addresses, one per **network-object** command, instead of identifying whole networks. The example then modifies a GTP map to permit responses from the GSN pool to the SGSN.

```
ciscoasa(config)# object-group network gsnpool32
ciscoasa(config-network)# network-object 192.168.100.0 255.255.255.0
ciscoasa(config)# object-group network sgsn32
ciscoasa(config-network)# network-object host 192.168.50.100
ciscoasa(config)# gtp-map gtp-policy
ciscoasa(config-gtp-map)# permit response to-object-group sgsn32 from-object-group
gsnpool32
```

h. To specify the maximum number of GTP requests that will be queued waiting for a response, enter the following command:

```
ciscoasa(config-gtp-map)# request-queue max_requests
```

where the *max_requests* argument sets the maximum number of GTP requests that will be queued waiting for a response, from 1 to 4294967295. The default is 200.

When the limit has been reached and a new request arrives, the request that has been in the queue for the longest time is removed. The Error Indication, the Version Not Supported and the SGSN Context Acknowledge messages are not considered as requests and do not enter the request queue to wait for a response.

i. To change the inactivity timers for a GTP session, enter the following command:

```
ciscoasa(config-gtp-map)# timeout {gsn | pdp-context | request | signaling | tunnel}
hh:mm:ss
```

Enter this command separately for each timeout.

The **gsn** keyword specifies the period of inactivity after which a GSN will be removed.

The **pdp-context** keyword specifies the maximum period of time allowed before beginning to receive the PDP context.

The **request** keyword specifies the maximum period of time allowed before beginning to receive the GTP message.

The **signaling** keyword specifies the period of inactivity after which the GTP signaling will be removed.

The **tunnel** keyword specifies the period of inactivity after which the GTP tunnel will be torn down.

The *hh***:***mm***:***ss* argument is the timeout where *hh* specifies the hour, *mm* specifies the minutes, and *ss* specifies the seconds. The value **0** means never tear down.

j. To specify the maximum number of GTP tunnels allowed to be active on the ASA, enter the following command:

```
ciscoasa(config-gtp-map)# tunnel-limit max_tunnels
```

where the *max_tunnels* argument is the maximum number of tunnels allowed, from 1 to 4294967295. The default is 500.

New requests will be dropped once the number of tunnels specified by this command is reached.

The following example shows how to limit the number of tunnels in the network:

```
ciscoasa(config)# policy-map type inspect gtp gmap
ciscoasa(config-pmap)# parameters
ciscoasa(config-pmap-p)# tunnel-limit 3000

ciscoasa(config)# policy-map global_policy
ciscoasa(config-pmap)# class inspection_default
ciscoasa(config-pmap-c)# inspect gtp gmap

ciscoasa(config)# service-policy global_policy global
```

Verifying and Monitoring GTP Inspection

To display GTP configuration, enter the **show service-policy inspect gtp** command in privileged EXEC mode. For the detailed syntax for this command, see the command page in the command reference.

Use the **show service-policy inspect gtp statistics** command to show the statistics for GTP inspection. The following is sample output from the **show service-policy inspect gtp statistics** command:

```
ciscoasa# show service-policy inspect gtp statistics
GPRS GTP Statistics:
  version_not_support              0     msg_too_short            0
  unknown_msg                      0     unexpected_sig_msg       0
  unexpected_data_msg              0     ie_duplicated            0
  mandatory_ie_missing             0     mandatory_ie_incorrect   0
  optional_ie_incorrect            0     ie_unknown               0
  ie_out_of_order                  0     ie_unexpected            0
  total_forwarded                  0     total_dropped            0
  signalling_msg_dropped           0     data_msg_dropped         0
  signalling_msg_forwarded         0     data_msg_forwarded       0
  total created_pdp                0     total deleted_pdp        0
  total created_pdpmcb              0     total deleted_pdpmcb      0
  pdp_non_existent                  0
```

You can use the vertical bar (|) to filter the display. Type **?|** for more display filtering options.

The following is sample GSN output from the **show service-policy inspect gtp statistics gsn** command:

```
ciscoasa# show service-policy inspect gtp statistics gsn 9.9.9.9
1 in use, 1 most used, timeout 0:00:00

GTP GSN Statistics for 9.9.9.9, Idle 0:00:00, restart counter 0
    Tunnels Active 0Tunnels Created 0
    Tunnels Destroyed 0
    Total Messages Received 2
    Signaling Messages Data Messages
    total received 2 0
    dropped 0 0
    forwarded 2 0
```

Use the **show service-policy inspect gtp pdp-context** command to display PDP context-related information. The following is sample output from the **show service-policy inspect gtp pdp-context** command:

```
ciscoasa# show service-policy inspect gtp pdp-context detail
1 in use, 1 most used, timeout 0:00:00

Version TID                  MS Addr        SGSN Addr    Idle      APN
v1      1234567890123425        10.0.1.1        10.0.0.2 0:00:13  gprs.cisco.com

    user_name (IMSI): 214365870921435    MS address:         1.1.1.1
    primary pdp: Y                       nsapi: 2
    sgsn_addr_signal:        10.0.0.2    sgsn_addr_data:          10.0.0.2
    ggsn_addr_signal:        10.1.1.1    ggsn_addr_data:          10.1.1.1
    sgsn control teid:     0x000001d1    sgsn data teid:        0x000001d3
    ggsn control teid:     0x6306ffa0    ggsn data teid:        0x6305f9fc
    seq_tpdu_up:                    0    seq_tpdu_down:                  0
    signal_sequence:                0
    upstream_signal_flow:          0     upstream_data_flow:           0
    downstream_signal_flow:        0     downstream_data_flow:         0
    RAupdate_flow:                 0
```

The PDP context is identified by the tunnel ID, which is a combination of the values for IMSI and NSAPI. A GTP tunnel is defined by two associated PDP contexts in different GSN nodes and is identified with a Tunnel ID. A GTP tunnel is necessary to forward packets between an external packet data network and a MS user.

You can use the vertical bar (|) to filter the display, as in the following example:

```
ciscoasa# show service-policy gtp statistics  |  grep gsn
```

RADIUS Accounting Inspection

This section describes the RADIUS Accounting inspection engine. This section includes the following topics:

- RADIUS Accounting Inspection Overview, page 13-9
- Configuring a RADIUS Inspection Policy Map for Additional Inspection Control, page 13-9

RADIUS Accounting Inspection Overview

One of the well known problems is the over-billing attack in GPRS networks. The over-billing attack can cause consumers anger and frustration by being billed for services that they have not used. In this case, a malicious attacker sets up a connection to a server and obtains an IP address from the SGSN. When the attacker ends the call, the malicious server will still send packets to it, which gets dropped by the GGSN, but the connection from the server remains active. The IP address assigned to the malicious attacker gets released and reassigned to a legitimate user who will then get billed for services that the attacker will use.

RADIUS accounting inspection prevents this type of attack by ensuring the traffic seen by the GGSN is legitimate. With the RADIUS accounting feature properly configured, the security appliance tears down a connection based on matching the Framed IP attribute in the Radius Accounting Request Start message with the Radius Accounting Request Stop message. When the Stop message is seen with the matching IP address in the Framed IP attribute, the security appliance looks for all connections with the source matching the IP address.

You have the option to configure a secret pre-shared key with the RADIUS server so the security appliance can validate the message. If the shared secret is not configured, the security appliance does not need to validate the source of the message and will only check that the source IP address is one of the configured addresses allowed to send the RADIUS messages.

Note When using RADIUS accounting inspection with GPRS enabled, the ASA checks for the 3GPP-Session-Stop-Indicator in the Accounting Request STOP messages to properly handle secondary PDP contexts. Specifically, the ASA requires that the Accounting Request STOP messages include the 3GPP-SGSN-Address attribute before it will terminate the user sessions and all associated connections. Some third-party GGSNs might not send this attribute by default.

Configuring a RADIUS Inspection Policy Map for Additional Inspection Control

In order to use this feature, the **radius-accounting-map** will need to be specified in the **policy-map type management** and then applied to the service-policy using the new **control-plane** keyword to specify that this traffic is for to-the-box inspection.

The following example shows the complete set of commands in context to properly configure this feature:

Step 1 Configure the class map and the port:

```
class-map type management c1
  match port udp eq 1888
```

Step 2 Create the policy map, and configure the parameters for RADIUS accounting inspection using the parameter command to access the proper mode to configure the attributes, host, and key.

```
policy-map type inspect radius-accounting radius_accounting_map
  parameters
    host 10.1.1.1 inside key 123456789
    send response
    enable gprs
    validate-attribute 22
```

Step 3 Configure the service policy.

```
policy-map global_policy
  class c1
     inspect radius-accounting radius_accounting_map
```

```
service-policy global_policy global
```

RSH Inspection

RSH inspection is enabled by default. The RSH protocol uses a TCP connection from the RSH client to the RSH server on TCP port 514. The client and server negotiate the TCP port number where the client listens for the STDERR output stream. RSH inspection supports NAT of the negotiated port number if necessary.

SNMP Inspection

This section describes the SNMP inspection engine. This section includes the following topics:

- SNMP Inspection Overview, page 13-10
- Configuring an SNMP Inspection Policy Map for Additional Inspection Control, page 13-10

SNMP Inspection Overview

SNMP application inspection lets you restrict SNMP traffic to a specific version of SNMP. Earlier versions of SNMP are less secure; therefore, denying certain SNMP versions may be required by your security policy. The ASA can deny SNMP versions 1, 2, 2c, or 3. You control the versions permitted by creating an SNMP map.

You then apply the SNMP map when you enable SNMP inspection according to the "Configuring Application Layer Protocol Inspection" section on page 9-7.

Configuring an SNMP Inspection Policy Map for Additional Inspection Control

To create an SNMP inspection policy map, perform the following steps:

Step 1 To create an SNMP map, enter the following command:

```
ciscoasa(config)# snmp-map map_name
ciscoasa(config-snmp-map)#
```

where *map_name* is the name of the SNMP map. The CLI enters SNMP map configuration mode.

Step 2 To specify the versions of SNMP to deny, enter the following command for each version:

```
ciscoasa(config-snmp-map)# deny version version
ciscoasa(config-snmp-map)#
```

where *version* is 1, 2, 2c, or 3.

The following example denies SNMP Versions 1 and 2:

```
ciscoasa(config)# snmp-map sample_map
ciscoasa(config-snmp-map)# deny version 1
ciscoasa(config-snmp-map)# deny version 2
```

XDMCP Inspection

XDMCP inspection is enabled by default; however, the XDMCP inspection engine is dependent upon proper configuration of the **established** command.

XDMCP is a protocol that uses UDP port 177 to negotiate X sessions, which use TCP when established.

For successful negotiation and start of an XWindows session, the ASA must allow the TCP back connection from the Xhosted computer. To permit the back connection, use the **established** command on the ASA. Once XDMCP negotiates the port to send the display, The **established** command is consulted to verify if this back connection should be permitted.

During the XWindows session, the manager talks to the display Xserver on the well-known port 6000 | n. Each display has a separate connection to the Xserver, as a result of the following terminal setting.

```
setenv DISPLAY Xserver:n
```

where *n* is the display number.

When XDMCP is used, the display is negotiated using IP addresses, which the ASA can NAT if needed. XDCMP inspection does not support PAT.

PART 5

Configuring Unified Communications

CHAPTER 14

Information About Cisco Unified Communications Proxy Features

This chapter describes how to configure the adaptive security appliance for Cisco Unified Communications Proxy features.

This chapter includes the following sections:

Information About the Adaptive Security Appliance in Cisco Unified Communications

This section describes the Cisco UC Proxy features on the Cisco ASA 5500 series appliances. The purpose of a proxy is to terminate and reoriginate connections between a client and server. The proxy delivers a range of security functions such as traffic inspection, protocol conformance, and policy control to ensure security for the internal network. An increasingly popular function of a proxy is to terminate encrypted connections in order to apply security policies while maintaining confidentiality of connections. The Cisco ASA 5500 Series appliances are a strategic platform to provide proxy functions for unified communications deployments.

The Cisco UC Proxy includes the following solutions:

Phone Proxy: Secure remote access for Cisco encrypted endpoints, and VLAN traversal for Cisco softphones

The phone proxy feature enables termination of Cisco SRTP/TLS-encrypted endpoints for secure remote access. The phone proxy allows large scale deployments of secure phones without a large scale VPN remote access hardware deployment. End-user infrastructure is limited to just the IP endpoint, without VPN tunnels or hardware.

The Cisco adaptive security appliance phone proxy is the replacement product for the Cisco Unified Phone Proxy. Additionally, the phone proxy can be deployed for voice/data VLAN traversal for softphone applications. Cisco IP Communicator (CIPC) traffic (both media and signaling) can be proxied through the ASA, thus traversing calls securely between voice and data VLANs.

For information about the differences between the TLS proxy and phone proxy, go to the following URL for Unified Communications content, including TLS Proxy vs. Phone Proxy white paper:

http://www.cisco.com/go/secureuc

TLS Proxy: Decryption and inspection of Cisco Unified Communications encrypted signaling

End-to-end encryption often leaves network security appliances “blind” to media and signaling traffic, which can compromise access control and threat prevention security functions. This lack of visibility can result in a lack of interoperability between the firewall functions and the encrypted voice, leaving businesses unable to satisfy both of their key security requirements.

The ASA is able to intercept and decrypt encrypted signaling from Cisco encrypted endpoints to the Cisco Unified Communications Manager (Cisco UCM), and apply the required threat protection and access control. It can also ensure confidentiality by re-encrypting the traffic onto the Cisco UCM servers.

Typically, the ASA TLS Proxy functionality is deployed in campus unified communications network. This solution is ideal for deployments that utilize end to end encryption and firewalls to protect Unified Communications Manager servers.

Mobility Proxy: Secure connectivity between Cisco Unified Mobility Advantage server and Cisco Unified Mobile Communicator clients

Cisco Unified Mobility solutions include the Cisco Unified Mobile Communicator (Cisco UMC), an easy-to-use software application for mobile handsets that extends enterprise communications applications and services to mobile phones and the Cisco Unified Mobility Advantage (Cisco UMA) server. The Cisco Unified Mobility solution streamlines the communication experience, enabling single number reach and integration of mobile endpoints into the Unified Communications infrastructure.

The security appliance acts as a proxy, terminating and reoriginating the TLS signaling between the Cisco UMC and Cisco UMA. As part of the proxy security functionality, inspection is enabled for the Cisco UMA Mobile Multiplexing Protocol (MMP), the protocol between Cisco UMC and Cisco UMA.

Presence Federation Proxy: Secure connectivity between Cisco Unified Presence servers and Cisco/Microsoft Presence servers

Cisco Unified Presence solution collects information about the availability and status of users, such as whether they are using communication devices, such as IP phones at particular times. It also collects information regarding their communications capabilities, such as whether web collaboration or video conferencing is enabled. Using user information captured by Cisco Unified Presence, applications such as Cisco Unified Personal Communicator and Cisco UCM can improve productivity by helping users connect with colleagues more efficiently through determining the most effective way for collaborative communication.

Using the ASA as a secure presence federation proxy, businesses can securely connect their Cisco Unified Presence (Cisco UP) servers to other Cisco or Microsoft Presence servers, enabling intra-enterprise communications. The security appliance terminates the TLS connectivity between the servers, and can inspect and apply policies for the SIP communications between the servers.

Cisco Intercompany Media Engine Proxy: Secure connectivity between Cisco UCM servers in different enterprises for IP Phone traffic

As more unified communications are deployed within enterprises, cases where business-to-business calls utilize unified communications on both sides with the Public Switched Network (PSTN) in the middle become increasingly common. All outside calls go over circuits to telephone providers and from there are delivered to all external destinations.

The Cisco Intercompany Media Engine gradually creates dynamic, encrypted VoIP connections between businesses, so that a collection of enterprises that work together end up looking like one giant business with secure VoIP interconnections between them.

There are three components to a Cisco Intercompany Media Engine deployment within an enterprise: a Cisco Intercompany Media Engine server, a call agent (the Cisco Unified Communications Manager) and an ASA running the Cisco Intercompany Media Engine Proxy.

The ASA provides perimeter security by encrypting signaling connections between enterprises and preventing unathorized calls. An ASA running the Cisco Intercompany Media Engine Proxy can either be deployed as an Internet firewall or be designated as a Cisco Intercompany Media Engine Proxy and placed in the DMZ, off the path of the regular Internet traffic.

TLS Proxy Applications in Cisco Unified Communications

Table 14-1 shows the Cisco Unified Communications applications that utilize the TLS proxy on the ASA.

Table 14-1 TLS Proxy Applications and the Security Appliance

Application	TLS Client	TLS Server	Client Authentication	Security Appliance Server Role	Security Appliance Client Role
Phone Proxy and TLS Proxy	IP phone	Cisco UCM	Yes	Proxy certificate, self-signed or by internal CA	Local dynamic certificate signed by the ASA CA (might not need certificate for phone proxy application)
Mobility Proxy	Cisco UMC	Cisco UMA	No	Using the Cisco UMA private key or certificate impersonation	Any static configured certificate
Presence Federation Proxy	Cisco UP or MS LCS/OCS	Cisco UP or MS LCS/OCS	Yes	Proxy certificate, self-signed or by internal CA	Using the Cisco UP private key or certificate impersonation

The ASA supports TLS proxy for various voice applications. For the phone proxy, the TLS proxy running on the ASA has the following key features:

- The ASA forces remote IP phones connecting to the phone proxy through the Internet to be in secured mode even when the Cisco UCM cluster is in non-secure mode.
- The TLS proxy is implemented on the ASA to intercept the TLS signaling from IP phones.
- The TLS proxy decrypts the packets, sends packets to the inspection engine for NAT rewrite and protocol conformance, optionally encrypts packets, and sends them to Cisco UCM or sends them in clear text if the IP phone is configured to be in nonsecure mode on the Cisco UCM.
- The ASA acts as a media terminator as needed and translates between SRTP and RTP media streams.
- The TLS proxy is a transparent proxy that works based on establishing trusted relationship between the TLS client, the proxy (the ASA), and the TLS server.

For the Cisco Unified Mobility solution, the TLS client is a Cisco UMA client and the TLS server is a Cisco UMA server. The ASA is between a Cisco UMA client and a Cisco UMA server. The mobility proxy (implemented as a TLS proxy) for Cisco Unified Mobility allows the use of an imported PKCS-12 certificate for server proxy during the handshake with the client. Cisco UMA clients are not required to present a certificate (no client authentication) during the handshake.

For the Cisco Unified Presence solution, the ASA acts as a TLS proxy between the Cisco UP server and the foreign server. This allows the ASA to proxy TLS messages on behalf of the server that initiates the TLS connection, and route the proxied TLS messages to the client. The ASA stores certificate trustpoints for the server and the client, and presents these certificates on establishment of the TLS session.

Licensing for Cisco Unified Communications Proxy Features

The Cisco Unified Communications proxy features supported by the ASA require a Unified Communications Proxy license:

- Phone proxy
- TLS proxy for encrypted voice inspection
- Presence federation proxy
- Intercompany media engine proxy

Note In Version 8.2(2) and later, the Mobility Advantage proxy no longer requires a Unified Communications Proxy license.

The following table shows the Unified Communications Proxy license details by platform for the phone proxy, TLS proxy for encrypted voice inspection, and presence federation proxy:

Note This feature is not available on No Payload Encryption models.

Model	License Requirement[1]
ASA 5505	Base License and Security Plus License: 2 sessions. *Optional license: 24 sessions.*
ASA 5510	Base License and Security Plus License: 2 sessions. *Optional licenses: 24, 50, or 100 sessions.*
ASA 5520	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, or 1000 sessions.*
ASA 5540	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, or 2000 sessions.*
ASA 5550	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*
ASA 5580	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]

Model	License Requirement[1]
ASA 5512-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, or 500 sessions.*
ASA 5515-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, or 500 sessions.*
ASA 5525-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, or 1000 sessions.*
ASA 5545-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, or 2000 sessions.*
ASA 5555-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*
ASA 5585-X with SSP-10	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*
ASA 5585-X with SSP-20, -40, or -60	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]
ASA SM	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]

[1] The following applications use TLS proxy sessions for their connections. Each TLS proxy session used by these applications (and only these applications) is counted against the UC license limit:
- Phone Proxy
- Presence Federation Proxy
- Encrypted Voice Inspection

Other applications that use TLS proxy sessions do not count towards the UC limit, for example, Mobility Advantage Proxy (which does not require a license) and IME (which requires a separate IME license).

Some UC applications might use multiple sessions for a connection. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS proxy connections, so 2 UC Proxy sessions are used.

You independently set the TLS proxy limit using the **tls-proxy maximum-sessions** command. To view the limits of your model, enter the **tls-proxy maximum-sessions ?** command. When you apply a UC license that is higher than the default TLS proxy limit, the security appliance automatically sets the TLS proxy limit to match the UC limit. The TLS proxy limit takes precedence over the UC license limit; if you set the TLS proxy limit to be less than the UC license, then you cannot use all of the sessions in your UC license.

Note: For license part numbers ending in “K8” (for example, licenses under 250 users), TLS proxy sessions are limited to 1000. For license part numbers ending in “K9” (for example, licenses 250 users or larger), the TLS proxy limit depends on the configuration, up to the model limit. K8 and K9 refer to whether the license is restricted for export: K8 is unrestricted, and K9 is restricted.

Note: If you clear the configuration (using the **clear configure all** command, for example), then the TLS proxy limit is set to the default for your model; if this default is lower than the UC license limit, then you see an error message to use the tls-proxy maximum-sessions command to raise the limit again . If you use failover and enter the **write standby** command on the primary unit to force a configuration synchronization, the **clear configure all** command is generated on the secondary unit automatically, so you may see the warning message on the secondary unit. Because the configuration synchronization restores the TLS proxy limit set on the primary unit, you can ignore the warning.

You might also use SRTP encryption sessions for your connections:
- For K8 licenses, SRTP sessions are limited to 250.
- For K9 licenses, there is not limit.

Note: Only calls that require encryption/decryption for media are counted towards the SRTP limit; if passthrough is set for the call, even if both legs are SRTP, they do not count towards the limit.

[2] With the 10,000-session UC license, the total combined sessions can be 10,000, but the maximum number of Phone Proxy sessions is 5000.

Table 14-2 shows the default and maximum TLS session details by platform.

Table 14-2 Default and Maximum TLS Sessions on the Security Appliance

Security Appliance Platform	Default TLS Sessions	Maximum TLS Sessions
ASA 5505	10	80
ASA 5510	100	200
ASA 5520	300	1200
ASA 5540	1000	4500
ASA 5550	2000	4500
ASA 5580	4000	13,000

The following table shows the Unified Communications Proxy license details by platform for intercompany media engine proxy:

Note This feature is not available on No Payload Encryption models.

Model	License Requirement
All models	Intercompany Media Engine license. When you enable the Intercompany Media Engine (IME) license, you can use TLS proxy sessions up to the configured TLS proxy limit. If you also have a Unified Communications (UC) license installed that is higher than the default TLS proxy limit, then the ASA sets the limit to be the UC license limit plus an additional number of sessions depending on your model. You can manually configure the TLS proxy limit using the **tls-proxy maximum-sessions** command. To view the limits of your model, enter the **tls-proxy maximum-sessions ?** command. If you also install the UC license, then the TLS proxy sessions available for UC are also available for IME sessions. For example, if the configured limit is 1000 TLS proxy sessions, and you purchase a 750-session UC license, then the first 250 IME sessions do not affect the sessions available for UC. If you need more than 250 sessions for IME, then the remaining 750 sessions of the platform limit are used on a first-come, first-served basis by UC and IME. • For a license part number ending in “K8”, TLS proxy sessions are limited to 1000. • For a license part number ending in “K9”, the TLS proxy limit depends on your configuration and the platform model. **Note** K8 and K9 refer to whether the license is restricted for export: K8 is unrestricted, and K9 is restricted. You might also use SRTP encryption sessions for your connections: • For a K8 license, SRTP sessions are limited to 250. • For a K9 license, there is no limit. **Note** Only calls that require encryption/decryption for media are counted toward the SRTP limit; if passthrough is set for the call, even if both legs are SRTP, they do not count toward the limit.

For more information about licensing, see the general operations configuration guide.

CHAPTER 15

Using the Cisco Unified Communication Wizard

This chapter describes how to configure the adaptive security appliance for Cisco Unified Communications Proxy features.

This chapter includes the following sections:

Information about the Cisco Unified Communication Wizard

Note The Unified Communication Wizard is supported for the ASA version 8.3(1) and later.

The Unified Communication Wizard assists you in configuring the following Unified Communications proxies on the ASA:

- Cisco Phone Proxy

  See Configuring the Phone Proxy by using the Unified Communication Wizard, page 15-4.

- Cisco Mobility Advantage Proxy

  See Configuring the Mobility Advantage by using the Unified Communication Wizard, page 15-11.

- Cisco Presence Federation Proxy

  See Configuring the Presence Federation Proxy by using the Unified Communication Wizard, page 15-14.

- Cisco Intercompany Media Engine Proxy

  See Configuring the UC-IME by using the Unified Communication Wizard, page 15-16.

The wizard simplifies the configuration of the Unified Communications proxies in the following ways:

- You enter all required data in the wizard steps. You are not required to navigate various ASDM screens to configure the Unified Communications proxies.
- The wizard generates configuration settings for the Unified Communications proxies where possible, automatically, without requiring you to enter data. For example, the wizard configures the required ACLs, IP address translation (NAT and PAT) statements, self-signed certificates, TLS proxies, and application inspection.
- The wizard displays network diagrams to illustrate data collection.

To access the Unified Communication Wizard, choose one of the following paths in the main ASDM application window:

- **Wizards > Unified Communication Wizard**.
- **Configuration > Firewall > Unified Communications,** and then click **Unified Communication Wizard**.

Phone Proxy: Secure remote access for Cisco encrypted endpoints, and VLAN traversal for Cisco softphones

The phone proxy feature enables termination of Cisco SRTP/TLS-encrypted endpoints for secure remote access. The phone proxy allows large scale deployments of secure phones without a large scale VPN remote access hardware deployment. End-user infrastructure is limited to just the IP endpoint, without VPN tunnels or hardware.

The Cisco adaptive security appliance phone proxy is the replacement product for the Cisco Unified Phone Proxy. Additionally, the phone proxy can be deployed for voice/data VLAN traversal for softphone applications. Cisco IP Communicator (CIPC) traffic (both media and signaling) can be proxied through the ASA, thus traversing calls securely between voice and data VLANs.

For information about the differences between the TLS proxy and phone proxy, go to the following URL for Unified Communications content, including TLS Proxy vs. Phone Proxy white paper:

http://www.cisco.com/go/secureuc

Mobility Advantage Proxy: Secure connectivity between Cisco Mobility Advantage server and Cisco Unified Mobile Communicator clients

Cisco Mobility Advantage solutions include the Cisco Unified Mobile Communicator (Cisco UMC), an easy-to-use software application for mobile handsets that extends enterprise communications applications and services to mobile phones and the Cisco Unified Mobility Advantage (Cisco UMA) server. The Cisco Mobility Advantage solution streamlines the communication experience, enabling single number reach and integration of mobile endpoints into the Unified Communications infrastructure.

The security appliance acts as a proxy, terminating and reoriginating the TLS signaling between the Cisco UMC and Cisco UMA. As part of the proxy security functionality, inspection is enabled for the Cisco UMA Mobile Multiplexing Protocol (MMP), the protocol between Cisco UMC and Cisco UMA.

Presence Federation Proxy: Secure connectivity between Cisco Unified Presence servers and Cisco/Microsoft Presence servers

Cisco Unified Presence solution collects information about the availability and status of users, such as whether they are using communication devices, such as IP phones at particular times. It also collects information regarding their communications capabilities, such as whether web collaboration or video conferencing is enabled. Using user information captured by Cisco Unified Presence, applications such as Cisco Unified Personal Communicator and Cisco UCM can improve productivity by helping users connect with colleagues more efficiently through determining the most effective way for collaborative communication.

Using the ASA as a secure presence federation proxy, businesses can securely connect their Cisco Unified Presence (Cisco UP) servers to other Cisco or Microsoft Presence servers, enabling intra-enterprise communications. The security appliance terminates the TLS connectivity between the servers, and can inspect and apply policies for the SIP communications between the servers.

Cisco Intercompany Media Engine Proxy: Secure connectivity between Cisco UCM servers in different enterprises for IP Phone traffic

As more unified communications are deployed within enterprises, cases where business-to-business calls utilize unified communications on both sides with the Public Switched Network (PSTN) in the middle become increasingly common. All outside calls go over circuits to telephone providers and from there are delivered to all external destinations.

The Cisco Intercompany Media Engine (UC-IME) gradually creates dynamic, encrypted VoIP connections between businesses, so that a collection of enterprises that work together end up looking like one giant business with secure VoIP interconnections between them.

There are three components to a Cisco Intercompany Media Engine deployment within an enterprise: a Cisco Intercompany Media Engine server, a call agent (the Cisco Unified Communications Manager) and an ASA running the Cisco Intercompany Media Engine Proxy.

The ASA provides perimeter security by encrypting signaling connections between enterprises and preventing unauthorized calls. An ASA running the Cisco Intercompany Media Engine Proxy can either be deployed as an Internet firewall or be designated as a Cisco Intercompany Media Engine Proxy and placed in the DMZ, off the path of the regular Internet traffic.

Licensing Requirements for the Unified Communication Wizard

To run the Unified Communication Wizard in ASDM, you require the following license:

Model	License Requirement
All models	Base License

However, to run each of the Unified Communications proxy features created by the wizard, you must have the appropriate Unified Communications Proxy licenses.

The Cisco Unified Communications proxy features supported by the ASA require a Unified Communications Proxy license:

- Cisco Phone Proxy
- TLS proxy for encrypted voice inspection
- Presence Federation Proxy
- Cisco Intercompany Media Engine Proxy

See Licensing for Cisco Unified Communications Proxy Features, page 14-4 for more information.

Note The Cisco Intercompany Media Engine Proxy does not appear as an option in the Unified Communication Wizard unless the license required for this proxy is installed on the ASA.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Supports IPv6 addresses.

Additional Guidelines and Limitations

Using the Unified Communication Wizard to create the Unified Communications proxies has the following limitations and requirements:

- You must configure at least two interfaces on the ASA to use the UC Wizard to configure a Unified Communications proxy.
- For all Unified Communications proxies to function correctly, you must synchronize the clock on the ASA and all servers associated with each proxy, such as the Cisco Unified Communication Manager server, the Cisco Mobility Advantage server, the Cisco Unified Presence server, and the Cisco Intercompany Media Engine server.
- When you configure the Cisco Intercompany Media Engine Proxy for an off-path deployment, you must ensure that the public IP addresses and ports of the Cisco Unified Communications Manager servers and the public IP address for the media termination address are accessible from the Internet. The summary page of the Unified Communication Wizard reminds you of the requirements.
- If the ASA on which you configure the Cisco Mobility Advantage Proxy and the Cisco Presence Federation Proxy is located behind another firewall, you must ensure that the public IP addresses for the Cisco Mobility Advantage server and the Cisco Unified Presence server are accessible from the Internet.
- If you use the Unified Communication Wizard to create to the Presence Federation Proxy and the Cisco Intercompany Media Engine Proxy, you might be required to adjust the configuration of the ACLs created automatically by the wizard for each proxy. See Chapter 19, “Configuring Cisco Unified Presence” and Chapter 20, “Configuring Cisco Intercompany Media Engine Proxy”, respectively, for information about the ACL requirements required by each proxy.

Configuring the Phone Proxy by using the Unified Communication Wizard

To configure the Cisco Unified Presence proxy by using ASDM, choose Wizards > Unified Communications Wizard from the menu. The Unified Communications Wizard opens. From the first page, select the Phone Proxy option under the Remote Access section.

The wizard automatically creates the necessary TLS proxy, then guides you through creating the Phone Proxy instance, importing and installing the required certificates, and finally enables the SIP and SCCP inspection for the Phone Proxy traffic automatically.

Note Any configuration created by the wizard should be maintained through the wizard to ensure proper synchronization. For example, if you create a phone proxy configuration through the UC wizard and then modify the configuration outside of the wizard, the rest of the wizard configuration is not updated, and the wizard configuration is not synchronized.

Therefore, if you choose to change some part of the phone proxy configuration outside of the wizard, it is your responsibility to keep the rest of the configuration in synchronization.

The wizard guides you through four steps to configure the Phone Proxy:

Step 1 Select the Phone Proxy option.

Step 2 Specify settings to define the Cisco Unified Communications Manager (UCM) servers and TFTP servers, such the IP address and the address translation settings of each server, and the Cisco UCM cluster security mode. See Configuring the Private Network for the Phone Proxy, page 15-5 and Configuring Servers for the Phone Proxy, page 15-6.

Step 3 If required, enable Certificate Authority Proxy Function (CAPF). See Enabling Certificate Authority Proxy Function (CAPF) for IP Phones, page 15-8.

Step 4 Configure the public IP phone network, such as address translation settings for remote IP phones, whether to enable service setting for IP phones, and the HTTP proxy used by the IP phones. Configuring the Public IP Phone Network, page 15-9

Step 5 Specify the media termination address settings of the Cisco UCM. Configuring the Media Termination Address for Unified Communication Proxies, page 15-10.

The wizard completes by displaying a summary of the configuration created for Phone Proxy.

Configuring the Private Network for the Phone Proxy

The values that you specify in this page configure the connection from the ASA to the Cisco UCMs and TFTP servers by creating the necessary address translation settings and access control list entries.

Additionally, you specify the security mode for the Cisco UCM cluster. In a nonsecure cluster mode or a mixed mode where the phones are configured as nonsecure, the phone proxy behaves in the following ways:

- The TLS connections from the phones are terminated on the ASA and a TCP connection is initiated to the Cisco UCM.
- SRTP sent from external IP phones to the internal network IP phone via the ASA is converted to RTP.

In a mixed mode cluster where the internal IP phones are configured as authenticated, the TLS connection is not converted to TCP to the Cisco UCM but the SRTP is converted to RTP.

In a mixed mode cluster where the internal IP phone is configured as encrypted, the TLS connection remains a TLS connection to the Cisco UCM and the SRTP from the remote phone remains SRTP to the internal IP phone.

Step 1 From the Interface drop-down list, choose the interface on which the ASA listens for the Cisco UCM servers and TFTP servers. The Cisco UCM servers and TFTP servers must reside on the same interface.

Step 2 Specify each entity in the network (all Cisco UCM and TFTP servers) that the IP phones must trust. Click **Add** to add the servers. See Configuring Servers for the Phone Proxy, page 15-6.

To modify the configuration of a server already added to the configuration, select the server in the table and click **Edit**. The Edit Server dialog appears. See Configuring Servers for the Phone Proxy, page 15-6. At least one Cisco UCM and at least one TFTP server must be configured for the phone proxy.

Step 3 Specify the security mode of the Cisco UCM cluster by clicking one of the following options in the Unified CM Cluster Mode field:

- Non-secure—Specifies the cluster to be in nonsecure mode when configuring the Phone Proxy feature.
- Mixed—Specifies the cluster to be in mixed mode when configuring the Phone Proxy feature.

  If you selected the Mixed security mode, the Generate and Export LDC Certificate button becomes available.

Step 4 For a Mixed security mode only, configure local dynamic certificates (LDC) for the IP phones by performing the following steps:

a. Click the **Generate and Export LDC Certificate** button.

   A dialog box appears stating "Enrollment succeeded," which indicates that the LDC was generated.

b. Click **OK** to close the Enrollment Status dialog box. The Export certificate dialog box appears.

c. In the Export to File field, enter the file name and path for the LDC or click browse to locate and select an existing file.

d. Click the **Export Certificate** button. A dialog box appears indicating that the file was exported successfully.

e. Click **OK** to close the dialog box. A dialog box appears reminding you to install the LDC on the Cisco UCMs.

f. Click **OK** to close the dialog box.

   Once configured, the ASA presents this unique, dynamically-created certificate to the Cisco UCM on behalf of the IP phones.

Step 5 Click **Next**.

Configuring Servers for the Phone Proxy

The values that you specify in this page generate address translation settings, access list entries, trustpoints, and the corresponding CTL file entries for each server.

You must add a server for each entity in the network that the IP phones must trust. These servers include all Cisco UCM servers in the cluster and all the TFTP servers.

You must add at least one TFTP server and at least one Cisco UCM server for the phone proxy. You can configure up to five TFTP servers for the phone proxy. The TFTP server is assumed to be behind the firewall on the trusted network; therefore, the phone proxy intercepts the requests between the IP phones and TFTP server.

Note When you delete a TFTP server from the Server list in Step 2 of the wizard, ASDM deletes only the TFTP server IP address from the configuration and does not remove from the configuration all the ACLs, NAT statements, object groups, etc. attached to the TFTP server. To remove those attached configuration statements, you must delete them manually by using the appropriate area of ASDM or rerun the Unified Communications wizard without making any changes and apply the configuration to to remove these statements.

The servers that the IP phones must trust can be deployed on the network in one of the following ways:

- All the services required by the Cisco UCM server, namely the Cisco UCM, TFTP, and CAPF services, are running on one server. In this deployment, only one instance of each service exists. For this deployment, you can select Unified CM+ TFTP as the server type. You can either use Address only or Address and ports for address translation. Cisco recommends that you specify Address and ports for increased security.
- Deployments for larger enterprises might have redundant Cisco UCMs and dedicated servers for TFTP and CAPF services. In that type of deployment, use Address only for voice address translation and Address only or Address and ports for TFTP.

Table 15-1 lists the ports that are configured for Address and port translation by default:

Table 15-1 Port Configuration

Address	Default Port	Description
TFTP Server	69	Allows incoming TFTP
Cisco UCM	2000	Allows incoming non-secure SCCP
Cisco UCM	2443	Allows incoming secure SCCP
Cisco UCM	5061	Allows incoming secure SIP

Step 1 In the Server Type field, select the server from the drop-down list: Unified CM, TFTP, or Unified CM + TFTP. Select Unified CM + TFTP when the Cisco UCM and TFTP server reside on the same device.

Note Depending on which type of server you select (Unified CM or TFTP), only the necessary fields in this dialog box become available. Specifically, if the server type is Unified CM, the TFTP section in the dialog is unavailable. If the server type is TFTP, the Voice section is unavailable.

Step 2 In the Private Address field, specify the actual internal IP address of the server.

Step 3 In the FQDN field, enter the fully-qualified domain name of the server, which includes the hostname and domain name; for example, `ucm.cisco.com` (where `ucm` is the hostname and `cisco.com` is the domain name).

If you are configuring a Unified CM server, enter the fully-qualified domain name configured on the Cisco UCM.

If you are configuring a TFTP server, only specify the TFTP server fully-qualified domain name when that server is configured with FQDN. If the TFTP server is not configured with FQDN, you can leave the field blank.

Note Entering the fully-qualified domain name allows the ASA to perform hostname resolution when DNS lookup is not configured on the ASA or the configured DNS servers are unavailable.See the command reference for information about the **dns domain-lookup** command.

Step 4 In the Address Translation section, select whether to use the interface IP address or to enter a different IP address.

Selecting the Use interface IP radio button configures the server to use the IP address of the public interface. You select the public interface in step 4 of the wizard when you configure the public network for the phone proxy.

If the Use interface IP radio button is selected, you must specify port translation settings in the Voice and TFTP sections. Address-only translation is available only when you specify an IP address other than the IP address of the public interface.

When you select the Address only radio button, the ASA performs address translation on all traffic between the server and the IP phones. Selecting the Address and ports radio button limits address translation to the specified ports.

Step 5 (Unified CM or Unified CM + TFTP servers only) In the Voice section, configure inspection of SIP or SCCP protocol traffic, or both SIP and SCCP protocol traffic by completing the following fields:

a. In the Translation Type field, specify whether to use the Address only or the Address and ports.

When the deployment has redundant Cisco UCM servers and dedicated servers for TFTP and CAPF services, select Address only for voice address translation.

Select the Address and ports option when you want to limit address translation to the specified ports.

b. In the Voice Protocols field, select the inspection protocols supported by the IP phones deployed in the enterprise. Depending on which inspection protocols you select—SCCP, SIP, or SCCP and SIP—only the ports fields for the selected voice protocols are available.

c. In the Port Translation section, enter the private and public ports for the voice protocols.

The default values for the voice ports appear in the text fields. If necessary, change the private ports to match the settings on the Cisco UCM. The values you set for the public ports are used by the IP phones to traverse the ASA and communicate with the Cisco UCM.

The secure SCCP private port and public port are automatically configured. These port numbers are automatically set to the value of the non-secure port number plus 443.

Step 6 (TFTP or Unified CM + TFTP servers only) In the TFTP section, you can select either Address only or Address and port for address translation. Cisco recommends that you specify Address and port for increased security. Specifying Address and port configures the TFTP server to listen on port 69 for TFTP requests.

When the server type is Unified CM + TFTP, the wizard configures the same type of address translation for Voice and TFTP; for example, when the server type is Unified CM + TFTP and the Address only option is selected, the wizard creates a global address translation rule for all traffic to and from the server. In this case, configuring port translation for the TFTP server would be redundant.

Step 7 Click **OK** to add the server to the phone proxy configuration and return to step 2 of the wizard.

Enabling Certificate Authority Proxy Function (CAPF) for IP Phones

As an alternative to authenticating remote IP phones through the TLS handshake, you can configure authentication via locally significant certificate (LSC) provisioning. With LSC provisioning, you create a password for each remote IP phone user and each user enters the password on the remote IP phones to retrieve the LSC.

Because using LSC provisioning to authenticate remote IP phones requires the IP phones first register in nonsecure mode, Cisco recommends LSC provisioning be done inside the corporate network before giving the IP phones to end-users. Otherwise, having the IP phones register in nonsecure mode requires the Administrator to open the nonsecure signaling port for SIP and SCCP on the ASA.

See also the Cisco Unified Communications Manager Security Guide for information on Using the Certificate Authority Proxy Function (CAPF) to install a locally significant certificate (LSC).

If your network includes Cisco IP Communicators (CIPC) or you have LSC enabled IP phones, you must import the CAPF certificate from the Cisco UCM. The certificate will be used to generate the LSC on the IP phones.

If the Cisco UCM has more than one CAPF certificate, you must import all of them to the ASA. However, the wizard supports configuring only one CAPF certificate, which is the default. To import more than one CAPF certificate, go to Configuration > Device Management > Certificate Management > Identity Certificates.

You can configure LSC provisioning for additional end-user authentication. See the Cisco Unified Communications Manager configuration guide for information.

Step 1 Check the **Enable Certificate Authority Proxy Function** check box. The remaining fields in the page become available.

Step 2 Enter the private IP address of the LSC provider.

Step 3 In the Public Address field, specify whether to use the IP address of the ASA public interface or enter an IP address.

Specifying the private and public IP addresses for the LSC provider, creates an access list entry that allows the IP phones to contact the Cisco UCM by opening the CAPF port for LSC provisioning.

Step 4 In the Translation Type field, select the Address only or Address and ports radio button.

The IP phones must contact the CAPF service on the Cisco UCM. The address translation type (Address only versus Address and ports) you select for CAPF must match the address translation type of the Cisco UCM on which the CAPF service is running. You set the address translation type for that Cisco UCM server in the previous step of this wizard (see Configuring Servers for the Phone Proxy, page 15-6),

By default, the CAPF Service uses port 3804. Modify this default value only when it is modified on the Cisco UCM.

Step 5 If you selected the Address and ports radio button, enter the private and public ports for the CAPF service.

Step 6 Click the **Install CAPF Certificate** button. The Install Certificate dialog box appears. See Installing a Certificate, page 15-23.

Step 7 Click **Next**.

Configuring the Public IP Phone Network

The values that you specify in this page generate the address translation rules used for the IP phones and configure how the ASA handles IP phone settings.

Step 1 From the Interface drop-down list, choose the interface on which the ASA listens for connections from IP phones.

Step 2 To preserve Call Manager configuration on the IP phones, check the Preserve the Unified CM's configuration on the phone's service check box. When this check box is uncheck, the following service settings are disabled on the IP phones:

- Web Access

- PC Port
- Voice VLAN access
- Gratuitous ARP
- Span to PC Port

Step 3 To configure address translation for IP phones, check the Enable address translation for IP phones check box. Select whether to use the IP address of the ASA private interface (which you selected in step 2 of the wizard) or enter an IP address.

Configuring address translation for IP phone configures the address used by the IP phones. All traffic from the outside network converges into one source IP address so that, if there is another corporate firewall in the network, a pinhole needs to be opened only for that IP address rather than for all traffic.

Step 4 To configure an HTTP proxy for the Phone Proxy feature that is written into the IP phone's configuration file under the <proxyServerURL> tag, do the following:

a. Check the Configure an HTTP proxy to redirect phone URLs... check box.

b. In the IP Address field, type the IP address of the HTTP proxy

c. In the Port field, enter the listening port of the HTTP proxy.

The IP address you enter should be the global IP address based on where the IP phone and HTTP proxy server is located. You can enter a hostname in the IP Address field when that hostname can be resolved to an IP address by the adaptive security appliance (for example, DNS lookup is configured) because the adaptive security appliance will resolve the hostname to an IP address. If a port is not specified, the default will be 8080.

d. In the Interface field, select the interface on which the HTTP proxy resides on the adaptive security appliance.

Setting the proxy server configuration option for the Phone Proxy allows for an HTTP proxy on the DMZ or external network in which all the IP phone URLs are directed to the proxy server for services on the phones. This setting accommodates nonsecure HTTP traffic, which is not allowed back into the corporate network.

Step 5 Click **Next**.

Configuring the Media Termination Address for Unified Communication Proxies

The data from this step generates the MTA instance to be added to the Phone Proxy and the UC-IME proxy.

The phone proxy and the UC-IME proxy use the media termination address for Secure RTP (SRTP) and RTP traffic. SRTP traffic sent from external IP phones to the internal network IP phone via the ASA is converted to RTP traffic. The traffic is terminated on the adaptive security appliance. SRTP provides message authentication and replay protection to Internet media traffic such as audio and video. RTP defines a standardized packet format for delivering audio and video over the Internet.

For the UC-IME proxy and the Phone Proxy to be fully functional, you must ensure that the public IP address for the media termination address (MTA) is accessible from the Internet. The summary page of the Unified Communication Wizard reminds you of this requirement.

The MTA IP addresses that you specify must meet specific requirements. See Media Termination Instance Prerequisites, page 16-6 for information.

Step 1 In the field for the private IP address, enter the IP address on which private media traffic terminates. The IP address must be within the same subnet as the private interface IP address. The correct subnet range is provided to the right of the field for the private IP address.

Step 2 In the field for the public IP address, enter the IP address on which public media traffic terminates. The IP address must be within the same subnet as the public interface IP address. The correct subnet range is provided to the right of the field for the public IP address.

Step 3 Specify the minimum and maximum values for the RTP port range for the media termination instance.

Port values must be within the range of 1024 to 65535.

Step 4 Click **Next**.

The wizard completes by displaying a summary of the configuration created for proxy.

Configuring the Mobility Advantage by using the Unified Communication Wizard

Note The Unified Communication Wizard is supported for the ASA version 8.3(1) and later.

The Unified Communication wizard guides you through the steps to configure the Mobility Advantage proxy. Choose **Wizards** > **Unified Communication Wizard** from the menu. The Unified Communication Wizard opens. Click the Cisco Mobility Advantage Proxy radio button under the Remote Access section.

When using the wizard to create the Mobility Advantage proxy, ASDM automatically creates the necessary TLS proxies, enables MMP inspection for the Mobility Advantage traffic, generates address translation (NAT) statements, and creates the access rules that are necessary to allow traffic between the Cisco Mobility Advantage server and the mobility clients.

The following steps provide the high-level overview for configuring the Mobility Advantage proxy:

Step 1 Specify settings to define the private and public network topology, such the public and private network interfaces, and the IP addresses of the Cisco Mobility Advantage server. See Configuring the Topology for the Cisco Mobility Advantage Proxy, page 15-12.

Step 2 Configure the certificates that are exchanged between the Cisco Mobility Advantage server and the ASA. See Configuring the Server-Side Certificates for the Cisco Mobility Advantage Proxy, page 15-12.

Step 3 Configure the client-side certificate management, namely the certificates that are exchanged between the Unified Mobile Communicator clients and the ASA. See Configuring the Client-Side Certificates for the Cisco Mobility Advantage Proxy, page 15-13.

The wizard completes by displaying a summary of the configuration created for Mobility Advantage Proxy.

Configuring the Topology for the Cisco Mobility Advantage Proxy

When configuring the Mobility Advantage Proxy, you specify settings to define the private and public network topology, such the private and public network interfaces, and the private and public IP addresses of the Cisco Mobility Advantage server.

The values that you specify in this page generate the following configuration settings for the Mobility Advantage Proxy:

- Static PAT for the Cisco Mobility Advantage server
- Static NAT for Cisco Unified Mobile Communicator clients if the Enable address translation for Mobility clients check box is checked.
- ACLs to allow Cisco Unified Mobile Communicator clients to access the Cisco Mobility Advantage server

Step 1 In the Private Network area, choose the interface from the drop-down list.

Step 2 In the Unified MA Server area, enter the private and public IP address for the Cisco Mobility Advantage server. Entering ports for these IP addresses is optional. By default port number 5443 is entered, which is the default TCP port for MMP inspection.

Step 3 In the FQDN field, enter the domain name for the Cisco Mobility Advantage server. This domain name is included in the certificate signing request that you generate later in this wizard.

Step 4 In the Public Network area, choose an interface from the drop-down list.

The proxy uses this interface for configuring static PAT for the Cisco Mobility Advantage server and the ACLs to allow Cisco Unified Mobile Communicator clients to access the Cisco Mobility Advantage server.

Step 5 **To configure whether address translation (NAT) is used by** Cisco Unified Mobile Communicator clients, check the **Enable address translation for Mobility clients** check box and choose whether to use the IP address of the public interface or whether to enter an IP address.

Step 6 Click **Next**.

Configuring the Server-Side Certificates for the Cisco Mobility Advantage Proxy

A trusted relationship between the ASA and the Cisco UMA server can be established with self-signed certificates. The ASA's identity certificate is exported, and then uploaded on the Cisco UMA server truststore. The Cisco UMA server certificate is downloaded, and then uploaded on the ASA truststore.

The supports using self-signed certificates only at this step.

Step 1 In the ASA’s Identity Certificate area, click **Generate and Export ASA’s Identity Certificate**.

An information dialog boxes appear indicating that the enrollment seceded. In the Enrollment Status dialog box, click **OK**. The Export certificate dialog box appears.

Note

- If an identity certificate for the ASA has already been created, the button in this area appears as **Export ASA’s Identity Certificate** and the Export certificate dialog box immediately appears.

- When using the wizard to configure the Cisco Mobility Advantage proxy, the wizard only supports installing self-signed certificates.

Step 2 Export the identity certificate generated by the wizard for the ASA. See Exporting an Identity Certificate, page 15-23.

Step 3 In the Unified MA Server's Certificate area, click **Install Unified MA Server's Certificate**. The Install Certificate dialog appears.

Step 4 Locate the file containing the Cisco Mobility Advantage server certificate or paste the certificate details in the dialog box. See Installing a Certificate, page 15-23.

Step 5 Click **Next**.

Note See the Cisco Mobility Advantage server documentation for information on how to export the certificate for this server.

Configuring the Client-Side Certificates for the Cisco Mobility Advantage Proxy

To establish a trust relationship between the Cisco Unified Mobile Communicator (UMC) clients and the ASA, the ASA uses a CA-signed certificate that is configured with the Cisco Mobility Advantage server's FQDN (also referred to as certificate impersonation).

In the Client-Side Certificate Management page, you enter both the intermediate CA certificate (if applicable, as in the cases of Verisign) and the signed ASA identity certificate.

Note If the ASA already has a signed identity certificate, you can skip Step 1 in this procedure and proceed directly to Step 2.

Step 1 In the ASA's Identity Certificate area, click **Generate CSR**. The CSR parameters dialog box appears.

For information about specifying additional parameters for the certificate signing request (CSR), see Generating a Certificate Signing Request (CSR) for a Unified Communications Proxy, page 15-24.

Information dialog boxes appear indicating that the wizard is delivering the settings to the ASA and retrieving the certificate key pair information. The Identity Certificate Request dialog box appears.

For information about saving the CSR that was generated and submitting it to a CA, see Saving the Identity Certificate Request, page 15-25.

Step 2 Click **Install ASA's Identity Certificate**. Install the certificate. See Installing the ASA Identity Certificate on the Mobility Advantage Server, page 15-26.

Step 3 Click **Install Root CA's Certificate**. The Install Certificate dialog box appears. Install the certificate. See Installing a Certificate, page 15-23.

Step 4 Click **Next**.

The wizard completes by displaying a summary of the configuration created for Mobility Advantage Proxy.

Configuring the Presence Federation Proxy by using the Unified Communication Wizard

Note The Unified Communication Wizard is supported for the ASA version 8.3(1) and later.

To configure the Cisco Unified Presence proxy by using ASDM, choose **Wizards** > **Unified Communication Wizard** from the menu. The Unified Communication Wizard opens. From the first page, select the Cisco Unified Presence Proxy option under the Business-to-Business section.

When using the wizard to create the Cisco Presence Federation proxy, ASDM automatically creates the necessary TLS proxies, enables SIP inspection for the Presence Federation traffic, generates address translation (static PAT) statements for the local Cisco Unified Presence server, and creates ACLs to allow traffic between the local Cisco Unified Presence server and remote servers.

The following steps provide the high-level overview for configuring the Presence Federation Proxy:

Step 1 Specify settings to define the private and public network topology, such the private and public IP address of the Presence Federation server. See Configuring the Topology for the Cisco Presence Federation Proxy, page 15-14.

Step 2 Configure the local-side certificate management, namely the certificates that are exchanged between the local Unified Presence Federation server and the ASA. See Configuring the Local-Side Certificates for the Cisco Presence Federation Proxy, page 15-15.

Step 3 Configure the remote-side certificate management, namely the certificates that are exchanged between the remote server and the ASA. See Configuring the Remote-Side Certificates for the Cisco Presence Federation Proxy, page 15-15.

The wizard completes by displaying a summary of the configuration created for the Presence Federation proxy.

Configuring the Topology for the Cisco Presence Federation Proxy

When configuring the Presence Federation Proxy, you specify settings to define the private and public network topology, such the private and public network interfaces, and the private and public IP addresses of the Cisco Unified Presence server.

The values that you specify in this page generate the following configuration settings for the Presence Federation Proxy:

- Static PAT for the local Cisco Unified Presence server
- ACLs for traffic between the local Cisco Unified Presence server and remote servers

Step 1 In the Private Network area, choose the interface from the drop-down list.

Step 2 In the Unified Presence Server area, enter the private and public IP address for the Unified Presence server. Entering ports for these IP addresses is optional. By default port number 5061 is entered, which is the default TCP port for SIP inspection.

Step 3 In the FQDN field, enter the domain name for the Unified Presence server. This domain name is included in the certificate signing request that you generate later in this wizard.

Step 4 In the Public Network area, choose the interface of the public network from the drop-down list. The proxy uses this interface for configuring static PAT for the local Cisco Unified Presence server and for configuring ACLs to allow remote servers to access the Cisco Unified Presence server.

Step 5 Click **Next**.

Configuring the Local-Side Certificates for the Cisco Presence Federation Proxy

Within an enterprise, setting up a trust relationship is achievable by using self-signed certificates. The supports using self-signed certificates only at this step.

Step 1 In the ASA's Identity Certificate area, click **Generate and Export ASA's Identity Certificate**.

An information dialog box appears indicating that enrollment succeeded. In the Enrollment Status dialog box, click **OK**. The Export certificate dialog box appears.

Note

- If an identity certificate for the ASA has already been created, the button in this area appears as **Export ASA's Identity Certificate** and the Export certificate dialog box immediately appears.
- When using the wizard to configure the Cisco Presence Federation proxy, the wizard only supports installing self-signed certificates.

Step 2 Export the identity certificate generated by the wizard for the ASA. See Exporting an Identity Certificate, page 15-23.

Step 3 Local Unified Presence Server's Certificate area, click **Install Server's Certificate**. The Install Certificate dialog appears.

Step 4 Locate the file containing the Cisco Unified Presence server certificate or paste the certificate details in the dialog box. See Installing a Certificate, page 15-23.

Step 5 Click **Next**.

Note See the Cisco Unified Presence server documentation for information on how to export the certificate for this server.

Configuring the Remote-Side Certificates for the Cisco Presence Federation Proxy

Establishing a trust relationship across enterprises or across administrative domains is key for federation. Across enterprises you must use a trusted third-party CA (such as, VeriSign). The security appliance obtains a certificate with the FQDN of the Cisco Unified Presence server (certificate impersonation).

For the TLS handshake, the two entities, namely the local entity and a remote entity, could validate the peer certificate via a certificate chain to trusted third-party certificate authorities. The local entity and the remote entity enroll with the CAs. The ASA as the TLS proxy must be trusted by both the local and remote entities. The security appliance is always associated with one of the enterprises. Within that enterprise, the entity and the security appliance authenticate each other by using a self-signed certificate.

To establish a trusted relationship between the security appliance and the remote entity, the security appliance can enroll with the CA on behalf of the Cisco Unified Presence server for the local entity. In the enrollment request, the local entity identity (domain name) is used.

To establish the trust relationship, the security appliance enrolls with the third party CA by using the Cisco Unified Presence server FQDN as if the security appliance is the Cisco Unified Presence server.

Note If the ASA already has a signed identity certificate, you can skip Step 1 in this procedure and proceed directly to Step 2.

Step 1 In the ASA's Identity Certificate area, click **Generate CSR**. The CSR parameters dialog box appears.

For information about specifying additional parameters for the certificate signing request (CSR), see Generating a Certificate Signing Request (CSR) for a Unified Communications Proxy, page 15-24.

Information dialog boxes appear indicating that the wizard is delivering the settings to the ASA and retrieving the certificate key pair information. The Identity Certificate Request dialog box appears.

For information about saving the CSR that was generated and submitting it to a CA, see Saving the Identity Certificate Request, page 15-25.

Step 2 Click **Install ASA's Identity Certificate**. See Installing the ASA Identity Certificate on the Presence Federation and Cisco Intercompany Media Engine Servers, page 15-26.

Step 3 Click **Remote Server's CA's Certificate**. The Install Certificate dialog box appears. Install the certificate. See Installing a Certificate, page 15-23.

Note You must install a root CA certificate for each remote entity that communicates with the ASA because different organizations might be using different CAs.

Step 4 Click **Next**.

The wizard completes by displaying a summary of the configuration created for the Presence Federation proxy.

Configuring the UC-IME by using the Unified Communication Wizard

Note The Unified Communication Wizard is supported for the ASA version 8.3(1) and later.

To configure the Cisco Intercompany Media Engine Proxy by using ASDM, choose **Wizards** > **Unified Communication Wizard** from the menu. The Unified Communication Wizard opens. From the first page, select the Cisco Intercompany Media Engine Proxy option under the Business-to-Business section and click **Next**.

Note The Cisco Intercompany Media Engine Proxy does not appear as an option in the Unified Communication Wizard unless the license required for this proxy is installed on the ASA.

When using the wizard to create the Cisco Intercompany Media Engine Proxy, ASDM automatically creates the necessary TLS proxies, enables SIP inspection for Cisco Intercompany Media Engine traffic, generates address translation (static PAT) statements for local Cisco Unified Communications Manager servers, and creates ACLs to allow traffic between the local Cisco Unified Communications Manager servers and the remote servers.

The following steps provide the high-level overview for configuring the Cisco Intercompany Media Engine Proxy:

Step 1 Select the topology of the Cisco Intercompany Media Engine Proxy, namely whether the security appliance is an edge firewall with all Internet traffic flowing through it or whether the security appliance is off the path of the main Internet traffic (referred to as an off-path deployment). See Configuring the Topology for the Cisco Intercompany Media Engine Proxy, page 15-17.

Step 2 Specify private network settings such as the Cisco UCM IP addresses and the ticket settings. See Configuring the Private Network Settings for the Cisco Intercompany Media Engine Proxy, page 15-18.

Step 3 Specify the public network settings. See Configuring the Public Network Settings for the Cisco Intercompany Media Engine Proxy, page 15-20.

Step 4 Specify the media termination address settings of the Cisco UMC. See Configuring the Media Termination Address for Unified Communication Proxies, page 15-10.

Step 5 Configure the local-side certificate management, namely the certificates that are exchanged between the local Cisco Unified Communications Manager servers and the security appliance. See Configuring the Local-Side Certificates for the Cisco Intercompany Media Engine Proxy, page 15-21.

Step 6 Configure the remote-side certificate management, namely the certificates that are exchanged between the remote server and the ASA. This certificate is presented to remote servers so that they can authenticate the ASA as a trusted server. See Configuring the Remote-Side Certificates for the Cisco Intercompany Media Engine Proxy, page 15-22.

The wizard completes by displaying a summary of the configuration created for the Cisco Intercompany Media Engine.

Configuring the Topology for the Cisco Intercompany Media Engine Proxy

Step 1 Select the topology of your ICME deployment by clicking one of the following options:

- All Internet traffic flows through the ASA radio button. This option is also referred to as a basic deployment.
- This ASA is off the path of the regular Internet traffic. This option is also referred to as an off-path deployment.

Step 2 Click **Next**.

Basic Deployment

In a basic deployment, the Cisco Intercompany Media Engine Proxy sits in-line with the Internet firewall such that all Internet traffic traverses the ASA. In this deployment, a single Cisco UCM or a Cisco UCM cluster is centrally deployed within the enterprise, along with a Cisco Intercompany Media Engine server (and perhaps a backup). A single Internet connection traverses the ASA, which is enabled with the Cisco Intercompany Media Engine Proxy.

The ASA sits on the edge of the enterprise and inspects SIP signaling by creating dynamic SIP trunks between enterprises.

Off-path Deployment

In an off path deployment, inbound and outbound Cisco Intercompany Media Engine calls pass through an ASA enabled with the Cisco Intercompany Media Engine Proxy. The ASA is located in the DMZ and configured to support primarily Cisco Intercompany Media Engine. Normal Internet facing traffic does not flow through this ASA.

For all inbound calls, the signaling is directed to the ASA because destined Cisco UCMs are configured with the global IP address on the ASA. For outbound calls, the called party could be any IP address on the Internet; therefore, the ASA is configured with a mapping service that dynamically provides an internal IP address on the ASA for each global IP address of the called party on the Internet.

Cisco UCM sends all outbound calls directly to the mapped internal IP address on the ASA instead of the global IP address of the called party on the Internet. The ASA then forwards the calls to the global IP address of the called party.

Note When you configure the Cisco Intercompany Media Engine for an off-path deployment, you must ensure that the public IP addresses and ports of the Cisco Unified Communications Manager servers and the public IP address for the media termination address are accessible from the Internet. The summary page of the Unified Communication Wizard reminds you of the requirements.

Configuring the Private Network Settings for the Cisco Intercompany Media Engine Proxy

When configuring the Cisco Intercompany Media Engine Proxy, you specify settings to define the private network topology, such the private network interface, the IP addresses of the Cisco Unified Communications servers, and ticket verification. Additionally, when the Cisco Unified Communications servers are operating in secure mode, you specify the X.509 subject name for the Cisco Intercompany Media Engine Proxy,

The values that you specify in this page generate the following configuration settings for the Cisco Intercompany Media Engine Proxy:

- The list of Cisco Unified Communications servers
- The ticket epoch and password used by the Cisco Intercompany Media Engine Proxy
- For an off-path deployment only, the mapping service on the same interface as the Cisco Unified Communications server

Step 1 To configure the Cisco Intercompany Media Engine Proxy as part of a basic deployment, select the interface that connects to the local Cisco Unified Communications servers.

Or

To configure the Cisco Intercompany Media Engine Proxy as part of an off-path deployment, complete the following steps:

a. From the Listening Interface drop-down list, choose the interface on which the ASA listens for the mapping requests.

b. In the Port field, enter a number between 1024 and 65535 as the TCP port on which the ASA listens for the mapping requests. The port number must be 1024 or higher to avoid conflicts with other services on the device, such as Telnet or SSH. By default, the port number is TCP 8060.

c. From the UC-IME Interface drop-down list, choose the interface that the ASA uses to connect to the remote ASA that is enabled with the Cisco Intercompany Media Engine Proxy.

Note In a basic and an off-path deployment, all Cisco Unified Communications servers must be on the same interface.

Step 2 In the Unified CM Servers area, the wizard displays the private IP address, public IP address, and security mode of any Cisco Unified Communications server configured on the ASA. If necessary, click **Add** to add a Cisco Unified Communications server. You must include an entry for each Cisco UCM in the cluster with Cisco Intercompany Media Engine that has a SIP trunk enabled.

Step 3 In the Ticket Epoch field, enter a integer from 1-255.

The epoch indicates the number of times that password has changed. When the proxy is configured the first time and a password entered for the first time, enter 1 for the epoch integer. Each time you change the password, increment the epoch to indicate the new password. You must increment the epoch value each time your change the password. Typically, you increment the epoch sequentially; however, the security appliance allows you to choose any value when you update the epoch.

If you change the epoch value, the current password is invalidated and you must enter a new password.

Step 4 In the Ticket Password field, enter a minimum of 10 and a maximum of 64 printable character from the US-ASCII character set. The allowed characters include 0x21 to 0x73 inclusive, and exclude the space character. The ticket password is stored onto flash.

Note We recommend a password of at least 20 characters. Only one password can be configured at a time.

The epoch and password that you configure on the ASA must match the epoch and password configured on the Cisco Intercompany Media Engine server. See the Cisco Intercompany Media Engine server documentation for information.

Step 5 In the Confirm Password field, reenter the password.

Step 6 In the X.509 Subject Name field, enter the distinguished name (DN) of the local enterprise. The name that you enter must match the name configured for the Cisco Unified Communications servers in the cluster. See the Cisco Unified Communications server documentation for information.

Step 7 Click **Next**.

Adding a Cisco Unified Communications Manager Server for the UC-IME Proxy

You must include an entry for each Cisco UCM in the cluster with Cisco Intercompany Media Engine Proxy that has a SIP trunk enabled.

Step 1 Enter the private IP address and port number (in the range 5000-6000) for the Cisco UCM server.

Step 2 In the Address Translation area, enter the public IP address for the Cisco UCM server.

Step 3 If necessary, enter the port number for the public IP address by clicking the Translate address and port radio button and entering a number (in the range 5000-6000) in the Port field.

Step 4 In the Security Mode area, click the Secure or Non-secure radio button. Specifying secure for Cisco UCM or Cisco UCM cluster indicates that Cisco UCM or Cisco UCM cluster is initiating TLS.

If you specify that some of the Cisco UCM servers are operating in secure mode, the Unified Communications Wizard includes a step in the proxy configuration to generate certificates for the local-side communication between the ASA and that Cisco UCM server. See Configuring the Local-Side Certificates for the Cisco Intercompany Media Engine Proxy, page 15-21.

Step 5 Click **OK**.

Configuring the Public Network Settings for the Cisco Intercompany Media Engine Proxy

The public network configuration depends on the deployment scenario you selected in the topology step of this wizard. Specifically, when you are configuring the UC-IME proxy as part of an off-path deployment, this step of the wizard displays fields for address translation, requiring that you specify the private IP address for the UC-IME proxy. Specifying this private IP address, translates IP addresses for inbound traffic.

In an off-path deployment, any existing ASA that you have deployed in your environment are not capable of transmitting Cisco Intercompany Media Engine traffic. Therefore, off-path signaling requires that outside addresses translate to an inside (private) IP address. The inside interface address can be used for this mapping service configuration. For the Cisco Intercompany Media Engine Proxy, the ASA creates dynamic mappings for external addresses to the internal IP address.

The values that you specify in this page generate the following configuration settings for the Cisco Intercompany Media Engine Proxy:

- Static PAT for the Cisco Unified Communications servers
- ACLs for traffic between the local and the remote servers

Step 1 In the Configure public network area, choose an interface from the Interface drop-down list.

Step 2 When configuring an off-path deployment, in the Address Translation area, specify whether to use the private IP address for the public network.

Or

Click the Specify IP address radio button and enter an IP address in the field.

Step 3 Click **Next**.

Configuring the Local-Side Certificates for the Cisco Intercompany Media Engine Proxy

Completing this step of the wizard generates a self-signed certificate for the ASA. The server proxy certificate is automatically generated using the subject name provided in an earlier step of this wizard.

The wizard supports using self-signed certificates only.

A trusted relationship between the ASA and the Cisco UMA server can be established with self-signed certificates. The certificates are used by the security appliance and the Cisco UCMs to authenticate each other, respectively, during TLS handshakes.

The ASA's identity certificate is exported, and then needs to be installed on each Cisco Unified Communications Manager (UCM) server in the cluster with the proxy and each identity certificate from the Cisco UCMs need to be installed on the security appliance.

This step in the Unified Communications Wizard only appears when the UC-IME proxy that you are creating has at least one secure Cisco Unified Communications Manager server defined. See Configuring the Topology for the Cisco Intercompany Media Engine Proxy, page 15-17 for information.

Step 1 In the ASA's Identity Certificate area, click **Generate and Export ASA's Identity Certificate**.

An information dialog boxes appear indicating that the enrollment seceded. In the Enrollment Status dialog box, click **OK**. The Export certificate dialog box appears.

Note

- If an identity certificate for the ASA has already been created, the button in this area appears as **Export ASA's Identity Certificate** and the Export certificate dialog box immediately appears.
- When using the wizard to configure the Cisco Intercompany Media Engine Proxy, the wizard only supports installing self-signed certificates.

Step 2 Export the identity certificate generated by the wizard for the ASA. See Exporting an Identity Certificate, page 15-23.

Step 3 In the Local Unified CM's Certificate area, click **Install Local Unified CM's Certificate**. The Install Certificate dialog appears.

Step 4 Locate the file containing the certificate from the Cisco Unified Communications Manager server or paste the certificate details in the dialog box. See Installing a Certificate, page 15-23. You must install the certificate from each Cisco Unified Communications Manager server in the cluster.

Step 5 Click **Next**.

Note See the Cisco Intercompany Media Engine server documentation for information on how to export the certificate for this server.

Configuring the Remote-Side Certificates for the Cisco Intercompany Media Engine Proxy

Establishing a trust relationship cross enterprises or across administrative domains is key. Cross enterprises you must use a trusted third-party CA (such as, VeriSign). The ASA obtains a certificate with the FQDN of the Cisco Unified Communications Manager server (certificate impersonation).

For the TLS handshake, the two entities could validate the peer certificate via a certificate chain to trusted third-party certificate authorities. Both entities enroll with the CAs. The ASA as the TLS proxy must be trusted by both entities. The ASA is always associated with one of the enterprises. Within that enterprise, the entity and the ASA could authenticate each other via a local CA, or by using self-signed certificates.

To establish a trusted relationship between the ASA and the remote entity, the ASA can enroll with the CA on behalf of the local enterprise. In the enrollment request, the local Cisco UCM identity (domain name) is used.

To establish the trust relationship, the ASA enrolls with the third party CA by using the Cisco Unified Communications Manager server FQDN as if the security appliance is the Cisco UCM.

Note If the ASA already has a signed identity certificate, you can skip Step 1 in this procedure and proceed directly to Step 3.

Step 1 In the ASA's Identity Certificate area, click **Generate CSR**. The CSR parameters dialog box appears.

For information about specifying additional parameters for the certificate signing request (CSR), see Generating a Certificate Signing Request (CSR) for a Unified Communications Proxy, page 15-24.

Information dialog boxes appear indicating that the wizard is delivering the settings to the ASA and retrieving the certificate key pair information. The Identity Certificate Request dialog box appears.

For information about saving the CSR that was generated and submitting it to a CA, see Saving the Identity Certificate Request, page 15-25.

Step 2 In the ASA's Identity Certificate area, click **Install ASA's Identity Certificate**. Installing the ASA Identity Certificate on the Presence Federation and Cisco Intercompany Media Engine Servers, page 15-26.

Step 3 In the Remote Server's CA's Certificate area, click **Install Remote Server's CA's Certificate**. Installing the root certificates of the CA for the remote servers is necessary so that the ASA can determine that the remote servers are trusted.

The Install Certificate dialog box appears. Install the certificate. See Installing a Certificate, page 15-23.

Note You must install the root certificates only when the root certificates for the remote servers are received from a CA other than the one that provided the identity certificate for the ASA

Step 4 Click **Next**.

The wizard completes by displaying a summary of the configuration created for the Cisco Intercompany Media Engine.

Working with Certificates in the Unified Communication Wizard

This section includes the following topics:

Exporting an Identity Certificate

The Cisco Mobility Advantage Proxy, Cisco Presence Federation Proxy, or Cisco Intercompany Media Engine Proxy require that you export the ASA identity certificate to install on the Cisco Mobility Advantage server, Cisco Presence Federation server, and Cisco Unified Communications server, respectfully.

You use the wizard to export a self-signed identity certificate. The identity certificate has all associated keys and is in PKCS12 format, which is the public key cryptography standard. When configuring a Unified Communications proxy by using the wizard, you click the Generate and Export ASA's Identify Certificate button while in the local-side or server-side certificate management step of the wizard. The Export certificate dialog box appears.

From the Export certificate dialog box, perform these steps:

Step 1 Enter the name of the PKCS12 format file to use in exporting the certificate configuration. Alternatively, click Browse to display the Export ID Certificate File dialog box to find the file to which you want to export the certificate configuration.

Step 2 Click Export Certificate to export the certificate configuration.

An information dialog box appears informing you that the certificate configuration file has been successfully exported to the location that you specified.

You complete the configuration of the Cisco Mobility Advantage Proxy, Cisco Presence Federation Proxy, or Cisco Intercompany Media Engine Proxy, you must import the generated ASA identify certificate in to the Cisco Mobility Advantage server, Cisco Presence Federation server, and Cisco Unified Communications server, respectfully, depending on which proxy you are configuring.

See the documentation for the for each of these products for information about importing an identity certificate into each.

Installing a Certificate

When configuring certificates for the Phone Proxy, Cisco Mobility Advantage Proxy, the Cisco Presence Federation Proxy, and Cisco Intercompany Media Engine Proxy, you must install the certificates from the Cisco Unified Communications Manager servers, the Cisco Mobility Advantage server, the Cisco

Presence Federation server, and the Cisco Unified Communications Manager servers, respectively, on the ASA. See the documentation for each of these products for information about obtaining the identity certificates from each.

When configuring the Cisco Phone Proxy, if LSC provisioning is required or you have LSC enabled IP phones, you must install the CAPF certificate from the Cisco UCM on the ASA. If the Cisco UCM has more than one CAPF certificate, you must import all of them to the ASA. See Enabling Certificate Authority Proxy Function (CAPF) for IP Phones, page 15-8.

Additionally, when configuring the Cisco Mobility Advantage Proxy, you use the Install Certificate dialog box to install the root certificate received from the certificate authority. The root certificate from the certificate authority is used to sign other certificates. The root certificate is used by the ASA to authenticate your signed identity certificate received from the certificate authority.

Note When using the wizard to configure the Unified Communications proxies, the wizard only supports installing self-signed certificates.

From the Install Certificate dialog box, perform these steps:

Step 1 Perform one of the following actions:

- To add a certificate configuration from an existing file, click the **Install from a file** radio button (this is the default setting). Enter the path and file name, or click **Browse** to search for the file. Then click **Install Certificate**.
- To enroll manually, click the **Paste certificate in PEM format** radio button. Copy and paste the PEM format (base64 or hexadecimal) certificate into the area provided.

Step 2 Click **Install Certificate**.

An information dialog box appears informing you that the certificate was installed on the ASA successfully.

Generating a Certificate Signing Request (CSR) for a Unified Communications Proxy

When configuring certificates for the Cisco Mobility Advantage Proxy, Cisco Presence Federation Proxy, or Cisco Intercompany Media Engine Proxy, you must generate and identity certificate request for the ASA.

Note If the ASA already has a signed identity certificate, you do not need to generate a CSR and can proceed directly to installing this certificate on the ASA. See Installing the ASA Identity Certificate on the Mobility Advantage Server, page 15-26 and Installing the ASA Identity Certificate on the Presence Federation and Cisco Intercompany Media Engine Servers, page 15-26 for the steps to install the identity certificate.

The identify certificate that you receive is presented to the following entities for each of the Unified Communication Proxies:

- Unified Mobile Communicator clients for the Cisco Mobility Advantage Proxy

- Remote Presence Federation servers for the Cisco Presence Federation Proxy
- The remote ASAfor the Cisco Intercompany Media Engine Proxy

Before generating the CSR, you can enter additional parameters.

When configuring a Unified Communications proxy by using the wizard, you click the Generate CSR button while in the client-side or remote-side certificate management step of the wizard. The CSR Parameters dialog box appears.

In the CSR Parameters dialog box, perform the following steps:

Step 1 From the Key Pair Size drop-down list, choose the size required for you certificate.

The key size that you select depends on the level of security that you want to configure and on any limitations imposed by the CA from which you are obtaining the certificate. The larger the number that you select, the higher the security level will be for the certificate. Most CAs recommend 2048 for the key modulus size; however, GoDaddy requires a key modulus size of 2048.

Step 2 (Cisco Intercompany Media Engine Proxy only) In the CN field, enter the domain name used by your enterprise or network. The subject DN you configure for the Cisco Intercompany Media Engine Proxy must match the domain name that set in the local Cisco Unified Communications Manager server.

Note For the Cisco Mobility Advantage Proxy and Cisco Presence Federation Proxy, the wizard provides the common name (CN), which is the FQDN of the Cisco Mobility Advantage server or Cisco Unified Presence server, respectively.

Step 3 In the Additional DN Attributes field, enter an attribute.

Or

Click **Select** to display the Additional DN Attributes dialog box.

a. In the Additional DN Attributes dialog box, choose an attribute from the drop-down list.

b. Enter a value for the attribute.

c. Click Add. The attribute appears in the list.

d. Click OK to return to the CSR Parameters dialog box.

The value you added appears in the Additional DN Attributes field in the CSR Parameters dialog box.

Step 4 Click **OK**.

Saving the Identity Certificate Request

After successfully generating the identity certificate request for one of the Unified Communications proxies, the Identity Certificate Request dialog box appears and prompts you to save the request.

Step 1 In the Save CSR to File field, enter the CSR file name and path; for example, c:\asa-csr.txt.

Step 2 Click **OK**. An information dialog box appears indicating the CSR was saved successfully.

Step 3 Click **OK** to close the dialog and return to the wizard.

Submit the CSR to the certificate authority (CA), for example, by pasting the CSR text into the CSR enrollment page on the CA website.

When the CA returns the signed identity certificate, rerun the Unified Communications Wizard. From the client-side or remote-side certificate management step of the wizard, click **Install ASA's Identity Certificate**. See Installing the ASA Identity Certificate on the Mobility Advantage Server, page 15-26 and Installing the ASA Identity Certificate on the Presence Federation and Cisco Intercompany Media Engine Servers, page 15-26 for the steps to install the identity certificate.

Installing the ASA Identity Certificate on the Mobility Advantage Server

When configuring certificates for the Cisco Mobility Advantage Proxy, you must install the ASA identity certificate on the Cisco Mobility Advantage server.

Typically, a certificate authority returns two certificates: your signed identity certificate and the certificate authority's certificate (referred to as the root certificate). However, some certificate authorities (for example, VeriSign) might also send you an intermediate certificate.

The root certificate from the certificate authority is used to sign other certificates. The root certificate is used by the ASA to authenticate your signed identity certificate received from the certificate authority.

If the certificate authority provided an intermediate certificate, you must enter the certificate text in the Intermediate Certificate (If Applicable) area of the Install ASA's Identity Certificate dialog box.

For the Cisco Mobility Advantage Proxy, you install the root certificate in another dialog box. See Installing a Certificate, page 15-23 for the steps to install the root certificate.

Step 1 In the Intermediate Certificate (If Applicable) area, perform on of the following actions:

- To add a certificate configuration from an existing file, click the **Install from a file** radio button (this is the default setting). Enter the path and file name, or click **Browse** to search for the file. Then click **Install Certificate**.
- To enroll manually, click the **Paste the certificate data in base-64 format** radio button. Copy and paste the PEM format (base64 or hexadecimal) certificate into the area provided.

Step 2 In the ASA's Identity Certificate area, perform on of the following actions:

- To add a certificate configuration from an existing file, click the **Install from a file** radio button (this is the default setting). Enter the path and file name, or click **Browse** to search for the file. Then click **Install Certificate**.
- To enroll manually, click the **Paste the certificate data in base-64 format** radio button. Copy and paste the PEM format (base64 or hexadecimal) certificate into the area provided.

Step 3 Click **Install Certificate**.

Installing the ASA Identity Certificate on the Presence Federation and Cisco Intercompany Media Engine Servers

When configuring certificates for the Cisco Presence Federation Proxy and Cisco Intercompany Media Engine Proxy, you must install the ASA identity certificate and the root certificate on the Cisco Presence Federation server and Cisco Intercompany Media Engine server, respectively.

Typically, a certificate authority returns two certificates: your signed identity certificate and the certificate authority's certificate (referred to as the root certificate). The root certificate from the certificate authority is used to sign other certificates. The root certificate is used by the ASA to authenticate your signed identity certificate received from the certificate authority.

Step 1 In the Root CA's Certificate area, perform on of the following actions:

- To add a certificate configuration from an existing file, click the **Install from a file** radio button (this is the default setting). Enter the path and file name, or click **Browse** to search for the file. Then click **Install Certificate**.
- To enroll manually, click the **Paste the certificate data in base-64 format** radio button. Copy and paste the PEM format (base64 or hexadecimal) certificate into the area provided.

Step 2 In the ASA's Identity Certificate area, perform on of the following actions:

- To add a certificate configuration from an existing file, click the **Install from a file** radio button (this is the default setting). Enter the path and file name, or click **Browse** to search for the file. Then click **Install Certificate**.
- To enroll manually, click the **Paste the certificate data in base-64 format** radio button. Copy and paste the PEM format (base64 or hexadecimal) certificate into the area provided.

Step 3 Click **Install Certificate**.

Configuring the Cisco Phone Proxy

This chapter describes how to configure the ASA for Cisco Phone Proxy feature.

This chapter includes the following sections:

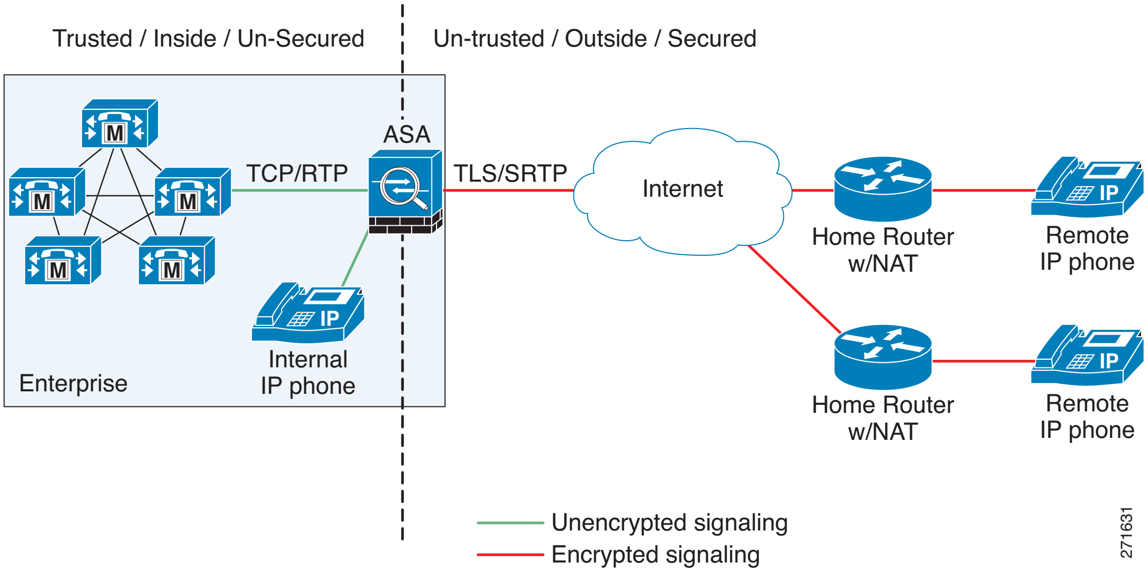
Information About the Cisco Phone Proxy

The Cisco Phone Proxy on the ASA bridges IP telephony between the corporate IP telephony network and the Internet in a secure manner by forcing data from remote phones on an untrusted network to be encrypted.

Phone Proxy Functionality

Telecommuters can connect their IP phones to the corporate IP telephony network over the Internet securely via the phone proxy without the need to connect over a VPN tunnel as illustrated by Figure 16-1.

Figure 16-1 Phone Proxy Secure Deployment



The phone proxy supports a Cisco UCM cluster in mixed mode or nonsecure mode. Regardless of the cluster mode, the remote phones that are capable of encryption are always forced to be in encrypted mode. TLS (signaling) and SRTP (media) are always terminated on the ASA. The ASA can also perform NAT, open pinholes for the media, and apply inspection policies for the SCCP and SIP protocols. In a nonsecure cluster mode or a mixed mode where the phones are configured as nonsecure, the phone proxy behaves in the following ways:

- The TLS connections from the phones are terminated on the ASA and a TCP connection is initiated to the Cisco UCM.
- SRTP sent from external IP phones to the internal network IP phone via the ASA is converted to RTP.

In a mixed mode cluster where the internal IP phones are configured as authenticated, the TLS connection is not converted to TCP to the Cisco UCM but the SRTP is converted to RTP.

In a mixed mode cluster where the internal IP phone is configured as encrypted, the TLS connection remains a TLS connection to the Cisco UCM and the SRTP from the remote phone remains SRTP to the internal IP phone.

Since the main purpose of the phone proxy is to make the phone behave securely while making calls to a nonsecure cluster, the phone proxy performs the following major functions:

- Creates the certificate trust list (CTL) file, which is used to perform certificate based authentication with remote phones.
- Modifies the IP phone configuration file when it is requested via TFTP, changes security fields from nonsecure to secure, and signs all files sent to the phone. These modifications secure remote phones by forcing the phones to perform encrypted signaling and media.
- Terminates TLS signaling from the phone and initiates TCP or TLS to Cisco UCM
- Inserts itself into the media path by modifying the Skinny and SIP signaling messages.
- Terminates SRTP and initiates RTP/SRTP to the called party.

Note As an alternative to authenticating remote IP phones through the TLS handshake, you can configure authentication via LSC provisioning. With LSC provisioning you create a password for each remote IP phone user and each user enters the password on the remote IP phones to retrieve the LSC.

Because using LSC provisioning to authenticate remote IP phones requires the IP phones first register in nonsecure mode, Cisco recommends LSC provisioning be done inside the corporate network before giving the IP phones to end-users. Otherwise, having the IP phones register in nonsecure mode requires the Administrator to open the nonsecure signaling port for SIP and SCCP on the ASA.

See "Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher, page 16-50". See also the Cisco Unified Communications Manager Security Guide for information on Using the Certificate Authority Proxy Function (CAPF) to install a locally significant certificate (LSC).

Supported Cisco UCM and IP Phones for the Phone Proxy

Cisco Unified Communications Manager

The following release of the Cisco Unified Communications Manager are supported with the phone proxy:

- Cisco Unified CallManager Version 4.*x*
- Cisco Unified CallManager Version 5.0
- Cisco Unified CallManager Version 5.1
- Cisco Unified Communications Manager 6.1
- Cisco Unified Communications Manager 7.0
- Cisco Unified Communications Manager 8.0

Cisco Unified IP Phones

The phone proxy supports these IP phone features:

- Enterprise features like conference calls on remote phones connected through the phone proxy
- XML services

The following IP phones in the Cisco Unified IP Phones 7900 Series are supported with the phone proxy:

- Cisco Unified IP Phone 7975
- Cisco Unified IP Phone 7971
- Cisco Unified IP Phone 7970
- Cisco Unified IP Phone 7965
- Cisco Unified IP Phone 7962
- Cisco Unified IP Phone 7961
- Cisco Unified IP Phone 7961G-GE
- Cisco Unified IP Phone 7960 (SCCP protocol support only)
- Cisco Unified IP Phone 7945
- Cisco Unified IP Phone 7942

- Cisco Unified IP Phone 7941
- Cisco Unified IP Phone 7941G-GE
- Cisco Unified IP Phone 7940 (SCCP protocol support only)
- Cisco Unified Wireless IP Phone 7921
- Cisco Unified Wireless IP Phone 7925

Note To support Cisco Unified Wireless IP Phone 7925, you must also configure MIC or LSC on the IP phone so that it properly works with the phone proxy.

- CIPC for softphones (CIPC versions with Authenticated mode only)

Note The Cisco IP Communicator is supported with the phone proxy VLAN Traversal in authenticated TLS mode. We do not recommend it for remote access because SRTP/TLS is not supported currently on the Cisco IP Communicator.

Note The ASA supports inspection of traffic from Cisco IP Phones running SCCP protocol version 19 and earlier.

Licensing Requirements for the Phone Proxy

The Cisco Phone Proxy feature supported by the ASA require a Unified Communications Proxy license.

The following table shows the Unified Communications Proxy license details by platform:

Note This feature is not available on No Payload Encryption models.

Model	License Requirement[1]
ASA 5505	Base License and Security Plus License: 2 sessions. *Optional license: 24 sessions.*
ASA 5510	Base License and Security Plus License: 2 sessions. *Optional licenses: 24, 50, or 100 sessions.*
ASA 5520	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, or 1000 sessions.*
ASA 5540	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, or 2000 sessions.*
ASA 5550	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*
ASA 5580	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]

Model	License Requirement[1]
ASA 5512-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, or 500 sessions.*
ASA 5515-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, or 500 sessions.*
ASA 5525-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, or 1000 sessions.*
ASA 5545-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, or 2000 sessions.*
ASA 5555-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*
ASA 5585-X with SSP-10	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*
ASA 5585-X with SSP-20, -40, or -60	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]
ASA SM	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]

[1] The following applications use TLS proxy sessions for their connections. Each TLS proxy session used by these applications (and only these applications) is counted against the UC license limit:
- Phone Proxy
- Presence Federation Proxy
- Encrypted Voice Inspection

Other applications that use TLS proxy sessions do not count towards the UC limit, for example, Mobility Advantage Proxy (which does not require a license) and IME (which requires a separate IME license).

Some UC applications might use multiple sessions for a connection. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS proxy connections, so 2 UC Proxy sessions are used.

You independently set the TLS proxy limit using the **tls-proxy maximum-sessions** command. To view the limits of your model, enter the **tls-proxy maximum-sessions ?** command. When you apply a UC license that is higher than the default TLS proxy limit, the security appliance automatically sets the TLS proxy limit to match the UC limit. The TLS proxy limit takes precedence over the UC license limit; if you set the TLS proxy limit to be less than the UC license, then you cannot use all of the sessions in your UC license.

Note: For license part numbers ending in “K8” (for example, licenses under 250 users), TLS proxy sessions are limited to 1000. For license part numbers ending in “K9” (for example, licenses 250 users or larger), the TLS proxy limit depends on the configuration, up to the model limit. K8 and K9 refer to whether the license is restricted for export: K8 is unrestricted, and K9 is restricted.

Note: If you clear the configuration (using the **clear configure all** command, for example), then the TLS proxy limit is set to the default for your model; if this default is lower than the UC license limit, then you see an error message to use the tls-proxy maximum-sessions command to raise the limit again . If you use failover and enter the **write standby** command on the primary unit to force a configuration synchronization, the **clear configure all** command is generated on the secondary unit automatically, so you may see the warning message on the secondary unit. Because the configuration synchronization restores the TLS proxy limit set on the primary unit, you can ignore the warning.

You might also use SRTP encryption sessions for your connections:
- For K8 licenses, SRTP sessions are limited to 250.
- For K9 licenses, there is not limit.

Note: Only calls that require encryption/decryption for media are counted towards the SRTP limit; if passthrough is set for the call, even if both legs are SRTP, they do not count towards the limit.

[2] With the 10,000-session UC license, the total combined sessions can be 10,000, but the maximum number of Phone Proxy sessions is 5000.

For more information about licensing, see the general operations configuration guide.

Prerequisites for the Phone Proxy

This section contains the following topics:

Media Termination Instance Prerequisites

The ASA must have a media termination instance that meets the following criteria:

- You must configure one media termination for each phone proxy on the ASA. Multiple media termination instances on the ASA are not supported.
- For the media termination instance, you can configure a global media-termination address for all interfaces or configure a media-termination address for different interfaces. However, you cannot use a global media-termination address and media-termination addresses configured for each interface at the same time.
- If you configure a media termination address for multiple interfaces, you must configure an address on each interface that the ASA uses when communicating with IP phones.

  For example, if you had three interfaces on the ASA (one internal interface and two external interfaces) and only one of the external interfaces were used to communicate with IP phones, you would configure two media termination addresses: one on the internal interface and one on the external interface that communicated with the IP phones.
- Only one media-termination address can be configured per interface.
- The IP addresses are publicly routable addresses that are unused IP addresses within the address range on that interface.
- The IP address on an interface cannot be the same address as that interface on the ASA.
- The IP addresses cannot overlap with existing static NAT pools or NAT rules.
- The IP addresses cannot be the same as the Cisco UCM or TFTP server IP address.

- For IP phones behind a router or gateway, you must also meet this prerequisite. On the router or gateway, add routes to the media termination address on the ASA interface that the IP phones communicate with so that the phone can reach the media termination address.

Certificates from the Cisco UCM

Import the following certificates which are stored on the Cisco UCM. These certificates are required by the ASA for the phone proxy.

- Cisco_Manufacturing_CA
- CAP-RTP-001
- CAP-RTP-002
- CAPF certificate (Optional)

If LSC provisioning is required or you have LSC enabled IP phones, you must import the CAPF certificate from the Cisco UCM. If the Cisco UCM has more than one CAPF certificate, you must import all of them to the ASA.

Note You can configure LSC provisioning for additional end-user authentication. See the Cisco Unified Communications Manager configuration guide for information.

See Importing Certificates from the Cisco UCM, page 16-15. For example, the CA Manufacturer certificate is required by the phone proxy to validate the IP phone certificate.

DNS Lookup Prerequisites

- If you have an fully qualified domain name (FQDN) configured for the Cisco UCM rather than an IP address, you must configure and enable DNS lookup on the ASA. For information about the **dns domain-lookup** command and how to use it to configure DNS lookup, see command reference.
- After configuring the DNS lookup, make sure that the ASA can ping the Cisco UCM with the configured FQDN.
- You must configure DNS lookup when you have a CAPF service enabled and the Cisco UCM is not running on the Publisher but the Publisher is configured with a FQDN instead of an IP address.

Cisco Unified Communications Manager Prerequisites

- The TFTP server must reside on the same interface as the Cisco UCM.
- The Cisco UCM can be on a private network on the inside but you need to have a static mapping for the Cisco UCM on the ASA to a public routable address.
- If NAT is required for Cisco UCM, it must be configured on the ASA, not on the existing firewall.

ACL Rules

If the phone proxy is deployed behind an existing firewall, access-list rules to permit signaling, TFTP requests, and media traffic to the phone proxy must be configured.

If NAT is configured for the TFTP server or Cisco UCMs, the translated "global" address must be used in the ACLs.

Table 16-1 lists the ports that are required to be configured on the existing firewall:

Table 16-1 Port Configuration Requirements

Address	Port	Protocol	Description
Media Termination	1024-65535	UDP	Allow incoming SRTP
TFTP Server	69	UDP	Allow incoming TFTP
Cisco UCM	2443	TCP	Allow incoming secure SCCP
Cisco UCM	5061	TCP	Allow incoming secure SIP
CAPF Service (on Cisco UCM)	3804	TCP	Allow CAPF service for LSC provisioning

Note All these ports are configurable on the Cisco UCM, except for TFTP. These are the default values and should be modified if they are modified on the Cisco UCM. For example, 3804 is the default port for the CAPF Service. This default value should be modified if it is modified on the Cisco UCM.

NAT and PAT Prerequisites

NAT Prerequisites

- If NAT is configured for the TFTP server, the NAT configuration must be configured prior to configuring the **tftp-server** command under the phone proxy.
- If NAT is configured for the TFTP server or Cisco UCMs, the translated "global" address must be used in the ACLs.

PAT Prerequisites

- When the Skinny inspection global port is configured to use a non-default port, then you must configure the nonsecure port as the `global_sccp_port+443`.

  Therefore, if *global_sccp_port* is 7000, then the global secure SCCP port is 7443. Reconfiguring the port might be necessary when the phone proxy deployment has more than one Cisco UCM and they must share the interface IP address or a global IP address.

```
/* use the default ports for the first CUCM */
object network obj-10.0.0.1-01
    host 10.0.0.1
    nat (inside,outside) static interface service tcp 2000 2000
object network obj-10.0.0.1-02
    host 10.0.0.1
    nat (inside,outside) static interface service tcp 2443 2443
/* use non-default ports for the 2nd CUCM */
object network obj-10.0.0.2-01
    host 10.0.0.2
    nat (inside,outside) static interface service tcp 2000 7000
object network obj-10.0.0.2-02
```

```
        host 10.0.0.2
        nat (inside,outside) static interface service tcp 2443 7443
```

Note Both PAT configurations—for the nonsecure and secure ports—must be configured.

- When the IP phones must contact the CAPF on the Cisco UCM and the Cisco UCM is configured with static PAT (LCS provisioning is required), you must configure static PAT for the default CAPF port 3804.

Prerequisites for IP Phones on Multiple Interfaces

When IP phones reside on multiple interfaces, the phone proxy configuration must have the correct IP address set for the Cisco UCM in the CTL file.

See the following example topology for information about how to correctly set the IP address:

```
phones --- (dmz)-----|
                     |----- ASA PP --- (outside Internet) --- phones
phones --- (inside)--|
```

In this example topology, the following IP address are set:

- Cisco UCM on the inside interface is set to 10.0.0.5
- The DMZ network is 192.168.1.0/24
- The inside network is 10.0.0.0/24

The Cisco UCM is mapped with different global IP addresses from DMZ > outside and inside interfaces > outside interface.

In the CTL file, the Cisco UCM must have two entries because of the two different IP addresses. For example, if the static statements for the Cisco UCM are as follows:

```
    object network obj-10.0.0.5-01
        host 10.0.0.5
        nat (inside,outside) static 209.165.202.129
    object network obj-10.0.0.5-02
        host 10.0.0.5
        nat (inside,dmz) static 198.168.1.2
```

There must be two CTL file record entries for the Cisco UCM:

```
    record-entry cucm trustpoint cucm_in_to_out address 209.165.202.129
    record-entry cucm trustpoint cucm_in_to_dmz address 192.168.1.2
```

7960 and 7940 IP Phones Support

- An LSC must be installed on these IP phones because they do not come pre installed with a MIC. Install the LSC on each phone before using them with the phone proxy to avoid opening the nonsecure SCCP port for the IP phones to register in nonsecure mode with the Cisco UCM.

  See the following document for the steps to install an LSC on IP phones:

  http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/security/7_0_1/secugd/secucapf.html#wp1093518

Note If an IP phone already has an LSC installed on it from a different Cisco UCM cluster, delete the LSC from the different cluster and install an LSC from the current Cisco UCM cluster.

Note You can configure LSC provisioning for additional end-user authentication. See the Cisco Unified Communications Manager configuration guide for information.

- The CAPF certificate must be imported onto the ASA.
- The CTL file created on the ASA must be created with a CAPF record-entry.
- The phone must be configured to use only the SCCP protocol because the SIP protocol does not support encryption on these IP phones.
- If LSC provisioning is done via the phone proxy, you must add an ACL to allow the IP phones to register with the Cisco UCM on the nonsecure port 2000.

Cisco IP Communicator Prerequisites

To configure Cisco IP Communicator (CIPC) with the phone proxy, you must meet the following prerequisites:

- Include the **cipc security-mode authenticated** command under the **phone-proxy** command when configuring the phone proxy instance.
- Create an ACL to allow CIPC to register with the Cisco UCM in nonsecure mode.
- Configure null-sha1 as one of the SSL encryption ciphers.

Current versions of Cisco IP Communicator (CIPC) support authenticated mode and perform TLS signaling but not voice encryption. Therefore, you must include the following command when configuring the phone proxy instance:

cipc security-mode authenticated

Because CIPC requires an LSC to perform the TLS handshake, CIPC needs to register with the Cisco UCM in nonsecure mode using cleartext signaling. To allow the CIPC to register, create an ACL that allows the CIPC to connect to the Cisco UCM on the nonsecure SIP/SCCP signalling ports (5060/2000).

Note You can configure LSC provisioning for additional end-user authentication. See the Cisco Unified Communications Manager configuration guide for information.

CIPC uses a different cipher when doing the TLS handshake and requires the null-sha1 cipher and SSL encryption be configured. To add the null-shal cipher, use the show run all ssl command to see the output for the ssl encryption command and add null-shal to the end of the SSL encryption list.

Note When used with CIPC, the phone proxy does not support end-users resetting their device name in CIPC (Preferences > Network tab > Use this Device Name field) or Administrators resetting the device name in Cisco Unified CM Administration console (Device menu > Phone Configuration > Device Name field). To function with the phone proxy, the CIPC configuration file must be in the format: SEP<mac_address>.cnf.xml. If the device name does not follow this format (SEP<mac_address>), CIPC cannot retrieve its configuration file from Cisco UMC via the phone proxy and CIPC will not function.

Prerequisites for Rate Limiting TFTP Requests

In a remote access scenario, we recommend that you configure rate limiting of TFTP requests because any IP phone connecting through the Internet is allowed to send TFTP requests to the TFTP server.

To configure rate limiting of TFTP requests, configure the **police** command in the Modular Policy Framework. See the command reference for information about using the **police** command.

Policing is a way of ensuring that no traffic exceeds the maximum rate (in bits/second) that you configure, thus ensuring that no one traffic flow can take over the entire resource. When traffic exceeds the maximum rate, the ASA drops the excess traffic. Policing also sets the largest single burst of traffic allowed.

Rate Limiting Configuration Example

The following example describes how you configure rate limiting for TFTP requests by using the **police** command and the Modular Policy Framework.

Begin by determining the conformance rate that is required for the phone proxy. To determine the conformance rate, use the following formula:

```
X * Y * 8
```

Where

X = requests per second

Y = size of each packet, which includes the L2, L3, and L4 plus the payload

Therefore, if a rate of 300 TFTP requests/second is required, then the conformance rate would be calculated as follows:

```
300 requests/second * 80 bytes * 8 = 192000
```

The example configuration below shows how the calculated conformance rate is used with the **police** command:

```
access-list tftp extended permit udp any host 192.168.0.1 eq tftp

class-map tftpclass
   match access-list tftp

policy-map tftpmap
   class tftpclass
   police output 192000

service-policy tftpmap interface inside
```

About ICMP Traffic Destined for the Media Termination Address

To control which hosts can ping the media termination address, use the **icmp** command and apply the access rule to the outside interface on the ASA.

Any rules for ICMP access applied to the outside interface apply to traffic destined for the media termination address.

For example, use the following command to deny ICMP pings from any host destined for the media termination address:

icmp deny any outside

End-User Phone Provisioning

The phone proxy is a transparent proxy with respect to the TFTP and signaling transactions. If NAT is not configured for the Cisco UCM TFTP server, then the IP phones need to be configured with the Cisco UCM cluster TFTP server address.

If NAT is configured for the Cisco UCM TFTP server, then the Cisco UCM TFTP server global address is configured as the TFTP server address on the IP phones.

Ways to Deploy IP Phones to End Users

In both options, deploying a remote IP phone behind a commercial Cable/DSL router with NAT capabilities is supported.

Option 1 (Recommended)

Stage the IP phones at corporate headquarters before sending them to the end users:

- The phones register inside the network. IT ensures there are no issues with the phone configurations, image downloads, and registration.
- If Cisco UCM cluster was in mixed mode, the CTL file should be erased before sending the phone to the end user.

Advantages of this option are:

- Easier to troubleshoot and isolate problems with the network or phone proxy because you know whether the phone is registered and working with the Cisco UCM.
- Better user experience because the phone does not have to download firmware from over a broadband connection, which can be slow and require the user to wait for a longer time.

Option 2

Send the IP phone to the end user. When using option 2, the user must be provided instructions to change the settings on phones with the appropriate Cisco UCM and TFTP server IP address.

Note As an alternative to authenticating remote IP phones through the TLS handshake, you can configure authentication via LSC provisioning. With LSC provisioning you create a password for each remote IP phone user and each user enters the password on the remote IP phones to retrieve the LSC.

Because using LSC provisioning to authenticate remote IP phones requires the IP phones first register in nonsecure mode, Cisco recommends LSC provisioning be done inside the corporate network before giving the IP phones to end-users. Otherwise, having the IP phones register in nonsecure mode requires the Administrator to open the nonsecure signaling port for SIP and SCCP on the ASA.

See “Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher, page 16-50“. See also the Cisco Unified Communications Manager Security Guide for information on Using the Certificate Authority Proxy Function (CAPF) to install a locally significant certificate (LSC).

Phone Proxy Guidelines and Limitations

This section includes the following topics:

- General Guidelines and Limitations, page 16-13
- Media Termination Address Guidelines and Limitations, page 16-14

General Guidelines and Limitations

The phone proxy has the following general limitations:

- Only one phone proxy instance can be configured on the ASA by using the **phone-proxy** command. See the command reference for information about the **phone-proxy** command. See also Creating the Phone Proxy Instance, page 16-24.
- The phone proxy only supports one Cisco UCM cluster. See Creating the CTL File, page 16-18 for the steps to configure the Cisco UCM cluster for the phone proxy.
- The phone proxy is not supported when the ASA is running in transparent mode or multiple context mode.
- When a remote IP phone calls an invalid internal or external extension, the phone proxy does not support playing the annunciator message from the Cisco UCM. Instead, the remote IP phone plays a fast busy signal instead of the annunciator message "Your call cannot be completed ..." However, when an internal IP phone dials in invalid extension, the annunciator messages plays "Your call cannot be completed ..."
- Packets from phones connecting to the phone proxy over a VPN tunnel are not inspected by the ASA inspection engines.
- The phone proxy does not support IP phones sending Real-Time Control Protocol (RTCP) packets through the ASA. Disable RTCP packets in the Cisco Unified CM Administration console from the Phone Configuration page. See your Cisco Unified Communications Manager (CallManager) documentation for information about setting this configuration option.
- When used with CIPC, the phone proxy does not support end-users resetting their device name in CIPC (Preferences > Network tab > Use this Device Name field) or Administrators resetting the device name in Cisco Unified CM Administration console (Device menu > Phone Configuration > Device Name field). To function with the phone proxy, the CIPC configuration file must be in the format: SEP<mac_address>.cnf.xml. If the device name does not follow this format (SEP<mac_address>), CIPC cannot retrieve its configuration file from Cisco UMC via the phone proxy and CIPC will not function.
- The phone proxy does not support IP phones sending SCCP video messages using Cisco VT Advantage because SCCP video messages do not support SRTP keys.
- For mixed-mode clusters, the phone proxy does not support the Cisco Unified Call Manager using TFTP to send encrypted configuration files to IP phones through the ASA.
- Multiple IP phones behind one NAT device must be configured to use the same security mode.

  When the phone proxy is configured for a mixed-mode cluster and multiple IP phones are behind one NAT device and registering through the phone proxy, all the SIP and SCCP IP phones must be configured as authenticated or encrypted, or all as non-secure on the Unified Call Manager.

  For example, if there are four IP phones behind one NAT device where two IP phones are configured using SIP and two IP phones are configured using SCCP, the following configurations on the Unified Call Manager are acceptable:

  - Two SIP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, or both in encrypted mode

    Two SCCP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, or both in encrypted mode

- Two SIP IP phones: both in non-secure mode

  Two SCCP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, both in encrypted mode
- Two SIP IP phones: one IP phone in authenticated mode and one in encrypted mode, both in authenticated mode, both in encrypted mode

  Two SCCP IP phones: both in non-secure mode

This limitation results from the way the application-redirect rules (rules that convert TLS to TCP) are created for the IP phones.

Media Termination Address Guidelines and Limitations

The phone proxy has the following limitations relating to configuring the media-termination address:

- When configuring the media-termination address, the phone proxy does not support having internal IP phones (IP phones on the inside network) being on a different network interface from the Cisco UCM unless the IP phones are forced to use the non-secure Security mode.

  When internal IP phones are on a different network interface than the Cisco UCM, the IP phones signalling sessions still go through ASA; however, the IP phone traffic does not go through the phone proxy. Therefore, Cisco recommends that you deploy internal IP phones on the same network interface as the Cisco UMC.

  If the Cisco UMC and the internal IP phones must be on different network interfaces, you must add routes for the internal IP phones to access the network interface of the media-termination address where Cisco UMC resides.

  When the phone proxy is configured to use a global media-termination address, all IP phones see the same global address, which is a public routable address.
- If you decide to configure a media-termination address on interfaces (rather than using a global interface), you must configure a media-termination address on at least two interfaces (the inside and an outside interface) before applying the phone-proxy service policy. Otherwise, you will receive an error message when enabling the Phone Proxy with SIP and Skinny Inspection.
- The phone proxy can use only one type of media termination instance at a time; for example, you can configure a global media-termination address for all interfaces or configure a media-termination address for different interfaces. However, you cannot use a global media-termination address and media-termination addresses configured for each interface at the same time.

Configuring the Phone Proxy

This section includes the following topics:

- Creating the TLS Proxy for a Mixed-mode Cisco UCM Cluster, page 16-21
- Creating the Media Termination Instance, page 16-23
- Creating the Phone Proxy Instance, page 16-24
- Enabling the Phone Proxy with SIP and Skinny Inspection, page 16-26
- Configuring Linksys Routers with UDP Port Forwarding for the Phone Proxy, page 16-27

Task Flow for Configuring the Phone Proxy in a Non-secure Cisco UCM Cluster

Follow these tasks to configure the phone proxy in a Non-secure Cisco UCM Cluster:

Step 1 Create trustpoints and generate certificates for each entity in the network (Cisco UCM, Cisco UCM and TFTP, TFTP server, CAPF) that the IP phone must trust. The certificates are used in creating the CTL file. See Creating Trustpoints and Generating Certificates, page 16-17.

Note Before you create the trustpoints and generate certificates, you must have imported the required certificates, which are stored on the Cisco UCM. See Certificates from the Cisco UCM, page 16-7 and Importing Certificates from the Cisco UCM, page 16-15

Step 2 Create the CTL file for the phone proxy. See Creating the CTL File, page 16-18.

Step 3 Create the TLS proxy instance. See Creating the TLS Proxy Instance for a Non-secure Cisco UCM Cluster, page 16-20.

Step 4 Create the media termination instance for the phone proxy. See Creating the Media Termination Instance, page 16-23.

Step 5 Create the phone proxy instance. See Creating the Phone Proxy Instance, page 16-24.

Step 6 Enable the phone proxy y with SIP and Skinny inspection. See Enabling the Phone Proxy with SIP and Skinny Inspection, page 16-26.

Importing Certificates from the Cisco UCM

For the TLS proxy used by the phone proxy to complete the TLS handshake successfully, it needs to verify the certificates from the IP phone (and the Cisco UCM if doing TLS with Cisco UCM). To validate the IP phone certificate, we need the CA Manufacturer certificate which is stored on the Cisco UCM. Follow these steps to import the CA Manufacturer certificate to the ASA.

Step 1 Go to the Cisco UCM Operating System Administration web page.

Step 2 Choose **Security** > **Certificate Management**.

Note Earlier versions of Cisco UCM have a different UI and way to locate the certificates. For example, in Cisco UCM version 4.x, certificates are located in the directory `C:\Program Files\Cisco\Certificates`. See your Cisco Unified Communications Manager (CallManager) documentation for information about locating certificates.

Step 3 Click Find and it will display all the certificates.

Step 4 Find the filename `Cisco_Manufacturing_CA`. This is the certificate need to verify the IP phone certificate. Click the .PEM file `Cisco_Manufacturing_CA.pem`. This will show you the certificate information and a dialog box that has the option to download the certificate.

Note If the certificate list contains more than one certificate with the filename `Cisco_Manufacturing_CA`, make you select the certificate `Cisco_Manufacturing_CA.pem`—the one with the .pem file extension.

Step 5 Click Download and save the file as a text file.

Step 6 On the ASA, create a trustpoint for the Cisco Manufacturing CA and enroll via terminal by entering the following commands. Enroll via terminal because you will paste the certificate you downloaded in Step 4.

```
hostname(config)# crypto ca trustpoint trustpoint_name
hostname(config-ca-trustpoint)# enrollment terminal
```

Step 7 Authenticate the trustpoint by entering the following command:

```
hostname(config)# crypto ca authenticate trustpoint
```

Step 8 You are prompted to "Enter the base 64 encoded CA Certificate." Copy the .PEM file you downloaded in Step 4 and paste it at the command line. The file is already in base-64 encoding so no conversion is required. If the certificate is OK, you are prompted to accept it: "Do you accept this certificate? [yes/no]." Enter **yes**.

Note When you copy the certificate, make sure that you also copy also the lines with BEGIN and END.

Tip If the certificate is not ok, use the **debug crypto ca** command to show debug messages for PKI activity (used with CAs).

Step 9 Repeat the Step 1 through Step 8 for the next certificate. Table 16-2 shows the certificates that are required by the ASA.

Table 16-2 Certificates Required by the Security Appliance for the Phone Proxy

Certificate Name	Required for...
CallManager	Authenticating the Cisco UCM during TLS handshake; only required for mixed-mode clusters.
Cisco_Manufacturing_CA	Authenticating IP phones with a Manufacturer Installed Certificate (MIC).
CAP-RTP-001	Authenticating IP phones with a MIC.
CAP-RTP-002	Authenticating IP phones with a MIC.
CAPF	Authenticating IP phones with an LSC.

Task Flow for Configuring the Phone Proxy in a Mixed-mode Cisco UCM Cluster

Note For mixed-mode clusters, the phone proxy does not support the Cisco Unified Call Manager using TFTP to send encrypted configuration files to IP phones through the ASA.

Follow these tasks to configure the phone proxy in a Non-secure Cisco UCM Cluster:

Step 1 Create trustpoints and generate certificates for each entity in the network (Cisco UCM, Cisco UCM and TFTP, TFTP server, CAPF) that the IP phone must trust. The certificates are used in creating the CTL file. See Creating Trustpoints and Generating Certificates, page 16-17.

Note Before you create the trustpoints and generate certificates, you must have imported the required certificates, which are stored on the Cisco UCM. See Certificates from the Cisco UCM, page 16-7 and Importing Certificates from the Cisco UCM, page 16-15

Step 2 Create the CTL file for the phone proxy. See Creating the CTL File, page 16-18.

Note When the phone proxy is being configured to run in mixed-mode clusters, you have the following option to use an existing CTL file to install the trustpoints. See Using an Existing CTL File, page 16-20.

Step 3 Create the TLS proxy instance. See Creating the TLS Proxy for a Mixed-mode Cisco UCM Cluster, page 16-21.

Step 4 Create the media termination instance for the phone proxy. See Creating the Media Termination Instance, page 16-23.

Step 5 Create the phone proxy instance. See Creating the Phone Proxy Instance, page 16-24.

Step 6 While configuring the phone proxy instance (in the Phone Proxy Configuration mode), enter the following command to configure the mode of the cluster to be mixed mode because the default is nonsecure:

```
hostname(config-phone-proxy)# cluster-mode mixed
```

Step 7 Enable the phone proxy y with SIP and Skinny inspection. See Enabling the Phone Proxy with SIP and Skinny Inspection, page 16-26.

Creating Trustpoints and Generating Certificates

Create trustpoints and generate certificates for each entity in the network (Cisco UCM, Cisco UCM and TFTP, TFTP server, CAPF) that the IP phone must trust. The certificates are used in creating the CTL file.

You need to create trustpoints for each Cisco UCM (primary and secondary if a secondary Cisco UCM is used) and TFTP server in the network. The trustpoints need to be in the CTL file for the phones to trust the Cisco UCM.

Prerequisites

Import the required certificates, which are stored on the Cisco UCM. See Certificates from the Cisco UCM, page 16-7 and Importing Certificates from the Cisco UCM, page 16-15.

	Command	Purpose
Step 1	hostname(config)# **crypto key generate rsa label** *key-pair-label* **modulus** *size* **Example:** crypto key generate rsa label cucmtftp_kp modulus 1024	Creates a keypair that can be used for the trustpoints.
Step 2	hostname(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** crypto ca trustpoint cucm_tftp_server	Creates the trustpoints for each entity in the network (primary Cisco UCM, secondary Cisco UCM, and TFTP server). **Note** You are only required to create a separate trustpoint for the TFTP server when the TFTP server resides on a different server from the Cisco UCM. See Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers, page 16-47 for an example of this configuration.
Step 3	hostname(config-ca-trustpoint)# **enrollment self**	Generates a self-signed certificate.
Step 4	hostname(config-ca-trustpoint)# **keypair** *keyname* **Example:** keypair cucmtftp_kp	Specifies the keypair whose public key is being certified.
Step 5	hostname(config-ca-trustpoint)# **exit**	Exits from the Configure Trustpoint mode.
Step 6	hostname(config)# **crypto ca enroll** *trustpoint* **Example:** crypto ca enroll cucm_tftp_server	Requests the certificate from the CA server and causes the ASA to generate the certificate. When prompted to include the device serial number in the subject name, type **Y** to include the serial number or type **N** to exclude it. When prompted to generate the self-signed certificate, type **Y**.

What to Do Next

Once you have created the trustpoints and generated the certificates, create the CTL file for the phone proxy. See Creating the CTL File, page 16-18.

If you are configuring the phone proxy in a mixed-mode cluster, you can use an existing CTL file. See Using an Existing CTL File, page 16-20.

Creating the CTL File

Create the CTL file that will be presented to the IP phones during the TFTP requests.

Prerequisites

If you are using domain names for your Cisco UCM and TFTP server, you must configure DNS lookup on the ASA. Add an entry for each of the outside interfaces on the ASA into your DNS server, if such entries are not already present. Each ASA outside IP address should have a DNS entry associated with it for lookups. These DNS entries must also be enabled for Reverse Lookup.

Enable DNS lookups on your ASA with the **dns domain-lookup** *interface_name* command (where the *interface_name* specifies the interface that has a route to your DNS server). Additionally, define your DNS server IP address on the ASA; for example: `dns name-server 10.2.3.4` (IP address of your DNS server).

Note You can enter the **dns domain-lookup** command multiple times to enable DNS lookup on multiple interfaces. If you enter multiple commands, the ASA tries each interface in the order it appears in the configuration until it receives a response.

See the command reference for information about the **dns domain-lookup** command.

	Command	Purpose

Step 1	`hostname(config)#` **`ctl-file`** *`ctl_name`*
	`Example:`
	`ctl-file myctl`
		Creates the CTL file instance.

Step 2	`hostname(config-ctl-file)#` **`record-entry tftp trustpoint`** *`trustpoint_name`* **`address`** *`TFTP_IP_address`*
	`Example:`
	`record-entry cucm-tftp trustpoint cucm_tftp_server address 10.10.0.26`
		Creates the record entry for the TFTP server.
		Note Use the global or mapped IP address of the TFTP server or Cisco UCM if NAT is configured.

Step 3	`hostname(config-ctl-file)#` **`record-entry cucm trustpoint`** *`trustpoint_name`* **`address`** *`IP_address`*
	`Example:`
	`record-entry cucm trustpoint cucm_server address 10.10.0.26`
		Creates the record entry for the each Cisco UCM (primary and secondary).
		Note Use the global or mapped IP address of the Cisco UCM.

Step 4	`hostname(config-ctl-file)#` **`record-entry capf trustpoint`** *`trust_point address`*
	`Example:`
	`record-entry capf trustpoint capf address 10.10.0.26`
		Creates the record entry for CAPF.
		Note You only enter this command when LSC provisioning is required or you have LSC enabled IP phones.

Step 5	`hostname(config-ctl-file)#` **`no shutdown`**	Creates the CTL file. When the file is created, it creates an internal trustpoint used by the phone proxy to sign the TFTP files. The trustpoint is named **_internal_PP_***ctl-instance_filename*.
Step 6	`hostname(config)#` **`copy running-configuration startup-configuration`**	Saves the certificate configuration to Flash memory.

What to Do Next

Once you have configured the CTL file for the phone proxy, create the TLS proxy instance. See Creating the TLS Proxy Instance for a Non-secure Cisco UCM Cluster, page 16-20 to add the TLS proxy when configuring the phone proxy in a non-secure mode or see Creating the TLS Proxy for a Mixed-mode Cisco UCM Cluster, page 16-21 if the phone proxy is running in a mixed-mode cluster.

Using an Existing CTL File

Note Only when the phone proxy is running in mixed-mode clusters, you have the option to use an existing CTL file to install trustpoints.

If you have an existing CTL file that contains the correct IP addresses of the entities (namely, the IP address that the IP phones use for the Cisco UCM or TFTP servers), you can be use it to create a new CTL file thereby using the existing CTL file to install the trustpoints for each entity in the network (Cisco UCM, Cisco UCM and TFTP, TFTP server, CAPF) that the IP phones must trust.

Prerequisites

If a CTL file exists for the cluster, copy the CTL file to Flash memory. When you copy the CTL file to Flash memory, rename the file and do not name the file `CTLFile.tlv`.

If you are using domain names for your Cisco UCM and TFTP server, you must configure DNS lookup on the ASA. See the prerequisites for Creating the CTL File, page 16-18.

	Command	Purpose

Step 1	`hostname(config)#` **`ctl-file`** *`ctl_name`*
	`Example:`
	`ctl-file myctl`
		Creates the CTL file instance.

Step 2	`hostname(config-ctl-file)#` **`cluster-ctl-file`** *`filename_path`*
	`Example:`
	`hostname(config-ctl-file)# cluster-ctl-file disk0:/old_ctlfile.tlv`
		Uses the trustpoints that are already in the existing CTL file stored in Flash memory.
		Where the existing CTL file was saved to Flash memory with a filename other than `CTLFile.tlv`; for example, `old_ctlfile.tlv`.

What to Do Next

When using an existing CTL file to configure the phone proxy, you can add additional entries to the file as necessary. See Creating the CTL File, page 16-18.

Once you have configured the CTL file for the phone proxy, create the TLS proxy instance. See Creating the TLS Proxy Instance for a Non-secure Cisco UCM Cluster, page 16-20 to add the TLS proxy when configuring the phone proxy in a non-secure mode or see Creating the TLS Proxy for a Mixed-mode Cisco UCM Cluster, page 16-21 if the phone proxy is running in a mixed-mode cluster.

Creating the TLS Proxy Instance for a Non-secure Cisco UCM Cluster

Create the TLS proxy instance to handle the encrypted signaling.

	Command	Purpose
Step 1	hostname(config)# **tls-proxy** *proxy_name* **Example:** tls-proxy mytls	Creates the TLS proxy instance.
Step 2	hostname(config-tlsp)# **server trust-point _internal_PP_***ctl-instance_filename* **Example:** server trust-point _internal_PP_myctl	Configures the server trustpoint and references the internal trustpoint named **_internal_PP_***ctl-instance_filename*.

What to Do Next

Once you have created the TLS proxy instance, create the phone proxy instance. See Creating the Phone Proxy Instance, page 16-24.

Creating the TLS Proxy for a Mixed-mode Cisco UCM Cluster

For mixed mode clusters, there might be IP phones that are already configured as encrypted so it requires TLS to the Cisco UCM. You must configure the LDC issuer for the TLS proxy.

	Command	Purpose
Step 1	hostname(config)# **crypto key generate rsa label** *key-pair-label* **modulus** *size* **Examples:** hostname(config)# crypto key generate rsa label ldc_signer_key modulus 1024 hostname(config)# crypto key generate rsa label phone_common modulus 1024	Creates the necessary RSA key pairs. Where the *key-pair-label* is the LDC signer key and the key for the IP phones.
Step 2	hostname(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** hostname(config)# crypto ca trustpoint ldc_server	Creates an internal local CA to sign the LDC for Cisco IP phones. Where the *trustpoint_name* is for the LDC.
Step 3	hostname(config-ca-trustpoint)# **enrollment self**	Generates a self-signed certificate.
Step 4	hostname(config-ca-trustpoint)# **proxy-ldc-issuer**	Defines the local CA role for the trustpoint to issue dynamic certificates for the TLS proxy.
Step 5	hostname(config-ca-trustpoint)# **fqdn** *fqdn* **Example:** hostname(config-ca-trustpoint)# fqdn my-ldc-ca.example.com	Includes the indicated FQDN in the Subject Alternative Name extension of the certificate during enrollment. Where the *fqdn* is for the LDC.

	Command	Purpose
Step 6	hostname(config-ca-trustpoint)# **subject-name** *X.500_name* **Example:** hostname(config-ca-trustpoint)# subject-name cn=FW_LDC_SIGNER_172_23_45_200	Includes the indicated subject DN in the certificate during enrollment Where the *X.500_name* is for the LDC. Use commas to separate attribute-value pairs. Insert quotation marks around any value that contains commas or spaces. For example: `cn=crl,ou=certs,o="cisco systems, inc.",c=US` The maximum length is 500 characters.
Step 7	hostname(config-ca-trustpoint)# **keypair** *keypair* **Example:** hostname(config-ca-trustpoint)# keypair ldc_signer_key	Specifies the key pair whose public key is to be certified. Where the *keypair* is for the LDC.
Step 8	hostname(config)# **crypto ca enroll** *ldc_server* **Example:** hostname(config)# crypto ca enroll ldc_server	Starts the enrollment process with the CA.
Step 9	hostname(config)# **tls-proxy** *proxy_name* **Example:** tls-proxy mytls	Creates the TLS proxy instance.
Step 10	hostname(config-tlsp)# **server trust-point _internal_PP_***ctl-instance_filename* **Example:** hostname(config-tlsp)# server trust-point _internal_PP_myctl	Configures the server trustpoint and references the internal trustpoint named **_internal_PP_***ctl-instance_filename*.
Step 11	hostname(config-tlsp)# **client ldc issuer** *ca_tp_name* **Example:** client ldc issuer ldc_server	Specifies the local CA trustpoint to issue client dynamic certificates.
Step 12	hostname(config-tlsp)# **client ldc keypair** *key_label* **Example:** hostname(config-tlsp)# client ldc keypair phone_common	Specifies the RSA keypair to be used by client dynamic certificates.
Step 13	hostname(config-tlsp)# **client cipher-suite** *cipher-suite* **Example:** hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1	Specifies the cipher suite. Options include des-sha1, 3des-sha1, aes128-sha1, aes256-sha1, or null-sha1.
Step 14		Exports the local CA certificate and installs it as a trusted certificate on the Cisco Unified Communications Manager server by performing one of the following actions.

	Command	Purpose

•	`hostname(config)# crypto ca export trustpoint identity-certificate`
	Example:
	`hostname(config)# crypto ca export ldc_server identity-certificate`
		Exports the certificate if a trustpoint with proxy-ldc-issuer is used as the signer of the dynamic certificates.

•	`hostname(config)# show crypto ca server certificates`	Exports the certificate for the embedded local CA server LOCAL-CA-SERVER. After exporting the certificate, you must save the output to a file and import it on the Cisco Unified Communications Manager. You can use the Display Certificates function in the Cisco Unified Communications Manager software to verify the installed certificate. For information about performing these procedures, see the following URLs: http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/cucos/5_0_4/iptpch6.html#wp1040848 http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/cucos/5_0_4/iptpch6.html#wp1040354

What To Do Next

Once you have created the TLS proxy instance and installed the certificate on the Cisco Unified Communications Manager, create the phone proxy instance. See Creating the Phone Proxy Instance, page 16-24.

Creating the Media Termination Instance

Create the media termination instance that you will use in the phone proxy.

The media termination address you configure must meet the requirements as described in Media Termination Instance Prerequisites, page 16-6.

	Command	Purpose

Step 1	`hostname(config)#` **`media-termination`** *`instance_name`*
	`Example:`
	`hostname(config)#` **`media-termination`** *`mediaterm1`*
		Creates the media termination instance that you attach to the phone proxy.

Step 2	`hostname(config-media-termination)#` **`address`** *`ip_address`* `[`**`interface`** *`intf_name`*`]`
	`Examples:`
	`hostname(config-media-termination)# address 192.0.2.25 interface inside`
	`hostname(config-media-termination)# address 10.10.0.25 interface outside`
		Configures the media-termination address used by the media termination instance. The phone proxy uses this address for SRTP and RTP.
		For the media termination instance, you can configure a global media-termination address for all interfaces or configure a media-termination address for different interfaces. However, you cannot use a global media-termination address and media-termination addresses configured for each interface at the same time.
		If you configure a media termination address for multiple interfaces, you must configure an address on each interface that the ASA uses when communicating with IP phones.
		The IP addresses are publicly routable addresses that are unused IP addresses within the address range on that interface.
		See Media Termination Instance Prerequisites, page 16-6 for the complete list of prerequisites that you must follow when creating the media termination instance and configuring the media termination addresses.

Step 3	(Optional)
	`hostname(config-media-termination)#` **`rtp-min-port`** *`port1`* **`rtp-max-port`** *`port2`*
	`Example:`
	`hostname(config-media-termination)# rtp-min-port 2001 rtp-maxport 32770`
		Specifies the minimum and maximum values for the RTP port range for the media termination instance.
		Where *port1* and *port2* can be a value from 1024 to 65535.

What To Do Next

Once you have created the media termination instance, create the phone proxy instance. See Creating the Phone Proxy Instance, page 16-24.

Creating the Phone Proxy Instance

Create the phone proxy instance.

Prerequisites

You must have already created the CTL file and TLS proxy instance for the phone proxy.

See Creating the CTL File, page 16-18 and Creating the TLS Proxy Instance for a Non-secure Cisco UCM Cluster, page 16-20

	Command	Purpose
Step 1	hostname(config)# **phone-proxy** *phone_proxy_name* **Example:** hostname(config)# phone-proxy myphoneproxy	Creates the phone proxy instance. Only one phone proxy instance can be configured on the security appliance.
Step 2	hostname(config-phone-proxy)# **media-termination** *instance_name* **Examples:** hostname(config-phone-proxy)# media-termination my_mt	Specifies the media termination instance used by the phone proxy for SRTP and RTP. **Note** You must create the media termination instance before you specify it in the phone proxy instance. See Creating the Media Termination Instance, page 16-23 for the steps to create the media termination instance.
Step 3	hostname(config-phone-proxy)# **tftp-server address** *ip_address* **interface** *interface* **Example:** hostname(config-phone-proxy)# tftp-server address 192.0.2.101 interface inside	Creates the TFTP server using the actual internal address and specify the interface on which the TFTP server resides.
Step 4	hostame(config-phone-proxy)# **tls-proxy** *proxy_name* **Example:** hostame(config-phone-proxy)# tls-proxy mytls	Configures the TLS proxy instance that you have already created.
Step 5	hostname(config-phone-proxy)# **ctl-file** *ctl_name* **Example:** hostame(config-phone-proxy)# ctl-file myctl	Configures the CTL file instance that you have already created,
Step 6	hostname(config-phone-proxy)# **proxy-server address** *ip_address* [*listen_port*] **interface** *ifc* **Example:** hostname(config-phone-proxy)# proxy-server 192.168.1.2 interface inside	(Optional) If the operational environment has an external HTTP proxy to which the IP phones direct all HTTP request, configures a proxy server. You can configure only one proxy server while the phone proxy is in use. By default, the Phone URL Parameters configured under the Enterprise Parameters use an FQDN in the URLs. The parameters might need to be changed to use an IP address if the DNS lookup for the HTTP proxy does not resolve the FQDNs. **Note** If the IP phones have already downloaded their configuration files after you have configured the proxy server, you must restart the IP phones so that they get the configuration file with the proxy server address in the file.

	Command	Purpose
Step 7	`hostname(config-phone-proxy)# `**`cipc security-mode authenticated`**	(Optional) Forces Cisco IP Communicator (CIPC) softphones to operate in authenticated mode when CIPC softphones are deployed in a voice and data VLAN scenario. See Cisco IP Communicator Prerequisites, page 16-10 for all requirements for using the phone proxy with CIPC.
Step 8	`hostname(config-phone-proxy)# `**`no disable service-settings`**	(Optional) Preserve the settings configured on the Cisco UCM for each IP phone configured. By default, the following settings are disabled on the IP phones: • PC Port • Gratuitous ARP • Voice VLAN access • Web Access • Span to PC Port

What to Do Next

Once you have created the phone proxy instance, configuring SIP and Skinny for the phone proxy. See Enabling the Phone Proxy with SIP and Skinny Inspection, page 16-26.

Enabling the Phone Proxy with SIP and Skinny Inspection

Enables the phone proxy instance that you created to inspect SIP and Skinny protocol traffic.

Prerequisites

You must have already created the phone proxy instance. See Creating the Phone Proxy Instance, page 16-24.

	Command	Purpose

Step 1	`hostname(config)# `**`class-map`** *`class_map_name`*
	Example:
	`class-map sec_sccp`
		Configures the secure Skinny class of traffic to inspect. Traffic between the Cisco Unified Communications Manager and Cisco IP Phones uses SCCP and is handled by SCCP inspection.
		Where *class_map_name* is the name of the Skinny class map.

Step 2	`hostname(config-cmap)# `**`match port tcp eq 2443`**	Matches the TCP port 2443 to which you want to apply actions for secure Skinny inspection.
Step 3	`hostname(config-cmap)# `**`exit`**	Exits from the Class Map configuration mode.

	Command	Purpose

Step 4	`hostname(config)# class-map class_map_name`
	Example:
	`class-map sec_sip`
		Configures the secure SIP class of traffic to inspect.
		Where *class_map_name* is the name of the SIP class map.

Step 5	`hostname(config-cmap)# match port tcp eq 5061`	Matches the TCP port 5061 to which you want to apply actions for secure SIP inspection
Step 6	`hostname(config-cmap)# exit`	Exits from the Class Map configuration mode.

Step 7	`hostname(config)# policy-map name`
	Example:
	`policy-map pp_policy`
		Configure the policy map and attach the action to the class of traffic.

Step 8	`hostname(config-pmap)# class classmap-name`
	Example:
	`class sec_sccp`
		Assigns a class map to the policy map so that you can assign actions to the class map traffic.
		Where *classmap_name* is the name of the Skinny class map.

Step 9	`hostname(config-pmap-c)# inspect skinny phone-proxy pp_name`
	Example:
	`inspect skinny phone-proxy mypp`
		Enables SCCP (Skinny) application inspection and enables the phone proxy for the specified inspection session.

Step 10	`hostnae(config-pmap)# class classmap-name`
	Example:
	`class sec_sip`
		Assigns a class map to the policy map so that you can assign actions to the class map traffic.
		Where *classmap_name* is the name of the SIP class map.

Step 11	`hostname(config-pmap-c)# inspect sip phone-proxy pp_name`
	Example:
	`inspect sip phone-proxy mypp`
		Enables SIP application inspection and enables the phone proxy for the specified inspection session.

Step 12	`hostname(config-pmap-c)# exit`	Exits from Policy Map configuration mode.

Step 13	`hostname(config)# service-policy policymap_name interface intf`
	Example:
	`service-policy pp_policy interface outside`
		Enables the service policy on the outside interface.

Configuring Linksys Routers with UDP Port Forwarding for the Phone Proxy

When IP phones are behind a NAT-capable router, the router can be configured to forward the UDP ports to the IP address of the IP phone. Specifically, configure the router for UDP port forwarding when an IP phone is failing during TFTP requests and the failure is due to the router dropping incoming TFTP data packets. Configure the router to enable UDP port forwarding on port 69 to the IP phone.

As an alternative of explicit UDP forwarding, some Cable/DSL routers require you to designate the IP phone as a DMZ host. For Cable/DSL routers, this host is a special host that receives all incoming connections from the public network.

When configuring the phone proxy, there is no functional difference between an IP phone that has UDP ports explicitly forwarded or an IP phone designated as a DMZ host. The choice is entirely dependent upon the capabilities and preference of the end user.

Configuring Your Router

Your firewall/router needs to be configured to forward a range of UDP ports to the IP phone. This will allow the IP phone to receive audio when you make/receive calls.

Note Different Cable/DSL routers have different procedures for this configuration. Furthermore most NAT-capable routers will only allow a given port range to be forwarded to a single IP address

The configuration of each brand/model of firewall/router is different, but the task is the same. For specific instructions for your brand and model of router, please contact the manufacturer's website.

Linksys Routers

Step 1 From your web browser, connect to the router administrative web page. For Linksys, this is typically something like `http://192.168.1.1`.

Step 2 Click Applications & Gaming or the Port Forwarding tab (whichever is present on your router).

Step 3 Locate the table containing the port forwarding data and add an entry containing the following values:

Table 16-3 Port Forwarding Values to Add to Router

Application	Start	End	Protocol	IP Address	Enabled
IP phone	1024	65535	UDP	*Phone IP address*	**Checked**
TFTP	69	69	UDP	*Phone IP address*	**Checked**

Step 4 Click Save Settings. Port forwarding is configured.

Troubleshooting the Phone Proxy

This section includes the following topics:

- Debugging Information from the Security Appliance, page 16-28
- Debugging Information from IP Phones, page 16-32
- IP Phone Registration Failure, page 16-33
- Media Termination Address Errors, page 16-41
- Audio Problems with IP Phones, page 16-42
- Saving SAST Keys, page 16-42

Debugging Information from the Security Appliance

This section describes how to use the **debug**, **capture**, and **show** commands to obtain debugging information for the phone proxy. See the command reference for detailed information about the syntax for these commands.

Table 16-4 lists the **debug** commands to use with the phone proxy.

Table 16-4 Security Appliance Debug Commands to Use with the Phone Proxy

To	Use the Command	Notes
To show error and event messages for TLS proxy inspection.	**debug inspect tls-proxy** [**events** | **errors**]	Use this command when your IP phone has successfully downloaded all TFTP files but is failing to complete the TLS handshake with the TLS proxy configured for the phone proxy.
To show error and event messages of media sessions for SIP and Skinny inspections related to the phone proxy.	**debug phone-proxy media** [**events** | **errors**]	Use this command in conjunction with the **debug sip** command and the **debug skinny** command if your IP phone is experiencing call failures or audio problems.
To show error and event messages of signaling sessions for SIP and Skinny inspections related to the phone proxy.	**debug phone-proxy signaling** [**events** | **errors**]	Use this command in conjunction with the **debug sip** command and the **debug skinny** command if your IP phone is failing to register with the Cisco UCM or if you are experiencing call failure.
To show error and event messages of TFTP inspection, including creation of the CTL file and configuration file parsing.	**debug phone-proxy tftp** [**events** | **errors**]	
To show debug messages for SIP application inspection.	**debug sip**	Use this command when your IP phones are experiencing connection problems; for example, you can connect within the network but cannot make calls off the network. In the output, check for 4XX or 5XX messages.
To show debug messages for SCCP (Skinny) application inspection.	**debug skinny**	Use this command when your IP phones are experiencing connection problems; for example, you can connect within the network but cannot make calls off the network. In the output, check for 4XX or 5XX messages.

Table 16-5 lists the capture commands to use with the phone proxy. Use the **capture** command on the appropriate interfaces (IP phones and Cisco UCM) to enable packet capture capabilities for packet sniffing and network fault isolation.

Table 16-5 Security Appliance Capture Commands to Use with the Phone Proxy

To	Use the Command	Notes
To capture packets on the ASA interfaces.	**capture** *capture_name* **interface** *interface_name*	Use this command if you are experiencing any problems that might require looking into the packets. For example, if there is a TFTP failure and the output from the **debug** command does not indicate the problem clearly, run the **capture** command on the interface on which the IP phone resides and the interface on which the TFTP server resides to see the transaction and where the problem could be.
To capture data from the TLS proxy when there is a non-secure IP phone connecting to the phone proxy on the inside interface.	**capture** *capture_name* **packet-length** *bytes* **interface inside buffer** *buf_size*	
To capture encrypted data from the TLS proxy when there are secure IP phones connecting to the phone proxy on the inside interface.	**capture** *capture_name* **type tls-proxy buffer** *buf_size* **packet-length** *bytes* **interface inside**	
To capture encrypted inbound and outbound data from the TLS proxy on one or more interfaces.	**capture** *capture_name* **type tls-proxy buffer** *buf_size* **packet-length** *bytes* **interface** *interface_nam*e	If signaling fails, you might require capturing decrypted packets to see the contents of the SIP and SCCP signaling message. Use the **type tls-proxy** option in the **capture** command.

Table 16-6 lists the **show** commands to use with the phone proxy.

Table 16-6 Security Appliance Show Commands to Use with the Phone Proxy

To	Use the Command	Notes
To show the packets or connections dropped by the accelerated security path.	**show asp drop**	Use this command to troubleshoot audio quality issues with the IP phones or other traffic issues with the phone proxy. In addition to running this command, get call status from the phone to check for any dropped packets or jitter. See Debugging Information from IP Phones, page 16-32.
To show the classifier contents of the accelerated security path for the specific classifier domain.	**show asp table classify domain** *domain_name*	If the IP phones are not downloading TFTP files, use this command to check that the classification rule for the domain `inspect-phone-proxy` is set for hosts to the configured TFTP server under the phone proxy instance. If the IP phones are failing to register, use this command to make sure there is a classification rule for the domain `app-redirect` set for the IP phones that cannot register.
To show the connections that are to the ASA or from the ASA, in addition to through-traffic connections.	**show conn all**	If you are experiencing problems with audio, use this command to make sure that there are connections opened from the IP phone to the media termination address. **Note** Use the **show conn** command with following options to display TFTP connections that have replicated (unused) connections: `hostname#` **`show conn | include p`** The output for the TFTP connections should have a "p" flag at the end: `UDP out 64.169.58.181:9014 in 192.168.200.101:39420 idle 0:01:51 bytes 522 flags p` Using this command shows that the phone proxy has connections that are going through "inspect-phone-proxy", which inspects TFTP connections. Using this command verifies that the TFTP requests are being inspected because the p flag is there.

Table 16-6 Security Appliance Show Commands to Use with the Phone Proxy

To	Use the Command	Notes
To show the logs in the buffer and logging settings.	**show logging**	Before entering the **show logging** command, enable the **logging buffered** command so that the **show logging command** displays the current message buffer and the current settings. Use this command to determine if the phone proxy and IP phones are successfully completing the TLS handshake. **Note** Using the **show logging** command is useful for troubleshooting many problems where packets might be denied or there are translation failures.
To show the corresponding media sessions stored by the phone proxy.	**show phone-proxy media-sessions**	Use this command to display output from successful calls. Additionally, use this command to troubleshoot problems with IP phone audio, such as one-way audio.
To show the IP phones capable of Secure mode stored in the database.	**show phone-proxy secure-phones**	For any problems, make sure there is an entry for the IP phone in this output and that the port for this IP phone is non-zero, which indicates that it has successfully registered with the Cisco UCM.
To show the corresponding signaling sessions stored by the phone proxy.	**show phone-proxy signaling-sessions**	Use this command to troubleshoot media or signaling failure.
To show the configured service policies.	**show service-policy**	Use this command to show statistics for the service policy.
To show active TLS proxy sessions related to the phone proxy.	**show tls-proxy sessions**	If the IP phone has failed to register, use this command to see if the IP phone has successfully completed the handshake with the TLS proxy configured for the phone proxy.

Debugging Information from IP Phones

On the IP phone, perform the following actions:

- Check the Status messages on the IP phone by selecting the **Settings** button > Status > Status Messages and selecting the status item that you want to view.
- Collect the call-statistics data from the IP phone by selecting the **Settings** button > Status > Call Statistic. Data like the following displays:

```
RxType: G.729                TxType: G.729
RxSize:   20 ms              TxSize:  20 ms
RxCnt: 0                     TxCnt: 014174
AvgJtr:   10                 MaxJtr:  59
RxDisc: 0000                 RxLost: 014001
```

- Check the Security settings on the IP phone by selecting the **Settings** button > Security Configuration. Settings for web access, Security mode, MIC, LSC, CTL file, trust list, and CAPF appear. Under Security mode, make sure the IP phone is set to Encrypted.
- Check the IP phone to determine which certificates are installed on the phone by selecting the **Settings** button > Security Configuration > Trust List. In the trustlist, verify the following:
  - Make sure that there is an entry for each entity that the IP phone will need to contact. If there is a primary and backup Cisco UCM, the trustlist should contain entries for each Cisco UCM.
  - If the IP phone needs an LSC, the record entry should contain a CAPF entry.
  - Make sure that the IP addresses listed for each entry are the mapped IP addresses of the entities that the IP phone can reach.
- Open a web browser and access the IP phone console logs at the URL `http://`*`IP_phone_IP address`*. The device information appears in the page. In the Device Logs section in the left pane, click Console Logs.

IP Phone Registration Failure

The following errors can make IP phones unable to register with the phone proxy:

- TFTP Auth Error Displays on IP Phone Console, page 16-33
- Configuration File Parsing Error, page 16-34
- Configuration File Parsing Error: Unable to Get DNS Response, page 16-34
- Non-configuration File Parsing Error, page 16-35
- Cisco UCM Does Not Respond to TFTP Request for Configuration File, page 16-35
- IP Phone Does Not Respond After the Security Appliance Sends TFTP Data, page 16-36
- IP Phone Requesting Unsigned File Error, page 16-37
- IP Phone Unable to Download CTL File, page 16-37
- IP Phone Registration Failure from Signaling Connections, page 16-38
- SSL Handshake Failure, page 16-40
- Certificate Validation Errors, page 16-41

TFTP Auth Error Displays on IP Phone Console

Problem The IP phone displays the following Status message:

```
TFTP Auth Error
```

Solution This Status message can indicate a problem with the IP phone CTL file.

To correct problems with the IP phone CTL file, perform the following:

Step 1 From the IP phone, select the **Setting** button > Security Configuration > Trust List. Verify that each entity in the network—Primary Cisco UCM, Secondary Cisco UCM, TFTP server—has its own entry in the trustlist and that each entity IP address is reachable by the IP phone.

Step 2 From the ASA, verify that the CTL file for the phone proxy contains one record entry for each entity in the network—Primary Cisco UCM, Secondary Cisco UCM, TFTP server—by entering the following command:

```
ciscoasa# show running-config all ctl-file [ctl_name]
```

Each of these record entries creates one entry on the IP phone trustlist. The phone proxy creates one entry internally with the function CUCM+TFTP.

Step 3 In the CTL file, verify that each IP address is the global or mapped IP address of the entity. If the IP phones are on multiple interfaces, additional addressing requirements apply. See Prerequisites for IP Phones on Multiple Interfaces, page 16-9.

Configuration File Parsing Error

Problem When the ASA receives the configuration file from the Cisco UCM and tries to parse it, the following error appears in the debug output (**debug phone-proxy tftp errors**):

```
PP: 192.168.10.5/49357 requesting SEP00010002003.cnf.xml.sgn
PP: opened 0x193166
.......
PP: Beginning of element tag is missing, got !
PP: error parsing config file
PP: Error modifying config file, dropping packet
```

Solution Perform the following actions to troubleshoot this problem:

Step 1 Enter the following URL in a web browser to obtain the IP phone configuration file from the Cisco Unified CM Administration console:

```
http://<cucm_ip>:6970/<config_file_name>
```

For example, if the Cisco UCM IP address is 128.106.254.2 and the IP phone configuration file name is SEP000100020003.cnf.xml, enter:

```
http://128.106.254.2:6970/SEP000100020003.cnf.xml
```

Step 2 Save this file, open a case with TAC and send them this file and the output from running the **debug phone-proxy tftp** command on the ASA.

Configuration File Parsing Error: Unable to Get DNS Response

Problem When the ASA receives the configuration file from the Cisco UCM and tries to parse it, the following error appears in the debug output (**debug phone-proxy tftp errors**):

```
PP: 192.168.10.5/49357 requesting SEP00010002003.cnf.xml.sgn
PP: opened 0x193166
.......
PP: Callback required for parsing config file
PP: Unable to get dns response for id 7
PP: Callback, error modifying config file
```

The error indicates that the Cisco UCM is configured as an FQDN and the phone proxy is trying to do a DNS lookup but failed to get a response.

Solution

Step 1 Verify that DNS lookup is configured on the ASA.

Step 2 If DNS lookup is configured, determine whether you can ping the FQDN for the Cisco UCM from the ASA.

Step 3 If ASA cannot ping the Cisco UCM FQDN, check to see if there is a problem with the DNS server.

Step 4 Additionally, use the **name** command to associate a name with an IP address with the FQDN. See the command reference for information about using the **name** command.

Non-configuration File Parsing Error

Problem The ASA receives a file other than an IP phone configuration file from the Cisco UCM and attempts to parse it. The following error appears in the debug output (**debug phone-proxy tftp**):

```
PP: 192.168.10.5/49357 requesting SK72f64050-7ad5-4b47-9bfa-5e9ad9cd4aa9.xml.sgn
PP: opened 0x193166
.......
PP: Beginning of element tag is missing, got !
PP: error parsing config file
PP: Error modifying config file, dropping packet
```

Solution The phone proxy should parse only the IP phone configuration file. When the phone proxy TFTP state gets out of state, the phone proxy cannot detect when it is attempting to parse a file other than the IP phone configuration file and the error above appears in the ASA output from the **debug phone-proxy tftp** command.

Perform the following actions to troubleshoot this problem:

Step 1 Reboot the IP phone.

Step 2 On the ASA, enter the following command to obtain the error information from the first TFTP request to the point where the first error occurred.

```
hostname# debug phone-proxy tftp
```

Step 3 Capture the packets from the IP phone to the ASA. Make sure to capture the packets on the interface facing the IP phone and the interface facing the Cisco UCM. See Debugging Information from the Security Appliance, page 16-28.

Step 4 Save this troubleshooting data, open a case with TAC and give them this information.

Cisco UCM Does Not Respond to TFTP Request for Configuration File

Problem When the ASA forwards the TFTP request to the Cisco UCM for the IP phone configuration file, the Cisco UCM does not respond and the following errors appear in the debug output (**debug phone-proxy tftp**):

```
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
```

```
PP: Client outside:192.168.10.5/49355 retransmitting request for Config file
SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
PP: Client outside:192.168.10.5/49355 retransmitting request for Config file
SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
PP: 192.168.10.5/49355 requesting SEP001562106AF3.cnf.xml.sgn
PP: Client outside:192.168.10.5/49355 retransmitting request for Config file
SEP001562106AF3.cnf.xml.sgn
PP: opened 0x17ccde
```

Solution Perform the following actions to troubleshoot this problem:

Step 1 Determine why the Cisco UCM is not responding to the TFTP request by performing the following troubleshooting actions:

- Use the Cisco UCM to ping the ASA inside interface when PAT is configured for the outside interface so that the IP phone IP address is uses NAT for the ASA inside interface IP address.
- Use the Cisco UCM to ping the IP phone IP address when NAT and PAT are not configured.

Step 2 Verify that the ASA is forwarding the TFTP request. Capture the packets on the interface between the ASA and Cisco UCM. See Debugging Information from the Security Appliance, page 16-28.

IP Phone Does Not Respond After the Security Appliance Sends TFTP Data

Problem When the ASA receives a TFTP request from the IP phone for the CTL file and forwards the data to the IP phone, the phone might not see the data and the TFTP transaction fails.

The following errors appear in the debug output (**debug phone-proxy tftp**):

```
PP: Client outside:68.207.118.9/33606 retransmitting request for CTL file
CTLSEP001DA2B78E91.tlv
PP: opened 0x214b27a
PP: Data Block 1 forwarded from 168.215.146.220/20168 to 68.207.118.9/33606 ingress ifc
outside
PP: 68.207.118.9/33606 requesting CTLSEP001DA2B78E91.tlv
PP: Client outside:68.207.118.9/33606 retransmitting request for CTL file
CTLSEP001DA2B78E91.tlv
PP: 68.207.118.9/33606 requesting CTLSEP001DA2B78E91.tlv
PP: Client outside:68.207.118.9/33606 retransmitting request for CTL file
CTLSEP001DA2B78E91.tlv
```

Solution Perform the following actions to determine why the IP phone is not responding and to troubleshoot the problem:

Step 1 Verify that the ASA is forwarding the TFTP request by entering the following command to capture the packets on the interface between the ASA and the IP phone:

```
hostname# capture out interface outside
```

See the command reference for more information about using the **capture** command.

Step 2 If the IP phone is behind a router, the router might be dropping the data. Make sure UDP port forwarding is enabled on the router.

Step 3 If the router is a Linksys router, see Configuring Linksys Routers with UDP Port Forwarding for the Phone Proxy, page 16-27 for information on the configuration requirements.

IP Phone Requesting Unsigned File Error

Problem The IP phone should always request a signed file. Therefore, the TFTP file being requested always has the .SGN extension.

When the IP phone does not request a signed file, the following error appears in the debug output (**debug phone-proxy tftp errors**):

```
Error: phone requesting for unsigned config file
```

Solution Most likely, this error occurs because the IP phone has not successfully installed the CTL file from the ASA.

Determine whether the IP phone has successfully downloaded and installed the CTL file from the ASA by checking the Status messages on the IP phone. See Debugging Information from IP Phones, page 16-32 for information.

IP Phone Unable to Download CTL File

Problem The IP phone Status message indicates it cannot download its CTL file and the IP phone cannot be converted to Secure (encrypted) mode.

Solution If the IP phone did not have an existing CTL file, check the Status messages by selecting the **Settings** button > Status > Status Messages. If the list contains a Status message indicating the IP phone encountered a CTL File Auth error, obtain the IP phone console logs, open a TAC case, and send them the logs.

Solution This error can appear in the IP phone Status messages when the IP phone already has an existing CTL file.

Step 1 Check the IP phone to see if a CTL file already exists on it. This can occur if the IP phone previously registered with a mixed mode cluster Cisco UCM. On the IP phone, select the **Settings** button > Security Configuration > CTL file.

Step 2 Erase the existing CTL file by selecting the **Settings** button > Security Configuration > CTL file > Select. Press **# on the keypad and select Erase.

Solution Problems downloading the CTL file might be caused by issues with media termination. Enter the following command to determine if the media-termination address in the phone proxy configuration is set correctly:

```
hostname(config)# show running-config all phone-proxy
!
phone-proxy mypp
    media-termination address 10.10.0.25
    cipc security-mode authenticated
    cluster-mode mixed
    disable service-settings
    timeout secure-phones 0:05:00
hostname(config)#
```

Make sure that each media-termination instance is created correctly and that the address or addresses are set correctly. The ASA must meet specific criteria for media termination. See Media Termination Instance Prerequisites, page 16-6 for the complete list of prerequisites that you must follow when creating the media termination instance and configuring the media termination addresses.

IP Phone Registration Failure from Signaling Connections

Problem The IP phone is unable to complete the TLS handshake with the phone proxy and download its files using TFTP.

Solution

Step 1 Determine if the TLS handshake is occurring between the phone proxy and the IP phone, perform the following:

a. Enable logging with the following command:

```
hostname(config)# logging buffered debugging
```

b. To check the output from the syslogs captured by the **logging buffered** command, enter the following command:

```
hostname# show logging
```

The syslogs will contain information showing when the IP phone is attempting the TLS handshake, which happens after the IP phone downloads its configuration file.

Step 2 Determine if the TLS proxy is configured correctly for the phone proxy:

a. Display all currently running TLS proxy configurations by entering the following command:

```
hostname# show running-config tls-proxy
    tls-proxy proxy
    server trust-point _internal_PP_<ctl_file_instance_name>
    client ldc issuer ldc_signer
    client ldc key-pair phone_common
    no client cipher-suite
hostname#
```

b. Verify that the output contains the **server trust-point** command under the **tls-proxy** command (as shown in substep a.).

If you are missing the **server trust-point** command, modify the TLS proxy in the phone proxy configuration.

See Step 3 in the "Task Flow for Configuring the Phone Proxy in a Non-secure Cisco UCM Cluster" section on page 16-15, or Step 3 in the "Task Flow for Configuring the Phone Proxy in a Mixed-mode Cisco UCM Cluster" section on page 16-17.

Having this command missing from the TLS proxy configuration for the phone proxy will cause TLS handshake failure.

Step 3 Verify that all required certificates are imported into the ASA so that the TLS handshake will succeed.

a. Determine which certificates are installed on the ASA by entering the following command:

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

Additionally, determine which certificates are installed on the IP phones. See Debugging Information from IP Phones, page 16-32 for information about checking the IP phone to determine if it has MIC installed on it.

b. Verify that the list of installed certificates contains all required certificates for the phone proxy.

   See Table 16-2, Certificates Required by the Security Appliance for the Phone Proxy, for information.

c. Import any missing certificates onto the ASA. See also Importing Certificates from the Cisco UCM, page 16-15.

Step 4 If the steps above fail to resolve the issue, perform the following actions to obtain additional troubleshooting information for Cisco Support.

a. Enter the following commands to capture additional debugging information for the phone proxy:

```
hostname# debug inspect tls-proxy error
hostname# show running-config ssl
hostname(config) show tls-proxy tls_name session host host_addr detail
```

b. Enable the **capture** command on the inside and outside interfaces (IP phones and Cisco UCM) to enable packet capture capabilities for packet sniffing and network fault isolation. See the command reference for information.

Problem The TLS handshake succeeds, but signaling connections are failing.

Solution Perform the following actions:

- Check to see if SIP and Skinny signaling is successful by using the following commands:
  - **debug sip**
  - **debug skinny**
- If the TLS handshake is failing and you receive the following syslog, the SSL encryption method might not be set correctly:

```
%ASA-6-725001: Starting SSL handshake with client dmz:171.169.0.2/53097 for TLSv1
session.
%ASA-7-725010: Device supports the following 1 cipher(s).
%ASA-7-725011: Cipher[1] : RC4-SHA
%ASA-7-725008: SSL client dmz:171.169.0.2/53097 proposes the following 2 cipher(s).
%ASA-7-725011: Cipher[1] : AES256-SHA
%ASA-7-725011: Cipher[2] : AES128-SHA
%ASA-7-725014: SSL lib error. Function: SSL3_GET_CLIENT_HELLO Reason: no shared cipher
%ASA-6-725006: Device failed SSL handshake with dmz client:171.169.0.2/53097

Set the correct ciphers by completing the following procedure:
```

Step 1 To see the ciphers being used by the phone proxy, enter the following command:

```
hostname# show run all ssl
```

Step 2 To add the required ciphers, enter the following command:

```
hostname(config)# ssl encryption
```

The default is to have all algorithms available in the following order:

[3des-sha1] [des-sha1] [rc4-md5] [possibly others]

See the command reference for more information about setting ciphers with the **ssl encryption** command.

SSL Handshake Failure

Problem The phone proxy is not functioning. Initial troubleshooting uncovered the following errors in the ASA syslogs:

```
%ASA-7-725014: SSL lib error. Function: SSL3_READ_BYTES Reason: ssl handshake failure
%ASA-7-725014: SSL lib error. Function: SSL3_GET_CLIENT_CERTIFICATE Reason: no certificate
returned
%ASA-6-725006: Device failed SSL handshake with outside client:72.146.123.158/30519
%ASA-3-717009: Certificate validation failed. No suitable trustpoints found to validate
certificate serial number: 62D06172000000143FCC, subject name:
cn=CP-7962G-SEP002155554502,ou=EVVBU,o=Cisco Systems Inc.
%ASA-3-717027: Certificate chain failed validation. No suitable trustpoint was found to
validate chain.
```

Solution

Verify that all required certificates are imported into the ASA so that the TLS handshake will succeed.

Step 1 Determine which certificates are installed on the ASA by entering the following command:

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

Additionally, determine which certificates are installed on the IP phones. See Debugging Information from IP Phones, page 16-32 for information about checking the IP phone to determine if it has MIC installed on it.

Step 2 Verify that the list of installed certificates contains all required certificates for the phone proxy.

See Table 16-2, Certificates Required by the Security Appliance for the Phone Proxy, for information.

Step 3 Import any missing certificates onto the ASA. See also Importing Certificates from the Cisco UCM, page 16-15.

Problem The phone proxy is not functioning. Initial troubleshooting uncovered the following errors in the ASA syslogs:

```
%ASA-6-725001: Starting SSL handshake with client dmz:171.169.0.2/53097 for TLSv1
session.
%ASA-7-725010: Device supports the following 1 cipher(s).
%ASA-7-725011: Cipher[1] : RC4-SHA
%ASA-7-725008: SSL client dmz:171.169.0.2/53097 proposes the following 2 cipher(s).
%ASA-7-725011: Cipher[1] : AES256-SHA
%ASA-7-725011: Cipher[2] : AES128-SHA
%ASA-7-725014: SSL lib error. Function: SSL3_GET_CLIENT_HELLO Reason: no shared cipher
%ASA-6-725006: Device failed SSL handshake with dmz client:171.169.0.2/53097
```

Solution the SSL encryption method might not be set correctly. Set the correct ciphers by completing the following procedure:

Step 1 To see the ciphers being used by the phone proxy, enter the following command:

```
hostname# show run all ssl
```

Step 2 To add the required ciphers, enter the following command:

```
hostname(config)# ssl encryption
```

The default is to have all algorithms available in the following order:

[3des-sha1] [des-sha1] [rc4-md5] [possibly others]

See the command reference for more information about setting ciphers with the **ssl encryption** command.

Certificate Validation Errors

Problem Errors in the ASA log indicate that certificate validation errors occurred.

Entering the **show logging asdm** command, displayed the following errors:

```
3|Jun 19 2008 17:23:54|717009: Certificate validation failed. No suitable trustpoints
found to validate
certificate serial number: 348FD2760000000E6E27, subject name:
cn=CP-7961G-SEP001819A89CC3,ou=EVVBU,o=Cisco Systems Inc.
```

Solution

In order for the phone proxy to authenticate the MIC provided by the IP phone, it needs the Cisco Manufacturing CA (MIC) certificate imported into the ASA.

Verify that all required certificates are imported into the ASA so that the TLS handshake will succeed.

Step 1 Determine which certificates are installed on the ASA by entering the following command:

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

Additionally, determine which certificates are installed on the IP phones. The certificate information is shown under the Security Configuration menu. See Debugging Information from IP Phones, page 16-32 for information about checking the IP phone to determine if it has the MIC installed on it.

Step 2 Verify that the list of installed certificates contains all required certificates for the phone proxy.

See Table 16-2, Certificates Required by the Security Appliance for the Phone Proxy, for information.

Step 3 Import any missing certificates onto the ASA. See also Importing Certificates from the Cisco UCM, page 16-15.

Media Termination Address Errors

Problem Entering the **media-termination address** command displays the following errors:

```
hostname(config-phone-proxy)# media-termination address ip_address
ERROR: Failed to apply IP address to interface Virtual254, as the network overlaps with
interface GigabitEthernet0/0. Two interfaces cannot be in the same subnet.
ERROR: Failed to set IP address for the Virtual interface
ERROR: Could not bring up Phone proxy media termination interface
ERROR: Failed to find the HWIDB for the Virtual interface
```

Solution Enter the following command to determine if the media-termination address in the phone proxy configuration is set correctly:

```
hostname(config)# show running-config all phone-proxy
asa2(config)# show running-config all phone-proxy
!
```

```
phone-proxy mypp
    media-termination address 10.10.0.25
    cipc security-mode authenticated
    cluster-mode mixed
    disable service-settings
    timeout secure-phones 0:05:00
hostname(config)#
```

Make sure that each media-termination instance is created correctly and that the address or addresses are set correctly. The ASA must meet specific criteria for media termination. See Media Termination Instance Prerequisites, page 16-6 for the complete list of prerequisites that you must follow when creating the media termination instance and configuring the media termination addresses.

Audio Problems with IP Phones

The following audio errors can occur when the IP phones connecting through the phone proxy.

Media Failure for a Voice Call

Problem The call signaling completes but there is one way audio or no audio.

Solution

- Problems with one way or no audio might be caused by issues with media termination. Enter the following command to determine if the media-termination address in the phone proxy configuration is set correctly:

```
hostname(config)# show running-config all phone-proxy
asa2(config)# show running-config all phone-proxy
!
phone-proxy mypp
    media-termination address 10.10.0.25
    cipc security-mode authenticated
    cluster-mode mixed
    disable service-settings
    timeout secure-phones 0:05:00
hostname(config)#
```

- Make sure that each media-termination instance is created correctly and that the address or addresses are set correctly. The ASA must meet specific criteria for media termination. See Media Termination Instance Prerequisites, page 16-6 for the complete list of prerequisites that you must follow when creating the media termination instance and configuring the media termination addresses.
- If each media-termination address meets the requirements, determine whether the IP addresses are reachable by all IP phones.
- If each IP address is set correctly and reachable by all IP phones, check the call statistics on an IP phone (see Debugging Information from IP Phones, page 16-32) and determine if there are Rcvr packets and Sender packets on the IP phone, or if there are any Rcvr Lost or Discarded packets.

Saving SAST Keys

Site Administrator Security Token (SAST) keys on the ASA can be saved in the event a recovery is required due to hardware failure and a replacement is required. The following steps shows how to recover the SAST keys and use them on the new hardware.

The SAST keys can be seen via the **show crypto key mypubkey rsa** command. The SAST keys are associated with a trustpoint that is labeled **_internal_***ctl-file_name***_SAST_***X* where *ctl-file-name* is the name of the CTL file instance that was configured, and *X* is an integer from 0 to N-1 where N is the number of SASTs configured for the CTL file (the default is 2).

Step 1 On the ASA, export all the SAST keys in PKCS-12 format by using the **crypto ca export** command:

```
hostname(config)# crypto ca export _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Exported pkcs12 follows:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]

MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH
---End - This line not part of the pkcs12---

hostname(config)# crypto ca export _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Exported pkcs12 follows:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]

mGF/hfDDNAICBAA=

---End - This line not part of the pkcs12---
hostname(config)#
```

Note Save this output somewhere secure.

Step 2 Import the SAST keys to a new ASA.

a. To import the SAST key, enter the following command:

```
hostname(config)# crypto ca import trustpoint pkcs12 passphrase
```

Where *trustpoint* is **_internal_***ctl-file_name***_SAST_***X* and *ctl-file-name* is the name of the CTL file instance that was configured, and *X* is an integer from 0 to 4 depending on what you exported from the ASA.

b. Using the PKCS-12 output you saved in Step 1, enter the following command and paste the output when prompted:

```
hostname(config)# crypto ca import _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Enter the base 64 encoded pkcs12.
hostname(config)# End with the word "quit" on a line by itself:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]

muMiZ6eClQICBAA=
hostname(config)# quit
INFO: Import PKCS12 operation completed successfully
hostname(config)# crypto ca import _internal_ctl-file_name_SAST_X pkcs12 passphrase

hostname(config)# Enter the base 64 encoded pkcs12.
hostname(config)# End with the word "quit" on a line by itself:
MIIGZwIBAzCCBiEGCSqGSIb3DQEHAaCCBhIEggYOMIIGCjCCBgYGCSqGSIb3DQEH

[snip]
```

```
mGF/hfDDNAICBAA=
hostname(config)# quit
INFO: Import PKCS12 operation completed successfully
hostname(config)#
```

Step 3 Create the CTL file instance on the new ASA using the same name as the one used in the SAST trustpoints created in Step 2 by entering the following commands. Create trustpoints for each Cisco UMC (primary and secondary).

```
hostname(config)# ctl-file ctl_name
hostname(config-ctl-file)# record-entry cucm trustpoint trust_point address address
hostname(config-ctl-file)# record-entry capf trustpoint trust_point address address
hostname(config-ctl-file)# no shutdown
```

Configuration Examples for the Phone Proxy

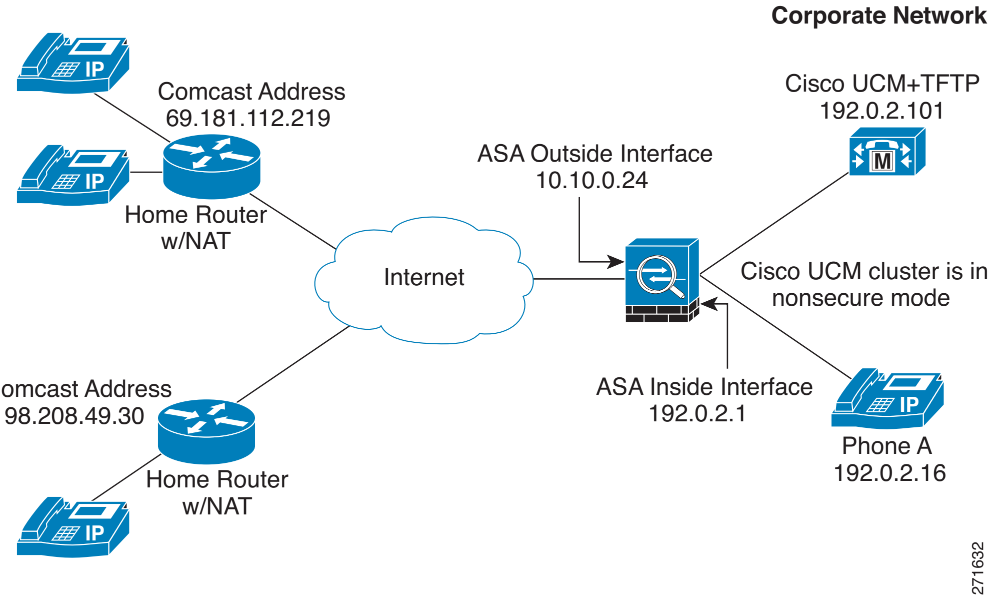
This section includes the following topics:

- Example 1: Nonsecure Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher, page 16-44
- Example 2: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher, page 16-46
- Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers, page 16-47
- Example 4: Mixed-mode Cisco UCM cluster, Primary Cisco UCM, Secondary and TFTP Server on Different Servers, page 16-48
- Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher, page 16-50
- Example 6: VLAN Transversal, page 16-52

Example 1: Nonsecure Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher

Figure 16-2 shows an example of the configuration for a non-secure Cisco UCM cluster using the following topology.

Figure 16-2 Nonsecure Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher

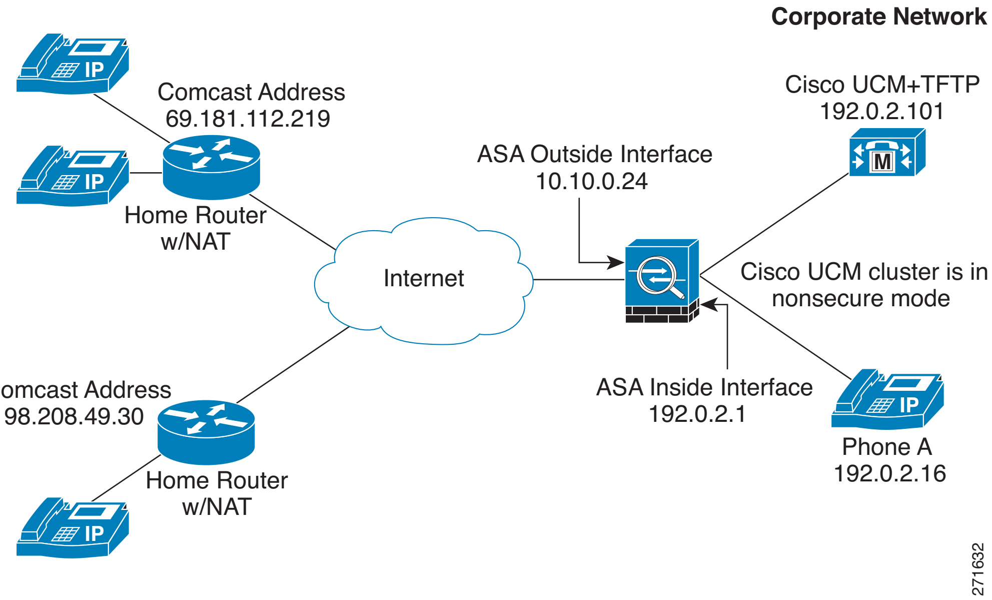


```
object network obj-192.0.2.101
    host 192.0.2.101
    nat (inside,outside) static 10.10.0.26
access-list pp extended permit udp any host 10.10.0.26 eq 69
access-group pp in interface outside
crypto key generate rsa label cucmtftp_kp modulus 1024
crypto ca trustpoint cucm_tftp_server
    enrollment self
    keypair cucmtftp_kp
crypto ca enroll cucm_tftp_server
ctl-file myctl
    record-entry cucm-tftp trustpoint cucm_tftp_server address 10.10.0.26
    no shutdown
tls-proxy mytls
    server trust-point _internal_PP_myctl
media-termination my_mediaterm
    address 192.0.2.25 interface inside
    address 10.10.0.25 interface outside
phone-proxy mypp
    media-termination my_mediaterm
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 2: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher

Figure 16-3 shows an example of the configuration for a mixed-mode Cisco UCM cluster using the following topology.

Figure 16-3 Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Publisher



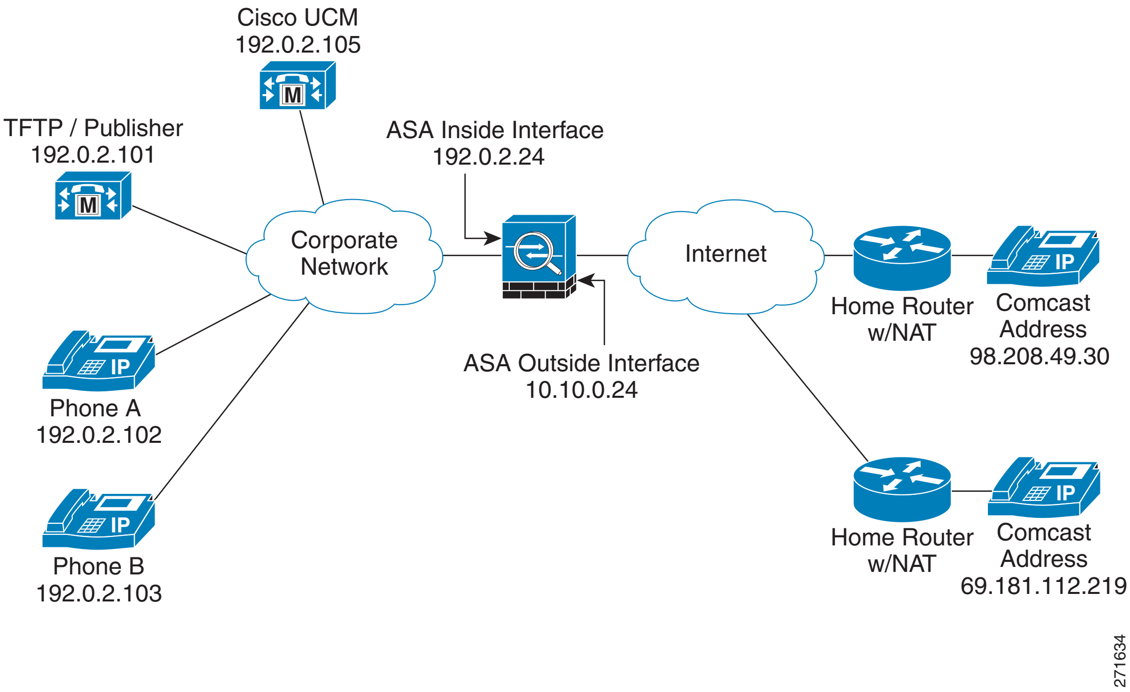
```
object network obj-192.0.2.101
    host 192.0.2.101
    nat (inside,outside) static 10.10.0.26
access-list pp extended permit udp any host 10.10.0.26 eq 69
access-group pp in interface outside
crypto key generate rsa label cucmtftp_kp modulus 1024
crypto ca trustpoint cucm_tftp_server
    enrollment self
    keypair cucmtftp_kp
crypto ca enroll cucm_tftp_server
ctl-file myctl
    record-entry cucm-tftp trustpoint cucm_tftp_server address 10.10.0.26
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
    crypto ca enroll ldc_server
tls-proxy my_proxy
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
media-termination my_mediaterm
    address 192.0.2.25 interface inside
    address 10.10.0.25 interface outside
phone-proxy mypp
    media-termination my_mediaterm
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 3: Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers

Figure 16-4 shows an example of the configuration for a mixed-mode Cisco UCM cluster using the following topology where the TFTP server resides on a different server from the Cisco UCM.

In this sample, the static interface PAT for the TFTP server is configured to appear like the ASA's outside interface IP address.

Figure 16-4 Mixed-mode Cisco UCM cluster, Cisco UCM and TFTP Server on Different Servers



```
object network obj-192.0.2.105
    host 192.0.2.105
    nat (inside,outside) static 10.10.0.26
object network obj-192.0.2.101
```

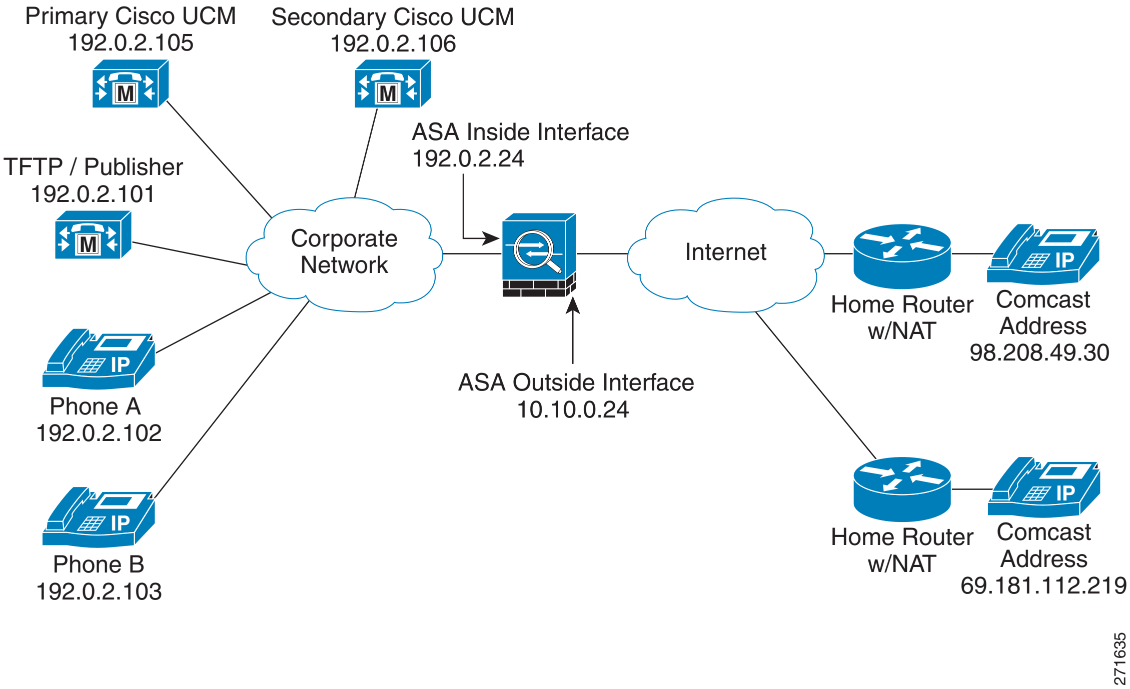
```
    host 192.0.2.101
    nat (inside,outside) static interface udp 69 69
access-list pp extended permit udp any host 10.10.0.24 eq 69
access-group pp in interface outside
crypto key generate rsa label cucm_kp modulus 1024
crypto ca trustpoint cucm
    enrollment self
    keypair cucm_kp
crypto ca enroll cucm
crypto key generate rsa label tftp_kp modulus 1024
crypto ca trustpoint tftp_server
    enrollment self
    keypair tftp_kp
crypto ca enroll tftp_server
ctl-file myctl
    record-entry cucm trustpoint cucm_server address 10.10.0.26
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
    crypto ca enroll ldc_server
tls-proxy my_proxy
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
media-termination my_mediaterm
    address 192.0.2.25 interface inside
    address 10.10.0.25 interface outside
phone-proxy mypp
    media-termination my_mediaterm
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 4: Mixed-mode Cisco UCM cluster, Primary Cisco UCM, Secondary and TFTP Server on Different Servers

Figure 16-5 shows an example of the configuration for a mixed-mode Cisco UCM cluster using the following topology where the TFTP server resides on a different server from the primary and secondary Cisco UCMs.

In this sample, the static interface PAT for the TFTP server is configured to appear like the ASA’s outside interface IP address.

Figure 16-5 Mixed-mode Cisco UCM cluster, Primary Cisco UCM, Secondary Cisco UCM, and TFTP Server on Different Servers



```
object network obj-192.0.2.105
    host 192.0.2.105
    nat (inside,outside) static 10.10.0.27
object network obj-192.0.2.101
    host 192.0.2.101
    nat (inside,outside) static interface udp 69 69
object network obj-192.0.2.106
    host 192.0.2.106
    nat (inside,outside) static 10.10.0.26
access-list pp extended permit udp any host 10.10.0.24 eq 69
access-group pp in interface outside
crypto key generate rsa label cluster_kp modulus 1024
crypto ca trustpoint pri_cucm
    enrollment self
    keypair cluster_kp
crypto ca enroll pri_cucm
crypto ca trustpoint sec_cucm
    enrollment self
    serial-number
    keypair cluster_kp
crypto ca enroll sec_cucm
crypto ca trustpoint tftp-server
    enrollment self
    fqdn my-tftp.example.com
    keypair cluster-kp
crypto ca enroll tftp_server
ctl-file myctl
    record-entry tftp trustpoint tftp_server address 10.10.0.24
    record-entry cucm trustpoint pri_cucm_server address 10.10.0.27
    record-entry cucm trustpoint sec_cucm_server address 10.10.0.2
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
    crypto ca enroll ldc_server
tls-proxy my_proxy
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
media-termination my_mediaterm
    address 192.0.2.25 interface inside
    address 10.10.0.25 interface outside
phone-proxy mypp
    media-termination my_mediaterm
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 5: LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher

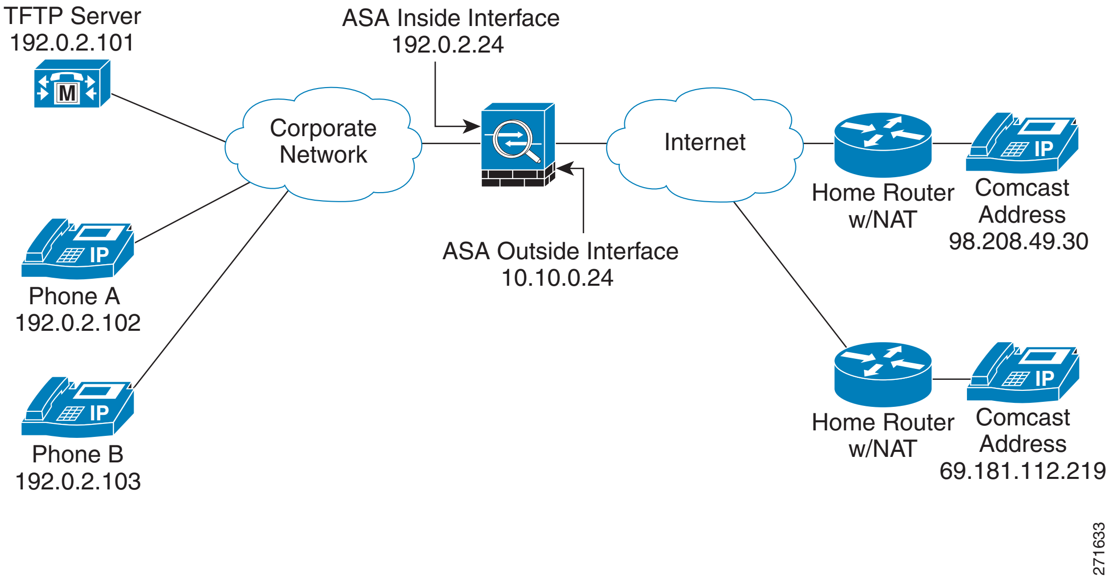
Figure 16-6 shows an example of the configuration for a mixed-mode Cisco UCM cluster where LSC provisioning is required using the following topology.

Note Doing LSC provisioning for remote IP phones is not recommended because it requires that the IP phones first register and they have to register in nonsecure mode. Having the IP phones register in nonsecure mode requires the Administrator to open the nonsecure signaling port for SIP and SCCP on the ASA. If possible, LSC provisioning should be done inside the corporate network before giving the IP phones to the end-users.

In this sample, you create an ACL to allow the IP phones to contact the TFTP server and to allow the IP phones to register in nonsecure mode by opening the nonsecure port for SIP and SCCP as well as the CAPF port for LSC provisioning.

Additionally, you create the CAPF trustpoint by copying and pasting the CAPF certificate from the Cisco UCM Certificate Management software.

Figure 16-6 LSC Provisioning in Mixed-mode Cisco UCM cluster; Cisco UCM and TFTP Server on Publisher



```
object network obj-192.0.2.105
    host 192.0.2.105
    nat (inside,outside) static 10.10.0.26
object network obj-192.0.2.101
    host 192.0.2.101
    nat (inside,outside) static interface udp 69 69
access-list pp extended permit udp any host 10.10.0.24 eq 69
access-list pp extended permit tcp any host 10.10.0.26 eq 2000
access-list pp extended permit tcp any host 10.10.0.26 eq 5060
access-list pp extended permit tcp any host 10.10.0.26 eq 3804
access-group pp in interface outside
crypto key generate rsa label cluster_kp modulus 1024
crypto ca trustpoint cucm
    enrollment self
    keypair cluster_kp
crypto ca enroll cucm
crypto ca trustpoint tftp_server
    enrollment self
    serial-number
    keypair cluster_kp
crypto ca enroll tftp_server
crypto ca trustpoint capf
    enroll terminal
crypto ca authenticate capf
ctl-file myctl
    record-entry cucm trustpoint cucm_server address 10.10.0.26
    record-entry capf trustpoint capf address 10.10.0.26
    no shutdown
crypto key generate rsa label ldc_signer_key modulus 1024
crypto key generate rsa label phone_common modulus 1024
crypto ca trustpoint ldc_server
    enrollment self
    proxy_ldc_issuer
    fqdn my-ldc-ca.exmaple.com
    subject-name cn=FW_LDC_SIGNER_172_23_45_200
    keypair ldc_signer_key
    crypto ca enroll ldc_server
tls-proxy my_proxy
```

```
    server trust-point _internal_PP_myctl
    client ldc issuer ldc_server
    client ldc keypair phone_common
    client cipher-suite aes128-sha1 aes256-sha1
media-termination my_mediaterm
    address 192.0.2.25 interface inside
    address 10.10.0.25 interface outside
phone-proxy mypp
    media-termination my_mediaterm
    tftp-server address 192.0.2.101 interface inside
    tls-proxy mytls
    ctl-file myctl
    cluster-mode mixed
class-map sec_sccp
    match port tcp 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface outside
```

Example 6: VLAN Transversal

Figure 16-7 shows an example of the configuration to force Cisco IP Communicator (CIPC) softphones to operate in authenticated mode when CIPC softphones are deployed in a voice and data VLAN scenario. VLAN transversal is required between CIPC softphones on the data VLAN and hard phones on the voice VLAN.

In this sample, the Cisco UCM cluster mode is nonsecure.

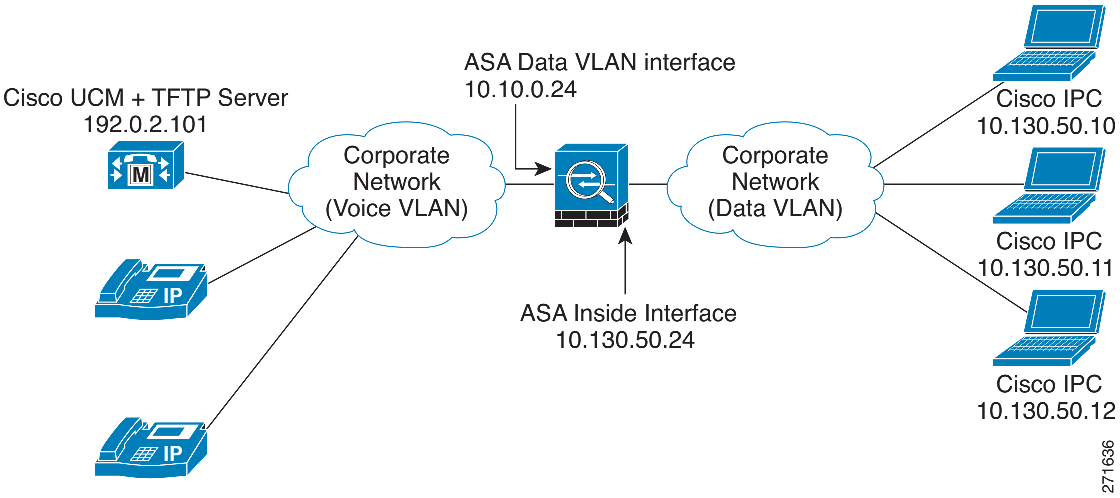
In this sample, you create an ACL to allow the IP phones to contact the TFTP server and to allow the IP phones to register in nonsecure mode by opening the nonsecure port for SIP and SCCP as well as the CAPF port for LSC provisioning.

In this sample, you configure NAT for the CIPC by using PAT so that each CIPC is mapped to an IP address space in the Voice VLAN.

Additionally, you create the CAPF trustpoint by copying and pasting the CAPF certificate from the Cisco UCM Certificate Management software.

Note Cisco IP Communicator supports authenticated mode only and does not support encrypted mode; therefore, there is no encrypted voice traffic (SRTP) flowing from the CIPC softphones.

Figure 16-7 VLAN Transversal Between CIPC Softphones on the Data VLAN and Hard Phones on the Voice VLAN



```
object network obj-10.130.50.0
    subnet 10.130.50.0 255.255.255.0
    nat (data,voice) dynamic 192.0.2.10
object network obj-10.130.50.5
    host 10.130.50.5
    nat (data,voice) static 192.0.2.101
access-list pp extended permit udp any host 10.130.50.5 eq 69
access-list pp extended permit tcp any host 10.130.50.5 eq 2000
access-list pp extended permit tcp any host 10.130.50.5 eq 5060
access-list pp extended permit tcp any host 10.130.50.5 eq 3804
access-group pp in interface data
crypto ca generate rsa label cucmtftp_kp modulus 1024
crypto ca trustpoint cucm_tftp_server
    enrollment self
    keypair cucmtftp_kp
crypto ca enroll cucm_tftp_server
crypto ca trustpoint capf
    enrollment terminal
crypto ca authenticate capf
ctl-file myctl
    record-entry cucm-tftp trustpoint cucm_tftp_server address 10.130.50.5
    record-entry capf trustpoint capf address 10.130.50.5
    no shutdown
tls-proxy mytls
    server trust-point _internal_PP_myctl
media-termination my_mediaterm
    address 10.130.50.2
phone-proxy mypp
    media-termination my_mediaterm
    tftp-server address 10.10.0.20 interface inside
    tls-proxy mytls
    ctl-file myctl
    cipc security-mode authenticated
class-map sec_sccp
    match port tcp eq 2443
class-map sec_sip
    match port tcp eq 5061
policy-map pp_policy
    class sec_sccp
        inspect skinny phone-proxy mypp
    class sec_sip
        inspect sip phone-proxy mypp
service-policy pp_policy interface data
```

Feature History for the Phone Proxy

Table 16-7 lists the release history for this feature.

Table 16-7 Feature History for Cisco Phone Proxy

Feature Name	Releases	Feature Information
Cisco Phone Proxy	8.0(4)	The phone proxy feature was introduced. The following new commands were introduced. **cipc security-mode authenticated**, **clear configure ctl**, **clear configure phone-proxy**, **cluster-ctl-file**, **cluster-mode nonsecure**, **ctl-file (global)**, **ctl-file (phone proxy)**, **debug phone proxy**, **disable service-settings**, **media-termination address**, **phone-proxy**, **proxy-server**, **record-entry**, **sast**, **show phone-proxy**, **show running-config ctl**, s**how running-config phone-proxy**, **timeout secure-phones**, **tftp-server address**.
NAT for the media termination address	8.1(2)	The **media-termination address** command was changed to allow for NAT: [**no**] **media-termination address** *ip_address* **interface** *intf_name* Where the **interface** *inft_name* keyword was added. The **rtp-min-port** and **rtp-max-ports** keywords were removed from the command syntax and included as a separate command: **rtp-min-port** *port1* **rtp-max-port** *port2*

Configuring the TLS Proxy for Encrypted Voice Inspection

This chapter describes how to configure the ASA for the TLS Proxy for Encrypted Voice Inspection feature.

This chapter includes the following sections:

Information about the TLS Proxy for Encrypted Voice Inspection

End-to-end encryption often leaves network security appliances “blind” to media and signaling traffic, which can compromise access control and threat prevention security functions. This lack of visibility can result in a lack of interoperability between the firewall functions and the encrypted voice, leaving businesses unable to satisfy both of their key security requirements.

The ASA is able to intercept and decrypt encrypted signaling from Cisco encrypted endpoints to the Cisco Unified Communications Manager (Cisco UCM), and apply the required threat protection and access control. It can also ensure confidentiality by re-encrypting the traffic onto the Cisco UCM servers.

Typically, the ASA TLS Proxy functionality is deployed in campus unified communications network. This solution is ideal for deployments that utilize end to end encryption and firewalls to protect Unified Communications Manager servers.

Decryption and Inspection of Unified Communications Encrypted Signaling

With encrypted voice inspection, the security appliance decrypts, inspects and modifies (as needed, for example, performing NAT fixup), and re-encrypts voice signaling traffic while all of the existing VoIP inspection functions for Skinny and SIP protocols are preserved. Once voice signaling is decrypted, the plaintext signaling message is passed to the existing inspection engines.

The security appliance acts as a TLS proxy between the Cisco IP Phone and Cisco UCM. The proxy is transparent for the voice calls between the phone and theCisco UCM. Cisco IP Phones download a Certificate Trust List from the Cisco UCM before registration which contains identities (certificates) of the devices that the phone should trust, such as TFTP servers and Cisco UCM servers. To support server proxy, the CTL file must contain the certificate that the security appliance creates for the Cisco UCMs. To proxy calls on behalf of the Cisco IP Phone, the security appliance presents a certificate that the Cisco UCM can verify, which is a Local Dynamic Certificate for the phone, issued by the certificate authority on the security appliance.

TLS proxy is supported by the Cisco Unified CallManager Release 5.1 and later. You should be familiar with the security features of the Cisco UCM. For background and detailed description of Cisco UCM security, see the Cisco Unified CallManager document:

http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_0/sec_vir/ae/sec504/index.htm

TLS proxy applies to the encryption layer and must be configured with an application layer protocol inspection. You should be familiar with the inspection features on the ASA, especially Skinny and SIP inspection.

Supported Cisco UCM and IP Phones for the TLS Proxy

Cisco Unified Communications Manager

The following releases of the Cisco Unified Communications Manager are supported with the TLS proxy:

- Cisco Unified CallManager Version 4.*x*
- Cisco Unified CallManager Version 5.0
- Cisco Unified CallManager Version 5.1
- Cisco Unified Communications Manager 6.1
- Cisco Unified Communications Manager 7.0
- Cisco Unified Communications Manager 8.0

Cisco Unified IP Phones

The following IP phones in the Cisco Unified IP Phones 7900 Series are supported with the TLS proxy:

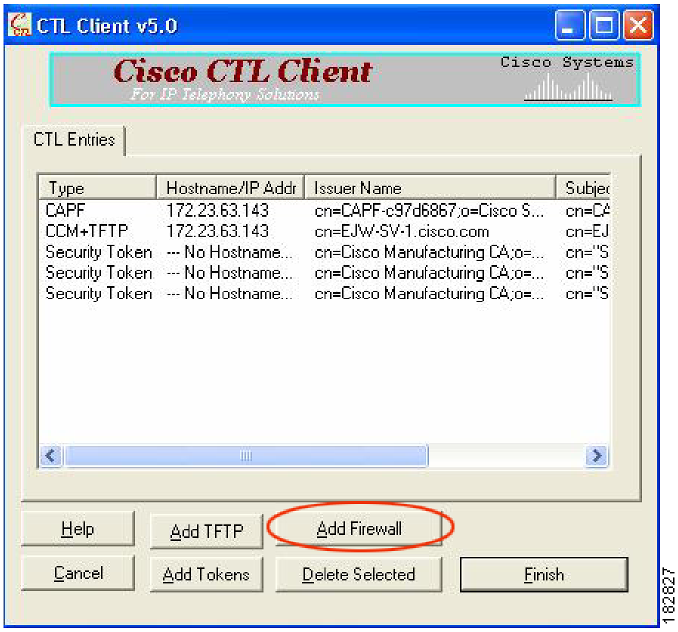
- Cisco Unified IP Phone 7985
- Cisco Unified IP Phone 7975
- Cisco Unified IP Phone 7971
- Cisco Unified IP Phone 7970
- Cisco Unified IP Phone 7965
- Cisco Unified IP Phone 7962
- Cisco Unified IP Phone 7961
- Cisco Unified IP Phone 7961G-GE
- Cisco Unified IP Phone 7960
- Cisco Unified IP Phone 7945
- Cisco Unified IP Phone 7942
- Cisco Unified IP Phone 7941

- Cisco Unified IP Phone 7941G-GE
- Cisco Unified IP Phone 7940
- Cisco Unified Wireless IP Phone 7921
- Cisco Unified Wireless IP Phone 7925
- Cisco IP Communicator (CIPC) for softphones

CTL Client Overview

The CTL Client application supplied by Cisco Unified CallManager Release 5.1 and later supports a TLS proxy server (firewall) in the CTL file. Figure 17-1 through Figure 17-4 illustrate the TLS proxy features supported in the CTL Client.

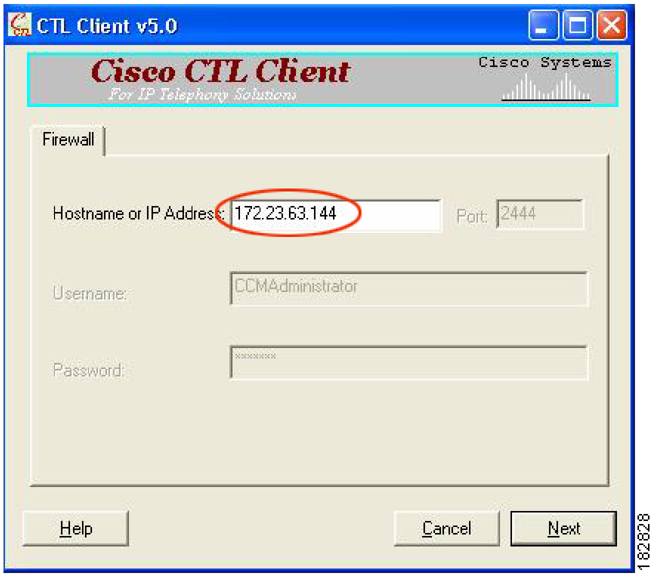
Figure 17-1 CTL Client TLS Proxy Features — Add Firewall



Type	Hostname/IP Addr	Issuer Name	Subjec
CAPF	172.23.63.143	cn=CAPF-c97d6867;o=Cisco S...	cn=CA
CCM+TFTP	172.23.63.143	cn=EJW-SV-1.cisco.com	cn=EJ
Security Token	--- No Hostname...	cn=Cisco Manufacturing CA;o=...	cn="S
Security Token	--- No Hostname...	cn=Cisco Manufacturing CA;o=...	cn="S
Security Token	--- No Hostname...	cn=Cisco Manufacturing CA;o=...	cn="S

Figure 17-1 shows support for adding a CTL entry consisting of the security appliance as the TLS proxy.

Figure 17-2 CTL Client TLS Proxy Features — ASA IP Address or Domain Name



Hostname or IP Address: 172.23.63.144

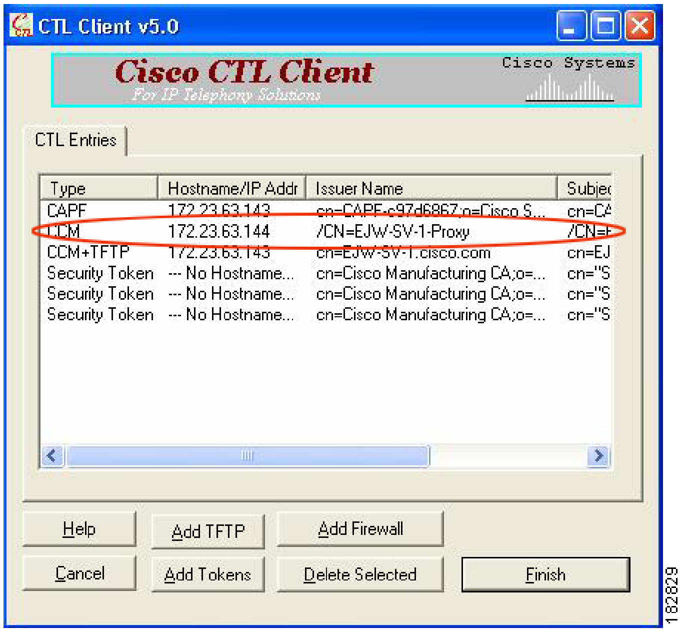
Port: 2444

Username: CCMAdministrator

Password: *******

[Figure 17-2](#) shows support for entering the security appliance IP address or domain name in the CTL Client.

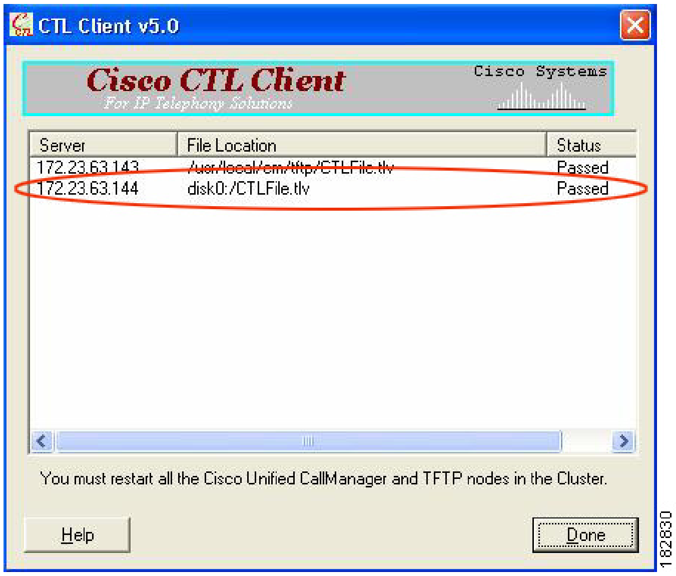
Figure 17-3 CTL Client TLS Proxy Features — CTL Entry for ASA



Type	Hostname/IP Addr	Issuer Name	Subjec
CAPF	172.23.63.143	cn=CAPF-c97d6867;o=Cisco S...	cn=CA
CCM	172.23.63.144	/CN=EJW-SV-1-Proxy	/CN=E
CCM+TFTP	172.23.63.143	cn=EJW-SV-1.cisco.com	cn=EJ
Security Token	--- No Hostname...	cn=Cisco Manufacturing CA;o=...	cn="S
Security Token	--- No Hostname...	cn=Cisco Manufacturing CA;o=...	cn="S
Security Token	--- No Hostname...	cn=Cisco Manufacturing CA;o=...	cn="S

[Figure 17-3](#) shows that the CTL entry for the security appliance as the TLS proxy has been added. The CTL entry is added after the CTL Client connects to the CTL Provider service on the security appliance and retrieves the proxy certificate.

Figure 17-4 CTL Client TLS Proxy Features — CTL File Installed on the ASA



Server	File Location	Status
172.23.63.143	/usr/local/cm/tftp/CTLFile.tlv	Passed
172.23.63.144	disk0:/CTLFile.tlv	Passed

You must restart all the Cisco Unified CallManager and TFTP nodes in the Cluster.

The security appliance does not store the raw CTL file in the flash, rather, it parses the CTL file and installs appropriate trustpoints. Figure 17-4 indicates the installation was successful.

Licensing for the TLS Proxy

The TLS proxy for encrypted voice inspection feature supported by the ASA require a Unified Communications Proxy license.

The following table shows the Unified Communications Proxy license details by platform:

Note This feature is not available on No Payload Encryption models.

Model	License Requirement[1]
ASA 5505	Base License and Security Plus License: 2 sessions. *Optional license: 24 sessions.*
ASA 5510	Base License and Security Plus License: 2 sessions. *Optional licenses: 24, 50, or 100 sessions.*
ASA 5520	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, or 1000 sessions.*
ASA 5540	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, or 2000 sessions.*
ASA 5550	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*

Model	License Requirement[1]
ASA 5580	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]
ASA 5512-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, or 500 sessions.*
ASA 5515-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, or 500 sessions.*
ASA 5525-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, or 1000 sessions.*
ASA 5545-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, or 2000 sessions.*
ASA 5555-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*
ASA 5585-X with SSP-10	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*
ASA 5585-X with SSP-20, -40, or -60	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]
ASA SM	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]

[1] The following applications use TLS proxy sessions for their connections. Each TLS proxy session used by these applications (and only these applications) is counted against the UC license limit:
- Phone Proxy
- Presence Federation Proxy
- Encrypted Voice Inspection

Other applications that use TLS proxy sessions do not count towards the UC limit, for example, Mobility Advantage Proxy (which does not require a license) and IME (which requires a separate IME license).

Some UC applications might use multiple sessions for a connection. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS proxy connections, so 2 UC Proxy sessions are used.

You independently set the TLS proxy limit using the **tls-proxy maximum-sessions** command. To view the limits of your model, enter the **tls-proxy maximum-sessions ?** command. When you apply a UC license that is higher than the default TLS proxy limit, the security appliance automatically sets the TLS proxy limit to match the UC limit. The TLS proxy limit takes precedence over the UC license limit; if you set the TLS proxy limit to be less than the UC license, then you cannot use all of the sessions in your UC license.

Note: For license part numbers ending in “K8” (for example, licenses under 250 users), TLS proxy sessions are limited to 1000. For license part numbers ending in “K9” (for example, licenses 250 users or larger), the TLS proxy limit depends on the configuration, up to the model limit. K8 and K9 refer to whether the license is restricted for export: K8 is unrestricted, and K9 is restricted.

Note: If you clear the configuration (using the **clear configure all** command, for example), then the TLS proxy limit is set to the default for your model; if this default is lower than the UC license limit, then you see an error message to use the tls-proxy maximum-sessions command to raise the limit again . If you use failover and enter the **write standby** command on the primary unit to force a configuration synchronization, the **clear configure all** command is generated on the secondary unit automatically, so you may see the warning message on the secondary unit. Because the configuration synchronization restores the TLS proxy limit set on the primary unit, you can ignore the warning.

You might also use SRTP encryption sessions for your connections:
- For K8 licenses, SRTP sessions are limited to 250.
- For K9 licenses, there is not limit.

Note: Only calls that require encryption/decryption for media are counted towards the SRTP limit; if passthrough is set for the call, even if both legs are SRTP, they do not count towards the limit.

2. With the 10,000-session UC license, the total combined sessions can be 10,000, but the maximum number of Phone Proxy sessions is 5000.

Table 17-1 shows the default and maximum TLS session details by platform.

Table 17-1 Default and Maximum TLS Sessions on the Security Appliance

Security Appliance Platform	Default TLS Sessions	Maximum TLS Sessions
ASA 5505	10	80
ASA 5510	100	200
ASA 5520	300	1200
ASA 5540	1000	4500
ASA 5550	2000	4500
ASA 5580	4000	13,000

For more information about licensing, see the general operations configuration guide.

Prerequisites for the TLS Proxy for Encrypted Voice Inspection

Before configuring TLS proxy, the following prerequisites are required:

- You must set clock on the security appliance before configuring TLS proxy. To set the clock manually and display clock, use the **clock set** and **show clock** commands. We recommend that the security appliance use the same NTP server as the Cisco Unified CallManager cluster. TLS handshake may fail due to certificate validation failure if clock is out of sync between the security appliance and the Cisco Unified CallManager server.
- 3DES-AES license is needed to interoperate with the Cisco Unified CallManager. AES is the default cipher used by the Cisco Unified CallManager and Cisco IP Phone.
- Import the following certificates which are stored on the Cisco UCM. These certificates are required by the ASA for the phone proxy.
  - Cisco_Manufacturing_CA
  - CAP-RTP-001
  - CAP-RTP-002
  - CAPF certificate (Optional)

    If LSC provisioning is required or you have LSC enabled IP phones, you must import the CAPF certificate from the Cisco UCM. If the Cisco UCM has more than one CAPF certificate, you must import all of them to the ASA.

  See Chapter 16, “Configuring the Cisco Phone Proxy.”For example, the CA Manufacturer certificate is required by the phone proxy to validate the IP phone certificate.

Configuring the TLS Proxy for Encrypted Voice Inspection

This section includes the following topics:

- Task flow for Configuring the TLS Proxy for Encrypted Voice Inspection, page 17-8

- Creating Trustpoints and Generating Certificates, page 17-9
- Creating an Internal CA, page 17-10
- Creating a CTL Provider Instance, page 17-11
- Creating the TLS Proxy Instance, page 17-12
- Enabling the TLS Proxy Instance for Skinny or SIP Inspection, page 17-13

Task flow for Configuring the TLS Proxy for Encrypted Voice Inspection

To configure the security appliance for TLS proxy, perform the following steps:

Step 1 (Optional) Set the maximum number of TLS proxy sessions to be supported by the security appliance using the following command, for example:

```
ciscoasa(config)# tls-proxy maximum-sessions 1200
```

Note The **tls-proxy maximum-sessions** command controls the memory size reserved for cryptographic applications such as TLS proxy. Crypto memory is reserved at the time of system boot. You may need to reboot the security appliance for the configuration to take effect if the configured maximum sessions number is greater than the currently reserved.

Step 2 Create trustpoints and generate certificates for the TLS Proxy for Encrypted Voice Inspection. See Creating Trustpoints and Generating Certificates, page 17-9.

Step 3 Create the internal CA to sign the LDC for Cisco IP Phones. See Creating an Internal CA, page 17-10.

Step 4 Create the CTL provider instance. See Creating a CTL Provider Instance, page 17-11.

Step 5 Create the TLS proxy instance. See Creating the TLS Proxy Instance, page 17-12.

Step 6 Enable the TLS proxy y with SIP and Skinny inspection. See Enabling the TLS Proxy Instance for Skinny or SIP Inspection, page 17-13.

Step 7 Export the local CA certificate (ldc_server) and install it as a trusted certificate on the Cisco UCM server.

a. Use the following command to export the certificate if a trust-point with **proxy-ldc-issuer** is used as the signer of the dynamic certificates, for example:

```
ciscoasa(config)# crypto ca export ldc_server identity-certificate
```

b. For the embedded local CA server LOCAL-CA-SERVER, use the following command to export its certificate, for example:

```
ciscoasa(config)# show crypto ca server certificate
```

Save the output to a file and import the certificate on the Cisco UCM. For more information, see the Cisco Unified CallManager document:
http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_0/iptp_adm/504/iptpch6.htm#wp1040848

After this step, you may use the Display Certificates function on the Cisco Unified CallManager GUI to verify the installed certificate:

http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_0/iptp_adm/504/iptpch6.htm#wp1040354

Step 8 Run the CTL Client application to add the server proxy certificate (ccm_proxy) to the CTL file and install the CTL file on the security appliance. See the Cisco Unified CallManager document for information on how to configure and use CTL Client:

http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_1/nci/p08/secuauth.htm

Note You will need the CTL Client that is released with Cisco Unified CallManager Release 5.1 to interoperate with the security appliance. See the "CTL Client Overview" section on page 17-3 for more information regarding TLS proxy support.

Creating Trustpoints and Generating Certificates

The Cisco UCM proxy certificate could be self-signed or issued by a third-party CA. The certificate is exported to the CTL client.

Prerequisites

Import the required certificates, which are stored on the Cisco UCM. See the "Certificates from the Cisco UCM" section on page 16-7 and the "Importing Certificates from the Cisco UCM" section on page 16-15.

	Command	Purpose
Step 1	hostname(config)# **crypto key generate rsa label** *key-pair-label* **modulus** *size* **Examples:** ciscoasa(config)# **crypto key generate rsa label ccm_proxy_key modulus 1024** ciscoasa(config)# **crypto key generate rsa label ldc_signer_key modulus 1024** ciscoasa(config)# **crypto key generate rsa label phone_common modulus 1024**	Creates the RSA keypair that can be used for the trustpoints. The keypair is used by the self-signed certificate presented to the local domain containing the Cisco UP (proxy for the remote entity). **Note** We recommend that you create a different key pair for each role.
Step 2	ciscoasa(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** ciscoasa(config)# **! for self-signed CCM proxy certificate** ciscoasa(config)# **crypto ca trustpoint ccm_proxy**	Enters the trustpoint configuration mode for the specified trustpoint so that you can create the trustpoint for the Cisco UMA server. A trustpoint represents a CA identity and possibly a device identity, based on a certificate issued by the CA.
Step 3	ciscoasa(config-ca-trustpoint)# **enrollment self**	Generates a self-signed certificate.
Step 4	ciscoasa(config-ca-trustpoint)# **fqdn none**	Specifies not to include a fully qualified domain name (FQDN) in the Subject Alternative Name extension of the certificate during enrollment.

	Command	Purpose

Step 5	`ciscoasa(config-ca-trustpoint)#` **`subject-name`** *`X.500_name`*
	`Example:`
	`ciscoasa(config-ca-trustpoint)#` **`subject-name cn=EJW-SV-1-Proxy`**
		Includes the indicated subject DN in the certificate during enrollment
		Cisco IP Phones require certain fields from the X.509v3 certificate to be present to validate the certificate via consulting the CTL file. Consequently, the **subject-name** entry must be configured for a proxy certificate trustpoint. The subject name must be composed of the ordered concatenation of the CN, OU and O fields. The CN field is mandatory; the others are optional.
		Note Each of the concatenated fields (when present) are separated by a semicolon, yielding one of the following forms:
		CN=xxx;OU=yyy;O=zzz
		CN=xxx;OU=yyy
		CN=xxx;O=zzz
		CN=xxx

Step 6	`hostname(config-ca-trustpoint)#` **`keypair`** *`keyname`*
	`Example:`
	`ciscoasa(config-ca-trustpoint)#` **`keypair ccm_proxy_key`**
		Specifies the key pair whose public key is to be certified.

Step 7	`hostname(config-ca-trustpoint)#` **`exit`**	Exits from the CA Trustpoint configuration mode.

Step 8	`hostname(config)#` **`crypto ca enroll`** *`trustpoint`*
	`Example:`
	`ciscoasa(config)#` **`crypto ca enroll ccm_proxy`**
		Starts the enrollment process with the CA and specifies the name of the trustpoint to enroll with.

What to Do Next

Once you have created the trustpoints and generated the certificates, create the internal CA to sign the LDC for Cisco IP Phones. See Creating an Internal CA, page 17-10.

Creating an Internal CA

Create an internal local CA to sign the LDC for Cisco IP Phones.

This local CA is created as a regular self-signed trustpoint with **proxy-ldc-issuer** enabled. You can use the embedded local CA LOCAL-CA-SERVER on the ASA to issue the LDC.

	Command	Purpose

Step 1	`ciscoasa(config)#` **`crypto ca trustpoint`** *`trustpoint_name`*
	`Example:`
	`ciscoasa(config)#` **`! for the internal local LDC issuer`**
	`ciscoasa(config)#` **`crypto ca trustpoint ldc_server`**
		Enters the trustpoint configuration mode for the specified trustpoint so that you can create the trustpoint for the LDC issuer.

Step 2	`ciscoasa(config-ca-trustpoint)#` **`enrollment self`**	Generates a self-signed certificate.

	Command	Purpose
Step 3	ciscoasa(config-ca-trustpoint)# **proxy-ldc-issue**r	Issues TLS proxy local dynamic certificates. The **proxy-ldc-issuer** command grants a crypto trustpoint the role as local CA to issue the LDC and can be accessed from crypto ca trustpoint configuration mode. The **proxy-ldc-issuer** command defines the local CA role for the trustpoint to issue dynamic certificates for TLS proxy. This command can only be configured under a trustpoint with "enrollment self."
Step 4	ciscoasa(config-ca-trustpoint)# **fqdn** *fqdn* **Example:** ciscoasa(config-ca-trustpoint)# **fqdn my-ldc-ca.exmaple.com**	Includes the indicated FQDN in the Subject Alternative Name extension of the certificate during enrollment.
Step 5	ciscoasa(config-ca-trustpoint)# **subject-name** *X.500_name* **Example:** ciscoasa(config-ca-trustpoint)# **subject-name cn=FW_LDC_SIGNER_172_23_45_200**	Includes the indicated subject DN in the certificate during enrollment
Step 6	hostname(config-ca-trustpoint)# **keypair** *keyname* **Example:** ciscoasa(config-ca-trustpoint)# **keypair ldc_signer_key**	Specifies the key pair whose public key is to be certified.
Step 7	ciscoasa(config-ca-trustpoint)# **exit**	Exits from the CA Trustpoint configuration mode.
Step 8	hostname(config)# **crypto ca enroll** *trustpoint* **Example:** ciscoasa(config)# **crypto ca enroll ldc_server**	Starts the enrollment process with the CA and specifies the name of the trustpoint to enroll with.

What to Do Next

Once you have created the internal CA, create the CTL provider instance. See Creating a CTL Provider Instance, page 17-11.

Creating a CTL Provider Instance

Create a CTL Provider instance in preparation for a connection from the CTL Client.

The default port number listened by the CTL Provider is TCP 2444, which is the default CTL port on the Cisco UCM. Use the **service port** command to change the port number if a different port is used by the Cisco UCM cluster.

	Command	Purpose
Step 1	ciscoasa(config)# **ctl-provider** *ctl_name* **Example:** ciscoasa(config)# **ctl-provider my_ctl**	Enters the CTL provider configuration mode so that you can create the Certificate Trust List provider instance.
Step 2	ciscoasa(config-ctl-provider)# **client interface** *if_name ipv4_addr* **Example:** ciscoasa(config-ctl-provider)# **client interface inside address 172.23.45.1**	Specifies clients allowed to connect to the Certificate Trust List provider. Where **interface** *if_name* specifies the interface allowed to connect and *ipv4_addr* specifies the IP address of the client. More than one command may be issued to define multiple clients.
Step 3	ciscoasa(config-ctl-provider)# **client username** *user_name* **password** *password* **encrypted** **Example:** ciscoasa(config-ctl-provider)# **client username CCMAdministrator password XXXXXX encrypted**	Specifies the username and password for client authentication. The username and password must match the username and password for Cisco UCM administration.
Step 4	ciscoasa(config-ctl-provider)# **export certificate** *trustpoint_name* **Example:** ciscoasa(config-ctl-provider)# **export certificate**	Specifies the certificate to be exported to the client. The certificate will be added to the Certificate Trust List file composed by the CTL client. The trustpoint name in the **export** command is the proxy certificate for the Cisco UCM server.
Step 5	ciscoasa(config-ctl-provider)# **ctl install**	Enables the CTL provider to parse the CTL file from the CTL client and install trustpoints for entries from the CTL file. Ttrustpoints installed by this command have names prefixed with "_internal_CTL_<ctl_name>."

What to Do Next

Once you have created the CTL provider instance, create the TLS proxy instance. See Creating the TLS Proxy Instance, page 17-12.

Creating the TLS Proxy Instance

Create the TLS proxy instance to handle the encrypted signaling.

	Command	Purpose

Step 1	`ciscoasa(config)#` **`tls-proxy`** *`proxy_name`*
	`Example:`
	`ciscoasa(config)# tls-proxy my_proxy`
		Creates the TLS proxy instance.

Step 2	`ciscoasa(config-tlsp)#` **`server trust-point`** *`proxy_trustpoint`*
	`Example:`
	`ciscoasa(config-tlsp)# server trust-point ccm_proxy`
		Specifies the proxy trustpoint certificate to present during TLS handshake.
		The **server** command configures the proxy parameters for the original TLS server. In other words, the parameters for the ASA to act as the server during a TLS handshake, or facing the original TLS client.

Step 3	`ciscoasa(config-tlsp)#` **`client ldc issuer`** *`ca_tp_name`*
	`Example:`
	`ciscoasa(config-tlsp)# client ldc issuer ldc_server`
		Sets the local dynamic certificate issuer. The local CA to issue client dynamic certificates is defined by the **crypto ca trustpoint** command and the trustpoint must have **proxy-ldc-issuer** configured, or the default local CA server (LOCAL-CA-SERVER).
		Where **`ldc issuer`** *`ca_tp_name`* specifies the local CA trustpoint to issue client dynamic certificates.

Step 4	`ciscoasa(config-tlsp)#` **`client ldc key-pair`** *`key_label`*
	`Example:`
	`ciscoasa(config-tlsp)# client ldc key-pair phone_common`
		Sets the keypair.
		The keypair value must have been generated with the crypto key generate command.

Step 5	`hostname(config-tlsp)#` **`client cipher-suite`** *`cipher_suite`*
	`Example:`
	`hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1`
		Sets the user-defined cipher suite.
		For client proxy (the proxy acts as a TLS client to the server), the user-defined cipher suite replaces the default cipher suite, or the one defined by the ssl encryption command. You can use this command to achieve difference ciphers between the two TLS sessions. You should use AES ciphers with the CallManager server.

What to Do Next

Once you have created TLS proxy instance, enable the TLS proxy instance for Skinny and SIP inspection. See Enabling the TLS Proxy Instance for Skinny or SIP Inspection, page 17-13.

Enabling the TLS Proxy Instance for Skinny or SIP Inspection

Enable TLS proxy for the Cisco IP Phones and Cisco UCMs in Skinny or SIP inspection. The following procedure shows how to enable the TLS proxy instance for Skinny inspection.

	Command	Purpose

Step 1	`hostname(config)#` **`class-map`** *`class_map_name`*
	`Example:`
	`ciscoasa(config)#` **`class-map sec_skinny`**
		Configures the secure Skinny class of traffic to inspect.
		Where *class_map_name* is the name of the Skinny class map.

Step 2	`ciscoasa(config-cmap)#` **`match port tcp eq 2443`**	Matches the TCP port 2443 to which you want to apply actions for secure Skinny inspection
Step 3	`ciscoasa(config-cmap)#` **`exit`**	

Step 4	`hostname(config)#` **`policy-map type inspect skinny`** *`policy_map_name`*
	`Example:`
	`ciscoasa(config)#` **`policy-map type inspect skinny skinny_inspect`**
		Defines special actions for Skinny inspection application traffic.

Step 5	`ciscoasa(config-pmap)#` **`parameters`**
	`ciscoasa(config-pmap-p)#` **`!`** `Skinny inspection parameters`
		Specifies the parameters for Skinny inspection. Parameters affect the behavior of the inspection engine.
		The commands available in parameters configuration mode depend on the application.

Step 6	ciscoasa(config-pmap-p)# **exit**	Exits from Policy Map configuration mode.

Step 7	`hostname(config)#` **`policy-map`** *`name`*
	`Example:`
	`ciscoasa(config)#` **`policy-map global_policy`**
		Configure the policy map and attach the action to the class of traffic.

Step 8	`ciscoasa(config-pmap)#` **`class inspection_default`**	Specifies the default class map. The configuration includes a default Layer 3/4 class map that the ASA uses in the default global policy. It is called **inspection_default** and matches the default inspection traffic,

Step 9	`ciscoasa(config-pmap-c)#` **`inspect skinny`** *`skinny_map`*
	`Example:`
	`ciscoasa(config-pmap-c)#` **`inspect skinny skinny_inspect`**
		Enables SCCP (Skinny) application inspection.

Step 10	`ciscoasa(config-pmap)#` **`class`** *`classmap_name`*
	`Example:`
	`ciscoasa(config-pmap)#` **`class sec_skinny`**
		Assigns a class map to the policy map where you can assign actions to the class map traffic.

Step 11	`ciscoasa(config-pmap-c)#` **`inspect skinny`** *`skinny_map`* **`tls-proxy`** *`proxy_name`*
	`Example:`
	`ciscoasa(config-pmap-c)#` **`inspect skinny skinny_inspect tls-proxy my_proxy`**
		Enables TLS proxy for the specified inspection session.

Step 12	`ciscoasa(config-pmap-c)#` **`exit`**	Exits from the Policy Map configuration mode.

Step 13	`ciscoasa(config)#` **`service-policy`** *`policymap_name`* **`global`**
	`Example:`
	`ciscoasa(config)#` **`service-policy global_policy global`**
		Enables the service policy on all interfaces.

Monitoring the TLS Proxy

You can enable TLS proxy debug flags along with SSL syslogs to debug TLS proxy connection problems. For example, using the following commands to enable TLS proxy-related debug and syslog output only:

```
ciscoasa(config)# debug inspect tls-proxy events
ciscoasa(config)# debug inspect tls-proxy errors
ciscoasa(config)# logging enable
ciscoasa(config)# logging timestamp
ciscoasa(config)# logging list loglist message 711001
ciscoasa(config)# logging list loglist message 725001-725014
ciscoasa(config)# logging list loglist message 717001-717038
ciscoasa(config)# logging buffer-size 1000000
ciscoasa(config)# logging buffered loglist
ciscoasa(config)# logging debug-trace
```

The following is sample output reflecting a successful TLS proxy session setup for a SIP phone:

```
ciscoasa(config)# show log

Apr 17 2007 23:13:47: %ASA-6-725001: Starting SSL handshake with client
outside:133.9.0.218/49159 for TLSv1 session.
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Set up proxy for Client
outside:133.9.0.218/49159 <-> Server inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Using trust point 'local_ccm' with the
Client, RT proxy cbae1538
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Waiting for SSL handshake from Client
outside:133.9.0.218/49159.
Apr 17 2007 23:13:47: %ASA-7-725010: Device supports the following 4 cipher(s).
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : RC4-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[3] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[4] : DES-CBC3-SHA
Apr 17 2007 23:13:47: %ASA-7-725008: SSL client outside:133.9.0.218/49159 proposes the
following 2 cipher(s).
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725012: Device chooses cipher : AES128-SHA for the SSL
session with client outside:133.9.0.218/49159
Apr 17 2007 23:13:47: %ASA-7-725014: SSL lib error. Function: SSL23_READ Reason: ssl
handshake failure
Apr 17 2007 23:13:47: %ASA-7-717025: Validating certificate chain containing 1
certificate(s).
Apr 17 2007 23:13:47: %ASA-7-717029: Identified client certificate within certificate
chain. serial number: 01, subject name: cn=SEP0017593F50A8.
Apr 17 2007 23:13:47: %ASA-7-717030: Found a suitable trustpoint
_internal_ejw-sv-2_cn=CAPF-08a91c01 to validate certificate.
Apr 17 2007 23:13:47: %ASA-6-717022: Certificate was successfully validated. serial
number: 01, subject name:  cn=SEP0017593F50A8.
Apr 17 2007 23:13:47: %ASA-6-717028: Certificate chain was successfully validated with
warning, revocation status was not checked.
Apr 17 2007 23:13:47: %ASA-6-725002: Device completed SSL handshake with client
outside:133.9.0.218/49159
Apr 17 2007 23:13:47: %ASA-6-725001: Starting SSL handshake with server
inside:195.168.2.201/5061 for TLSv1 session.
Apr 17 2007 23:13:47: %ASA-7-725009: Device proposes the following 2 cipher(s) to server
inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Generating LDC for client
'cn=SEP0017593F50A8', key-pair 'phone_common', issuer 'LOCAL-CA-SERVER', RT proxy cbae1538
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Started SSL handshake with Server
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Data channel ready for the Client
Apr 17 2007 23:13:47: %ASA-7-725013: SSL Server inside:195.168.2.201/5061 choose cipher :
AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-717025: Validating certificate chain containing 1
certificate(s).
Apr 17 2007 23:13:47: %ASA-7-717029: Identified client certificate within certificate
chain. serial number: 76022D3D9314743A, subject name: cn=EJW-SV-2.inside.com.
Apr 17 2007 23:13:47: %ASA-6-717022: Certificate was successfully validated. Certificate
is resident and trusted, serial number: 76022D3D9314743A, subject name:
cn=EJW-SV-2.inside.com.
Apr 17 2007 23:13:47: %ASA-6-717028: Certificate chain was successfully validated with
revocation status check.
Apr 17 2007 23:13:47: %ASA-6-725002: Device completed SSL handshake with server
inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Data channel ready for the Server
```

Use the **show tls-proxy** commands with different options to check the active TLS proxy sessions. The following are some sample outputs:

```
ciscoasa(config-tlsp)# show tls-proxy
Maximum number of sessions: 1200

TLS-Proxy 'sip_proxy': ref_cnt 1, seq# 3
    Server proxy:
        Trust-point: local_ccm
    Client proxy:
        Local dynamic certificate issuer: LOCAL-CA-SERVER
        Local dynamic certificate key-pair: phone_common
        Cipher suite:  aes128-sha1 aes256-sha1
    Run-time proxies:
        Proxy 0xcbae1538: Class-map: sip_ssl, Inspect: sip
            Active sess 1, most sess 3, byte 3456043

TLS-Proxy 'proxy': ref_cnt 1, seq# 1
    Server proxy:
        Trust-point: local_ccm
    Client proxy:
        Local dynamic certificate issuer: ldc_signer
        Local dynamic certificate key-pair: phone_common
        Cipher-suite: <unconfigured>
    Run-time proxies:
        Proxy 0xcbadf720: Class-map: skinny_ssl, Inspect: skinny
            Active sess 1, most sess 1, byte 42916

ciscoasa(config-tlsp)# show tls-proxy session count
2 in use, 4 most used

ciscoasa(config-tlsp)# show tls-proxy session
2 in use, 4 most used
outside 133.9.0.211:50437 inside 195.168.2.200:2443 P:0xcbadf720(proxy) S:0xcbc48a08 byte
42940
outside 133.9.0.218:49159 inside 195.168.2.201:5061 P:0xcbae1538(sip_proxy) S:0xcbad5120
byte 8786

ciscoasa(config-tlsp)# show tls-proxy session detail
2 in use, 4 most used
outside 133.9.0.211:50437 inside 195.168.2.200:2443 P:0xcbadf720(proxy) S:0xcbc48a08 byte
42940
    Client: State SSLOK  Cipher AES128-SHA Ch 0xca55e498 TxQSize 0 LastTxLeft 0 Flags 0x1
    Server: State SSLOK  Cipher AES128-SHA Ch 0xca55e478 TxQSize 0 LastTxLeft 0 Flags 0x9
Local Dynamic Certificate
    Status: Available
    Certificate Serial Number: 29
    Certificate Usage: General Purpose
    Public Key Type: RSA (1024 bits)
    Issuer Name:
        cn=TLS-Proxy-Signer
    Subject Name:
        cn=SEP0002B9EB0AAD
        o=Cisco Systems Inc
        c=US
    Validity Date:
        start date: 09:25:41 PDT Apr 16 2007
        end   date: 09:25:41 PDT Apr 15 2008
    Associated Trustpoints:

outside 133.9.0.218:49159 inside 195.168.2.201:5061 P:0xcbae1538(sip_proxy) S:0xcbad5120
byte 8786
    Client: State SSLOK  Cipher AES128-SHA Ch 0xca55e398 TxQSize 0 LastTxLeft 0 Flags 0x1
    Server: State SSLOK  Cipher AES128-SHA Ch 0xca55e378 TxQSize 0 LastTxLeft 0 Flags 0x9
Local Dynamic Certificate
    Status: Available
    Certificate Serial Number: 2b
    Certificate Usage: General Purpose
    Public Key Type: RSA (1024 bits)
    Issuer Name:
        cn=F1-ASA.default.domain.invalid
    Subject Name:
        cn=SEP0017593F50A8
    Validity Date:
        start date: 23:13:47 PDT Apr 16 2007
        end   date: 23:13:47 PDT Apr 15 2008
    Associated Trustpoints:
```

Feature History for the TLS Proxy for Encrypted Voice Inspection

Table 17-2 lists the release history for this feature.

Table 17-2 Feature History for Cisco Phone Proxy

Feature Name	Releases	Feature Information
TLS Proxy	8.0(2)	The TLS proxy feature was introduced.

CHAPTER 18

Configuring Cisco Mobility Advantage

This chapter describes how to configure the ASA for Cisco Unified Communications Mobility Advantage Proxy features.

This chapter includes the following sections:

Information about the Cisco Mobility Advantage Proxy Feature

This section contains the following topics:

Cisco Mobility Advantage Proxy Functionality

To support Cisco UMA for the Cisco Mobility Advantage solution, the mobility advantage proxy (implemented as a TLS proxy) includes the following functionality:

- The ability to allow no client authentication during the handshake with clients.
- Allowing an imported PKCS-12 certificate to server as a proxy certificate.

The ASA includes an inspection engine to validate the Cisco UMA Mobile Multiplexing Protocol (MMP).

MMP is a data transport protocol for transmitting data entities between Cisco UMA clients and servers. MMP must be run on top of a connection-oriented protocol (the underlying transport) and is intended to be run on top of a secure transport protocol such as TLS. The Orative Markup Language (OML) protocol is intended to be run on top of MMP for the purposes of data synchronization, as well as the HTTP protocol for uploading and downloading large files.

The TCP/TLS default port is 5443. There are no embedded NAT or secondary connections.

Cisco UMA client and server communications can be proxied via TLS, which decrypts the data, passes it to the inspect MMP module, and re-encrypt the data before forwarding it to the endpoint. The inspect MMP module verifies the integrity of the MMP headers and passes the OML/HTTP to an appropriate handler. The ASA takes the following actions on the MMP headers and data:

- Verifies that client MMP headers are well-formed. Upon detection of a malformed header, the TCP session is terminated.
- Verifies that client to server MMP header lengths are not exceeded. If an MMP header length is exceeded (4096), then the TCP session is terminated.
- Verifies that client to server MMP content lengths are not exceeded. If an entity content length is exceeded (4096), the TCP session is terminated.

Note 4096 is the value currently used in MMP implementations.

Because MMP headers and entities can be split across packets, the ASA buffers data to ensure consistent inspection. The SAPI (stream API) handles data buffering for pending inspection opportunities. MMP header text is treated as case insensitive and a space is present between header text and values. Reclaiming of MMP state is performed by monitoring the state of the TCP connection.

Mobility Advantage Proxy Deployment Scenarios

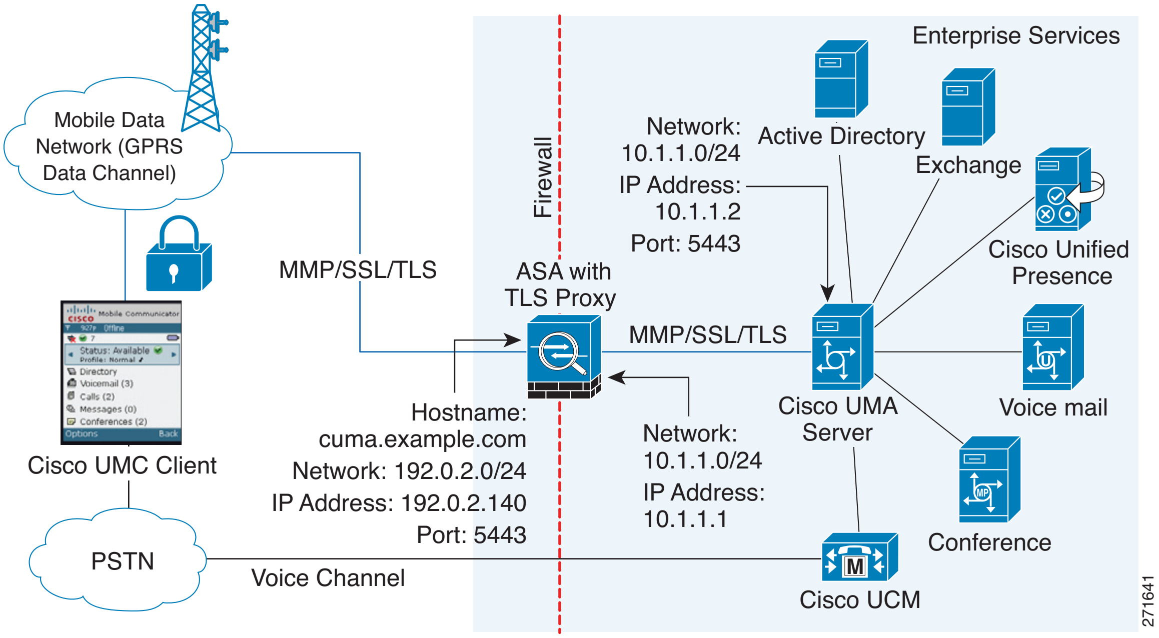
Figure 18-1 and Figure 18-2 show the two deployment scenarios for the TLS proxy used by the Cisco Mobility Advantage solution. In scenario 1 (the recommended deployment architecture), the ASA functions as both the firewall and TLS proxy. In scenario 2, the ASA functions as the TLS proxy only and works with an existing firewall. In both scenarios, the clients connect from the Internet.

In the scenario 1 deployment, the ASA is between a Cisco UMA client and a Cisco UMA server. The Cisco UMA client is an executable that is downloaded to each smartphone. The Cisco UMA client applications establishes a data connection, which is a TLS connection, to the corporate Cisco UMA server. The ASA intercepts the connections and inspects the data that the client sends to the Cisco UMA server.

Note The TLS proxy for the Cisco Mobility Advantage solution does not support client authentication because the Cisco UMA client cannot present a certificate. The following commands can be used to disable authentication during the TLS handshake.

```
hostname(config)# tls-proxy my_proxy
hostname(config-tlsp)# no server authenticate-client
```

Figure 18-1 Security Appliance as Firewall with Mobility Advantage Proxy and MMP Inspection



In Figure 18-1, the ASA performs static NAT by translating the Cisco UMA server 10.1.1.2 IP address to 192.0.2.140.

Figure 18-2 shows deployment scenario 2, where the ASA functions as the TLS proxy only and does not function as the corporate firewall. In this scenario, the ASA and the corporate firewall are performing NAT. The corporate firewall will not be able to predict which client from the Internet needs to connect to the corporate Cisco UMA server. Therefore, to support this deployment, you can take the following actions:

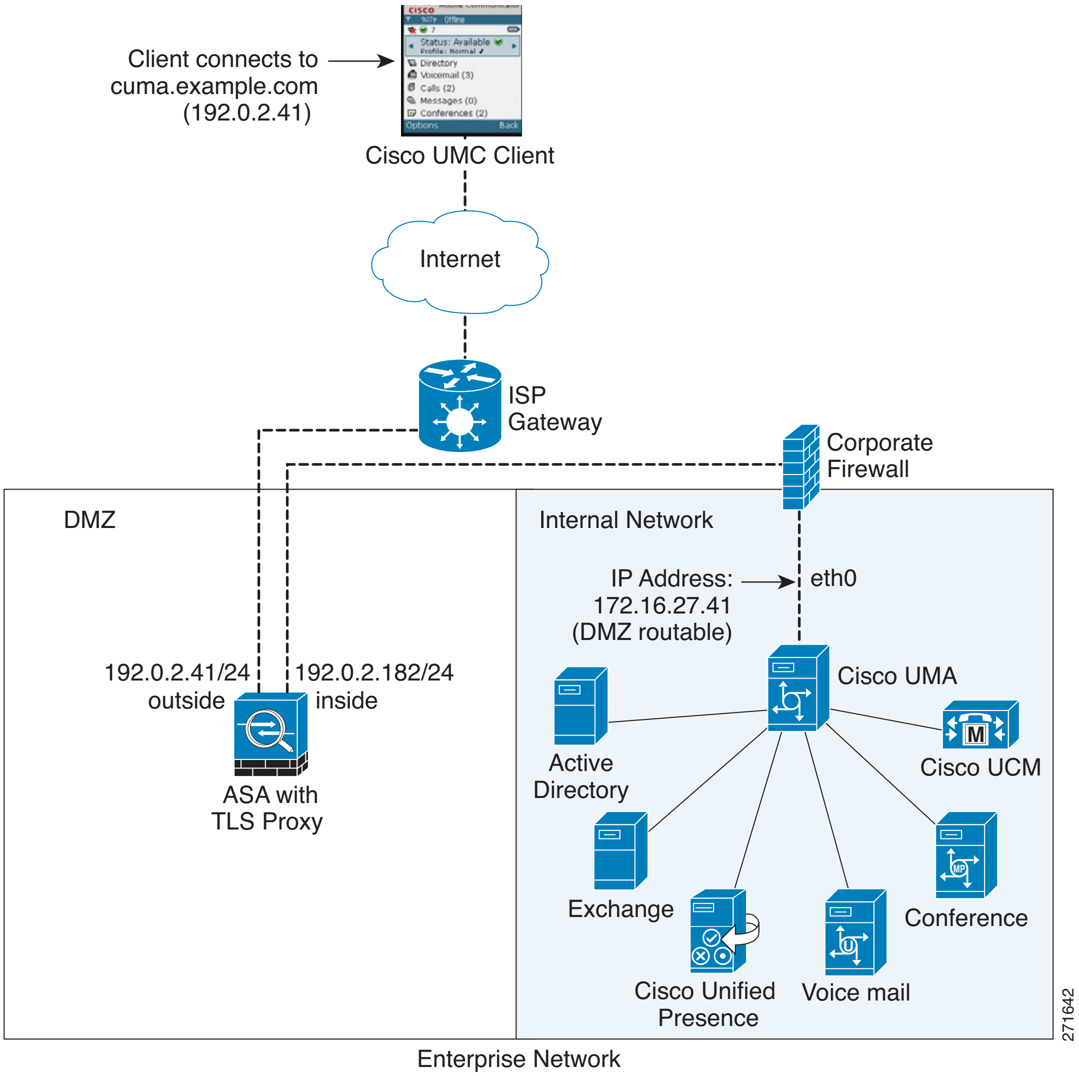
- Set up a NAT rule for inbound traffic that translates the destination IP address 192.0.2.41 to 172.16.27.41.
- Set up an interface PAT rule for inbound traffic translating the source IP address of every packet so that the corporate firewall does not need to open up a wildcard pinhole. The Cisco UMA server receives packets with the source IP address 192.0.12.183.

```
hostname(config)# object network obj-0.0.0.0-01
hostname(config-network-object)# subnet 0.0.0.0 0.0.0.0
hostname(config-network-object)# nat (outside,inside) dynamic 192.0.2.183
```

See Chapter 4, “Configuring Network Object NAT” and Chapter 5, “Configuring Twice NAT” for information.

Note This interface PAT rule converges the Cisco UMA client IP addresses on the outside interface of the ASA into a single IP address on the inside interface by using different source ports. Performing this action is often referred as “outside PAT”. “Outside PAT” is not recommended when TLS proxy for Cisco Mobility Advantage is enabled on the same interface of the ASA with phone proxy, Cisco Unified Presence, or any other features involving application inspection. “Outside PAT” is not supported completely by application inspection when embedded address translation is needed.

Figure 18-2 Cisco UMC/Cisco UMA Architecture – Scenario 2: Security Appliance as Mobility Advantage Proxy Only



Mobility Advantage Proxy Using NAT/PAT

In both scenarios (Figure 18-1 and Figure 18-2), NAT can be used to hide the private address of the Cisco UMA servers.

In scenario 2 (Figure 18-2), PAT can be used to converge all client traffic into one source IP, so that the firewall does not have to open up a wildcard pinhole for inbound traffic.

```
hostname(config)# access-list cumc extended permit tcp any host 172.16.27.41 eq 5443
```

versus

```
hostname(config)# access-list cumc extended permit tcp host 192.0.2.183 host 172.16.27.41
eq 5443
```

Trust Relationships for Cisco UMA Deployments

To establish a trust relationship between the Cisco UMC client and the ASA, the ASA uses the Cisco UMA server certificate and keypair or the ASA obtains a certificate with the Cisco UMA server FQDN (certificate impersonation). Between the ASA and the Cisco UMA server, the ASA and Cisco UMA server use self-signed certificates or certificates issued by a local certificate authority.

Figure 18-3 shows how you can import the Cisco UMA server certificate onto the ASA. When the Cisco UMA server has already enrolled with a third-party CA, you can import the certificate with the private key onto the ASA. Then, the ASA has the full credentials of the Cisco UMA server. When a Cisco UMA client connects to the Cisco UMA server, the ASA intercepts the handshake and uses the Cisco UMA server certificate to perform the handshake with the client. The ASA also performs a handshake with the server.

Figure 18-3 How the Security Appliance Represents Cisco UMA – Private Key Sharing

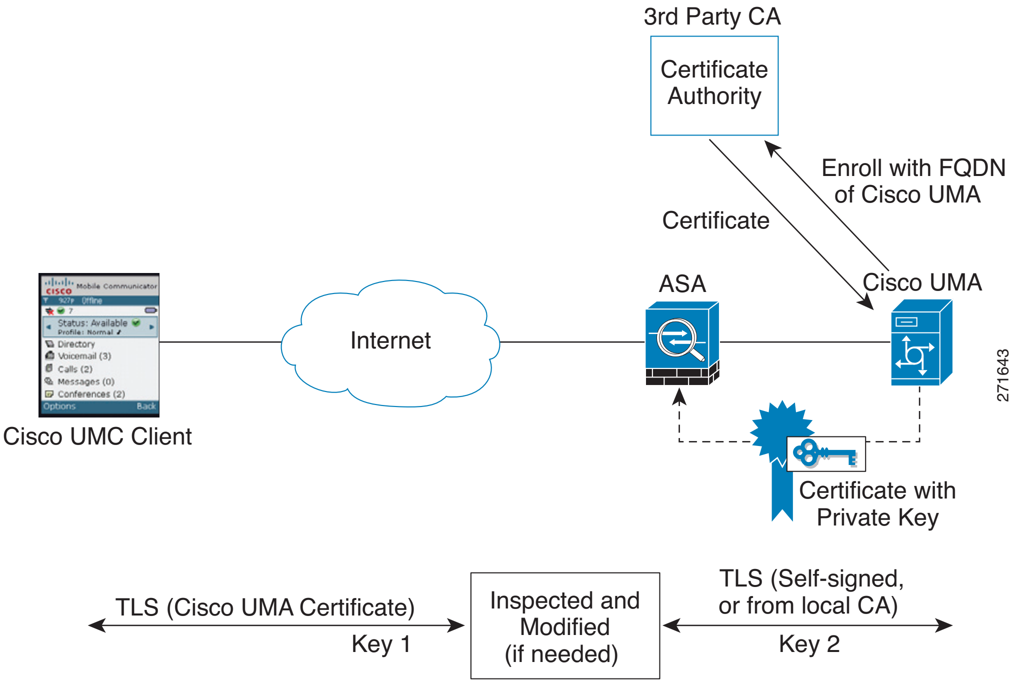
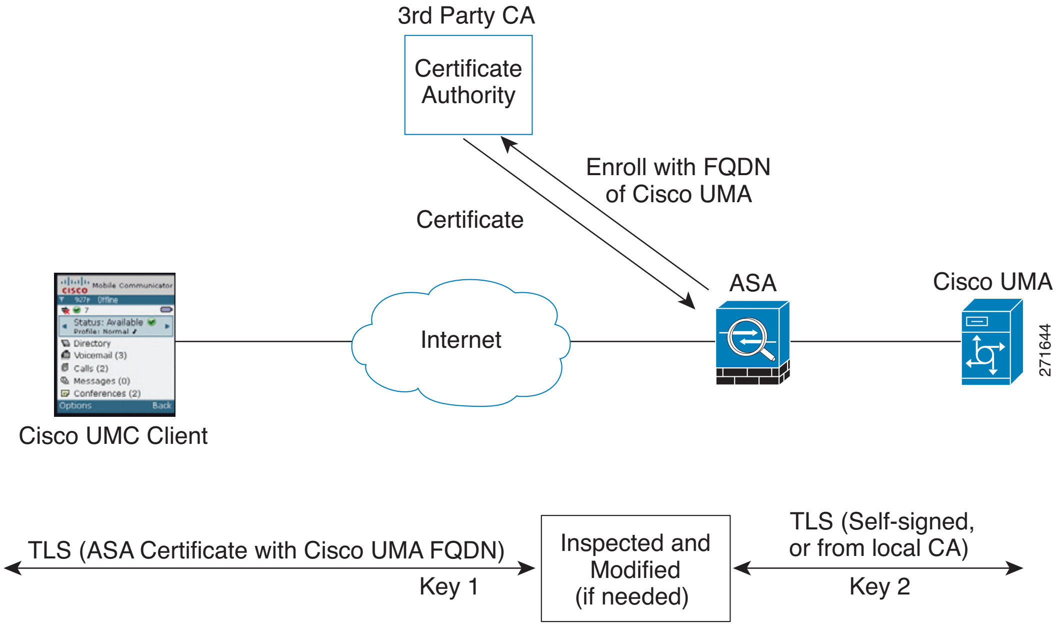

3rd Party CA
Certificate Authority
Enroll with FQDN of Cisco UMA
Certificate
ASA
Cisco UMA
Internet
Cisco UMC Client
Certificate with Private Key
271643
TLS (Cisco UMA Certificate)
Key 1
Inspected and Modified (if needed)
TLS (Self-signed, or from local CA)
Key 2

Figure 18-4 shows another way to establish the trust relationship. Figure 18-4 shows a green field deployment, because each component of the deployment has been newly installed. The ASA enrolls with the third-party CA by using the Cisco UMA server FQDN as if the ASA is the Cisco UMA server. When the Cisco UMA client connects to the ASA, the ASA presents the certificate that has the Cisco UMA server FQDN. The Cisco UMA client believes it is communicating to with the Cisco UMA server.

Figure 18-4 How the Security Appliance Represents Cisco UMA – Certificate Impersonation


3rd Party CA
Certificate Authority
Enroll with FQDN of Cisco UMA
Certificate
ASA
Cisco UMA
Internet
Cisco UMC Client
271644
TLS (ASA Certificate with Cisco UMA FQDN)
Key 1
Inspected and Modified (if needed)
TLS (Self-signed, or from local CA)
Key 2

A trusted relationship between the ASA and the Cisco UMA server can be established with self-signed certificates. The ASA's identity certificate is exported, and then uploaded on the Cisco UMA server truststore. The Cisco UMA server certificate is downloaded, and then uploaded on the ASA truststore by creating a trustpoint and using the **crypto ca authenticate** command.

Licensing for the Cisco Mobility Advantage Proxy Feature

The Cisco Unified Communications proxy features (Cisco Phone Proxy, TLS proxy for encrypted voice inspection, and the Cisco Presence Federation Proxy) supported by the ASA require a Unified Communications Proxy license. However, in Version 8.2(2) and later, the Mobility Advantage proxy no longer requires a Unified Communications Proxy license.

The following table shows the licensing requirements for the Mobility Advantage proxy:

Model	License Requirement
All models	Base License.

For more information about licensing, see the general operations configuration guide.

Configuring Cisco Mobility Advantage

This section includes the following topics:

- Task Flow for Configuring Cisco Mobility Advantage, page 18-7
- Installing the Cisco UMA Server Certificate, page 18-7
- Creating the TLS Proxy Instance, page 18-8

- Enabling the TLS Proxy for MMP Inspection, page 18-9

Task Flow for Configuring Cisco Mobility Advantage

To configure for the ASA to perform TLS proxy and MMP inspection as shown in Figure 18-1 and Figure 18-2, perform the following tasks.

It is assumed that self-signed certificates are used between the ASA and the Cisco UMA server.

Prerequisites

Export the Cisco UMA server certificate and keypair in PKCS-12 format so that you can import it onto the ASA. The certificate will be used during the handshake with the Cisco UMA clients.

Step 1 Create the static NAT for the Cisco UMA server by entering the following commands:

```
hostname(config)# object network name
hostname(config-network-object)# host real_ip
hostname(config-network-object)# nat (real_ifc,mapped_ifc) static mapped_ip
```

Step 2 Import the Cisco UMA server certificate onto the ASA by entering the following commands:

```
hostname(config)# crypto ca import trustpoint pkcs12 passphrase
[paste base 64 encoded pkcs12]
hostname(config)# quit
```

Step 3 Install the Cisco UMA server certificate on the ASA. See Installing the Cisco UMA Server Certificate, page 18-7.

Step 4 Create the TLS proxy instance for the Cisco UMA clients connecting to the Cisco UMA server. See Creating the TLS Proxy Instance, page 18-8.

Step 5 Enable the TLS proxy for MMP inspection. See Enabling the TLS Proxy for MMP Inspection, page 18-9.

Installing the Cisco UMA Server Certificate

Install the Cisco UMA server self-signed certificate in the ASA truststore. This task is necessary for the ASA to authenticate the Cisco UMA server during the handshake between the ASA proxy and Cisco UMA server.

Prerequisites

Export the Cisco UMA server certificate and keypair in PKCS-12 format so that you can import it onto the ASA.

	Command	Purpose
Step 1	hostname(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** hostname(config)# crypto ca trustpoint cuma_server	Enters the trustpoint configuration mode for the specified trustpoint so that you can create the trustpoint for the Cisco UMA server. A trustpoint represents a CA identity and possibly a device identity, based on a certificate issued by the CA.
Step 2	hostname(config-ca-trustpoint)# **enrollment terminal**	Specifies cut and paste enrollment with this trustpoint (also known as manual enrollment).
Step 3	hostname(config-ca-trustpoint)# **exit**	Exits from the CA Trustpoint configuration mode.
Step 4	hostname(config)# **crypto ca authenticate** *trustpoint* **Example:** hostname(config)# crypto ca authenticate cuma_server Enter the base 64 encoded CA certificate. End with a blank line or the word "quit" on a line by itself [certificate data omitted] Certificate has the following attributes: Fingerprint: 21B598D5 4A81F3E5 0B24D12E 3F89C2E4 % Do you accept this certificate? [yes/no]: yes Trustpoint CA certificate accepted. % Certificate successfully imported hostname(config)#	Installs and authenticates the CA certificates associated with a trustpoint created for the Cisco UMA server. Where *trustpoint* specifies the trustpoint from which to obtain the CA certificate. Maximum name length is 128 characters. The ASA prompts you to paste the base-64 formatted CA certificate onto the terminal.

What to Do Next

Once you have created the trustpoints and installed the Cisco UMA certificate on the ASA, create the TLS proxy instance. See Creating the TLS Proxy Instance, page 18-8.

Creating the TLS Proxy Instance

Create a TLS proxy instance for the Cisco UMA clients connecting to the Cisco UMA server.

Prerequisites

Before you can create the TLS proxy instance, you must have installed the Cisco UMA server self-signed certificate in the ASA truststore.

	Command	Purpose
Step 1	hostname(config)# **tls-proxy** *proxy_name* **Example:** tls-proxy cuma_tlsproxy	Creates the TLS proxy instance.
Step 2	hostname(config-tlsp)# **server trust-point** *proxy_name* **Example:** hostname(config-tlsp)# server trust-point cuma_proxy	Specifies the proxy trustpoint certificate presented during TLS handshake. The certificate must be owned by the ASA (identity certificate).

	Command	Purpose
Step 3	hostname(config-tlsp)# **client trust-point** *proxy_name* **Example:** hostname(config-tlsp)# client trust-point cuma_proxy	Specifies the trustpoint and associated certificate that the ASA uses in the TLS handshake when the ASA assumes the role of the TLS client. The certificate must be owned by the ASA (identity certificate).
Step 4	hostname(config-tlsp)# **no server authenticate-client**	Disables client authentication. Disabling TLS client authentication is required when the ASA must interoperate with a Cisco UMA client or clients such as a Web browser that are incapable of sending a client certificate.
Step 5	hostname(config-tlsp)# **client cipher-suite** *cipher_suite* **Example:** hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1	Specifies cipher suite configuration. For client proxy (the proxy acts as a TLS client to the server), the user-defined cipher suite replaces the default cipher suite.

What to Do Next

Once you have created the TLS proxy instance, enable it for MMP inspection. See Enabling the TLS Proxy for MMP Inspection, page 18-9.

Enabling the TLS Proxy for MMP Inspection

Cisco UMA client and server communications can be proxied via TLS, which decrypts the data, passes it to the inspect MMP module, and re-encrypt the data before forwarding it to the endpoint. The inspect MMP module verifies the integrity of the MMP headers and passes the OML/HTTP to an appropriate handler.

	Command	Purpose
Step 1	hostname(config)# **class-map** *class_map_name* **Example:** hostname(config)# class-map cuma_tlsproxy	Configures the class of traffic to inspect. Traffic between the Cisco UMA server and client uses MMP and is handled by MMP inspection. Where *class_map_name* is the name of the MMP class map.
Step 2	hostname(config-cmap)# **match port tcp eq** *port* **Example:** hostname(config-cmap)# match port tcp eq 5443	Matches the TCP port to which you want to apply actions for MMP inspection. The TCP/TLS default port for MMP inspection is 5443.
Step 3	hostname(config-cmap)# **exit**	Exits from the Class Map configuration mode.
Step 4	hostname(config)# **policy-map** *name* **Example:** hostname(config)# policy-map global_policy	Configures the policy map and attaches the action to the class of traffic.
Step 5	hostname(config-pmap)# **class** *classmap-name* **Example:** hostname(config-pmap)# class cuma_proxy	Assigns a class map to the policy map so that you can assign actions to the class map traffic. Where *classmap_name* is the name of the Skinny class map.

	Command	Purpose
Step 6	hostname(config-pmap)# **inspect mmp tls-proxy** *proxy_name* **Example:** hostname(config-pmap)# inspect mmp tls-proxy cuma_proxy	Enables SCCP (Skinny) application inspection and enables the phone proxy for the specified inspection session.
Step 7	hostname(config-pmap)# **exit**	Exits from the Policy Map configuration mode.
Step 8	hostname(config)# **service-policy** *policy_map_name* **global** **Example:** service-policy global_policy global	Enables the service policy on all interfaces.

Monitoring for Cisco Mobility Advantage

Mobility advantage proxy can be debugged the same way as IP Telephony. You can enable TLS proxy debug flags along with SSL syslogs to debug TLS proxy connection problems.

For example, using the following commands to enable TLS proxy-related debugging and syslog output only:

```
hostname# debug inspect tls-proxy events
hostname# debug inspect tls-proxy errors
hostname# config terminal
hostname(config)# logging enable
hostname(config)# logging timestamp
hostname(config)# logging list loglist message 711001
hostname(config)# logging list loglist message 725001-725014
hostname(config)# logging list loglist message 717001-717038
hostname(config)# logging buffer-size 1000000
hostname(config)# logging buffered loglist
hostname(config)# logging debug-trace
```

For information about TLS proxy debugging techniques and sample output, see the Monitoring the TLS Proxy, page 17-15.

Enable the **debug mmp** command for MMP inspection engine debugging:

```
MMP:: received 60 bytes from outside:1.1.1.1/2000 to inside:2.2.2.2/5443
MMP:: version OLWP-2.0
MMP:: forward 60/60 bytes from outside:1.1.1.1/2000 to inside:2.2.2.2/5443
MMP:: received 100 bytes from inside:2.2.2.2/5443 to outside:1.1.1.1/2000
MMP:: session-id: ABCD_1234
MMP:: status: 201
MMP:: forward 100/100 bytes from inside:2.2.2.2/5443 to outside 1.1.1.1/2000
MMP:: received 80 bytes from outside:1.1.1.1/2000 to inside:2.2.2.2/5443
MMP:: content-type: http/1.1
MMP:: content-length: 40
```

You can also capture the raw and decrypted data by the TLS proxy by entering the following commands:

```
hostname# capture mycap interface outside (capturing raw packets)
hostname# capture mycap-dec type tls-proxy interface outside (capturing decrypted data)
hostname# show capture capture_name
hostname# copy /pcap capture:capture_name tftp://tftp_location
```

Configuration Examples for Cisco Mobility Advantage

- Example 1: Cisco UMC/Cisco UMA Architecture – Security Appliance as Firewall with TLS Proxy and MMP Inspection, page 18-11
- Example 2: Cisco UMC/Cisco UMA Architecture – Security Appliance as TLS Proxy Only, page 18-12

This section describes sample configurations that apply to two deployment scenarios for the TLS proxy used by the Cisco Mobility Advantage solution—scenario 1 where the ASA functions as both the firewall and TLS proxy and scenario 2 where the ASA functions as the TLS proxy only. In both scenarios, the clients connect from the Internet.

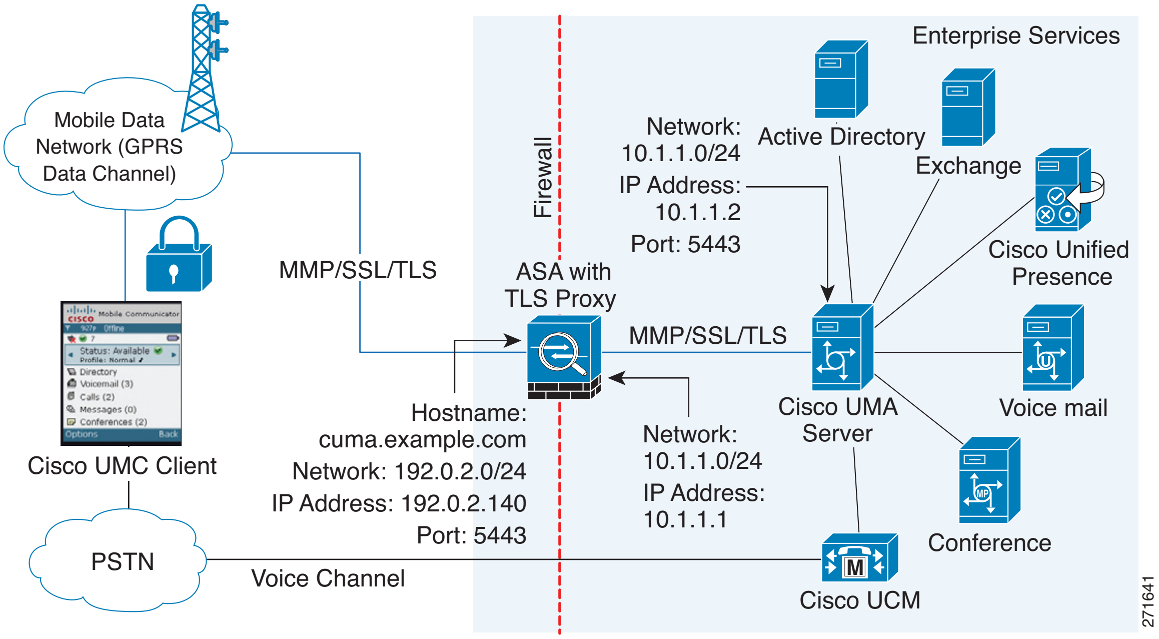
In the samples, you export the Cisco UMA server certificate and key-pair in PKCS-12 format and import it to the ASA. The certificate will be used during handshake with the Cisco UMA clients.

Installing the Cisco UMA server self-signed certificate in the ASA truststore is necessary for the ASA to authenticate the Cisco UMA server during handshake between the ASA proxy and Cisco UMA server. You create a TLS proxy instance for the Cisco UMA clients connecting to the Cisco UMA server. Lastly, you must enable TLS proxy for MMP inspection.

Example 1: Cisco UMC/Cisco UMA Architecture – Security Appliance as Firewall with TLS Proxy and MMP Inspection

As shown in Figure 18-5 (scenario 1—the recommended architecture), the ASA functions as both the firewall and TLS proxy. In the scenario 1 deployment, the ASA is between a Cisco UMA client and a Cisco UMA server. In this scenario, the ASA performs static NAT by translating the Cisco UMA server 10.1.1.2 IP address to 192.0.2.140.

Figure 18-5 Cisco UMC/Cisco UMA Architecture – Scenario 1: Security Appliance as Firewall with TLS Proxy and MMP Inspection



```
object network obj-10.1.1.2-01
    host 10.1.1.2
    nat (inside,outside) static 192.0.2.140
crypto ca import cuma_proxy pkcs12 sample_passphrase
    <cut-paste base 64 encoded pkcs12 here>
    quit
! for CUMA server’s self-signed certificate
crypto ca trustpoint cuma_server
    enrollment terminal
crypto ca authenticate cuma_server
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
    [ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit
tls-proxy cuma_proxy
    server trust-point cuma_proxy
    no server authenticate-client
    client cipher-suite aes128-sha1 aes256-sha1
class-map cuma_proxy
    match port tcp eq 5443
policy-map global_policy
    class cuma_proxy
        inspect mmp tls-proxy cuma_proxy
service-policy global_policy global
```

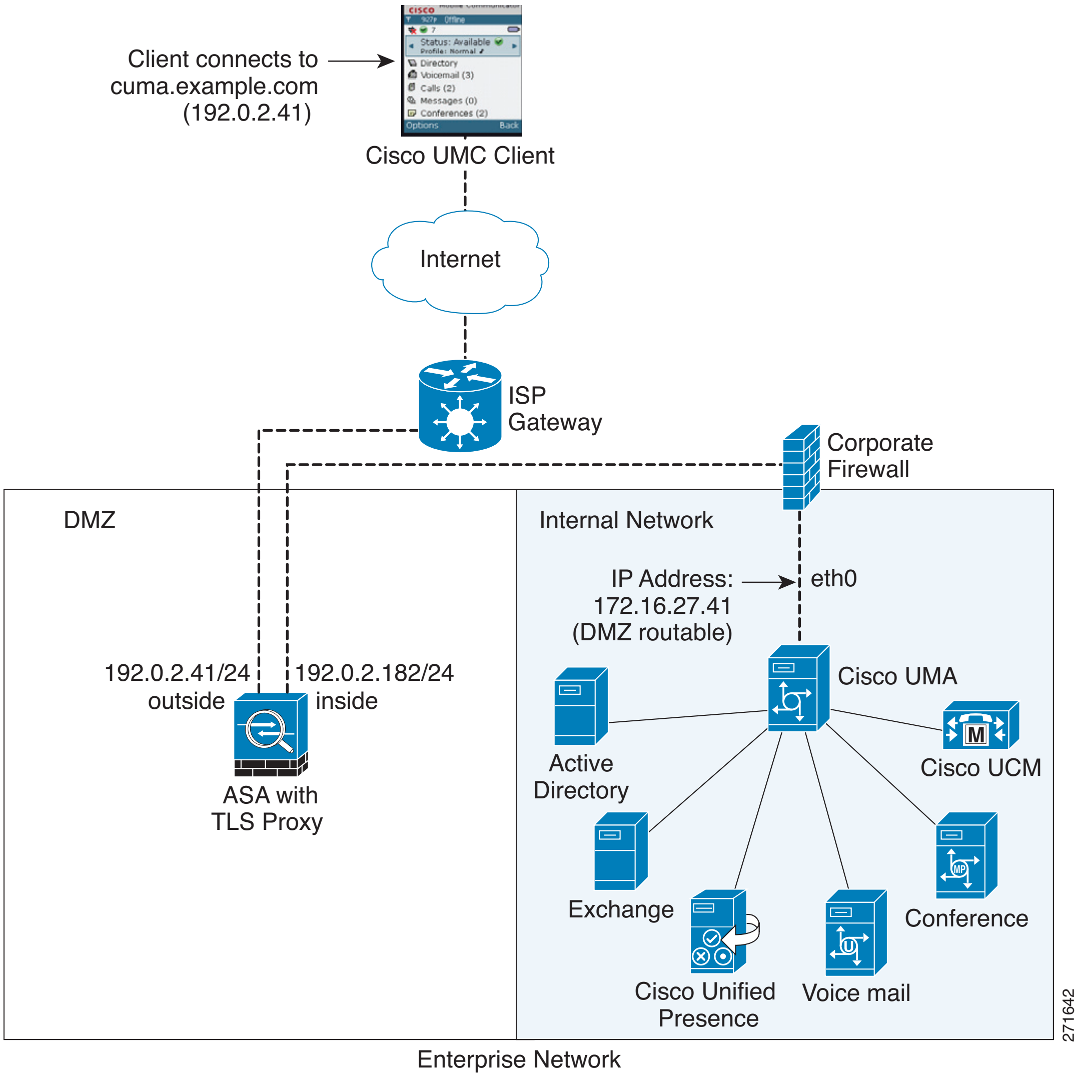
Example 2: Cisco UMC/Cisco UMA Architecture – Security Appliance as TLS Proxy Only

As shown in Figure 18-6 (scenario 2), the ASA functions as the TLS proxy only and works with an existing firewall. The ASA and the corporate firewall are performing NAT. The corporate firewall will not be able to predict which client from the Internet needs to connect to the corporate Cisco UMA server. Therefore, to support this deployment, you can take the following actions:

- Set up a NAT rule for inbound traffic that translates the destination IP address 192.0.2.41 to 172.16.27.41.
- Set up an interface PAT rule for inbound traffic translating the source IP address of every packet so that the corporate firewall does not need to open up a wildcard pinhole. The Cisco UMA server receives packets with the source IP address 192.0.2.183.

```
hostname(config)# object network obj-0.0.0.0-01
hostname(config-network-object)# subnet 0.0.0.0 0.0.0.0
hostname(config-network-object)# nat (outside,inside) dynamic 192.0.2.183
```

Figure 18-6 Cisco UMC/Cisco UMA Architecture – Scenario 2: Security Appliance as TLS Proxy Only



```
object network obj-172.16.27.41-01
    host 172.16.27.41
    nat (inside,outside) static 192.0.2.140
object network obj-0.0.0.0-01
    subnet 0.0.0.0 0.0.0.0
    nat (outside,inside) dynamic 192.0.2.183
crypto ca import cuma_proxy pkcs12 sample_passphrase
    <cut-paste base 64 encoded pkcs12 here>
    quit
! for CUMA server's self-signed certificate
crypto ca trustpoint cuma_server
    enrollment terminal
crypto ca authenticate cuma_server
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
    [ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit
```

```
tls-proxy cuma_proxy
    server trust-point cuma_proxy
    no server authenticate-client
    client cipher-suite aes128-sha1 aes256-sha1
class-map cuma_proxy
    match port tcp eq 5443
policy-map global_policy
    class cuma_proxy
        inspect mmp tls-proxy cuma_proxy
service-policy global_policy global
```

Feature History for Cisco Mobility Advantage

Table 18-1 lists the release history for this feature.

Table 18-1 Feature History for Cisco Phone Proxy

Feature Name	Releases	Feature Information
Cisco Mobility Advantage Proxy	8.0(4)	The Cisco Mobility Advantage Proxy feature was introduced.
Cisco Mobility Advantage Proxy	8.3(1)	The Unified Communications Wizard was added to ASDM. By using the wizard, you can configure the Cisco Mobility Advantage Proxy.

Configuring Cisco Unified Presence

This chapter describes how to configure the adaptive security appliance for Cisco Unified Presence.

This chapter includes the following sections:

Information About Cisco Unified Presence

This section includes the following topics:

Architecture for Cisco Unified Presence for SIP Federation Deployments

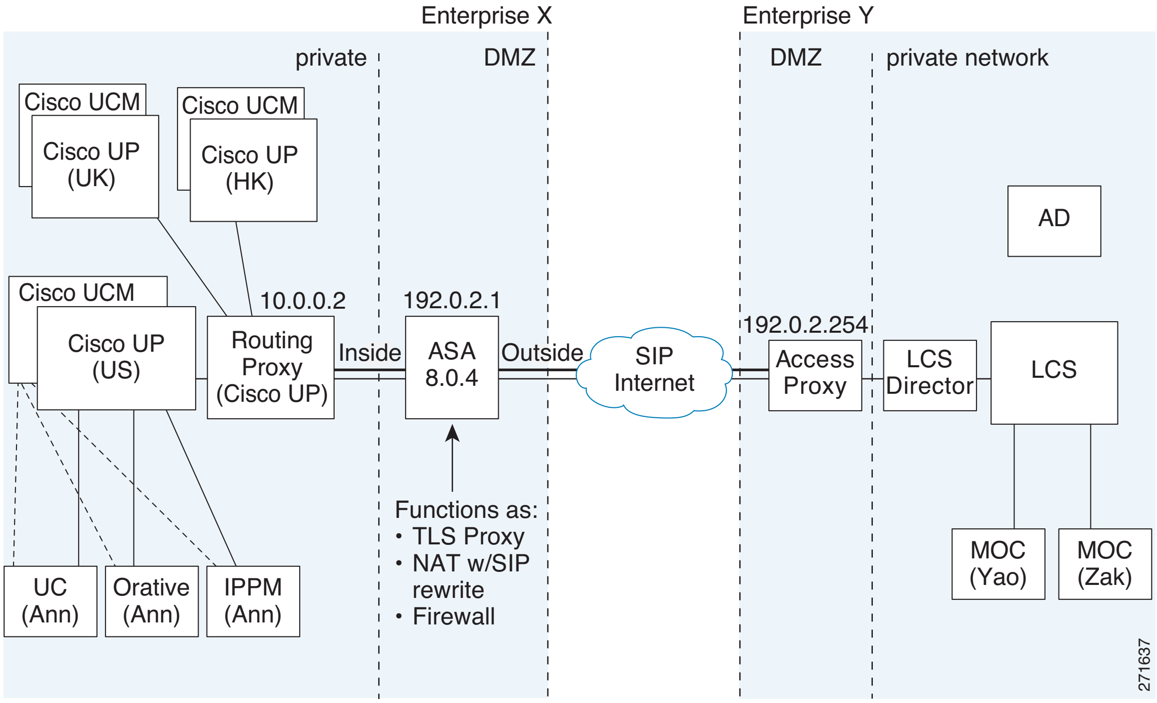
Figure 19-1 depicts a Cisco Unified Presence/LCS Federation scenario with the ASA as the presence federation proxy (implemented as a TLS proxy). The two entities with a TLS connection are the "Routing Proxy" (a dedicated Cisco UP) in Enterprise X and the Microsoft Access Proxy in Enterprise Y. However, the deployment is not limited to this scenario. Any Cisco UP or Cisco UP cluster could be deployed on the left side of the ASA; the remote entity could be any server (an LCS, an OCS, or another Cisco UP).

The following architecture is generic for two servers using SIP (or other ASA inspected protocols) with a TLS connection.

Entity X: Cisco UP/Routing Proxy in Enterprise X

Entity Y: Microsoft Access Proxy/Edge server for LCS/OCS in Enterprise Y

Figure 19-1 Typical Cisco Unified Presence/LCS Federation Scenario



In the above architecture, the ASA functions as a firewall, NAT, and TLS proxy, which is the recommended architecture. However, the ASA can also function as NAT and the TLS proxy alone, working with an existing firewall.

Either server can initiate the TLS handshake (unlike IP Telephony or Cisco Unified Mobility, where only the clients initiate the TLS handshake). There are by-directional TLS proxy rules and configuration. Each enterprise can have an ASA as the TLS proxy.

In Figure 19-1, NAT or PAT can be used to hide the private address of Entity X. In this situation, static NAT or PAT must be configured for foreign server (Entity Y) initiated connections or the TLS handshake (inbound). Typically, the public port should be 5061. The following static PAT command is required for the Cisco UP that accepts inbound connections:

```
ciscoasa(config)# object network obj-10.0.0.2-01
ciscoasa(config-network-object)# host 10.0.0.2
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5061
5061
```

The following static PAT must be configured for each Cisco UP that could initiate a connection (by sending SIP SUBSCRIBE) to the foreign server.

For Cisco UP with the address 10.0.0.2, enter the following command:

```
ciscoasa(config)# object network obj-10.0.0.2-02
ciscoasa(config-network-object)# host 10.0.0.2
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5062
5062
ciscoasa(config)# object network obj-10.0.0.2-03
ciscoasa(config-network-object)# host 10.0.0.2
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service udp 5070
5070
ciscoasa(config)# object network obj-10.0.0.2-04
ciscoasa(config-network-object)# host 10.0.0.2
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5060
5060
```

For another Cisco UP with the address 10.0.0.3, you must use a different set of PAT ports, such as 45062 or 45070:

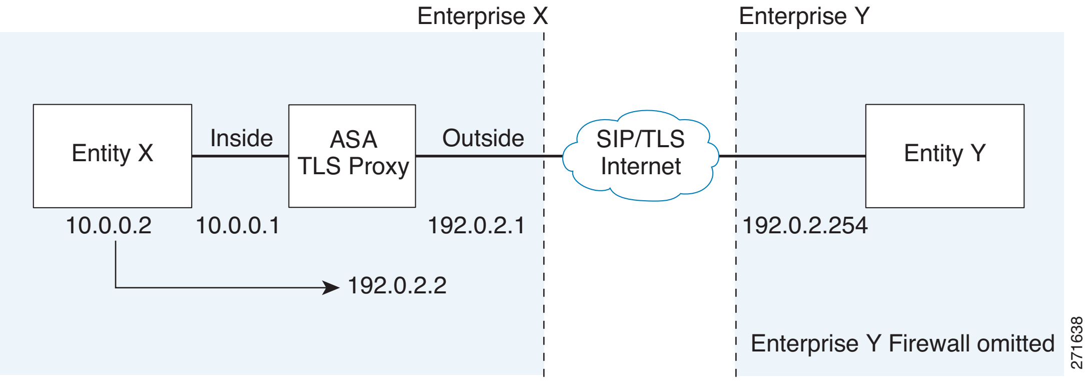
```
ciscoasa(config)# object network obj-10.0.0.3-01
ciscoasa(config-network-object)# host 10.0.0.3
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5061
45061
ciscoasa(config)# object network obj-10.0.0.3-02
ciscoasa(config-network-object)# host 10.0.0.3
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5062
45062
ciscoasa(config)# object network obj-10.0.0.3-03
ciscoasa(config-network-object)# host 10.0.0.3
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service udp 5070
5070
ciscoasa(config)# object network obj-10.0.0.2-03
ciscoasa(config-network-object)# host 10.0.0.2
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5070
45070
ciscoasa(config)# object network obj-10.0.0.3-04
ciscoasa(config-network-object)# host 10.0.0.3
ciscoasa(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5060
45060
```

Dynamic NAT or PAT can be used for the rest of the outbound connections or the TLS handshake. The ASA SIP inspection engine takes care of the necessary translation (fixup).

```
ciscoasa(config)# object network obj-0.0.0.0-01
ciscoasa(config-network-object)# subnet 0.0.0.0 0.0.0.0
ciscoasa(config-network-object)# nat (inside,outside) dynamic 192.0.2.1
```

Figure 19-2 illustrates an abstracted scenario with Entity X connected to Entity Y through the presence federation proxy on the ASA. The proxy is in the same administrative domain as Entity X. Entity Y could have another ASA as the proxy but this is omitted for simplicity.

Figure 19-2 Abstracted Presence Federation Proxy Scenario between Two Server Entities



For the Entity X domain name to be resolved correctly when the ASA holds its credential, the ASA could be configured to perform NAT for Entity X, and the domain name is resolved as the Entity X public address for which the ASA provides proxy service.

For further information about configuring Cisco Unified Presence Federation for SIP Federation, see the Integration Guide for Configuring Cisco Unified Presence for Interdomain Federation.:

http://www.cisco.com/en/US/products/ps6837/products_installation_and_configuration_guides_list.html

Trust Relationship in the Presence Federation

Within an enterprise, setting up a trust relationship is achievable by using self-signed certificates or you can set it up on an internal CA.

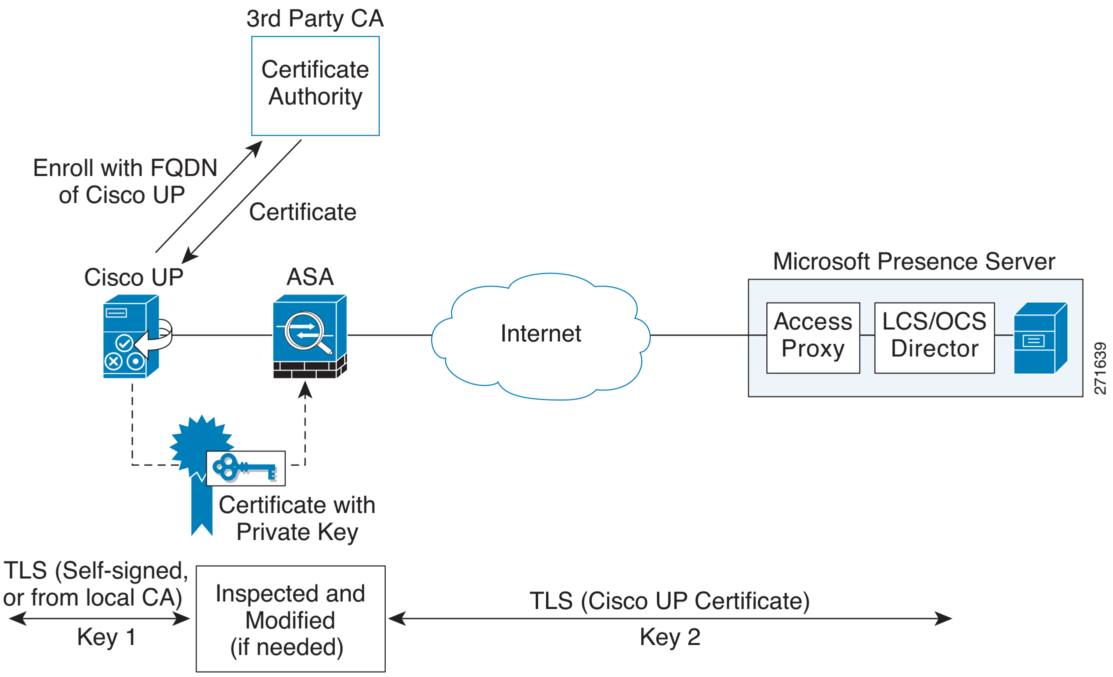
Establishing a trust relationship cross enterprises or across administrative domains is key for federation. Cross enterprises you must use a trusted third-party CA (such as, VeriSign). The ASA obtains a certificate with the FQDN of the Cisco UP (certificate impersonation).

For the TLS handshake, the two entities could validate the peer certificate via a certificate chain to trusted third-party certificate authorities. Both entities enroll with the CAs. The ASA as the TLS proxy must be trusted by both entities. The ASA is always associated with one of the enterprises. Within that enterprise (Enterprise X in Figure 19-1), the entity and the ASA could authenticate each other via a local CA, or by using self-signed certificates.

To establish a trusted relationship between the ASA and the remote entity (Entity Y), the ASA can enroll with the CA on behalf of Entity X (Cisco UP). In the enrollment request, the Entity X identity (domain name) is used.

Figure 19-3 shows the way to establish the trust relationship. The ASA enrolls with the third party CA by using the Cisco UP FQDN as if the ASA is the Cisco UP.

Figure 19-3 How the Security Appliance Represents Cisco Unified Presence – Certificate Impersonate


3rd Party CA
Certificate Authority
Enroll with FQDN of Cisco UP
Certificate
Cisco UP
ASA
Internet
Microsoft Presence Server
Access Proxy
LCS/OCS Director
271639
Certificate with Private Key
TLS (Self-signed, or from local CA)
Key 1
Inspected and Modified (if needed)
TLS (Cisco UP Certificate)
Key 2

Security Certificate Exchange Between Cisco UP and the Security Appliance

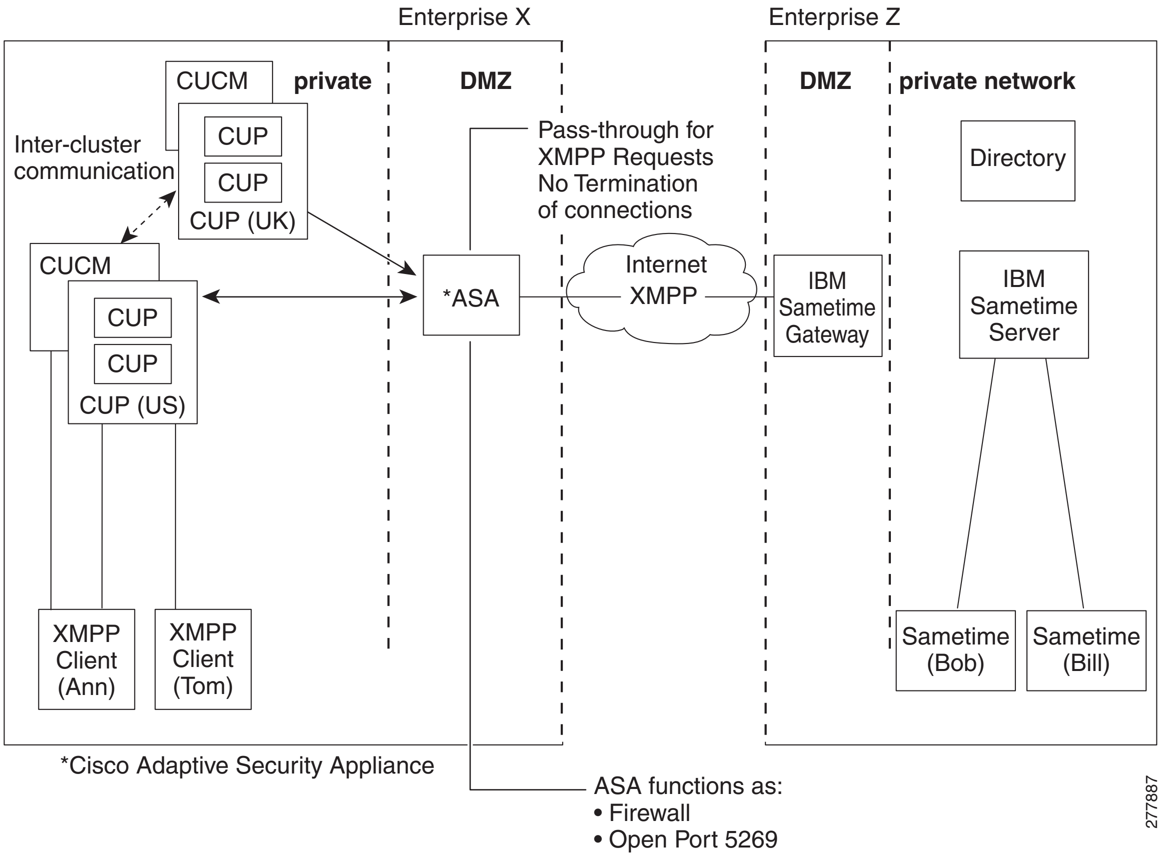
You need to generate the keypair for the certificate (such as `cup_proxy_key`) used by the ASA, and configure a trustpoint to identify the self-signed certificate sent by the ASA to Cisco UP (such as `cup_proxy`) in the TLS handshake.

For the ASA to trust the Cisco UP certificate, you need to create a trustpoint to identify the certificate from the Cisco UP (such as `cert_from_cup`), and specify the enrollment type as terminal to indicate that you will paste the certificate received from the Cisco UP into the terminal.

XMPP Federation Deployments

Figure 19-4 provides an example of an XMPP federated network between Cisco Unified Presence enterprise deployment and an IBM Sametime enterprise deployment. TLS is optional for XMPP federation. ASA acts only as a firewall for XMPP federation; it does not provide TLS proxy functionality or PAT for XMPP federation.

Figure 19-4 Basic XMPP Federated Network between Cisco Unified Presence and IBM Sametime



There are two DNS servers within the internal Cisco Unified Presence enterprise deployment. One DNS server hosts the Cisco Unified Presence private address. The other DNS server hosts the Cisco Unified Presence public address and a DNS SRV records for SIP federation (_sipfederationtle), and XMPP federation (_xmpp-server) with Cisco Unified Presence. The DNS server that hosts the Cisco Unified Presence public address is located in the local DMZ.

For further information about configuring Cisco Unified Presence Federation for XMPP Federation, see the *Integration Guide for Configuring Cisco Unified Presence Release 8.0 for Interdomain Federation*:

http://www.cisco.com/en/US/products/ps6837/products_installation_and_configuration_guides_list.html

Configuration Requirements for XMPP Federation

For XMPP Federation, ASA acts as a firewall only. You must open port 5269 for both incoming and outgoing XMPP federated traffic on ASA.

These are sample ACLs to open port 5269 on ASA.

Allow traffic from any address to any address on port 5269:

```
access-list ALLOW-ALL extended permit tcp any any eq 5269
```

Allow traffic from any address to any single node on port 5269:

```
access-list ALLOW-ALL extended permit tcp any host <private cup IP address> eq 5269
```

If you do not configure the ACL above, and you publish additional XMPP federation nodes in DNS, you must configure access to each of these nodes, for example:

```
object network obj_host_<private cup ip address>
#host <private cup ip address>
object network obj_host_<private cup2 ip address>
#host <private cup2 ip address>
object network obj_host_<public cup ip address>
#host <public cup ip address>
....
```

Configure the following NAT commands:

```
nat (inside,outside) source static obj_host_<private cup1 IP> obj_host_<public cup IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_<private cup1 IP> obj_host_<public cup IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

If you publish a single public IP address in DNS, and use arbitrary ports, configure the following:

(This example is for two additional XMPP federation nodes)

```
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_25269
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_25269

nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_35269
nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_35269
```

If you publish multiple public IP addresses in DNS all using port 5269, configure the following:

(This example is for two additional XMPP federation nodes)

```
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup2 IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup2 IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269

nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup3 IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

Licensing for Cisco Unified Presence

The Cisco Unified Presence feature supported by the ASA require a Unified Communications Proxy license.

The following table shows the Unified Communications Proxy license details by platform:

Note This feature is not available on No Payload Encryption models.

Model	License Requirement[1]
ASA 5505	Base License and Security Plus License: 2 sessions. *Optional license: 24 sessions.*
ASA 5510	Base License and Security Plus License: 2 sessions. *Optional licenses: 24, 50, or 100 sessions.*
ASA 5520	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, or 1000 sessions.*
ASA 5540	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, or 2000 sessions.*
ASA 5550	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*
ASA 5580	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]
ASA 5512-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, or 500 sessions.*
ASA 5515-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, or 500 sessions.*
ASA 5525-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, or 1000 sessions.*

Model	License Requirement[1]
ASA 5545-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, or 2000 sessions.*
ASA 5555-X	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*
ASA 5585-X with SSP-10	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, or 3000 sessions.*
ASA 5585-X with SSP-20, -40, or -60	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]
ASA SM	Base License: 2 sessions. *Optional licenses: 24, 50, 100, 250, 500, 750, 1000, 2000, 3000, 5000, or 10,000 sessions.*[2]

[1] The following applications use TLS proxy sessions for their connections. Each TLS proxy session used by these applications (and only these applications) is counted against the UC license limit:
- Phone Proxy
- Presence Federation Proxy
- Encrypted Voice Inspection

Other applications that use TLS proxy sessions do not count towards the UC limit, for example, Mobility Advantage Proxy (which does not require a license) and IME (which requires a separate IME license).

Some UC applications might use multiple sessions for a connection. For example, if you configure a phone with a primary and backup Cisco Unified Communications Manager, there are 2 TLS proxy connections, so 2 UC Proxy sessions are used.

You independently set the TLS proxy limit using the **tls-proxy maximum-sessions** command. To view the limits of your model, enter the **tls-proxy maximum-sessions ?** command. When you apply a UC license that is higher than the default TLS proxy limit, the security appliance automatically sets the TLS proxy limit to match the UC limit. The TLS proxy limit takes precedence over the UC license limit; if you set the TLS proxy limit to be less than the UC license, then you cannot use all of the sessions in your UC license.

Note: For license part numbers ending in “K8” (for example, licenses under 250 users), TLS proxy sessions are limited to 1000. For license part numbers ending in “K9” (for example, licenses 250 users or larger), the TLS proxy limit depends on the configuration, up to the model limit. K8 and K9 refer to whether the license is restricted for export: K8 is unrestricted, and K9 is restricted.

Note: If you clear the configuration (using the **clear configure all** command, for example), then the TLS proxy limit is set to the default for your model; if this default is lower than the UC license limit, then you see an error message to use the tls-proxy maximum-sessions command to raise the limit again . If you use failover and enter the **write standby** command on the primary unit to force a configuration synchronization, the **clear configure all** command is generated on the secondary unit automatically, so you may see the warning message on the secondary unit. Because the configuration synchronization restores the TLS proxy limit set on the primary unit, you can ignore the warning.

You might also use SRTP encryption sessions for your connections:
- For K8 licenses, SRTP sessions are limited to 250.
- For K9 licenses, there is not limit.

Note: Only calls that require encryption/decryption for media are counted towards the SRTP limit; if passthrough is set for the call, even if both legs are SRTP, they do not count towards the limit.

[2] With the 10,000-session UC license, the total combined sessions can be 10,000, but the maximum number of Phone Proxy sessions is 5000.

For more information about licensing, see the general operations configuration guide.

Configuring Cisco Unified Presence Proxy for SIP Federation

This section contains the following topics:

- Task Flow for Configuring Cisco Unified Presence Federation Proxy for SIP Federation, page 19-9

Task Flow for Configuring Cisco Unified Presence Federation Proxy for SIP Federation

To configure a Cisco Unified Presence/LCS Federation scenario with the ASA as the TLS proxy where there is a single Cisco UP that is in the local domain and self-signed certificates are used between the Cisco UP and the ASA (like the scenario shown in Figure 19-1), perform the following tasks.

Step 1 Create the following static NAT for the local domain containing the Cisco UP.

For the inbound connection to the local domain containing the Cisco UP, create static PAT by entering the following command:

```
hostname(config)# object network name
hostname(config-network-object)# host real_ip
hostname(config-network-object)# nat (real_ifc,mapped_ifc) static mapped_ip service {tcp |
udp} real_port mapped_port
```

Note For each Cisco UP that could initiate a connection (by sending SIP SUBSCRIBE) to the foreign server, you must also configure static PAT by using a different set of PAT ports.

For outbound connections or the TLS handshake, use dynamic NAT or PAT. The ASA SIP inspection engine takes care of the necessary translation (fixup).

```
hostname(config)# object network name
hostname(config-network-object)# subnet real_ip netmask
hostname(config-network-object)# nat (real_ifc,mapped_ifc) dynamic mapped_ip
```

For information about configuring NAT and PAT for the Cisco Presence Federation proxy, see Chapter 4, “Configuring Network Object NAT” and Chapter 5, “Configuring Twice NAT”.

Step 2 Create the necessary RSA keypairs and proxy certificate, which is a self-signed certificate, for the remote entity. See Creating Trustpoints and Generating Certificates, page 19-9.

Step 3 Install the certificates. See Installing Certificates, page 19-10.

Step 4 Create the TLS proxy instance for the Cisco UP clients connecting to the Cisco UP server. See Creating the TLS Proxy Instance, page 19-12.

Step 5 Enable the TLS proxy for SIP inspection. See Enabling the TLS Proxy for SIP Inspection, page 19-13.

Creating Trustpoints and Generating Certificates

You need to generate the keypair for the certificate (such as `cup_proxy_key`) used by the ASA, and configure a trustpoint to identify the self-signed certificate sent by the ASA to Cisco UP (such as `cup_proxy`) in the TLS handshake.

	Command	Purpose
Step 1	hostname(config)# **crypto key generate rsa label** *key-pair-label* **modulus** *size* **Example:** crypto key generate rsa label ent_y_proxy_key modulus 1024 INFO: The name for the keys will be: ent_y_proxy_key Keypair generation process begin. Please wait... hostname(config)#	Creates the RSA keypair that can be used for the trustpoints. The keypair is used by the self-signed certificate presented to the local domain containing the Cisco UP (proxy for the remote entity).
Step 2	hostname(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** hostname(config)# crypto ca trustpoint ent_y_proxy	Enters the trustpoint configuration mode for the specified trustpoint so that you can create the trustpoint for the remote entity. A trustpoint represents a CA identity and possibly a device identity, based on a certificate issued by the CA.
Step 3	hostname(config-ca-trustpoint)# **enrollment self**	Generates a self-signed certificate.
Step 4	hostname(config-ca-trustpoint)# **fqdn none**	Specifies not to include a fully qualified domain name (FQDN) in the Subject Alternative Name extension of the certificate during enrollment.
Step 5	hostname(config-ca-trustpoint)# **subject-name** *X.500_name* **Example:** hostname(config-ca-trustpoint)# subject-name cn=Ent-Y-Proxy	Includes the indicated subject DN in the certificate during enrollment
Step 6	hostname(config-ca-trustpoint)# **keypair** *keyname* **Example:** hostname(config-ca-trustpoint)# keypair ent_y_proxy_key	Specifies the key pair whose public key is to be certified.
Step 7	hostname(config-ca-trustpoint)# **exit**	Exits from the CA Trustpoint configuration mode.
Step 8	hostname(config)# **crypto ca enroll** *trustpoint* **Example:** hostname(config)# crypto ca enroll ent_y_proxy	Starts the enrollment process with the CA and specifies the name of the trustpoint to enroll with.

What to Do Next

Install the certificate on the local entity truststore. You could also enroll the certificate with a local CA trusted by the local entity. See the "Installing Certificates" section on page 19-10.

Installing Certificates

Export the self-signed certificate for the ASA created in the "Creating Trustpoints and Generating Certificates" section on page 19-9 and install it as a trusted certificate on the local entity. This task is necessary for local entity to authenticate the ASA.

Prerequisites

To create a proxy certificate on the ASA that is trusted by the remote entity, obtain a certificate from a trusted CA. For information about obtaining a certificate from a trusted CA, see the general operations configuration guide.

	Command	Purpose
Step 1	hostname(config)# **crypto ca export** *trustpoint* **identity-certificate** **Example:** hostname(config)# crypto ca export ent_y_proxy identity-certificate	Export the ASA self-signed (identity) certificate.
Step 2	hostname(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** hostname(config)# crypto ca trustpoint ent_x_cert ! for Entity X's self-signed certificate	Enters the trustpoint configuration mode for the specified trustpoint so that you can create the trustpoint for the local entity. A trustpoint represents a CA identity and possibly a device identity, based on a certificate issued by the CA.
Step 3	hostname(config-ca-trustpoint)# **enrollment terminal**	Specifies cut and paste enrollment with this trustpoint (also known as manual enrollment). If the local entity uses a self-signed certificate, the self-signed certificate must be installed; if the local entity uses a CA-issued certificate, the CA certificate needs to be installed. This configuration shows the commands for using a self-signed certificate.
Step 4	hostname(config-ca-trustpoint)# **exit**	Exits from the CA Trustpoint configuration mode.
Step 5	hostname(config)# **crypto ca authenticate** *trustpoint* **Example:** hostname(config)# crypto ca authenticate ent_x_cert Enter the base 64 encoded CA certificate. End with a blank line or the word "quit" on a line by itself [certificate data omitted] Certificate has the following attributes: Fingerprint: 21B598D5 4A81F3E5 0B24D12E 3F89C2E4 % Do you accept this certificate? [yes/no]: yes Trustpoint CA certificate accepted. % Certificate successfully imported	Installs and authenticates the CA certificates associated with a trustpoint created for the local entity. Where *trustpoint* specifies the trustpoint from which to obtain the CA certificate. Maximum name length is 128 characters. The ASA prompts you to paste the base-64 formatted CA certificate onto the terminal.
Step 6	hostname(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** hostname(config)# crypto ca trustpoint ent_y_ca ! for Entity Y's CA certificate	Install the CA certificate that signs the remote entity certificate on the ASA by entering the following commands. This step is necessary for the ASA to authenticate the remote entity.
Step 7	hostname(config-ca-trustpoint)# **enrollment terminal**	Specifies cut and paste enrollment with this trustpoint (also known as manual enrollment).
Step 8	hostname(config-ca-trustpoint)# **exit**	Exits from the CA Trustpoint configuration mode.
Step 9	hostname(config)# **crypto ca authenticate** *trustpoint* **Example:** hostname(config)# crypto ca authenticate ent_y_ca Enter the base 64 encoded CA certificate. End with a blank line or the word "quit" on a line by itself MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG 9w0BAQUFADCB [certificate data omitted] /7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==	Installs and authenticates the CA certificates associated with a trustpoint created for the local entity. The ASA prompts you to paste the base-64 formatted CA certificate onto the terminal.

What to Do Next

Once you have created the trustpoints and installed the certificates for the local and remote entities on the ASA, create the TLS proxy instance. See Creating the TLS Proxy Instance, page 19-12.

Creating the TLS Proxy Instance

Because either server can initiate the TLS handshake (unlike IP Telephony or Cisco Unified Mobility, where only the clients initiate the TLS handshake), you must configure by-directional TLS proxy rules. Each enterprise can have an ASA as the TLS proxy.

Create TLS proxy instances for the local and remote entity initiated connections respectively. The entity that initiates the TLS connection is in the role of "TLS client". Because the TLS proxy has a strict definition of "client" and "server" proxy, two TLS proxy instances must be defined if either of the entities could initiate the connection.

	Command	Purpose
Step 1	! Local entity to remote entity hostname(config)# **tls-proxy** proxy_name **Example:** hostname(config)# tls-proxy ent_x_to_y	Creates the TLS proxy instance.
Step 2	hostname(config-tlsp)# **server trust-point** *proxy_name* **Example:** hostname(config-tlsp)# server trust-point ent_y_proxy	Specifies the proxy trustpoint certificate presented during TLS handshake. The certificate must be owned by the ASA (identity certificate). Where the *proxy_name* for the **server trust-point** command is the remote entity proxy name.
Step 3	hostname(config-tlsp)# **client trust-point** *proxy_trustpoint* **Example:** hostname(config-tlsp)# client trust-point ent_x_cert	Specifies the trustpoint and associated certificate that the ASA uses in the TLS handshake when the ASA assumes the role of the TLS client. The certificate must be owned by the ASA (identity certificate). Where the *proxy_trustpoint* for the **client trust-point** command is the local entity proxy.
Step 4	hostname(config-tlsp)# **client cipher-suite** *cipher_suite* **Example:** hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1	Specifies cipher suite configuration. For client proxy (the proxy acts as a TLS client to the server), the user-defined cipher suite replaces the default cipher suite.
Step 5	! Remote entity to local entity hostname(config)# **tls-proxy** *proxy_name* **Example:** tls-proxy ent_y_to_x	Creates the TLS proxy instance.
Step 6	hostname(config-tlsp)# **server trust-point** *proxy_name* **Example:** hostname(config-tlsp)# server trust-point ent_x_cert	Specifies the proxy trustpoint certificate presented during TLS handshake. Where the *proxy_name* for the **server trust-point** command is the local entity proxy name

	Command	Purpose
Step 7	hostname(config-tlsp)# **client trust-point** *proxy_trustpoint* **Example:** hostname(config-tlsp)# client trust-point ent_y_proxy	Specifies the trustpoint and associated certificate that the ASA uses in the TLS handshake when the ASA assumes the role of the TLS client. Where the *proxy_trustpoint* for the **client trust-point** command is the remote entity proxy.
Step 8	hostname(config-tlsp)# **client cipher-suite** *cipher_suite* **Example:** hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1	Specifies cipher suite configuration.

What to Do Next

Once you have created the TLS proxy instance, enable it for SIP inspection. See Enabling the TLS Proxy for SIP Inspection, page 19-13.

Enabling the TLS Proxy for SIP Inspection

Enable the TLS proxy for SIP inspection and define policies for both entities that could initiate the connection.

	Command	Purpose
Step 1	hostname(config)# **access-list** *id* **extended permit tcp host** *src_ip* **host** *dest_ip* **eq** *port* **Examples:** access-list ent_x_to_y extended permit tcp host 10.0.0.2 host 192.0.2.254 eq 5061 access-list ent_y_to_x extended permit tcp host 192.0.2.254 host 192.0.2.1 eq 5061	Adds an Access Control Entry. The ACL is used to specify the class of traffic to inspect.
Step 2	hostname(config)# **class-map** *class_map_name* **Example:** hostname(config)# class-map ent_x_to_y	Configures the secure SIP class of traffic to inspect. Where *class_map_name* is the name of the SIP class map.
Step 3	hostname(config-cmap)# **match access-list** *access_list_name* **Example:** hostname(config-cmap)# match access-list ent_x_to_y	Identifies the traffic to inspect.
Step 4	hostname(config-cmap)# **exit**	Exits from Class Map configuration mode.
Step 5	hostname(config)# **policy-map type inspect sip** *policy_map_name* **Example:** hostname(config)# policy-map type inspect sip sip_inspect	Defines special actions for SIP inspection application traffic.
Step 6	hostname(config-pmap)# **parameters** ! SIP inspection parameters	Specifies the parameters for SIP inspection. Parameters affect the behavior of the inspection engine. The commands available in parameters configuration mode depend on the application.
Step 7	hostname(config-pmap)# **exit**	Exits from Policy Map configuration mode.

	Command	Purpose
Step 8	hostname(config)# **policy-map** name **Example:** hostname(config)# policy-map global_policy	Configure the policy map and attach the action to the class of traffic.
Step 9	hostname(config-pmap)# **class** classmap_*name* **Example:** hostname(config-pmap)# class ent_x_to_y	Assigns a class map to the policy map so that you can assign actions to the class map traffic. Where *classmap_name* is the name of the SIP class map.
Step 10	hostname(config-pmap)# **inspect sip** *sip_map* **tls-proxy** *proxy_name* hostname(config-pmap)# inspect sip sip_inspect tls-proxy ent_x_to_y	Enables TLS proxy for the specified SIP inspection session.
Step 11	hostname(config-pmap)# **exit**	Exits from Policy Map configuration mode.
Step 12	hostname(config)# **service-policy** *policy_map_name* **global** **Example:** hostname(config)# service-policy global_policy global	Enables the service policy for SIP inspection for all interfaces. Where name for the policy-map command is the name of the global policy map.

Monitoring Cisco Unified Presence

Debugging is similar to debugging TLS proxy for IP Telephony. You can enable TLS proxy debug flags along with SSL syslogs to debug TLS proxy connection problems.

For example, use the following commands to enable TLS proxy-related debug and syslog output only:

```
hostname(config)# debug inspect tls-proxy events
hostname(config)# debug inspect tls-proxy errors
hostname(config)# logging enable
hostname(config)# logging timestamp
hostname(config)# logging list loglist message 711001
hostname(config)# logging list loglist message 725001-725014
hostname(config)# logging list loglist message 717001-717038
hostname(config)# logging buffer-size 1000000
hostname(config)# logging buffered loglist
hostname(config)# logging debug-trace
```

For information about TLS proxy debugging techniques and sample output, see Monitoring the TLS Proxy, page 17-15.

Enable the **debug sip** command for SIP inspection engine debugging. See the command reference.

Additionally, you can capture the raw and decrypted data by the TLS proxy by entering the following commands:

```
hostname# capture mycap interface outside (capturing raw packets)
hostname# capture mycap-dec type tls-proxy interface outside (capturing decrypted data)
hostname# show capture capture_name
hostname# copy /pcap capture:capture_name tftp://tftp_location
```

Configuration Example for Cisco Unified Presence

This section contains the following topics:

- Example Configuration for SIP Federation Deployments, page 19-15

- Example ACL Configuration for XMPP Federation, page 19-17
- Example NAT Configuration for XMPP Federation, page 19-18

Example Configuration for SIP Federation Deployments

The following sample illustrates the necessary configuration for the ASA to perform TLS proxy for Cisco Unified Presence as shown in Figure 19-5. It is assumed that a single Cisco UP (Entity X) is in the local domain and self-signed certificates are used between Entity X and the ASA.

For each Cisco UP that could initiate a connection (by sending SIP SUBSCRIBE) to the foreign server, you must also configure static PAT and if you have another Cisco UP with the address (10.0.0.3 in this sample), it must use a different set of PAT ports (such as 45062 or 45070). Dynamic NAT or PAT can be used for outbound connections or TLS handshake. The ASA SIP inspection engine takes care of the necessary translation (fixup).

When you create the necessary RSA key pairs, a key pair is used by the self-signed certificate presented to Entity X (proxy for Entity Y). When you create a proxy certificate for Entity Y, the certificate is installed on the Entity X truststore. It could also be enrolled with a local CA trusted by Entity X.

Exporting the ASA self-signed certificate (ent_y_proxy) and installing it as a trusted certificate on Entity X is necessary for Entity X to authenticate the ASA. Exporting the Entity X certificate and installing it on the ASA is needed for the ASA to authenticate Entity X during handshake with X. If Entity X uses a self-signed certificate, the self-signed certificate must be installed; if Entity X uses a CA issued the certificate, the CA’s certificated needs to be installed.

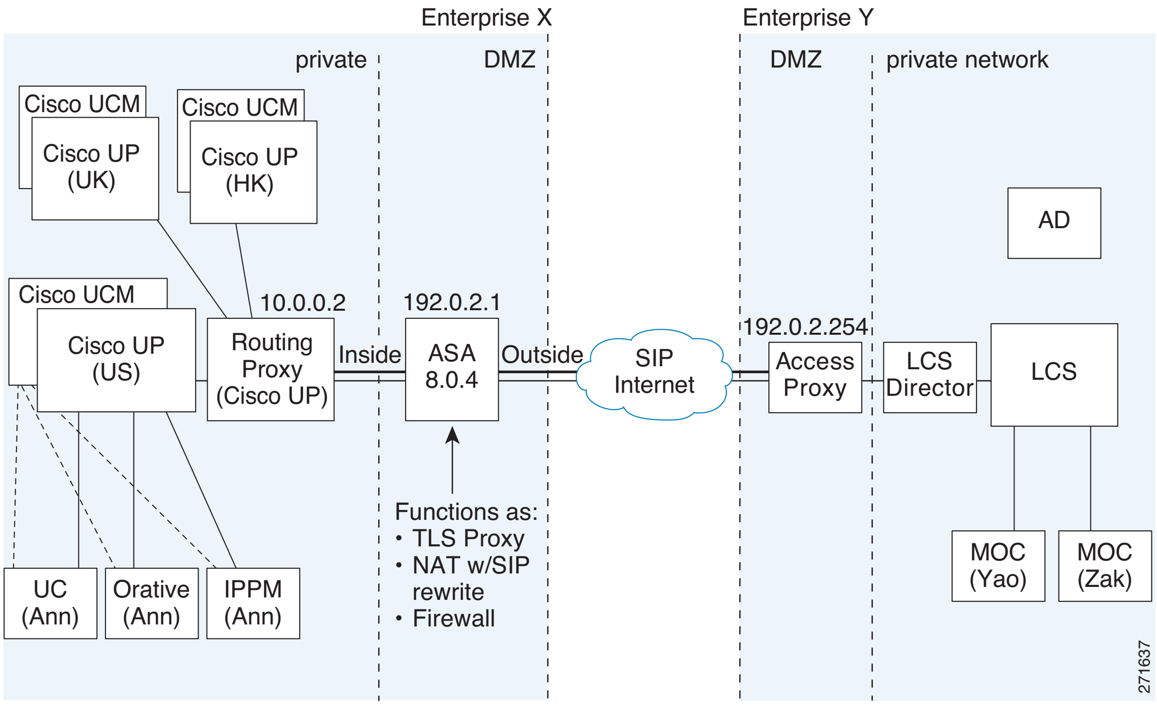
For about obtaining a certificate from a trusted CA, see the general operations configuration guide.

Installing the CA certificate that signs the Entity Y certificate on the ASA is necessary for the ASA to authenticate Entity Y.

When creating TLS proxy instances for Entity X and Entity Y, the entity that initiates the TLS connection is in the role of “TLS client”. Because the TLS proxy has strict definition of “client” and “server” proxy, two TLS proxy instances must be defined if either of the entities could initiate the connection.

When enabling the TLS proxy for SIP inspection, policies must be defined for both entities that could initiate the connection.

Figure 19-5 Typical Cisco Unified Presence/LCS Federation Scenario



```
object network obj-10.0.0.2-01
    host 10.0.0.2
    nat (inside,outside) static 192.0.2.1 service tcp 5061 5061
object network obj-10.0.0.2-02
    host 10.0.0.2
    nat (inside,outside) static 192.0.2.1 service tcp 5062 5062
object network obj-10.0.0.2-03
    host 10.0.0.2
    nat (inside,outside) static 192.0.2.1 service udp 5070 5070
object network obj-10.0.0.3-01
    host 10.0.0.3
    nat (inside,outside) static 192.0.2.1 service tcp 5062 45062
object network obj-10.0.0.3-02
    host 10.0.0.3
    nat (inside,outside) static 192.0.2.1 service udp 5070 45070
object network obj-0.0.0.0-01
    subnet 0.0.0.0 0.0.0.0
    nat (inside,outside) dynamic 192.0.2.1
crypto key generate rsa label ent_y_proxy_key modulus 1024
    ! for self-signed Entity Y proxy certificate
crypto ca trustpoint ent_y_proxy
    enrollment self
    fqdn none
    subject-name cn=Ent-Y-Proxy
    keypair ent_y_proxy_key
crypto ca enroll ent_y_proxy
crypto ca export ent_y_proxy identity-certificate
! for Entity X's self-signed certificate
crypto ca trustpoint ent_x_cert
    enrollment terminal
crypto ca authenticate ent_x_cert
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
    [ certificate data omitted ]
```

```
quit
! for Entity Y’s CA certificate
crypto ca trustpoint ent_y_ca
    enrollment terminal
crypto ca authenticate ent_y_ca
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
    [ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit
! Entity X to Entity Y
tls-proxy ent_x_to_y
    server trust-point ent_y_proxy
    client trust-point ent_x_cert
    client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1
! Entity Y to Entity X
tls-proxy ent_y_to_x
    server trust-point ent_x_cert
    client trust-point ent_y_proxy
    client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1
access-list ent_x_to_y extended permit tcp host 10.0.0.2 host 192.0.2.254 eq 5061
access-list ent_y_to_x extended permit tcp host 192.0.2.254 host 192.0.2.1 eq 5061
class-map ent_x_to_y
    match access-list ent_x_to_y
class-map ent_y_to_x
    match access-list ent_y_to_x
policy-map type inspect sip sip_inspect
    parameters
        ! SIP inspection parameters
policy-map global_policy
    class ent_x_to_y
        inspect sip sip_inspect tls-proxy ent_x_to_y
    class ent_y_to_x
        inspect sip sip_inspect tls-proxy ent_y_to_x
service-policy global_policy global
```

Example ACL Configuration for XMPP Federation

Example 1: This example ACL configuration allows from any address to any address on port 5269:

```
access-list ALLOW-ALL extended permit tcp any any eq 5269
```

Example 2: This example ACL configuration allows from any address to any single XMPP federation node on port 5269. The following values are used in this example:

- Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1
- XMPP federation listening port = 5269

```
access-list ALLOW-ALL extended permit tcp any host 1.1.1.1 eq 5269
```

Example 3: This example ACL configuration allows from any address to specific XMPP federation nodes published in DNS.

Note The public addresses are published in DNS, but the private addresses are configured in the access-list command.

The following values are used in this sample configuration:

• Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1

• Private second Cisco Unified Presence Release 8.0 IP address= 2.2.2.2

• Private third Cisco Unified Presence Release 7.x IP address = 3.3.3.3

• XMPP federation listening port = 5269

```
access-list ALLOW-ALL extended permit tcp any host 1.1.1.1 eq 5269
access-list ALLOW-ALL extended permit tcp any host 2.2.2.2 eq 5269
access-list ALLOW-ALL extended permit tcp any host 3.3.3.3 eq 5269
```

Example 4: This example ACL configuration allows only from a specific federated domain interface to specific XMPP federation nodes published in DNS.

Note The public addresses are published in DNS, but the private addresses are configured in the access-list command.

The following values are used in this sample configuration:

- Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1
- Private second Cisco Unified Presence Release 8.0 IP address = 2.2.2.2
- Private third Cisco Unified Presence Release 7.x IP address = 3.3.3.3
- XMPP federation listening port = 5269
- External interface of the foreign XMPP enterprise = 100.100.100.100

```
access-list ALLOW-ALL extended permit tcp host 100.100.100.100 host 1.1.1.1 eq 5269
access-list ALLOW-ALL extended permit tcp host 100.100.100.100 host 2.2.2.2 eq 5269
access-list ALLOW-ALL extended permit tcp host 100.100.100.100 host 3.3.3.3 eq 5269
```

Example NAT Configuration for XMPP Federation

Example 1: Single node with XMPP federation enabled

The following values are used in this sample configuration:

- Public Cisco Unified Presence IP address = 10.10.10.10
- Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1
- XMPP federation listening port = 5269

```
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

Example 2: Multiple nodes with XMPP federation, each with a public IP address in DNS

The following values are used in this sample configuration:

- Public Cisco Unified Presence IP addresses = 10.10.10.10, 20.20.20.20, 30.30.30.30
- Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1
- Private second Cisco Unified Presence Release 8.0 IP address = 2.2.2.2

- Private third Cisco Unified Presence Release 7.x IP address = 3.3.3.3
- XMPP federation listening port = 5269

```
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269

nat (inside,outside) source static obj_host_2.2.2.2 obj_host_20.20.20.20 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_2.2.2.2 obj_host_20.20.20.20 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269

nat (inside,outside) source static obj_host_3.3.3.3 obj_host_30.30.30.30 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_3.3.3.3 obj_host_30.30.30.30 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

Example 3: Multiple nodes with XMPP federation, but a single public IP address in DNS with arbitrary ports published in DNS (PAT).

The following values are used in this sample configuration:

- Public Cisco Unified Presence IP Address = 10.10.10.10
- Private XMPP federation Cisco Unified Presence Release 8.0 IP address = 1.1.1.1, port 5269
- Private second Cisco Unified Presence Release 8.0 IP address = 2.2.2.2, arbitrary port 25269
- Private third Cisco Unified Presence Release 7.x IP address = 3.3.3.3, arbitrary port 35269

```
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269

nat (inside,outside) source static obj_host_2.2.2.2 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_25269
nat (inside,outside) source static obj_host_2.2.2.2 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_25269

nat (inside,outside) source static obj_host_3.3.3.3 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_35269
nat (inside,outside) source static obj_host_3.3.3.3 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_35269
```

Feature History for Cisco Unified Presence

Table 19-1 lists the release history for this feature.

Table 19-1 Feature History for Cisco Unified Presence

Feature Name	Releases	Feature Information
Cisco Presence Federation Proxy	8.0(4)	The Cisco Unified Presence proxy feature was introduced.
Cisco Presence Federation Proxy	8.3(1)	The Unified Communications Wizard was added to ASDM. By using the wizard, you can configure the Cisco Presence Federation Proxy. Support for XMPP Federation was introduced.

CHAPTER 20

Configuring Cisco Intercompany Media Engine Proxy

This chapter describes how to configure the ASA for Cisco Intercompany Media Engine Proxy.

This chapter includes the following sections:

Information About Cisco Intercompany Media Engine Proxy

This section includes the following topics:

Features of Cisco Intercompany Media Engine Proxy

Cisco Intercompany Media Engine enables companies to interconnect on-demand, over the Internet with advanced features made available by VoIP technologies. Cisco Intercompany Media Engine allows for business-to-business federation between Cisco Unified Communications Manager clusters in different enterprises by utilizing peer-to-peer, security, and SIP protocols to create dynamic SIP trunks between businesses. A collection of enterprises work together to end up looking like one large business with inter-cluster trunks between them.

The adaptive security appliance applies its existing TLS proxy, SIP Application Layer Gateway (ALG), and SIP verification features to the functioning of Cisco Intercompany Media Engine.

Cisco Intercompany Media Engine has the following key features:

- Works with existing phone numbers: Cisco Intercompany Media Engine works with the phone numbers an enterprise currently has and does not require an enterprise to learn new numbers or change providers to use Cisco Intercompany Media Engine.
- Works with existing IP phones: Cisco Intercompany Media Engine works with the existing IP phones within an enterprise. However, the feature set in business-to-business calls is limited to the capabilities of the IP phones.
- Does not require purchasing new services: Cisco Intercompany Media Engine does not require any new services from any service providers. Customers continue to use the PSTN connectivity they have and the Internet connectivity they have today. Cisco Intercompany Media Engine gradually moves calls off the PSTN and onto the Internet.
- Provides a full Cisco Unified Communications experience: Because Cisco Intercompany Media Engine creates inter-cluster SIP trunks between enterprises, any Unified Communication features that work over the SIP trunk and only require a SIP trunk work with the Cisco Intercompany Media Engine, thus providing a Unified Communication experience across enterprises.
- Works on the Internet: Cisco Intercompany Media Engine was designed to work on the Internet. It can also work on managed extranets.
- Provides worldwide reach: Cisco Intercompany Media Engine can connect to any enterprise anywhere in the world, as long as the enterprise is running Cisco Intercompany Media Engine technology. There are no regional limitations. This is because Cisco Intercompany Media Engine utilizes two networks that both have worldwide reach—the Internet and the PSTN.
- Allows for unlimited scale: Cisco Intercompany Media Engine can work with any number of enterprises.
- Is self-learning: The system is primarily self-learning. Customers do not have to enter information about other businesses: no phone prefixes, no IP address, no ports, no domain names, nor certificates. Customers need to configure information about their own networks, and provide policy information if they want to limit the scope of Cisco Intercompany Media Engine.
- Is secure: Cisco Intercompany Media Engine is secure, utilizing a large number of different technologies to accomplish this security.
- Includes anti-spam: Cisco Intercompany Media Engine prevents people from setting up software on the Internet that spams enterprises with phone calls. It provides an extremely high barrier to entry.
- Provides for QoS management: Cisco Intercompany Media Engine provides features that help customers manage the QoS on the Internet, such as the ability to monitor QoS of the RTP traffic in real-time and fallback to PSTN automatically if problems arise.

How the UC-IME Works with the PSTN and the Internet

The Cisco Intercompany Media Engine utilizes two networks that both have worldwide reach—the Internet and the PSTN. Customers continue to use the PSTN connectivity they have. The Cisco Intercompany Media Engine gradually moves calls off the PSTN and onto the Internet. However, if QoS problems arise, the Cisco Intercompany Media Engine Proxy monitors QoS of the RTP traffic in real-time and fallbacks to PSTN automatically.

The Cisco Intercompany Media Engine uses information from PSTN calls to validate that the terminating side owns the number that the originated side had called. After the PSTN call terminates, the enterprises involved in the call send information about the call to their Cisco IME server. The Cisco IME server on the originating side validates the call.

On successful verification, the terminating side creates a ticket that grants permission to the call originator to make a Cisco IME call to a specific number. See Tickets and Passwords, page 20-3 for information.

Tickets and Passwords

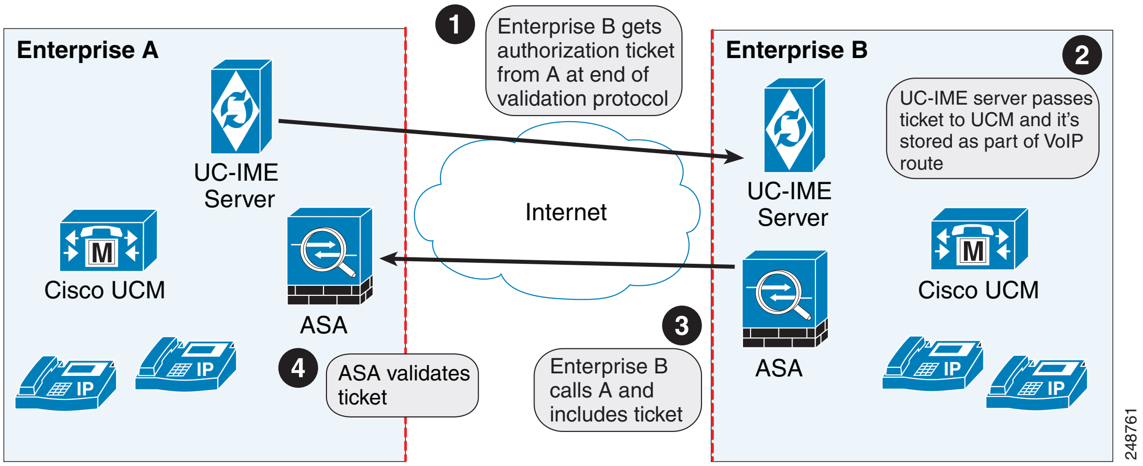
Cisco Intercompany Media Engine utilizes tickets and passwords to provide enterprise verification. Verification through the creation of tickets ensures an enterprise is not subject to denial-of-service (DOS) attacks from the Internet or endless VoIP spam calls. Ticket verification prevents spam and DOS attacks because it introduces a cost to the VoIP caller; namely, the cost of a PSTN call. A malicious user cannot set up just an open source asterisk PBX on the Internet and begin launching SIP calls into an enterprise running Cisco Intercompany Media Engine. Having the Cisco Intercompany Media Engine Proxy verify tickets allows incoming calls from a particular enterprise to a particular number only when that particular enterprise has previously called that phone number on the PSTN.

To send a spam VoIP call to every phone within an enterprise, an organization would have to purchase the Cisco Intercompany Media Engine and Cisco Unified Communications Manager and have called each phone number within the enterprise over the PSTN and completed each call successfully. Only then can it launch a VoIP call to each number.

The Cisco Intercompany Media Engine server creates tickets and the ASA validates them. The ASA and Cisco Intercompany Media Engine server share a password that is configured so that the ASA detects the ticket was created by a trusted Cisco Intercompany Media Engine server. The ticket contains information that indicates that the enterprise is authorized to call specific phone numbers at the target enterprise. See Figure 20-1 for the ticket verification process and how it operates between the originating and terminating-call enterprises.

Note Because the initial calls are over the PSTN, they are subject to any national regulations regarding telemarketing calling. For example, within the United States, they would be subject to the national do-not-call registry.

Figure 20-1 Ticket Verification Process with Cisco Intercompany Media Engine


Enterprise A
UC-IME Server
Cisco UCM
ASA
IP
IP
1
Enterprise B gets authorization ticket from A at end of validation protocol
Internet
4
ASA validates ticket
3
Enterprise B calls A and includes ticket
Enterprise B
2
UC-IME server passes ticket to UCM and it's stored as part of VoIP route
UC-IME Server
Cisco UCM
ASA
IP
IP
248761

As illustrated in Figure 20-1. Enterprise B makes a PSTN call to enterprise A. That call completes successfully. Later, Enterprise B Cisco Intercompany Media Engine server initiates validation procedures with Enterprise A. These validation procedures succeed. During the validation handshake, Enterprise B sends Enterprise A its domain name. Enterprise A verifies that this domain name is not on the blacklisted set of domains. Assuming it is not, Enterprise A creates a ticket.

Subsequently, someone in Enterprise B calls that number again. That call setup message from Enterprise B to Enterprise A includes the ticket in the X-Cisco-UC-IME-Ticket header field in the SIP INVITE message. This message arrives at the Enterprise A ASA. The ASA verifies the signature and computes several checks on the ticket to make sure it is valid. If the ticket is valid, the ASA forwards the request to Cisco UCM (including the ticket). Because the ASA drops requests that lack a valid ticket, unauthorized calls are never received by Cisco UCM.

The ticket password is a 128 bit random key, which can be thought of as a shared password between the adaptive security appliance and the Cisco Intercompany Media Engine server. This password is generated by the Cisco Intercompany Media Engine server and is used by a Cisco Intercompany Media Engine SIP trunk to generate a ticket to allow a call to be made between Cisco Intercompany Media Engine SIP trunks. A ticket is a signed object that contains a number of fields that grant permission to the calling domain to make a Cisco Intercompany Media Engine call to a specific number. The ticket is signed by the ticket password.

The Cisco Intercompany Media Engine also required that you configure an epoch for the password. The epoch contains an integer that updates each time that the password is changed. When the proxy is configured the first time and a password entered for the first time, enter 1 for the epoch integer. Each time you change the password, increment the epoch to indicate the new password. You must increment the epoch value each time your change the password.

Typically, you increment the epoch sequentially; however, the ASA allows you to choose any value when you update the epoch. If you change the epoch value, the tickets in use at remote enterprises become invalid. The incoming calls from the remote enterprises fallback to the PSTN until the terminating enterprise reissues tickets with the new epoch value and password.

The epoch and password that you configure on the ASA must match the epoch and password configured on the Cisco Intercompany Media Engine server. If you change the password or epoch on the ASA, you must update them on the Cisco Intercompany Media Engine server. See the Cisco Intercompany Media Engine server documentation for information.

Call Fallback to the PSTN

Cisco Intercompany Media Engine provides features that manage the QoS on the Internet, such as the ability to monitor QoS of the RTP traffic in real-time and fallback to PSTN automatically if problems arise. Call fallback from Internet VoIP calls to the public switched telephone network (PSTN) can occur for two reasons changes in connection quality and signal failure for the Cisco Intercompany Media Engine.

Internet connections can vary wildly in their quality and vary over time. Therefore, even if a call is sent over VoIP because the quality of the connection was good, the connection quality might worsen mid-call. To ensure an overall good experience for the end user, Cisco Intercompany Media Engine attempts to perform a mid-call fallback.

Performing a mid-call fallback requires the adaptive security appliance to monitor the RTP packets coming from the Internet and send information into an RTP Monitoring Algorithm (RMA) API, which will indicates to the adaptive security appliance whether fallback is required. If fallback is required, the adaptive security appliance sends a REFER message to Cisco UCM to tell it that it needs to fallback the call to PSTN.

The TLS signaling connections from the Cisco UCM are terminated on the adaptive security appliance and a TCP or TLS connection is initiated to the Cisco UCM. SRTP (media) sent from external IP phones to the internal network IP phone via the adaptive security appliance is converted to RTP. The adaptive security appliance inserts itself into the media path by modifying the SIP signaling messages that are sent over the SIP trunk between Cisco UCMs. TLS (signaling) and SRTP are always terminated on the adaptive security appliance.

If signaling problems occur, the call falls back to the PSTN; however, the Cisco UCM initiates the PSTN fall back and the adaptive security appliance does not send REFER message.

Architecture and Deployment Scenarios for Cisco Intercompany Media Engine

This section includes the following topics:

- Architecture, page 20-5
- Basic Deployment, page 20-6
- Off Path Deployment, page 20-7

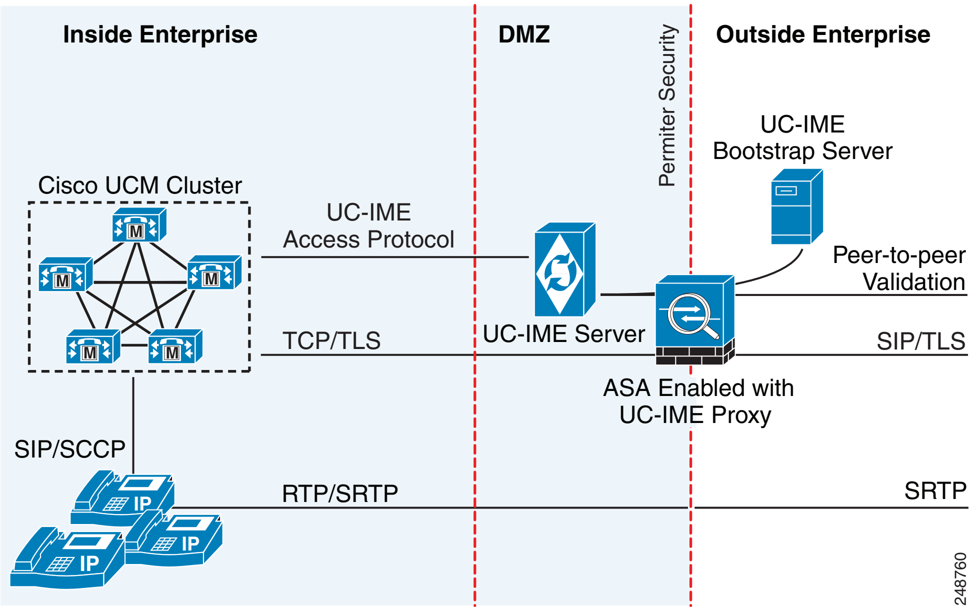
Architecture

Within the enterprise, Cisco Intercompany Media Engine is deployed with the following components for the following purposes:

- The adaptive security appliance—Enabled with the Cisco Intercompany Media Engine Proxy, provides perimeter security functions and inspects SIP signaling between SIP trunks.
- Cisco Intercompany Media Engine (UC-IME) server— Located in the DMZ, provides an automated provisioning service by learning new VoIP routes to particular phone numbers, and recording those routes in Cisco UCM. The Cisco Intercompany Media Engine server does not perform call control.
- Cisco Unified Communications Manager (Cisco UCM)—Responsible for call control and processing. Cisco UCM connects to the Cisco Intercompany Media Engine server by using the Access Protocol to publish and exchange updates. The architecture can consist of a single Cisco UCM or a Cisco UCM cluster within the enterprise.
- Cisco Intercompany Media Engine (UC-IME) Bootstrap server—Provides a certificate required admission onto the public peer-to-peer network for Cisco Intercompany Media Engine.

Figure 20-2 illustrates the components of the Cisco Intercompany Media Engine in a basic deployment.

Figure 20-2 Cisco Intercompany Media Engine Architecture in a Basic Deployment

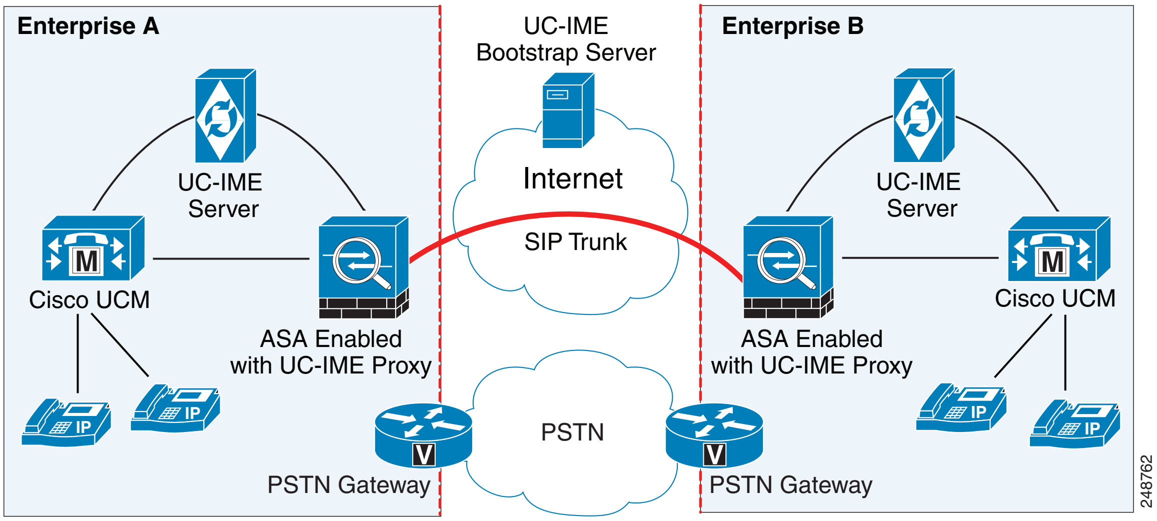


Basic Deployment

In a basic deployment, the Cisco Intercompany Media Engine Proxy sits in-line with the Internet firewall such that all Internet traffic traverses the adaptive security appliance. In this deployment, a single Cisco UCM or a Cisco UCM cluster is centrally deployed within the enterprise, along with a Cisco Intercompany Media Engine server (and perhaps a backup).

As shown in Figure 20-3, the adaptive security appliance sits on the edge of the enterprise and inspects SIP signaling by creating dynamic SIP trunks between enterprises.

Figure 20-3 Basic Deployment Scenario



Off Path Deployment

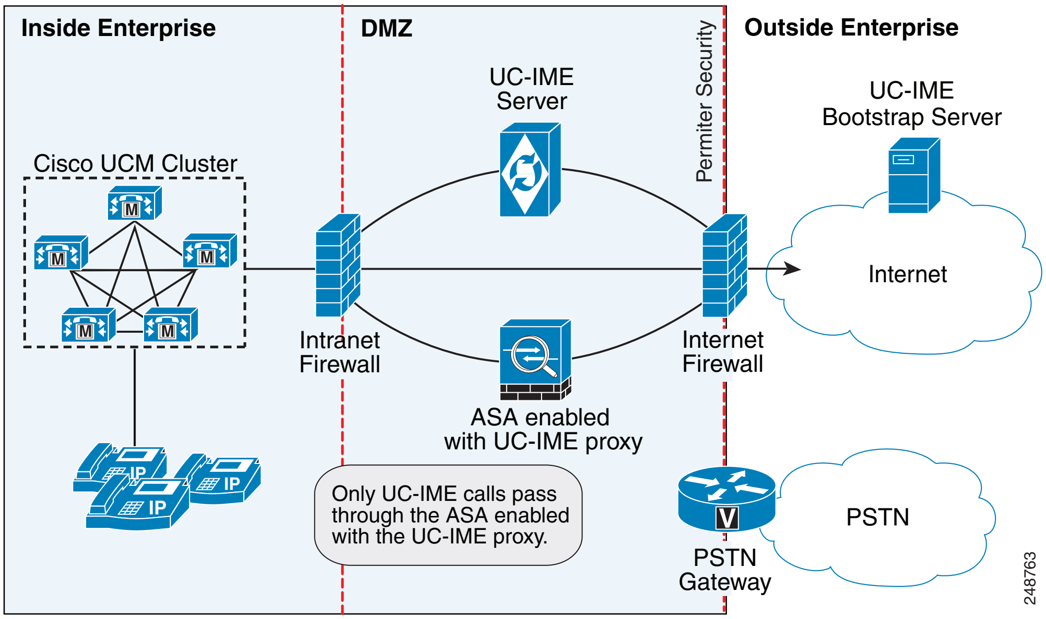
In an off path deployment, inbound and outbound Cisco Intercompany Media Engine calls pass through an adaptive security appliance enabled with the Cisco Intercompany Media Engine Proxy. The adaptive security appliance is located in the DMZ and is configured to support only the Cisco Intercompany Media Engine traffic (SIP signaling and RTP traffic). Normal Internet facing traffic does not flow through this adaptive security appliance.

For all inbound calls, the signaling is directed to the adaptive security appliance because destined Cisco UCMs are configured with the global IP address on the adaptive security appliance. For outbound calls, the called party could be any IP address on the Internet; therefore, the adaptive security appliance is configured with a mapping service that dynamically provides an internal IP address on the adaptive security appliance for each global IP address of the called party on the Internet.

Cisco UCM sends all outbound calls directly to the mapped internal IP address on the adaptive security appliance instead of the global IP address of the called party on the Internet. The adaptive security appliance then forwards the calls to the global IP address of the called party.

Figure 20-4 illustrates the architecture of the Cisco Intercompany Media Engine in an off path deployment.

Figure 20-4 Off Path Deployment of the Adaptive Security Appliance



Licensing for Cisco Intercompany Media Engine

The Cisco Intercompany Media Engine feature supported by the ASA require a Unified Communications Proxy license.

The following table shows the details of the Unified Communications Proxy license:

Note This feature is not available on No Payload Encryption models.

Model	License Requirement
All models	Intercompany Media Engine license. When you enable the Intercompany Media Engine (IME) license, you can use TLS proxy sessions up to the configured TLS proxy limit. If you also have a Unified Communications (UC) license installed that is higher than the default TLS proxy limit, then the ASA sets the limit to be the UC license limit plus an additional number of sessions depending on your model. You can manually configure the TLS proxy limit using the **tls-proxy maximum-sessions** command. To view the limits of your model, enter the **tls-proxy maximum-sessions ?** command. If you also install the UC license, then the TLS proxy sessions available for UC are also available for IME sessions. For example, if the configured limit is 1000 TLS proxy sessions, and you purchase a 750-session UC license, then the first 250 IME sessions do not affect the sessions available for UC. If you need more than 250 sessions for IME, then the remaining 750 sessions of the platform limit are used on a first-come, first-served basis by UC and IME. • For a license part number ending in “K8”, TLS proxy sessions are limited to 1000. • For a license part number ending in “K9”, the TLS proxy limit depends on your configuration and the platform model. **Note** K8 and K9 refer to whether the license is restricted for export: K8 is unrestricted, and K9 is restricted. You might also use SRTP encryption sessions for your connections: • For a K8 license, SRTP sessions are limited to 250. • For a K9 license, there is no limit. **Note** Only calls that require encryption/decryption for media are counted toward the SRTP limit; if passthrough is set for the call, even if both legs are SRTP, they do not count toward the limit.

For more information about licensing, see Chapter 5, “Managing Feature Licenses,” in the general operations configuration guide.

Guidelines and Limitations

Context Mode Guidelines

Supported in single context mode only.

Firewall Mode Guidelines

Supported in routed firewall mode only.

IPv6 Guidelines

Does not support IPv6 addresses.

Additional Guidelines and Limitations

Cisco Intercompany Media Engine has the following limitations:

- Fax is not supported. Fax capability needs to be disabled on the SIP trunk.

- Stateful failover of Cisco Unified Intercompany Media Engine is not supported. During failover, existing calls traversing the Cisco Intercompany Media Engine Proxy disconnect; however, new calls successfully traverse the proxy after the failover completes.
- Having Cisco UCMs on more than one of the ASA interfaces is not supported with the Cisco Intercompany Media Engine Proxy. Having the Cisco UCMs on one trusted interface is especially necessary in an off path deployment because the ASA requires that you specify the listening interface for the mapping service and the Cisco UCMs must be connected on one trusted interface.
- Multipart MIME is not supported.
- Only existing SIP features and messages are supported.
- H.264 is not supported.
- RTCP is not supported. The ASA drops any RTCP traffic sent from the inside interface to the outside interface. The ASA does not convert RTCP traffic from the inside interface into SRTP traffic.
- The Cisco Intercompany Media Engine Proxy configured on the ASA creates a dynamic SIP trunk for each connection to a remote enterprise. However, you cannot configure a unique subject name for each SIP trunk. The Cisco Intercompany Media Engine Proxy can have only one subject name configured for the proxy.

  Additionally, the subject DN you configure for the Cisco Intercompany Media Engine Proxy match the domain name that has been set for the local Cisco UCM.
- If a service policy rule for the Cisco Intercompany Media Engine Proxy is removed (by using the no service policy command) and reconfigured, the first call traversing the ASA will fail. The call fails over to the PSTN because the Cisco UCM does not know the connections are cleared and tries to use the recently cleared IME SIP trunk for the signaling.

  To resolve this issue, you must additionally enter the **clear connection all** command and restart the ASA. If the failure is due to failover, the connections from the primary ASA are not synchronized to the standby ASA.
- After the **clear connection all** command is issued on an ASA enabled with a UC-IME Proxy and the IME call fails over to the PSTN, the next IME call between an originating and terminating SCCP IP phone completes but does not have audio and is dropped after the signaling session is established.

  An IME call between SCCP IP phones use the IME SIP trunk in both directions. Namely, the signaling from the calling to called party uses the IME SIP trunk. Then, the called party uses the reverse IME SIP trunk for the return signaling and media exchange. However, this connection is already cleared on the ASA, which causes the IME call to fail.

  The next IME call (the third call after the **clear connection all** command is issued), will be completely successful.

  Note This limitation does not apply when the originating and terminating IP phones are configured with SIP.
- The ASA must be licensed and configured with enough TLS proxy sessions to handle the IME call volume. See "Licensing for Cisco Intercompany Media Engine" section on page 20-7 for information about the licensing requirements for TLS proxy sessions.

  This limitation occurs because an IME call cannot fall back to the PSTN when there are not enough TLS proxy sessions left to complete the IME call. An IME call between two SCCP IP phones requires the ASA to use two TLS proxy sessions to successfully complete the TLS handshake.

Assume for example, the ASA is configured to have a maximum of 100 TLS proxy sessions and IME calls between SCCP IP phones establish 101 TLS proxy sessions. In this example, the next IME call is initiated successfully by the originating SCCP IP phone but fails after the call is accepted by the terminating SCCP IP phone. The terminating IP phone rings and on answering the call, the call hangs due to an incomplete TLS handshake. The call does not fall back to the PSTN.

Configuring Cisco Intercompany Media Engine Proxy

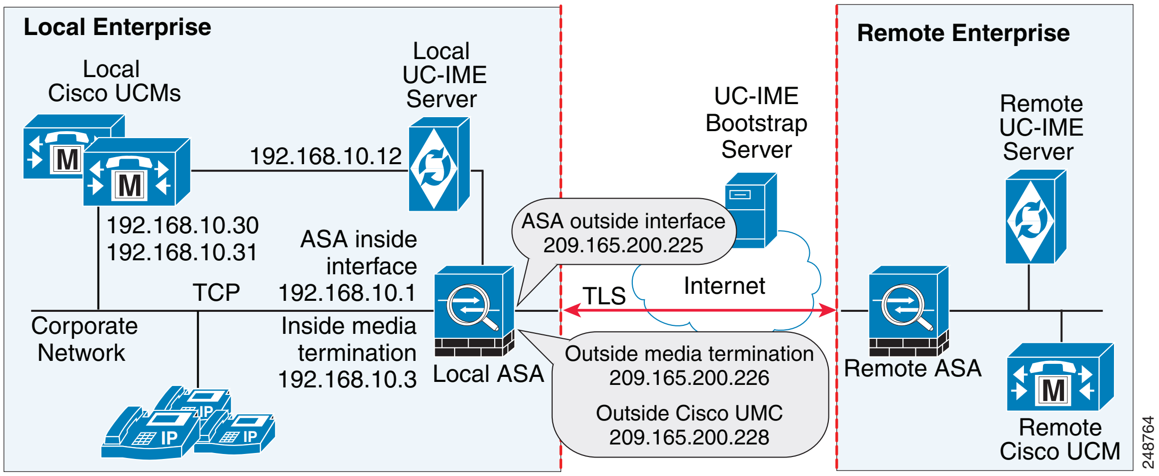
This section contains the following topics:

- Task Flow for Configuring Cisco Intercompany Media Engine, page 20-10
- Configuring NAT for Cisco Intercompany Media Engine Proxy, page 20-11
- Configuring PAT for the Cisco UCM Server, page 20-13
- Creating ACLs for Cisco Intercompany Media Engine Proxy, page 20-15
- Creating the Media Termination Instance, page 20-16
- Creating the Cisco Intercompany Media Engine Proxy, page 20-17
- Creating Trustpoints and Generating Certificates, page 20-20
- Creating the TLS Proxy, page 20-23
- Enabling SIP Inspection for the Cisco Intercompany Media Engine Proxy, page 20-24
- (Optional) Configuring TLS within the Local Enterprise, page 20-26
- (Optional) Configuring Off Path Signaling, page 20-29

Task Flow for Configuring Cisco Intercompany Media Engine

Figure 20-5 provides an example for a basic deployment of the Cisco Intercompany Media Engine. The following tasks include command line examples based on Figure 20-5.

Figure 20-5 Example for Basic (in-line) Deployment Tasks



Note Step 1 through Step 8 apply to both basic (in-line) and off path deployments and Step 9 applies only to off path deployment.

To configure a Cisco Intercompany Media Engine for a basic deployment, perform the following tasks.

Step 1 Configure static NAT for Cisco UCM. See Configuring NAT for Cisco Intercompany Media Engine Proxy, page 20-11.

Or

Configure PAT for the UCM server. See Configuring PAT for the Cisco UCM Server, page 20-13.

Step 2 Create ACLs for Cisco Intercompany Media Engine Proxy. See Creating ACLs for Cisco Intercompany Media Engine Proxy, page 20-15.

Step 3 Create the media termination address instance for Cisco Intercompany Media Engine Proxy. See Creating the Media Termination Instance, page 20-16.

Step 4 Create the Cisco Intercompany Media Engine Proxy. See Creating the Cisco Intercompany Media Engine Proxy, page 20-17.

Step 5 Create trustpoints and generate certificates for the Cisco Intercompany Media Engine Proxy. See Creating Trustpoints and Generating Certificates, page 20-20.

Step 6 Create the TLS proxy. See Creating the TLS Proxy, page 20-23.

Step 7 Configure SIP inspection for the Cisco Intercompany Media Engine Proxy. See Enabling SIP Inspection for the Cisco Intercompany Media Engine Proxy, page 20-24.

Step 8 (Optional) Configure TLS within the enterprise. See (Optional) Configuring TLS within the Local Enterprise, page 20-26.

Step 9 (Optional) Configure off path signaling. See (Optional) Configuring Off Path Signaling, page 20-29.

Note You only perform Step 9 when you are configuring the Cisco Intercompany Media Engine Proxy in an off path deployment.

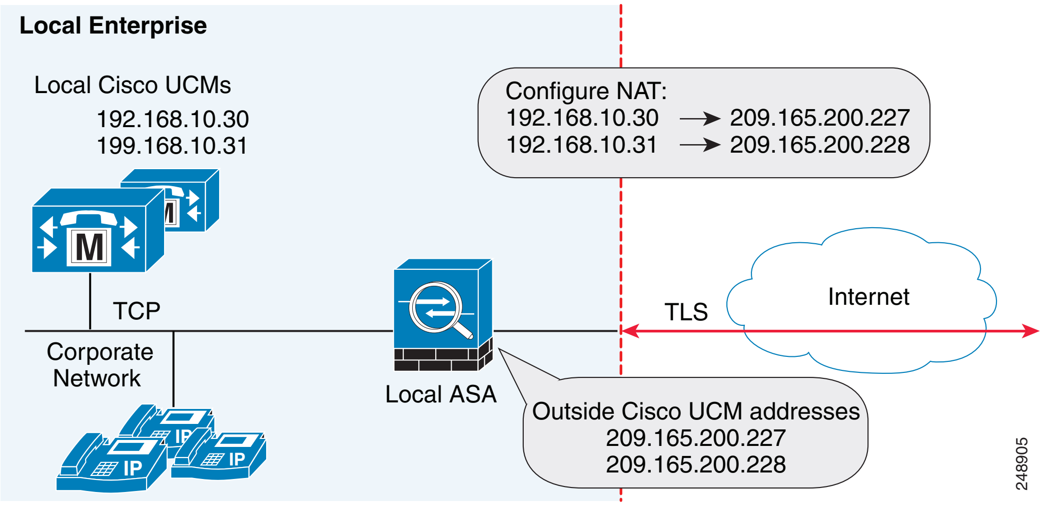
Configuring NAT for Cisco Intercompany Media Engine Proxy

To configure auto NAT, you first configure an object; then use the **nat** command in the object configuration mode.

The example command lines in this task are based on a basic (in-line) deployment. See Figure 20-5 on page 20-10 for an illustration explaining the example command lines in this task.

Alternatively, you can configure PAT for the Cisco Intercompany Media Engine Proxy. See Configuring PAT for the Cisco UCM Server, page 20-13.

Figure 20-6 Example for Configuring NAT for a Deployment



To configure auto NAT rules for the Cisco UCM server, perform the following steps:

	Command	Purpose

Step 1	`hostname(config)#` **`object network`** *`name`*
	`Examples:`
	`hostname(config)# object network ucm_real_192.168.10.30`
	`hostname(config)# object network ucm_real_192.168.10.31`
		Configures a network object for the real address of Cisco UCM that you want to translate.

Step 2	`hostname(config-network-object)#` **`host`** *`ip_address`*
	`Examples:`
	`hostname(config-network-object)# host 192.168.10.30`
	`hostname(config-network-object)# host 192.168.10.31`
		Specifies the real IP address of the Cisco UCM host for the network object.

Step 3	(Optional)
	`hostname(config-network-object)#` **`description`** *`string`*
	`Example:`
	`hostname(config-network-object)# description "Cisco UCM Real Address"`
		Provides a description of the network object.

Step 4	`hostname(config-network-object)#` **`exit`**	Exits from the objects configuration mode.

Step 5	`hostname(config)#` **`object network`** *`name`*
	`Example:`
	`hostname(config)# object network ucm_map_209.165.200.228`
		Configures a network object for the mapped address of the Cisco UCM.

Step 6	`hostname(config-network-object)#` **`host`** *`ip_address`*
	`Example:`
	`hostname(config-network-object)# host 209.165.200.228`
		Specifies the mapped IP address of the Cisco UCM host for the network object.

Step 7	(Optional)
	`hostname(config-network-object)#` **`description`** *`string`*
	`Example:`
	`hostname(config-network-object)# description "Cisco UCM Mapped Address"`
		Provides a description of the network object.

	Command	Purpose
Step 8	`hostname(config-network-object)# exit`	Exits from the objects configuration mode.

Step 9	`hostname(config)# nat (inside,outside) source static`
	`real_obj mapped_obj`
	Examples:
	`hostname(config)# nat (inside,outside) source static`
	`ucm_real_192.168.10.30 ucm_209.165.200.228`
	`hostname(config)# nat (inside,outside) source static`
	`ucm_real_192.168.10.31 ucm_209.165.200.228`
		Specifies the address translation on the network objects created in this procedure.
		Where *real_obj* is the *name* that you created in Step 1 in this task.
		Where *mapped_obj* is the *name* that you created in Step 5 in this task.

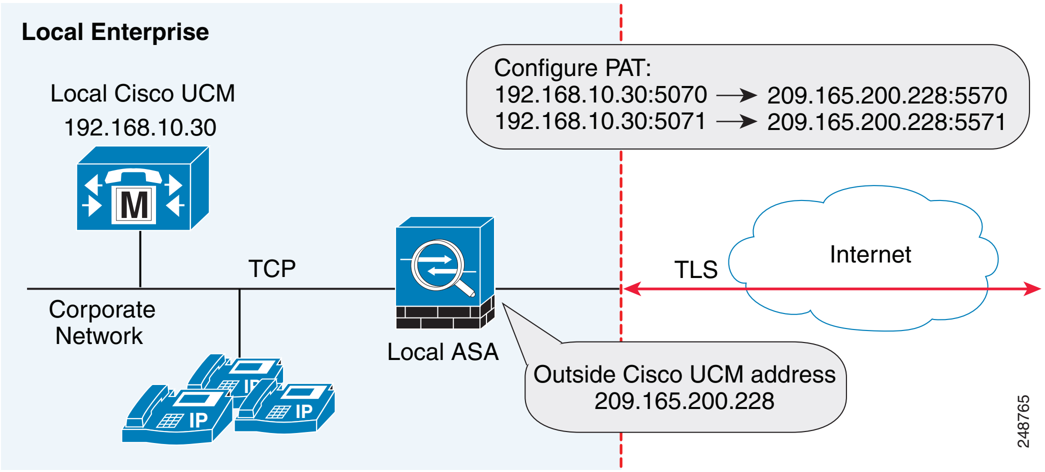
What to Do Next

Create the ACLs for the Cisco Intercompany Media Engine Proxy. See Creating ACLs for Cisco Intercompany Media Engine Proxy, page 20-15.

Configuring PAT for the Cisco UCM Server

Perform this task as an alternative to configuring NAT for the Cisco Intercompany Media Engine Proxy.

Figure 20-7 Example for Configuring PAT for a Deployment



Note You only perform this step when NAT is not configured for the Cisco UCM server.

To configure PAT for the Cisco UCM server, perform the following steps:

	Command	Purpose
Step 1	hostname(config)# **object network** *name* **Examples:** hostname(config)# object network ucm-pat-209.165.200.228	Configures a network object for the outside IP address of Cisco UCM that you want to translate.
Step 2	hostname(config-network-object)# **host** *ip_address* **Example:** hostname(config-network-object)# host 209.165.200.228	Specifies the real IP address of the Cisco UCM host for the network object.
Step 3	hostname(config-network-object)# **exit**	Exits from the objects configuration mode.
Step 4	hostname(config)# **object service** *name* **Examples:** hostname(config)# object service tcp_5070 hostname(config)# object service tcp_5071	Creates a service object for the outside Cisco Intercompany Media Engine port.
Step 5	hostname(config-service-object)# **tcp source eq** *port* **Examples:** hostname(config-service-object)# tcp source eq 5070 hostname(config-service-object)# tcp source eq 5071	Specifies the port number.
Step 6	hostname(config-service-object)# **exit**	Exits from the objects configuration mode.
Step 7	hostname(config)# **object network** *name* **Examples:** hostname(config)# object network ucm-real-192.168.10.30 hostname(config)# object network ucm-real-192.168.10.31	Configures a network object to represent the real IP address of Cisco UCM.
Step 8	hostname(config-network-object)# **host** *ip_address* **Examples:** hostname(config-network-object)# host 192.168.10.30 hostname(config-network-object)# host 192.168.10.31	Specifies the real IP address of the Cisco UCM host for the network object.
Step 9	hostname(config-network-object)# **exit**	Exits from the objects configuration mode.
Step 10	hostname(config)# **object service** *name* **Examples:** hostname(config)# object service tcp_5570 hostname(config)# object service tcp_5571	Creates a service objects for Cisco UCM SIP port.
Step 11	hostname(config-service-object)# **tcp source eq** *port* **Example:** hostname(config-service-object)# tcp source eq 5570 hostname(config-service-object)# tcp source eq 5571	Specifies the port number.
Step 12	hostname(config-service-object)# **exit**	Exits from the objects configuration mode.
Step 13	hostname(config)# **nat (inside,outside) source static** *real_obj mapped_obj* **service** *real_port mapped_port* **Examples:** hostname(config)# nat (inside,outside) source static ucm-real-192.168.10.30 ucm-pat-209.165.200.228 service tcp_5070 tcp_5570 hostname(config)# nat (inside,outside) source static ucm-real-192.168.10.31 ucm-pat-128.106.254.5 service tcp_5071 tcp_5571	Creates a static mapping for Cisco UCM. *Where real_obj is the name that you created in Step 1 in this task.* *Where mapped_obj is the name that you created in Step 7 in this task.* *Where real_port is the name that you created in Step 4 in this task.* *Where mapped_obj is the name that you created in Step 10 in this task.*

Creating ACLs for Cisco Intercompany Media Engine Proxy

To configure ACLs for the Cisco Intercompany Media Engine Proxy to reach the Cisco UCM server, perform the following steps.

The example command lines in this task are based on a basic (in-line) deployment. See Figure 20-5 on page 20-10 for an illustration explaining the example command lines in this task.

	Command	Purpose
Step 1	hostname(config)# **access-list** *id* **extended permit tcp any host** *ip_address* **eq** *port* **Example:** hostname(config)# access-list incoming extended permit tcp any host 192.168.10.30 eq 5070	Adds an Access Control Entry (ACE). An ACL is made up of one or more ACEs with the same ACL ID. This ACE provides access control by allowing incoming access for Cisco Intercompany Media Engine connections on the specified port. In the *ip_address* argument, provide the real IP address of Cisco UCM.
Step 2	hostname(config)# **access-group** *access-list* **in interface** *interface_name* **Example:** hostname(config)# access-group incoming in interface outside	Binds the ACL to an interface.
Step 3	hostname(config)# **access-list** *id* **extended permit tcp any host** *ip_address* **eq** *port* **Example:** hostname(config)# access-list ime-inbound-sip extended permit tcp any host 192.168.10.30 eq 5070	Adds an ACE. This ACE allows the ASA to allow inbound SIP traffic for Cisco Intercompany Media Engine. This entry is used to classify traffic for the class and policy map. **Note** The port that you configure here must match the trunk settings configured on Cisco UCM. See the Cisco Unified Communications Manager documentation for information about this configuration setting.
Step 4	hostname(config)# **access-list** *id* **extended permit tcp** *ip_address mask* **any range** *range* **Example:** hostname(config)# access-list ime-outbound-sip extended permit tcp 192.168.10.30 255.255.255.255 any range 5000 6000	Adds an ACE. This ACE allows the ASA to allow outbound SIP traffic for Cisco Intercompany Media Engine (in the example, any TCP traffic with source as 192.168.10.30 and destination port range between 5000 and 6000). This entry is used to classify traffic for the class and policy map. **Note** Ensure that TCP traffic between Cisco UCM and the Cisco Intercompany Media Engine server does not use this port range (if that connection goes through the ASA).
Step 5	hostname(config)# **access-list** *id* **permit tcp any host** *ip_address* **eq 6084** **Example:** hostname(config)# access-list ime-traffic permit tcp any host 192.168.10.12 eq 6084	Adds an ACE. This ACE allows the ASA to allow traffic from the Cisco Intercompany Media Engine server to remote Cisco Intercompany Media Engine servers.
Step 6	hostname(config)# **access-list** *id* **permit tcp any host** *ip_address* **eq 8470** **Example:** hostname(config)# access-list ime-bootserver-traffic permit tcp any host 192.168.10.12 eq 8470	Adds an ACE. This ACE allows the ASA to allow traffic from the Cisco Intercompany Media Engine server to the Bootstrap server for the Cisco Intercompany Media Engine.

What to Do Next

Create the media termination instance on the ASA for the Cisco Intercompany Media Engine Proxy. See Creating the Media Termination Instance, page 20-16.

Creating the Media Termination Instance

Guidelines

The media termination address you configure must meet these requirements:

- If you decide to configure a media-termination address on interfaces (rather than using a global interface), you must configure a media-termination address on at least two interfaces (the inside and an outside interface) before applying the service policy for the Cisco Intercompany Media Engine Proxy. Otherwise, you will receive an error message when enabling the proxy with SIP inspection.

Note Cisco recommends that you configure the media-termination address for the Cisco Intercompany Media Engine Proxy on interfaces rather than configuring a global media-termination address.

- The Cisco Intercompany Media Engine Proxy can use only one type of media termination instance at a time; for example, you can configure a global media-termination address for all interfaces or configure a media-termination address for different interfaces. However, you cannot use a global media-termination address and media-termination addresses configured for each interface at the same time.

Note If you change any Cisco Intercompany Media Engine Proxy settings after you create the media-termination address for the proxy, you must reconfigure the media-termination address by using the **no media-termination** command, and then reconfiguring it as described in this procedure.

Procedure

Create the media termination instance to use with the Cisco Intercompany Media Engine Proxy.

The example command lines in this task are based on a basic (in-line) deployment. See Figure 20-5 on page 20-10 for an illustration explaining the example command lines in this task.

To create the media termination instance for the Cisco Intercompany Media Engine Proxy, perform the following steps:

	Command	Purpose
Step 1	hostname(config)# **media-termination** *instance_name* **Example:** hostname(config)# **media-termination** *uc-ime-media-term*	Creates the media termination instance that you attach to the Cisco Intercompany Media Engine Proxy.
Step 2	hostname(config-media-termination)# **address** *ip_address* **interface** *intf_name* **Examples:** hostname(config-media-termination)# address 209.165.200.228 interface outside	Configures the media-termination address used by the outside interface of the ASA. The outside IP address must be a publicly routable address that is an unused IP address within the address range on that interface. See Creating the Cisco Intercompany Media Engine Proxy, page 20-17 for information about the UC-IME proxy settings. See CLI configuration guide for information about the **no service-policy** command.
Step 3	hostname(config-media-termination)# **address** *ip_address* **interface** *intf_name* **Examples:** hostname(config-media-termination)# address 192.168.10.3 interface inside	Configures a media termination address used by the inside interface of the ASA. **Note** The IP address must be an unused IP address within the same subnet on that interface.
Step 4	(Optional) hostname(config-media-termination)# ***rtp-min-port*** *port1* ***rtp-maxport*** *port2* **Examples:** hostname(config-media-termination)# rtp-min-port 1000 rtp-maxport 2000	Configures the rtp-min-port and rtp-max-port limits for the Cisco Intercompany Media Engine Proxy. Configure the RTP port range for the media termination point when you need to scale the number of calls that the Cisco Intercompany Media Engine supports. Where *port1* specifies the minimum value for the RTP port range for the media termination point, where port1 can be a value from 1024 to 65535. By default, the value for *port1* is 16384. Where *port2* specifies the maximum value for the RTP port range for the media termination point, where port2 can be a value from 1024 to 65535. By default, the value for *port2* is 32767.

What To Do Next

Once you have created the media termination instance, create the Cisco Intercompany Media Engine Proxy. See Creating the Cisco Intercompany Media Engine Proxy, page 20-17.

Creating the Cisco Intercompany Media Engine Proxy

To create the Cisco Intercompany Media Engine Proxy, perform the following steps.

The example command lines in this task are based on a basic (in-line) deployment. See Figure 20-5 on page 20-10 for an illustration explaining the example command lines in this task.

Note You cannot change any of the configuration settings for the Cisco Intercompany Media Engine Proxy described in this procedure when the proxy is enabled for SIP inspection. Remove the Cisco Intercompany Media Engine Proxy from SIP inspection before changing any of the settings described in this procedure.

	Command	Purpose

Step 1	`hostname(config)#` **`uc-ime`** *`uc_ime_name`*
	`Example:`
	`hostname(config)# uc-ime local-ent-ime`
		Configures the Cisco Intercompany Media Engine Proxy.
		Where *`uc_ime_name`* is the name of the Cisco Intercompany Media Engine Proxy. The name is limited to 64 characters.
		Only one Cisco Intercompany Media Engine Proxy can be configured on the ASA.

Step 2	`hostname(config-uc-ime)#` **`media-termination`** *`mta_instance_name`*
	`Example:`
	`hostname(config-uc-ime)# media-termination ime-media-term`
		Specifies the media termination instance used by the Cisco Intercompany Media Engine Proxy.
		Note You must create the media termination instance before you specify it in the Cisco Intercompany Media Engine Proxy.
		Where *`mta_instance_name`* is the *`instance_name`* that you created in Step 1 of Creating the Media Termination Instance.
		See Creating the Media Termination Instance, page 20-16 for the steps to create the media termination instance.

Step 3	`hostname(config-uc-ime)#` **`ucm address`** *`ip_address`* **`trunk-security-mode`** `[`**`nonsecure`** `|` **`secure`**`]`
	`Example:`
	`hostname(config-uc-ime)# ucm address 192.168.10.30 trunk-security-mode non-secure`
		Specifies the Cisco UCM server in the enterprise. You must specify the real IP address of the Cisco UCM server. Do not specify a mapped IP address for the server.
		Note You must include an entry for each Cisco UCM in the cluster with Cisco Intercompany Media Engine that has a SIP trunk enabled.
		Where the **nonsecure** and **secure** options specify the security mode of the Cisco UCM or cluster of Cisco UCMs.
		Note Specifying **secure** for Cisco UCM or Cisco UCM cluster indicates that Cisco UCM or Cisco UCM cluster is initiating TLS; therefore, you must configure TLS for components. See (Optional) Configuring TLS within the Local Enterprise, page 20-26.
		You can specify the **secure** option in this task or you can update it later while configuring TLS for the enterprise. See Step 11 in (Optional) Configuring TLS within the Local Enterprise, page 20-26.

	Command	Purpose
Step 4	hostname(config-uc-ime)# **ticket epoch** *n* **password** *password* **Example:** hostname(config-uc-ime)# ticket epoch 1 password password1234	Configures the ticket epoch and password for Cisco Intercompany Media Engine. Where *n* is an integer from 1-255. The epoch contains an integer that updates each time that the password is changed. When the proxy is configured the first time and a password entered for the first time, enter 1 for the epoch integer. Each time you change the password, increment the epoch to indicate the new password. You must increment the epoch value each time your change the password. Typically, you increment the epoch sequentially; however, the ASA allows you to choose any value when you update the epoch. If you change the epoch value, the current password is invalidated and you must enter a new password. Where *password* contains a minimum of 10 and a maximum of 64 printable character from the US-ASCII character set. The allowed characters include 0x21 to 0x73 inclusive, and exclude the space character. We recommend a password of at least 20 characters. Only one password can be configured at a time. The ticket password is stored onto flash. The output of the **show running-config uc-ime** command displays ***** instead of the password string. **Note** The epoch and password that you configure on the ASA must match the epoch and password configured on the Cisco Intercompany Media Engine server. See the Cisco Intercompany Media Engine server documentation for information.

	Command	Purpose
Step 5	(Optional) hostname(config-uc-ime)# **fallback monitoring timer** *timer_millisec* | **hold-down timer** *timer_sec* **Examples:** hostname(config-uc-ime)# fallback monitoring timer 120 hostname(config-uc-ime)# fallback hold-down timer 30	Specifies the fallback timers for Cisco Intercompany Media Engine. Specifying **monitoring timer** sets the time between which the ASA samples the RTP packets received from the Internet. The ASA uses the data sample to determine if fallback to the PSTN is needed for a call. Where *timer_millisec* specifies the length of the monitoring timer. By default, the length is 100 milliseconds for the monitoring timer and the allowed range is 10-600 ms. Specifying **hold-down timer** sets the amount of time that ASA waits before notifying Cisco UCM whether to fall back to PSTN. Where *timer_sec* specifies the length of the hold-down timer. By default, the length is 20 seconds for the hold-down timer and the allowed range is 10-360 seconds. If you do not use this command to specify fallback timers, the ASA uses the default settings for the fallback timers.
Step 6	(Optional) hostname(config-uc-ime)# **fallback sensitivity-file** *file_name* **Example:** hostname(config-uc-ime)# fallback sensitivity-file ime-fallback-sensitvity.fbs	Specifies the file to use for mid-call PSTN fallback. Where *file_name* must be the name of a file on disk that includes the .fbs file extension. The fallback file is used to determine whether the QoS of the call is poor enough for the Cisco Intercompany Media Engine to move the call to the PSTN.

What to Do Next

Install the certificate on the local entity truststore. You could also enroll the certificate with a local CA trusted by the local entity.

Creating Trustpoints and Generating Certificates

You need to generate the keypair for the certificate used by the ASA, and configure a trustpoint to identify the certificate sent by the ASA in the TLS handshake.

The example command lines in this task are based on a basic (in-line) deployment. See Figure 20-5 on page 20-10 for an illustration explaining the example command lines in this task.

Note This task instructs you on how to create trustpoints for the local enterprise and the remote enterprise and how to exchange certificates between these two enterprises. This task does not provide steps for creating trustpoints and exchanging certificates between the local Cisco UCM and the local ASA. However, if you require additional security within the local enterprise, you must perform the optional task (Optional)

Configuring TLS within the Local Enterprise, page 20-26. Performing that task allows for secure TLS connections between the local Cisco UCM and the local ASA. The instructions in that task describe how to create trustpoints between the local Cisco UCM and the local ASA.

Prerequisites for Installing Certificates

To create a proxy certificate on the ASA that is trusted by the remote entity, obtain a certificate from a trusted CA or export it from the remote enterprise ASA.

To export the certificate from the remote enterprise, you enter the following command on the remote ASA:

hostname(config)# **crypto ca export trustpoint identity-certificate**

The ASA prompts displays the certificate in the terminal screen. Copy the certificate from the terminal screen. You will need the certificate text in Step 5 of this task.

Procedure

To create the trustpoints and generate certificates, perform the following steps:

	Command	Purpose

Step 1	`hostname(config)#` **`crypto key generate rsa label`** *`key-pair-label`* **`modulus`** *`size`*
	`Example:`
	`hostname(config)# crypto key generate rsa label local-ent-key modulus 2048`
		On the local ASA, creates the RSA keypair that can be used for the trustpoints. This is the keypair and trustpoint for the local entities signed certificate.
		The modulus key size that you select depends on the level of security that you want to configure and on any limitations imposed by the CA from which you are obtaining the certificate. The larger the number that you select, the higher the security level will be for the certificate. Most CAs recommend 2048 for the key modulus size; however,
		Note GoDaddy requires a key modulus size of 2048.

Step 2	`hostname(config)#` **`crypto ca trustpoint`** *`trustpoint_name`*
	`Example:`
	`hostname(config)# crypto ca trustpoint local_ent`
		Enters the trustpoint configuration mode for the specified trustpoint so that you can create the trustpoint for the local entity.
		A trustpoint represents a CA identity and possibly a device identity, based on a certificate issued by the CA. Maximum name length is 128 characters.

Step 3	`hostname(config-ca-trustpoint)#` **`subject-name`** *`X.500_name`*
	`Example:`
	`hostname(config-ca-trustpoint)# subject-name cn=Ent-local-domain-name**`
		Includes the indicated subject DN in the certificate during enrollment.
		Note The domain name that you enter here must match the domain name that has been set for the local Cisco UCM. For information about how to configure the domain name for Cisco UCM, see the Cisco Unified Communications Manager documentation for information.

	Command	Purpose
Step 4	hostname(config-ca-trustpoint)# **keypair** *keyname* **Example:** hostname(config-ca-trustpoint)# keypair local-ent-key	Specifies the key pair whose public key is to be certified.
Step 5	hostname(config-ca-trustpoint)# **enroll terminal**	Specifies that you will use the "copy and paste" method of enrollment with this trustpoint (also known as manual enrollment).
Step 6	hostname(config-ca-trustpoint)# **exit**	Exits from the CA Trustpoint configuration mode.
Step 7	hostname(config)# **crypto ca enroll** *trustpoint* **Example:** hostname(config)# crypto ca enroll remote-ent % % Start certificate enrollment ... % The subject name in the certificate will be: % cn=enterpriseA % The fully-qualified domain name in the certificate will @ be: ciscoasa % Include the device serial number in the subject name? [yes/no]: no Display Certificate Request to terminal? [yes/no]: yes	Starts the enrollment process with the CA. Where *trustpoint* is the same as the value you entered for *trustpoint_name* in Step 2. When the trustpoint is configured for manual enrollment (**enroll terminal** command), the ASA writes a base-64-encoded PKCS10 certification request to the console and then displays the CLI prompt. Copy the text from the prompt. Submit the certificate request to the CA, for example, by pasting the text displayed at the prompt into the certificate signing request enrollment page on the CA website. When the CA returns the signed identity certificate, proceed to Step 8 in this procedure.
Step 8	hostname(config)# **crypto ca import** *trustpoint* **certificate** **Example:** hostname(config)# crypto ca import remote-ent certificate	Imports the signed certificate received from the CA in response to a manual enrollment request. Where *trustpoint* specifies the trustpoint you created in Step 2. The ASA prompts you to paste the base-64 formatted signed certificate onto the terminal.
Step 9	hostname(config)# **crypto ca authenticate** *trustpoint* **Example:** hostname(config)# crypto ca authenticate remote-ent	Authenticates the third-party identity certificate received from the CA. The identity certificate is associated with a trustpoint created for the remote enterprise. The ASA prompts you to paste the base-64 formatted identity certificate from the CA onto the terminal.

What to Do Next

Create the TLS proxy for the Cisco Intercompany Media Engine. See the "Creating the TLS Proxy" section on page 20-23.

Creating the TLS Proxy

Because either enterprise, namely the local or remote Cisco UCM servers, can initiate the TLS handshake (unlike IP Telephony or Cisco Mobility Advantage, where only the clients initiate the TLS handshake), you must configure by-directional TLS proxy rules. Each enterprise can have an ASA as the TLS proxy.

Create TLS proxy instances for the local and remote entity initiated connections respectively. The entity that initiates the TLS connection is in the role of "TLS client." Because the TLS proxy has a strict definition of "client" and "server" proxy, two TLS proxy instances must be defined if either of the entities could initiate the connection.

The example command lines in this task are based on a basic (in-line) deployment. See Figure 20-5 on page 20-10 for an illustration explaining the example command lines in this task.

To create the TLS proxy, perform the following steps:

	Command	Purpose

Step 1	`hostname(config)# tls-proxy proxy_name`
	Example:
	`hostname(config)# tls-proxy local_to_remote-ent`
		Creates the TLS proxy for the outbound connections.

Step 2	`hostname(config-tlsp)# client trust-point proxy_trustpoint`
	Example:
	`hostname(config-tlsp)# client trust-point local-ent`
		For **outbound** connections, specifies the trustpoint and associated certificate that the adaptive security appliance uses in the TLS handshake when the adaptive security appliance assumes the role of the TLS client. The certificate must be owned by the adaptive security appliance (identity certificate).
		Where *proxy_trustpoint* specifies the trustpoint defined by the **crypto ca trustpoint** command in Step 2 in "Creating Trustpoints and Generating Certificates" section on page 20-20.

Step 3	`hostname(config-tlsp)# client cipher-suite cipher_suite`
	Example:
	`hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1`
		For outbound connections, controls the TLS handshake parameter for the cipher suite.
		Where `cipher_suite` includes des-sha1, 3des-sha1, aes128-sha1, aes256-sha1, or null-sha1.
		For client proxy (the proxy acts as a TLS client to the server), the user-defined cipher suite replaces the default cipher suite, or the one defined by the **ssl encryption** command. Use this command to achieve difference ciphers between the two TLS sessions. You should use AES ciphers with the Cisco UCM server.

Step 4	`hostname(config-tlsp)# exit`	Exits from the TLS proxy configuration mode.

Step 5	`hostname(config)# tls-proxy proxy_name`
	Example:
	`hostname(config)# tls-proxy remote_to_local-ent`
		Create the TLS proxy for inbound connections.

	Command	Purpose
Step 6	hostname(config-tlsp)# **server trust-point** *proxy_trustpoint* **Example:** hostname(config-tlsp)# server trust-point local-ent	For **inbound** connections, specifies the proxy trustpoint certificate presented during TLS handshake. The certificate must be owned by the adaptive security appliance (identity certificate). Where *proxy_trustpoint* specifies the trustpoint defined by the **crypto ca trustpoint** command in Step 2 in "Creating Trustpoints and Generating Certificates" section on page 20-20. Because the TLS proxy has strict definition of client proxy and server proxy, two TLS proxy instances must be defined if either of the entities could initiate the connection.
Step 7	hostname(config-tlsp)# **client cipher-suite** *cipher_suite* **Example:** hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1	For inbound connections, controls the TLS handshake parameter for the cipher suite. Where *cipher_suite* includes des-sha1, 3des-sha1, aes128-sha1, aes256-sha1, or null-sha1.
Step 8	hostname(config-tlsp)# **exit**	Exits from the TSL proxy configuration mode.
Step 9	hostname(config)# **ssl encryption 3des-shal aes128-shal** [*algorithms*]	Specifies the encryption algorithms that the SSL/TLS protocol uses. Specifying the 3des-shal and aes128-shal is required. Specifying other algorithms is optional. **Note** The Cisco Intercompany Media Engine Proxy requires that you use strong encryption. You must specify this command when the proxy is licensed using a K9 license.

What to Do Next

Once you have created the TLS proxy, enable it for SIP inspection.

Enabling SIP Inspection for the Cisco Intercompany Media Engine Proxy

Enable the TLS proxy for SIP inspection and define policies for both entities that could initiate the connection.

The example command lines in this task are based on a basic (in-line) deployment. See Figure 20-5 on page 20-10 for an illustration explaining the example command lines in this task.

Note If you want to change any Cisco Intercompany Media Engine Proxy settings after you enable SIP inspection, you must enter the **no service-policy** command, and then reconfigure the service policy as described in this procedure. Removing and reconfiguring the service policy does not affect existing calls; however, the first call traversing the Cisco Intercompany Media Engine Proxy will fail. Enter the **clear connection** command and restart the ASA.

To enable SIP inspection for the Cisco Intercompany Media Engine Proxy, perform the following steps:

	Command	Purpose
Step 1	hostname(config)# **class-map** *class_map_name* **Examples:** hostname(config)# class-map ime-inbound-sip	Defines a class for the inbound Cisco Intercompany Media Engine SIP traffic.
Step 2	hostname(config-cmap)# **match access-list** *access_list_name* **Examples:** hostname(config-cmap)# match access-list ime-inbound-sip	Identifies the SIP traffic to inspect. Where the *access_list_name* is the ACL you created in Step 3, page 20-15 of the task Creating ACLs for Cisco Intercompany Media Engine Proxy.
Step 3	hostname(config-cmap)# **exit**	Exits from the class map configuration mode.
Step 4	hostname(config)# **class-map** *class_map_name* **Examples:** hostname(config)# class-map ime-outbound-sip	Defines a class for the outbound SIP traffic from Cisco Intercompany Media Engine.
Step 5	hostname(config)# **match access-list** *access_list_name* **Examples:** hostname(config-cmap)# match access-list ime-outbound-sip	Identifies which outbound SIP traffic to inspect. Where the *access_list_name* is the ACL you created in Step 4, page 20-15 of the task Creating ACLs for Cisco Intercompany Media Engine Proxy.
Step 6	hostname(config-cmap)# **exit**	Exits from the class map configuration mode.
Step 7	hostname(config)# **policy-map** *name* **Examples:** hostname(config)# policy-map ime-policy	Defines the policy map to which to attach the actions for the class of traffic.
Step 8	hostname(config-pmap)# **class** *classmap_name* **Examples:** hostname(config-pmap)# class ime-outbound-sip	Assigns a class map to the policy map so that you can assign actions to the class map traffic. Where *classmap_name* is the name of the SIP class map that you created in Step 1 in this task.
Step 9	hostname(config-pmap-c)# **inspect sip** [*sip_map*] **tls-proxy** *proxy_name* **uc-ime** *uc_ime_map* **Examples:** hostname(config-pmap-c)# inspect sip tls-proxy local_to_remote-ent uc-ime local-ent-ime	Enables the TLS proxy and Cisco Intercompany Media Engine Proxy for the specified SIP inspection session.
Step 10	hostname(config-cmap-c)# **exit**	Exits from the policy map class configuration mode.
Step 11	hostname(config-pmap)# **class** *class_map_name* **Examples:** hostname(config-pmap)# class ime-inbound-sip	Assigns a class map to the policy map so that you can assign actions to the class map traffic. Where *classmap_name* is the name of the SIP class map that you created in Step 4 in this task.
Step 12	hostname(config-pmap-c)# **inspect sip** [*sip_map*] **tls-proxy** *proxy_name* **uc-ime** *uc_ime_map* **Examples:** hostname(config-pmap-c)# inspect sip tls-proxy remote-to-local-ent uc-ime local-ent-ime	Enables the TLS proxy and Cisco Intercompany Media Engine Proxy for the specified SIP inspection session.
Step 13	hostname(config-pmap-c)# **exit**	Exits from the policy map class configuration mode.

	Command	Purpose
Step 14	`hostname(config-pmap)# `**`exit`**	Exits from the policy map configuration mode.

Step 15	`hostname(config)# `**`service-policy`** *`policymap_name`*
	`global`
	`Examples:`
	`hostname(config)# service-policy ime-policy global`
		Enables the service policy for SIP inspection for all interfaces.
		Where *`policymap_name`* is the name of the policy map you created in Step 7 of this task.
		See Creating the Cisco Intercompany Media Engine Proxy, page 20-17 for information about the UC-IME proxy settings. See CLI configuration guide for information about the **no service-policy** command.

What to Do Next

Once you have enabled the TLS proxy for SIP inspection, if necessary, configure TLS within the enterprise. See (Optional) Configuring TLS within the Local Enterprise, page 20-26.

(Optional) Configuring TLS within the Local Enterprise

This task is not required if TCP is allowable within the inside network.

TLS within the enterprise refers to the security status of the Cisco Intercompany Media Engine trunk as seen by the ASA.

Note If the transport security for the Cisco Intercompany Media Engine trunk changes on Cisco UCM, it must be changed on the ASA as well. A mismatch will result in call failure. The ASA does not support SRTP with non-secure IME trunks. The ASA assumes SRTP is allowed with secure trunks. So ‘SRTP Allowed’ must be checked for IME trunks if TLS is used. The ASA supports SRTP fallback to RTP for secure IME trunk calls.

Prerequisites

On the local Cisco UCM, download the Cisco UCM certificate. See the Cisco Unified Communications Manager documentation for information. You will need this certificate when performing Step 6 of this procedure.

Procedure

To configure TLS within the local enterprise, perform the following steps on the local ASA:

	Commands	Purpose
Step 1	hostname(config)# **crypto key generate rsa label** *key-pair-label* hostname(config)# **crypto ca trustpoint** *trustpoint_name* hostname(config-ca-trustpoint)# **enroll self** hostname(config-ca-trustpoint)# **keypair** *keyname* hostname(config-ca-trustpoint)# **subject-name** *x.500_name* **Example:** hostname(config)# crypto key generate rsa label local-ent-key hostname(config)# crypto ca trustpoint local-asa hostname(config-ca-trustpoint)# enroll self hostname(config-ca-trustpoint)# keypair key-local-asa hostname(config-ca-trustpoint)# subject-name cn=Ent-local-domain-name**., o="Example Corp"	Creates an RSA key and trustpoint for the self-signed certificate. Where *key-pair-label* is the RSA key for the local ASA. Where *trustpoint_name* is the trustpoint for the local ASA. Where *keyname* is key pair for the local ASA. Where *x.500_name* includes the X.500 distinguished name of the local ASA; for example, cn=Ent-local-domain-name**. **Note** The domain name that you enter here must match the domain name that has been set for the local Cisco UCM. For information about how to configure the domain name for Cisco UCM, see the Cisco Unified Communications Manager documentation for information.
Step 2	hostname(config-ca-trustpoint)# **exit**	Exits from Trustpoint Configuration mode.
Step 3	hostname(config)# **crypto ca export** *trustpoint* **identity-certificate** **Example:** hostname(config)# crypto ca export local-asa identity-certificate	Exports the certificate you created in Step 1. The certificate contents appear on the terminal screen. Copy the certificate from the terminal screen. This certificate enables Cisco UCM to validate the certificate that the ASA sends in the TLS handshake. On the local Cisco UCM, upload the certificate into the Cisco UCM trust store. See the Cisco Unified Communications Manager documentation for information. **Note** The subject name you enter while uploading the certificate to the local Cisco UCM is compared with the X.509 Subject Name field entered on the SIP Trunk Security Profile on Cisco UCM. For example, "Ent-local-domain-name" was entered in Step 1 of this task; therefore, "Ent-local-domain-name" should be entered in the Cisco UCM configuration.
Step 4	hostname(config)# **crypto ca trustpoint** *trustpoint_name* hostname(config-ca-trustpoint)# **enroll terminal** **Example:** hostname(config)# crypto ca trustpoint local-ent-ucm hostname(config-ca-trustpoint)# enroll terminal	Creates a trustpoint for local Cisco UCM. Where *trustpoint_name* is the trustpoint for the local Cisco UCM.
Step 5	hostname(config-ca-trustpoint)# **exit**	Exits from Trustpoint Configuration mode.

	Commands	Purpose
Step 6	hostname(config)# **crypto ca authenticate** *trustpoint* **Example:** hostname(config)# crypto ca authenticate local-ent-ucm	Imports the certificate from local Cisco UCM. Where *trustpoint* is the trustpoint for the local Cisco UCM. Paste the certificate downloaded from the local Cisco UCM. This certificate enables the ASA to validate the certificate that Cisco UCM sends in the TLS handshake.
Step 7	hostname(config)# **tls-proxy** *proxy_name* hostname(config-tlsp)# **server trust-point** *proxy_trustpoint* hostname(config-tlsp)# **client trust-point** *proxy_trustpoint* hostname(config-tlsp)# **client cipher-suite** aes128-sha1 aes256-sha1 3des-sha1 null-sha1 **Example:** hostname(config)# tls-proxy local_to_remote-ent hostname(config-tlsp)# server trust-point local-ent-ucm hostname(config-tlsp)# client trust-point local-ent hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1	Updates the TLS proxy for **outbound** connections. Where *proxy_name* is the name you entered in Step 1 of the task Creating the TLS Proxy. Where *proxy_trustpoint* for the **server trust-point** command is the name you entered in Step 4 of this procedure. Where *proxy_trustpoint* for the **client trust-point** command is the name you entered in Step 2 of the task Creating Trustpoints and Generating Certificates. **Note** In this step, you are creating different trustpoints for the client and the server.
Step 8	hostname(config-tlsp)# **exit**	Exits from TLS Proxy Configuration mode.
Step 9	hostname(config)# **tls-proxy** proxy_name hostname(config-tlsp)# **server trust-point** *proxy_trustpoint* hostname(config-tlsp)# **client trust-point** *proxy_trustpoint* hostname(config-tlsp)# **client cipher-suite** aes128-sha1 aes256-sha1 3des-sha1 null-sha1 **Example:** hostname(config)# tls-proxy remote_to_local-ent hostname(config-tlsp)# server trust-point local-ent hostname(config-tlsp)# client trust-point local-ent-ucm hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1	Updates the TLS proxy for **inbound** connections. Where *proxy_name* is the name you entered in Step 5 of the task Creating the TLS Proxy. Where *proxy_trustpoint* for the **server trust-point** command is the name you entered in Step 2 of the task Creating Trustpoints and Generating Certificates. Where *proxy_trustpoint* for the **client trust-point** command is the name you entered in Step 4 of this procedure.
Step 10	hostname(config-tlsp)# **exit**	Exits from TLS Proxy Configuration mode.
Step 11	hostname(config)# **uc-ime** *uc_ime_name* hostname(config-uc-ime)# **ucm address** *ip_address* **trunk-security-mode secure** **Example:** hostname(config)# uc-ime local-ent-ime hostname(config-uc-ime)# ucm address 192.168.10.30 trunk-security-mode secure	Updates the Cisco Intercompany Media Engine Proxy for trunk-security-mode. Where *uc_ime_name* is the name you entered in Step 1 of the task Creating the Cisco Intercompany Media Engine Proxy. Only perform this step if you entered nonsecure in Step 3 of the task Creating the Cisco Intercompany Media Engine Proxy.

What to Do Next

Once you have configured the TLS within the enterprise, if necessary, configure off path signaling for an off path deployment. See (Optional) Configuring Off Path Signaling, page 20-29.

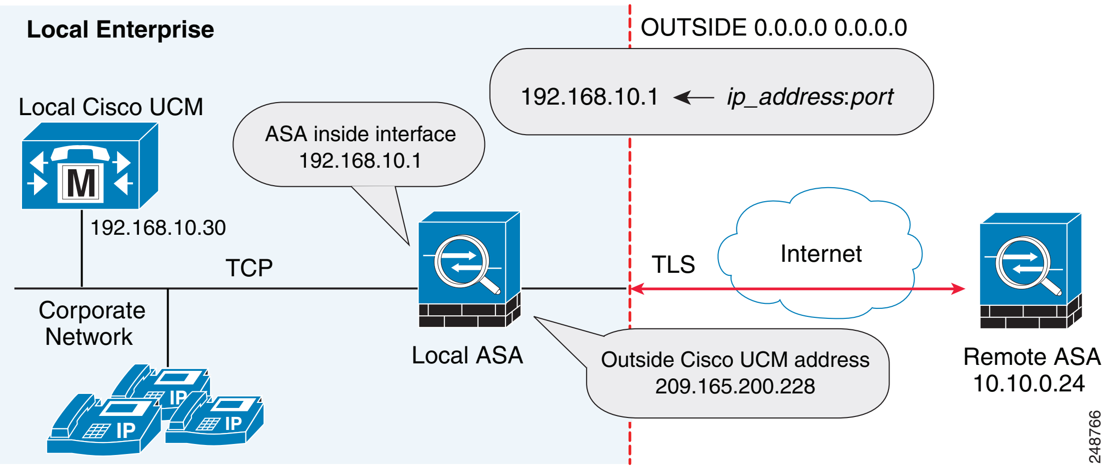
(Optional) Configuring Off Path Signaling

Perform this task only when you are configuring the Cisco Intercompany Media Engine Proxy as part of an off path deployment. You might choose to have an off path deployment when you want to use the Cisco Intercompany Media Engine but do not want to replace your existing Internet firewall with an ASA enabled with the Cisco Intercompany Media Engine Proxy.

In an off path deployment, the existing firewall that you have deployed in your environment is not capable of transmitting Cisco Intercompany Media Engine traffic.

Off path signaling requires that outside IP addresses translate to an inside IP address. The inside interface address can be used for this mapping service configuration. For the Cisco Intercompany Media Engine Proxy, the ASA creates dynamic mappings for external addresses to the internal IP address; therefore, using the dynamic NAT configuration on outbound calls, Cisco UCM sends SIP traffic to this internal IP address, and the ASA uses that mapping to determine the real destination on inbound calls. The static NAT or PAT mapping is used for inbound calls in an off path configuration.

Figure 20-8 Example for Configuring Off Path Signaling in an Off Path Deployment



After you configure off path signaling, the ASA mapping service listens on interface "inside" for requests. When it receives a request, it creates a dynamic mapping for the "outside" as the destination interface.

To configure off path signaling for the Cisco Intercompany Media Engine Proxy, perform the following steps:

	Command	Purpose

Step 1	`hostname(config)# `**`object network`** *`name`*
	Example:
	`hostname(config)# object network outside-any`
		For the off path ASA, creates a network object to represent all outside addresses.

Step 2	`hostname(config-network-object)# `**`subnet`** *`ip_address`*
	Example:
	`hostname(config-network-object)# subnet 0.0.0.0 0.0.0.0`
		Specifies the IP address of the subnet.

Step 3	`hostname(config-network-object)# `**`nat (outside,inside) dynamic interface inside`**	Creates a mapping for the Cisco UCM of remote enterprises.
Step 4	`hostname(config-network-object)# `**`exit`**	Exits from the objects configuration mode.

	Command	Purpose

Step 5	`hostname(config)#` **`uc-ime`** *`uc_ime_name`*
	`Example:`
	`hostname(config)# uc-ime local-ent-ime`
		Specifies the Cisco Intercompany Media Engine Proxy that you created in the task Creating the Cisco Intercompany Media Engine Proxy, page 20-17.
		Where *`uc_ime_name`* is the name you specified in Step 1 of Creating the Cisco Intercompany Media Engine Proxy, page 20-17.

Step 6	`hostname(config)#` **`mapping-service listening-interface`** *`interface_name`* `[`**`listening-port`** *`port`*`]` **`uc-ime-interface`** *`uc-ime-interface_name`*
	`Example:`
	`hostname(config-uc-ime)# mapping-service listening-interface inside listening-port 8060 uc-ime-interface outside`
		For the off path ASA, adds the mapping service to the Cisco Intercompany Media Engine Proxy.
		Specifies the interface and listening port for the ASA mapping service.
		You can only configure one mapping server for the Cisco Intercompany Media Engine Proxy.
		Where *`interface_name`* is the name of the interface on which the ASA listens for the mapping requests.
		Where port is the TCP port on which the ASA listens for the mapping requests. The port number must be between 1024 and 65535 to avoid conflicts with other services on the device, such as Telnet or SSH. By default, the port number is TCP 8060.
		Where *`uc-ime-interface_name`* is the name of the interface that connects to the remote Cisco UCM.

This section contains the following sections:

- Configuring the Cisco UC-IMC Proxy by using the UC-IME Proxy Pane, page 20-30
- Configuring the Cisco UC-IMC Proxy by using the Unified Communications Wizard, page 20-32

Configuring the Cisco UC-IMC Proxy by using the UC-IME Proxy Pane

Use the Configure Cisco Intercompany Media Engine (UC-IME) proxy pane to add or edit a Cisco Intercompany Media Engine Proxy instance.

Note The Cisco Intercompany Media Engine Proxy does not appear as an option under the Unified Communications section of the navigation pane unless the license required for this proxy is installed on the ASA.

Use this pane to create the proxy instance; however, for the UC-IME proxy to be fully functionally, you must complete additional tasks, such as create the required NAT statements, ACLs, and MTA, set up the certificates, create the TLS Proxy, and enable SIP inspection.

Depending on whether the UC-IME proxy is deployed off path or in-line of Internet traffic, you must create the appropriate network objects with embedded NAT/PAT statements for the Cisco UCMs.

This pane is available from the Configuration > Firewall > Unified Communications > UC-IME Proxy.

Step 1 Open the Configuration > Firewall > Unified Communications > UC-IME Proxy pane.

Step 2 Check the Enable Cisco UC-IME proxy check box to enable the feature.

Step 3 In the Unified CM Servers area, enter an IP address or hostname for the Cisco Unified Communications Manager (Cisco UCM) or click the ellipsis to open a dialog and browse for an IP address or hostname.

Step 4 In the Trunk Security Mode field, click a security option. Specifying **secure** for Cisco UCM or Cisco UCM cluster indicates that Cisco UCM or Cisco UCM cluster is initiating TLS.

Step 5 Click **Add** to add the Cisco UCM for the Cisco Intercompany Media Engine Proxy. You must include an entry for each Cisco UCM in the cluster with Cisco Intercompany Media Engine that has a SIP trunk enabled.

Step 6 In the Ticket Epoch field, enter an integer from 1-255.

The epoch contains an integer that updates each time that the password is changed. When the proxy is configured the first time and a password entered for the first time, enter 1 for the epoch integer. Each time you change the password, increment the epoch to indicate the new password. You must increment the epoch value each time your change the password.

Typically, you increment the epoch sequentially; however, the ASA allows you to choose any value when you update the epoch.

If you change the epoch value, the current password is invalidated and you must enter a new password.

Note The epoch and password that you configure in this step on the ASA must match the epoch and password that you configure on the Cisco Intercompany Media Engine server. See the Cisco Intercompany Media Engine server documentation for information.

Step 7 In the Ticket Password field, enter a minimum of 10 printable character from the US-ASCII character set. The allowed characters include 0x21 to 0x73 inclusive, and exclude the space character. The ticket password can be up to 64 characters. Confirm the password you entered. Only one password can be configured at a time.

Step 8 Check the Apply MTA to UC-IME Link proxy check box to associate the media termination address with the Cisco Intercompany Media Engine Proxy.

Note You must create the media termination instance before you associate it with the Cisco Intercompany Media Engine Proxy. If necessary, click the Configure MTA button to configure a media termination address instance.

Step 9 If the Cisco Intercompany Media Engine Proxy is being configured as part of off path deployment, check the Enable off path address mapping service checkbox and configure the off path deployment settings:

a. From the Listening Interface field, select an ASA interface. This is the interface on which the ASA listens for the mapping requests.

b. In the Port field, enter a number between 1024 and 65535 as the TCP port on which the ASA listens for the mapping requests. The port number must be 1024 or higher to avoid conflicts with other services on the device, such as Telnet or SSH. By default, the port number is TCP 8060.

c. From the UC-IME Interface field, select an interface from the list. This is the interface that the ASA uses to connect to the remote Cisco UCM.

Note In an off path deployment any existing ASA that you have deployed in your environment are not capable of transmitting Cisco Intercompany Media Engine traffic. Off-path signaling requires that outside addresses are translated (using NAT) to an inside IP address. The inside interface address can be used for this mapping service configuration. For the Cisco Intercompany Media Engine Proxy, the ASA creates dynamic mappings for external addresses to the internal IP address.

Step 10 In the Fallback area, configure the fallback timer for the Cisco Intercompany Media Engine by specifying the following settings:

a. In the Fallback Sensitivity File field, enter the path to a file in flash memory that the ASA uses for mid-call PSTN fallback. The file name that you enter must be the name of a file on disk that includes the .fbs file extension. Alternatively, click the Browse Flash button to locate and select the file from flash memory.

b. In the Call Quality Evaluation Interval field, enter a number between 10-600 (in milliseconds). This number controls the frequency at which the ASA samples the RTP packets received from the Internet. The ASA uses the data sample to determine if fallback to the PSTN is needed for a call. By default, the length is 100 milliseconds for the timer.

c. In the Notification Interval field, enter a number between 10-360 (in seconds). This number controls the amount of time that the ASA waits before notifying Cisco UCM whether to fall back to PSTN. By default, the length is 20 seconds for this timer.

Note When you change the fallback timer for the Cisco Intercompany Media Engine Proxy, ASDM automatically removes the proxy from SIP inspection and then reapplies SIP inspection when the proxy is re-enabled.

Step 11 Click Apply to save the configuration changes for the Cisco Intercompany Media Engine Proxy.

Configuring the Cisco UC-IMC Proxy by using the Unified Communications Wizard

To configure the Cisco Intercompany Media Engine Proxy by using ASDM, choose Wizards > Unified Communications Wizard from the menu. The Unified Communications Wizard opens. From the first page, select the Cisco Intercompany Media Engine Proxy option under the Business-to-Business section.

The wizard automatically creates the necessary TLS proxy, then guides you through creating the Intercompany Media Engine proxy, importing and installing the required certificates, and finally enables the SIP inspection for the Intercompany Media Engine traffic automatically.

The wizard guides you through these steps to create the Cisco Intercompany Media Engine Proxy:

Step 1 Select the Intercompany Media Engine Proxy option.

Step 2 Select the topology of the Cisco Intercompany Media Engine Proxy, namely whether the ASA is an edge firewall with all Internet traffic flowing through it or whether the ASA is off the path of the main Internet traffic (referred to as an off path deployment).

Step 3 Specify private network settings such as the Cisco UCM IP addresses and the ticket settings.

Step 4 Specify the public network settings.

Step 5 Specify the media termination address settings of Cisco UCM.

Step 6 Configure the local-side certificate management, namely the certificates that are exchanged between the local Cisco Unified Communications Manager servers and the ASA. The identity certificate that the wizard generates in this step needs to be installed on each Cisco Unified Communications Manager (UCM) server in the cluster with the proxy and each identity certificate from the Cisco UCMs need to be installed on the ASA. The certificates are used by the ASA and the Cisco UCMs to authenticate each other, respectively, during TLS handshakes. The wizard only supports self-signed certificates for this step.

Step 7 Configure the remote-side certificate management, namely the certificates that are exchanged between the remote server and the ASA. In this step, the wizard generates a certificate signing request (CSR). After successfully generating the identity certificate request for the proxy, the wizard prompts you to save the file.

You must send the CSR text file to a certificate authority (CA), for example, by pasting the text file into the CSR enrollment page on the CA website. When the CA returns the Identity Certificate, you must install it on the ASA. This certificate is presented to remote servers so that they can authenticate the ASA as a trusted server.

Finally, this step of the wizard assists you in installing the root certificates of the CA from the remote servers so that the ASA can determine that the remote servers are trusted.

The wizard completes by displaying a summary of the configuration created for Cisco Intercompany Media Engine. See the Unified Communications Wizard section in this documentation for more information.

Troubleshooting Cisco Intercompany Media Engine Proxy

This section describes how to certain options of the **show uc-ime** command to obtain troubleshooting information for the Cisco Intercompany Media Engine Proxy. See the command reference for detailed information about the syntax for these commands.

show uc-ime signaling-sessions

Displays the corresponding SIP signaling sessions stored by the Cisco Intercompany Media Engine Proxy. Use this command to troubleshoot media or signaling failure. The command also displays the fallback parameters extracted from the SIP message headers, whether RTP monitoring is enabled or disabled, and whether SRTP keys are set.

Through the use of the Cisco Intercompany Media Engine Proxy, not only signaling but also media is secured for communication. It provides signaling encryption and SRTP/RTP conversion with SRTP enforced on the Internet side. The Cisco Intercompany Media Engine Proxy inserts itself into the media path by modifying the SIP signaling messages from Cisco UCMs.The Cisco Intercompany Media Engine Proxy sits on the edge of the enterprise and inspects SIP signaling between SIP trunks created between enterprises. It terminates TLS signaling from the Internet and initiates TCP or TLS to the local Cisco UCM.

```
hostname# show uc-ime signaling-sessions
 1 in use, 3 most used
 inside 192.168.10.30:39608 outside 10.194.108.118:5070
   Local Media (audio) conn: 10.194.108.119/29824 to 10.194.108.109/21558
```

```
      Local SRTP key set : Remote SRTP key set
    Remote Media (audio) conn: 192.168.10.51/19520 to 192.168.10.3/30930
    Call-ID: ab6d7980-a7d11b08-50-1e0aa8c0@192.168.10.30
    FB Sensitivity: 3
    Session ID: 2948-32325449-0@81a985c9-f3a1-55a0-3b19-96549a027259
    SIP Trunk URI: 81a985c9-f3a1-55a0-3b19-9654@UCM-30;maddr=192.168.10.30
    Codec-name: G722
    Payload type: 9
```

Note If calls are not going through the Cisco Intercompany Media Engine, you can also use the **show tls-proxy session** command to troubleshoot the success of the TLS handshake between the components in the Cisco Intercompany Media Engine system. See the command reference for information about this command.

show uc-ime signaling-sessions statistics

Displays statistical information about corresponding signaling sessions stored by Cisco Intercompany Media Engine Proxy. Failure of signaling sessions in the Cisco Intercompany Media Engine can occur for different call-related reasons; such as failure of ticket verification or domain name verification, or offering RTP over the Internet.

```
hostname# show uc-ime signaling-sessions statistics
10 in use, 20 most used
15 terminated
  Ticket integrity check failed: 2
  Ticket decode failed: 1
  Ticket epoch mismatch: 1
  Ticket DID mismatch: 0
  Ticket timestamp invalid: 4
  Ticket domain check failed: 2
  Ticket not found: 0
  Route domain name check failed: 1
  RTP over UC-IME: 2
```

Note Call-related failures, for example, can be due to the service policy rule being reconfigured or the primary ASA operating in failover mode. If a service policy rule for the Cisco Intercompany Media Engine Proxy is removed (by using the **no service policy** command) and reconfigured, the first call trasversing the ASA will fail. To resolve this issue, you must additionally enter the **clear connection** command and restart the ASA. If the failure is due to failover, the connections from the primary ASA are not synchronized to the standby ASA.

show uc-ime media-sessions detail

Displays the details about all active media sessions (calls) stored for the Cisco Intercompany Media Engine Proxy. Use this command to display output from successful calls. Additionally, use this command to troubleshoot problems with IP phone audio, such as one-way audio. If no calls are currently up, this output will be blank.

```
hostname(config)# show uc-ime media-sessions detail
 2 in use, 5 most used
 Media-session: 10.194.108.109/21558 :: client ip 192.168.10.51/19520
 Call ID: ab6d7980-a7d11b08-50-1e0aa8c0@192.168.10.30
 Session ID: 2948-32325449-0@81a985c9-f3a1-55a0-3b19-96549a027259
   Lcl SRTP conn 10.194.108.109/21558 to 10.194.108.119/29824 tx_pkts 20203 rx_pkts 20200
   refcnt 3 : created by Inspect SIP, passthrough not set
   RTP monitoring is enabled
      Failover_state                :  0
      Sum_all_packets                :  20196
      Codec_payload_format           :  9
      RTP_ptime_ms                   :  20
      Max_RBLR_pct_x100              :  0
      Max_ITE_count_in_8_sec         :  0
      Max_BLS_ms                     :  0
      Max_PDV_usec                   :  1000
      Min_PDV_usec                   :  0
      Mov_avg_PDV_usec               :  109
      Total_ITE_count                :  0
      Total_sec_count                :  403
      Concealed_sec_count            :  0
      Severely_concealed_sec_count   :  0
      Max_call_interval_ms           :  118
      Total_SequenceNumber_Resets    :  0
 Media-session: 192.168.10.3/30930 :: client ip 10.194.108.119/29824
 Call ID: N/A
   Lcl RTP conn 192.168.10.3/30930 to 192.168.10.51/19520 tx_pkts 20201 rx_pkts 20203
```

show uc-ime fallback-notification statistics

Displays statistics about the PSTN fallback notifications to the Cisco UMC. Even if a call is sent over VoIP because the quality of the connection was good, the connection quality might worsen mid-call. To ensure an overall good experience for the end user, Cisco Intercompany Media Engine attempts to perform a mid-call fallback. Performing a mid-call fallback requires the adaptive security appliance to monitor the RTP packets coming from the Internet. If fallback is required, the adaptive security appliance sends a REFER message to Cisco UCM to tell it that it needs to fallback the call to PSTN.

Cisco Intercompany Media Engine uses a configurable hold-down timer to set the amount of time that adaptive security appliance waits before notifying Cisco UCM whether to fall back to PSTN.

```
hostname# show uc-ime fallback-notification statistics
UCM address: 172.23.32.37
 Total Notifications Sent: 10
```

show uc-ime mapping-service-sessions

When the Cisco Intercompany Media Engine Proxy is configured for an off path deployment, displays mapping-service requests and replies between the proxy and the local Cisco UMC. A TCP port on the ASA is configured to listen for mapping requests.

The port number must be 1024 or higher to avoid conflicts with other services on the device, such as Telnet or SSH. By default, the port number is TCP 8060.

```
Hostname# show uc-b2blink mapping-service-sessions
Total active sessions:  2
Session client (IP:Port)      Idle time
192.168.1.10:2001             0:01:01
192.168.1.20:3001             0:10:20
```

show uc-ime mapping-service-sessions statistics

Displays statistical information about the Cisco Intercompany Media Engine Proxy mapping service used in off path signaling.

```
Hostname# show uc-ime mapping-service-sessions statistics
Total active sessions:  2
Session client      Total       Responses   Failed      Pending     Idle
(IP:Port)           requests    sent        requests    responses   time
192.168.1.10:2001   10          9           1           0           0:01:01
192.168.1.20:3001   19          19          0           0           0:10:20
```

Feature History for Cisco Intercompany Media Engine Proxy

Table 20-1 lists the release history for this feature.

Table 20-1 Feature History for Cisco Phone Proxy

Feature Name	Releases	Feature Information
Cisco Intercompany Media Engine Proxy	8.3(1)	The Cisco Intercompany Media Engine Proxy was introduced. The following commands were added to the CLI to support configuration of this new feature. [no] **uc-ime** *uc_ime_name* [no] **fallback hold-down** | **monitoring timer** *value* [no] **fallback sensitivity-file** *filename* [no] **mapping-service listening-interface** *ifc_name* [**listening-port** *port*] **uc-ime-interface** *b2b-ifc* [no] **ticket epoch** *epoch* **password** *pwd* [no] **ucm address** *ip_addr* **trunk-security-mode nonsecure** | **secure** **clear configure uc-ime** [*uc_ime_name*] [no] **debug uc-ime** [**mapping-service** | **media** | **notification** | **rma** | **signaling**] [**errors** | **events**] **show uc-ime** **show running-config** [**all**] **uc-ime** *[uc_ime_map*] The following command was updated by adding options for the UC-IME proxy. **inspect sip uc-ime** *uc-ime-name* **tls-proxy** *tls-proxy-name*

PART 6

Configuring Connection Settings and QoS

CHAPTER 22

Configuring Connection Settings

This chapter describes how to configure connection settings for connections that go through the ASA, or for management connections, that go to the ASA. Connection settings include:

- Maximum connections (TCP and UDP connections, embryonic connections, per-client connections)
- Connection timeouts
- Dead connection detection
- TCP sequence randomization
- TCP normalization customization
- TCP state bypass
- Global timeouts

This chapter includes the following sections:

- Information About Connection Settings, page 22-1
- Licensing Requirements for Connection Settings, page 22-4
- Guidelines and Limitations, page 22-5
- Default Settings, page 22-5
- Configuring Connection Settings, page 22-6
- Monitoring Connection Settings, page 22-15
- Configuration Examples for Connection Settings, page 22-15
- Feature History for Connection Settings, page 22-17

Information About Connection Settings

This section describes why you might want to limit connections and includes the following topics:

- TCP Intercept and Limiting Embryonic Connections, page 22-2
- Disabling TCP Intercept for Management Packets for Clientless SSL Compatibility, page 22-2
- Dead Connection Detection (DCD), page 22-2
- TCP Sequence Randomization, page 22-3
- TCP Normalization, page 22-3
- TCP State Bypass, page 22-3

TCP Intercept and Limiting Embryonic Connections

Limiting the number of embryonic connections protects you from a DoS attack. The ASA uses the per-client limits and the embryonic connection limit to trigger TCP Intercept, which protects inside systems from a DoS attack perpetrated by flooding an interface with TCP SYN packets. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. TCP Intercept uses the SYN cookies algorithm to prevent TCP SYN-flooding attacks. A SYN-flooding attack consists of a series of SYN packets usually originating from spoofed IP addresses. The constant flood of SYN packets keeps the server SYN queue full, which prevents it from servicing connection requests. When the embryonic connection threshold of a connection is crossed, the ASA acts as a proxy for the server and generates a SYN-ACK response to the client SYN request. When the ASA receives an ACK back from the client, it can then authenticate the client and allow the connection to the server.

Note When you use TCP SYN cookie protection to protect servers from SYN attacks, you must set the embryonic connection limit lower than the TCP SYN backlog queue on the server that you want to protect. Otherwise, valid clients can nolonger access the server during a SYN attack.

To view TCP Intercept statistics, including the top 10 servers under attack, see Chapter 27, "Configuring Threat Detection."

Disabling TCP Intercept for Management Packets for Clientless SSL Compatibility

By default, TCP management connections have TCP Intercept always enabled. When TCP Intercept is enabled, it intercepts the 3-way TCP connection establishment handshake packets and thus deprives the ASA from processing the packets for clientless SSL. Clientless SSL requires the ability to process the 3-way handshake packets to provide selective ACK and other TCP options for clientless SSL connections. To disable TCP Intercept for management traffic, you can set the embryonic connection limit; only after the embryonic connection limit is reached is TCP Intercept enabled.

Dead Connection Detection (DCD)

DCD detects a dead connection and allows it to expire, without expiring connections that can still handle traffic. You configure DCD when you want idle, but valid connections to persist.

When you enable DCD, idle timeout behavior changes. With idle timeout, DCD probes are sent to each of the two end-hosts to determine the validity of the connection. If an end-host fails to respond after probes are sent at the configured intervals, the connection is freed, and reset values, if configured, are sent to each of the end-hosts. If both end-hosts respond that the connection is valid, the activity timeout is updated to the current time and the idle timeout is rescheduled accordingly.

Enabling DCD changes the behavior of idle-timeout handling in the TCP normalizer. DCD probing resets the idle timeout on the connections seen in the **show conn** command. To determine when a connection that has exceeded the configured timeout value in the timeout command but is kept alive due to DCD probing, the **show service-policy** command includes counters to show the amount of activity from DCD.

TCP Sequence Randomization

Each TCP connection has two ISNs: one generated by the client and one generated by the server. The ASA randomizes the ISN of the TCP SYN passing in both the inbound and outbound directions.

Randomizing the ISN of the protected host prevents an attacker from predecting the next ISN for a new connection and potentially hijacking the new session.

TCP initial sequence number randomization can be disabled if required. For example:

- If another in-line firewall is also randomizing the initial sequence numbers, there is no need for both firewalls to be performing this action, even though this action does not affect the traffic.
- If you use eBGP multi-hop through the ASA, and the eBGP peers are using MD5. Randomization breaks the MD5 checksum.
- You use a WAAS device that requires the ASA not to randomize the sequence numbers of connections.

TCP Normalization

The TCP normalization feature identifies abnormal packets that the ASA can act on when they are detected; for example, the ASA can allow, drop, or clear the packets. TCP normalization helps protect the ASA from attacks. TCP normalization is always enabled, but you can customize how some features behave.

The TCP normalizer includes non-configurable actions and configurable actions. Typically, non-configurable actions that drop or clear connections apply to packets that are always bad. Configurable actions (as detailed in "Customizing the TCP Normalizer with a TCP Map" section on page 22-6) might need to be customized depending on your network needs.

See the following guidelines for TCP normalization:

- The normalizer does not protect from SYN floods. The ASA includes SYN flood protection in other ways.
- The normalizer always sees the SYN packet as the first packet in a flow unless the ASA is in loose mode due to failover.

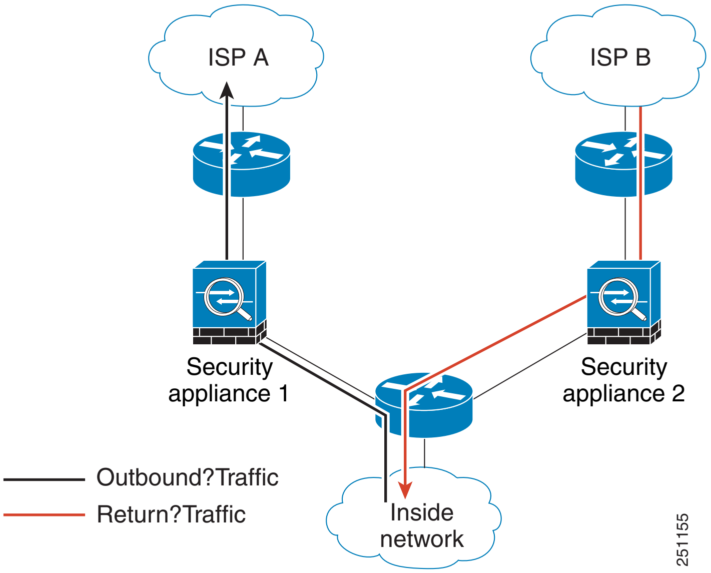
TCP State Bypass

By default, all traffic that goes through the ASA is inspected using the Adaptive Security Algorithm and is either allowed through or dropped based on the security policy. The ASA maximizes the firewall performance by checking the state of each packet (is this a new connection or an established connection?) and assigning it to either the session management path (a new connection SYN packet), the fast path (an established connection), or the control plane path (advanced inspection). See the "Stateful Inspection Overview" section on page 1-22 in the general operations configuration guide for more detailed information about the stateful firewall.

TCP packets that match existing connections in the fast path can pass through the ASA without rechecking every aspect of the security policy. This feature maximizes performance. However, the method of establishing the session in the fast path using the SYN packet, and the checks that occur in the fast path (such as TCP sequence number), can stand in the way of asymmetrical routing solutions: both the outbound and inbound flow of a connection must pass through the same ASA.

For example, a new connection goes to ASA 1. The SYN packet goes through the session management path, and an entry for the connection is added to the fast path table. If subsequent packets of this connection go through ASA 1, then the packets will match the entry in the fast path, and are passed through. But if subsequent packets go to ASA 2, where there was not a SYN packet that went through the session management path, then there is no entry in the fast path for the connection, and the packets are dropped. Figure 22-1 shows an asymmetric routing example where the outbound traffic goes through a different ASA than the inbound traffic:

Figure 22-1 Asymmetric Routing



If you have asymmetric routing configured on upstream routers, and traffic alternates between two ASAs, then you can configure TCP state bypass for specific traffic. TCP state bypass alters the way sessions are established in the fast path and disables the fast path checks. This feature treats TCP traffic much as it treats a UDP connection: when a non-SYN packet matching the specified networks enters the ASA, and there is not an fast path entry, then the packet goes through the session management path to establish the connection in the fast path. Once in the fast path, the traffic bypasses the fast path checks.

Licensing Requirements for Connection Settings

Model	License Requirement
All models	Base License.

Guidelines and Limitations

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent mode.

Failover Guidelines

Failover is supported.

TCP State Bypass Unsupported Features

The following features are not supported when you use TCP state bypass:

- Application inspection—Application inspection requires both inbound and outbound traffic to go through the same ASA, so application inspection is not supported with TCP state bypass.
- AAA authenticated sessions—When a user authenticates with one ASA, traffic returning via the other ASA will be denied because the user did not authenticate with that ASA.
- TCP Intercept, maximum embryonic connection limit, TCP sequence number randomization—The ASA does not keep track of the state of the connection, so these features are not applied.
- TCP normalization—The TCP normalizer is disabled.
- SSM and SSC functionality—You cannot use TCP state bypass and any application running on an SSM or SSC, such as IPS or CSC.

TCP State Bypass NAT Guidelines

Because the translation session is established separately for each ASA, be sure to configure static NAT on both ASAs for TCP state bypass traffic; if you use dynamic NAT, the address chosen for the session on ASA 1 will differ from the address chosen for the session on ASA 2.

Maximum Concurrent and Embryonic Connection Guidelines

Depending on the number of CPU cores on your ASA model, the maximum concurrent and embryonic connections may exceed the configured numbers due to the way each core manages connections. In the worst case scenario, the ASA allows up to *n*-1 extra connections and embryonic connections, where *n* is the number of cores. For example, if your model has 4 cores, if you configure 6 concurrent connections and 4 embryonic connections, you could have an additional 3 of each type. To determine the number of cores for your model, enter the **show cpu core** command.

Default Settings

TCP State Bypass

TCP state bypass is disabled by default.

TCP Normalizer

The default configuration includes the following settings:

```
no check-retransmission
no checksum-verification
exceed-mss allow
queue-limit 0 timeout 4
reserved-bits allow
syn-data allow
synack-data drop
invalid-ack drop
seq-past-window drop
tcp-options range 6 7 clear
tcp-options range 9 255 clear
tcp-options selective-ack allow
tcp-options timestamp allow
tcp-options window-scale allow
ttl-evasion-protection
urgent-flag clear
window-variation allow-connection
```

Configuring Connection Settings

This section includes the following topics:

- Customizing the TCP Normalizer with a TCP Map, page 22-6
- Configuring Connection Settings, page 22-11

Task Flow For Configuring Connection Settings

Step 1 For TCP normalization customization, create a TCP map according to the "Customizing the TCP Normalizer with a TCP Map" section on page 22-6.

Step 2 For all connection settings, configure a service policy according to Chapter 1, "Configuring a Service Policy Using the Modular Policy Framework."

Step 3 Configure connection settings according to the "Configuring Connection Settings" section on page 22-11.

Customizing the TCP Normalizer with a TCP Map

To customize the TCP normalizer, first define the settings using a TCP map.

Detailed Steps

Step 1 To specify the TCP normalization criteria that you want to look for, create a TCP map by entering the following command:

```
ciscoasa(config)# tcp-map tcp-map-name
```

For each TCP map, you can customize one or more settings.

Step 2 (Optional) Configure the TCP map criteria by entering one or more of the following commands (see Table 22-1). If you want to customize some settings, then the defaults are used for any commands you do not enter.

Table 22-1 tcp-map Commands

Command	Notes
check-retransmission	Prevents inconsistent TCP retransmissions.
checksum-verification	Verifies the checksum.
exceed-mss {**allow** | **drop**}	Sets the action for packets whose data length exceeds the TCP maximum segment size. (Default) The **allow** keyword allows packets whose data length exceeds the TCP maximum segment size. The **drop** keyword drops packets whose data length exceeds the TCP maximum segment size.
invalid-ack {**allow** | **drop**}	Sets the action for packets with an invalid ACK. You might see invalid ACKs in the following instances: • In the TCP connection SYN-ACK-received status, if the ACK number of a received TCP packet is not exactly same as the sequence number of the next TCP packet sending out, it is an invalid ACK. • Whenever the ACK number of a received TCP packet is greater than the sequence number of the next TCP packet sending out, it is an invalid ACK. The **allow** keyword allows packets with an invalid ACK. (Default) The **drop** keyword drops packets with an invalid ACK. **Note** TCP packets with an invalid ACK are automatically allowed for WAAS connections.

Table 22-1 tcp-map Commands (continued)

Command	Notes
queue-limit *pkt_num* [**timeout** *seconds*]	Sets the maximum number of out-of-order packets that can be buffered and put in order for a TCP connection, between 1 and 250 packets. The default is 0, which means this setting is disabled and the default system queue limit is used depending on the type of traffic: • Connections for application inspection (the **inspect** command), IPS (the **ips** command), and TCP check-retransmission (the TCP map **check-retransmission** command) have a queue limit of 3 packets. If the ASA receives a TCP packet with a different window size, then the queue limit is dynamically changed to match the advertised setting. • For other TCP connections, out-of-order packets are passed through untouched. If you set the **queue-limit** command to be 1 or above, then the number of out-of-order packets allowed for all TCP traffic matches this setting. For example, for application inspection, IPS, and TCP check-retransmission traffic, any advertised settings from TCP packets are ignored in favor of the **queue-limit** setting. For other TCP traffic, out-of-order packets are now buffered and put in order instead of passed through untouched. The **timeout** *seconds* argument sets the maximum amount of time that out-of-order packets can remain in the buffer, between 1 and 20 seconds; if they are not put in order and passed on within the timeout period, then they are dropped. The default is 4 seconds. You cannot change the timeout for any traffic if the *pkt_num* argument is set to 0; you need to set the limit to be 1 or above for the **timeout** keyword to take effect.
reserved-bits {**allow** | **clear** | **drop**}	Sets the action for reserved bits in the TCP header. (Default) The **allow** keyword allows packets with the reserved bits in the TCP header. The **clear** keyword clears the reserved bits in the TCP header and allows the packet. The **drop** keyword drops the packet with the reserved bits in the TCP header.
seq-past-window {**allow** | **drop**}	Sets the action for packets that have past-window sequence numbers, namely the sequence number of a received TCP packet is greater than the right edge of the TCP receiving window. The **allow** keyword allows packets that have past-window sequence numbers. This action is only allowed if the **queue-limit** command is set to 0 (disabled). (Default) The **drop** keyword drops packets that have past-window sequence numbers.

Table 22-1 tcp-map Commands (continued)

Command	Notes
synack-data {**allow** | **drop**}	Sets the action for TCP SYNACK packets that contain data. The **allow** keyword allows TCP SYNACK packets that contain data. (Default) The **drop** keyword drops TCP SYNACK packets that contain data.
syn-data {**allow** | **drop**}	Sets the action for SYN packets with data. (Default) The **allow** keyword allows SYN packets with data. The **drop** keyword drops SYN packets with data.

tcp-options {**selective-ack** | **timestamp** | **window-scale**} {**allow** | **clear**}
Or
tcp-options range *lower upper* {**allow** | **clear** | **drop**}
	Sets the action for packets with TCP options, including the selective-ack, timestamp, or window-scale TCP options.
	(Default) The **allow** keyword allows packets with the specified option.
	(Default for **range**) The **clear** keyword clears the option and allows the packet.
	The **drop** keyword drops the packet with the specified option.
	The **selective-ack** keyword sets the action for the SACK option.
	The **timestamp** keyword sets the action for the timestamp option. Clearing the timestamp option disables PAWS and RTT.
	The **widow-scale** keyword sets the action for the window scale mechanism option.
	The **range** keyword specifies a range of options. The *lower* argument sets the lower end of the range as 6, 7, or 9 through 255.
	The *upper* argument sets the upper end of the range as 6, 7, or 9 through 255.

ttl-evasion-protection	Disables the TTL evasion protection. Do not enter this command it you want to prevent attacks that attempt to evade security policy. For example, an attacker can send a packet that passes policy with a very short TTL. When the TTL goes to zero, a router between the ASA and the endpoint drops the packet. It is at this point that the attacker can send a malicious packet with a long TTL that appears to the ASA to be a retransmission and is passed. To the endpoint host, however, it is the first packet that has been received by the attacker. In this case, an attacker is able to succeed without security preventing the attack.

Table 22-1 tcp-map Commands (continued)

Command	Notes
urgent-flag {**allow** | **clear**}	Sets the action for packets with the URG flag. The URG flag is used to indicate that the packet contains information that is of higher priority than other data within the stream. The TCP RFC is vague about the exact interpretation of the URG flag, therefore end systems handle urgent offsets in different ways, which may make the end system vulnerable to attacks. The **allow** keyword allows packets with the URG flag. (Default) The **clear** keyword clears the URG flag and allows the packet.
window-variation {**allow** | **drop**}	Sets the action for a connection that has changed its window size unexpectedly. The window size mechanism allows TCP to advertise a large window and to subsequently advertise a much smaller window without having accepted too much data. From the TCP specification, "shrinking the window" is strongly discouraged. When this condition is detected, the connection can be dropped. (Default) The **allow** keyword allows connections with a window variation. The **drop** keyword drops connections with a window variation.

Configuring Connection Settings

To set connection settings, perform the following steps.

Detailed Steps

	Command	Purpose

Step 1	`class-map name`
	Example:
	`ciscoasa(config)# class-map bypass_traffic`
		Creates a class map to identify the traffic for which you want to disable stateful firewall inspection.

Step 2	`match parameter`
	Example:
	`ciscoasa(config-cmap)# match access-list bypass`
		Specifies the traffic in the class map. See the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12 for more information.

	Command	Purpose

Step 3	**`policy-map`** *`name`*
	Example:
	`ciscoasa(config)# policy-map tcp_bypass_policy`
		Adds or edits a policy map that sets the actions to take with the class map traffic.

Step 4	**`class`** *`name`*
	Example:
	`ciscoasa(config-pmap)# class bypass_traffic`
		Identifies the class map created in Step 1

Step 5	Do one or more of the following:	

Command	Purpose

set connection {[**conn-max** *n*] [**embryonic-conn-max** *n*] [**per-client-embryonic-max** *n*] [**per-client-max** *n*] [**random-sequence-number** {**enable** | **disable**}]}
Example:
`ciscoasa(config-pmap-c)# set connection conn-max 256 random-sequence-number disable`
	Sets maximum connection limits or whether TCP sequence randomization is enabled.
	The **conn-max** *n* argument sets the maximum number of simultaneous TCP and/or UDP connections that are allowed, between 0 and 2000000. The default is 0, which allows unlimited connections.
	If two servers are configured to allow simultaneous TCP and/or UDP connections, the connection limit is applied to each configured server separately.
	When configured under a class, this argument restricts the maximum number of simultaneous connections that are allowed for the entire class. In this case, one attack host can consume all the connections and leave none of the rest of the hosts matched in the ACL under the class.
	The **embryonic-conn-max** *n* argument sets the maximum number of simultaneous embryonic connections allowed, between 0 and 2000000. The default is 0, which allows unlimited connections.
	The **per-client-embryonic-max** *n* argument sets the maximum number of simultaneous embryonic connections allowed per client, between 0 and 2000000. The default is 0, which allows unlimited connections.
	The **per-client-max** *n* argument sets the maximum number of simultaneous connections allowed per client, between 0 and 2000000. The default is 0, which allows unlimited connections. When configured under a class, this argument restricts the maximum number of simultaneous connections that are allowed for each host that is matched through an ACL under the class.
	The **random-sequence-number** {**enable** | **disable**} keyword enables or disables TCP sequence number randomization. See the "TCP Sequence Randomization" section on page 22-3 section for more information.
	You can enter this command all on one line (in any order), or you can enter each attribute as a separate command. The ASA combines the command into one line in the running configuration.
	Note For management traffic, you can only set the **conn-max** and **embryonic-conn-max** keywords.

Command	Purpose

set connection timeout {[**embryonic** *hh*:*mm*:*ss*] {**idle** *hh*:*mm*:*ss* [**reset**]] [**half-closed** *hh*:*mm*:*ss*] [dcd *hh*:*mm*:*ss* [*max_retries*]]}
Example:
`ciscoasa(config-pmap-c)# set connection timeout idle 2:0:0 embryonic 0:40:0 half-closed 0:20:0 dcd`
	Sets connection timeouts. For global timeouts, see the **timout** command in the command reference.
	The **embryonic** *hh:mm:ss* keyword sets the timeout period until a TCP embryonic (half-open) connection is closed, between 0:0:5 and 1193:00:00. The default is 0:0:30. You can also set this value to 0, which means the connection never times out.
	The **idle** *hh:mm:ss* keyword sets the idle timeout period after which an established connection of any protocol closes, between 0:0:1 and 1193:0:0. The default is 1:0:0. You can also set this value to 0, which means the connection never times out. For TCP traffic, the **reset** keyword sends a reset to TCP endpoints when the connection times out.
	The **half-closed** *hh:mm:ss* keyword sets the idle timeout period until a half-closed connection is closed, between 0:5:0 (for 9.1(1) and earlier) or 0:0:30 (for 9.1(2) and later) and 1193:0:0. The default is 0:10:0. Half-closed connections are not affected by DCD. Also, the ASA does not send a reset when taking down half-closed connections.
	The **dcd** keyword enables DCD. DCD detects a dead connection and allows it to expire, without expiring connections that can still handle traffic. You configure DCD when you want idle, but valid connections to persist. After a TCP connection times out, the ASA sends DCD probes to the end hosts to determine the validity of the connection. If one of the end hosts fails to respond after the maximum retries are exhausted, the ASA frees the connection. If both end hosts respond that the connection is valid, the ASA updates the activity timeout to the current time and reschedules the idle timeout accordingly. The *retry-interval* sets the time duration in *hh*:*mm*:*ss* format to wait after each unresponsive DCD probe before sending another probe, between 0:0:1 and 24:0:0. The default is 0:0:15. The *max-retries* sets the number of consecutive failed retries for DCD before declaring the connection as dead. The minimum value is 1 and the maximum value is 255. The default is 5.
	The default **tcp** idle timeout is 1 hour.
	The default **udp** idle timeout is 2 minutes.
	The default **icmp** idle timeout is 2 seconds.
	The default **esp** and **ha** idle timeout is 30 seconds.
	For all other protocols, the default idle timeout is 2 minutes.
	To never time out, enter 0:0:0.
	You can enter this command all on one line (in any order), or you can enter each attribute as a separate command. The command is combined onto one line in the running configuration.
	Note This command is not available for management traffic.

	Command	Purpose
	set connection advanced-options *tcp-map-name* **Example:** `ciscoasa(config-pmap-c)# set connection advanced-options tcp_map1`	Customizes the TCP normalizer. See the "Customizing the TCP Normalizer with a TCP Map" section on page 22-6 to create a TCP map.
	set connection advanced-options tcp-state-bypass **Example:** `ciscoasa(config-pmap-c)# set connection advanced-options tcp-state-bypass`	Enables TCP state bypass.

Step 6	**service-policy** *policymap_name* {**global** | **interface** *interface_name*}
	Example:
	`ciscoasa(config)# service-policy tcp_bypass_policy outside`
		Activates the policy map on one or more interfaces. **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

Monitoring Connection Settings

To monitor TCP state bypass, perform one of the following tasks:

Command	Purpose
show conn	If you use the **show conn** command, the display for connections that use TCP state bypass includes the flag "b."

Configuration Examples for Connection Settings

This section includes the following topics:

- Configuration Examples for Connection Limits and Timeouts, page 22-15
- Configuration Examples for TCP State Bypass, page 22-16
- Configuration Examples for TCP Normalization, page 22-16

Configuration Examples for Connection Limits and Timeouts

The following example sets the connection limits and timeouts for all traffic:

```
ciscoasa(config)# class-map CONNS
ciscoasa(config-cmap)# match any
ciscoasa(config-cmap)# policy-map CONNS
ciscoasa(config-pmap)# class CONNS
ciscoasa(config-pmap-c)# set connection conn-max 1000 embryonic-conn-max 3000
ciscoasa(config-pmap-c)# set connection timeout idle 2:0:0 embryonic 0:40:0 half-closed
0:20:0 dcd
ciscoasa(config-pmap-c)# service-policy CONNS interface outside
```

You can enter **set connection** commands with multiple parameters or you can enter each parameter as a separate command. The ASA combines the commands into one line in the running configuration. For example, if you entered the following two commands in class configuration mode:

```
ciscoasa(config-pmap-c)# set connection conn-max 600
ciscoasa(config-pmap-c)# set connection embryonic-conn-max 50
```

the output of the **show running-config policy-map** command would display the result of the two commands in a single, combined command:

```
set connection conn-max 600 embryonic-conn-max 50
```

Configuration Examples for TCP State Bypass

The following is a sample configuration for TCP state bypass:

```
ciscoasa(config)# access-list tcp_bypass extended permit tcp 10.1.1.0 255.255.255.224 any

ciscoasa(config)# class-map tcp_bypass
ciscoasa(config-cmap)# description "TCP traffic that bypasses stateful firewall"
ciscoasa(config-cmap)# match access-list tcp_bypass

ciscoasa(config-cmap)# policy-map tcp_bypass_policy
ciscoasa(config-pmap)# class tcp_bypass
ciscoasa(config-pmap-c)# set connection advanced-options tcp-state-bypass

ciscoasa(config-pmap-c)# service-policy tcp_bypass_policy outside

ciscoasa(config-pmap-c)# static (inside,outside) 209.165.200.224 10.1.1.0 netmask
255.255.255.224
```

Configuration Examples for TCP Normalization

For example, to allow urgent flag and urgent offset packets for all traffic sent to the range of TCP ports between the well known FTP data port and the Telnet port, enter the following commands:

```
ciscoasa(config)# tcp-map tmap
ciscoasa(config-tcp-map)# urgent-flag allow
ciscoasa(config-tcp-map)# class-map urg-class
ciscoasa(config-cmap)# match port tcp range ftp-data telnet
ciscoasa(config-cmap)# policy-map pmap
ciscoasa(config-pmap)# class urg-class
ciscoasa(config-pmap-c)# set connection advanced-options tmap
ciscoasa(config-pmap-c)# service-policy pmap global
```

Feature History for Connection Settings

Table 22-2 lists each feature change and the platform release in which it was implemented.

Table 22-2 Feature History for Connection Settings

Feature Name	Platform Releases	Feature Information
TCP state bypass	8.2(1)	This feature was introduced. The following command was introduced: **set connection advanced-options tcp-state-bypass**.
Connection timeout for all protocols	8.2(2)	The idle timeout was changed to apply to all protocols, not just TCP. The following command was modified: **set connection timeout**
Timeout for connections using a backup static route	8.2(5)/8.4(2)	When multiple static routes exist to a network with different metrics, the ASA uses the one with the best metric at the time of connection creation. If a better route becomes available, then this timeout lets connections be closed so a connection can be reestablished to use the better route. The default is 0 (the connection never times out). To take advantage of this feature, change the timeout to a new value. We modified the following command: **timeout floating-conn**.
Configurable timeout for PAT xlate	8.4(3)	When a PAT xlate times out (by default after 30 seconds), and the ASA reuses the port for a new translation, some upstream routers might reject the new connection because the previous connection might still be open on the upstream device. The PAT xlate timeout is now configurable, to a value between 30 seconds and 5 minutes. We introduced the following command: **timeout pat-xlate**. *This feature is not available in 8.5(1) or 8.6(1).*

Table 22-2 Feature History for Connection Settings (continued)

Feature Name	Platform Releases	Feature Information
Increased maximum connection limits for service policy rules	9.0(1)	The maximum number of connections for service policy rules was increased from 65535 to 2000000. We modified the following commands: **set connection conn-max**, **set connection embryonic-conn-max**, **set connection per-client-embryonic-max**, **set connection per-client-max**.
Decreased the half-closed timeout minimum value to 30 seconds	9.1(2)	The half-closed timeout minimum value for both the global timeout and connection timeout was lowered from 5 minutes to 30 seconds to provide better DoS protection. We modified the following commands: **set connection timeout half-closed**, **timeout half-closed**.

Configuring QoS

Have you ever participated in a long-distance phone call that involved a satellite connection? The conversation might be interrupted with brief, but perceptible, gaps at odd intervals. Those gaps are the time, called the latency, between the arrival of packets being transmitted over the network. Some network traffic, such as voice and video, cannot tolerate long latency times. Quality of service (QoS) is a feature that lets you give priority to critical traffic, prevent bandwidth hogging, and manage network bottlenecks to prevent packet drops.

Note For the ASASM, we suggest performing QoS on the switch instead of the ASASM. Switches have more capability in this area.

This chapter describes how to apply QoS policies and includes the following sections:

- Information About QoS, page 23-1
- Licensing Requirements for QoS, page 23-5
- Guidelines and Limitations, page 23-5
- Configuring QoS, page 23-6
- Monitoring QoS, page 23-16
- Feature History for QoS, page 23-19

Information About QoS

You should consider that in an ever-changing network environment, QoS is not a one-time deployment, but an ongoing, essential part of network design.

This section describes the QoS features supported by the ASA and includes the following topics:

- Supported QoS Features, page 23-2
- What is a Token Bucket?, page 23-2
- Information About Policing, page 23-3
- Information About Priority Queuing, page 23-3
- Information About Traffic Shaping, page 23-4
- DSCP and DiffServ Preservation, page 23-5

Supported QoS Features

The ASA supports the following QoS features:

- Policing—To prevent individual flows from hogging the network bandwidth, you can limit the maximum bandwidth used per flow. See the "Information About Policing" section on page 23-3 for more information.
- Priority queuing—For critical traffic that cannot tolerate latency, such as Voice over IP (VoIP), you can identify traffic for Low Latency Queuing (LLQ) so that it is always transmitted ahead of other traffic. See the "Information About Priority Queuing" section on page 23-3 for more information.
- Traffic shaping—If you have a device that transmits packets at a high speed, such as a ASA with Fast Ethernet, and it is connected to a low speed device such as a cable modem, then the cable modem is a bottleneck at which packets are frequently dropped. To manage networks with differing line speeds, you can configure the ASA to transmit packets at a fixed slower rate. See the "Information About Traffic Shaping" section on page 23-4 for more information.

What is a Token Bucket?

A token bucket is used to manage a device that regulates the data in a flow. For example, the regulator might be a traffic policer or a traffic shaper. A token bucket itself has no discard or priority policy. Rather, a token bucket discards tokens and leaves to the flow the problem of managing its transmission queue if the flow overdrives the regulator.

A token bucket is a formal definition of a rate of transfer. It has three components: a burst size, an average rate, and a time interval. Although the average rate is generally represented as bits per second, any two values may be derived from the third by the relation shown as follows:

average rate = burst size / time interval

Here are some definitions of these terms:

- Average rate—Also called the committed information rate (CIR), it specifies how much data can be sent or forwarded per unit time on average.
- Burst size—Also called the Committed Burst (Bc) size, it specifies in bits or bytes per burst how much traffic can be sent within a given unit of time to not create scheduling concerns. (For traffic shaping, it specifies bits per burst; for policing, it specifies bytes per burst.)
- Time interval—Also called the measurement interval, it specifies the time quantum in seconds per burst.

In the token bucket metaphor, tokens are put into the bucket at a certain rate. The bucket itself has a specified capacity. If the bucket fills to capacity, newly arriving tokens are discarded. Each token is permission for the source to send a certain number of bits into the network. To send a packet, the regulator must remove from the bucket a number of tokens equal in representation to the packet size.

If not enough tokens are in the bucket to send a packet, the packet either waits until the bucket has enough tokens (in the case of traffic shaping) or the packet is discarded or marked down (in the case of policing). If the bucket is already full of tokens, incoming tokens overflow and are not available to future packets. Thus, at any time, the largest burst a source can send into the network is roughly proportional to the size of the bucket.

Note that the token bucket mechanism used for traffic shaping has both a token bucket and a data buffer, or queue; if it did not have a data buffer, it would be a policer. For traffic shaping, packets that arrive that cannot be sent immediately are delayed in the data buffer.

For traffic shaping, a token bucket permits burstiness but bounds it. It guarantees that the burstiness is bounded so that the flow will never send faster than the token bucket capacity, divided by the time interval, plus the established rate at which tokens are placed in the token bucket. See the following formula:

(token bucket capacity in bits / time interval in seconds) + established rate in bps = maximum flow speed in bps

This method of bounding burstiness also guarantees that the long-term transmission rate will not exceed the established rate at which tokens are placed in the bucket.

Information About Policing

Policing is a way of ensuring that no traffic exceeds the maximum rate (in bits/second) that you configure, thus ensuring that no one traffic flow or class can take over the entire resource. When traffic exceeds the maximum rate, the ASA drops the excess traffic. Policing also sets the largest single burst of traffic allowed.

Information About Priority Queuing

LLQ priority queuing lets you prioritize certain traffic flows (such as latency-sensitive traffic like voice and video) ahead of other traffic.

The ASA supports two types of priority queuing:

- Standard priority queuing—Standard priority queuing uses an LLQ priority queue on an interface (see the "Configuring the Standard Priority Queue for an Interface" section on page 23-8), while all other traffic goes into the "best effort" queue. Because queues are not of infinite size, they can fill and overflow. When a queue is full, any additional packets cannot get into the queue and are dropped. This is called *tail drop*. To avoid having the queue fill up, you can increase the queue buffer size. You can also fine-tune the maximum number of packets allowed into the transmit queue. These options let you control the latency and robustness of the priority queuing. Packets in the LLQ queue are always transmitted before packets in the best effort queue.
- Hierarchical priority queuing—Hierarchical priority queuing is used on interfaces on which you enable a traffic shaping queue. A subset of the shaped traffic can be prioritized. The standard priority queue is not used. See the following guidelines about hierarchical priority queuing:
  - Priority packets are always queued at the head of the shape queue so they are always transmitted ahead of other non-priority queued packets.
  - Priority packets are never dropped from the shape queue unless the sustained rate of priority traffic exceeds the shape rate.
  - For IPsec-encrypted packets, you can only match traffic based on the DSCP or precedence setting.
  - IPsec-over-TCP is not supported for priority traffic classification.

Information About Traffic Shaping

Traffic shaping is used to match device and link speeds, thereby controlling packet loss, variable delay, and link saturation, which can cause jitter and delay.

Note Traffic shaping is only supported on the ASA 5505, 5510, 5520, 5540, and 5550.

- Traffic shaping must be applied to all outgoing traffic on a physical interface or in the case of the ASA 5505, on a VLAN. You cannot configure traffic shaping for specific types of traffic.
- Traffic shaping is implemented when packets are ready to be transmitted on an interface, so the rate calculation is performed based on the actual size of a packet to be transmitted, including all the possible overhead such as the IPsec header and L2 header.
- The shaped traffic includes both through-the-box and from-the-box traffic.
- The shape rate calculation is based on the standard token bucket algorithm. The token bucket size is twice the Burst Size value. See the "What is a Token Bucket?" section on page 23-2.
- When bursty traffic exceeds the specified shape rate, packets are queued and transmitted later. Following are some characteristics regarding the shape queue (for information about hierarchical priority queuing, see the "Information About Priority Queuing" section on page 23-3):
  - The queue size is calculated based on the shape rate. The queue can hold the equivalent of 200-milliseconds worth of shape rate traffic, assuming a 1500-byte packet. The minimum queue size is 64.
  - When the queue limit is reached, packets are tail-dropped.
  - Certain critical keep-alive packets such as OSPF Hello packets are never dropped.
  - The time interval is derived by *time_interval* = *burst_size* / *average_rate*. The larger the time interval is, the burstier the shaped traffic might be, and the longer the link might be idle. The effect can be best understood using the following exaggerated example:

    Average Rate = 1000000

    Burst Size = 1000000

    In the above example, the time interval is 1 second, which means, 1 Mbps of traffic can be bursted out within the first 10 milliseconds of the 1-second interval on a 100 Mbps FE link and leave the remaining 990 milliseconds idle without being able to send any packets until the next time interval. So if there is delay-sensitive traffic such as voice traffic, the Burst Size should be reduced compared to the average rate so the time interval is reduced.

How QoS Features Interact

You can configure each of the QoS features alone if desired for the ASA. Often, though, you configure multiple QoS features on the ASA so you can prioritize some traffic, for example, and prevent other traffic from causing bandwidth problems.

See the following supported feature combinations per interface:

- Standard priority queuing (for specific traffic) + Policing (for the rest of the traffic).

  You cannot configure priority queuing and policing for the same set of traffic.
- Traffic shaping (for all traffic on an interface) + Hierarchical priority queuing (for a subset of traffic).

You cannot configure traffic shaping and standard priority queuing for the same interface; only hierarchical priority queuing is allowed. For example, if you configure standard priority queuing for the global policy, and then configure traffic shaping for a specific interface, the feature you configured last is rejected because the global policy overlaps the interface policy.

Typically, if you enable traffic shaping, you do not also enable policing for the same traffic, although the ASA does not restrict you from configuring this.

DSCP and DiffServ Preservation

- DSCP markings are preserved on all traffic passing through the ASA.
- The ASA does not locally mark/remark any classified traffic, but it honors the Expedited Forwarding (EF) DSCP bits of every packet to determine if it requires “priority” handling and will direct those packets to the LLQ.
- DiffServ marking is preserved on packets when they traverse the service provider backbone so that QoS can be applied in transit (QoS tunnel pre-classification).

Licensing Requirements for QoS

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single context mode only. Does not support multiple context mode.

Firewall Mode Guidelines

Supported in routed firewall mode only. Does not support transparent firewall mode.

IPv6 Guidelines

Does not support IPv6.

Model Guidelines

- Traffic shaping is only supported on the ASA 5505, 5510, 5520, 5540, and 5550. Multi-core models (such as the ASA 5500-X) do not support shaping.
- (ASA 5580) You cannot create a standard priority queue for a Ten Gigabit Ethernet interface. **Note**: For the ASA 5585-X, standard priority queuing is supported on a Ten Gigabit Interface.

- (ASA 5512-X through ASA 5555-X) Priority queuing is not supported on the Management 0/0 interface.
- (ASASM) Only policing is supported.

Additional Guidelines and Limitations

- QoS is applied unidirectionally; only traffic that enters (or exits, depending on the QoS feature) the interface to which you apply the policy map is affected. See the "Feature Directionality" section on page 1-2 for more information.
- For traffic shaping, you can only use the **class-default** class map, which is automatically created by the ASA, and which matches all traffic.
- For priority traffic, you cannot use the **class-default** class map.
- For hierarchical priority queuing, for encrypted VPN traffic, you can only match traffic based on the DSCP or precedence setting; you cannot match a tunnel group.
- For hierarchical priority queuing, IPsec-over-TCP traffic is not supported.
- You cannot configure traffic shaping and standard priority queuing for the same interface; only hierarchical priority queuing is allowed.
- For standard priority queuing, the queue must be configured for a physical interface or, for the ASA 5505 or ASASM, a VLAN.
- For policing, to-the-box traffic is not supported.
- For policing, traffic to and from a VPN tunnel bypass interface is not supported.
- For policing, when you match a tunnel group class map, only outbound policing is supported.

Configuring QoS

This section includes the following topics:

- Determining the Queue and TX Ring Limits for a Standard Priority Queue, page 23-7
- Configuring the Standard Priority Queue for an Interface, page 23-8
- Configuring a Service Rule for Standard Priority Queuing and Policing, page 23-9
- Configuring a Service Rule for Traffic Shaping and Hierarchical Priority Queuing, page 23-13

Determining the Queue and TX Ring Limits for a Standard Priority Queue

To determine the priority queue and TX ring limits, use the worksheets below.

Table 23-1 shows how to calculate the priority queue size. Because queues are not of infinite size, they can fill and overflow. When a queue is full, any additional packets cannot get into the queue and are dropped (called *tail drop*). To avoid having the queue fill up, you can adjust the queue buffer size according to the "Configuring the Standard Priority Queue for an Interface" section on page 23-8.

Table 23-1 Queue Limit Worksheet

Step 1	__________ *Outbound bandwidth (Mbps or Kbps)*[1]	Mbps	x	**125**	=	__________ *# of bytes/ms*		
		Kbps	x	**.125**	=	__________ *# of bytes/ms*		
Step 2	__________ *# of bytes/ms from Step 1*		÷	__________ *Average packet size (bytes)*[2]	x	__________ *Delay (ms)*[3]	=	__________ *Queue limit (# of packets)*

[1] For example, DSL might have an uplink speed of 768 Kbps. Check with your provider.

[2] Determine this value from a codec or sampling size. For example, for VoIP over VPN, you might use 160 bytes. We recommend 256 bytes if you do not know what size to use.

[3] The delay depends on your application. For example, the recommended maximum delay for VoIP is 200 ms. We recommend 500 ms if you do not know what delay to use.

Table 23-2 shows how to calculate the TX ring limit. This limit determines the maximum number of packets allowed into the Ethernet transmit driver before the driver pushes back to the queues on the interface to let them buffer packets until the congestion clears. This setting guarantees that the hardware-based transmit ring imposes a limited amount of extra latency for a high-priority packet.

Table 23-2 TX Ring Limit Worksheet

Step 1	__________ *Outbound bandwidth (Mbps or Kbps)*[1]	Mbps	x	**125**	=	__________ *# of bytes/ms*		
		Kbps	x	**0.125**	=	__________ *# of bytes/ms*		
Step 2	__________ *# of bytes/ms from Step 1*		÷	__________ *Maximum packet size (bytes)*[2]	x	__________ *Delay (ms)*[3]	=	__________ *TX ring limit (# of packets)*

[1] For example, DSL might have an uplink speed of 768 Kbps.Check with your provider.

2. Typically, the maximum size is 1538 bytes, or 1542 bytes for tagged Ethernet. If you allow jumbo frames (if supported for your platform), then the packet size might be larger.
3. The delay depends on your application. For example, to control jitter for VoIP, you should use 20 ms.

Configuring the Standard Priority Queue for an Interface

If you enable standard priority queuing for traffic on a physical interface, then you need to also create the priority queue on each interface. Each physical interface uses two queues: one for priority traffic, and the other for all other traffic. For the other traffic, you can optionally configure policing.

Note The standard priority queue is not required for hierarchical priority queuing with traffic shaping; see the "Information About Priority Queuing" section on page 23-3 for more information.

Restrictions

- (ASASM) The ASASM does not support priority queuing.
- (ASA 5580) You cannot create a standard priority queue for a Ten Gigabit Ethernet interface. **Note**: For the ASA 5585-X, standard priority queuing is supported on a Ten Gigabit Interface.
- (ASA 5512-X through ASA 5555-X) Priority queuing is not supported on the Management 0/0 interface.

Detailed Steps

	Command	Purpose
Step 1	**priority-queue** *interface_name* **Example:** `ciscoasa(config)# priority-queue inside`	Creatse the priority queue, where the *interface_name* argument specifies the physical interface name on which you want to enable the priority queue, or for the ASA 5505 or ASASM, the VLAN interface name.

	Command	Purpose
Step 2	**queue-limit** *number_of_packets* **Example:** `ciscoasa(config-priority-queue)# queue-limit 260`	Changes the size of the priority queues. The default queue limit is 1024 packets. Because queues are not of infinite size, they can fill and overflow. When a queue is full, any additional packets cannot get into the queue and are dropped (called *tail drop*). To avoid having the queue fill up, you can use the **queue-limit** command to increase the queue buffer size. The upper limit of the range of values for the **queue-limit** command is determined dynamically at run time. To view this limit, enter **queue-limit ?** on the command line. The key determinants are the memory needed to support the queues and the memory available on the device. The **queue-limit** that you specify affects both the higher priority low-latency queue and the best effort queue.
Step 3	**tx-ring-limit** *number_of_packets* **Example:** `ciscoasa(config-priority-queue)# tx-ring-limit 3`	Specifies the depth of the priority queues. The default tx-ring-limit is 128 packets. This command sets the maximum number of low-latency or normal priority packets allowed into the Ethernet transmit driver before the driver pushes back to the queues on the interface to let them buffer packets until the congestion clears. This setting guarantees that the hardware-based transmit ring imposes a limited amount of extra latency for a high-priority packet. The upper limit of the range of values for the **tx-ring-limit** command is determined dynamically at run time. To view this limit, enter **tx-ring-limit ?** on the command line. The key determinants are the memory needed to support the queues and the memory available on the device. The **tx-ring-limit** that you specify affects both the higher priority low-latency queue and the best-effort queue.

Examples

The following example establishes a priority queue on interface “outside” (the GigabitEthernet0/1 interface), with the default queue-limit and tx-ring-limit:

```
ciscoasa(config)# priority-queue outside
```

The following example establishes a priority queue on the interface “outside” (the GigabitEthernet0/1 interface), sets the queue-limit to 260 packets, and sets the tx-ring-limit to 3:

```
ciscoasa(config)# priority-queue outside
ciscoasa(config-priority-queue)# queue-limit 260
ciscoasa(config-priority-queue)# tx-ring-limit 3
```

Configuring a Service Rule for Standard Priority Queuing and Policing

You can configure standard priority queuing and policing for different class maps within the same policy map. See the “How QoS Features Interact” section on page 23-4 for information about valid QoS configurations.

To create a policy map, perform the following steps.

Restrictions

- You cannot use the **class-default** class map for priority traffic.
- You cannot configure traffic shaping and standard priority queuing for the same interface; only hierarchical priority queuing is allowed.
- (ASASM) The ASASM only supports policing.
- For policing, to-the-box traffic is not supported.
- For policing, traffic to and from a VPN tunnel bypass interface is not supported.
- For policing, when you match a tunnel group class map, only outbound policing is supported.

Guidelines

- For priority traffic, identify only latency-sensitive traffic.
- For policing traffic, you can choose to police all other traffic, or you can limit the traffic to certain types.

Detailed Steps

	Command	Purpose

Step 1	**`class-map`** *`priority_map_name`*
	Example:
	`ciscoasa(config)# class-map priority_traffic`
		For priority traffic, creates a class map to identify the traffic for which you want to perform priority queuing.

Step 2	**`match`** *`parameter`*
	Example:
	`ciscoasa(config-cmap)# match access-list priority`
		Specifies the traffic in the class map. See the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12 for more information.

Step 3	**`class-map`** *`policing_map_name`*
	Example:
	`ciscoasa(config)# class-map policing_traffic`
		For policing traffic, creates a class map to identify the traffic for which you want to perform policing.

Step 4	**`match`** *`parameter`*
	Example:
	`ciscoasa(config-cmap)# match access-list policing`
		Specifies the traffic in the class map. See the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12 for more information.

Step 5	**`policy-map`** *`name`*
	Example:
	`ciscoasa(config)# policy-map QoS_policy`
		Adds or edits a policy map.

	Command	Purpose

Step 6	**class** *priority_map_name*
	Example:
	`ciscoasa(config-pmap)# class priority_class`
		Identifies the class map you created for prioritized traffic in Step 1.

Step 7	**priority** **Example:** `ciscoasa(config-pmap-c)# priority`	Configures priority queuing for the class.

Step 8	**class** *policing_map_name*
	Example:
	`ciscoasa(config-pmap)# class policing_class`
		Identifies the class map you created for policed traffic in Step 3.

Step 9	**police** {**output** | **input**} *conform-rate* [*conform-burst*] [**conform-action** [**drop** | **transmit**]] [**exceed-action** [**drop** | **transmit**]] **Example:** `ciscoasa(config-pmap-c)# police output 56000 10500`	Configures policing for the class. See the followingoptions: • *conform-burst argument*—Specifies the maximum number of instantaneous bytes allowed in a sustained burst before throttling to the conforming rate value, between 1000 and 512000000 bytes. • **conform-action**—Sets the action to take when the rate is less than the *conform_burst* value. • *conform-rate*—Sets the rate limit for this traffic flow; between 8000 and 2000000000 bits per second.] • **drop**—Drops the packet. • **exceed-action**—Sets the action to take when the rate is between the *conform-rate* value and the *conform-burst* value. • **input**—Enables policing of traffic flowing in the input direction. • **output**—Enables policing of traffic flowing in the output direction. • **transmit**—Transmits the packet.

Step 10	**service-policy** *policymap_name* {**global** | **interface** *interface_name*}
	Example:
	`ciscoasa(config)# service-policy QoS_policy interface inside`
		Activates the policy map on one or more interfaces. **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

Examples

Example 23-1 Class Map Examples for VPN Traffic

In the following example, the **class-map** command classifies all non-tunneled TCP traffic, using an ACL named tcp_traffic:

```
ciscoasa(config)# access-list tcp_traffic permit tcp any any
ciscoasa(config)# class-map tcp_traffic
ciscoasa(config-cmap)# match access-list tcp_traffic
```

In the following example, other, more specific match criteria are used for classifying traffic for specific, security-related tunnel groups. These specific match criteria stipulate that a match on tunnel-group (in this case, the previously-defined Tunnel-Group-1) is required as the first match characteristic to classify traffic for a specific tunnel, and it allows for an additional match line to classify the traffic (IP differential services code point, expedited forwarding).

```
ciscoasa(config)# class-map TG1-voice
ciscoasa(config-cmap)# match tunnel-group tunnel-grp1
ciscoasa(config-cmap)# match dscp ef
```

In the following example, the **class-map** command classifies both tunneled and non-tunneled traffic according to the traffic type:

```
ciscoasa(config)# access-list tunneled extended permit ip 10.10.34.0 255.255.255.0 
192.168.10.0 255.255.255.0
ciscoasa(config)# access-list non-tunneled extended permit tcp any any
ciscoasa(config)# tunnel-group tunnel-grp1 type IPsec_L2L

ciscoasa(config)# class-map browse
ciscoasa(config-cmap)# description "This class-map matches all non-tunneled tcp traffic."
ciscoasa(config-cmap)# match access-list non-tunneled

ciscoasa(config-cmap)# class-map TG1-voice
ciscoasa(config-cmap)# description "This class-map matches all dscp ef traffic for 
tunnel-grp 1."
ciscoasa(config-cmap)# match dscp ef
ciscoasa(config-cmap)# match tunnel-group tunnel-grp1

ciscoasa(config-cmap)# class-map TG1-BestEffort
ciscoasa(config-cmap)# description "This class-map matches all best-effort traffic for 
tunnel-grp1."
ciscoasa(config-cmap)# match tunnel-group tunnel-grp1
ciscoasa(config-cmap)# match flow ip destination-address
```

The following example shows a way of policing a flow within a tunnel, provided the classed traffic is not specified as a tunnel, but does go *through* the tunnel. In this example, 192.168.10.10 is the address of the host machine on the private side of the remote tunnel, and the ACL is named “host-over-l2l”. By creating a class-map (named “host-specific”), you can then police the “host-specific” class before the LAN-to-LAN connection polices the tunnel. In this example, the “host-specific” traffic is rate-limited before the tunnel, then the tunnel is rate-limited:

```
ciscoasa(config)# access-list host-over-l2l extended permit ip any host 192.168.10.10
ciscoasa(config)# class-map host-specific
ciscoasa(config-cmap)# match access-list host-over-l2l
```

The following example builds on the configuration developed in the previous section. As in the previous example, there are two named class-maps: tcp_traffic and TG1-voice.

```
ciscoasa(config)# class-map TG1-best-effort
ciscoasa(config-cmap)# match tunnel-group Tunnel-Group-1
ciscoasa(config-cmap)# match flow ip destination-address
```

Adding a third class map provides a basis for defining a tunneled and non-tunneled QoS policy, as follows, which creates a simple QoS policy for tunneled and non-tunneled traffic, assigning packets of the class TG1-voice to the low latency queue and setting rate limits on the tcp_traffic and TG1-best-effort traffic flows.

Example 23-2 Priority and Policing Example

In this example, the maximum rate for traffic of the tcp_traffic class is 56,000 bits/second and a maximum burst size of 10,500 bytes per second. For the TC1-BestEffort class, the maximum rate is 200,000 bits/second, with a maximum burst of 37,500 bytes/second. Traffic in the TC1-voice class has no policed maximum speed or burst rate because it belongs to a priority class.

```
ciscoasa(config)# access-list tcp_traffic permit tcp any any
ciscoasa(config)# class-map tcp_traffic
ciscoasa(config-cmap)# match access-list tcp_traffic

ciscoasa(config)# class-map TG1-voice
ciscoasa(config-cmap)# match tunnel-group tunnel-grp1
ciscoasa(config-cmap)# match dscp ef

ciscoasa(config-cmap)# class-map TG1-BestEffort
ciscoasa(config-cmap)# match tunnel-group tunnel-grp1
ciscoasa(config-cmap)# match flow ip destination-address

ciscoasa(config)# policy-map qos
ciscoasa(config-pmap)# class tcp_traffic
ciscoasa(config-pmap-c)# police output 56000 10500

ciscoasa(config-pmap-c)# class TG1-voice
ciscoasa(config-pmap-c)# priority

ciscoasa(config-pmap-c)# class TG1-best-effort
ciscoasa(config-pmap-c)# police output 200000 37500

ciscoasa(config-pmap-c)# class class-default
ciscoasa(config-pmap-c)# police output 1000000 37500

ciscoasa(config-pmap-c)# service-policy qos global
```

Configuring a Service Rule for Traffic Shaping and Hierarchical Priority Queuing

You can configure traffic shaping for all traffic on an interface, and optionally hierarchical priority queuing for a subset of latency-sensitive traffic.

This section includes the following topics:

- (Optional) Configuring the Hierarchical Priority Queuing Policy, page 23-13
- Configuring the Service Rule, page 23-14

(Optional) Configuring the Hierarchical Priority Queuing Policy

You can optionally configure priority queuing for a subset of latency-sensitive traffic.

Guidelines

- One side-effect of priority queuing is packet re-ordering. For IPsec packets, out-of-order packets that are not within the anti-replay window generate warning syslog messages. These warnings are false alarms in the case of priority queuing. You can configure the IPsec anti-replay window size to avoid possible false alarms. See the **crypto ipsec security-association replay** command in the command reference.

- For hierarchical priority queuing, you do not need to create a priority queue on an interface.

Restrictions

- For hierarchical priority queuing, for encrypted VPN traffic, you can only match traffic based on the DSCP or precedence setting; you cannot match a tunnel group.
- For hierarchical priority queuing, IPsec-over-TCP traffic is not supported.

Detailed Steps

	Command	Purpose

Step 1	**class-map** *priority_map_name*
	Example:
	`ciscoasa(config)# class-map priority_traffic`
		For hierarchical priority queuing, creates a class map to identify the traffic for which you want to perform priority queuing.

Step 2	**match** *parameter* **Example:** `ciscoasa(config-cmap)# match access-list priority`	Specifies the traffic in the class map. See the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12 for more information. For encrypted VPN traffic, you can only match traffic based on the DSCP or precedence setting; you cannot match a tunnel group.
Step 3	**policy-map** *priority_map_name* **Example:** `ciscoasa(config)# policy-map priority-sub-policy`	Creates a policy map.

Step 4	**class** *priority_map_name*
	Example:
	`ciscoasa(config-pmap)# class priority-sub-map`
		Specifies the class map you created in Step 1.

Step 5	**priority** **Example:** `ciscoasa(config-pmap-c)# priority`	Applies the priority queuing action to a class map. **Note** This policy has not yet been activated. You must activate it as part of the shaping policy. See the "Configuring the Service Rule" section on page 23-14.

Configuring the Service Rule

To configure traffic shaping and optional hiearchical priority queuing, perform the following steps.

Restrictions

- Traffic shaping is only supported on the ASA 5505, 5510, 5520, 5540, and 5550. Multi-core models (such as the ASA 5500-X) do not support shaping.
- For traffic shaping, you can only use the **class-default** class map, which is automatically created by the ASA, and which matches all traffic.

- You cannot configure traffic shaping and standard priority queuing for the same interface; only hierarchical priority queuing is allowed. See the "How QoS Features Interact" section on page 23-4 for information about valid QoS configurations.
- You cannot configure traffic shaping in the global policy.

Detailed Steps

	Command	Purpose
Step 1	**policy-map** *name* **Example:** ciscoasa(config)# policy-map shape_policy	Adds or edits a policy map. This policy map must be different from the hierarchical priority-queuing map.
Step 2	**class class-default** **Example:** ciscoasa(config-pmap)# class class-default	Identifies all traffic for traffic shaping; you can only use the **class-default** class map, which is defined as **match any**, because the ASA requires all traffic to be matched for traffic shaping.
Step 3	**shape average** *rate* [*burst_size*] **Example:** ciscoasa(config-pmap-c)# shape average 70000 4000	Enables traffic shaping, where the **average** *rate* argument sets the average rate of traffic in bits per second over a given fixed time period, between 64000 and 154400000. Specify a value that is a multiple of 8000. See the "Information About Traffic Shaping" section on page 23-4 for more information about how the time period is calculated. The *burst_size* argument sets the average burst size in bits that can be transmitted over a given fixed time period, between 2048 and 154400000. Specify a value that is a multiple of 128. If you do not specify the *burst_size*, the default value is equivalent to 4-milliseconds of traffic at the specified average rate. For example, if the average rate is 1000000 bits per second, 4 ms worth = 1000000 * 4/1000 = 4000.
Step 4	(Optional) **service-policy** *priority_policy_map_name* **Example:** ciscoasa(config-pmap-c)# **service-policy priority-sub-policy**	Configures hierarchical priority queuing, where the *priority_policy_map_name* is the policy map you created for prioritized traffic in the "(Optional) Configuring the Hierarchical Priority Queuing Policy" section on page 23-13.
Step 5	**service-policy** *policymap_name* **interface** *interface_name* **Example:** ciscoasa(config)# service-policy shape-policy interface inside	Activates the shaping policy map on an interface.

Examples

The following example enables traffic shaping on the outside interface, and limits traffic to 2 Mbps; priority queuing is enabled for VoIP traffic that is tagged with DSCP EF and AF13 and for IKE traffic:

```
ciscoasa(config)# access-list ike permit udp any any eq 500
ciscoasa(config)# class-map ike
ciscoasa(config-cmap)# match access-list ike

ciscoasa(config-cmap)# class-map voice_traffic
ciscoasa(config-cmap)# match dscp EF AF13

ciscoasa(config-cmap)# policy-map qos_class_policy
ciscoasa(config-pmap)# class voice_traffic
ciscoasa(config-pmap-c)# priority
ciscoasa(config-pmap-c)# class ike
ciscoasa(config-pmap-c)# priority

ciscoasa(config-pmap-c)# policy-map qos_outside_policy
ciscoasa(config-pmap)# class class-default
ciscoasa(config-pmap-c)# shape average 2000000 16000
ciscoasa(config-pmap-c)# service-policy qos_class_policy

ciscoasa(config-pmap-c)# service-policy qos_outside_policy interface outside
```

Monitoring QoS

This section includes the following topics:

- Viewing QoS Police Statistics, page 23-16
- Viewing QoS Standard Priority Statistics, page 23-17
- Viewing QoS Shaping Statistics, page 23-17
- Viewing QoS Standard Priority Queue Statistics, page 23-18

Viewing QoS Police Statistics

To view the QoS statistics for traffic policing, use the **show service-policy** command with the **police** keyword:

```
ciscoasa# show service-policy police
```

The following is sample output for the **show service-policy police** command:

```
ciscoasa# show service-policy police

Global policy:
    Service-policy: global_fw_policy

Interface outside:
    Service-policy: qos
        Class-map: browse
            police Interface outside:
                cir 56000 bps, bc 10500 bytes
                conformed 10065 packets, 12621510 bytes; actions: transmit
                exceeded 499 packets, 625146 bytes; actions: drop
                conformed 5600 bps, exceed 5016 bps
        Class-map: cmap2
            police Interface outside:
                cir 200000 bps, bc 37500 bytes
                conformed 17179 packets, 20614800 bytes; actions: transmit
                exceeded 617 packets, 770718 bytes; actions: drop
                conformed 198785 bps, exceed 2303 bps
```

Viewing QoS Standard Priority Statistics

To view statistics for service policies implementing the **priority** command, use the **show service-policy** command with the **priority** keyword:

```
ciscoasa# show service-policy priority
```

The following is sample output for the **show service-policy priority** command:

```
ciscoasa# show service-policy priority
Global policy:
    Service-policy: global_fw_policy
Interface outside:
    Service-policy: qos
        Class-map: TG1-voice
            Priority:
                Interface outside: aggregate drop 0, aggregate transmit 9383
```

Note “Aggregate drop” denotes the aggregated drop in this interface; “aggregate transmit” denotes the aggregated number of transmitted packets in this interface.

Viewing QoS Shaping Statistics

To view statistics for service policies implementing the **shape** command, use the **show service-policy** command with the **shape** keyword:

```
ciscoasa# show service-policy shape
```

The following is sample output for the **show service-policy shape** command:

```
ciscoasa# show service-policy shape
Interface outside
  Service-policy: shape
    Class-map: class-default

      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0

      shape (average) cir 2000000, bc 8000, be 8000
```

The following is sample output of the **show service policy shape** command, which includes service policies that include the **shape** command and the **service-policy** command that calls the hierarchical priority policy and the related statistics:

```
ciscoasa# show service-policy shape

Interface outside:
  Service-policy: shape
    Class-map: class-default

      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0

      shape (average) cir 2000000, bc 16000, be 16000
      Service-policy: voip
        Class-map: voip

          Queueing
          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
        Class-map: class-default

          queue limit 64 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0
```

Viewing QoS Standard Priority Queue Statistics

To display the priority-queue statistics for an interface, use the **show priority-queue statistics** command in privileged EXEC mode. The results show the statistics for both the best-effort (BE) queue and the low-latency queue (LLQ). The following example shows the use of the **show priority-queue statistics** command for the interface named test, and the command output.

```
ciscoasa# show priority-queue statistics test

Priority-Queue Statistics interface test

Queue Type         = BE
Packets Dropped    = 0
Packets Transmit   = 0
Packets Enqueued   = 0
Current Q Length   = 0
Max Q Length       = 0

Queue Type         = LLQ
Packets Dropped    = 0
Packets Transmit   = 0
Packets Enqueued   = 0
Current Q Length   = 0
Max Q Length       = 0
ciscoasa#
```

In this statistical report, the meaning of the line items is as follows:

- "Packets Dropped" denotes the overall number of packets that have been dropped in this queue.
- "Packets Transmit" denotes the overall number of packets that have been transmitted in this queue.
- "Packets Enqueued" denotes the overall number of packets that have been queued in this queue.
- "Current Q Length" denotes the current depth of this queue.
- "Max Q Length" denotes the maximum depth that ever occurred in this queue.

Feature History for QoS

Table 23-3 lists each feature change and the platform release in which it was implemented.

Table 23-3 Feature History for QoS

Feature Name	Platform Releases	Feature Information
Priority queuing and policing	7.0(1)	We introduced QoS priority queuing and policing. We introduced the following commands: **priority-queue**, **queue-limit**, **tx-ring-limit**, **priority, police, show priority-queue statistics, show service-policy police, show service-policy priority, show running-config priority-queue, clear configure priority-queue** .
Shaping and hierarchical priority queuing	7.2(4)/8.0(4)	We introduced QoS shaping and hierarchical priority queuing. We introduced the following commands: **shape**, **show service-policy shape**.
Ten Gigabit Ethernet support for a standard priority queue on the ASA 5585-X	8.2(3)/8.4(1)	We added support for a standard priority queue on Ten Gigabit Ethernet interfaces for the ASA 5585-X.

CHAPTER 24

Troubleshooting Connections and Resources

This chapter describes how to troubleshoot the ASA and includes the following sections:

Testing Your Configuration

This section describes how to test connectivity for the single mode ASA or for each security context, how to ping the ASA interfaces, and how to allow hosts on one interface to ping through to hosts on another interface.

We recommend that you only enable pinging and debugging messages during troubleshooting. When you are done testing the ASA, follow the steps in the "Disabling the Test Configuration" section on page 24-6.

This section includes the following topics:

Enabling ICMP Debugging Messages and Syslog Messages

Debugging messages and syslog messages can help you troubleshoot why your pings are not successful. The ASA only shows ICMP debugging messages for pings to the ASA interfaces, and not for pings through the ASA to other hosts.

To enable debugging and syslog messages, perform the following steps:

	Command	Purpose

Step 1	**`debug icmp trace`**
	Example:
	`hostname(config)# debug icmp trace`
		Shows ICMP packet information for pings to the ASA interfaces.

Step 2	**`logging monitor debug`**
	Example:
	`hostname(config)# logging monitor debug`
		Sets syslog messages to be sent to Telnet or SSH sessions.
		Note You can alternately use the **logging buffer debug** command to send log messages to a buffer, and then view them later using the **show logging** command.

Step 3	**`terminal monitor`**
	Example:
	`hostname(config)# terminal monitor`
		Sends the syslog messages to a Telnet or SSH session.

Step 4	**`logging on`**
	Example:
	`hostname(config)# logging on`
		Enables syslog message generation.

Examples

The following example shows a successful ping from an external host (209.165.201.2) to the ASA outside interface (209.165.201.1):

```
ciscoasa(config)# debug icmp trace
Inbound ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 512) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 768) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 768) 209.165.201.1 > 209.165.201.2
Outbound ICMP echo request (len 32 id 1 seq 1024) 209.165.201.2 > 209.165.201.1
Inbound ICMP echo reply (len 32 id 1 seq 1024) 209.165.201.1 > 209.165.201.2
```

The output shows the ICMP packet length (32 bytes), the ICMP packet identifier (1), and the ICMP sequence number (the ICMP sequence number starts at 0, and is incremented each time that a request is sent).

Pinging ASA Interfaces

To test whether the ASA interfaces are up and running and that the ASA and connected routers are operating correctly, you can ping the ASA interfaces.

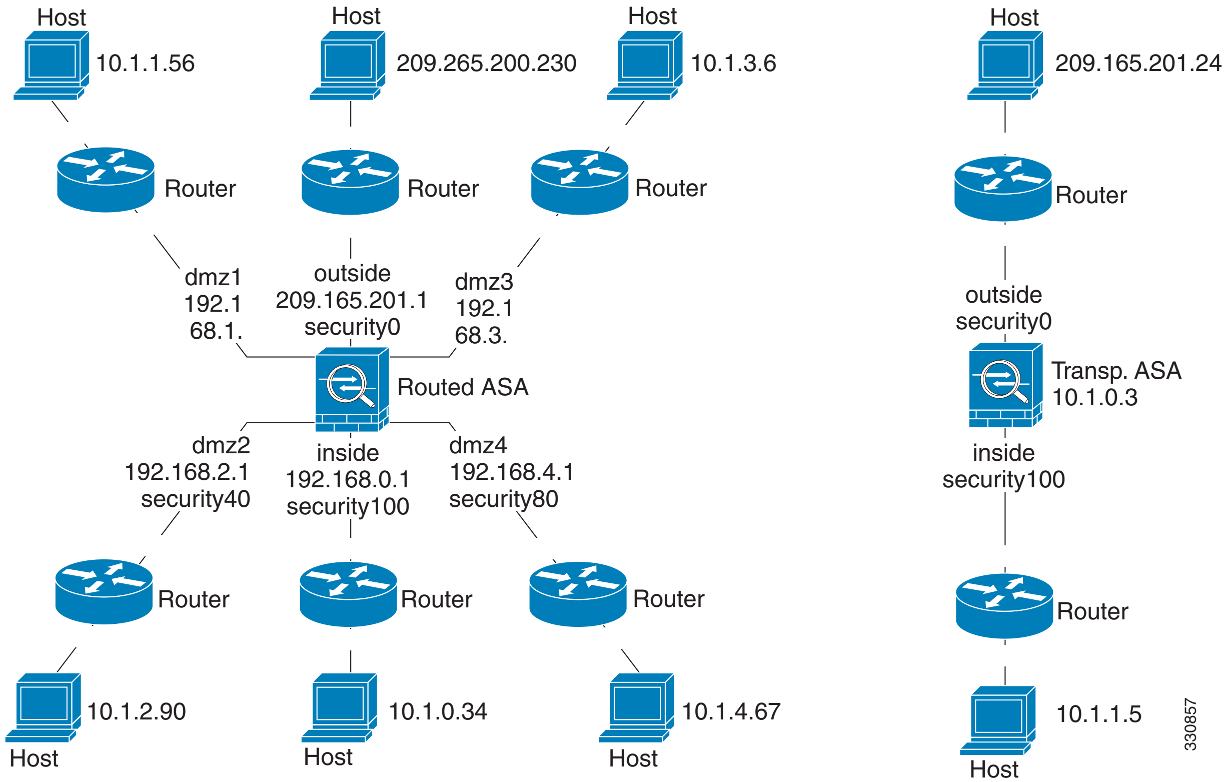
To ping the ASA interfaces, perform the following steps:

Step 1 Draw a diagram of your single-mode ASA or security context that shows the interface names, security levels, and IP addresses.

Note Although this procedure uses IP addresses, the **ping** command also supports DNS names and names that are assigned to a local IP address with the **name** command.

The diagram should also include any directly connected routers and a host on the other side of the router from which you will ping the ASA. You will use this information in this procedure and in the procedure in the "Passing Traffic Through the ASA" section on page 24-5. (See Figure 24-1.)

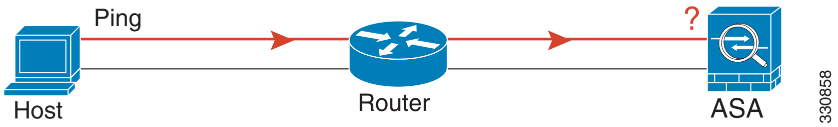
Figure 24-1 Network Diagram with Interfaces, Routers, and Hosts



Step 2 Ping each ASA interface from the directly connected routers. For transparent mode, ping the management IP address. This test ensures that the ASA interfaces are active and that the interface configuration is correct.

A ping might fail if the ASA interface is not active, the interface configuration is incorrect, or if a switch between the ASA and a router is down (see Figure 24-2). In this case, no debugging messages or syslog messages appear, because the packet never reaches the ASA.

Figure 24-2 Ping Failure at the ASA Interface

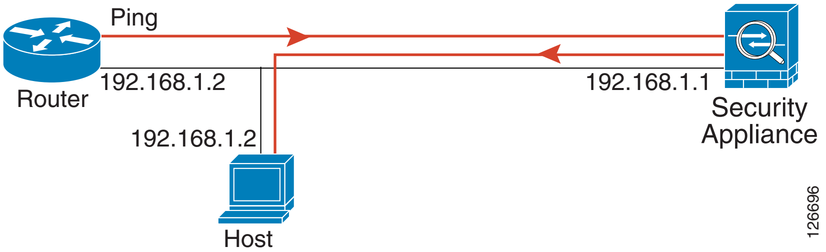


If the ping reaches the ASA, and it responds, debugging messages similar to the following appear:

```
ICMP echo reply (len 32 id 1 seq 256) 209.165.201.1 > 209.165.201.2
ICMP echo request (len 32 id 1 seq 512) 209.165.201.2 > 209.165.201.1
```

If the ping reply does not return to the router, then a switch loop or redundant IP addresses may exist (see Figure 24-3).

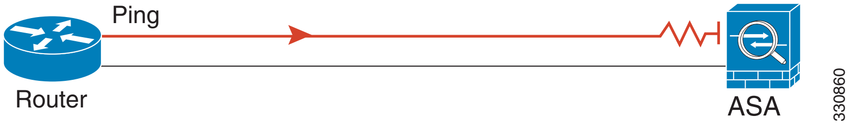
Figure 24-3 Ping Failure Because of IP Addressing Problems



Step 3 Ping each ASA interface from a remote host. For transparent mode, ping the management IP address. This test checks whether the directly connected router can route the packet between the host and the ASA, and whether the ASA can correctly route the packet back to the host.

A ping might fail if the ASA does not have a return route to the host through the intermediate router (see Figure 24-4). In this case, the debugging messages show that the ping was successful, but syslog message 110001 appears, indicating a routing failure has occurred.

Figure 24-4 Ping Failure Because the ASA Has No Return Route



Passing Traffic Through the ASA

After you successfully ping the ASA interfaces, make sure that traffic can pass successfully through the ASA. By default, you can ping from a high security interface to a low security interface. You just need to enable ICMP inspection to allow returning traffic through. If you want to ping from high to low, then you need to apply an ACL to allow traffic. If you use NAT, this test shows that NAT is operating correctly.

Ping from the host or router through the source interface to another host or router on another interface. Repeat this step for as many interface pairs as you want to check.

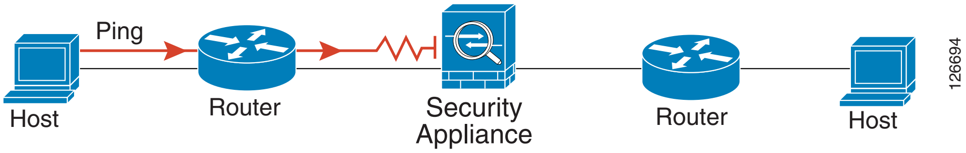
If the ping succeeds, a syslog message appears to confirm the address translation for routed mode (305009 or 305011) and that an ICMP connection was established (302020). You can also enter either the **show xlate** or **show conns** command to view this information.

The ping might fail because NAT is not configured correctly. In this case, a syslog message appears, showing that the NAT failed (305005 or 305006). If the ping is from an outside host to an inside host, and you do not have a static translation, the following syslog message appears:

```
%ASA-3-106010: deny inbound icmp.
```

Note The ASA only shows ICMP debugging messages for pings to the ASA interfaces, and not for pings through the ASA to other hosts.

Figure 24-5 Ping Failure Because the ASA is Not Translating Addresses



Detailed Steps

	Command	Purpose
Step 1	**policy-map global_policy**	Edits the default global policy and enters policy-map configuration mode.
Step 2	**class inspection_default**	Edits the default class map, which matches application traffic for standard protocols and ports. For ICMP, this class matches all ICMP traffic.
Step 3	**inspect icmp**	Enables the ICMP inspection engine and ensures that ICMP responses can return to the source host.

Step 4	(Optional, for low security interfaces) `access-list ICMPACL extended permit icmp any any`	Adds an ACL to allow ICMP traffic from any source host.
Step 5	`access-group ICMPACL in interface outside`	Assigns the ACL to the outside interface. Replace “outside” with your interface name if it is different. Repeat the command for each interface that you want to allow ICMP traffic from high to low. **Note** After you apply this ACL to an interface that is not the lowest security interface, only ICMP traffic is allowed; the implicit permit from high to low is removed. For example, to allow a DMZ interface (level 50) to ping the inside interface (level 100), you need to apply this ACL. However, now traffic from DMZ to outside (level 0) is limited to ICMP traffic only, as opposed to all traffic that the implicit permit allowed before. After testing ping, be sure to remove this ACL from your interfaces, especially interfaces to which you want to restore the implicit permit (**no access-list ICMPACL**).

Disabling the Test Configuration

After you complete your testing, disable the test configuration that allows ICMP to and through the ASA and that prints debugging messages. If you leave this configuration in place, it can pose a serious security risk. Debugging messages also slow ASA performance.

To disable the test configuration, perform the following steps:

	Command	Purpose
Step 1	`no debug icmp trace`	Disables ICMP debugging messages.
Step 2	`no logging on`	Disables logging.
Step 3	`no access-list ICMPACL`	Removes the ICMPACL ACL, and deletes the related **access-group** commands.

Step 4	`policy-map global_policy`
	`  class inspection_default`
	`    no inspect icmp`
		(Optional) Disables the ICMP inspection engine.

Determining Packet Routing with Traceroute

You can trace the route of a packet using the traceroute feature, which is accessed with the **traceroute** command. A traceroute works by sending UDP packets to a destination on an invalid port. Because the port is not valid, the routers along the way to the destination respond with an ICMP Time Exceeded Message, and report that error to the ASA.

Tracing Packets with Packet Tracer

The packet tracer tool provides packet tracing for packet sniffing and network fault isolation, as well as detailed information about the packets and how they are processed by the ASA. If a configuration command did not cause the packet to drop, the packet tracer tool can provide information about the cause in an easily readable format.

In addition, you can trace the lifespan of a packet through the ASA to see whether the packet is operating correctly with the packet tracer tool. This tool enables you to do the following:

- Debug all packet drops in a production network.
- Verify the configuration is working as intended.
- Show all rules applicable to a packet, along with the CLI commands that caused the rule addition.
- Show a time line of packet changes in a data path.
- Inject tracer packets into the data path.
- Search for an IPv4 or IPv6 address based on the user identity and the FQDN.

To trace packets, enter the following command:

Command	Purpose
packet-tracer input [*ifc_name*] [**icmp** [*sip* | **user** *username* | **fqdn** *fqdn-string*] *type code ident* [*dip* | **fqdn** *fqdn-string*]] | [**tcp** [*sip* | **user** *username* | **fqdn** *fqdn-string*] *sport* [*dip* | **fqdn** *fqdn-string*] *dport*] | [**udp** [*sip* | **user** *username* | **fqdn** *fqdn-string*] *sport* [*dip* | **fqdn** *fqdn-string*] *dport*] | [**rawip** [*sip* | **user** *username* | **fqdn** *fqdn-string*] [*dip* | **fqdn** *fqdn-string*]] [**detailed**] [**xml**] **Example:** `ciscoasa# packet-tracer input inside tcp 10.2.25.3 www 209.165.202.158 aol detailed`	Provides detailed information about the packets and how they are processed by the ASA. The example shows how to enable packet tracing from inside host 10.2.25.3 to external host 209.165.202.158, including detailed information.

Monitoring Per-Process CPU Usage

You can monitor the processes that run on the CPU. You can obtain information about the percentage of CPU that is used by a certain process. CPU usage statistics are sorted in descending order to display the highest consumer at the top. Also included is information about the load on the CPU per process, at 5 seconds, 1 minute, and 5 minutes before the log time. This information is updated automatically every 5 seconds to provide real-time statistics.

You can use the **show process cpu-usage sorted** command to find a breakdown of the process-related load-to-CPU that is consumed by any configured contexts.

PART 7

Configuring Advanced Network Protection

CHAPTER 25

Configuring the ASA for Cisco Cloud Web Security

Cisco Cloud Web Security provides web security and web filtering services through the Software-as-a-Service (SaaS) model. Enterprises with the ASA in their network can use Cloud Web Security services without having to install additional hardware.

When Cloud Web Security is enabled on the ASA, the ASA transparently redirects selected HTTP and HTTPS traffic to the Cloud Web Security proxy servers. The Cloud Web Security proxy servers then scan the content and allow, block, or send a warning about the traffic based on the policy configured in Cisco ScanCenter to enforce acceptable use and to protect users from malware.

The ASA can optionally authenticate and identify users with Identity Firewall (IDFW) and AAA rules. The ASA encrypts and includes the user credentials (including usernames and/or user groups) in the traffic it redirects to Cloud Web Security. The Cloud Web Security service then uses the user credentials to match the traffic to the policy. It also uses these credentials for user-based reporting. Without user authentication, the ASA can supply an (optional) default username and/or group, although usernames and groups are not required for the Cloud Web Security service to apply policy.

You can customize the traffic you want to send to Cloud Web Security when you create your service policy rules. You can also configure a "whitelist" so that a subset of web traffic that matches the service policy rule instead goes directly to the originally requested web server and is not scanned by Cloud Web Security.

You can configure a primary and a backup Cloud Web Security proxy server, each of which the ASA polls regularly to check for availability.

Note This feature is also called "ScanSafe," so the ScanSafe name appears in some commands.

This chapter includes the following sections:

Information About Cisco Cloud Web Security

This section includes the following topics:

Redirection of Web Traffic to Cloud Web Security

When an end user sends an HTTP or HTTPS request, the ASA receives it and optionally retrieves the user and/or group information. If the traffic matches an ASA service policy rule for Cloud Web Security, then the ASA redirects the request to the Cloud Web Security proxy servers. The ASA acts as an intermediary between the end user and the Cloud Web Security proxy server by redirecting the connection to the proxy server. The ASA changes the destination IP address and port in the client requests and adds Cloud Web Security-specific HTTP headers and then sends the modified request to the Cloud Web Security proxy server. The Cloud Web Security HTTP headers include various kinds of information, including the username and user group (if available).

User Authentication and Cloud Web Security

User identity can be used to apply policy in Cloud Web Security. User identity is also useful for Cloud Web Security reporting. User identity is not required to use Cloud Web Security. There are other methods to identify traffic for Cloud Web Security policy.

The ASA supports the following methods of determining the identity of a user, or of providing a default identity:

- AAA rules—When the ASA performs user authentication using a AAA rule, the username is retrieved from the AAA server or local database. Identity from AAA rules does not include group information. If configured, the default group is used. For information about configuring AAA rules, see Chapter 7, "Configuring AAA Rules for Network Access."
- IDFW—When the ASA uses IDFW with the Active Directory (AD), the username and group is retrieved from the AD agent when you activate a user and/or group by using an ACL in a feature such as an access rule or in your service policy, or by configuring the user identity monitor to download user identity information directly.

  For information about configuring IDFW, see Chapter 38, "Configuring the Identity Firewall," in the general operations configuration guide.
- Default username and group—Without user authentication, the ASA uses an optional default username and/or group for all users that match a service policy rule for Cloud Web Security.

Authentication Keys

Each ASA must use an authentication key that you obtain from Cloud Web Security. The authentication key lets Cloud Web Security identify the company associated with web requests and ensures that the ASA is associated with valid customer.

You can use one of two types of authentication keys for your ASA: the company key or the group key.

- Company Authentication Key, page 25-3
- Group Authentication Key, page 25-3

Company Authentication Key

A Company authentication key can be used on multiple ASAs within the same company. This key simply enables the Cloud Web Security service for your ASAs. The administrator generates this key in ScanCenter (https://scancenter.scansafe.com/portal/admin/login.jsp); you have the opportunity to e-mail the key for later use. You cannot look up this key later in ScanCenter; only the last 4 digits are shown in ScanCenter. For more information, see the Cloud Web Security documentation: http://www.cisco.com/en/US/products/ps11720/products_installation_and_configuration_guides_list.html.

Group Authentication Key

A Group authentication key is a special key unique to each ASA that performs two functions:

- Enables the Cloud Web Security service for one ASA.
- Identifies all traffic from the ASA so you can create ScanCenter policy per ASA.

For information about using the Group authentication key for policy, see the "ScanCenter Policy" section on page 25-4).

The administrator generates this key in ScanCenter (https://scancenter.scansafe.com/portal/admin/login.jsp); you have the opportunity to e-mail the key for later use. You cannot look up this key later in ScanCenter; only the last 4 digits are shown in ScanCenter.

For more information, see the Cloud Web Security documentation: http://www.cisco.com/en/US/products/ps11720/products_installation_and_configuration_guides_list.html.

ScanCenter Policy

In ScanCenter, traffic is matched against policy rules in order until a rule is matched. Cloud Web Security then applies the configured action for the rule. User traffic can match a policy rule in ScanCenter based on group association: a *directory group* or a *custom group*.

- Directory Groups, page 25-4
- Custom Groups, page 25-4
- How Groups and the Authentication Key Interoperate, page 25-5

Directory Groups

Directory groups define the group to which traffic belongs. The group, if present, is included in the HTTP header of the client request. The ASA includes the group in the HTTP header when you configure IDFW. If you do not use IDFW, you can configure a default group for traffic matching an ASA rule for Cloud Web Security inspection.

When you configure a directory group, you must enter the group name exactly.

- IDFW group names are sent in the following format:

  domain-name\group-name

  When the ASA learns the IDFW group name, the format on the ASA is *domain-name\\group-name*. However, the ASA modifies the name to use only one backslash (\) to conform to typical ScanCenter notation.

- The default group name is sent in the following format:

  [*domain*\]*group-name*

  On the ASA, you need to configure the optional domain name to be followed by 2 backslashes (\\); however, the ASA modifies the name to use only one backslash (\) to conform to typical ScanCenter notation. For example, if you specify "Cisco\\Boulder1," the ASA modifies the group name to be "Cisco\Boulder1" with only one backslash (\) when sending the group name to Cloud Web Security.

Custom Groups

Custom groups are defined using one or more of the following criteria:

- ScanCenter Group authentication key—You can generate a Group authentication key for a custom group. Then, if you identify this group key when you configure the ASA, all traffic from the ASA is tagged with the Group key.
- Source IP address—You can identify source IP addresses in the custom group. Note that the ASA service policy is based on source IP address, so you might want to configure any IP address-based policy on the ASA instead.
- Username—You can identify usernames in the custom group.
  - IDFW usernames are sent in the following format:

    domain-name\username

- AAA usernames, when using RADIUS or TACACS+, are sent in the following format:

  LOCAL\username

- AAA usernames, when using LDAP, are sent in the following format:

  domain-name\username

- For the default username, it is sent in the following format:

  [*domain-name*\\]*username*

  For example, if you configure the default username to be “Guest,” then the ASA sends “Guest.” If you configure the default username to be “Cisco\Guest,” then the ASA sends “Cisco\Guest.”

How Groups and the Authentication Key Interoperate

Unless you need the per-ASA policy that a custom group+group key provides, you will likely use a company key. Note that not all custom groups are associated with a group key. Non-keyed custom groups can be used to identify IP addresses or usernames, and can be used in your policy along with rules that use directory groups.

Even if you do want per-ASA policy and are using a group key, you can also use the matching capability provided by directory groups and non-keyed custom groups. In this case, you might want an ASA-based policy, with some exceptions based on group membership, IP address, or username. For example, if you want to exempt users in the America\Management group across all ASAs:

1. Add a directory group for America\Management.
2. Add an exempt rule for this group.
3. Add rules for each custom group+group key after the exempt rule to apply policy per-ASA.
4. Traffic from users in America\Management will match the exempt rule, while all other traffic will match the rule for the ASA from which it originated.

Many combinations of keys, groups, and policy rules are possible.

Cloud Web Security Actions

After applying the configured policies, Cloud Web Security either blocks, allows, or sends a warning about the user request:

- Allows—When Cloud Web Security allows the client request, it contacts the originally requested server and retrieves the data. It forwards the server response to the ASA, which then forwards it to the user.
- Blocks—When Cloud Web Security blocks the client request, it notifies the user that access has been blocked. It sends an HTTP 302 “Moved Temporarily” response that redirects the client application to a web page hosted by the Cloud Web Security proxy server showing the blocked error message. The ASA forwards the 302 response to the client.
- Warns—When the Cloud Web Security proxy server determines that a site may be in breach of the acceptable use policy, it displays a warning page about the site. You can choose to heed the warning and drop the request to connect, or you can click through the warning and proceed to the requested site.

You can also choose how the ASA handles web traffic when it cannot reach either the primary or backup Cloud Web Security proxy server. It can block or allow all web traffic. By default, it blocks web traffic.

Bypassing Scanning with Whitelists

If you use AAA rules or IDFW, you can configure the ASA so that web traffic from specific users or groups that otherwise match the service policy rule is not redirected to the Cloud Web Security proxy server for scanning. When you bypass Cloud Web Security scanning, the ASA retrieves the content directly from the originally requested web server without contacting the proxy server. When it receives the response from the web server, it sends the data to the client. This process is called “whitelisting” traffic.

Although you can achieve the same results of exempting traffic based on user or group when you configure the class of traffic using ACLs to send to Cloud Web Security, you might find it more straightforward to use a whitelist instead. Note that the whitelist feature is only based on user and group, not on IP address.

IPv4 and IPv6 Support

Cloud Web Security currently supports only IPv4 addresses. If you use IPv6 internally, NAT 64 must be performed for any IPv6 flows that need to be sent to Cloud Web Security.

The following table shows the class map traffic that is supported by Cloud Web Security redirection:

Class Map Traffic	Cloud Web Security Inspection
From IPv4 to IPv4	Supported
From IPv6 to IPv4 (using NAT64)	Supported
From IPv4 to IPv6	Not Supported
From IPv6 to IPv6	Not Supported

Failover from Primary to Backup Proxy Server

When you subscribe to the Cisco Cloud Web Security service, you are assigned a primary Cloud Web Security proxy server and backup proxy server.

If any client is unable to reach the primary server, then the ASA starts polling the tower to determine availability. (If there is no client activity, the ASA polls every 15 miniutes.) If the proxy server is unavailable after a configured number of retries (the default is 5; this setting is configurable), the server is declared unreachable, and the backup proxy server becomes active.

If a client or the ASA can reach the server at least twice consecutively before the retry count is reached, the polling stops and the tower is determined to be reachable.

After a failover to the backup server, the ASA continues to poll the primary server. If the primary server becomes reachable, then the ASA returns to using the primary server.

Licensing Requirements for Cisco Cloud Web Security

Model	License Requirement
All models	Strong Encryption (3DES/AES) License to encrypt traffic between the security appliance and the Cloud Web Security server.

On the Cloud Web Security side, you must purchase a Cisco Cloud Web Security license and identify the number of users that the ASA handles. Then log into ScanCenter, and generate your authentication keys.

Prerequisites for Cloud Web Security

(Optional) User Authentication Prerequisites

To send user identity information to Cloud Web Security, configure one of the following on the ASA:

- AAA rules (username only)—See Chapter 7, "Configuring AAA Rules for Network Access."
- IDFW (username and group)—See Chapter 38, "Configuring the Identity Firewall," in the general operations configuration guide.

(Optional) Fully Qualified Domain Name Prerequisites

If you use FQDNs in ACLs for your service policy rule, or for the Cloud Web Security server, you must configure a DNS server for the ASA according to the "Configuring the DNS Server" section on page 16-8 in the general operations configuration guide.

Guidelines and Limitations

Context Mode Guidelines

Supported in single and multiple context modes.

In multiple context mode, the server configuration is allowed only in the system, and the service policy rule configuration is allowed only in the security contexts.

Each context can have its own authentication key, if desired.

Firewall Mode Guidelines

Supported in routed firewall mode only. Does not support transparent firewall mode.

IPv6 Guidelines

Does not support IPv6. See the "IPv4 and IPv6 Support" section on page 25-6.

Additional Guidelines

- Cloud Web Security is not supported with ASA clustering.
- Clientless SSL VPN is not supported with Cloud Web Security; be sure to exempt any clientless SSL VPN traffic from the ASA service policy for Cloud Web Security.

- When an interface to the Cloud Web Security proxy servers goes down, output from the **show scansafe server** command shows both servers up for approximately 15-25 minutes. This condition may occur because the polling mechanism is based on the active connection, and because that interface is down, it shows zero connection, and it takes the longest poll time approach.
- Cloud Web Security is not supported with the ASA CX module. If you configure both the ASA CX action and Cloud Web Security inspection for the same traffic, the ASA only performs the ASA CX action.
- Cloud Web Security inspection is compatibile with HTTP inspection for the same traffic. HTTP inspection is enabled by default as part of the default global policy.
- Cloud Web Security is not supported with extended PAT or any application that can potentially use the same source port and IP address for separate connections. For example, if two different connections (targeted to separate servers) use extended PAT, the ASA might reuse the same source IP and source port for both connection translations because they are differentiated by the separate destinations. When the ASA redirects these connections to the Cloud Web Security server, it replaces the destination with the Cloud Web Security server IP address and port (8080 by default). As a result, both connections now appear to belong to the same flow (same source IP/port and destination IP/port), and return traffic cannot be untranslated properly.
- The **match default-inspection-traffic** command does not include the default ports for the Cloud Web Security inspection (80 and 443).

Default Settings

By default, Cisco Cloud Web Security is not enabled.

Configuring Cisco Cloud Web Security

- Configuring Communication with the Cloud Web Security Proxy Server, page 25-8
- (Multiple Context Mode) Allowing Cloud Web Security Per Security Context, page 25-9
- Configuring a Service Policy to Send Traffic to Cloud Web Security, page 25-10
- (Optional) Configuring Whitelisted Traffic, page 25-15
- Configuring the Cloud Web Security Policy, page 25-16

Configuring Communication with the Cloud Web Security Proxy Server

Guidelines

The public key is embedded in the ASA software, so there is no need for you to configure it.

Detailed Steps

	Command	Purpose
Step 1	**scansafe general-options** **Example:** `ciscoasa(config)# scansafe general-options`	Enters scansafe general-options configuration mode.
Step 2	**server primary** {**ip** *ip_address* | **fqdn** *fqdn*} [**port** *port*] **Example:** `ciscoasa(cfg-scansafe)# server primary ip 192.168.43.10`	Configures the fully qualified domain name or IP address of the primary Cloud Web Security proxy server. By default, the Cloud Web Security proxy server uses port 8080 for both HTTP and HTTPS traffic; do not change this value unless directed to do so.
Step 3	**server backup** {**ip** *ip_address* | **fqdn** *fqdn*} [**port** *port*] **Example:** `ciscoasa(cfg-scansafe)# server backup fqdn server.example.com`	(Optional) Configures the fully qualified domain name or IP address of the backup Cloud Web Security proxy server. By default, the Cloud Web Security proxy server uses port 8080 for both HTTP and HTTPS traffic; do not change this value unless directed to do so.
Step 4	**retry-count** *value* **Example:** `ciscoasa(cfg-scansafe)# retry-count 2`	(Optional) Enters the value for the number of consecutive polling failures to the Cloud Web Security proxy server before determining the server is unreachable. Polls are performed every 30 seconds. Valid values are from 2 to 100, and the default is 5. See the "Failover from Primary to Backup Proxy Server" section on page 25-6.

Step 5	**license** *hex_key*
	Example:
	`ciscoasa(cfg-scansafe)#`
	`license F12A588FE5A0A4AE86C10D222FC658F3`
		Configures the authentication key that the ASA sends to the Cloud Web Security proxy servers to indicate from which organization the request comes. The authentication key is a 16-byte hexidecimal number.
		See the "Authentication Keys" section on page 25-3.

Examples

The following example configures a primary and backup server:

```
scansafe general-options
 server primary ip 10.24.0.62 port 8080
 server backup ip 10.10.0.7 port 8080
 retry-count 7
 license 366C1D3F5CE67D33D3E9ACEC265261E5
```

(Multiple Context Mode) Allowing Cloud Web Security Per Security Context

In multiple context mode, you must allow Cloud Web Security per context. See the "Configuring a Security Context" section on page 8-20 in the general operations configuration guide.

Note You must configure a route pointing to the Scansafe towers in both; the admin context and the specific context. This ensures that the Scansafe tower does not become unreachable in the Active/Active failover scenario.

The following sample configuration enables Cloud Web Security in context one with the default license and in context two with the license key override:

```
! System Context
!
scansafe general-options
server primary ip 180.24.0.62 port 8080
retry-count 5
license 366C1D3F5CE67D33D3E9ACEC265261E5
!
context one
 allocate-interface GigabitEthernet0/0.1
 allocate-interface GigabitEthernet0/1.1
 allocate-interface GigabitEthernet0/3.1
 scansafe
 config-url disk0:/one_ctx.cfg
!
context two
 allocate-interface GigabitEthernet0/0.2
 allocate-interface GigabitEthernet0/1.2
 allocate-interface GigabitEthernet0/3.2
 scansafe license 366C1D3F5CE67D33D3E9ACEC26789534
 config-url disk0:/two_ctx.cfg
!
```

Configuring a Service Policy to Send Traffic to Cloud Web Security

See Chapter 1, "Configuring a Service Policy Using the Modular Policy Framework," for more information about service policy rules.

Prerequisites

(Optional) If you need to use a whitelist to exempt some traffic from being sent to Cloud Web Security, first create the whitelist according to the "(Optional) Configuring Whitelisted Traffic" section on page 25-15 so you can refer to the whitelist in your service policy rule.

Detailed Steps

	Command	Purpose

Step 1	**policy-map type inspect scansafe** *name1*
	Example:
	`ciscoasa(config)# policy-map type inspect`
	`scansafe cws_inspect_pmap1`
		Creates an inspection policy map so you can configure essential parameters for the rule and also optionally identify the whitelist. An inspection policy map is required for each class of traffic that you want to send to Cloud Web Security.
		The *policy_map_name* argument can be up to 40 characters in length.
		You enter policy-map configuration mode.

Step 2	`parameters`
	Example:
	`ciscoasa(config-pmap)# parameters`
		Parameters lets you configure the protocol and the default user or group. You enter parameters configuration mode.

Step 3	**{http | https}**
	Example:
	`ciscoasa(config-pmap-p)# http`
		You can only specify one service type for this inspection policy map, either **http** or **https**.

Step 4	(Optional)
	default {[**user** *username*]
	[**group** *groupname*]}
	Example:
	`ciscoasa(config-pmap-p)# default group`
	`default_group`
		Specifies that if the ASA cannot determine the identity of the user coming into the ASA, then the default user and/or group is included in the HTTP header.

Step 5	(Optional, for a Whitelist)
	class *whitelist_name*
	Example:
	`ciscoasa(config-pmap-p)# class whitelist1`
		Identifies the whitelist class map name that you created in the "(Optional) Configuring Whitelisted Traffic" section on page 25-15.

Step 6	`whitelist`
	Example:
	`ciscoasa(config-pmap-p)# class whitelist1`
	`ciscoasa(config-pmap-c)# whitelist`
		Performs the whitelist action on the class of traffic.

	Command	Purpose

Step 7	**policy-map type inspect scansafe** *name2*
	parameters
	default {[**user** *user*] [**group** *group*]}
	class *whitelist_name2*
	whitelist
	
	Example:
	`ciscoasa(config)# policy-map type inspect scansafe cws_inspect_pmap2`
	`ciscoasa(config-pmap)# parameters`
	`ciscoasa(config-pmap-p)# default group2 default_group2`
	`ciscoasa(config-pmap-p)# class whitelist2`
	`ciscoasa(config-pmap-c)# whitelist`
		Repeat Step 1 to Step 6 to create a separate class map for HTTPS traffic (for example). You can create an inspection class map for each class of traffic you want to send to Cloud Web Security. You can reuse an inspection class map for multiple classes of traffic if desired.

Step 8	**access-list** *access_list_name* [**line** *line_number*] **extended** {**deny** | **permit**} **tcp** [*user_argument*] [*security_group_argument*] *source_address_argument* [*port_argument*] *dest_address_argument* [*port_argument*]
	
	Example:
	`ciscoasa(config)# object network cisco1`
	`ciscoasa(config-object-network)# fqdn www.cisco.com`
	
	`ciscoasa(config)# object network cisco2`
	`ciscoasa(config-object-network)# fqdn tools.cisco.com`
	
	`ciscoasa(config)# access-list SCANSAFE_HTTP extended deny tcp any4 object cisco1 eq 80`
	`ciscoasa(config)# access-list SCANSAFE_HTTP extended deny tcp any4 object cisco2 eq 80`
	`ciscoasa(config)# access-list SCANSAFE_HTTP extended permit tcp any4 any4 eq 80`
		Identifies the class of traffic you want to send to Cloud Web Security. Create an ACL consisting of one or more access control entries (ACEs). For detailed information about ACLs, see Chapter 19, “Adding an Extended Access Control List,” in the general operations configuration guide.
		
		Cloud Web Security only operates on HTTP and HTTPS traffic. Each type of traffic is treated separately by the ASA. Therefore, you need to create HTTP-only ACLs and HTTPS-only ACLs. Create as many ACLs as needed for your policy.
		
		A **permit** ACE sends matching traffic to Cloud Web Security. A **deny** ACE exempts traffic from the service policy rule, so it is not sent to Cloud Web Security.
		
		When creating your ACLs, consider how you can match appropriate traffic that is destined for the Internet, but not match traffic that is destined for other internal networks. For example, to prevent inside traffic from being sent to Cloud Web Security when the destination is an internal server on the DMZ, be sure to add a deny ACE to the ACL that exempts traffic to the DMZ.
		
		FQDN network objects might be useful in exempting traffic to specific servers.
		
		The *user_argument* lets you specify the IDFW username or group, either inline or by referring to an object group.
		
		The *security_group_argument* lets you specify the TrustSec security group, either inline or by referring to an object group. Note that although you can match traffic to send to Cloud Web Security by security group, the ASA does not send security group information to Cloud Web Security in the HTTP header; Cloud Web Security cannot create policy based on the security group.

Step 9	**class-map** *name1*
	
	Example:
	`ciscoasa(config)# class-map cws_class1`
		Creates a class map to identify the traffic for which you want to enable Cloud Web Security filtering.

	Command	Purpose
Step 10	**match access-list** *acl1* **Example:** `ciscoasa(config-cmap)# match access-list SCANSAFE_HTTP`	Specifies an ACL created in Step 8. Although you can use other match statements for this rule, we recommend using the **match access-list** command because it is the most versatile for identifying HTTP or HTTPS-only traffic. See the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12 for more information.

Step 11	**class-map** *name2*
	match access-list *acl2*
	Example:
	`ciscoasa(config)# class-map cws_class2`
	`ciscoasa(config-cmap)# match access-list SCANSAFE_HTTPS`
		(Optional) Creates an additional class map, for example for HTTPS traffic. You can create as many classes as needed for this service policy rule.

Step 12	**policy-map** *name* **Example:** `ciscoasa(config)# policy-map cws_policy`	Adds or edits a policy map that sets the actions to take with the class map traffic. The policy map in the default global policy is called global_policy. You can edit this policy, or create a new one. You can only apply one policy to each interface or globally.

Step 13	**class** *name1*
	Example:
	`ciscoasa(config-pmap)# class cws_class1`
		Identifies the class map created in Step 9.

Step 14	**inspect scansafe** *scansafe_policy_name1* [**fail-open** | **fail-close**] **Example:** `ciscoasa(config-pmap-c)# inspect scansafe cws_inspect_pmap1 fail-open`	Enables Cloud Web Security inspection on the traffic in this class. Specify the inspection class map name that you created in Step 1. Specify **fail-open** to allow traffic to pass through the ASA if the Cloud Web Security servers are unavailable. Specify **fail-close** to drop all traffic if the Cloud Web Security servers are unavailable. **fail-close** is the default.

Step 15	**class** *name2*
	inspect scansafe *scansafe_policy_name2* [**fail-open** | **fail-close**]
	Example:
	`ciscoasa(config-pmap)# class cws_class2`
	`ciscoasa(config-pmap-c)# inspect scansafe cws_inspect_pmap2 fail-open`
		(Optional) Identifies a second class map that you created in Step 11, and enables Cloud Web Security inspection for it.
		You can configure multiple class maps as needed.

Step 16	**service-policy** *policymap_name* {**global** | **interface** *interface_name*}
	Example:
	`ciscoasa(config)# service-policy cws_policy inside`
		Activates the policy map on one or more interfaces. **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface. See the "Applying Actions to an Interface (Service Policy)" section on page 1-17 for more information.

Examples

The following example configures two classes: one for HTTP and one for HTTPS. Each ACL exempts traffic to www.cisco.com and to tools.cisco.com, and to the DMZ network, for both HTTP and HTTPS. All other traffic is sent to Cloud Web Security, except for traffic from several whitelisted users and groups. The policy is then applied to the inside interface.

```
hostname(config)# class-map type inspect scansafe match-any whitelist1
hostname(config-cmap)# match user user1 group cisco
hostname(config-cmap)# match user user2
hostname(config-cmap)# match group group1
hostname(config-cmap)# match user user3 group group3

hostname(config)# policy-map type inspect scansafe cws_inspect_pmap1
hostname(config-pmap)# parameters
hostname(config-pmap-p)# http
hostname(config-pmap-p)# default group default_group
hostname(config-pmap-p)# class whitelist1
hostname(config-pmap-c)# whitelist

hostname(config)# policy-map type inspect scansafe cws_inspect_pmap2
hostname(config-pmap)# parameters
hostname(config-pmap-p)# https
hostname(config-pmap-p)# default group2 default_group2
hostname(config-pmap-p)# class whitelist1
hostname(config-pmap-c)# whitelist

hostname(config)# object network cisco1
hostname(config-object-network)# fqdn www.cisco.com
hostname(config)# object network cisco2
hostname(config-object-network)# fqdn tools.cisco.com
hostname(config)# object network dmz_network
hostname(config-object-network)# subnet 10.1.1.0 255.255.255.0

hostname(config)# access-list SCANSAFE_HTTP extended deny tcp any4 object cisco1 eq 80
hostname(config)# access-list SCANSAFE_HTTP extended deny tcp any4 object cisco2 eq 80
hostname(config)# access-list SCANSAFE_HTTP extended deny tcp any4 object dmz_network eq
80
hostname(config)# access-list SCANSAFE_HTTP extended permit tcp any4 any4 eq 80

hostname(config)# access-list SCANSAFE_HTTPS extended deny tcp any4 object cisco1 eq 443
hostname(config)# access-list SCANSAFE_HTTPS extended deny tcp any4 object cisco2 eq 443
hostname(config)# access-list SCANSAFE_HTTP extended deny tcp any4 object dmz_network eq
443
hostname(config)# access-list SCANSAFE_HTTPS extended permit tcp any4 any4 eq 443

hostname(config)# class-map cws_class1
hostname(config-cmap)# match access-list SCANSAFE_HTTP
hostname(config)# class-map cws_class2
hostname(config-cmap)# match access-list SCANSAFE_HTTPS

hostname(config)# policy-map cws_policy
hostname(config-pmap)# class cws_class1
hostname(config-pmap-c)# inspect scansafe cws_inspect_pmap1 fail-open
hostname(config-pmap)# class cws_class2
hostname(config-pmap-c)# inspect scansafe cws_inspect_pmap2 fail-open
hostname(config)# service-policy cws_policy inside
```

(Optional) Configuring Whitelisted Traffic

If you use user authentication, you can exempt some traffic from being filtered by Cloud Web Security based on the username and/or groupname. When you configure your Cloud Web Security service policy rule, you can reference the whitelisting inspection class map. Both IDFW and AAA user credentials can be used with this feature.

Although you can achieve the same results of exempting traffic based on user or group when you configure the service policy rule, you might find it more straightforward to use a whitelist instead. Note that the whitelist feature is only based on user and group, not on IP address.

Detailed Steps

	Command	Purpose

Step 1	**`class-map type inspect scansafe`** [**`match-all`** | **`match-any`**] *`name`*
	Example:
	`ciscoasa(config)# class-map type inspect scansafe match-any whitelist1`
		Creates an inspection class map for whitelisted users and groups.
		The *class_map_name* argument is the name of the class map up to 40 characters in length.
		The **match-all** keyword is the default, and specifies that traffic must match all criteria to match the class map.
		The **match-any** keyword specifies that the traffic matches the class map if it matches at least one of the criteria.
		The CLI enters class-map configuration mode, where you can enter one or more **match** commands.

Step 2	**`match`** [**`not`**] {[**`user`** *`username`*] [**`group`** *`groupname`*]}
	Example:
	`ciscoasa(config-cmap)# match`
		The **match** keyword, followed by a specific username or groupname, specifies a user or group to whitelist.
		The **match not** keyword specifies that the user and/or group should be filtered using Web Cloud Security. For example, if you whitelist the group "cisco," but you want to scan traffic from users "johncrichton" and "aerynsun," you can specify **match not** for those users. Repeat this command to add as many users and groups as needed.

Example

The following example whitelists the same users and groups for the HTTP and HTTPS inspection policy maps:

```
hostname(config)# class-map type inspect scansafe match-any whitelist1
hostname(config-cmap)# match user user1 group cisco
hostname(config-cmap)# match user user2
hostname(config-cmap)# match group group1
hostname(config-cmap)# match user user3 group group3

hostname(config)# policy-map type inspect scansafe cws_inspect_pmap1
hostname(config-pmap)# parameters
hostname(config-pmap-p)# http
hostname(config-pmap-p)# default group default_group
hostname(config-pmap-p)# class whitelist1
hostname(config-pmap-c)# whitelist

hostname(config)# policy-map type inspect scansafe cws_inspect_pmap2
hostname(config-pmap)# parameters
hostname(config-pmap-p)# https
hostname(config-pmap-p)# default group2 default_group2
hostname(config-pmap-p)# class whitelist1
hostname(config-pmap-c)# whitelist
```

(Optional) Configuring the User Identity Monitor

When you use IDFW, the ASA only downloads user identity information from the AD server for users and groups included in active ACLs; the ACL must be used in a feature such as an access rule, AAA rule, service policy rule, or other feature to be considered active. Because Cloud Web Security can base its policy on user identity, you may need to download groups that are not part of an active ACL to get full IDFW coverage for all your users. For example, although you can configure your Cloud Web Security service policy rule to use an ACL with users and groups, thus activating any relevant groups, it is not required; you could use an ACL based entirely on IP addresses.The user identity monitor feature lets you download group information directly from the AD agent.

Restrictions

The ASA can only monitor a maximum of 512 groups, including those configured for the user identity monitor and those monitored through active ACLs.

Detailed Steps

Command	Purpose

user-identity monitor {**user-group** [*domain-name*\\]*group-name* | **object-group-user** *object-group-name*}
Example:
`ciscoasa(config)# user-identity monitor user-group CISCO\\Engineering`
	Downloads the specified user or group information from the AD agent.
	• **user-group**—Specifies a group name inline. Although you specify 2 backslashes (\\) between the domain and the group, the ASA modifies the name to include only one backslash when it sends it to Cloud Web Security, to comply with Cloud Web Security notation conventions.
	• **object-group-user**—Specifies an **object-group user** name. This group can include multiple groups.

Configuring the Cloud Web Security Policy

After you configure the ASA service policy rules, launch the ScanCenter Portal to configure Web content scanning, filtering, malware protection services, and reports.

Detailed Steps

Go to: https://scancenter.scansafe.com/portal/admin/login.jsp.

For more information, see the Cisco ScanSafe Cloud Web Security Configuration Guides:

http://www.cisco.com/en/US/products/ps11720/products_installation_and_configuration_guides_list.html

Monitoring Cloud Web Security

Command	Purpose
show scansafe server	Shows the status of the server, whether it is the current active server, the backup server, or unreachable.
show scansafe statistics	Shows total and current HTTP(S) connections.
show conn scansafe	Shows all Cloud Web Security connections, as noted by the capitol Z flag.
show service policy inspect scansafe	Shows the number of connections that are redirected or white listed by a particular policy.
See the following URL: http://Whoami.scansafe.net	From a client, access this web site to determine if your traffic is going to the Cloud Web Security server.

The **show scansafe server** command shows whether or not the Cloud Web Security proxy servers are reachable:

```
hostname# show scansafe server
ciscoasa# Primary: proxy197.scansafe.net (72.37.244.115) (REACHABLE)*
ciscoasa# Backup: proxy137.scansafe.net (80.254.152.99)
```

The **show scansafe statistics** command shows information about Cloud Web Security activity, such as the number of connections redirected to the proxy server, the number of current connections being redirected, and the number of whitelisted connections:

```
hostname# show scansafe statistics
Current HTTP sessions : 0
Current HTTPS sessions : 0
Total HTTP Sessions : 0
Total HTTPS Sessions : 0
Total Fail HTTP sessions : 0
Total Fail HTTPS sessions : 0
Total Bytes In : 0 Bytes
Total Bytes Out : 0 Bytes
HTTP session Connect Latency in ms(min/max/avg) : 0/0/0
HTTPS session Connect Latency in ms(min/max/avg) : 0/0/0
```

The **show service policy inspect scansafe** command shows the number of connections that are redirected or whitelisted by a particular policy:

```
hostname(config)# show service-policy inspect scansafe
Global policy:
  Service-policy: global_policy
    Class-map: inspection_default
Interface inside:
  Service-policy: scansafe-pmap
    Class-map: scansafe-cmap
      Inspect: scansafe p-scansafe fail-open, packet 0, drop 0, reset-drop 0,
v6-fail-close 0
Number of whitelisted connections: 0
Number of connections allowed without scansafe inspection because of "fail-open" config: 0
Number of connections dropped because of "fail-close" config: 0
Number of HTTP connections inspected: 0
Number of HTTPS connections inspected: 0
Number of HTTP connections dropped because of errors: 0
Number of HTTPS connections dropped because of errors: 0
```

Configuration Examples for Cisco Cloud Web Security

Single Mode Example

The following example shows a complete configuration for Cisco Cloud Web Security:

Configure ACLs

We recommend that you split the traffic by creating separate HTTP and HTTPS class maps so that you know how many HTTP and HTTPS packets have gone through.

Then, if you need to troubleshoot you can run debug commands to distinguish how many packets have traversed each class map and find out if you are pushing through more HTTP or HTTPS traffic:

```
hostname(config)# access-list web extended permit tcp any any eq www
hostname(config)# access-list https extended permit tcp any any eq https
```

Configure Class Maps

```
hostname(config)# class-map cmap-http
hostname(config-cmap)# match access-list web

hostname(config)# class-map cmap-https
hostname(config-cmap)# match access-list https
```

Configure Inspection Policy Maps

```
hostname(config)# policy-map type inspect scansafe http-pmap
hostname(config-pmap)# parameters
hostname(config-pmap-p)# default group httptraffic
hostname(config-pmap-p)# http

hostname(config)# policy-map type inspect scansafe https-pmap
hostname(config-pmap)# parameters
hostname(config-pmap-p)# default group httpstraffic
hostname(config-pmap-p)# https
```

Configure Policy Maps

```
hostname(config)# policy-map pmap-webtraffic
hostname(config-pmap)# class cmap-http
hostname(config-pmap-c)# inspect scansafe http-pmap fail-close

hostname(config-pmap)# class cmap-https
hostname(config-pmap-c)# inspect scansafe https-pmap fail-close
```

Configure Service Policy

```
hostname(config)# service-policy pmap-webtraffic interface inside
```

Configure Cloud Web Security on the ASA

```
hostname(config)# scansafe general-options
hostname(cfg-scansafe)# server primary ip 192.168.115.225 web 8080
hostname(cfg-scansafe)# retry-count 5
hostname(cfg-scansafe)# license 366C1D3F5CE67D33D3E9ACEC265261E5
```

Multiple Mode Example

The following example enables Cloud Web Security in context one with the default license and in context two with the authentication key override:

```
! System Context
!
ciscoasa(config)#scansafe general-options
ciscoasa(cfg-scansafe)#server primary ip 180.24.0.62 port 8080
ciscoasa(cfg-scansafe)#retry-count 5
ciscoasa(cfg-scansafe)#license FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
ciscoasa(cfg-scansafe)#publickey <path to public key>
!
context one
 allocate-interface GigabitEthernet0/0.1
 allocate-interface GigabitEthernet0/1.1
 allocate-interface GigabitEthernet0/3.1
 scansafe
 config-url disk0:/one_ctx.cfg
!
context two
 allocate-interface GigabitEthernet0/0.2
 allocate-interface GigabitEthernet0/1.2
 allocate-interface GigabitEthernet0/3.2
 scansafe license 366C1D3F5CE67D33D3E9ACEC265261E5
!
config-url disk0:/two_ctx.cfg
!
```

Whitelist Example

Configure what access-list traffic should be sent to Cloud Web Security:

```
access-list 101 extended permit tcp any4 any4 eq www
access-list 102 extended permit tcp any4 any4 eq https

class-map web
 match access-list 101
class-map https
 match access-list 102
```

To configure the whitelist to ensure user1 is in this access-list range to bypass Cloud Web Security:

```
class-map type inspect scansafe match-any whiteListCmap
 match user LOCAL\user1
```

To attach class-maps to the Cloud Web Security Policy map:

```
policy-map type inspect scansafe ss
 parameters
  default user user1 group group1
  http
 class whiteListCmap
  whitelist

policy-map type inspect scansafe ss2
```

```
 parameters
  default user user1 group group1
  https
 class whiteListCmap
  whitelist
```

After creating this inspect policy, attach it to the policy map to be assigned to the service group:

```
policy-map pmap
 class web
  inspect scansafe ss fail-close
class https
  inspect scansafe ss2 fail-close
```

Then attach the policy map to a service-policy to make it in effect globally or by ASA interface:

```
service-policy pmap interface inside
```

Directory Integration Examples

This section contains various example configurations for directory integration. See also Chapter 38, "Configuring the Identity Firewall," in the general operations configuration guide.

- Configuring the Active Directory Server Using LDAP, page 25-20
- Configuring the Active Directory Agent Using RADIUS, page 25-21
- Creating the ASA as a Client on the AD Agent Server, page 25-21
- Creating a Link Between the AD Agent and DCs, page 25-21
- Testing the AD Agent, page 25-21
- Configuring the Identity Options on the ASA, page 25-21
- Configuring the User Identity Options and Enabling Granular Reporting, page 25-21
- Monitoring the Active Directory Groups, page 25-22
- Downloading the Entire Active-User Database from the Active Directory Server, page 25-22
- Downloading the Database from the AD Agent, page 25-22
- Showing a List of Active Users, page 25-22

Configuring the Active Directory Server Using LDAP

The following example shows how to configure the Active Directory server on your ASA using LDAP:

```
hostname(config)# aaa-server AD protocol ldap
hostname(config-aaa-server-group)# aaa-server AD (inside) host 192.168.116.220
hostname(config-aaa-server-host)# ldap-base-dn DC=ASASCANLAB,DC=local
hostname(config-aaa-server-host)# ldap-scope subtree
hostname(config-aaa-server-host)# server-type microsoft
hostname(config-aaa-server-host)# server-port 389
hostname(config-aaa-server-host)# ldap-login-dn
cn=administrator,cn=Users,dc=asascanlab,dc=local
hostname(config-aaa-server-host)# ldap-login-password Password1
```

Configuring the Active Directory Agent Using RADIUS

The following example shows how to configure the Active Directory Agent on your ASA using RADIUS:

```
hostname(config)# aaa-server adagent protocol radius
hostname(config-aaa-server-group)# ad-agent-mode
hostname(config-aaa-server-group)# aaa-server adagent (inside) host 192.168.116.220
hostname(config-aaa-server-host)# key cisco123
hostname(config-aaa-server-host)# user-identity ad-agent aaa-server adagent
```

Creating the ASA as a Client on the AD Agent Server

The following example shows how to create the ASA as a client on the Active Directory agent server:

```
c:\IBF\CLI\adacfg client create -name ASA5520DEVICE -ip 192.168.116.90 -secret cisco123
```

Creating a Link Between the AD Agent and DCs

The following example shows how to create a link between the Active Directory Agent and all DCs for which you want to monitor logon/logoff events:

```
c:\IBF\CLI\adacfg.exe dc create -name DCSERVER1 -host W2K3DC -domain
W2K3DC.asascanlab.local -user administrator -password Password1
c:\IBF\CLI\adacfg.exe dc list
```

Running the last command should show the status as "UP."

For the AD_Agent to monitor logon/logoff events, you need to ensure that these are logged on ALL DCs that are actively being monitored. To do this, choose:

Start > Administrative Tools > Domain Controller Security Policy

Local policies > Audit Policy > Audit account logon events (success and failure)

Testing the AD Agent

The following example shows how to configure the test Active Directory Agent so that it can communicate with the ASA:

```
hostname# test aaa-server ad-agent adagent
Server IP Address or name: 192.168.116.220
INFO: Attempting Ad-agent test to IP address <192.168.116.220> (timeout: 12 seconds)
INFO: Ad-agent Successful
```

See also the following command: **show user-identity ad-agent**.

Configuring the Identity Options on the ASA

The following example shows how to configure the identity options on the ASA:

```
hostname(config)# user-identity domain ASASCANLAB aaa-server AD
hostname(config)# user-identity default-domain ASASCANLAB
```

Configuring the User Identity Options and Enabling Granular Reporting

The following example shows how to configure the user identity options that send user credentials to the ASA and enable granular user reporting from the proxy server:

```
hostname(config)# user-identity inactive-user-timer minutes 60
hostname(config)# user-identity action netbios-response-fail remove-user-ip
hostname(config)# user-identity user-not-found enable
hostname(config)# user-identity action mac-address-mismatch remove-user-ip
hostname(config)# user-identity ad-agent active-user-database full-download
```

If you are using more than one domain, then enter the following command:

```
hostname(config)# user-identity domain OTHERDOMAINNAME
```

Monitoring the Active Directory Groups

The following example shows how to configure Active Directory groups to be monitored:

```
hostname(config)# user-identity monitor user-group ASASCANLAB\\GROUPNAME1
hostname(config)# user-identity monitor user-group ASASCANLAB\\GROUPNAME2
hostname(config)# user-identity monitor user-group ASASCANLAB\\GROUPNAME3
```

Caution Remember to save your configuration once the above is completed.

Downloading the Entire Active-User Database from the Active Directory Server

The following command updates the specified import user group database by querying the Active Directory server immediately without waiting for the expiration of poll-import-user-group-timer:

```
hostname(config)# user-identity update import-user
```

Downloading the Database from the AD Agent

The following example shows how to manually start the download of the database from the Active Directory Agent if you think the user database is out of sync with Active Directory:

```
hostname(config)# user-identity update active-user-database
```

Showing a List of Active Users

The following example shows how to show the Active users:

```
hostname# show user-identity user active list detail
```

There are two download modes with Identify Firewall: Full download and On-demand.

- Full download—Whenever a user logs into the network, the IDFW tells the ASA the User identity immediately (recommended on the ASA 5510 and above).
- On-demand—Whenever a user logs into the network, the ASA requests the user identity from AD (ADHOC) (recommended on the ASA 5505 due to memory constraints).

Cloud Web Security with Identity Firewall Example

The following example shows how to configure Cloud Web Security with Identity Firewall on the ASA:

```
hostname# sh run
ASA Version 100.8(24)32
!
hostname QFW-201-QASS
```

```
domain-name uk.scansafe.net
enable password liqhNWIOSfzvir2g encrypted
passwd liqhNWIOSfzvir2g encrypted
names
!
interface Ethernet0/0
 nameif inside
 security-level 100
 ip address 192.168.116.90 255.255.255.0
!
interface Ethernet0/1
 nameif outside
 security-level 0
 ip address 192.168.114.90 255.255.254.0
!
interface Ethernet0/2
 shutdown
 no nameif
 no security-level
 no ip address
!
interface Ethernet0/3
 shutdown
 no nameif
 no security-level
 no ip address
!
interface Management0/0
 shutdown
 no nameif
 no security-level
 no ip address
!
boot system disk0:/asa100824-32-k8.bin
ftp mode passive
dns server-group DefaultDNS
 domain-name uk.scansafe.net
object network obj0192.168.116.x
 subnet 192.168.116.0 255.255.255.0
access-list 101 extended permit tcp any any eq www
access-list 101 extended permit tcp any any eq https
access-list web extended permit tcp any any eq www
access-list icmp extended permit icmp any any
access-list https extended permit tcp any any eq https
!
scansafe general-options
 server primary ip 192.168.115.225 web 8080
 retry-count 5
 license 366C1D3F5CE67D33D3E9ACEC26789534f
!
pager lines 24
logging buffered debugging
mtu inside 1500
mtu outside 1500
no failover
icmp unreachable rate-limit 1 burst-size 1
no asdm history enable
arp timeout 14400
!
object network obj0192.168.116.x
 nat (inside,outside) dynamic interface
access-group 101 in interface outside
route outside 0.0.0.0 0.0.0.0 192.168.114.19 1
timeout xlate 3:00:00
timeout pat-xlate 0:00:30
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00
timeout sip 0:30:00 sip_media 0:02:00 sip-invite 0:03:00 sip-disconnect 0:02:00
timeout sip-provisional-media 0:02:00 uauth 0:05:00 absolute
timeout tcp-proxy-reassembly 0:01:00
timeout floating-conn 0:00:00
dynamic-access-policy-record DfltAccessPolicy
aaa-server AD protocol ldap
aaa-server AD (inside) host 192.168.116.220
 server-port 389
 ldap-base-dn DC=ASASCANLAB,DC=local
 ldap-scope subtree
 ldap-login-password *****
 ldap-login-dn cn=administrator,cn=Users,dc=asascanlab,dc=local
 server-type microsoft
aaa-server adagent protocol radius
 ad-agent-mode
aaa-server adagent (inside) host 192.168.116.220
 key *****
user-identity domain ASASCANLAB aaa-server AD
user-identity default-domain ASASCANLAB
user-identity action netbios-response-fail remove-user-ip
user-identity poll-import-user-group-timer hours 1
user-identity ad-agent aaa-server adagent
user-identity user-not-found enable
user-identity monitor user-group ASASCANLAB\\GROUP1
user-identity monitor user-group ASASCANLAB\\GROUPNAME
no snmp-server location
no snmp-server contact
crypto ca trustpool policy
telnet timeout 5
ssh 192.168.0.0 255.255.255.0 inside
ssh 192.168.21.0 255.255.255.0 inside
ssh timeout 30
console timeout 0
threat-detection basic-threat
threat-detection statistics access-list
no threat-detection statistics tcp-intercept
!
class-map cmap-https
 match access-list https
class-map inspection_default
 match default-inspection-traffic
class-map cmap-http
 match access-list web
!
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
policy-map type inspect scansafe ss
 parameters
  default user john group qa
  http
policy-map type inspect scansafe https-pmap
 parameters
  https
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect ip-options
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
policy-map type inspect scansafe http-pmap
 parameters
  default group http-scansafe
  http
policy-map pmap-http
 class cmap-http
  inspect scansafe http-pmap fail-open
 class cmap-https
  inspect scansafe https-pmap fail-open
!
service-policy pmap-http global
prompt hostname context
no call-home reporting anonymous
call-home
 profile CiscoTAC-1
  no active
  destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService
  destination address email callhome@cisco.com
  destination transport-method http
  subscribe-to-alert-group diagnostic
  subscribe-to-alert-group environment
  subscribe-to-alert-group inventory periodic monthly
  subscribe-to-alert-group configuration periodic monthly
  subscribe-to-alert-group telemetry periodic daily
Cryptochecksum:667ba936945b370c394806a63548e7a0
: end
QFW-201-QASS#
```

Related Documents

Related Documents	URL
Cisco ScanSafe Cloud Web Security Configuration Guides	http://www.cisco.com/en/US/products/ps11720/products_installation_and_configuration_guides_list.html

Feature History for Cisco Cloud Web Security

Table 25-1 lists each feature change and the platform release in which it was implemented.

Table 25-1 Feature History for Cloud Web Security

Feature Name	Platform Releases	Feature Information
Cloud Web Security	9.0(1)	This feature was introduced. Cisco Cloud Web Security provides content scanning and other malware protection service for web traffic. It can also redirect and report about web traffic based on user identity. We introduced or modified the following commands: **class-map type inspect scansafe**, **default user group**, **http**[s] (parameters), **inspect scansafe**, **license**, **match user group**, **policy-map type inspect scansafe**, **retry-count**, **scansafe**, **scansafe general-options**, **server** {**primary** | **backup**}, **show conn scansafe**, **show scansafe server**, **show scansafe statistics**, **user-identity monitor**, **whitelist**.

Configuring the Botnet Traffic Filter

Malware is malicious software that is installed on an unknowing host. Malware that attempts network activity such as sending private data (passwords, credit card numbers, key strokes, or proprietary data) can be detected by the Botnet Traffic Filter when the malware starts a connection to a known bad IP address. The Botnet Traffic Filter checks incoming and outgoing connections against a dynamic database of known bad domain names and IP addresses (the *blacklist*), and then logs or blocks any suspicious activity.

You can also supplement the Cisco dynamic database with blacklisted addresses of your choosing by adding them to a static blacklist; if the dynamic database includes blacklisted addresses that you think should not be blacklisted, you can manually enter them into a static *whitelist*. Whitelisted addresses still generate syslog messages, but because you are only targeting blacklist syslog messages, they are informational.

Note If you do not want to use the Cisco dynamic database at all, because of internal requirements, you can use the static blacklist alone if you can identify all the malware sites that you want to target.

This chapter describes how to configure the Botnet Traffic Filter and includes the following sections:

Information About the Botnet Traffic Filter

This section includes information about the Botnet Traffic Filter and includes the following topics:

- Botnet Traffic Filter Actions for Known Addresses, page 26-2
- Botnet Traffic Filter Databases, page 26-2
- How the Botnet Traffic Filter Works, page 26-5

Botnet Traffic Filter Address Types

Addresses monitored by the Botnet Traffic Filter include:

- Known malware addresses—These addresses are on the blacklist identified by the dynamic database and the static blacklist.
- Known allowed addresses—These addresses are on the whitelist. The whitelist is useful when an address is blacklisted by the dynamic database and also identified by the static whitelist.
- Ambiguous addresses—These addresses are associated with multiple domain names, but not all of these domain names are on the blacklist. These addresses are on the *greylist*.
- Unlisted addresses—These addresses are unknown, and not included on any list.

Botnet Traffic Filter Actions for Known Addresses

You can configure the Botnet Traffic Filter to log suspicious activity, and you can optionally configure it to block suspicious traffic automatically.

Unlisted addresses do not generate any syslog messages, but addresses on the blacklist, whitelist, and greylist generate syslog messages differentiated by type. See the "Botnet Traffic Filter Syslog Messaging" section on page 26-17 for more information.

Botnet Traffic Filter Databases

The Botnet Traffic Filter uses two databases for known addresses. You can use both databases together, or you can disable use of the dynamic database and use the static database alone. This section includes the following topics:

- Information About the Dynamic Database, page 26-2
- Information About the Static Database, page 26-3
- Information About the DNS Reverse Lookup Cache and DNS Host Cache, page 26-4

Information About the Dynamic Database

The Botnet Traffic Filter can receive periodic updates for the dynamic database from the Cisco update server. This database lists thousands of known bad domain names and IP addresses.

How the ASA Uses the Dynamic Database

The ASA uses the dynamic database as follows:

1. When the domain name in a DNS reply matches a name in the dynamic database, the Botnet Traffic Filter adds the name and IP address to the *DNS reverse lookup cache*.

2. When the infected host starts a connection to the IP address of the malware site, then the ASA sends a syslog message informing you of the suspicious activity and optionally drops the traffic if you configured the ASA to do so.
3. In some cases, the IP address itself is supplied in the dynamic database, and the Botnet Traffic Filter logs or drops any traffic to that IP address without having to inspect DNS requests.

Database Files

The database files are downloaded from the Cisco update server, and then stored in running memory; they are not stored in flash memory. Be sure to identify a DNS server for the ASA so that it can access the Cisco update server URL. In multiple context mode, the system downloads the database for all contexts using the admin context interface; be sure to identify a DNS server in the admin context.

If you need to delete the database, use the **dynamic-filter database purge** command instead. Be sure to first disable use of the database by entering the **no dynamic-filter use-database** command.

Note To filter on the domain names in the dynamic database, you need to enable DNS packet inspection with Botnet Traffic Filter snooping; the ASA looks inside the DNS packets for the domain name and associated IP address.

Database Traffic Types

The dynamic database includes the following types of addresses:

- Ads—These are advertising networks that deliver banner ads, interstitials, rich media ads, pop-ups, and pop-unders for websites, spyware and adware. Some of these networks send ad-oriented HTML emails and email verification services.
- Data Tracking—These are sources associated with companies and websites that offer data tracking and metrics services to websites and other online entities. Some of these also run small advertising networks.
- Spyware—These are sources that distribute spyware, adware, greyware, and other potentially unwanted advertising software. Some of these also run exploits to install such software.
- Malware—These are sources that use various exploits to deliver adware, spyware and other malware to victim computers. Some of these are associated with rogue online vendors and distributors of dialers which deceptively call premium-rate phone numbers.
- Adult—These are sources associated with adult networks/services offering web hosting for adult content, advertising, content aggregation, registration & billing, and age verification. These may be tied to distribution of adware, spyware, and dialers.
- Bot and Threat Networks—These are rogue systems that control infected computers. They are either systems hosted on threat networks or systems that are part of the botnet itself.

Information About the Static Database

You can manually enter domain names or IP addresses (host or subnet) that you want to tag as bad names in a blacklist. Static blacklist entries are always designated with a Very High threat level. You can also enter names or IP addresses in a whitelist, so that names or addresses that appear on both the *dynamic* blacklist and the whitelist are identified only as whitelist addresses in syslog messages and reports. Note that you see syslog messages for whitelisted addresses even if the address is not also in the dynamic blacklist.

When you add a domain name to the static database, the ASA waits 1 minute, and then sends a DNS request for that domain name and adds the domain name/IP address pairing to the *DNS host cache*. (This action is a background process, and does not affect your ability to continue configuring the ASA). We recommend also enabling DNS packet inspection with Botnet Traffic Filter snooping. The ASA uses Botnet Traffic Filter snooping instead of the regular DNS lookup to resolve static blacklist domain names in the following circumstances:

- The ASA DNS server is unavailable.
- A connection is initiated during the 1 minute waiting period before the ASA sends the regular DNS request.

If DNS snooping is used, when an infected host sends a DNS request for a name on the static database, the ASA looks inside the DNS packets for the domain name and associated IP address and adds the name and IP address to the DNS reverse lookup cache.

If you do not enable Botnet Traffic Filter snooping, and one of the above circumstances occurs, then that traffic will not be monitored by the Botnet Traffic Filter.

Information About the DNS Reverse Lookup Cache and DNS Host Cache

When you use the dynamic database with DNS snooping, entries are added to the DNS reverse lookup cache. If you use the static database, entries are added to the DNS host cache (see the "Information About the Static Database" section on page 26-3 about using the static database with DNS snooping and the DNS reverse lookup cache).

Entries in the DNS reverse lookup cache and the DNS host cache have a time to live (TTL) value provided by the DNS server. The largest TTL value allowed is 1 day (24 hours); if the DNS server provides a larger TTL, it is truncated to 1 day maximum.

For the DNS reverse lookup cache, after an entry times out, the ASA renews the entry when an infected host initiates a connection to a known address, and DNS snooping occurs.

For the DNS host cache, after an entry times out, the ASA periodically requests a refresh for the entry.

For the DNS host cache, the maximum number of blacklist entries and whitelist entries is 1000 each. The number of entries in the DNS reverse lookup cache varies per model.

How the Botnet Traffic Filter Works

Figure 26-1 shows how the Botnet Traffic Filter works with the dynamic database plus DNS inspection with Botnet Traffic Filter snooping.

Figure 26-1 How the Botnet Traffic Filter Works with the Dynamic Database

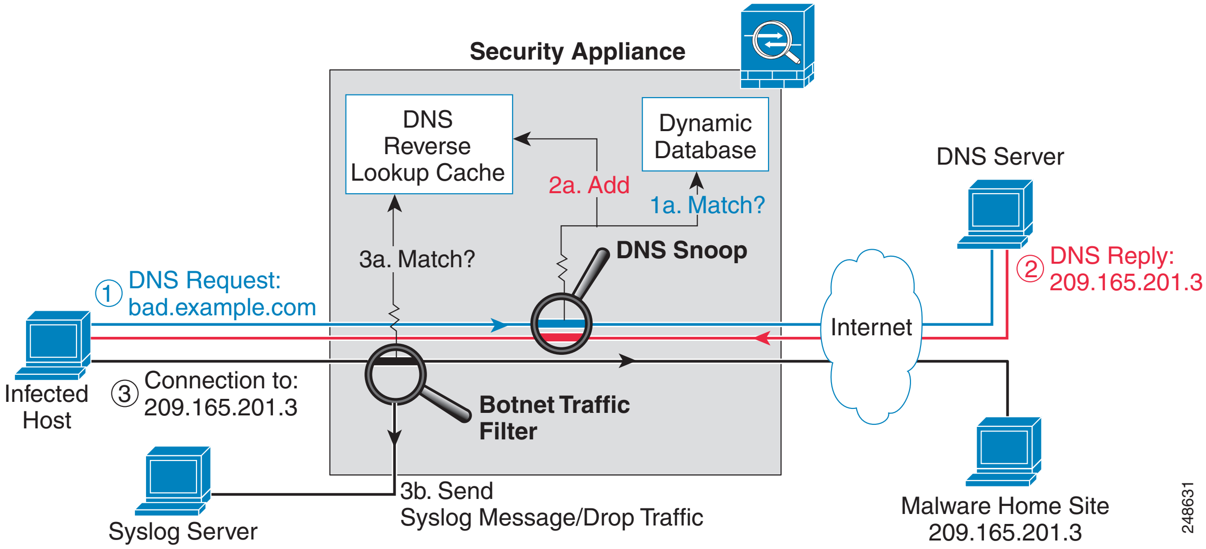
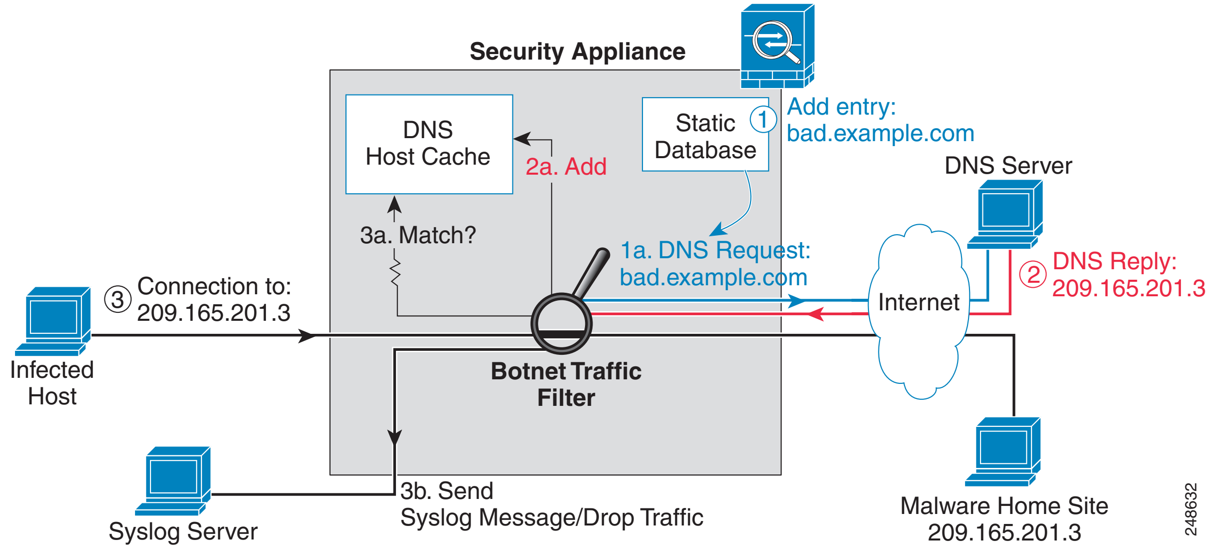


Figure 26-2 shows how the Botnet Traffic Filter works with the static database.

Figure 26-2 How the Botnet Traffic Filter Works with the Static Database


Security Appliance
DNS Host Cache
Static Database
1 Add entry: bad.example.com
2a. Add
DNS Server
3a. Match?
1a. DNS Request: bad.example.com
2 DNS Reply: 209.165.201.3
Internet
3 Connection to: 209.165.201.3
Infected Host
Botnet Traffic Filter
Syslog Server
3b. Send Syslog Message/Drop Traffic
Malware Home Site 209.165.201.3
248632

Licensing Requirements for the Botnet Traffic Filter

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	You need the following licenses: • Botnet Traffic Filter License. • Strong Encryption (3DES/AES) License to download the dynamic database.

Prerequisites for the Botnet Traffic Filter

To use the dynamic database, identify a DNS server for the ASA so that it can access the Cisco update server URL. In multiple context mode, the system downloads the database for all contexts using the admin context interface; be sure to identify a DNS server in the admin context.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

Failover Guidelines

Does not support replication of the DNS reverse lookup cache, DNS host cache, or the dynamic database in Stateful Failover.

IPv6 Guidelines

Does not support IPv6.

Additional Guidelines and Limitations

- TCP DNS traffic is not supported.
- You can add up to 1000 blacklist entries and 1000 whitelist entries in the static database.
- The packet tracer is not supported.

Default Settings

By default, the Botnet Traffic Filter is disabled, as is use of the dynamic database.

For DNS inspection, which is enabled by default, Botnet Traffic Filter snooping is disabled by default.

Configuring the Botnet Traffic Filter

This section includes the following topics:

Task Flow for Configuring the Botnet Traffic Filter

To configure the Botnet Traffic Filter, perform the following steps:

Step 1 Enable use of the dynamic database. See the "Configuring the Dynamic Database" section on page 26-8.

This procedure enables database updates from the Cisco update server, and also enables use of the downloaded dynamic database by the ASA. Disallowing use of the downloaded database is useful in multiple context mode so you can configure use of the database on a per-context basis.

Step 2 (Optional) Add static entries to the database. See the "Adding Entries to the Static Database" section on page 26-9.

This procedure lets you augment the dynamic database with domain names or IP addresses that you want to blacklist or whitelist. You might want to use the static database instead of the dynamic database if you do not want to download the dynamic database over the Internet.

Step 3 Enable DNS snooping. See the "Enabling DNS Snooping" section on page 26-10.

This procedure enables inspection of DNS packets, compares the domain name with those in the dynamic database or the static database (when a DNS server for the ASA is unavailable), and adds the name and IP address to the DNS reverse lookup cache. This cache is then used by the Botnet Traffic Filter when connections are made to the suspicious address.

Step 4 Enable traffic classification and actions for the Botnet Traffic Filter. See the "Enabling Traffic Classification and Actions for the Botnet Traffic Filter" section on page 26-12.

This procedure enables the Botnet Traffic Filter, which compares the source and destination IP address in each initial connection packet to the IP addresses in the dynamic database, static database, DNS reverse lookup cache, and DNS host cache, and sends a syslog message or drops any matching traffic.

Step 5 (Optional) Block traffic manually based on syslog message information. See the "Blocking Botnet Traffic Manually" section on page 26-15.

If you choose not to block malware traffic automatically, you can block traffic manually by configuring an access rule to deny traffic, or by using the **shun** command to block all traffic to and from a host.

Configuring the Dynamic Database

This procedure enables database updates, and also enables use of the downloaded dynamic database by the ASA. In multiple context mode, the system downloads the database for all contexts using the admin context interface. You can configure *use* of the database on a per-context basis.

By default, downloading and using the dynamic database is disabled.

Prerequisites

Enable ASA use of a DNS server according to the "Configuring the DNS Server" section on page 16-8 in the general operations configuration guide. In multiple context mode, the system downloads the database for all contexts using the admin context interface; be sure to identify a DNS server in the admin context.

Detailed Steps

	Command	Purpose

Step 1	**`dynamic-filter updater-client enable`**
	Example:
	`ciscoasa(config)# dynamic-filter updater-client enable`
		Enables downloading of the dynamic database from the Cisco update server. In multiple context mode, enter this command in the system execution space. If you do not have a database already installed on the ASA, it downloads the database after approximately 2 minutes. The update server determines how often the ASA polls the server for future updates, typically every hour.

Step 2	(Multiple context mode only)
	`changeto context` *`context_name`*
	Example:
	`ciscoasa# changeto context admin`
	`ciscoasa/admin#`
		Changes to the context so that you can configure use of the database on a per-context basis.

Step 3	**`dynamic-filter use-database`**
	Example:
	`ciscoasa(config)# dynamic-filter use-database`
		Enables use of the dynamic database. In multiple context mode, enter this command in the context execution space.

Examples

The following multiple mode example enables downloading of the dynamic database, and enables use of the database in context1 and context2:

```
ciscoasa(config)# dynamic-filter updater-client enable
ciscoasa(config)# changeto context context1
ciscoasa/context1(config)# dynamic-filter use-database
ciscoasa/context1(config)# changeto context context2
ciscoasa/context2(config)# dynamic-filter use-database
```

The following single mode example enables downloading of the dynamic database, and enables use of the database:

```
ciscoasa(config)# dynamic-filter updater-client enable
ciscoasa(config)# dynamic-filter use-database
```

What to Do Next

See the "Adding Entries to the Static Database" section on page 26-9.

Adding Entries to the Static Database

The static database lets you augment the dynamic database with domain names or IP addresses that you want to blacklist or whitelist. Static blacklist entries are always designated with a Very High threat level. See the "Information About the Static Database" section on page 26-3 for more information.

Prerequisites

- In multiple context mode, perform this procedure in the context execution space.
- Enable ASA use of a DNS server according to the "Configuring the DNS Server" section on page 16-8 in the general operations configuration guide.

Detailed Steps

	Command	Purpose
Step 1	**dynamic-filter blacklist** **Example:** `ciscoasa(config)# dynamic-filter blacklist`	Edits the Botnet Traffic Filter blacklist.
Step 2	Enter one or both of the following:	
	name *domain_name* **Example:** `ciscoasa(config-llist)# name bad.example.com`	Adds a name to the blacklist. You can enter this command multiple times for multiple entries. You can add up to 1000 blacklist entries.
	address *ip_address mask* **Example:** `ciscoasa(config-llist)# address 10.1.1.1 255.255.255.255`	Adds an IP address to the blacklist. You can enter this command multiple times for multiple entries. The *mask* can be for a single host or for a subnet.
Step 3	**dynamic-filter whitelist** **Example:** `ciscoasa(config)# dynamic-filter whitelist`	Edits the Botnet Traffic Filter whitelist.
Step 4	Enter one or both of the following:	

Command	Purpose
name *domain_name* **Example:** `ciscoasa(config-llist)# name good.example.com`	Adds a name to the whitelist. You can enter this command multiple times for multiple entries. You can add up to 1000 whitelist entries.
address *ip_address mask* **Example:** `ciscoasa(config-llist)# address 10.1.1.2 255.255.255.255`	Adds an IP address to the whitelist. You can enter this command multiple times for multiple entries. The *mask* can be for a single host or for a subnet.

Examples

The following example creates entries for the blacklist and whitelist:

```
ciscoasa(config)# dynamic-filter blacklist
ciscoasa(config-llist)# name bad1.example.com
ciscoasa(config-llist)# name bad2.example.com
ciscoasa(config-llist)# address 10.1.1.1 255.255.255.0
ciscoasa(config-llist)# dynamic-filter whitelist
ciscoasa(config-llist)# name good.example.com
ciscoasa(config-llist)# name great.example.com
ciscoasa(config-llist)# name awesome.example.com
ciscoasa(config-llist)# address 10.1.1.2 255.255.255.255
```

What to Do Next

See the "Enabling DNS Snooping" section on page 26-10.

Enabling DNS Snooping

This procedure enables inspection of DNS packets and enables Botnet Traffic Filter snooping, which compares the domain name with those on the dynamic database or static database, and adds the name and IP address to the Botnet Traffic Filter DNS reverse lookup cache. This cache is then used by the Botnet Traffic Filter when connections are made to the suspicious address.

The following procedure creates an interface-specific service policy for DNS inspection. See the "DNS Inspection" section on page 10-1 and Chapter 1, "Configuring a Service Policy Using the Modular Policy Framework," for detailed information about configuring advanced DNS inspection options using the Modular Policy Framework.

Prerequisites

In multiple context mode, perform this procedure in the context execution space.

Restrictions

TCP DNS traffic is not supported.

Default DNS Inspection Configuration and Recommended Configuration

The default configuration for DNS inspection inspects all UDP DNS traffic on all interfaces, and does not have DNS snooping enabled.

We suggest that you enable DNS snooping only on interfaces where external DNS requests are going. Enabling DNS snooping on all UDP DNS traffic, including that going to an internal DNS server, creates unnecessary load on the ASA.

For example, if the DNS server is on the outside interface, you should enable DNS inspection with snooping for all UDP DNS traffic on the outside interface. See the "Examples" section for the recommended commands for this configuration.

Detailed Steps

	Command	Purpose

Step 1	**`class-map`** *`name`*
	Example:
	`ciscoasa(config)# class-map dynamic-filter_snoop_class`
		Creates a class map to identify the traffic for which you want to inspect DNS.

Step 2	**`match`** *`parameters`*
	Example:
	`ciscoasa(config-cmap)# match port udp eq domain`
		Specifies traffic for the class map. See the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12 for more information about available parameters. For example, you can specify an ACL for DNS traffic to and from certain addresses, or you can specify all UDP DNS traffic.

Step 3	**`policy-map`** *`name`*
	Example:
	`ciscoasa(config)# policy-map dynamic-filter_snoop_policy`
		Adds or edits a policy map so you can set the actions to take with the class map traffic.

Step 4	**`class`** *`name`*
	Example:
	`ciscoasa(config-pmap)# class dynamic-filter_snoop_class`
		Identifies the class map you created in Step 1.

	Command	Purpose
Step 5	**inspect dns** [*map_name*] **dynamic-filter-snoop** **Example:** `ciscoasa(config-pmap-c)# inspect dns preset_dns_map dynamic-filter-snoop`	Enables DNS inspection with Botnet Traffic Filter snooping. To use the default DNS inspection policy map for the *map_name*, specify **preset_dns_map** for the map name. See the "DNS Inspection" section on page 10-1 for more information about creating a DNS inspection policy map.
Step 6	**service-policy** *policymap_name* **interface** *interface_name* **Example:** `ciscoasa(config)# service-policy dynamic-filter_snoop_policy interface outside`	Activates the policy map on an interface. The interface-specific policy overrides the global policy. You can only apply one policy map to each interface.

Examples

The following recommended configuration creates a class map for all UDP DNS traffic, enables DNS inspection and Botnet Traffic Filter snooping with the default DNS inspection policy map, and applies it to the outside interface:

```
ciscoasa(config)# class-map dynamic-filter_snoop_class
ciscoasa(config-cmap)# match port udp eq domain
ciscoasa(config-cmap)# policy-map dynamic-filter_snoop_policy
ciscoasa(config-pmap)# class dynamic-filter_snoop_class
ciscoasa(config-pmap-c)# inspect dns preset_dns_map dynamic-filter-snoop
ciscoasa(config-pmap-c)# service-policy dynamic-filter_snoop_policy interface outside
```

What to Do Next

See the "Enabling Traffic Classification and Actions for the Botnet Traffic Filter" section on page 26-12.

Enabling Traffic Classification and Actions for the Botnet Traffic Filter

This procedure enables the Botnet Traffic Filter. The Botnet Traffic Filter compares the source and destination IP address in each initial connection packet to the following:

- Dynamic database IP addresses
- Static database IP addresses
- DNS reverse lookup cache (for dynamic database domain names)
- DNS host cache (for static database domain names)

When an address matches, the ASA sends a syslog message. The only additional action currently available is to drop the connection.

Prerequisites

In multiple context mode, perform this procedure in the context execution space.

Recommended Configuration

Although DNS snooping is not required, we recommend configuring DNS snooping for maximum use of the Botnet Traffic Filter (see the "Enabling DNS Snooping" section on page 26-10). Without DNS snooping for the dynamic database, the Botnet Traffic Filter uses only the static database entries, plus any IP addresses in the dynamic database; domain names in the dynamic database are not used.

We recommend enabling the Botnet Traffic Filter on all traffic on the Internet-facing interface, and enabling dropping of traffic with a severity of moderate and higher. See the "Examples" section for the recommended commands used for this configuration.

Detailed Steps

	Command	Purpose

Step 1	(Optional)
	access-list *access_list_name* **extended** {**deny** | **permit**} *protocol source_address mask* [*operator port*] *dest_address mask* [*operator port*]
	Example:
	`ciscoasa(config)# access-list dynamic-filter_acl extended permit tcp any any eq 80`
	`ciscoasa(config)# access-list dynamic-filter_acl_subset extended permit tcp 10.1.1.0 255.255.255.0 any eq 80`
		Identifies the traffic that you want to monitor or drop. If you do not create an ACL for monitoring, by default you monitor all traffic. You can optionally use an ACL to identify a subset of monitored traffic that you want to drop; be sure the ACL is a subset of the monitoring ACL. See Chapter 19, "Adding an Extended Access Control List," in the general operations configuration guide for more information about creating an ACL.

Step 2	**dynamic-filter enable** [**interface** *name*] [**classify-list** *access_list*] **Example:** `ciscoasa(config)# dynamic-filter enable interface outside classify-list dynamic-filter_acl`	Enables the Botnet Traffic Filter; without any options, this command monitors all traffic. We recommend enabling the Botnet Traffic Filter on all traffic on the Internet-facing interface using the **interface** keyword. You can optionally limit monitoring to specific traffic by using the **classify-list** keyword with an ACL. You can enter this command one time for each interface and one time for the global policy (where you do not specify the **interface** keyword). Each interface and global command can have an optional **classify-list** keyword. Any interface-specific commands take precedence over the global command.

	Command	Purpose
Step 3	(Optional) **dynamic-filter drop blacklist** [**interface** *name*] [**action-classify-list** *subset_access_list*] [**threat-level** {**eq** *level* | **range** *min max*}] **Example:** `ciscoasa(config)# dynamic-filter drop blacklist interface outside action-classify-list dynamic-filter_acl_subset threat-level range moderate very-high`	Automatically drops malware traffic. To manually drop traffic, see the "Blocking Botnet Traffic Manually" section on page 26-15. Be sure to first configure a **dynamic-filter enable** command to monitor any traffic you also want to drop. You can set an interface policy using the **interface** keyword, or a global policy (where you do not specify the **interface** keyword). Any interface-specific commands take precedence over the global command. You can enter this command multiple times for each interface and global policy. The **action-classify-list** keyword limits the traffic dropped to a subset of monitored traffic. The dropped traffic must always be equal to or a subset of the monitored traffic. For example, if you specify an ACL for the **dynamic-filter enable** command, and you specify the **action-classify-list** for this command, then it must be a subset of the **dynamic-filter enable** ACL. Make sure you do not specify overlapping traffic in multiple commands for a given interface/global policy. Because you cannot control the exact order that commands are matched, overlapping traffic means you do not know which command will be matched. For example, do not specify both a command that matches all traffic (without the **action-classify-list** keyword) as well as a command with the **action-classify-list** keyword for a given interface. In this case, the traffic might never match the command with the **action-classify-list** keyword. Similarly, if you specify multiple commands with the **action-classify-list** keyword, make sure each ACL is unique, and that the networks do not overlap. You can additionally limit the traffic dropped by setting the threat level. If you do not explicitly set a threat level, the level used is **threat-level range moderate very-high**. **Note** We highly recommend using the default setting unless you have strong reasons for changing the setting. The *level* and *min* and *max* options are: • **very-low** • **low** • **moderate** • **high** • **very-high** **Note** Static blacklist entries are always designated with a Very High threat level.

	Command	Purpose
Step 4	(Optional) **dynamic-filter ambiguous-is-black** **Example:** `ciscoasa(config)# dynamic-filter ambiguous-is-black`	If you configured the **dynamic-filter drop blacklist** command, then this command treats greylisted traffic as blacklisted traffic for dropping purposes. If you do not enable this command, greylisted traffic will not be dropped. See the "Botnet Traffic Filter Address Types" section on page 26-2 for more information about the greylist.

Examples

The following recommended configuration monitors all traffic on the outside interface and drops all traffic at a threat level of moderate or higher:

```
ciscoasa(config)# dynamic-filter enable interface outside
ciscoasa(config)# dynamic-filter drop blacklist interface outside
```

If you decide not to monitor all traffic, you can limit the traffic using an ACL. The following example monitors only port 80 traffic on the outside interface, and drops traffic threat level very-high only:

```
ciscoasa(config)# access-list dynamic-filter_acl extended permit tcp any any eq 80
ciscoasa(config)# dynamic-filter enable interface outside classify-list dynamic-filter_acl
ciscoasa(config)# dynamic-filter drop blacklist interface outside threat-level eq
very-high
```

Blocking Botnet Traffic Manually

If you choose not to block malware traffic automatically (see the "Enabling Traffic Classification and Actions for the Botnet Traffic Filter" section on page 26-12), you can block traffic manually by configuring an access rule to deny traffic, or by using the **shun** command tool to block all traffic to and from a host.

For example, you receive the following syslog message:

```
ASA-4-338002: Dynamic Filter permitted black listed TCP traffic from inside:10.1.1.45/6798
(209.165.201.1/7890) to outside:209.165.202.129/80 (209.165.202.129/80), destination
209.165.202.129 resolved from dynamic list: bad.example.com
```

You can then perform one of the following actions:

- Create an access rule to deny traffic.

  For example, using the syslog message above, you might want to deny traffic from the infected host at 10.1.1.45 to the malware site at 209.165.202.129. Or, if there are many connections to different blacklisted addresses, you can create an ACL to deny all traffic from 10.1.1.45 until you resolve the infection on the host computer. For example, the following commands deny all traffic from 10.1.1.5 to 209.165.202.129, but permits all other traffic on the inside interface:

  ```
  ciscoasa(config)# access-list BLOCK_OUT extended deny ip host 10.1.1.45 host
  209.165.202.129
  ciscoasa(config)# access-list BLOCK_OUT extended permit ip any any
  ciscoasa(config)# access-group BLOCK_OUT in interface inside
  ```

  See Chapter 19, "Adding an Extended Access Control List," in the general operations configuration guide for more information about creating an ACL, and see Chapter 6, "Configuring Access Rules," for information about applying the ACL to the interface.

> **Note** ACLs block all future connections. To block the current connection, if it is still active, enter the **clear conn** command. For example, to clear only the connection listed in the syslog message, enter the **clear conn address 10.1.1.45 address 209.165.202.129** command. See the command reference for more information.

- Shun the infected host.

  Shunning blocks all connections from the host, so you should use an ACL if you want to block connections to certain destination addresses and ports. To shun a host, enter the following command. To drop the current connection as well as blocking all future connections, enter the destination address, source port, destination port, and optional protocol.

  ```
  ciscoasa(config)# shun src_ip [dst_ip src_port dest_port [protocol]]
  ```

  For example, to block future connections from 10.1.1.45, and also drop the current connection to the malware site in the syslog message, enter:

  ```
  ciscoasa(config)# shun 10.1.1.45 209.165.202.129 6798 80
  ```

  See "Blocking Unwanted Connections" section on page 28-2 for more information about shunning.

After you resolve the infection, be sure to remove the ACL or the shun. To remove the shun, enter **no shun** *src_ip*.

Searching the Dynamic Database

If you want to check if a domain name or IP address is included in the dynamic database, you can search the database for a string.

Detailed Steps

Command	Purpose
`dynamic-filter database find` *string* **Example:** `ciscoasa# dynamic-filter database find`	Searches the dynamic database for a domain name or IP address. The *string* can be the complete domain name or IP address, or you can enter part of the name or address, with a minimum search string of 3 characters. If there are multiple matches, the first two matches are shown. To refine your search for a more specific match, enter a longer string. **Note** Regular expressions are not supported for the database search.

Examples

The following example searches on the string "example.com", and finds 1 match:

```
ciscoasa# dynamic-filter database find bad.example.com

   bad.example.com
Found 1 matches
```

The following example searches on the string "bad", and finds more than 2 matches:

```
ciscoasa# dynamic-filter database find bad

   bad.example.com
```

```
   bad.example.net
Found more than 2 matches, enter a more specific string to find an exact
match
```

Monitoring the Botnet Traffic Filter

Whenever a known address is classified by the Botnet Traffic Filter, then a syslog message is generated. You can also monitor Botnet Traffic Filter statistics and other parameters by entering commands on the ASA. This section includes the following topics:

- Botnet Traffic Filter Syslog Messaging, page 26-17
- Botnet Traffic Filter Commands, page 26-17

Botnet Traffic Filter Syslog Messaging

The Botnet Traffic Filter generates detailed syslog messages numbered 338*nnn*. Messages differentiate between incoming and outgoing connections, blacklist, whitelist, or greylist addresses, and many other variables. (The greylist includes addresses that are associated with multiple domain names, but not all of these domain names are on the blacklist.)

See the syslog messages guide for detailed information about syslog messages.

Botnet Traffic Filter Commands

To monitor the Botnet Traffic Filter, enter one of the following commands:

Command	Purpose
show dynamic-filter statistics [**interface** *name*] [**detail**]	Shows how many connections were classified as whitelist, blacklist, and greylist connections, and how many connections were dropped. (The greylist includes addresses that are associated with multiple domain names, but not all of these domain names are on the blacklist.) The **detail** keyword shows how many packets at each threat level were classified or dropped. To clear the statistics, enter the **clear dynamic-filter statistics** [**interface** *name*] command.
show dynamic-filter reports top [**malware-sites** | **malware-ports** | **infected-hosts**]	Generates reports of the top 10 malware sites, ports, and infected hosts monitored. The top 10 malware-sites report includes the number of connections dropped, and the threat level and category of each site. This report is a snapshot of the data, and may not match the top 10 items since the statistics started to be collected. To clear the report data, enter the **clear dynamic-filter reports top** command.

Command	Purpose
show dynamic-filter reports infected-hosts {**max-connections** | **latest-active** | **highest-threat** | **subnet** *ip_address netmask* | **all**}	Generates reports about infected hosts. These reports contain detailed history about infected hosts, showing the correlation between infected hosts, visited malware sites, and malware ports. The **max-connections** keyword shows the 20 infected hosts with the most number of connections. The **latest-active** keyword shows the 20 hosts with the most recent activity. The **highest-threat** keyword shows the 20 hosts that connected to the malware sites with the highest threat level. The **subnet** keyword shows up to 20 hosts within the specified subnet. The **all** keyword shows all buffered infected-hosts information. This display might include thousands of entries. You might want to use ASDM to generate a PDF file instead of using the CLI. To clear the report data, enter the **clear dynamic-filter reports infected-hosts** command.
show dynamic-filter updater-client	Shows information about the updater server, including the server IP address, the next time the ASA will connect with the server, and the database version last installed.
show dynamic-filter dns-snoop [**detail**]	Shows the Botnet Traffic Filter DNS snooping summary, or with the **detail** keyword, the actual IP addresses and names. All inspected DNS data is included in this output, and not just matching names in the blacklist. DNS data from static entries are not included. To clear the DNS snooping data, enter the **clear dynamic-filter dns-snoop** command.
show dynamic-filter data	Shows information about the dynamic database, including when the dynamic database was last downloaded, the version of the database, how many entries the database contains, and 10 sample entries.
show asp table dynamic-filter [**hits**]	Shows the Botnet Traffic Filter rules that are installed in the accelerated security path.

Examples

The following is sample output from the **show dynamic-filter statistics** command:

```
ciscoasa# show dynamic-filter statistics
Enabled on interface outside
 Total conns classified 11, ingress 11, egress 0
 Total whitelist classified 0, ingress 0, egress 0
 Total greylist classified 0, dropped 0, ingress 0, egress 0
 Total blacklist classified 11, dropped 5, ingress 11, egress 0
Enabled on interface inside
 Total conns classified 1182, ingress 1182, egress 0
 Total whitelist classified 3, ingress 3, egress 0
 Total greylist classified 0, dropped 0, ingress 0, egress 0
 Total blacklist classified 1179, dropped 1000, ingress 1179, egress 0
```

The following is sample output from the **show dynamic-filter reports top malware-sites** command:

```
ciscoasa# show dynamic-filter reports top malware-sites
Site                                 Connections logged dropped Threat Level Category
--------------------------------------------------------------------------------------
bad1.example.com (10.67.22.34)                    11      0          2      Botnet
bad2.example.com (209.165.200.225)                 8      8          3      Virus
bad1.cisco.example(10.131.36.158)                  6      6          3      Virus
bad2.cisco.example(209.165.201.1)                  2      2          3      Trojan
```

```
horrible.example.net(10.232.224.2)                   2     2          3      Botnet
nono.example.org(209.165.202.130)                    1     1          3      Virus

Last clearing of the top sites report: at 13:41:06 UTC Jul 15 2009
```

The following is sample output from the **show dynamic-filter reports top malware-ports** command:

```
ciscoasa# show dynamic-filter reports top malware-ports
Port                                       Connections logged
----------------------------------------------------------------------
tcp 1000                                          617
tcp 2001                                          472
tcp 23                                             22
tcp 1001                                           19
udp 2000                                           17
udp 2001                                           17
tcp 8080                                            9
tcp 80                                              3
tcp >8192                                           2

Last clearing of the top sites report: at 13:41:06 UTC Jul 15 2009
```

The following is sample output from the **show dynamic-filter reports top infected-hosts** command:

```
ciscoasa# show dynamic-filter reports top infected-hosts
Host                                       Connections logged
----------------------------------------------------------------------
10.10.10.51(inside)                              1190
10.12.10.10(inside)                               10
10.10.11.10(inside)                                5

Last clearing of the top infected-hosts report: at 13:41:06 UTC Jul 15 2009
```

Configuration Examples for the Botnet Traffic Filter

This section includes the recommended configuration for single and multiple context mode, as well as other possible configurations. This section includes the following topics:

- Recommended Configuration Example, page 26-19
- Other Configuration Examples, page 26-20

Recommended Configuration Example

The following recommended example configuration for single context mode enables downloading of the dynamic database, and enables use of the database. It creates a class map for all UDP DNS traffic, enables DNS inspection and Botnet Traffic Filter snooping with the default DNS inspection policy map, and applies it to the outside interface, the Internet-facing interface.

Example 26-1 Single Mode Botnet Traffic Filter Recommended Example

```
ciscoasa(config)# dynamic-filter updater-client enable
ciscoasa(config)# dynamic-filter use-database
ciscoasa(config)# class-map dynamic-filter_snoop_class
ciscoasa(config-cmap)# match port udp eq domain
ciscoasa(config-cmap)# policy-map dynamic-filter_snoop_policy
ciscoasa(config-pmap)# class dynamic-filter_snoop_class
ciscoasa(config-pmap-c)# inspect dns preset_dns_map dynamic-filter-snoop
ciscoasa(config-pmap-c)# service-policy dynamic-filter_snoop_policy interface outside
ciscoasa(config)# dynamic-filter enable interface outside
ciscoasa(config)# dynamic-filter drop blacklist interface outside
```

The following recommended example configuration for multiple context mode enables the Botnet Traffic Filter for two contexts:

Example 26-2 Multiple Mode Botnet Traffic Filter Recommended Example

```
ciscoasa(config)# dynamic-filter updater-client enable

ciscoasa(config)# changeto context context1

ciscoasa/context1(config)# dynamic-filter use-database
ciscoasa/context1(config)# class-map dynamic-filter_snoop_class
ciscoasa/context1(config-cmap)# match port udp eq domain
ciscoasa/context1(config-cmap)# policy-map dynamic-filter_snoop_policy
ciscoasa/context1(config-pmap)# class dynamic-filter_snoop_class
ciscoasa/context1(config-pmap-c)# inspect dns preset_dns_map dynamic-filter-snoop
ciscoasa/context1(config-pmap-c)# service-policy dynamic-filter_snoop_policy interface
outside
ciscoasa/context1(config)# dynamic-filter enable interface outside
ciscoasa/context1(config)# dynamic-filter drop blacklist interface outside

ciscoasa/context1(config)# changeto context context2

ciscoasa/context2(config)# dynamic-filter use-database
ciscoasa/context2(config)# class-map dynamic-filter_snoop_class
ciscoasa/context2(config-cmap)# match port udp eq domain
ciscoasa/context2(config-cmap)# policy-map dynamic-filter_snoop_policy
ciscoasa/context2(config-pmap)# class dynamic-filter_snoop_class
ciscoasa/context2(config-pmap-c)# inspect dns preset_dns_map dynamic-filter-snoop
ciscoasa/context2(config-pmap-c)# service-policy dynamic-filter_snoop_policy interface
outside
ciscoasa/context2(config)# dynamic-filter enable interface outside
ciscoasa/context2(config)# dynamic-filter drop blacklist interface outside
```

Other Configuration Examples

The following sample configuration adds static entries are to the blacklist and to the whitelist. Then, it monitors all port 80 traffic on the outside interface, and drops blacklisted traffic. It also treats greylist addresses as blacklisted addresses.

```
ciscoasa(config)# dynamic-filter updater-client enable

ciscoasa(config)# changeto context context1

ciscoasa/context1(config)# dynamic-filter use-database
ciscoasa/context1(config)# class-map dynamic-filter_snoop_class
ciscoasa/context1(config-cmap)# match port udp eq domain
ciscoasa/context1(config-cmap)# policy-map dynamic-filter_snoop_policy
ciscoasa/context1(config-pmap)# class dynamic-filter_snoop_class
ciscoasa/context1(config-pmap-c)# inspect dns preset_dns_map dynamic-filter-snoop
ciscoasa/context1(config-pmap-c)# service-policy dynamic-filter_snoop_policy interface
outside
ciscoasa/context1(config-pmap-c)# dynamic-filter blacklist
ciscoasa/context1(config-llist)# name bad1.example.com
ciscoasa/context1(config-llist)# name bad2.example.com
```

```
ciscoasa/context1(config-llist)# address 10.1.1.1 255.255.255.0
ciscoasa/context1(config-llist)# dynamic-filter whitelist
ciscoasa/context1(config-llist)# name good.example.com
ciscoasa/context1(config-llist)# name great.example.com
ciscoasa/context1(config-llist)# name awesome.example.com
ciscoasa/context1(config-llist)# address 10.1.1.2 255.255.255.255
ciscoasa/context1(config-llist)# access-list dynamic-filter_acl extended permit tcp any
any eq 80
ciscoasa/context1(config)# dynamic-filter enable interface outside classify-list
dynamic-filter_acl
ciscoasa/context1(config)# dynamic-filter drop blacklist interface outside
ciscoasa/context1(config)# dynamic-filter ambiguous-is-black

ciscoasa/context1(config)# changeto context context2

ciscoasa/context2(config)# dynamic-filter use-database
ciscoasa/context2(config)# class-map dynamic-filter_snoop_class
ciscoasa/context2(config-cmap)# match port udp eq domain
ciscoasa/context2(config-cmap)# policy-map dynamic-filter_snoop_policy
ciscoasa/context2(config-pmap)# class dynamic-filter_snoop_class
ciscoasa/context2(config-pmap-c)# inspect dns preset_dns_map dynamic-filter-snoop
ciscoasa/context2(config-pmap-c)# service-policy dynamic-filter_snoop_policy interface
outside
ciscoasa/context2(config-pmap-c)# dynamic-filter blacklist
ciscoasa/context2(config-llist)# name bad1.example.com
ciscoasa/context2(config-llist)# name bad2.example.com
ciscoasa/context2(config-llist)# address 10.1.1.1 255.255.255.0
ciscoasa/context2(config-llist)# dynamic-filter whitelist
ciscoasa/context2(config-llist)# name good.example.com
ciscoasa/context2(config-llist)# name great.example.com
ciscoasa/context2(config-llist)# name awesome.example.com
ciscoasa/context2(config-llist)# address 10.1.1.2 255.255.255.255
ciscoasa/context2(config-llist)# access-list dynamic-filter_acl extended permit tcp any
any eq 80
ciscoasa/context2(config)# dynamic-filter enable interface outside classify-list
dynamic-filter_acl
ciscoasa/context2(config)# dynamic-filter drop blacklist interface outside
ciscoasa/context2(config)# dynamic-filter ambiguous-is-black
```

Where to Go Next

- To configure the syslog server, see Chapter 41, “Configuring Logging,” in the general operations configuration guide.
- To configure an ACL to block traffic, see Chapter 19, “Adding an Extended Access Control List,” in the general operations configuration guide and also see Chapter 6, “Configuring Access Rules,” for information about applying the ACL to the interface.
- To shun connections, see the “Blocking Unwanted Connections” section on page 28-2.

Feature History for the Botnet Traffic Filter

Table 26-1 lists each feature change and the platform release in which it was implemented.

Table 26-1 Feature History for the Botnet Traffic Filter

Feature Name	Platform Releases	Feature Information
Botnet Traffic Filter	8.2(1)	This feature was introduced.
Automatic blocking, and blacklist category and threat level reporting.	8.2(2)	The Botnet Traffic Filter now supports automatic blocking of blacklisted traffic based on the threat level. You can also view the category and threat level of malware sites in statistics and reports. The 1 hour timeout for reports for top hosts was removed; there is now no timeout. The following commands were introduced or modified: **dynamic-filter ambiguous-is-black**, **dynamic-filter drop blacklist**, **show dynamic-filter statistics**, **show dynamic-filter reports infected-hosts**, and **show dynamic-filter reports top**.

Configuring Threat Detection

This chapter describes how to configure threat detection statistics and scanning threat detection and includes the following sections:

Information About Threat Detection

The threat detection feature consists of the following elements:

- Different levels of statistics gathering for various threats.

  Threat detection statistics can help you manage threats to your ASA; for example, if you enable scanning threat detection, then viewing statistics can help you analyze the threat. You can configure two types of threat detection statistics:

  - Basic threat detection statistics—Includes information about attack activity for the system as a whole. Basic threat detection statistics are enabled by default and have no performance impact.
  - Advanced threat detection statistics—Tracks activity at an object level, so the ASA can report activity for individual hosts, ports, protocols, or ACLs. Advanced threat detection statistics can have a major performance impact, depending on the statistics gathered, so only the ACL statistics are enabled by default.

- Scanning threat detection, which determines when a host is performing a scan.

  You can optionally shun any hosts determined to be a scanning threat.

Licensing Requirements for Threat Detection

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Configuring Basic Threat Detection Statistics

Basic threat detection statistics include activity that might be related to an attack, such as a DoS attack.

This section includes the following topics:

- Information About Basic Threat Detection Statistics, page 27-2
- Guidelines and Limitations, page 27-3
- Default Settings, page 27-3
- Configuring Basic Threat Detection Statistics, page 27-4
- Monitoring Basic Threat Detection Statistics, page 27-5
- Feature History for Basic Threat Detection Statistics, page 27-6

Information About Basic Threat Detection Statistics

Using basic threat detection statistics, the ASA monitors the rate of dropped packets and security events due to the following reasons:

- Denial by ACLs
- Bad packet format (such as invalid-ip-header or invalid-tcp-hdr-length)
- Connection limits exceeded (both system-wide resource limits, and limits set in the configuration)
- DoS attack detected (such as an invalid SPI, Stateful Firewall check failure)
- Basic firewall checks failed (This option is a combined rate that includes all firewall-related packet drops in this bulleted list. It does not include non-firewall-related drops such as interface overload, packets failed at application inspection, and scanning attack detected.)
- Suspicious ICMP packets detected
- Packets failed application inspection
- Interface overload
- Scanning attack detected (This option monitors scanning attacks; for example, the first TCP packet is not a SYN packet, or the TCP connection failed the 3-way handshake. Full scanning threat detection (see the "Configuring Scanning Threat Detection" section on page 27-15) takes this scanning attack rate information and acts on it by classifying hosts as attackers and automatically shunning them, for example.)
- Incomplete session detection such as TCP SYN attack detected or no data UDP session attack detected

When the ASA detects a threat, it immediately sends a system log message (733100). The ASA tracks two types of rates: the average event rate over an interval, and the burst event rate over a shorter burst interval. The burst rate interval is 1/30th of the average rate interval or 10 seconds, whichever is higher.

For each received event, the ASA checks the average and burst rate limits; if both rates are exceeded, then the ASA sends two separate system messages, with a maximum of one message for each rate type per burst period.

Basic threat detection affects performance only when there are drops or potential threats; even in this scenario, the performance impact is insignificant.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature:

Security Context Guidelines

Supported in single mode only. Multiple mode is not supported.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

Types of Traffic Monitored

Only through-the-box traffic is monitored; to-the-box traffic is not included in threat detection.

Default Settings

Basic threat detection statistics are enabled by default.

Table 27-1 lists the default settings. You can view all these default settings using the **show running-config all threat-detection** command.

Table 27-1 Basic Threat Detection Default Settings

	Trigger Settings	
Packet Drop Reason	**Average Rate**	**Burst Rate**
• DoS attack detected • Bad packet format • Connection limits exceeded • Suspicious ICMP packets detected	100 drops/sec over the last 600 seconds.	400 drops/sec over the last 20 second period.
	80 drops/sec over the last 3600 seconds.	320 drops/sec over the last 120 second period.
Scanning attack detected	5 drops/sec over the last 600 seconds.	10 drops/sec over the last 20 second period.
	4 drops/sec over the last 3600 seconds.	8 drops/sec over the last 120 second period.
Incomplete session detected such as TCP SYN attack detected or no data UDP session attack detected (combined)	100 drops/sec over the last 600 seconds.	200 drops/sec over the last 20 second period.
	80 drops/sec over the last 3600 seconds.	160 drops/sec over the last 120 second period.

Table 27-1 Basic Threat Detection Default Settings (continued)

	Trigger Settings	
Packet Drop Reason	**Average Rate**	**Burst Rate**
Denial by ACLs	400 drops/sec over the last 600 seconds.	800 drops/sec over the last 20 second period.
	320 drops/sec over the last 3600 seconds.	640 drops/sec over the last 120 second period.
• Basic firewall checks failed • Packets failed application inspection	400 drops/sec over the last 600 seconds.	1600 drops/sec over the last 20 second period.
	320 drops/sec over the last 3600 seconds.	1280 drops/sec over the last 120 second period.
Interface overload	2000 drops/sec over the last 600 seconds.	8000 drops/sec over the last 20 second period.
	1600 drops/sec over the last 3600 seconds.	6400 drops/sec over the last 120 second period.

Configuring Basic Threat Detection Statistics

This section describes how to configure basic threat detection statistics, including enabling or disabling it and changing the default limits.

Detailed Steps

	Command	Purpose
Step 1	**threat-detection basic-threat** **Example:** `ciscoasa(config)# threat-detection basic-threat`	Enables basic threat detection statistics (if you previously disabled it). Basic threat detection is enabled by default.
Step 2	**threat-detection rate** {**acl-drop** | **bad-packet-drop** | **conn-limit-drop** | **dos-drop** | **fw-drop** | **icmp-drop** | **inspect-drop** | **interface-drop** | **scanning-threat** | **syn-attack**} **rate-interval** *rate_interval* **average-rate** *av_rate* **burst-rate** *burst_rate* **Example:** `ciscoasa(config)# threat-detection rate dos-drop rate-interval 600 average-rate 60 burst-rate 100`	(Optional) Changes the default settings for one or more type of event. For a description of each event type, see the "Information About Basic Threat Detection Statistics" section on page 27-2. When you use this command with the **scanning-threat** keyword, it is also used in the scanning threat detection feature (see the "Configuring Scanning Threat Detection" section). If you do not configure basic threat detection, you can still use this command with the **scanning-threat** keyword to configure the rate limits for scanning threat detection. You can configure up to three different rate intervals for each event type.

Monitoring Basic Threat Detection Statistics

To monitor basic threat detection statistics, perform one of the following tasks:

Command	Purpose
show threat-detection rate [**min-display-rate** *min_display_rate*] [**acl-drop** | **bad-packet-drop** | **conn-limit-drop** | **dos-drop** | **fw-drop** | **icmp-drop** | **inspect-drop** | **interface-drop** | **scanning-threat** | **syn-attack**]	Displays basic threat detection statistics. where the **min-display-rate** *min_display_rate* argument limits the display to statistics that exceed the minimum display rate in events per second. You can set the *min_display_rate* between 0 and 2147483647. For a description of each event type, see the "Information About Basic Threat Detection Statistics" section on page 27-2. The output shows the average rate in events/sec over two fixed time periods: the last 10 minutes and the last 1 hour. It also shows: the current burst rate in events/sec over the last completed burst interval, which is 1/30th of the average rate interval or 10 seconds, whichever is larger; the number of times the rates were exceeded (triggered); and the total number of events over the time periods. The ASA stores the count at the end of each burst period, for a total of 30 completed burst intervals. The unfinished burst interval presently occurring is not included in the average rate. For example, if the average rate interval is 20 minutes, then the burst interval is 20 seconds. If the last burst interval was from 3:00:00 to 3:00:20, and you use the **show** command at 3:00:25, then the last 5 seconds are not included in the output. The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 30) when calculating the total events. In that case, the ASA calculates the total events as the last 29 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.
clear threat-detection rate	Clears basic threat statistics.

Examples

The following is sample output from the **show threat-detection rate** command:

```
ciscoasa# show threat-detection rate

                          Average(eps)    Current(eps) Trigger         Total events
  10-min ACL  drop:                  0               0       0                   16
  1-hour ACL  drop:                  0               0       0                  112
  1-hour SYN attck:                  5               0       2                21438
  10-min  Scanning:                  0               0      29                  193
  1-hour  Scanning:                106               0      10               384776
  1-hour Bad  pkts:                 76               0       2               274690
  10-min  Firewall:                  0               0       3                   22
  1-hour  Firewall:                 76               0       2               274844
  10-min DoS attck:                  0               0       0                    6
  1-hour DoS attck:                  0               0       0                   42
  10-min Interface:                  0               0       0                  204
  1-hour Interface:                 88               0       0               318225
```

Feature History for Basic Threat Detection Statistics

Table 27-2 lists each feature change and the platform release in which it was implemented.

Table 27-2 Feature History for Basic Threat Detection Statistics

Feature Name	Platform Releases	Feature Information
Basic threat detection statistics	8.0(2)	Basic threat detection statistics was introduced. The following commands were introduced: **threat-detection basic-threat**, **threat-detection rate**, **show threat-detection rate**, **clear threat-detection rate**.
Burst rate interval changed to 1/30th of the average rate.	8.2(1)	In earlier releases, the burst rate interval was 1/60th of the average rate. To maximize memory usage, the sampling interval was reduced to 30 times during the average rate.
Improved memory usage	8.3(1)	The memory usage for threat detection was improved.

Configuring Advanced Threat Detection Statistics

You can configure the ASA to collect extensive statistics. This section includes the following topics:

- Information About Advanced Threat Detection Statistics, page 27-6
- Guidelines and Limitations, page 27-6
- Default Settings, page 27-7
- Configuring Advanced Threat Detection Statistics, page 27-7
- Monitoring Advanced Threat Detection Statistics, page 27-9
- Feature History for Advanced Threat Detection Statistics, page 27-14

Information About Advanced Threat Detection Statistics

Advanced threat detection statistics show both allowed and dropped traffic rates for individual objects such as hosts, ports, protocols, or ACLs.

Caution Enabling advanced statistics can affect the ASA performance, depending on the type of statistics enabled. The **threat-detection statistics host** command affects performance in a significant way; if you have a high traffic load, you might consider enabling this type of statistics temporarily. The **threat-detection statistics port** command, however, has modest impact.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature:

Security Context Guidelines

Only TCP Intercept statistics are available in multiple mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

Types of Traffic Monitored

Only through-the-box traffic is monitored; to-the-box traffic is not included in threat detection.

Default Settings

By default, statistics for ACLs are enabled.

Configuring Advanced Threat Detection Statistics

By default, statistics for ACLs are enabled. To enable other statistics, perform the following steps.

Detailed Steps

	Command	Purpose

Step 1	`threat-detection statistics`
	Example:
	`ciscoasa(config)# threat-detection statistics`
		(Optional) Enables *all* statistics.
		To enable only certain statistics, enter this command for each statistic type (shown in this table), and do not also enter the command without any options. You can enter **threat-detection statistics** (without any options) and then customize certain statistics by entering the command with statistics-specific options (for example, **threat-detection statistics host number-of-rate 2**). If you enter **threat-detection statistics** (without any options) and then enter a command for specific statistics, but without any statistic-specific options, then that command has no effect because it is already enabled.
		If you enter the **no** form of this command, it removes all **threat-detection statistics** commands, including the **threat-detection statistics access-list** command, which is enabled by default.

Step 2	`threat-detection statistics access-list`
	Example:
	`ciscoasa(config)# threat-detection statistics access-list`
		(Optional) Enables statistics for ACLs (if they were disabled previously). Statistics for ACLs are enabled by default. ACL statistics are only displayed using the **show threat-detection top access-list** command. This command is enabled by default.

	Command	Purpose
Step 3	**threat-detection statistics host** [**number-of-rate** {**1** | **2** | **3**}] **Example:** `ciscoasa(config)# threat-detection statistics host number-of-rate 2`	(Optional) Enables statistics for hosts. The **number-of-rate** keyword sets the number of rate intervals maintained for host statistics. The default number of rate intervals is **1**, which keeps the memory usage low. To view more rate intervals, set the value to **2** or **3**. For example, if you set the value to **3**, then you view data for the last 1 hour, 8 hours, and 24 hours. If you set this keyword to **1** (the default), then only the shortest rate interval statistics are maintained. If you set the value to **2**, then the two shortest intervals are maintained. The host statistics accumulate for as long as the host is active and in the scanning threat host database. The host is deleted from the database (and the statistics cleared) after 10 minutes of inactivity.
Step 4	**threat-detection statistics port** [**number-of-rate** {**1** | **2** | **3**}] **Example:** `ciscoasa(config)# threat-detection statistics port number-of-rate 2`	(Optional) Enables statistics for TCP and UDP ports. The **number-of-rate** keyword sets the number of rate intervals maintained for port statistics. The default number of rate intervals is **1**, which keeps the memory usage low. To view more rate intervals, set the value to **2** or **3**. For example, if you set the value to **3**, then you view data for the last 1 hour, 8 hours, and 24 hours. If you set this keyword to **1** (the default), then only the shortest rate interval statistics are maintained. If you set the value to **2**, then the two shortest intervals are maintained.

	Command	Purpose
Step 5	**threat-detection statistics protocol** [**number-of-rate** {**1** | **2** | **3**}] **Example:** `ciscoasa(config)# threat-detection statistics protocol number-of-rate 3`	(Optional) Enables statistics for non-TCP/UDP IP protocols. The **number-of-rate** keyword sets the number of rate intervals maintained for protocol statistics. The default number of rate intervals is **1**, which keeps the memory usage low. To view more rate intervals, set the value to **2** or **3**. For example, if you set the value to **3**, then you view data for the last 1 hour, 8 hours, and 24 hours. If you set this keyword to **1** (the default), then only the shortest rate interval statistics are maintained. If you set the value to **2**, then the two shortest intervals are maintained.
Step 6	**threat-detection statistics tcp-intercept** [**rate-interval** *minutes*] [**burst-rate** *attacks_per_sec*] [**average-rate** *attacks_per_sec*] **Example:** `ciscoasa(config)# threat-detection statistics tcp-intercept rate-interval 60 burst-rate 800 average-rate 600`	(Optional) Enables statistics for attacks intercepted by TCP Intercept (see the Chapter 22, "Configuring Connection Settings," to enable TCP Intercept). The **rate-interval** keyword sets the size of the history monitoring window, between 1 and 1440 minutes. The default is 30 minutes. During this interval, the ASA samples the number of attacks 30 times. The **burst-rate** keyword sets the threshold for syslog message generation, between 25 and 2147483647. The default is 400 per second. When the burst rate is exceeded, syslog message 733104 is generated. The **average-rate** keyword sets the average rate threshold for syslog message generation, between 25 and 2147483647. The default is 200 per second. When the average rate is exceeded, syslog message 733105 is generated. **Note** This command is available in multiple context mode.

Monitoring Advanced Threat Detection Statistics

The display output shows the following:

- The average rate in events/sec over fixed time periods.
- The current burst rate in events/sec over the last completed burst interval, which is 1/30th of the average rate interval or 10 seconds, whichever is larger
- The number of times the rates were exceeded (for dropped traffic statistics only)
- The total number of events over the fixed time periods.

The ASA stores the count at the end of each burst period, for a total of 30 completed burst intervals. The unfinished burst interval presently occurring is not included in the average rate. For example, if the average rate interval is 20 minutes, then the burst interval is 20 seconds. If the last burst interval was from 3:00:00 to 3:00:20, and you use the **show** command at 3:00:25, then the last 5 seconds are not included in the output.

The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 30) when calculating the total events. In that case, the ASA calculates the total events as the last 29 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.

To monitor advanced threat detection statistics, perform one of the following tasks:

Command	Purpose
show threat-detection statistics [**min-display-rate** *min_display_rate*] **top** [[**access-list** | **host** | **port-protocol**] [**rate-1** | **rate-2** | **rate-3**] | **tcp-intercept** [**all**] **detail**]]	Displays the top 10 statistics. The **min-display-rate** *min_display_rate* argument limits the display to statistics that exceed the minimum display rate in events per second. You can set the *min_display_rate* between 0 and 2147483647. If you do not enter any options, the top 10 statistics are shown for all categories. To view the top 10 ACEs that match packets, including both permit and deny ACEs, use the **access-list** keyword. Permitted and denied traffic are not differentiated in this display. If you enable basic threat detection using the **threat-detection basic-threat** command, you can track ACL denies using the **show threat-detection rate acl-drop** command. To view only host statistics, use the **host** keyword. **Note**: Due to the threat detction algorithm, an interface used as a combination failover and state link could appear in the top 10 hosts; this is expected behavior, and you can ignore this IP address in the display. To view statistics for ports and protocols, use the **port-protocol** keyword. The **port-protocol** keyword shows statistics for both ports and protocols (both must be enabled for the display), and shows the combined statistics of TCP/UDP port and IP protocol types. TCP (protocol 6) and UDP (protocol 17) are not included in the display for IP protocols; TCP and UDP ports are, however, included in the display for ports. If you only enable statistics for one of these types, port or protocol, then you will only view the enabled statistics. To view TCP Intercept statistics, use the **tcp-intercept** keyword. The display includes the top 10 protected servers under attack. The **all** keyword shows the history data of all the traced servers. The **detail** keyword shows history sampling data. The ASA samples the number of attacks 30 times during the rate interval, so for the default 30 minute period, statistics are collected every 60 seconds. The **rate-1** keyword shows the statistics for the smallest fixed rate intervals available in the display; **rate-2** shows the next largest rate interval; and **rate-3**, if you have three intervals defined, shows the largest rate interval. For example, the display shows statistics for the last 1 hour, 8 hours, and 24 hours. If you set the **rate-1** keyword, the ASA shows only the 1 hour time interval.
show threat-detection statistics [**min-display-rate** *min_display_rate*] **host** [*ip_address* [*mask*]]	Displays statistics for all hosts or for a specific host or subnet.
show threat-detection statistics [**min-display-rate** *min_display_rate*] **port** [*start_port*[-*end_port*]]	Displays statistics for all ports or for a specific port or range of ports.

Command	Purpose
show threat-detection statistics [**min-display-rate** *min_display_rate*] **protocol** [*protocol_number* | **ah** | **eigrp** | **esp** | **gre** | **icmp** | **igmp** | **igrp** | **ip** | **ipinip** | **ipsec** | **nos** | **ospf** | **pcp** | **pim** | **pptp** | **snp** | **tcp** | **udp**]	Displays statistics for all IP protocols or for a specific protocol. The *protocol_number* argument is an integer between 0 and 255.
show threat-detection memory	Displays how much memory is used by advanced threat detection statistics.

Examples

The following is sample output from the **show threat-detection statistics host** command:

```
ciscoasa# show threat-detection statistics host

                          Average(eps)    Current(eps) Trigger         Total events
Host:10.0.0.1: tot-ses:289235 act-ses:22571 fw-drop:0 insp-drop:0 null-ses:21438 bad-acc:0
  1-hour Sent byte:               2938               0       0             10580308
  8-hour Sent byte:                367               0       0             10580308
 24-hour Sent byte:                122               0       0             10580308
  1-hour Sent pkts:                 28               0       0               104043
  8-hour Sent pkts:                  3               0       0               104043
 24-hour Sent pkts:                  1               0       0               104043
  20-min Sent drop:                  9               0       1                10851
  1-hour Sent drop:                  3               0       1                10851
  1-hour Recv byte:               2697               0       0              9712670
  8-hour Recv byte:                337               0       0              9712670
 24-hour Recv byte:                112               0       0              9712670
  1-hour Recv pkts:                 29               0       0               104846
  8-hour Recv pkts:                  3               0       0               104846
 24-hour Recv pkts:                  1               0       0               104846
  20-min Recv drop:                 42               0       3                50567
  1-hour Recv drop:                 14               0       1                50567
Host:10.0.0.0: tot-ses:1 act-ses:0 fw-drop:0 insp-drop:0 null-ses:0 bad-acc:0
  1-hour Sent byte:                  0               0       0                  614
  8-hour Sent byte:                  0               0       0                  614
 24-hour Sent byte:                  0               0       0                  614
  1-hour Sent pkts:                  0               0       0                    6
  8-hour Sent pkts:                  0               0       0                    6
 24-hour Sent pkts:                  0               0       0                    6
  20-min Sent drop:                  0               0       0                    4
  1-hour Sent drop:                  0               0       0                    4
  1-hour Recv byte:                  0               0       0                  706
  8-hour Recv byte:                  0               0       0                  706
 24-hour Recv byte:                  0               0       0                  706
  1-hour Recv pkts:                  0               0       0                    7
```

Table 27-3 shows each field description.

Table 27-3 show threat-detection statistics host Command Fields

Field	Description
Host	Shows the host IP address.
tot-ses	Shows the total number of sessions for this host since it was added to the database.
act-ses	Shows the total number of active sessions that the host is currently involved in.

Table 27-3 show threat-detection statistics host Command Fields (continued)

Field	Description
fw-drop	Shows the number of firewall drops. Firewall drops is a combined rate that includes all firewall-related packet drops tracked in basic threat detection, including ACL denials, bad packets, exceeded connection limits, DoS attack packets, suspicious ICMP packets, TCP SYN attack packets, and no data UDP attack packets. It does not include non-firewall-related drops such as interface overload, packets failed at application inspection, and scanning attack detected.
insp-drop	Shows the number of packets dropped because they failed application inspection.
null-ses	Shows the number of null sessions, which are TCP SYN sessions that did not complete within the 3-second timeout, and UDP sessions that did not have any data sent by its server 3 seconds after the session starts.
bad-acc	Shows the number of bad access attempts to host ports that are in a closed state. When a port is determined to be in a null session (see the null-ses field description), the port state of the host is set to HOST_PORT_CLOSE. Any client accessing the port of the host is immediately classified as a bad access without the need to wait for a timeout.
Average(eps)	Shows the average rate in events/sec over each time period. The ASA stores the count at the end of each burst period, for a total of 30 completed burst intervals. The unfinished burst interval presently occurring is not included in the average rate. For example, if the average rate interval is 20 minutes, then the burst interval is 20 seconds. If the last burst interval was from 3:00:00 to 3:00:20, and you use the **show** command at 3:00:25, then the last 5 seconds are not included in the output. The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 30) when calculating the total events. In that case, the ASA calculates the total events as the last 29 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.
Current(eps)	Shows the current burst rate in events/sec over the last completed burst interval, which is 1/30th of the average rate interval or 10 seconds, whichever is larger. For the example specified in the Average(eps) description, the current rate is the rate from 3:19:30 to 3:20:00
Trigger	Shows the number of times the dropped packet rate limits were exceeded. For valid traffic identified in the sent and received bytes and packets rows, this value is always 0, because there are no rate limits to trigger for valid traffic.
Total events	Shows the total number of events over each rate interval. The unfinished burst interval presently occurring is not included in the total events. The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 30) when calculating the total events. In that case, the ASA calculates the total events as the last 29 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.

Table 27-3 show threat-detection statistics host Command Fields (continued)

Field	Description
20-min, 1-hour, 8-hour, and 24-hour	Shows statistics for these fixed rate intervals.
Sent byte	Shows the number of successful bytes sent from the host.
Sent pkts	Shows the number of successful packets sent from the host.
Sent drop	Shows the number of packets sent from the host that were dropped because they were part of a scanning attack.
Recv byte	Shows the number of successful bytes received by the host.
Recv pkts	Shows the number of successful packets received by the host.
Recv drop	Shows the number of packets received by the host that were dropped because they were part of a scanning attack.

Feature History for Advanced Threat Detection Statistics

Table 27-4 lists each feature change and the platform release in which it was implemented.

Table 27-4 Feature History for Advanced Threat Detection Statistics

Feature Name	Platform Releases	Feature Information
Advanced threat detection statistics	8.0(2)	Advanced threat detection statistics was introduced. The following commands were introduced: **threat-detection statistics**, **show threat-detection statistics**.
TCP Intercept statistics	8.0(4)/8.1(2)	TCP Intercept statistics were introduced. The following commands were modified or introduced: **threat-detection statistics tcp-intercept**, **show threat-detection statistics top tcp-intercept**, **clear threat-detection statistics**.
Customize host statistics rate intervals	8.1(2)	You can now customize the number of rate intervals for which statistics are collected. The default number of rates was changed from 3 to 1. The following command was modified: **threat-detection statistics host number-of-rates**.
Burst rate interval changed to 1/30th of the average rate.	8.2(1)	In earlier releases, the burst rate interval was 1/60th of the average rate. To maximize memory usage, the sampling interval was reduced to 30 times during the average rate.

Table 27-4 Feature History for Advanced Threat Detection Statistics (continued)

Feature Name	Platform Releases	Feature Information
Customize port and protocol statistics rate intervals	8.3(1)	You can now customize the number of rate intervals for which statistics are collected. The default number of rates was changed from 3 to 1. The following commands were modified: **threat-detection statistics port number-of-rates**, **threat-detection statistics protocol number-of-rates**.
Improved memory usage	8.3(1)	The memory usage for threat detection was improved. The following command was introduced: **show threat-detection memory**.

Configuring Scanning Threat Detection

This section includes the following topics:

- Information About Scanning Threat Detection, page 27-15
- Guidelines and Limitations, page 27-16
- Default Settings, page 27-16
- Configuring Scanning Threat Detection, page 27-17
- Monitoring Shunned Hosts, Attackers, and Targets, page 27-17

Information About Scanning Threat Detection

A typical scanning attack consists of a host that tests the accessibility of every IP address in a subnet (by scanning through many hosts in the subnet or sweeping through many ports in a host or subnet). The scanning threat detection feature determines when a host is performing a scan. Unlike IPS scan detection that is based on traffic signatures, the ASA scanning threat detection feature maintains an extensive database that contains host statistics that can be analyzed for scanning activity.

The host database tracks suspicious activity such as connections with no return activity, access of closed service ports, vulnerable TCP behaviors such as non-random IPID, and many more behaviors.

If the scanning threat rate is exceeded, then the ASA sends a syslog message (733101), and optionally shuns the attacker. The ASA tracks two types of rates: the average event rate over an interval, and the burst event rate over a shorter burst interval. The burst event rate is 1/30th of the average rate interval or 10 seconds, whichever is higher. For each event detected that is considered to be part of a scanning attack, the ASA checks the average and burst rate limits. If either rate is exceeded for traffic sent from a host, then that host is considered to be an attacker. If either rate is exceeded for traffic received by a host, then that host is considered to be a target.

Caution The scanning threat detection feature can affect the ASA performance and memory significantly while it creates and gathers host- and subnet-based data structure and information.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature:

Security Context Guidelines

Supported in single mode only. Multiple mode is not supported.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

Types of Traffic Monitored

- Only through-the-box traffic is monitored; to-the-box traffic is not included in threat detection.
- Traffic that is denied by an ACL does not trigger scanning threat detection; only traffic that is allowed through the ASA and that creates a flow is affected by scanning threat detection.

Default Settings

Table 27-5 lists the default rate limits for scanning threat detection.

Table 27-5 Default Rate Limits for Scanning Threat Detection

Average Rate	Burst Rate
5 drops/sec over the last 600 seconds.	10 drops/sec over the last 20 second period.
5 drops/sec over the last 3600 seconds.	10 drops/sec over the last 120 second period.

The burst rate is calculated as the average rate every *N* seconds, where *N* is the burst rate interval. The burst rate interval is 1/30th of the rate interval or 10 seconds, whichever is larger.

Configuring Scanning Threat Detection

Detailed Steps

	Command	Purpose
Step 1	**threat-detection scanning-threat** [**shun** [**except** {**ip-address** *ip_address mask* | **object-group** *network_object_group_id*}]] **Example:** `ciscoasa(config)# threat-detection scanning-threat shun except ip-address 10.1.1.0 255.255.255.0`	Enables scanning threat detection. By default, the system log message 733101 is generated when a host is identified as an attacker. Enter this command multiple times to identify multiple IP addresses or network object groups to exempt from shunning.
Step 2	**threat-detection scanning-threat shun duration** *seconds* **Example:** `ciscoasa(config)# threat-detection scanning-threat shun duration 2000`	(Optional) Sets the duration of the shun for attacking hosts.

Step 3	**threat-detection rate scanning-threat rate-interval** *rate_interval* **average-rate** *av_rate* **burst-rate** *burst_rate*
	Example:
	`ciscoasa(config)# threat-detection rate scanning-threat rate-interval 1200 average-rate 10 burst-rate 20`
	`ciscoasa(config)# threat-detection rate scanning-threat rate-interval 2400 average-rate 10 burst-rate 20`
		(Optional) Changes the default event limit for when the ASA identifies a host as an attacker or as a target. If you already configured this command as part of the basic threat detection configuration (see the "Configuring Basic Threat Detection Statistics" section on page 27-2), then those settings are shared with the scanning threat detection feature; you cannot configure separate rates for basic and scanning threat detection. If you do not set the rates using this command, the default values are used for both the scanning threat detection feature and the basic threat detection feature. You can configure up to three different rate intervals, by entering separate commands.

Monitoring Shunned Hosts, Attackers, and Targets

To monitor shunned hosts and attackers and targets, perform one of the following tasks:

Command	Purpose
show threat-detection shun	Displays the hosts that are currently shunned.

Command	Purpose
clear threat-detection shun [*ip_address* [*mask*]]	Releases a host from being shunned. If you do not specify an IP address, all hosts are cleared from the shun list.
show threat-detection scanning-threat [**attacker** | **target**]	Displays hosts that the ASA decides are attackers (including hosts on the shun list), and displays the hosts that are the target of an attack. If you do not enter an option, both attackers and target hosts are displayed.

Examples

The following is sample output from the **show threat-detection shun** command:

```
ciscoasa# show threat-detection shun
Shunned Host List:
10.1.1.6
192.168.6.7
```

To release the host at 10.1.1.6, enter the following command:

```
ciscoasa# clear threat-detection shun 10.1.1.6
```

The following is sample output from the **show threat-detection scanning-threat attacker** command:

```
ciscoasa# show threat-detection scanning-threat attacker
10.1.2.3
10.8.3.6
209.165.200.225
```

Feature History for Scanning Threat Detection

Table 27-6 lists each feature change and the platform release in which it was implemented.

Table 27-6 Feature History for Scanning Threat Detection

Feature Name	Platform Releases	Feature Information
Scanning threat detection	8.0(2)	Scanning threat detection was introduced. The following commands were introduced: **threat-detection scanning-threat**, **threat-detection rate scanning-threat**, **show threat-detection scanning-threat**, **show threat-detection shun**, **clear threat-detection shun**.
Shun duration	8.0(4)/8.1(2)	You can now set the shun duration, The following command was introduced: **threat-detection scanning-threat shun duration**.

Table 27-6 Feature History for Scanning Threat Detection (continued)

Feature Name	Platform Releases	Feature Information
Burst rate interval changed to 1/30th of the average rate.	8.2(1)	In earlier releases, the burst rate interval was 1/60th of the average rate. To maximize memory usage, the sampling interval was reduced to 30 times during the average rate.
Improved memory usage	8.3(1)	The memory usage for threat detection was improved.

Configuration Examples for Threat Detection

The following example configures basic threat detection statistics, and changes the DoS attack rate settings. All advanced threat detection statistics are enabled, with the host statistics number of rate intervals lowered to 2. The TCP Intercept rate interval is also customized. Scanning threat detection is enabled with automatic shunning for all addresses except 10.1.1.0/24. The scanning threat rate intervals are customized.

```
threat-detection basic-threat
threat-detection rate dos-drop rate-interval 600 average-rate 60 burst-rate 100
threat-detection statistics
threat-detection statistics host number-of-rate 2
threat-detection statistics tcp-intercept rate-interval 60 burst-rate 800 average-rate 600
threat-detection scanning-threat shun except ip-address 10.1.1.0 255.255.255.0
threat-detection rate scanning-threat rate-interval 1200 average-rate 10 burst-rate 20
threat-detection rate scanning-threat rate-interval 2400 average-rate 10 burst-rate 20
```

CHAPTER 28

Using Protection Tools

This chapter describes some of the many tools available to protect your network and includes the following sections:

Preventing IP Spoofing

This section lets you enable Unicast Reverse Path Forwarding on an interface. Unicast RPF guards against IP spoofing (a packet uses an incorrect source IP address to obscure its true source) by ensuring that all packets have a source IP address that matches the correct source interface according to the routing table.

Normally, the ASA only looks at the destination address when determining where to forward the packet. Unicast RPF instructs the ASA to also look at the source address; this is why it is called Reverse Path Forwarding. For any traffic that you want to allow through the ASA, the ASA routing table must include a route back to the source address. See RFC 2267 for more information.

For outside traffic, for example, the ASA can use the default route to satisfy the Unicast RPF protection. If traffic enters from an outside interface, and the source address is not known to the routing table, the ASA uses the default route to correctly identify the outside interface as the source interface.

If traffic enters the outside interface from an address that is known to the routing table, but is associated with the inside interface, then the ASA drops the packet. Similarly, if traffic enters the inside interface from an unknown source address, the ASA drops the packet because the matching route (the default route) indicates the outside interface.

Unicast RPF is implemented as follows:

- ICMP packets have no session, so each packet is checked.
- UDP and TCP have sessions, so the initial packet requires a reverse route lookup. Subsequent packets arriving during the session are checked using an existing state maintained as part of the session. Non-initial packets are checked to ensure they arrived on the same interface used by the initial packet.

To enable Unicast RPF, enter the following command:

```
ciscoasa(config)# ip verify reverse-path interface interface_name
```

Configuring the Fragment Size

By default, the ASA allows up to 24 fragments per IP packet, and up to 200 fragments awaiting reassembly. You might need to let fragments on your network if you have an application that routinely fragments packets, such as NFS over UDP. However, if you do not have an application that fragments traffic, we recommend that you do not allow fragments through the ASA. Fragmented packets are often used as DoS attacks.

To set disallow fragments, enter the following command:

```
ciscoasa(config)# fragment chain 1 [interface_name]
```

Enter an interface name if you want to prevent fragmentation on a specific interface. By default, this command applies to all interfaces.

Blocking Unwanted Connections

If you know that a host is attempting to attack your network (for example, system log messages show an attack), then you can block (or shun) connections based on the source IP address. All existing connections and new connections are blocked until you remove the shun.

Note If you have an IPS that monitors traffic, such as an AIP SSM, then the IPS can shun connections automatically.

To shun a connection manually, perform the following steps:

Step 1 If necessary, view information about the connection by entering the following command:

```
ciscoasa# show conn
```

The ASA shows information about each connection, such as the following:

```
TCP out 64.101.68.161:4300 in 10.86.194.60:23 idle 0:00:00 bytes 1297 flags UIO
```

Step 2 To shun connections from the source IP address, enter the following command:

```
ciscoasa(config)# shun src_ip [dst_ip src_port dest_port [protocol]] [vlan vlan_id]
```

If you enter only the source IP address, then all *future* connections are shunned; existing connections remain active.

To drop an existing connection, as well as blocking future connections from the source IP address, enter the destination IP address, source and destination ports, and the protocol. By default, the protocol is 0 for IP. Note that specifying the additional parameters is a convenient way to also drop a specific current connection; the shun, however, remains in place for all future connections from the source IP address, regardless of destination parameters.

For multiple context mode, you can enter this command in the admin context, and by specifying a VLAN ID that is assigned to an interface in other contexts, you can shun the connection in other contexts.

Step 3 To remove the shun, enter the following command:

```
ciscoasa(config)# no shun src_ip [vlan vlan_id]
```

Configuring IP Audit for Basic IPS Support

The IP audit feature provides basic IPS support for the ASA that does not have an AIP SSM. It supports a basic list of signatures, and you can configure the ASA to perform one or more actions on traffic that matches a signature.

This section includes the following topics:

- Configuring IP Audit, page 28-3
- IP Audit Signature List, page 28-4

Configuring IP Audit

To enable IP audit, perform the following steps:

Step 1 To define an IP audit policy for informational signatures, enter the following command:

```
ciscoasa(config)# ip audit name name info [action [alarm] [drop] [reset]]
```

Where **alarm** generates a system message showing that a packet matched a signature, **drop** drops the packet, and **reset** drops the packet and closes the connection. If you do not define an action, then the default action is to generate an alarm.

Step 2 To define an IP audit policy for attack signatures, enter the following command:

```
ciscoasa(config)# ip audit name name attack [action [alarm] [drop] [reset]]
```

Where **alarm** generates a system message showing that a packet matched a signature, **drop** drops the packet, and **reset** drops the packet and closes the connection. If you do not define an action, then the default action is to generate an alarm.

Step 3 To assign the policy to an interface, enter the following command:

```
ip audit interface interface_name policy_name
```

Step 4 To disable signatures, or for more information about signatures, see the **ip audit signature** command in the command reference.

IP Audit Signature List

Table 28-1 lists supported signatures and system message numbers.

Table 28-1 Signature IDs and System Message Numbers

Signature ID	Message Number	Signature Title	Signature Type	Description
1000	400000	IP options-Bad Option List	Informational	Triggers on receipt of an IP datagram where the list of IP options in the IP datagram header is incomplete or malformed. The IP options list contains one or more options that perform various network management or debugging tasks.
1001	400001	IP options-Record Packet Route	Informational	Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 7 (Record Packet Route).
1002	400002	IP options-Timestamp	Informational	Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 4 (Timestamp).
1003	400003	IP options-Security	Informational	Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 2 (Security options).
1004	400004	IP options-Loose Source Route	Informational	Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 3 (Loose Source Route).
1005	400005	IP options-SATNET ID	Informational	Triggers on receipt of an IP datagram where the IP option list for the datagram includes option 8 (SATNET stream identifier).
1006	400006	IP options-Strict Source Route	Informational	Triggers on receipt of an IP datagram in which the IP option list for the datagram includes option 9(Strict Source Routing).
1100	400007	IP Fragment Attack	Attack	Triggers when any IP datagram is received with an offset value less than 5 but greater than 0 indicated in the offset field.
1102	400008	IP Impossible Packet	Attack	Triggers when an IP packet arrives with source equal to destination address. This signature will catch the so-called Land Attack.

Table 28-1 Signature IDs and System Message Numbers (continued)

Signature ID	Message Number	Signature Title	Signature Type	Description
1103	400009	IP Overlapping Fragments (Teardrop)	Attack	Triggers when two fragments contained within the same IP datagram have offsets that indicate that they share positioning within the datagram. This could mean that fragment A is being completely overwritten by fragment B, or that fragment A is partially being overwritten by fragment B. Some operating systems do not properly handle fragments that overlap in this manner and may throw exceptions or behave in other undesirable ways upon receipt of overlapping fragments, which is how the Teardrop attack works to create a DoS.
2000	400010	ICMP Echo Reply	Informational	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 0 (Echo Reply).
2001	400011	ICMP Host Unreachable	Informational	Triggers when an IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 3 (Host Unreachable).
2002	400012	ICMP Source Quench	Informational	Triggers when an IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 4 (Source Quench).
2003	400013	ICMP Redirect	Informational	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 5 (Redirect).
2004	400014	ICMP Echo Request	Informational	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 8 (Echo Request).
2005	400015	ICMP Time Exceeded for a Datagram	Informational	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 11(Time Exceeded for a Datagram).
2006	400016	ICMP Parameter Problem on Datagram	Informational	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 12 (Parameter Problem on Datagram).
2007	400017	ICMP Timestamp Request	Informational	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 13 (Timestamp Request).

Table 28-1 Signature IDs and System Message Numbers (continued)

Signature ID	Message Number	Signature Title	Signature Type	Description
2008	400018	ICMP Timestamp Reply	Informational	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 14 (Timestamp Reply).
2009	400019	ICMP Information Request	Informational	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 15 (Information Request).
2010	400020	ICMP Information Reply	Informational	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 16 (ICMP Information Reply).
2011	400021	ICMP Address Mask Request	Informational	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 17 (Address Mask Request).
2012	400022	ICMP Address Mask Reply	Informational	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and the type field in the ICMP header set to 18 (Address Mask Reply).
2150	400023	Fragmented ICMP Traffic	Attack	Triggers when a IP datagram is received with the protocol field of the IP header set to 1 (ICMP) and either the more fragments flag is set to 1 (ICMP) or there is an offset indicated in the offset field.
2151	400024	Large ICMP Traffic	Attack	Triggers when a IP datagram is received with the protocol field of the IP header set to 1(ICMP) and the IP length > 1024.
2154	400025	Ping of Death Attack	Attack	Triggers when a IP datagram is received with the protocol field of the IP header set to 1(ICMP), the Last Fragment bit is set, and (IP offset * 8) + (IP data length) > 65535 that is to say, the IP offset (which represents the starting position of this fragment in the original packet, and which is in 8 byte units) plus the rest of the packet is greater than the maximum size for an IP packet.
3040	400026	TCP NULL flags	Attack	Triggers when a single TCP packet with none of the SYN, FIN, ACK, or RST flags set has been sent to a specific host.
3041	400027	TCP SYN+FIN flags	Attack	Triggers when a single TCP packet with the SYN and FIN flags are set and is sent to a specific host.

Table 28-1 Signature IDs and System Message Numbers (continued)

Signature ID	Message Number	Signature Title	Signature Type	Description
3042	400028	TCP FIN only flags	Attack	Triggers when a single orphaned TCP FIN packet is sent to a privileged port (having port number less than 1024) on a specific host.
3153	400029	FTP Improper Address Specified	Informational	Triggers if a port command is issued with an address that is not the same as the requesting host.
3154	400030	FTP Improper Port Specified	Informational	Triggers if a port command is issued with a data port specified that is <1024 or >65535.
4050	400031	UDP Bomb attack	Attack	Triggers when the UDP length specified is less than the IP length specified. This malformed packet type is associated with a denial of service attempt.
4051	400032	UDP Snork attack	Attack	Triggers when a UDP packet with a source port of either 135, 7, or 19 and a destination port of 135 is detected.
4052	400033	UDP Chargen DoS attack	Attack	This signature triggers when a UDP packet is detected with a source port of 7 and a destination port of 19.
6050	400034	DNS HINFO Request	Informational	Triggers on an attempt to access HINFO records from a DNS server.
6051	400035	DNS Zone Transfer	Informational	Triggers on normal DNS zone transfers, in which the source port is 53.
6052	400036	DNS Zone Transfer from High Port	Informational	Triggers on an illegitimate DNS zone transfer, in which the source port is not equal to 53.
6053	400037	DNS Request for All Records	Informational	Triggers on a DNS request for all records.
6100	400038	RPC Port Registration	Informational	Triggers when attempts are made to register new RPC services on a target host.
6101	400039	RPC Port Unregistration	Informational	Triggers when attempts are made to unregister existing RPC services on a target host.
6102	400040	RPC Dump	Informational	Triggers when an RPC dump request is issued to a target host.
6103	400041	Proxied RPC Request	Attack	Triggers when a proxied RPC request is sent to the portmapper of a target host.
6150	400042	ypserv (YP server daemon) Portmap Request	Informational	Triggers when a request is made to the portmapper for the YP server daemon (ypserv) port.
6151	400043	ypbind (YP bind daemon) Portmap Request	Informational	Triggers when a request is made to the portmapper for the YP bind daemon (ypbind) port.

Table 28-1 Signature IDs and System Message Numbers (continued)

Signature ID	Message Number	Signature Title	Signature Type	Description
6152	400044	yppasswdd (YP password daemon) Portmap Request	Informational	Triggers when a request is made to the portmapper for the YP password daemon (yppasswdd) port.
6153	400045	ypupdated (YP update daemon) Portmap Request	Informational	Triggers when a request is made to the portmapper for the YP update daemon (ypupdated) port.
6154	400046	ypxfrd (YP transfer daemon) Portmap Request	Informational	Triggers when a request is made to the portmapper for the YP transfer daemon (ypxfrd) port.
6155	400047	mountd (mount daemon) Portmap Request	Informational	Triggers when a request is made to the portmapper for the mount daemon (mountd) port.
6175	400048	rexd (remote execution daemon) Portmap Request	Informational	Triggers when a request is made to the portmapper for the remote execution daemon (rexd) port.
6180	400049	rexd (remote execution daemon) Attempt	Informational	Triggers when a call to the rexd program is made. The remote execution daemon is the server responsible for remote program execution. This may be indicative of an attempt to gain unauthorized access to system resources.
6190	400050	statd Buffer Overflow	Attack	Triggers when a large statd request is sent. This could be an attempt to overflow a buffer and gain access to system resources.

Configuring Filtering Services

This chapter describes how to use filtering services to provide greater control over traffic passing through the ASA and includes the following sections:

Information About Web Traffic Filtering

You can use web traffic filtering in two distinct ways:

- Filtering ActiveX objects or Java applets
- Filtering with an external filtering server

Instead of blocking access altogether, you can remove specific undesirable objects from web traffic, such as ActiveX objects or Java applets, that may pose a security threat in certain situations.

You can use web traffic filtering to direct specific traffic to an external filtering server, such an Secure Computing SmartFilter (formerly N2H2) or the Websense filtering server. You can enable long URL, HTTPS, and FTP filtering using either Websense or Secure Computing SmartFilter for web traffic filtering. Filtering servers can block traffic to specific sites or types of sites, as specified by the security policy.

Note URL caching will only work if the version of the URL server software from the URL server vendor supports it.

Because web traffic filtering is CPU-intensive, using an external filtering server ensures that the throughput of other traffic is not affected. However, depending on the speed of your network and the capacity of your web traffic filtering server, the time required for the initial connection may be noticeably slower when filtering traffic with an external filtering server.

Configuring ActiveX Filtering

This section includes the following topics:

Information About ActiveX Filtering

ActiveX objects may pose security risks because they can contain code intended to attack hosts and servers on a protected network. You can disable ActiveX objects with ActiveX filtering.

ActiveX controls, formerly known as OLE or OCX controls, are components that you can insert in a web page or another application. These controls include custom forms, calendars, or any of the extensive third-party forms for gathering or displaying information. As a technology, ActiveX creates many potential problems for network clients including causing workstations to fail, introducing network security problems, or being used to attack servers.

The **filter activex** command blocks the HTML **object** commands by commenting them out within the HTML web page. ActiveX filtering of HTML files is performed by selectively replacing the <APPLET> and </APPLET>, and <OBJECT CLASSID> and </OBJECT> tags with comments. Filtering of nested tags is supported by converting top-level tags to comments.

Caution The **filter activex** command also blocks any Java applets, image files, or multimedia objects that are embedded in object tags.

If the <object> or </object> HTML tags split across network packets or if the code in the tags is longer than the number of bytes in the MTU, the ASA cannot block the tag.

ActiveX blocking does not occur when users access an IP address referenced by the **alias** command or for clientless SSL VPN traffic.

Licensing Requirements for ActiveX Filtering

The following table shows the licensing requirements for this feature:

Model	License Requirement
All models	Base License.

Guidelines and Limitations for ActiveX Filtering

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Does not support IPv6.

Configuring ActiveX Filtering

To remove ActiveX objects in HTTP traffic that is passing through the ASA, enter the following command:

Command	Purpose
filter activex *port[-port] local_ip local_mask foreign_ip foreign_mask* **Example:** `ciscoasa# filter activex 80 0 0 0 0`	Removes ActiveX objects. To use this command, replace *port[-port]* with the TCP port to which filtering is applied. Typically, this is port 80, but other values are accepted. The **http** or **url** literal can be used for port 80. You can specify a range of ports by using a hyphen between the starting port number and the ending port number. The local IP address and mask identify one or more internal hosts that are the source of the traffic to be filtered. The foreign address and mask specify the external destination of the traffic to be filtered.

Configuration Examples for ActiveX Filtering

You can set either address to **0.0.0.0** (or in shortened form, **0**) to specify all hosts. You can use **0.0.0.0** for either mask (or in shortened form, **0**) to specify all masks. This command specifies that the ActiveX object blocking applies to HTTP traffic on port 80 from any local host and for connections to any foreign host.

The following example shows how to configure ActiveX filtering to block all outbound connections:

```
ciscoasa(config)# filter activex 80 0 0 0 0
```

The following example shows how to remove ActiveX filtering:

```
ciscoasa(config)# no filter activex 80 0 0 0 0
```

Feature History for ActiveX Filtering

Table 29-1 lists the release history for ActiveX Filtering. ASDM is backwards-compatible with multiple platform releases, so the specific ASDM release in which support was added is not listed.

Table 29-1 Feature History for ActiveX Filtering

Feature Name	Platform Releases	Feature Information
ActiveX filtering	7.0(1)	Filters specific undesirable objects from HTTP traffic, such as ActiveX objects, which may pose a security threat in certain situations.

Configuring Java Applet Filtering

This section includes the following topics:

Information About Java Applet Filtering

Java applets may pose security risks because they can contain code intended to attack hosts and servers on a protected network. You can remove Java applets with the **filter java** command.

Note Use the **filter activex** command to remove Java applets that are embedded in <object> tags.

The **filter java** command filters out Java applets that return to the ASA from an outbound connection. You still receive the HTML page, but the web page source for the applet is commented out so that the applet cannot execute. The **filter java** command does not filter clientless SSL VPN traffic.

Licensing Requirements for Java Applet Filtering

The following table shows the licensing requirements for Java applet filtering:

Table 29-2 Licensing Requirements

Model	License Requirement
All models	Base License.

Guidelines and Limitations for Java Applet Filtering

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Does not support IPv6.

Configuring Java Applet Filtering

To apply filtering to remove Java applets from HTTP traffic passing through the ASA, enter the following command:

Command	Purpose
filter java *port[-port] local_ip local_mask foreign_ip foreign_mask* **Example:** `ciscoasa# filter java 80 0 0 0 0`	Removes Java applets in HTTP traffic passing through the ASA. To use this command, replace *port[-port]* with the TCP port to which filtering is applied. Typically, this is port 80, but other values are accepted. The **http** or **url** literal can be used for port 80. You can specify a range of ports by using a hyphen between the starting port number and the ending port number. The local IP address and mask identify one or more internal hosts that are the source of the traffic to be filtered. The foreign address and mask specify the external destination of the traffic to be filtered. You can set either address to **0.0.0.0** (or in shortened form, **0**) to specify all hosts. You can use **0.0.0.0** for either mask (or in shortened form, **0**) to specify all hosts. You can set either address to **0.0.0.0** (or in shortened form, **0**) to specify all hosts. You can use **0.0.0.0** for either mask (or in shortened form, **0**) to specify all hosts.

Configuration Examples for Java Applet Filtering

The following example specifies that Java applets are blocked on all outbound connections:

```
ciscoasa(config)# filter java 80 0 0 0 0
```

This command specifies that the Java applet blocking applies to web traffic on port 80 from any local host and for connections to any foreign host.

The following example blocks downloading of Java applets to a host on a protected network:

```
ciscoasa(config)# filter java http 192.168.3.3 255.255.255.255 0 0
```

This command prevents host 192.168.3.3 from downloading Java applets.

The following example removes the configuration for downloading Java applets to a host on a protected network:

```
ciscoasa(config)# no filter java http 192.168.3.3 255.255.255.255 0 0
```

This command allows host 192.168.3.3 to download Java applets.

Feature History for Java Applet Filtering

Table 29-1 lists the release history for Java applet filtering. ASDM is backwards-compatible with multiple platform releases, so the specific ASDM release in which support was added is not listed.

Table 29-3 Feature History for Java Applet Filtering

Feature Name	Platform Releases	Feature Information
Java applet filtering	7.0(1)	Filters specific undesirable objects from HTTP traffic, such as Java applets, which may pose a security threat in certain situations.

Filtering URLs and FTP Requests with an External Server

This section describes how to filter URLs and FTP requests with an external server and includes the following topics:

- Information About URL Filtering, page 29-6
- Licensing Requirements for URL Filtering, page 29-7
- Guidelines and Limitations for URL Filtering, page 29-7
- Identifying the Filtering Server, page 29-8
- Configuring Additional URL Filtering Settings, page 29-10
- Feature History for URL Filtering, page 29-17

Information About URL Filtering

You can apply filtering to connection requests originating from a more secure network to a less secure network. Although you can use ACLs to prevent outbound access to specific content servers, managing usage this way is difficult because of the size and dynamic nature of the Internet. You can simplify configuration and improve ASA performance by using a separate server running one of the following Internet filtering products:

- Websense Enterprise for filtering HTTP, HTTPS, and FTP.
- McAfee SmartFilter (formerly N2H2) for filtering HTTP, HTTPS, FTP, and long URL filtering.

  In long URLs, the URL in the Referer field might contain a “host:” text string, which could cause the HTTP GET header to be incorrectly parsed as containing the HTTP Host parameter. The ASA, however, correctly parses the Referer field even when it contains a “host:” text string and forwards the header to the McAfee SmartFilter server with the correct Referer URL.

Note URL caching will only work if the version of the URL server software from the URL server vendor supports it.

Although ASA performance is less affected when using an external server, you might notice longer access times to websites or FTP servers when the filtering server is remote from the ASA.

When filtering is enabled and a request for content is directed through the ASA, the request is sent to the content server and to the filtering server at the same time. If the filtering server allows the connection, the ASA forwards the response from the content server to the originating client. If the filtering server denies the connection, the ASA drops the response and sends a message or return code indicating that the connection was not successful.

If user authentication is enabled on the ASA, then the ASA also sends the username to the filtering server. The filtering server can use user-specific filtering settings or provide enhanced reporting about usage.

Licensing Requirements for URL Filtering

The following table shows the licensing requirements for URL filtering:

Table 29-4 Licensing Requirements

Model	License Requirement
All models	Base License.

Guidelines and Limitations for URL Filtering

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

IPv6 Guidelines

Does not support IPv6.

Identifying the Filtering Server

You can identify up to four filtering servers per context. The ASA uses the servers in order until a server responds. In single mode, a maximum of 16 of the same type of filtering servers are allowed. You can only configure a single type of server (Websense or Secure Computing SmartFilter) in your configuration.

Note You must add the filtering server before you can configure filtering for HTTP or HTTPS with the **filter** command. If you remove the filtering servers from the configuration, then all **filter** commands are also removed.

To specify the external filtering server, enter the following command:

	Command	Purpose
	Choose from the following options:	

<table>
<tr><th></th><th>Command</th><th>Purpose</th></tr>
<tr><td></td><td>For Websense:
hostname(config)# **url-server** (*if_name*) **host** *local_ip* [**timeout** *seconds*] [**protocol TCP** | **UDP version [1|4]** [**connections** *num_conns*]]
Example:
ciscoasa(config)# url-server (perimeter) host 10.0.1.1 protocol TCP version 4</td><td>Identifies the address of the filtering server. *if_name* is the name of the ASA interface connected to the filtering server (the default is inside). For the **vendor** {*secure-computing* | *n2h2*} option, use *secure-computing* as the vendor string; however, *n2h2* is acceptable for backward compatibility. When the configuration entries are generated, *secure-computing* is saved as the vendor string. The **host** *local_ip* option is the IP address of the URL filtering server. The **port** *number* option is the Secure Computing SmartFilter server port number of the filtering server; the ASA also listens for UDP replies on this port.
Note The default port is 4005, which is used by the Secure Computing SmartFilter server to communicate to the ASA via TCP or UDP. For information about changing the default port, see the *Filtering by N2H2 Administrator's Guide*.
The **timeout** *seconds* option is the number of seconds that the ASA should keep trying to connect to the filtering server. The **connections** *number* option is the number of tries to make a connection between the host and server.
The example identifies a Websense filtering server with the IP address 10.0.1.1 on a perimeter interface of the ASA.Version 4, which is enabled in this example, is recommended by Websense because it supports caching.</td></tr>
<tr><td></td><td>For Secure Computing SmartFilter (formerly N2H2):
hostname(config)# **url-server** (*if_name*) **vendor** {*secure-computing* | *n2h2*} **host** *local_ip* [**port** *number*] [**timeout** *seconds*] [**protocol** {**TCP** [*connections number*]} | **UDP**]
Example:
ciscoasa(config)# url-server (perimeter) vendor n2h2 host 10.0.1.1
ciscoasa(config)# url-server (perimeter) vendor n2h2 host 10.0.1.2</td><td>The example identifies redundant Secure Computing SmartFilter servers that are both on a perimeter interface of the ASA.</td></tr>
</table>

Configuring Additional URL Filtering Settings

After you have accessed a website, the filtering server can allow the ASA to cache the server address for a certain period of time, as long as each website hosted at the address is in a category that is permitted at all times. When you access the server again, or if another user accesses the server, the ASA does not need to consult the filtering server again to obtain the server address.

Note Requests for cached IP addresses are not passed to the filtering server and are not logged. As a result, this activity does not appear in any reports.

This section describes how to configure additional URL filtering settings and includes the following topics:

- Buffering the Content Server Response, page 29-10
- Caching Server Addresses, page 29-11
- Filtering HTTP URLs, page 29-11
- Filtering HTTPS URLs, page 29-13
- Filtering FTP Requests, page 29-14

Buffering the Content Server Response

When you issue a request to connect to a content server, the ASA sends the request to the content server and to the filtering server at the same time. If the filtering server does not respond before the content server, the server response is dropped. This behavior delays the web server response for the web client, because the web client must reissue the request.

By enabling the HTTP response buffer, replies from web content servers are buffered, and the responses are forwarded to the requesting client if the filtering server allows the connection. This behavior prevents the delay that might otherwise occur.

To configure buffering for responses to HTTP or FTP requests, enter the following command:

	Command	Purpose
Step 1	**url-block block** *block-buffer-limit* **Example:** `ciscoasa# url-block 3000`	Enables buffering of responses for HTTP or FTP requests that are pending a response from the filtering server. Replaces *block-buffer* with the maximum number of HTTP responses that can be buffered while awaiting responses from the URL server. **Note** Buffering of URLs longer than 3072 bytes is not supported.
Step 2	**url-block mempool-size** *memory-pool-size* **Example:** `ciscoasa# url-block mempool-size 5000`	Configures the maximum memory available for buffering pending URLs (and for buffering long URLs). Replaces *memory-pool-size* with a value from 2 to 10240 for a maximum memory allocation of 2 KB to 10 MB.

Caching Server Addresses

After you access a website, the filtering server can allow the ASA to cache the server address for a certain period of time, as long as each website hosted at the address is in a category that is permitted at all times. When you access the server again, or if another user accesses the server, the ASA does not need to consult the filtering server again.

Note Requests for cached IP addresses are not passed to the filtering server and are not logged. As a result, this activity does not appear in any reports. You can accumulate Websense run logs before using the **url-cache** command.

To improve throughput, enter the following command:

Command	Purpose
url-cache dst | src_dst *size* **Example:** `ciscoasa## url-cache src_dst 100`	Replaces *size* with a value for the cache size within the range from 1 to 128 (KB). Uses the **dst** keyword to cache entries based on the URL destination address. Choose this option if all users share the same URL filtering policy on the Websense server. Uses the **src_dst** keyword to cache entries based on both the source address initiating the URL request as well as the URL destination address. Choose this option if users do not share the same URL filtering policy on the Websense server.

Filtering HTTP URLs

This section describes how to configure HTTP filtering with an external filtering server and includes the following topics:

- Enabling HTTP Filtering, page 29-12
- Enabling Filtering of Long HTTP URLs, page 29-12
- Truncating Long HTTP URLs, page 29-13
- Exempting Traffic from Filtering, page 29-13

Enabling HTTP Filtering

You must identify and enable the URL filtering server before enabling HTTP filtering. When the filtering server approves an HTTP connection request, the ASA allows the reply from the web server to reach the originating client. If the filtering server denies the request, the ASA redirects you to a block page, indicating that access was denied.

To enable HTTP filtering, enter the following command:

Command	Purpose
filter url [**http** | *port*[-*port*] *local_ip* *local_mask* *foreign_ip* *foreign_mask*] [**allow**] [**proxy-block**] **Example:** `ciscoasa# filter url http 80 allow proxy-block`	Replaces *port*[-*port*] with one or more port numbers if a different port than the default port for HTTP (80) is used. Replaces *local_ip* and *local_mask* with the IP address and subnet mask of a user or subnetwork making requests. Replaces *foreign_ip* and *foreign_mask* with the IP address and subnet mask of a server or subnetwork responding to requests. The **allow** option causes the ASA to forward HTTP traffic without filtering when the primary filtering server is unavailable. Use the **proxy-block** command to drop all requests to proxy servers.

Enabling Filtering of Long HTTP URLs

By default, the ASA considers an HTTP URL to be a long URL if it is greater than 1159 characters. You can increase the maximum length allowed.

To configure the maximum size of a single URL, enter the following command:

Command	Purpose
url-block url-size *long-url-size* **Example:** `ciscoasa# url-block url-size 3`	Replaces the *long-url-size* with the maximum size in KB for each long URL being buffered. For Websense servers, this is a value from 2 to 4 for a maximum URL size from 2 KB to 4 KB; for Secure Computing SmartFilter servers, this is a value between 2 and 3 for a maximum URL size from 2 KB to 3 KB. The default value is 2.

Truncating Long HTTP URLs

By default, if a URL exceeds the maximum permitted size, then it is dropped. To avoid this occurrence, truncate a long URL by entering the following command:

Command	Purpose

`filter url` `[`**`longurl-truncate`** `|` **`longurl-deny`** `|` **`cgi-truncate`**`]`
Example:
`ciscoasa# filter url longurl-truncate`
	The **longurl-truncate** option causes the ASA to send only the hostname or IP address portion of the URL for evaluation to the filtering server when the URL is longer than the maximum length permitted. Use the **longurl-deny** option to deny outbound URL traffic if the URL is longer than the maximum permitted.
	Use the **cgi-truncate** option to truncate CGI URLs to include only the CGI script location and the script name without any parameters. Many long HTTP requests are CGI requests. If the parameters list is very long, waiting and sending the complete CGI request, including the parameter list, can use up memory resources and affect ASA performance.

Exempting Traffic from Filtering

To exempt traffic from filtering, enter following command:

Command	Purpose

`filter url except` *`source_ip source_mask dest_ip dest_mask`*
Example:
`ciscoasa(config)# filter url http 0 0 0 0`
`ciscoasa(config)# filter url except 10.0.2.54 255.255.255.255 0 0`
	Exempts specific traffic from filtering.
	The example shows how to cause all HTTP requests to be forwarded to the filtering server, except for those from 10.0.2.54.

Filtering HTTPS URLs

You must identify and enable the URL filtering server before enabling HTTPS filtering.

Note Websense and Secure Computing Smartfilter currently support HTTPS; older versions of the Secure Computing SmartFilter (formerly N2H2) do not support HTTPS filtering.

Because HTTPS content is encrypted, the ASA sends the URL lookup without directory and filename information. When the filtering server approves an HTTPS connection request, the ASA allows the completion of SSL connection negotiation and allows the reply from the web server to reach the originating client. If the filtering server denies the request, the ASA prevents the completion of SSL connection negotiation. The browser displays an error message, such as “The Page or the content cannot be displayed.”

Note The ASA does not provide an authentication prompt for HTTPS, so you must authenticate with the ASA using HTTP or FTP before accessing HTTPS servers.

To enable HTTPS filtering, enter the following command:

Command	Purpose
filter https *port*[-*port*] *localIP* *local_mask* *foreign_IP* *foreign_mask* [**allow**] **Example:** `ciscoasa# filter https 443 0 0 0 0 0 0 0 0 allow`	Enables HTTPS filtering. Replaces *port*[-*port*] with a range of port numbers if a different port than the default port for HTTPS (443) is used. Replaces *local_ip* and *local_mask* with the IP address and subnet mask of a user or subnetwork making requests. Replaces *foreign_ip* and *foreign_mask* with the IP address and subnet mask of a server or subnetwork responding to requests. The **allow** option causes the ASA to forward HTTPS traffic without filtering when the primary filtering server is unavailable.

Filtering FTP Requests

You must identify and enable the URL filtering server before enabling FTP filtering.

Note Websense and Secure Computing Smartfilter currently support FTP; older versions of Secure Computing SmartFilter (formerly known as N2H2) did not support FTP filtering.

When the filtering server approves an FTP connection request, the ASA allows the successful FTP return code to reach the originating client. For example, a successful return code is “250: CWD command successful.” If the filtering server denies the request, the FTP return code is changed to show that the connection was denied. For example, the ASA changes code 250 to “550 Requested file is prohibited by URL filtering policy.”

To enable FTP filtering, enter the following command:

Command	Purpose
filter ftp *port*[-*port*] *localIP* *local_mask* *foreign_IP* *foreign_mask* [**allow**] [**interact-block**] **Example:** `ciscoasa# filter ftp 21 0 0 0 0 0 0 0 0 allow`	Enables FTP filtering. Replaces *port*[-*port*] with a range of port numbers if a different port than the default port for FTP (21) is used. Replaces *local_ip* and *local_mask* with the IP address and subnet mask of a user or subnetwork making requests. Replaces *foreign_ip* and *foreign_mask* with the IP address and subnet mask of a server or subnetwork responding to requests. The **allow** option causes the ASA to forward HTTPS traffic without filtering when the primary filtering server is unavailable. Use the **interact-block** option to prevent interactive FTP sessions that do not provide the entire directory path. An interactive FTP client allows you to change directories without typing the entire path. For example, you might enter **cd ./files** instead of **cd /public/files**.

Monitoring Filtering Statistics

To monitor filtering statistics, enter one of the following commands:

Command	Purpose
show url-server	Shows information about the URL filtering server.
show url-server statistics	Shows URL filtering statistics.
show url-block	Shows the number of packets held in the url-block buffer and the number (if any) dropped because of exceeding the buffer limit or retransmission.
show url-block block statistics	Shows the URL block statistics.
show url-cache stats	Shows the URL cache statistics.
show perfmon	Shows URL filtering performance statistics, along with other performance statistics.
show filter	Shows the filtering configuration.

Examples

The following is sample output from the **show url-server** command:

```
ciscoasa# show url-server
url-server (outside) vendor n2h2 host 128.107.254.202 port 4005 timeout 5 protocol TCP
```

The following is sample output from the **show url-server statistics** command:

```
ciscoasa# show url-server statistics

Global Statistics:
--------------------
URLs total/allowed/denied         13/3/10
URLs allowed by cache/server      0/3
URLs denied by cache/server       0/10
HTTPSs total/allowed/denied       138/137/1
HTTPSs allowed by cache/server    0/137
HTTPSs denied by cache/server     0/1
FTPs total/allowed/denied         0/0/0
FTPs allowed by cache/server      0/0
FTPs denied by cache/server       0/0
Requests dropped                  0
Server timeouts/retries           0/0
Processed rate average 60s/300s   0/0 requests/second
Denied rate average 60s/300s      0/0 requests/second
Dropped rate average 60s/300s     0/0 requests/second

Server Statistics:
--------------------
10.125.76.20                      UP
  Vendor                          websense
  Port                            15868
  Requests total/allowed/denied   151/140/11
  Server timeouts/retries         0/0
  Responses received              151
  Response time average 60s/300s  0/0

URL Packets Sent and Received Stats:
------------------------------------
Message                 Sent    Received
```

```
STATUS_REQUEST          1609    1601
LOOKUP_REQUEST          1526    1526
LOG_REQUEST             0       NA

Errors:
-------
RFC noncompliant GET method     0
URL buffer update failure       0
```

The following is sample output from the **show url-block** command:

```
ciscoasa# show url-block
    url-block url-mempool 128
    url-block url-size 4
    url-block block 128
```

The following is sample output from the **show url-block block statistics** command:

```
ciscoasa# show url-block block statistics

URL Pending Packet Buffer Stats with max block  128
-----------------------------------------------------
Cumulative number of packets held:              896
Maximum number of packets held (per URL):       3
Current number of packets held (global):        38
Packets dropped due to
       exceeding url-block buffer limit:        7546
       HTTP server retransmission:              10
Number of packets released back to client:      0
```

The following is sample output from the **show url-cache stats** command:

```
ciscoasa# show url-cache stats
URL Filter Cache Stats
----------------------
    Size :     128KB
 Entries :      1724
  In Use :       456
 Lookups :        45
    Hits :         8

This shows how the cache is used.
```

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

```
ciscoasa# show perfmon
PERFMON STATS:    Current      Average
Xlates               0/s          0/s
Connections          0/s          2/s
TCP Conns            0/s          2/s
UDP Conns            0/s          0/s
URL Access           0/s          2/s
URL Server Req       0/s          3/s
TCP Fixup            0/s          0/s
TCPIntercept         0/s          0/s
HTTP Fixup           0/s          3/s
FTP Fixup            0/s          0/s
AAA Authen           0/s          0/s
AAA Author           0/s          0/s
AAA Account          0/s          0/s
```

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

```
ciscoasa# show filter
filter url http 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
```

Feature History for URL Filtering

Table 29-5 lists the release history for URL filtering. ASDM is backwards-compatible with multiple platform releases, so the specific ASDM release in which support was added is not listed.

Table 29-5 Feature History for URL Filtering

Feature Name	Platform Releases	Feature Information
URL filtering	7.0(1)	Filters URLs based on an established set of filtering criteria.

PART 8

Configuring Modules

CHAPTER 30

Configuring the ASA CX Module

This chapter describes how to configure the ASA CX module that runs on the ASA.

Information About the ASA CX Module

The ASA CX module lets you enforce security based on the full context of a situation. This context includes the identity of the user (who), the application or website that the user is trying to access (what), the origin of the access attempt (where), the time of the attempted access (when), and the properties of the device used for the access (how). With the ASA CX module, you can extract the full context of a flow and enforce granular policies such as permitting access to Facebook but denying access to games on Facebook, or permitting finance employees access to a sensitive enterprise database but denying the same access to other employees.

How the ASA CX Module Works with the ASA

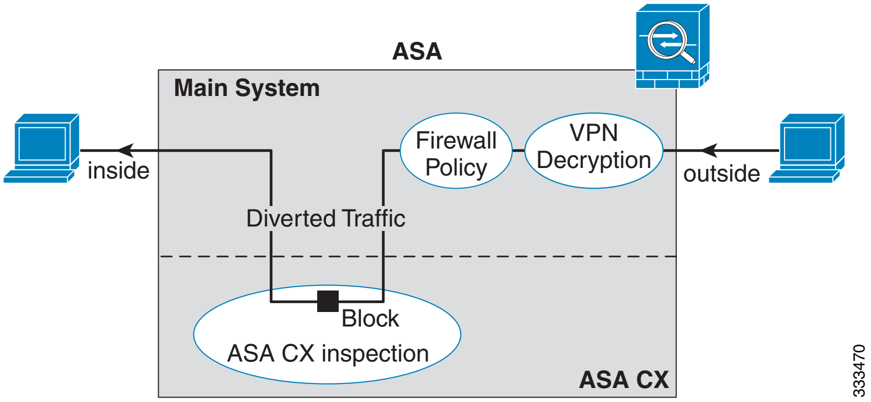
The ASA CX module runs a separate application from the ASA. The ASA CX module includes external management interface(s) so you can connect to the ASA CX module directly. Any data interfaces on the ASA CX module are used for ASA traffic only.

Traffic goes through the firewall checks before being forwarded to the ASA CX module. When you identify traffic for ASA CX inspection on the ASA, traffic flows through the ASA and the ASA CX module as follows:

1. Traffic enters the ASA.
2. Incoming VPN traffic is decrypted.
3. Firewall policies are applied.
4. Traffic is sent to the ASA CX module.
5. The ASA CX module applies its security policy to the traffic, and takes appropriate actions.
6. Valid traffic is sent back to the ASA; the ASA CX module might block some traffic according to its security policy, and that traffic is not passed on.
7. Outgoing VPN traffic is encrypted.
8. Traffic exits the ASA.

Figure 30-1 shows the traffic flow when using the ASA CX module. In this example, the ASA CX module automatically blocks traffic that is not allowed for a certain application. All other traffic is forwarded through the ASA.

Figure 30-1 ASA CX Module Traffic Flow in the ASA



Note If you have a connection between hosts on two ASA interfaces, and the ASA CX service policy is only configured for one of the interfaces, then all traffic between these hosts is sent to the ASA CX module, including traffic originating on the non-ASA CX interface (because the feature is bidirectional). However, the ASA only performs the authentication proxy on the interface to which the service policy is applied, because authentication proxy is applied only to ingress traffic (see the "Information About Authentication Proxy" section on page 30-5).

Monitor-Only Mode

For demonstration purposes, you can configure a service policy or a traffic-forwarding interface in monitor-only mode.

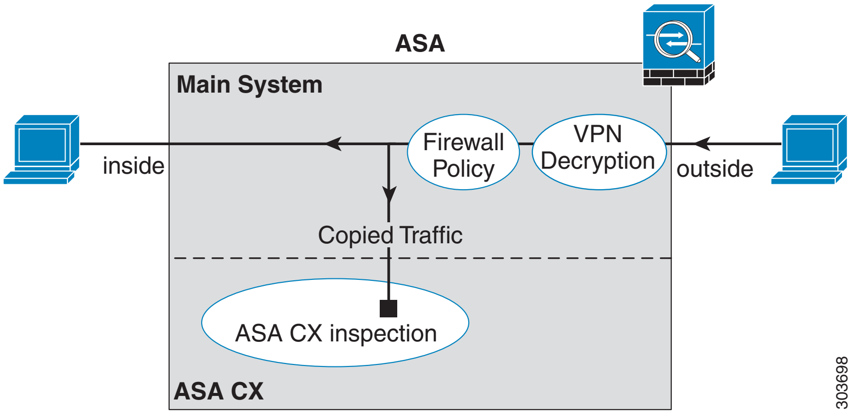
For guidelines and limitations for monitor-only mode, see the "Guidelines and Limitations" section on page 30-6.

- Service Policy in Monitor-Only Mode, page 30-3
- Traffic-Forwarding Interface in Monitor-Only Mode, page 30-3

Service Policy in Monitor-Only Mode

For testing and demonstration purposes, you can configure the ASA to send a duplicate stream of read-only traffic to the ASA CX module, so you can see how the module inspects the traffic without affecting the ASA traffic flow. In this mode, the ASA CX module inspects the traffic as usual, makes policy decisions, and generates events. However, because the packets are read-only copies, the module actions do not affect the actual traffic. Instead, the module drops the copies after inspection. Figure 30-2 shows the ASA CX module in monitor-only mode.

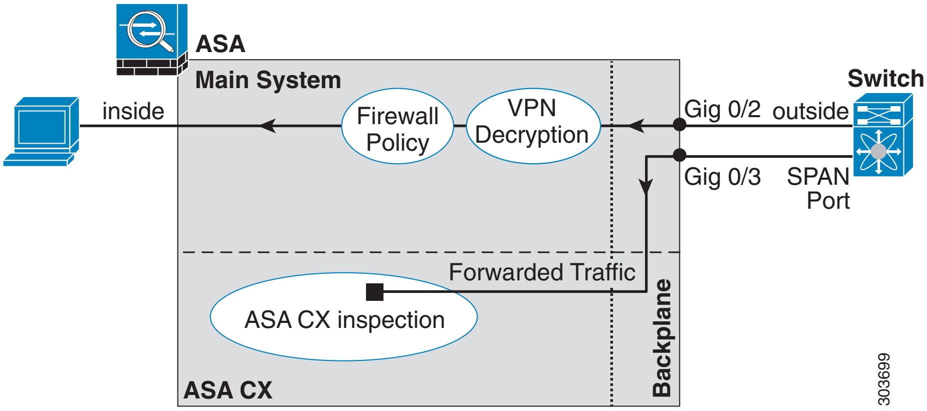
Figure 30-2 ASA CX Monitor-Only Mode



Traffic-Forwarding Interface in Monitor-Only Mode

You can alternatively configure ASA interfaces to be traffic-forwarding interfaces, where all traffic received is forwarded directly to the ASA CX module without any ASA processing. For testing and demonstration purposes, traffic-forwarding removes the extra complication of ASA processing. Traffic-forwarding is only supported in monitor-only mode, so the ASA CX module drops the traffic after inspecting it. Figure 30-3 shows the ASA GigabitEthernet 0/3 interface configured for traffic-forwarding. That interface is connected to a switch SPAN port so the ASA CX module can inspect all of the network traffic.

Figure 30-3 ASA CX Traffic-Forwarding



Information About ASA CX Management

- Initial Configuration, page 30-4
- Policy Configuration and Management, page 30-5

Initial Configuration

For initial configuration, you must use the CLI on the ASA CX module to run the **setup** command and configure other optional settings.

To access the CLI, you can use the following methods:

- ASA 5585-X:
  - ASA CX console port—The ASA CX console port is a separate external console port.
  - ASA CX Management 1/0 interface using SSH—You can connect to the default IP address (192.168.8.8), or you can use ASDM to change the management IP address and then connect using SSH. The ASA CX management interface is a separate external Gigabit Ethernet interface.

Note You cannot access the ASA CX hardware module CLI over the ASA backplane using the **session** command.

- ASA 5512-X through ASA 5555-X:
  - ASA session over the backplane—If you have CLI access to the ASA, then you can session to the module and access the module CLI.
  - ASA CX Management 0/0 interface using SSH—You can connect to the default IP address (192.168.1.2), or you can use ASDM to change the management IP address and then connect using SSH. These models run the ASA CX module as a software module. The ASA CX management interface shares the Management 0/0 interface with the ASA. Separate MAC addresses and IP addresses are supported for the ASA and ASA CX module. You must perform configuration of the ASA CX IP address within the ASA CX operating system (using the CLI

or ASDM). However, physical characteristics (such as enabling the interface) are configured on the ASA. You can remove the ASA interface configuration (specifically the interface name) to dedicate this interface as an ASA CX-only interface. This interface is management-only.

Policy Configuration and Management

After you perform initial configuration, configure the ASA CX policy using Cisco Prime Security Manager (PRSM). Then configure the ASA policy for sending traffic to the ASA CX module using ASDM or the ASA CLI.

Note When using PRSM in multiple device mode, you can configure the ASA policy for sending traffic to the ASA CX module within PRSM, instead of using ASDM or the ASA CLI. Using PRSM lets you consolodate management to a single management system. However, PRSM has some limitations when configuring the ASA service policy; see the ASA CX user guide for more information.

Information About Authentication Proxy

When the ASA CX needs to authenticate an HTTP user (to take advantage of identity policies), you must configure the ASA to act as an authentication proxy: the ASA CX module redirects authentication requests to the ASA interface IP address/proxy port. By default, the port is 885 (user configurable). Configure this feature as part of the service policy to divert traffic from the ASA to the ASA CX module. If you do not enable the authentication proxy, only passive authentication is available.

Note If you have a connection between hosts on two ASA interfaces, and the ASA CX service policy is only configured for one of the interfaces, then all traffic between these hosts is sent to the ASA CX module, including traffic orginiating on the non-ASA CX interface (the feature is bidirectional). However, the ASA only performs the authentication proxy on the interface to which the service policy is applied, because this feature is ingress-only.

Information About VPN and the ASA CX Module

The ASA includes VPN client and user authentication metadata from the Cisco AnyConnect client when forwarding traffic to the ASA CX module, which allows the ASA CX module to include this information as part of its policy lookup criteria. The VPN metadata is sent only at VPN tunnel establishment time along with a type-length-value (TLV) containing the session ID. The ASA CX module caches the VPN metadata for each session. Each tunneled connection sends the session ID so the ASA CX module can look up that session's metadata.

Compatibility with ASA Features

The ASA includes many advanced application inspection features, including HTTP inspection. However, the ASA CX module provides more advanced HTTP inspection than the ASA provides, as well as additional features for other applications, including monitoring and controlling application usage.

To take full advantage of the ASA CX module features, see the following guidelines for traffic that you send to the ASA CX module:

- Do not configure ASA inspection on HTTP traffic.
- Do not configure Cloud Web Security (ScanSafe) inspection. If you configure both the ASA CX action and Cloud Web Security inspection for the same traffic, the ASA only performs the ASA CX action.
- Other application inspections on the ASA are compatible with the ASA CX module, including the default inspections.
- Do not enable the Mobile User Security (MUS) server; it is not compatible with the ASA CX module.
- Do not enable ASA clustering; it is not compatible with the ASA CX module.
- If you enable failover, when the ASA fails over, any existing ASA CX flows are transferred to the new ASA, but the traffic is allowed through the ASA without being acted upon by the ASA CX module. Only new flows recieved by the new ASA are acted upon by the ASA CX module.
- (9.1(1) and earlier) Does not support NAT 64. In 9.1(2) and later, NAT 64 is supported.

Licensing Requirements for the ASA CX Module

Model	License Requirement
All models	Base License.

The ASA CX module and PRSM require additional licenses. See the ASA CX documentation for more information.

Prerequisites

To use PRSM to configure the ASA, you need to install a certificate on the ASA for secure communications. By default, the ASA generates a self-signed certificate. However, this certificate can cause browser prompts asking you to verify the certificate because the publisher is unknown. To avoid these browser prompts, you can instead install a certificate from a known certificate authority (CA). If you request a certificate from a CA, be sure the certificate type is both a server authentication certificate and a client authentication certificate. See the Chapter 40, “Configuring Digital Certificates,” in the general operations configuration guide for more information.

Guidelines and Limitations

Context Mode Guidelines

(9.1(2) and earlier) Supported in single context mode only. Does not support multiple context mode.

(9.1(3) and later) Supported in multiple context mode. See the following guidelines:

- The ASA CX module itself (configured in PRSM) is a single context mode device; the context-specific traffic coming from the ASA is checked against the common ASA CX policy.
- For ASA CX module support, you cannot use the same IP addresses in multiple contexts; each context must include unique networks.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode. Traffic-forwarding interfaces are only supported in transparent mode.

Failover Guidelines

Does not support failover directly; when the ASA fails over, any existing ASA CX flows are transferred to the new ASA, but the traffic is allowed through the ASA without being inspected by the ASA CX.

ASA Clustering Guidelines

Does not support clustering.

IPv6 Guidelines

- Supports IPv6.
- (9.1(1) and earlier) Does not support NAT 64. In 9.1(2) and later, NAT 64 is supported.

Model Guidelines

- Supported only on the ASA 5585-X and 5512-X through ASA 5555-X. See the *Cisco ASA Compatibility Matrix* for more information:

  http://www.cisco.com/en/US/docs/security/asa/compatibility/asamatrx.html
- For the 5512-X through ASA 5555-X, you must install a Cisco solid state drive (SSD). For more information, see the ASA 5500-X hardware guide.

Monitor-Only Mode Guidelines

- You cannot configure both monitor-only mode and normal inline mode at the same time on the ASA. Only one type of security policy is allowed. In multiple context mode, you cannot configure monitor-only mode for some contexts, and regular inline mode for others.
- The following features are not supported in monitor-only mode:
  - Deny policies
  - Active authentication
  - Decryption policies
- The ASA CX does not perform packet buffering in monitor-only mode, and events will be generated on a best-effort basis. For example, some events, such as ones with long URLs spanning packet boundaries, may be impacted by the lack of buffering.
- Be sure to configure both the ASA policy and the ASA CX to have matching modes: both in monitor-only mode, or both in normal inline mode.

Additional guidelines for traffic-forwarding interfaces:

- The ASA must be in transparent mode.
- You can configure up to 4 interfaces as traffic-forwarding interfaces. Other ASA interfaces can be used as normal.
- Traffic-forwarding interfaces must be physical interfaces, not VLANs or BVIs. The physical interface also cannot have any VLANs associated with it.
- Traffic-forwarding interfaces cannot be used for ASA traffic; you cannot name them or configure them for ASA features, including failover or management-only.
- You cannot configure both a traffic-forwarding interface and a service policy for ASA CX traffic.

Additional Guidelines and Limitations

- See the "Compatibility with ASA Features" section on page 30-5.
- You cannot change the software type installed on the hardware module; if you purchase an ASA CX module, you cannot later install other software on it.

Default Settings

Table 30-1 lists the default settings for the ASA CX module.

Table 30-1 Default Network Parameters

Parameters	Default
Management IP address	ASA 5585-X: Management 1/0 192.168.8.8/24 ASA 5512-X through ASA 5555-X: Management 0/0 192.168.1.2/24
Gateway	ASA 5585-X: 192.168.8.1/24 ASA 5512-X through ASA 5555-X: 192.168.1.1/24
SSH or session Username	admin
Password	Admin123

Configuring the ASA CX Module

This section describes how to configure the ASA CX module.

- Task Flow for the ASA CX Module, page 30-8
- Connecting the ASA CX Management Interface, page 30-9
- (ASA 5585-X) Changing the ASA CX Management IP Address, page 30-14
- (ASA 5512-X through ASA 5555-X; May Be Required) Installing the Software Module, page 30-12
- Configuring Basic ASA CX Settings at the ASA CX CLI, page 30-15
- Configuring the Security Policy on the ASA CX Module Using PRSM, page 30-16
- Redirecting Traffic to the ASA CX Module, page 30-18

Task Flow for the ASA CX Module

Configuring the ASA CX module is a process that includes configuration of the ASA CX security policy on the ASA CX module and then configuration of the ASA to send traffic to the ASA CX module. To configure the ASA CX module, perform the following steps:

Step 1 Cable the ASA CX management interfaces and optionally, the console interface. See the "Connecting the ASA CX Management Interface" section on page 30-9.

Step 2 (ASA 5512-X through ASA 5555-X; May be required) Install the software module. See the "(ASA 5512-X through ASA 5555-X; May Be Required) Installing the Software Module" section on page 30-12.

Step 3 (ASA 5585-X; Optional) Configure the ASA CX module management IP address for initial SSH access. See the "(ASA 5585-X) Changing the ASA CX Management IP Address" section on page 30-14.

Step 4 On the ASA CX module, configure basic settings. See the "Configuring Basic ASA CX Settings at the ASA CX CLI" section on page 30-15.

Step 5 On the ASA CX module, configure the security policy using PRSM. See the "Configuring the Security Policy on the ASA CX Module Using PRSM" section on page 30-16.

Step 6 (Optional) On the ASA, configure the authentication proxy port. See the "(Optional) Configuring the Authentication Proxy Port" section on page 30-17.

Step 7 On the ASA, identify traffic to divert to the ASA CX module. See the "Redirecting Traffic to the ASA CX Module" section on page 30-18.

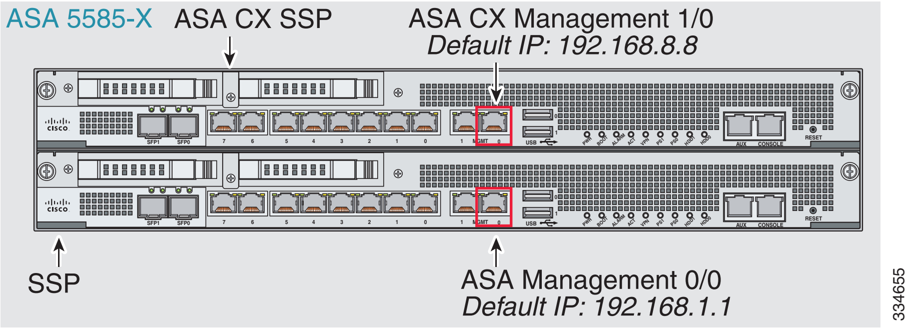
Note When using PRSM in multiple device mode, you can configure the ASA policy for sending traffic to the ASA CX module within PRSM, instead of using ASDM or the ASA CLI. However, PRSM has some limitations when configuring the ASA service policy; see the ASA CX user guide for more information.

Connecting the ASA CX Management Interface

In addition to providing management access to the ASA CX module, the ASA CX management interface needs access to an HTTP proxy server or a DNS server and the Internet for signature updates and more. This section describes recommended network configurations. Your network may differ.

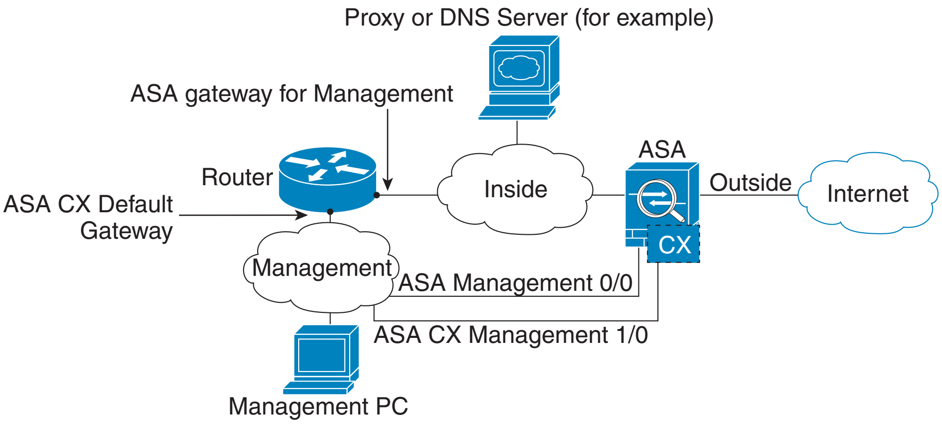
ASA 5585-X (Hardware Module)

The ASA CX module includes a separate management and console interface from the ASA. For initial setup, you can connect with SSH to the ASA CX Management 1/0 interface using the default IP address (192.168.8.8/24). If you cannot use the default IP address, you can either use the console port or use ASDM to change the management IP address so you can use SSH.



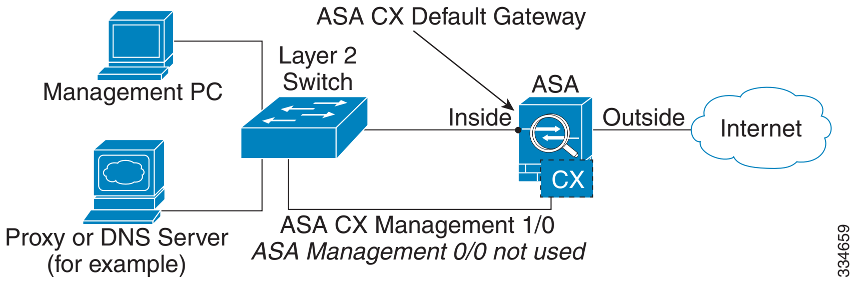
If you have an inside router

If you have an inside router, you can route between the management network, which can include both the ASA Management 0/0 and ASA CX Management 1/0 interfaces, and the ASA inside network for Internet access. Be sure to also add a route on the ASA to reach the Management network through the inside router.



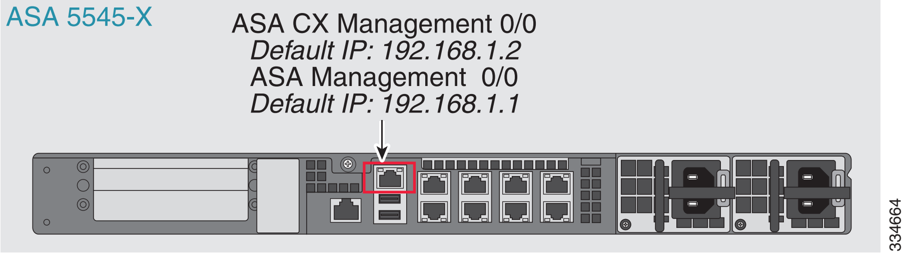
If you do not have an inside router

If you have only one inside network, then you cannot also have a separate management network, which would require an inside router to route between the networks. In this case, you can manage the ASA from the inside interface instead of the Management 0/0 interface. Because the ASA CX module is a separate device from the ASA, you can configure the ASA CX Management 1/0 address to be on the same network as the inside interface.



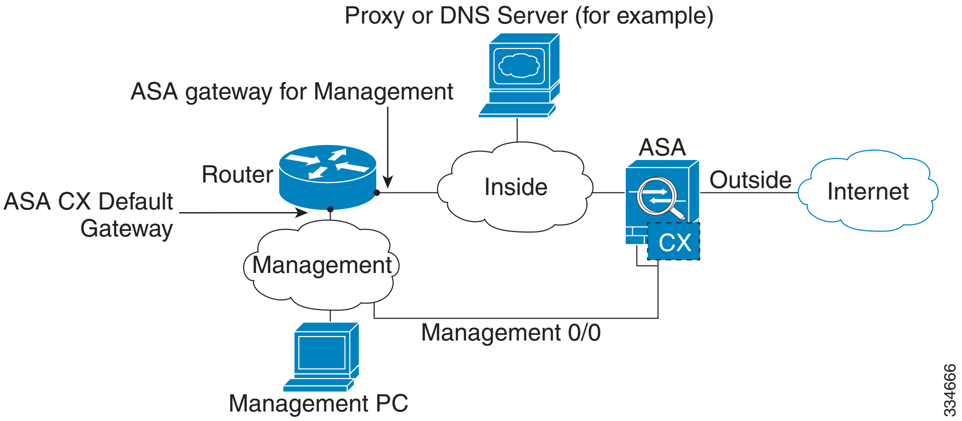
ASA 5512-X through ASA 5555-X (Software Module)

These models run the ASA CX module as a software module, and the ASA CX management interface shares the Management 0/0 interface with the ASA. For initial setup, you can connect with SSH to the ASA CX default IP address (192.168.1.2/24). If you cannot use the default IP address, you can either session to the ASA CX over the backplane or use ASDM to change the management IP address so you can use SSH.



If you have an inside router

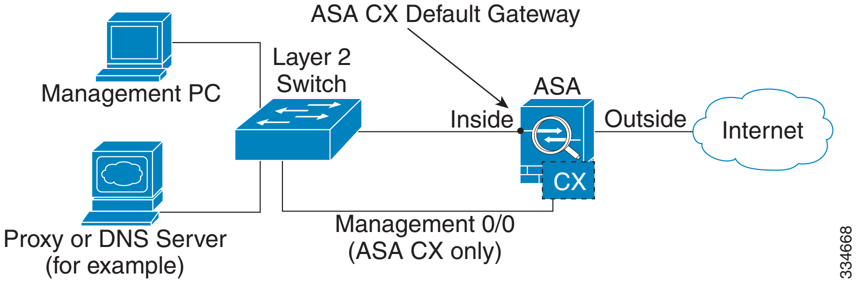
If you have an inside router, you can route between the Management 0/0 network, which includes both the ASA and ASA CX management IP addresses, and the inside network for Internet access. Be sure to also add a route on the ASA to reach the Management network through the inside router.



If you do not have an inside router

If you have only one inside network, then you cannot also have a separate management network. In this case, you can manage the ASA from the inside interface instead of the Management 0/0 interface. If you remove the ASA-configured name from the Management 0/0 interface, you can still configure the ASA

CX IP address for that interface. Because the ASA CX module is essentially a separate device from the ASA, you *can* configure the ASA CX management address to be on the same network as the inside interface.



Note You must remove the ASA-configured name for Management 0/0; if it is configured on the ASA, then the ASA CX address must be on the same network as the ASA, and that excludes any networks already configured on other ASA interfaces. If the name is not configured, then the ASA CX address can be on any network, for example, the ASA inside network.

What to Do Next

- (Optional) Configure the ASA CX management IP address. See the "(ASA 5585-X) Changing the ASA CX Management IP Address" section on page 30-14.
- Configure basic ASA CX settings. See the "Configuring Basic ASA CX Settings at the ASA CX CLI" section on page 30-15.

(ASA 5512-X through ASA 5555-X; May Be Required) Installing the Software Module

If you purchase the ASA with the ASA CX module, the module software and required solid state drive(s) (SSDs) come pre-installed and ready to go. If you want to add the ASA CX to an existing ASA, or need to replace the SSD, you need to install the ASA CX boot software and partition the SSD according to this procedure. To physically install the SSD, see the ASA hardware guide.

Note For the ASA 5585-X hardware module, you must install or upgrade your image from within the ASA CX module. See the ASA CX module documentation for more information.

Prerequisites

- The free space on flash (disk0) should be at least 3GB plus the size of the boot software.
- In multiple context mode, perform this procedure in the system execution space.

Detailed Steps

Step 1 Download the ASA CX boot software from Cisco.com to your computer. If you have a Cisco.com login, you can obtain the boot software from the following website:

http://www.cisco.com/cisco/software/release.html?mdfid=284325223&softwareid=284399946

The boot software lets you set basic ASA CX network configuration, partition the SSD, and download the larger system software from a server of your choice to the SSD.

Step 2 Download the ASA CX system software from Cisco.com to an HTTP, HTTPS, or FTP server accessible from the ASA CX management interface. If you have a Cisco.com login, you can obtain the boot software from the following website:

http://www.cisco.com/cisco/software/release.html?mdfid=284325223&softwareid=284399946

Step 3 Copy the boot software to disk0 on the ASA using the **copy** command. Do *not* transfer the system software; it is downloaded later to the SSD. For example:

```
ciscoasa# copy tftp://10.1.1.1/asacx-boot-9.1.1.img disk0:/asacx-boot-9.1.1.img
```

Step 4 If you are replacing the IPS module with the ASA CX module, shut down and uninstall the IPS module, and then reload the ASA:

```
ciscoasa# sw-module module ips shutdown
ciscoasa# sw-module module ips uninstall
ciscoasa# reload
```

After the ASA reloads, reconnect to the ASA CLI.

Step 5 Set the ASA CX module boot image location in ASA disk0 by entering the following command:

```
ciscoasa# sw-module module cxsc recover configure image disk0:file_path
```

Example:

```
ciscoasa# sw-module module cxsc recover configure image disk0:asacx-boot-9.1.1.img
```

Step 6 Load the ASA CX boot image by entering the following command:

```
ciscoasa# sw-module module cxsc recover boot
```

Step 7 Wait approximately 5 minutes for the ASA CX module to boot up, and then open a console session to the now-running ASA CX boot image. The default username is **admin** and the default password is **Admin123**.

```
ciscoasa# session cxsc console
Establishing console session with slot 1
Opening console session with module cxsc.
Connected to module cxsc. Escape character sequence is 'CTRL-SHIFT-6 then x'.
cxsc login: admin
Password: Admin123
```

Step 8 Partition the SSD:

```
asacx-boot> partition
....
Partition Successfully Completed
```

Step 9 Perform the basic network setup using the **setup** command according to the "Configuring Basic ASA CX Settings at the ASA CX CLI" section on page 30-15 (do not exit the ASA CX CLI), and then return to this procedure to install the software image.

Step 10 Install the system software from the server:

```
asacx-boot> system install url
```

Example:

The following command installs the asacx-sys-9.1.1.pkg system software.

```
asacx-boot> system install https://upgrades.example.com/packages/asacx-sys-9.1.1.pkg
```

```
Username: buffy
Password: angelforever
Verifying
Downloading
Extracting
Package Detail
        Description:
        Requires reboot:
Cisco ASA CX System Upgrade
Yes
Do you want to continue with upgrade? [n]: Y
Warning: Please do not interrupt the process or turn off the system. Doing so might leave
system in unusable state.
Upgrading
Stopping all the services ...
Starting upgrade process ...
Reboot is required to complete the upgrade. Press Enter to reboot the system.
```

Step 11 Press **Enter** to reboot the ASA CX module. Rebooting the module closes the console session. Allow 10 or more minutes for application component installation and for the ASA CX services to start.

(ASA 5585-X) Changing the ASA CX Management IP Address

If you cannot use the default management IP address (192.168.8.8), then you can set the management IP address from the ASA. After you set the management IP address, you can access the ASA CX module using SSH to perform initial setup.

Note For a software module, you can access the ASA CX CLI to perform setup by sessioning from the ASA CLI; you can then set the ASA CX management IP address as part of setup. See the "Configuring Basic ASA CX Settings at the ASA CX CLI" section on page 30-15.

Guidelines

In multiple context mode, perform this procedure in the system execution space.

Detailed Steps

Command	Purpose
session 1 do setup host ip *ip_address***/***mask***,***gateway_ip* **Example:** `ciscoasa# session 1 do setup host ip 10.1.1.2/24,10.1.1.1`	Sets the ASA CX management IP address, mask, and gateway.

Configuring Basic ASA CX Settings at the ASA CX CLI

You must configure basic network settings and other parameters on the ASA CX module before you can configure your security policy.

Detailed Steps

Step 1 Do one of the following:

- (All models) Use SSH to connect to the ASA CX management IP address.
- (ASA 5512-X through ASA 5555-X) Open a console session to the module from the ASA CLI (see the "Getting Started" chapter in the general operations configuration guide to access the ASA CLI). In multiple context mode, session from the system execution space.

```
ciscoasa# session cxsc console
```

Step 2 Log in with the username **admin** and the password **Admin123**. You will change the password as part of this procedure.

Step 3 Enter the following command:

```
asacx> setup
```

Example:

```
asacx> setup
Welcome to Cisco Prime Security Manager Setup
[hit Ctrl-C to abort]
Default values are inside [ ]
```

You are prompted through the setup wizard. The following example shows a typical path through the wizard; if you enter **Y** instead of **N** at a prompt, you will be able to configure some additional settings. This example shows how to configure both IPv4 and IPv6 static addresses. You can configure IPv6 stateless auto configuration by answering **N** when asked if you want to configure a static IPv6 address.

```
Enter a hostname [asacx]: asa-cx-host
Do you want to configure IPv4 address on management interface?(y/n) [Y]: Y
Do you want to enable DHCP for IPv4 address assignment on management interface?(y/n)[N]: N
Enter an IPv4 address [192.168.8.8]: 10.89.31.65
Enter the netmask [255.255.255.0]: 255.255.255.0
Enter the gateway [192.168.8.1]: 10.89.31.1
Do you want to configure static IPv6 address on management interface?(y/n) [N]: Y
Enter an IPv6 address: 2001:DB8:0:CD30::1234/64
Enter the gateway: 2001:DB8:0:CD30::1
Enter the primary DNS server IP address [ ]: 10.89.47.11
Do you want to configure Secondary DNS Server? (y/n) [N]: N
Do you want to configure Local Domain Name? (y/n) [N] Y
Enter the local domain name: example.com
Do you want to configure Search domains? (y/n) [N] Y
Enter the comma separated list for search domains: example.com
Do you want to enable the NTP service?(y/n) [N]: Y
Enter the NTP servers separated by commas: 1.ntp.example.com, 2.ntp.example.com
```

Step 4 After you complete the final prompt, you are presented with a summary of the settings. Look over the summary to verify that the values are correct, and enter **Y** to apply your changed configuration. Enter **N** to cancel your changes.

Example:

```
Apply the changes?(y,n) [Y]: Y
Configuration saved successfully!
```

```
Applying...
Done.
Generating self-signed certificate, the web server will be restarted after that
...
Done.
Press ENTER to continue...
asacx>
```

Note If you change the host name, the prompt does not show the new name until you log out and log back in.

Step 5 If you do not use NTP, configure the time settings. The default time zone is the UTC time zone. Use the **show time** command to see the current settings. You can use the following commands to change time settings:

```
asacx> config timezone
asacx> config time
```

Step 6 Change the admin password by entering the following command:

```
asacx> config passwd
```

Example:

```
asacx> config passwd
The password must be at least 8 characters long and must contain
at least one uppercase letter (A-Z), at least one lowercase letter
(a-z) and at least one digit (0-9).
Enter password: Farscape1
Confirm password: Farscape1
SUCCESS: Password changed for user admin
```

Step 7 Enter the **exit** command to log out.

Configuring the Security Policy on the ASA CX Module Using PRSM

This section describes how to launch PRSM to configure the ASA CX module application. For details on using PRSM to configure your ASA CX security policy, see the ASA CX user guide.

Detailed Steps

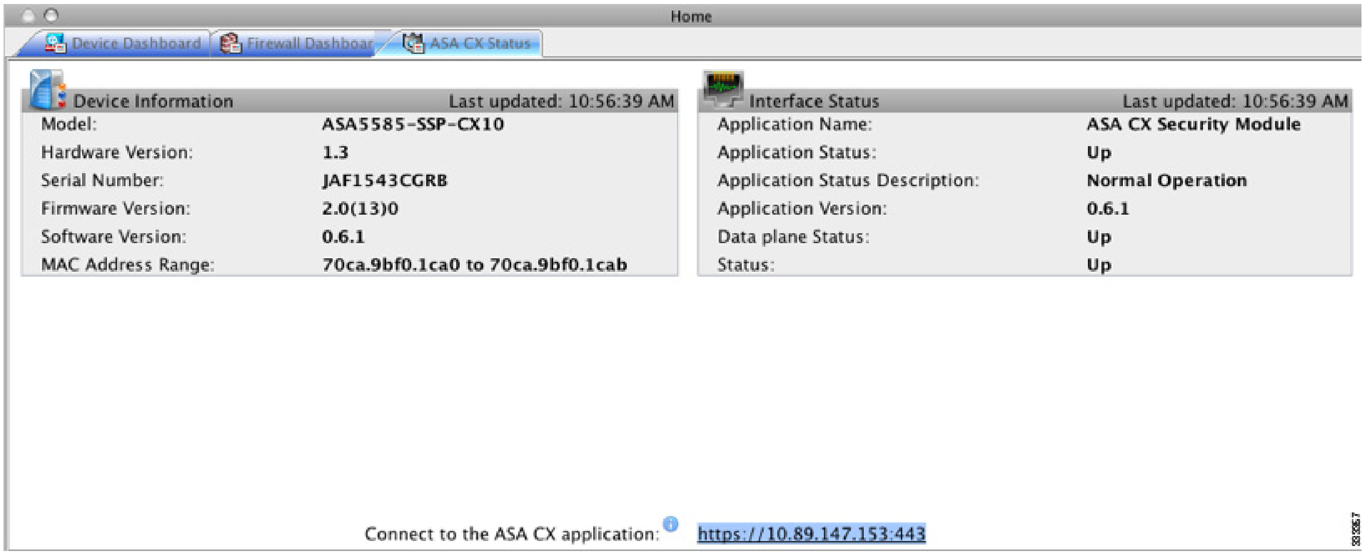
You can launch PRSM from your web browser, or you can launch it from ASDM.

- Launch PRSM from a web browser by enter the following URL:

  ```
  https://ASA_CX_management_IP
  ```

  Where the ASA CX management IP address is the one you set in the "Configuring Basic ASA CX Settings at the ASA CX CLI" section on page 30-15.

- Launch PRSM from ASDM by choosing **Home > ASA CX Status**, and clicking the **Connect to the ASA CX application** link.



Device Information	Last updated: 10:56:39 AM
Model:	ASA5585-SSP-CX10
Hardware Version:	1.3
Serial Number:	JAF1543CGRB
Firmware Version:	2.0(13)0
Software Version:	0.6.1
MAC Address Range:	70ca.9bf0.1ca0 to 70ca.9bf0.1cab

Interface Status	Last updated: 10:56:39 AM
Application Name:	ASA CX Security Module
Application Status:	Up
Application Status Description:	Normal Operation
Application Version:	0.6.1
Data plane Status:	Up
Status:	Up

Connect to the ASA CX application: https://10.89.147.153:443

What to Do Next

- (Optional) Configure the authentication proxy port. See the "(Optional) Configuring the Authentication Proxy Port" section on page 30-17.
- Redirect traffic to the ASA CX module. See the "Redirecting Traffic to the ASA CX Module" section on page 30-18.

(Optional) Configuring the Authentication Proxy Port

The default authentication port is 885. To change the authentication proxy port, perform the following steps. For more information about the authentication proxy, see the "Information About Authentication Proxy" section on page 30-5.

Guidelines

In multiple context mode, perform this procedure within each security context.

Detailed Steps

Command	Purpose
cxsc auth-proxy port *port* **Example:** `ciscoasa(config)# cxsc auth-proxy port 5000`	Sets the authentication proxy port greater than 1024. The default is 885.

Redirecting Traffic to the ASA CX Module

You can redirect traffic to the ASA CX module by creating a service policy that identifies specific traffic. For demonstration purposes only, you can also enable monitor-only mode for the service policy, which forwards a copy of traffic to the ASA CX module, while the original traffic remains unaffected.

Another option for demonstration purposes is to configure a traffic-forwarding interface instead of a service policy in monitor-only mode. The traffic-forwarding interface sends all traffic directly to the ASA CX module, bypassing the ASA.

- Creating the ASA CX Service Policy, page 30-18
- Configuring Traffic-Forwarding Interfaces (Monitor-Only Mode), page 30-20

Creating the ASA CX Service Policy

This section identifies traffic to redirect from the ASA to the ASA CX module. Configure this policy on the ASA. If you want to use a traffic-forwarding interface for demonstration purposes, skip this procedure and see the "Configuring Traffic-Forwarding Interfaces (Monitor-Only Mode)" section on page 30-20 instead.

Note When using PRSM in multiple device mode, you can configure the ASA policy for sending traffic to the ASA CX module within PRSM, instead of using ASDM or the ASA CLI. However, PRSM has some limitations when configuring the ASA service policy; see the ASA CX user guide for more information.

Prerequisites

- If you enable the authentication proxy on the ASA using this procedure, be sure to also configure a directory realm for authentication on the ASA CX module. See the ASA CX user guide for more information.
- If you have an active service policy redirecting traffic to an IPS module (that you replaced with the ASA CX), you must remove that policy before you configure the ASA CX service policy.
- Be sure to configure both the ASA policy and the ASA CX to have matching modes: both in monitor-only mode, or both in normal inline mode.
- In multiple context mode, perform this procedure within each security context.

Detailed Steps

	Command	Purpose

Step 1	**class-map** *name*
	Example:
	`ciscoasa(config)# class-map cx_class`
		Creates a class map to identify the traffic for which you want to send to the ASA CX module.
		If you want to send multiple traffic classes to the ASA CX module, you can create multiple class maps for use in the security policy.

Step 2	**match** *parameter* **Example:** `ciscoasa(config-cmap)# match access-list cx_traffic`	Specifies the traffic in the class map. See the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12 for more information.
Step 3	**policy-map** *name* **Example:** `ciscoasa(config)# policy-map cx_policy`	Adds or edits a policy map that sets the actions to take with the class map traffic.

Step 4	**class** *name*
	Example:
	`ciscoasa(config-pmap)# class cx_class`
		Identifies the class map you created in Step 1.

Step 5	**cxsc** {**fail-close** | **fail-open**} [**auth-proxy** | **monitor-only**] **Example:** `ciscoasa(config-pmap-c)# cxsc fail-close auth-proxy`	Specifies that the traffic should be sent to the ASA CX module. The **fail-close** keyword sets the ASA to block all traffic if the ASA CX module is unavailable. The **fail-open** keyword sets the ASA to allow all traffic through, uninspected, if the ASA CX module is unavailable. The optional **auth-proxy** keyword enables the authentication proxy, which is required for active authentication. For demonstration purposes only, specify **monitor-only** to send a read-only copy of traffic to the ASA CX module. When you configure this option, you see a warning message similar to the following: `WARNING: Monitor-only mode should be used for demonstrations and evaluations only.  This mode prevents CXSC from denying or altering traffic.` See the "Monitor-Only Mode" section on page 30-3 for more information. **Note** You must configure all classes and policies to be either in monitor-only mode, or in normal inline mode; you cannot mix both modes on the same ASA.

	Command	Purpose

Step 6	(Optional)
	class *name2*
	Example:
	`ciscoasa(config-pmap)# class cx_class2`
		If you created multiple class maps for ASA CX traffic, you can specify another class for the policy.
		See the "Feature Matching Within a Service Policy" section on page 1-3 for detailed information about how the order of classes matters within a policy map. Traffic cannot match more than one class map for the same action type.

Step 7	(Optional) **cxsc** {**fail-close** | **fail-open**} [**auth-proxy** | **monitor-only**] **Example:** `ciscoasa(config-pmap-c)# cxsc fail-close auth-proxy`	Specifies that the second class of traffic should be sent to the ASA CX module. Add as many classes as desired by repeating these steps.

Step 8	**service-policy** *policymap_name* {**global** | **interface** *interface_name*}
	Example:
	`ciscoasa(config)# service-policy cx_policy interface outside`
		Activates the policy map on one or more interfaces. **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

Configuring Traffic-Forwarding Interfaces (Monitor-Only Mode)

This section configures traffic-forwarding interfaces, where all traffic is forwarded directly to the ASA CX module. This method is for demonstration purposes only. For a normal ASA CX service policy, see the "Creating the ASA CX Service Policy" section on page 30-18.

For more information see the "Monitor-Only Mode" section on page 30-3. See also the "Guidelines and Limitations" section on page 30-6 for guidelines and limitations specific to traffic-forwarding interfaces.

Prerequisites

- Be sure to configure both the ASA policy and the ASA CX to have matching modes: both in monitor-only.
- In multiple context mode, perform this procedure within each security context.

Detailed Steps

	Command	Purpose
Step 1	**interface** *physical_interface* **Example:** `ciscoasa(config)# interface gigabitethernet 0/5`	Enters interface configuration mode for the physical interface you want to use for traffic-forwarding.
Step 2	**no nameif** **Example:** `ciscoasa(config-ifc)# no nameif`	Removes any name configured for the interface. If this interface was used in any ASA configuration, that configuration is removed. You cannot configure traffic-forwarding on a named interface.
Step 3	**traffic-forward cxsc monitor-only** **Example:** `ciscoasa(config-ifc)# traffic-forward cxsc monitor-only`	Enables traffic-forwarding. You see a warning similar to the following: `WARNING: This configuration is purely for demo of CX functionality and shouldn't be used on a production ASA and any issues found when mixing demo feature with production ASA is not supported.`
Step 4	**no shutdown** **Example:** `ciscoasa(config-ifc)# no shutdown`	Enables the interface.

Step 8 Repeat for any additional interfaces.

Step 9 Click **Send**.

Examples

The following example makes GigabitEthernet 0/5 a traffic-forwarding interface:

```
interface gigabitethernet 0/5
  no nameif
  traffic-forward cxsc monitor-only
  no shutdown
```

Managing the ASA CX Module

This section includes procedures that help you manage the module.

- Resetting the Password, page 30-22
- Reloading or Resetting the Module, page 30-22
- Shutting Down the Module, page 30-23
- (ASA 5512-X through ASA 5555-X) Uninstalling a Software Module Image, page 30-24
- (ASA 5512-X through ASA 5555-X) Sessioning to the Module From the ASA, page 30-24

Resetting the Password

You can reset the module password to the default. For the user **admin**, the default password is **Admin123**. After resetting the password, you should change it to a unique value using the module application.

Resetting the module password causes the module to reboot. Services are not available while the module is rebooting.

To reset the module password to the default of Admin123, perform the following steps.

Guidelines

In multiple context mode, perform this procedure in the system execution space.

Detailed Steps

Command	Purpose
For a hardware module (ASA 5585-X): **hw-module module 1 password-reset** For a software module (ASA 5512-X through ASA 5555-X): **sw-module module cxsc password-reset** **Example:** `ciscoasa# hw-module module 1 password-reset`	Resets the module password to **Admin123** for user **admin**.

Reloading or Resetting the Module

To reload or reset the module, enter one of the following commands at the ASA CLI.

Guidelines

In multiple context mode, perform this procedure in the system execution space.

Detailed Steps

Command	Purpose
For a hardware module (ASA 5585-X): **hw-module module 1 reload** For a software module (ASA 5512-X through ASA 5555-X): **sw-module module cxsc reload** **Example:** `ciscoasa# hw-module module 1 reload`	Reloads the module software.
For a hardware module: **hw-module module 1 reset** For a software module: **sw-module module cxsc reset** **Example:** `ciscoasa# hw-module module 1 reset`	Performs a reset, and then reloads the module.

Shutting Down the Module

Shutting down the module software prepares the module to be safely powered off without losing configuration data. **Note**: If you reload the ASA, the module is not automatically shut down, so we recommend shutting down the module before reloading the ASA. To gracefully shut down the module, perform the following steps at the ASA CLI.

Guidelines

In multiple context mode, perform this procedure in the system execution space.

Detailed Steps

Command	Purpose
For a hardware module (ASA 5585-X): **hw-module module 1 shutdown** For a software module (ASA 5512-X through ASA 5555-X): **sw-module module cxsc shutdown** **Example:** `ciscoasa# hw-module module 1 shutdown`	Shuts down the module.

(ASA 5512-X through ASA 5555-X) Uninstalling a Software Module Image

To uninstall a software module image and associated configuration, perform the following steps.

Guidelines

In multiple context mode, perform this procedure in the system execution space.

Detailed Steps

	Command	Purpose

Step 1	**`sw-module module cxsc uninstall`**
	Example:
	`ciscoasa# sw-module module cxsc uninstall`
	`Module cxsc will be uninstalled. This will`
	`completely remove the`
	`disk image associated with the sw-module`
	`including any configuration`
	`that existed within it.`
	`Uninstall module <id>? [confirm]`
		Permanently uninstalls the software module image and associated configuration.

Step 2	**`reload`**
	Example:
	`ciscoasa# reload`
		Reloads the ASA. You must reload the ASA before you can install a new module type.

(ASA 5512-X through ASA 5555-X) Sessioning to the Module From the ASA

To access the ASA CX software module CLI from the ASA, you can session from the ASA. You can either session to the module (using Telnet) or create a virtual console session. A console session might be useful if the control plane is down and you cannot establish a Telnet session.

Guidelines

In multiple context mode, perform this procedure in the system execution space.

Detailed Steps

Command	Purpose

Telnet session.
`session cxsc`
 	
Example:
`ciscoasa# session cxsc`
 	
`Opening command session with slot 1.`
`Connected to module cxsc. Escape character`
`sequence is 'CTRL-^X'.`
 	
`cxsc login: admin`
`Password: Admin123`
	Accesses the module using Telnet. You are prompted for the username and password. The default username is **admin**, and the default password is **Admin123**.

Console session.
`session cxsc console`
 	
Example:
`ciscoasa# session cxsc console`
 	
`Establishing console session with slot 1`
`Opening console session with module cxsc.`
`Connected to module cxsc. Escape character`
`sequence is 'CTRL-SHIFT-6 then x'.`
 	
`cxsc login: admin`
`Password: Admin123`
	Accesses the module console. You are prompted for the username and password. The default username is **admin**, and the default password is **Admin123**.
	
	Note Do not use this command in conjunction with a terminal server where **Ctrl-Shift-6, x** is the escape sequence to return to the terminal server prompt. **Ctrl-Shift-6, x** is also the sequence to escape the ASA CX console and return to the ASA prompt. Therefore, if you try to exit the ASA CX console in this situation, you instead exit all the way to the terminal server prompt. If you reconnect the terminal server to the ASA, the ASA CX console session is still active; you can never exit to the ASA prompt. You must use a direct serial connection to return the console to the ASA prompt.
	
	Use the **session cxsc** command instead.

Monitoring the ASA CX Module

Note For ASA CX-related syslog messages, see the syslog messages guide. ASA CX syslog messages start with message number 429001.

Showing Module Status

To check the status of a module, enter one of the following commands:

Command	Purpose
show module	Displays the status.
show module {**1** | **cxsc**} **details**	Displays additional status information. Specify **1** for a hardware module and **cxsc** for a software module.
show module cxsc recover	Displays the network parameters for transferring a software module boot image.

Examples

The following is sample output from the **show module** command for an ASA with an ASA CX SSP installed:

```
hostname# show module
Mod Card Type                                    Model              Serial No.
--- -------------------------------------------- ------------------ -----------
  0 ASA 5585-X Security Services Processor-10 wi ASA5585-SSP-10     JAF1507AMKE
  1 ASA 5585-X CX Security Services Processor-10 ASA5585-SSP-CX10   JAF1510BLSA

Mod MAC Address Range                 Hw Version   Fw Version   Sw Version
--- --------------------------------- ------------ ------------ ---------------
  0 5475.d05b.1100 to 5475.d05b.110b  1.0          2.0(7)0      100.7(6)78
  1 5475.d05b.2450 to 5475.d05b.245b  1.0          2.0(13)0     0.6.1

Mod SSM Application Name           Status           SSM Application Version
--- ------------------------------ ---------------- --------------------------
  1 ASA CX Security Module         Up               0.6.1

Mod Status             Data Plane Status     Compatibility
--- ------------------ --------------------- -------------
  0 Up Sys             Not Applicable
  1 Up                 Up
```

Showing Module Statistics

To show module statistics, enter the following command:

Command	Purpose
show service-policy cxsc	Displays the ASA CX statistics and status per service policy.

Examples

The following is sample output from the **show service-policy** command showing the ASA CX policy and the current statistics as well as the module status when the authentication proxy is disabled:

```
hostname# show service-policy cxsc
Global policy:
  Service-policy: global_policy
    Class-map: bypass
      CXSC: card status Up, mode fail-open, auth-proxy disabled
        packet input 2626422041, packet output 2626877967, drop 0, reset-drop 0, proxied 0
```

The following is sample output from the **show service-policy** command showing the ASA CX policy and the current statistics as well as the module status when the authentication proxy is enabled; in this case, the proxied counters also increment:

```
hostname# show service-policy cxsc
Global policy:
  Service-policy: pmap
    Class-map: class-default
      Default Queueing      Set connection policy: random-sequence-number disable
        drop 0
      CXSC: card status Up, mode fail-open, auth-proxy enabled
        packet input 7724, packet output 7701, drop 0, reset-drop 0, proxied 10
```

Monitoring Module Connections

To show connections through the ASA CX module, enter one of the following commands:

Command	Purpose
show asp table classify domain cxsc	Shows the NP rules created to send traffic to the ASA CX module.
show asp table classify domain cxsc-auth-proxy	Shows the NP rules created for the authentication proxy for the ASA CX module.
show asp drop	Shows dropped packets. The following drop types are used: Frame Drops: • cxsc-bad-tlv-received—This occurs when ASA receives a packet from CXSC without a Policy ID TLV. This TLV must be present in non-control packets if it does not have the Standy Active bit set in the actions field. • cxsc-request—The frame was requested to be dropped by CXSC due a policy on CXSC whereby CXSC would set the actions to Deny Source, Deny Destination, or Deny Pkt. • cxsc-fail-close—The packet is dropped because the card is not up and the policy configured was 'fail-close' (rather than 'fail-open' which allows packets through even if the card was down). • cxsc-fail—The CXSC configuration was removed for an existing flow and we are not able to process it through CXSC it will be dropped. This should be very unlikely. • cxsc-malformed-packet—The packet from CXSC contains an invalid header. For instance, the header length may not be correct. Flow Drops: • cxsc-request—The CXSC requested to terminate the flow. The actions bit 0 is set. • reset-by-cxsc—The CXSC requested to terminate and reset the flow. The actions bit 1 is set. • cxsc-fail-close—The flow was terminated because the card is down and the configured policy was 'fail-close'.

Command	Purpose
show asp event dp-cp cxsc-msg	This output shows how many ASA CX module messages are on the dp-cp queue. Currently, only VPN queries from the ASA CX module are sent to dp-cp.
show conn	This command already shows if a connection is being forwarded to a module by displaying the ‘X - inspected by service module’ flag. Connections being forwarded to the ASA CX module will also display the ‘X’ flag.

Examples

The following is sample output from the **show asp table classify domain cxsc** command:

```
ciscoasa# show asp table classify domain cxsc
Input Table
in  id=0x7ffedb4acf40, priority=50, domain=cxsc, deny=false
    hits=15485658, user_data=0x7ffedb4ac840, cs_id=0x0, use_real_addr, flags=0x0,
protocol=0
    src ip/id=0.0.0.0, mask=0.0.0.0, port=0
    dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, dscp=0x0
    input_ifc=outside, output_ifc=any
in  id=0x7ffedb4ad4a0, priority=50, domain=cxsc, deny=false
    hits=992053, user_data=0x7ffedb4ac840, cs_id=0x0, use_real_addr, flags=0x0, protocol=0
    src ip/id=0.0.0.0, mask=0.0.0.0, port=0
    dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, dscp=0x0
    input_ifc=inside, output_ifc=any
in  id=0x7ffedb4ada00, priority=50, domain=cxsc, deny=false
    hits=0, user_data=0x7ffedb4ac840, cs_id=0x0, use_real_addr, flags=0x0, protocol=0
    src ip/id=0.0.0.0, mask=0.0.0.0, port=0
    dst ip/id=0.0.0.0, mask=0.0.0.0, port=0, dscp=0x0
    input_ifc=m, output_ifc=any

Output Table:

L2 - Output Table:

L2 - Input Table:

Last clearing of hits counters: Never
```

The following is sample output from the **show asp table classify domain cxsc-auth-proxy** command. For the first rule in the output, the destination “port=2000” is the auth-proxy port configured by the **cxsc auth-proxy port 2000** command, and the destination “ip/id=192.168.0.100” is the ASA interface IP address.

```
ciscoasa# show asp table classify domain cxsc-auth-proxy
Input Table
in  id=0x7ffed86cc470, priority=121, domain=cxsc-auth-proxy, deny=false
    hits=0, user_data=0x7ffed86ca220, cs_id=0x0, flags=0x0, protocol=6
    src ip/id=0.0.0.0, mask=0.0.0.0, port=0
    dst ip/id=192.168.0.100, mask=255.255.255.255, port=2000, dscp=0x0
    input_ifc=inside, output_ifc=identity
in  id=0x7ffed86cce20, priority=121, domain=cxsc-auth-proxy, deny=false
    hits=0, user_data=0x7ffed86ca220, cs_id=0x0, flags=0x0, protocol=6
    src ip/id=0.0.0.0, mask=0.0.0.0, port=0
    dst ip/id=2.2.2.2, mask=255.255.255.255, port=2000, dscp=0x0
    input_ifc=new2, output_ifc=identity
in  id=0x7ffed86cd7d0, priority=121, domain=cxsc-auth-proxy, deny=false
    hits=0, user_data=0x7ffed86ca220, cs_id=0x0, flags=0x0, protocol=6
    src ip/id=0.0.0.0, mask=0.0.0.0, port=0
    dst ip/id=172.23.58.52, mask=255.255.255.255, port=2000, dscp=0x0
    input_ifc=mgmt, output_ifc=identity
in  id=0x7ffed86caa80, priority=121, domain=cxsc-auth-proxy, deny=false
    hits=0, user_data=0x7ffed86ca220, cs_id=0x0, flags=0x0, protocol=6
    src ip/id=0.0.0.0, mask=0.0.0.0, port=0
    dst ip/id=192.168.5.172, mask=255.255.255.255, port=2000, dscp=0x0
    input_ifc=outside, output_ifc=identity
in  id=0x7ffed86cb3c0, priority=121, domain=cxsc-auth-proxy, deny=false
    hits=0, user_data=0x7ffed86ca220, cs_id=0x0, flags=0x0, protocol=6
    src ip/id=::/0, port=0
        dst ip/id=fe80::5675:d0ff:fe5b:1102/128, port=2000
    input_ifc=outside, output_ifc=identity
in  id=0x7ffed742be10, priority=121, domain=cxsc-auth-proxy, deny=false
    hits=0, user_data=0x7ffed86ca220, cs_id=0x0, flags=0x0, protocol=6
    src ip/id=::/0, port=0
    dst ip/id=1:1:1:1::10/128, port=2000
    input_ifc=outside, output_ifc=identity

Output Table:

L2 - Output Table:

L2 - Input Table:

Last clearing of hits counters: Never
```

The following is sample output from the **show asp drop** command. This output is just an example and lists all the possible reasons for a dropped frame or flow from the ASA CX module:

```
ciscoasa# show asp drop
Frame drop:
  CXSC Module received packet with bad TLV's (cxsc-bad-tlv-received)           2
  CXSC Module requested drop (cxsc-request)                                    1
  CXSC card is down (cxsc-fail-close)                                          1
  CXSC config removed for flow (cxsc-fail)                                     3
  CXSC Module received malformed packet (cxsc-malformed-packet)                1

Last clearing: 18:12:58 UTC May 11 2012 by enable_15

Flow drop:
  Flow terminated by CXSC (cxsc-request)                                       2
  Flow reset by CXSC (reset-by-cxsc)                                           1
  CXSC fail-close (cxsc-fail-close)                                            1

Last clearing: 18:12:58 UTC May 11 2012 by enable_15
```

The following is sample output from the **show asp event dp-cp cxsc-msg** command:

```
ciscoasa# show asp event dp-cp cxsc-msg
DP-CP EVENT QUEUE                  QUEUE-LEN  HIGH-WATER
Punt Event Queue                           0           5
Identity-Traffic Event Queue               0           0
General Event Queue                        0           4
Syslog Event Queue                         4          90
Non-Blocking Event Queue                   0           2
Midpath High Event Queue                   0          53
Midpath Norm Event Queue                8074        8288
SRTP Event Queue                           0           0
HA Event Queue                             0           0
Threat-Detection Event Queue               0           3
ARP Event Queue                            0        2048
IDFW Event Queue                           0           0
CXSC Event Queue                           0           1
EVENT-TYPE          ALLOC ALLOC-FAIL ENQUEUED ENQ-FAIL  RETIRED 15SEC-RATE
```

```
cxsc-msg                1          0        1        0        1          0
```

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

```
ciscoasa# show conn detail
0 in use, 105 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       B - initial SYN from outside, b - TCP state-bypass or nailed, C - CTIQBE media,
       D - DNS, d - dump, E - outside back connection, F - outside FIN, f - inside FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
       i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
       k - Skinny media, M - SMTP data, m - SIP media, n - GUP
       O - outbound data, P - inside back connection, p - Phone-proxy TFTP connection,
       q - SQL*Net data, R - outside acknowledged FIN,
       R - UDP SUNRPC, r - inside acknowledged FIN, S - awaiting inside SYN,
       s - awaiting outside SYN, T - SIP, t - SIP transient, U - up,
       V - VPN orphan, W - WAAS,
       X - inspected by service module

TCP outside 208.80.152.2:80 inside 192.168.1.20:59928, idle 0:00:10, bytes 79174, flags
XUIO
```

Capturing Module Traffic

To configure and view packet captures for the ASA CX module, enter one of the following commands:

Command	Purpose
capture *name* **interface asa_dataplane**	Captures packets between ASA CX module and the ASA on the backplane.
copy capture	Copies the capture file to a server.
show capture	Shows the capture at the ASA console.

Note Captured packets contain an additional AFBP header that your PCAP viewer might not understand; be sure to use the appropriate plugin to view these packets.

Troubleshooting the ASA CX Module

- Debugging the Module, page 30-30
- Problems with the Authentication Proxy, page 30-31

Debugging the Module

To enable ASA CX debugging, enter the following command:

<table>
<tr><th>Command</th><th>Purpose</th></tr>
<tr><td><b>debug cxsc</b> [<b>error</b> | <b>event</b> | <b>message</b>]</td><td>Enables debugs at error, event, or message level.</td></tr>
</table>

When you enable the authentication proxy, the ASA generates a debug messge when it sends an authentication proxy TLV to the ASA CX module, giving IP and port details:

```
DP CXSC Event: Sent Auth proxy tlv for adding Auth Proxy on interface: inside4.
DP CXSC Event: Sent Auth proxy tlv for adding Auth Proxy on interface: cx_inside.
DP CXSC Event: Sent Auth proxy tlv for adding Auth Proxy on interface: cx_outside.
```

When the interface IP address is changed, auth-proxy tlv updates are sent to the ASA CX module:

```
DP CXSC Event: Sent Auth proxy tlv for removing Auth Proxy for interface inside.
DP CXSC Event: Sent Auth proxy tlv for adding Auth Proxy on interface: inside.
```

When a flow is freed on the ASA, the ASA CX module is notified so it can clean up the flow:

```
DP CXSC Msg: Notifying CXSC that flow (handle:275233990) is being freed  for
192.168.18.5:2213 -> 10.166.255.18:80.
```

When the ASA CX module sends a redirect to a client to authenticate, and that redirect is sent to the ASA, the ASA sends it to the ASA CX module. In this example, 192.168.18.3 is the interface address and port 8888 is the authentication proxy port reserved on that interface for the authentication proxy feature:

```
DP CXSC Msg: rcvd authentication proxy data from 192.168.18.5:2214 -> 192.168.18.3:8888,
forwarding to cx
```

When a VPN connection is established on the ASA, and the ASA sends connection information to the ASA CX module:

```
CXSC Event:   Dumping attributes from the vpn session record
CXSC Event:   tunnel->Protocol:        17
CXSC Event:   tunnel->ClientVendor:    SSL VPN Client
CXSC Event:   tunnel->ClientVersion:   Cisco AnyConnect VPN Agent for Windows 2.4.1012
CXSC Event:   Sending VPN RA session data to CXSC
CXSC Event:   sess index:      0x3000
CXSC Event:   sess type id:    3
CXSC Event:   username:        devuser
CXSC Event:   domain:          CN=Users,DC=test,DC=priv
CXSC Event:   directory type:  1
CXSC Event:   login time:      1337124762
CXSC Event:   nac result:      0
CXSC Event:   posture token:
CXSC Event:   public IP:       172.23.34.108
CXSC Event:   assigned IP:     192.168.17.200
CXSC Event:   client OS id:    1
CXSC Event:   client OS:
CXSC Event:   client type:     Cisco AnyConnect VPN Agent for Windows 2.4.1012
CXSC Event:   anyconnect data: , len: 0
```

Problems with the Authentication Proxy

If you are having a problem using the authentication proxy feature, follow these steps to troubleshoot your configuration and connections:

1. Check your configurations.

- On the ASA, check the output of the **show asp table classify domain cxsc-auth-proxy** command and make sure there are rules installed and that they are correct.
- In PRSM, ensure the directory realm is created with the correct credentials and test the connection to make sure you can reach the authentication server; also ensure that a policy object or objects are configured for authentication.

2. Check the output of the **show service-policy cxsc** command to see if any packets were proxied.
3. Perform a packet capture on the backplane, and check to see if traffic is being redirected on the correct configured port. See the "Capturing Module Traffic" section on page 30-30. You can check the configured port using the **show running-config cxsc** command or the **show asp table classify domain cxsc-auth-proxy** command.

Note If you have a connection between hosts on two ASA interfaces, and the ASA CX service policy is only configured for one of the interfaces, then all traffic between these hosts is sent to the ASA CX module, including traffic orginiating on the non-ASA CX interface (the feature is bidirectional). However, the ASA only performs the authentication proxy on the interface to which the service policy is applied, because this feature is ingress-only.

Example 30-1 Make sure port 2000 is used consistently:

1. Check the authentication proxy port:

```
ciscoasa# show running-config cxsc
cxsc auth-proxy port 2000
```

2. Check the authentication proxy rules:

```
ciscoasa# show asp table classify domain cxsc-auth-proxy

Input Table
in  id=0x7ffed86cc470, priority=121, domain=cxsc-auth-proxy, deny=false
    hits=0, user_data=0x7ffed86ca220, cs_id=0x0, flags=0x0, protocol=6
    src ip/id=0.0.0.0, mask=0.0.0.0, port=0
    dst ip/id=192.168.0.100, mask=255.255.255.255, port=2000, dscp=0x0
    input_ifc=inside, output_ifc=identity
```

3. In the packet captures, the redirect request should be going to destination port 2000.

Configuration Examples for the ASA CX Module

The following example diverts all HTTP traffic to the ASA CX module, and blocks all HTTP traffic if the ASA CX module card fails for any reason:

```
ciscoasa(config)# access-list ASACX permit tcp any any eq port 80
ciscoasa(config)# class-map my-cx-class
ciscoasa(config-cmap)# match access-list ASACX
ciscoasa(config-cmap)# policy-map my-cx-policy
ciscoasa(config-pmap)# class my-cx-class
ciscoasa(config-pmap-c)# cxsc fail-close auth-proxy
ciscoasa(config-pmap-c)# service-policy my-cx-policy global
```

The following example diverts all IP traffic destined for the 10.1.1.0 network and the 10.2.1.0 network to the ASA CX module, and allows all traffic through if the ASA CX module fails for any reason.

```
ciscoasa(config)# access-list my-cx-acl permit ip any 10.1.1.0 255.255.255.0
ciscoasa(config)# access-list my-cx-acl2 permit ip any 10.2.1.0 255.255.255.0
ciscoasa(config)# class-map my-cx-class
ciscoasa(config-cmap)# match access-list my-cx-acl
ciscoasa(config)# class-map my-cx-class2
ciscoasa(config-cmap)# match access-list my-cx-acl2
ciscoasa(config-cmap)# policy-map my-cx-policy
ciscoasa(config-pmap)# class my-cx-class
ciscoasa(config-pmap-c)# cxsc fail-open auth-proxy
ciscoasa(config-pmap)# class my-cx-class2
ciscoasa(config-pmap-c)# cxsc fail-open auth-proxy
ciscoasa(config-pmap-c)# service-policy my-cx-policy interface outside
```

Feature History for the ASA CX Module

Table 30-2 lists each feature change and the platform release in which it was implemented.

Table 30-2 Feature History for the ASA CX Module

Feature Name	Platform Releases	Feature Information
ASA 5585-X with SSP-10 and -20 support for the ASA CX SSP-10 and -20	ASA 8.4(4.1) ASA CX 9.0(1)	The ASA CX module lets you enforce security based on the complete context of a situation. This context includes the identity of the user (who), the application or website that the user is trying to access (what), the origin of the access attempt (where), the time of the attempted access (when), and the properties of the device used for the access (how). With the ASA CX module, you can extract the full context of a flow and enforce granular policies such as permitting access to Facebook but denying access to games on Facebook or permitting finance employees access to a sensitive enterprise database but denying the same access to other employees. We introduced or modified the following commands: **capture**, **cxsc**, **cxsc auth-proxy**, **debug cxsc**, **hw-module module password-reset**, **hw-module module reload**, **hw-module module reset**, **hw-module module shutdown**, **session do setup host ip, session do get-config, session do password-reset, show asp table classify domain cxsc**, **show asp table classify domain cxsc-auth-proxy**, **show capture**, **show conn**, **show module**, **show service-policy**.
ASA 5512-X through ASA 5555-X support for the ASA CX SSP	ASA 9.1(1) ASA CX 9.1(1)	We introduced support for the ASA CX SSP software module for the ASA 5512-X, ASA 5515-X, ASA 5525-X, ASA 5545-X, and ASA 5555-X. We modified the following commands: **session cxsc**, **show module cxsc**, **sw-module cxsc**.

Table 30-2 Feature History for the ASA CX Module (continued)

Feature Name	Platform Releases	Feature Information
Monitor-only mode for demonstration purposes	ASA 9.1(2) ASA CX 9.1(2)	For demonstration purposes only, you can enable monitor-only mode for the service policy, which forwards a copy of traffic to the ASA CX module, while the original traffic remains unaffected. Another option for demonstration purposes is to configure a traffic-forwarding interface instead of a service policy in monitor-only mode. The traffic-forwarding interface sends all traffic directly to the ASA CX module, bypassing the ASA. We modified or introduced the following commands: **cxsc** {**fail-close** | **fail-open**} **monitor-only**, **traffic-forward cxsc monitor-only**.
NAT 64 support for the ASA CX module	ASA 9.1(2) ASA CX 9.1(2)	You can now use NAT 64 in conjunction with the ASA CX module. We did not modify any commands.
ASA 5585-X with SSP-40 and -60 support for the ASA CX SSP-40 and -60	ASA 9.1(3) ASA CX 9.2(1)	ASA CX SSP-40 and -60 modules can be used with the matching level ASA 5585-X with SSP-40 and -60. We did not modify any commands.

Table 30-2 Feature History for the ASA CX Module (continued)

Feature Name	Platform Releases	Feature Information
Multiple context mode support for the ASA CX module	ASA 9.1(3) ASA CX 9.2(1)	You can now configure ASA CX service policies per context on the ASA. **Note** Although you can configure per context ASA service policies, the ASA CX module itself (configured in PRSM) is a single context mode device; the context-specific traffic coming from the ASA is checked against the common ASA CX policy. We did not modify any commands.
Filtering packets captured on the ASA CX backplane	ASA 9.1(3) ASA CX 9.2(1)	You can now filter packets captured on the ASA CX backplane using the **match** or **access-list** keyword with the **capture interface asa_dataplane** command. Control traffic specific to the ASA CX module is not affected by the access-list or match filtering; the ASA captures all control traffic. In multiple context mode, configure the packet capture per context. Note that all control traffic in multiple context mode goes only to the system execution space. Because control traffic cannot be filtered using an access-list or match, these options are not available in the system execution space. We modified the following command: **capture interface asa_dataplane**.

Configuring the ASA IPS Module

This chapter describes how to configure the ASA IPS module. The ASA IPS module might be a hardware module or a software module, depending on your ASA model. For a list of supported ASA IPS modules per ASA model, see the *Cisco ASA Compatibility Matrix*:

http://www.cisco.com/en/US/docs/security/asa/compatibility/asamatrx.html

This chapter includes the following sections:

Information About the ASA IPS Module

The ASA IPS module runs advanced IPS software that provides proactive, full-featured intrusion prevention services to stop malicious traffic, including worms and network viruses, before they can affect your network. This section includes the following topics:

How the ASA IPS Module Works with the ASA

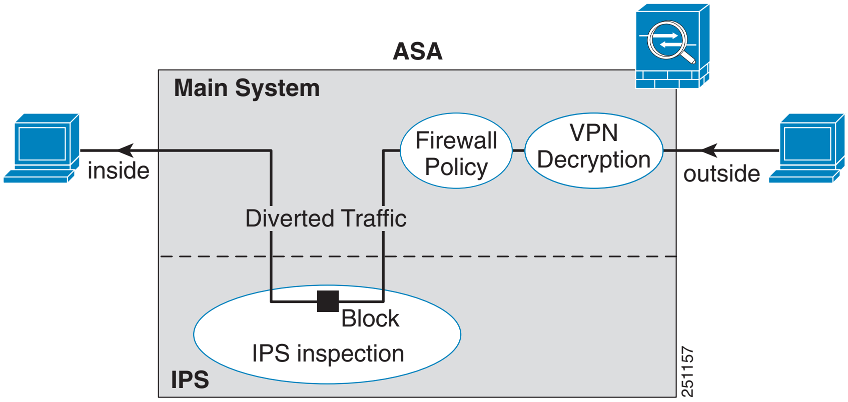
The ASA IPS module runs a separate application from the ASA. The ASA IPS module might include an external management interface so you can connect to the ASA IPS module directly; if it does not have a management interface, you can connect to the ASA IPS module through the ASA interface. The ASA IPS SSP on the ASA 5585-X includes data interfaces; these interfaces provide additional port-density for the ASA. However, the overall through-put of the ASA is not increased.

Traffic goes through the firewall checks before being forwarded to the ASA IPS module. When you identify traffic for IPS inspection on the ASA, traffic flows through the ASA and the ASA IPS module as follows. **Note**: This example is for "inline mode." See the "Operating Modes" section on page 31-3 for information about "promiscuous mode," where the ASA only sends a copy of the traffic to the ASA IPS module.

1. Traffic enters the ASA.
2. Incoming VPN traffic is decrypted.
3. Firewall policies are applied.
4. Traffic is sent to the ASA IPS module.
5. The ASA IPS module applies its security policy to the traffic, and takes appropriate actions.
6. Valid traffic is sent back to the ASA; the ASA IPS module might block some traffic according to its security policy, and that traffic is not passed on.
7. Outgoing VPN traffic is encrypted.
8. Traffic exits the ASA.

Figure 31-1 shows the traffic flow when running the ASA IPS module in inline mode. In this example, the ASA IPS module automatically blocks traffic that it identified as an attack. All other traffic is forwarded through the ASA.

Figure 31-1 ASA IPS module Traffic Flow in the ASA: Inline Mode

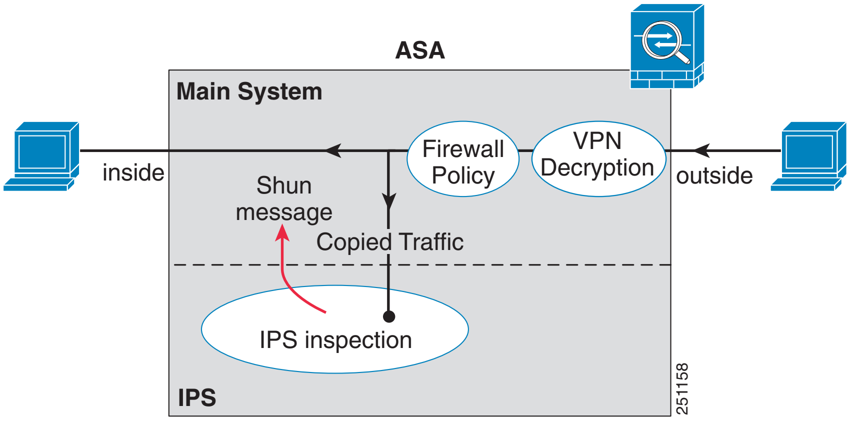


Operating Modes

You can send traffic to the ASA IPS module using one of the following modes:

- Inline mode—This mode places the ASA IPS module directly in the traffic flow (see Figure 31-1). No traffic that you identified for IPS inspection can continue through the ASA without first passing through, and being inspected by, the ASA IPS module. This mode is the most secure because every packet that you identify for inspection is analyzed before being allowed through. Also, the ASA IPS module can implement a blocking policy on a packet-by-packet basis. This mode, however, can affect throughput.
- Promiscuous mode—This mode sends a duplicate stream of traffic to the ASA IPS module. This mode is less secure, but has little impact on traffic throughput. Unlike inline mode, in promiscuous mode the ASA IPS module can only block traffic by instructing the ASA to shun the traffic or by resetting a connection on the ASA. Also, while the ASA IPS module is analyzing the traffic, a small amount of traffic might pass through the ASA before the ASA IPS module can shun it. Figure 31-2 shows the ASA IPS module in promiscuous mode. In this example, the ASA IPS module sends a shun message to the ASA for traffic it identified as a threat.

Figure 31-2 ASA IPS module Traffic Flow in the ASA: Promiscuous Mode



Using Virtual Sensors (ASA 5510 and Higher)

The ASA IPS module running IPS software Version 6.0 and later can run multiple virtual sensors, which means you can configure multiple security policies on the ASA IPS module. You can assign each ASA security context or single mode ASA to one or more virtual sensors, or you can assign multiple security contexts to the same virtual sensor. See the IPS documentation for more information about virtual sensors, including the maximum number of sensors supported.

Figure 31-3 shows one security context paired with one virtual sensor (in inline mode), while two security contexts share the same virtual sensor.

Figure 31-3 Security Contexts and Virtual Sensors

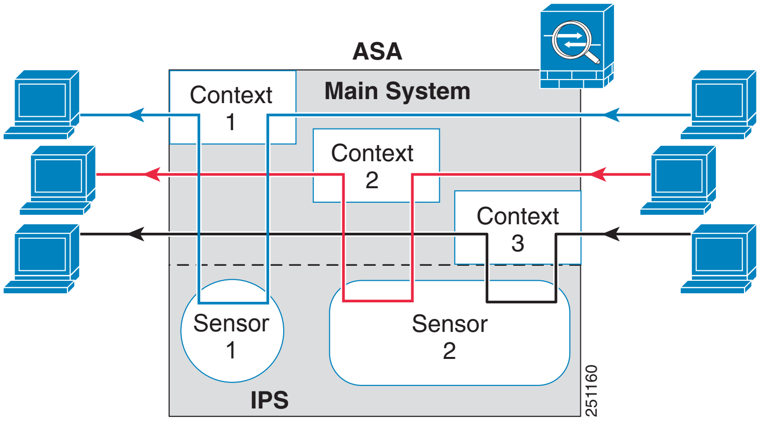
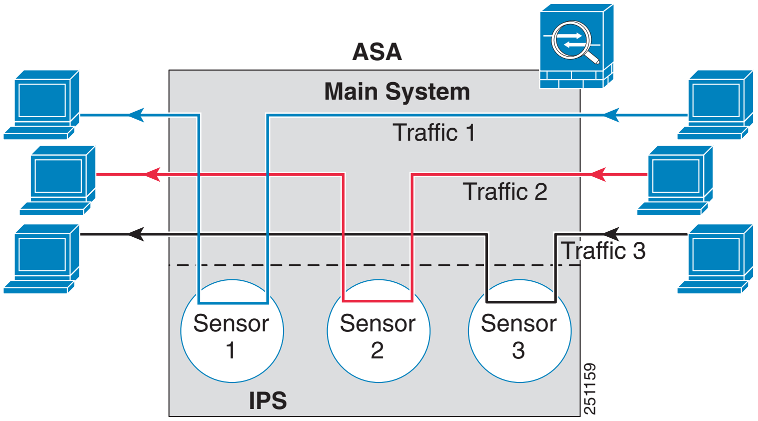


Figure 31-4 shows a single mode ASA paired with multiple virtual sensors (in inline mode); each defined traffic flow goes to a different sensor.

Figure 31-4 Single Mode ASA with Multiple Virtual Sensors



Information About Management Access

You can manage the IPS application using the following methods:

- Sessioning to the module from the ASA—If you have CLI access to the ASA, then you can session to the module and access the module CLI. See the "Sessioning to the Module from the ASA" section on page 31-11.
- Connecting to the IPS management interface using ASDM or SSH—After you launch ASDM from the ASA, your management station connects to the module management interface to configure the IPS application. For SSH, you can access the module CLI directly on the module management interface. (Telnet access requires additional configuration in the module application). The module management interface can also be used for sending syslog messages or allowing updates for the module application, such as signature database updates.

See the following information about the management interface:

- ASA 5510, ASA 5520, ASA 5540, ASA 5580, ASA 5585-X—The IPS management interface is a separate external Gigabit Ethernet interface.
- ASA 5512-X, ASA 5515-X, ASA 5525-X, ASA 5545-X, ASA 5555-X—These models run the ASA IPS module as a software module. The IPS management interface shares the Management 0/0 interface with the ASA. Separate MAC addresses and IP addresses are supported for the ASA and ASA IPS module. You must perform configuration of the IPS IP address within the IPS operating system (using the CLI or ASDM). However, physical characteristics (such as enabling the interface) are configured on the ASA. You can remove the ASA interface configuration (specifically the interface name) to dedicate this interface as an IPS-only interface. This interface is management-only.
- ASA 5505—You can use an ASA VLAN to allow access to an internal management IP address over the backplane.

Licensing Requirements for the ASA IPS module

The following table shows the licensing requirements for this feature:

Model	License Requirement
ASA 5512-X, ASA 5515-X, ASA 5525-X, ASA 5545-X, ASA 5555-X	IPS Module License. **Note** The IPS module license lets you run the IPS software module on the ASA. You must also purchase a separate IPS signature subscription; for failover, purchase a subscription for each unit. To obtain IPS signature support, you must purchase the ASA with IPS pre-installed (the part number must include “IPS”). The combined failover cluster license does not let you pair non-IPS and IPS units. For example, if you buy the IPS version of the ASA 5515-X (part number ASA5515-IPS-K9) and try to make a failover pair with a non-IPS version (part number ASA5515-K9), then you will not be able to obtain IPS signature updates for the ASA5515-K9 unit, even though it has an IPS module license inherited from the other unit.
All other models	Base License.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

The ASA 5505 does not support multiple context mode, so multiple context features, such as virtual sensors, are not supported on the AIP SSC.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

Model Guidelines

- See the *Cisco ASA Compatibility Matrix* for information about which models support which modules:

http://www.cisco.com/en/US/docs/security/asa/compatibility/asamatrx.html

- The ASA 5505 does not support multiple context mode, so multiple context features, such as virtual sensors, are not supported on the AIP SSC.
- The ASA IPS module for the ASA 5510 and higher supports higher performance requirements, while the ASA IPS module for the ASA 5505 is designed for a small office installation. The following features are not supported for the ASA 5505:
  - Virtual sensors
  - Anomaly detection
  - Unretirement of default retired signatures

Additional Guidelines

- The total throughput for the ASA plus the IPS module is lower than ASA throughput alone.
  - ASA 5512-X through ASA 5555-X—See http://www.cisco.com/en/US/prod/collateral/vpndevc/ps6032/ps6094/ps6120/qa_c67-700608.html
  - ASA 5585-X—See http://www.cisco.com/en/US/prod/collateral/vpndevc/ps6032/ps6094/ps6120/qa_c67-617018.html
  - ASA 5505 through ASA 5540—See http://www.cisco.com/en/US/prod/collateral/vpndevc/ps6032/ps6094/ps6120/product_data_sheet0900aecd802930c5.html
- You cannot change the software type installed on the module; if you purchase an ASA IPS module, you cannot later install other software on it.

Default Settings

Table 31-1 lists the default settings for the ASA IPS module.

Table 31-1 Default Network Parameters

Parameters	Default
Management VLAN (ASA 5505 only)	VLAN 1
Management IP address	192.168.1.2/24
Gateway	192.168.1.1/24 (the default ASA management IP address)
Username	cisco
Password	cisco

Note The default management IP address on the ASA is 192.168.1.1/24.

Configuring the ASA IPS module

This section describes how to configure the ASA IPS module and includes the following topics:

Task Flow for the ASA IPS Module

Configuring the ASA IPS module is a process that includes configuration of the IPS security policy on the ASA IPS module and then configuration of the ASA to send traffic to the ASA IPS module. To configure the ASA IPS module, perform the following steps:

Step 1 Cable the ASA IPS management interface. See the "Connecting the ASA IPS Management Interface" section on page 31-8.

Step 2 Session to the module. Access the IPS CLI over the backplane.See the "Sessioning to the Module from the ASA" section on page 31-11.

Step 3 (ASA 5512-X through ASA 5555-X; may be required) Install the software module. See the "(ASA 5512-X through ASA 5555-X) Booting the Software Module" section on page 31-11.

Step 4 Depending on your ASA model:

- (ASA 5510 and higher) Configure basic network settings for the IPS module. See the "(ASA 5510 and Higher) Configuring Basic Network Settings" section on page 31-13.
- (ASA 5505) Configure the management VLAN and IP address for the IPS module. See the "(ASA 5505) Configuring Basic Network Settings" section on page 31-13.

Step 5 On the module, configure the inspection and protection policy, which determines how to inspect traffic and what to do when an intrusion is detected. See the "Configuring the Security Policy on the ASA IPS Module" section on page 31-15.

Step 6 (ASA 5510 and higher, optional) On the ASA in multiple context mode, specify which IPS virtual sensors are available for each context (if you configured virtual sensors). See the "Assigning Virtual Sensors to a Security Context (ASA 5510 and Higher)" section on page 31-16.

Step 7 On the ASA, identify traffic to divert to the ASA IPS module. See the "Diverting Traffic to the ASA IPS module" section on page 31-18.

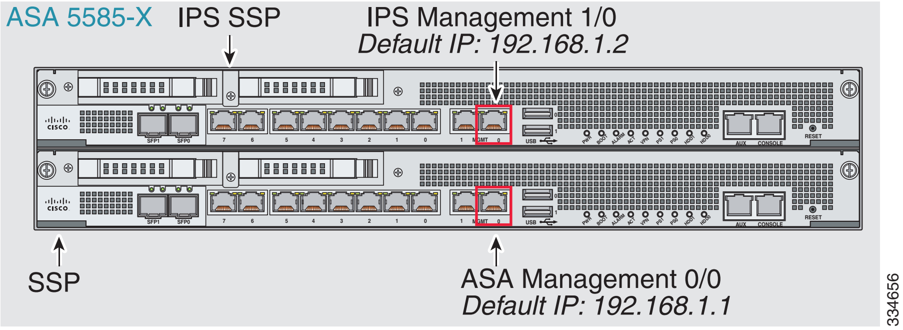
Connecting the ASA IPS Management Interface

In addition to providing management access to the IPS module, the IPS management interface needs access to an HTTP proxy server or a DNS server and the Internet so it can download global correlation, signature updates, and license requests. This section describes recommended network configurations. Your network may differ.

- ASA 5510, ASA 5520, ASA 5540, ASA 5580, ASA 5585-X (Hardware Module), page 31-8
- ASA 5512-X through ASA 5555-X (Software Module), page 31-9
- ASA 5505, page 31-10

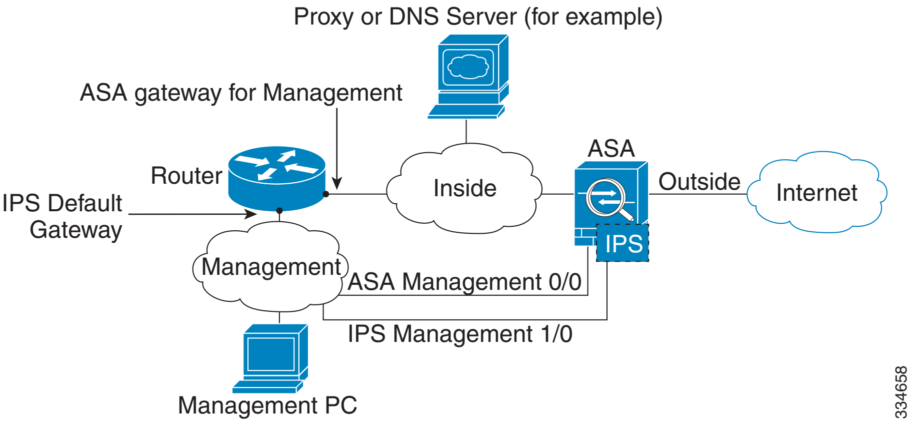
ASA 5510, ASA 5520, ASA 5540, ASA 5580, ASA 5585-X (Hardware Module)

The IPS module includes a separate management interface from the ASA.



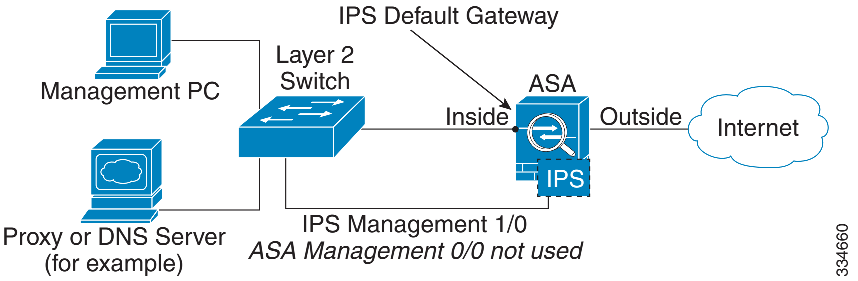
If you have an inside router

If you have an inside router, you can route between the management network, which can include both the ASA Management 0/0 and IPS Management 1/0 interfaces, and the ASA inside network. Be sure to also add a route on the ASA to reach the Management network through the inside router.



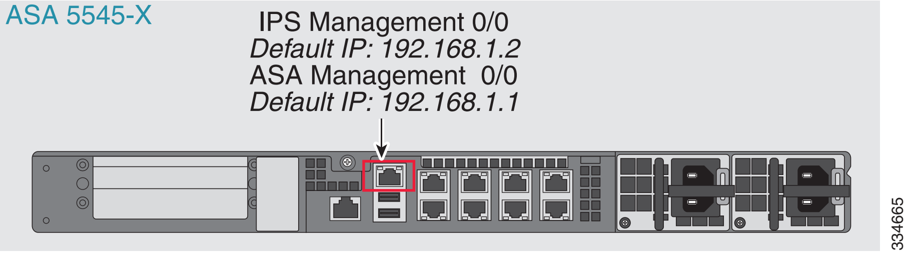
If you do not have an inside router

If you have only one inside network, then you cannot also have a separate management network, which would require an inside router to route between the networks. In this case, you can manage the ASA from the inside interface instead of the Management 0/0 interface. Because the IPS module is a separate device from the ASA, you can configure the IPS Management 1/0 address to be on the same network as the inside interface.



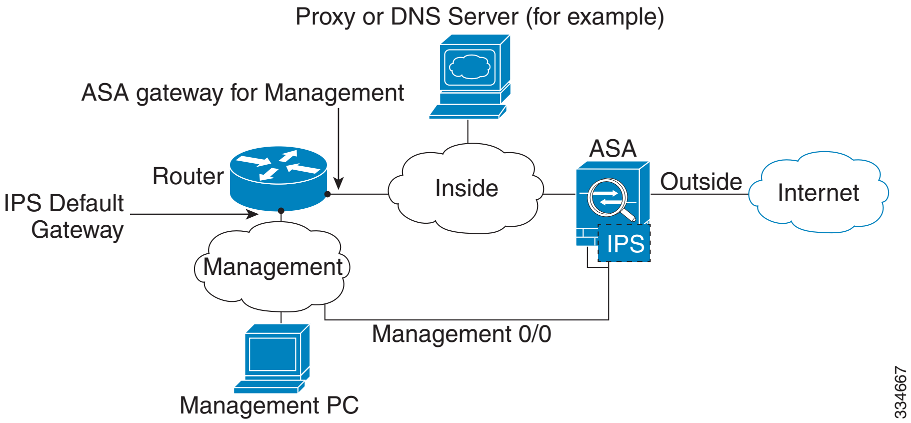
ASA 5512-X through ASA 5555-X (Software Module)

These models run the IPS module as a software module, and the IPS management interface shares the Management 0/0 interface with the ASA.



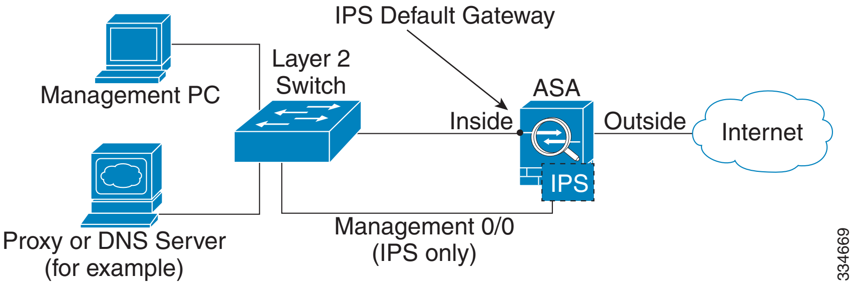
If you have an inside router

If you have an inside router, you can route between the Management 0/0 network, which includes both the ASA and IPS management IP addresses, and the inside network. Be sure to also add a route on the ASA to reach the Management network through the inside router.



If you do not have an inside router

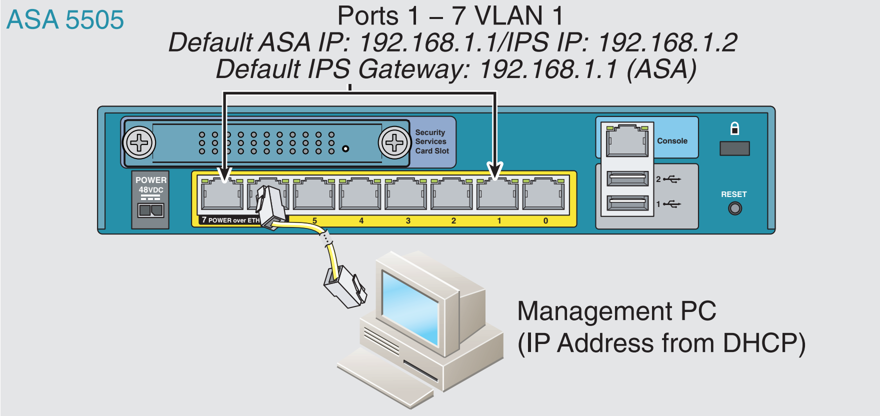
If you have only one inside network, then you cannot also have a separate management network. In this case, you can manage the ASA from the inside interface instead of the Management 0/0 interface. If you remove the ASA-configured name from the Management 0/0 interface, you can still configure the IPS IP address for that interface. Because the IPS module is essentially a separate device from the ASA, you *can* configure the IPS management address to be on the same network as the inside interface.



Note You must remove the ASA-configured name for Management 0/0; if it is configured on the ASA, then the IPS address must be on the same network as the ASA, and that excludes any networks already configured on other ASA interfaces. If the name is not configured, then the IPS address can be on any network, for example, the ASA inside network.

ASA 5505

The ASA 5505 does not have a dedicated management interface. You must use an ASA VLAN to access an internal management IP address over the backplane. Connect the management PC to one of the following ports: Ethernet 0/1 through 0/7, which are assigned to VLAN 1.


ASA 5505
Ports 1 – 7 VLAN 1
Default ASA IP: 192.168.1.1/IPS IP: 192.168.1.2
Default IPS Gateway: 192.168.1.1 (ASA)
Security Services Card Slot
Console
POWER 48VDC
7 POWER over ETH
5
4
3
2
1
0
RESET
Management PC
(IP Address from DHCP)

What to Do Next

- (ASA 5510 and higher) Configure basic network settings. See the "(ASA 5510 and Higher) Configuring Basic Network Settings" section on page 31-13.
- (ASA 5505) Configure management interface settings. See the "(ASA 5505) Configuring Basic Network Settings" section on page 31-13.

Sessioning to the Module from the ASA

To access the IPS module CLI from the ASA, you can session from the ASA. For software modules, you can either session to the module (using Telnet) or create a virtual console session. A console session might be useful if the control plane is down and you cannot establish a Telnet session.

Detailed Steps

Command	Purpose

Telnet session.
For a hardware module (for example, the ASA 5585-X):
`session 1`
For a software module (for example, the ASA 5545-X):
`session ips`
Example:
`ciscoasa# session 1`
`Opening command session with slot 1.`
`Connected to slot 1. Escape character`
`sequence is 'CTRL-^X'.`
`sensor login: cisco`
`Password: cisco`
	Accesses the module using Telnet. You are prompted for the username and password. The default username is **cisco**, and the default password is **cisco**.
	Note The first time you log in to the module, you are prompted to change the default password. Passwords must be at least eight characters long and cannot be a word in the dictionary.

Console session (software module only).
`session ips console`
Example:
`ciscoasa# session ips console`
`Establishing console session with slot 1`
`Opening console session with module ips.`
`Connected to module ips. Escape character`
`sequence is 'CTRL-SHIFT-6 then x'.`
`sensor login: cisco`
`Password: cisco`
	Accesses the module console. You are prompted for the username and password. The default username is **cisco**, and the default password is **cisco**.
	Note Do not use this command in conjunction with a terminal server where **Ctrl-Shift-6, x** is the escape sequence to return to the terminal server prompt. **Ctrl-Shift-6, x** is also the sequence to escape the IPS console and return to the ASA prompt. Therefore, if you try to exit the IPS console in this situation, you instead exit all the way to the terminal server prompt. If you reconnect the terminal server to the ASA, the IPS console session is still active; you can never exit to the ASA prompt. You must use a direct serial connection to return the console to the ASA prompt.
	Use the **session ips** command instead.

(ASA 5512-X through ASA 5555-X) Booting the Software Module

Your ASA typically ships with IPS module software present on Disk0. If the module is not running, or if you are adding the IPS module to an existing ASA, you must boot the module software. If you are unsure if the module is running, you will not be able to session it.

Detailed Steps

Step 1 Do one of the following:

- New ASA with IPS pre-installed—To view the IPS module software filename in flash memory, enter:.

  ```
  ciscoasa# dir disk0:
  ```

  For example, look for a filename like IPS-SSP_5512-K9-sys-1.1-a-7.1-4-E4.aip. Note the filename; you will need this filename later in the procedure.

- Existing ASA with new IPS installation—Download the IPS software from Cisco.com to a TFTP server. If you have a Cisco.com login, you can obtain the software from the following website:

  http://www.cisco.com/cisco/software/navigator.html?mdfid=282164240

  Copy the software to the ASA:

  ```
  ciscoasa# copy tftp://server/file_path disk0:/file_path
  ```

  For other download server types, see Chapter 46, "Managing Software and Configurations," in the general operations configuration guide.

  Note the filename; you will need this filename later in the procedure.

Step 2 To set the IPS module software location in disk0, enter the following command:

```
ciscoasa# sw-module module ips recover configure image disk0:file_path
```

For example, using the filename in the example in Step 1, enter:

```
ciscoasa# sw-module module ips recover configure image
disk0:IPS-SSP_5512-K9-sys-1.1-a-7.1-4-E4.aip
```

Step 3 To install and load the IPS module software, enter the following command:

```
ciscoasa# sw-module module ips recover boot
```

Step 4 To check the progress of the image transfer and module restart process, enter the following command:

```
ciscoasa# show module ips details
```

The Status field in the output indicates the operational status of the module. A module operating normally shows a status of "Up." While the ASA transfers an application image to the module, the Status field in the output reads "Recover." When the ASA completes the image transfer and restarts the module, the newly transferred image is running.

Configuring Basic IPS Module Network Settings

- (ASA 5510 and Higher) Configuring Basic Network Settings, page 31-13
- (ASA 5505) Configuring Basic Network Settings, page 31-13

(ASA 5510 and Higher) Configuring Basic Network Settings

Session to the module from the ASA and configure basic settings using the **setup** command.

Note (ASA 5512-X through ASA 5555-X) If you cannot session to the module, then the IPS module is not running. See the "(ASA 5512-X through ASA 5555-X) Booting the Software Module" section on page 31-11, and then repeat this procedure after you install the module.

Detailed Steps

	Command	Purpose
Step 1	Session to the IPS module according to the "Sessioning to the Module from the ASA" section on page 31-11.	

Step 2	`setup`
	Example:
	`sensor# setup`
		Runs the setup utility for initial configuration of the ASA IPS module. You are prompted for basic settings. For the default gateway, specify the IP address of the upstream router. See the "Connecting the ASA IPS Management Interface" section on page 31-8 to understand the requirements for your network. The default setting of the ASA management IP address will not work.

(ASA 5505) Configuring Basic Network Settings

An ASA IPS module on the ASA 5505 does not have any external interfaces. You can configure a VLAN to allow access to an internal IPS management IP address over the backplane. By default, VLAN 1 is enabled for IPS management. You can only assign one VLAN as the management VLAN. This section describes how to change the management VLAN and IP address if you do not want to use the default, and how to set other required network parameters.

Note Perform this configuration on the ASA 5505, not on the ASA IPS module.

Prerequisites

When you change the IPS VLAN and management address from the default, be sure to also configure the matching ASA VLAN and switch port(s) according to the procedures listed in Chapter 12, "Starting Interface Configuration (ASA 5505)," in the general operations configuration guide. You must define and configure the VLAN for the ASA so the IPS management interface is accessible on the network.

Restrictions

Do not configure NAT for the management address if you intend to access it using ASDM. For initial setup with ASDM, you need to access the real address. After initial setup (where you set the password on the ASA IPS module), you can configure NAT and supply ASDM with the translated address for accessing the ASA IPS module.

Detailed Steps

	Command	Purpose

Step 1	**`interface vlan`** *`number`*
	Example:
	`ciscoasa(config)# interface vlan 1`
		Specifies the current management VLAN for which you want to disable IPS management. By default, this is VLAN 1.

Step 2	**`no allow-ssc-mgmt`**
	Example:
	`ciscoasa(config-if)# no allow-ssc-mgmt`
		Disables IPS management for the old VLAN so that you can enable it for a different VLAN.

Step 3	**`interface vlan`** *`number`*
	Example:
	`ciscoasa(config)# interface vlan 20`
		Specifies the VLAN you want to use as the new IPS management VLAN.

Step 4	**`allow-ssc-mgmt`**
	Example:
	`ciscoasa(config-if)# allow-ssc-mgmt`
		Sets this interface as the IPS management interface.

	Command	Purpose
Step 5	**hw-module module 1 ip** *ip_address netmask gateway* **Example:** `ciscoasa# hw-module module 1 ip 10.1.1.2 255.255.255.0 10.1.1.1`	Configures the management IP address for the ASA IPS module. Make sure this address is on the same subnet as the ASA VLAN IP address. For example, if you assigned 10.1.1.1 to the VLAN for the ASA, then assign another address on that network, such as 10.1.1.2, for the IPS management address. Set the gateway to be the ASA IP address for the management VLAN. By default, this IP address is 192.168.1.1. **Note** These settings are written to the IPS application configuration, not the ASA configuration. You can view these settings from the ASA using the **show module details** command. You can alternatively use the IPS application **setup** command to configure this setting from the IPS CLI.
Step 6	**hw-module module 1 allow-ip** *ip_address netmask* **Example:** `ciscoasa# hw-module module 1 allow-ip 10.1.1.30 255.255.255.0`	Sets the hosts that are allowed to access the management IP address. **Note** These settings are written to the IPS application configuration, not the ASA configuration. You can view these settings from the ASA using the **show module details** command. You can alternatively use the IPS application **setup** command to configure this setting from the IPS CLI.

Examples

The following example configures VLAN 20 as the IPS management VLAN. Only the host at 10.1.1.30 can access the IPS management IP address. VLAN 20 is assigned to switch port Ethernet 0/0. When you connect to ASDM on ASA interface 10.1.1.1, ASDM then accesses the IPS on 10.1.1.2.

```
ciscoasa(config)# interface vlan 1
ciscoasa(config-if)# no allow-ssc-mgmt

ciscoasa(config-if)# interface vlan 20
ciscoasa(config-if)# nameif management
ciscoasa(config-if)# ip address 10.1.1.1 255.255.255.0
ciscoasa(config-if)# security-level 100
ciscoasa(config-if)# allow-ssc-mgmt
ciscoasa(config-if)# no shutdown

ciscoasa(config-if)# hw-module module 1 ip 10.1.1.2 255.255.255.0 10.1.1.1
ciscoasa(config)# hw-module module 1 allow-ip 10.1.1.30 255.255.255.255

ciscoasa(config)# interface ethernet 0/0
ciscoasa(config-if)# switchport access vlan 20
ciscoasa(config-if)# no shutdown
```

Configuring the Security Policy on the ASA IPS Module

This section describes how to configure the ASA IPS module application.

Detailed Steps

Step 1 Access the ASA IPS module CLI using one of the following methods:

- Session from the ASA to the ASA IPS module. See the "Sessioning to the Module from the ASA" section on page 31-11.
- Connect to the IPS management interface using SSH. If you did not change it, the default management IP address is 192.168.1.2. The default username is **cisco**, and the default password is **cisco**. See the "Information About Management Access" section on page 31-4 for more information about the management interface.

Step 2 Configure the IPS security policy according to the IPS documentation.

To access all documents related to IPS, go to:
http://www.cisco.com/en/US/products/hw/vpndevc/ps4077/products_documentation_roadmaps_list.html

Step 3 (ASA 5510 and higher) If you configure virtual sensors, you identify one of the sensors as the default. If the ASA does not specify a virtual sensor name in its configuration, the default sensor is used.

Step 4 When you are done configuring the ASA IPS module, exit the IPS software by entering the following command:

```
sensor# exit
```

If you sessioned to the ASA IPS module from the ASA, you return to the ASA prompt.

What to Do Next

- For the ASA in multiple context mode, see the "Assigning Virtual Sensors to a Security Context (ASA 5510 and Higher)" section on page 31-16.
- For the ASA in single context mode, see the "Diverting Traffic to the ASA IPS module" section on page 31-18.

Assigning Virtual Sensors to a Security Context (ASA 5510 and Higher)

If the ASA is in multiple context mode, then you can assign one or more IPS virtual sensors to each context. Then, when you configure the context to send traffic to the ASA IPS module, you can specify a sensor that is assigned to the context; you cannot specify a sensor that you did not assign to the context. If you do not assign any sensors to a context, then the default sensor configured on the ASA IPS module is used. You can assign the same sensor to multiple contexts.

Note You do not need to be in multiple context mode to use virtual sensors; you can be in single mode and use different sensors for different traffic flows.

Prerequisites

For more information about configuring contexts, see the "Configuring Multiple Contexts" section on page 8-15 in the general operations configuration guide.

Detailed Steps

	Command	Purpose

Step 1	**`context`** *`name`*
	Example:
	`ciscoasa(config)# context admin`
	`ciscoasa(config-ctx)#`
		Identifies the context you want to configure. Enter this command in the system execution space.

Step 2	**`allocate-ips`** *`sensor_name`* `[`*`mapped_name`*`]` `[`**`default`**`]`
	Example:
	`ciscoasa(config-ctx)# allocate-ips`
	`sensor1 highsec`
		Enter this command for each sensor you want to assign to the context.
		The *sensor _name* argument is the sensor name configured on the ASA IPS module. To view the sensors that are configured on the ASA IPS module, enter **allocate-ips ?**. All available sensors are listed. You can also enter the **show ips** command. In the system execution space, the **show ips** command lists all available sensors; if you enter it in the context, it shows the sensors you already assigned to the context. If you specify a sensor name that does not yet exist on the ASA IPS module, you get an error, but the **allocate-ips** command is entered as is. Until you create a sensor of that name on the ASA IPS module, the context assumes the sensor is down.
		Use the *mapped_name* argument as an alias for the sensor name that can be used within the context instead of the actual sensor name. If you do not specify a mapped name, the sensor name is used within the context. For security purposes, you might not want the context administrator to know which sensors are being used by the context. Or you might want to genericize the context configuration. For example, if you want all contexts to use sensors called "sensor1" and "sensor2," then you can map the "highsec" and "lowsec" sensors to sensor1 and sensor2 in context A, but map the "medsec" and "lowsec" sensors to sensor1 and sensor2 in context B.
		The **default** keyword sets one sensor per context as the default sensor; if the context configuration does not specify a sensor name, the context uses this default sensor. You can only configure one default sensor per context. If you want to change the default sensor, enter the **no allocate-ips** *sensor_name* command to remove the current default sensor before you allocate a new default sensor. If you do not specify a sensor as the default, and the context configuration does not include a sensor name, then traffic uses the default sensor as specified on the ASA IPS module.

Step 3	**`changeto context`** *`context_name`*
	Example:
	`ciscoasa# changeto context customer1`
	`ciscoasa/customer1#`
		Changes to the context so you can configure the IPS security policy as described in "Diverting Traffic to the ASA IPS module" section on page 31-18.

Examples

The following example assigns sensor1 and sensor2 to context A, and sensor1 and sensor3 to context B. Both contexts map the sensor names to “ips1” and “ips2.” In context A, sensor1 is set as the default sensor, but in context B, no default is set so the default that is configured on the ASA IPS module is used.

```
ciscoasa(config-ctx)# context A
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/0.100 int1
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/0.102 int2
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/0.110-gigabitethernet0/0.115
int3-int8
ciscoasa(config-ctx)# allocate-ips sensor1 ips1 default
ciscoasa(config-ctx)# allocate-ips sensor2 ips2
ciscoasa(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/test.cfg
ciscoasa(config-ctx)# member gold

ciscoasa(config-ctx)# context sample
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.200 int1
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.212 int2
ciscoasa(config-ctx)# allocate-interface gigabitethernet0/1.230-gigabitethernet0/1.235
int3-int8
ciscoasa(config-ctx)# allocate-ips sensor1 ips1
ciscoasa(config-ctx)# allocate-ips sensor3 ips2
ciscoasa(config-ctx)# config-url ftp://user1:passw0rd@10.1.1.1/configlets/sample.cfg
ciscoasa(config-ctx)# member silver

ciscoasa(config-ctx)# changeto context A
...
```

What to Do Next

Change to each context to configure the IPS security policy as described in “Diverting Traffic to the ASA IPS module” section on page 31-18.

Diverting Traffic to the ASA IPS module

This section identifies traffic to divert from the ASA to the ASA IPS module.

Prerequisites

In multiple context mode, perform these steps in each context execution space. To change to a context, enter the **changeto context** *context_name* command.

Detailed Steps

	Command	Purpose

Step 1	**class-map** *name*
	Example:
	`ciscoasa(config)# class-map ips_class`
		Creates a class map to identify the traffic for which you want to send to the ASA IPS module.
		If you want to send multiple traffic classes to the ASA IPS module, you can create multiple class maps for use in the security policy.

Step 2	**match** *parameter* **Example:** `ciscoasa(config-cmap)# match access-list ips_traffic`	Specifies the traffic in the class map. See the "Identifying Traffic (Layer 3/4 Class Maps)" section on page 1-12 for more information.
Step 3	**policy-map** *name* **Example:** `ciscoasa(config)# policy-map ips_policy`	Adds or edits a policy map that sets the actions to take with the class map traffic.

Step 4	**class** *name*
	Example:
	`ciscoasa(config-pmap)# class ips_class`
		Identifies the class map you created in Step 1.

	Command	Purpose
Step 5	**ips** {**inline** | **promiscuous**} {**fail-close** | **fail-open**} [**sensor** {*sensor_name* | *mapped_name*}] **Example:** `ciscoasa(config-pmap-c)# ips promiscuous fail-close`	Specifies that the traffic should be sent to the ASA IPS module. The **inline** and **promiscuous** keywords control the operating mode of the ASA IPS module. See the "Operating Modes" section on page 31-3 for more details. The **fail-close** keyword sets the ASA to block all traffic if the ASA IPS module is unavailable. The **fail-open** keyword sets the ASA to allow all traffic through, uninspected, if the ASA IPS module is unavailable. (ASA 5510 and higher) If you use virtual sensors, you can specify a sensor name using the **sensor** *sensor_name* argument. To see available sensor names, enter the **ips** {**inline** | **promiscuous**} {**fail-close** | **fail-open**} **sensor ?** command. Available sensors are listed. You can also use the **show ips** command. If you use multiple context mode on the ASA, you can only specify sensors that you assigned to the context (see the "Assigning Virtual Sensors to a Security Context (ASA 5510 and Higher)" section on page 31-16). Use the *mapped_name* if configured in the context. If you do not specify a sensor name, then the traffic uses the default sensor. In multiple context mode, you can specify a default sensor for the context. In single mode or if you do not specify a default sensor in multiple mode, the traffic uses the default sensor that is set on the ASA IPS module. If you enter a name that does not yet exist on the ASA IPS module, you get an error, and the command is rejected.

Step 6	(Optional)
	class *name2*
	Example:
	`ciscoasa(config-pmap)# class ips_class2`
		If you created multiple class maps for IPS traffic, you can specify another class for the policy.
		See the "Feature Matching Within a Service Policy" section on page 1-3 for detailed information about how the order of classes matters within a policy map. Traffic cannot match more than one class map for the same action type; so if you want network A to go to sensorA, but want all other traffic to go to sensorB, then you need to enter the **class** command for network A before you enter the **class** command for all traffic; otherwise all traffic (including network A) will match the first **class** command, and will be sent to sensorB.

	Command	Purpose
Step 7	(Optional) **ips** {**inline** | **promiscuous**} {**fail-close** | **fail-open**} [**sensor** {*sensor_name* | *mapped_name*}] **Example:** `ciscoasa(config-pmap-c)# ips promiscuous fail-close`	Specifies that the second class of traffic should be sent to the ASA IPS module. Add as many classes as desired by repeating these steps.

Step 8	**service-policy** *policymap_name* {**global** | **interface** *interface_name*}
	Example:
	`ciscoasa(config)# service-policy tcp_bypass_policy outside`
		Activates the policy map on one or more interfaces. **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

Managing the ASA IPS module

This section includes procedures that help you recover or troubleshoot the module and includes the following topics:

- Installing and Booting an Image on the Module, page 31-21
- Shutting Down the Module, page 31-23
- Uninstalling a Software Module Image, page 31-23
- Resetting the Password, page 31-24
- Reloading or Resetting the Module, page 31-25

Installing and Booting an Image on the Module

If the module suffers a failure, and the module application image cannot run, you can reinstall a new image on the module from a TFTP server (for a hardware module), or from the local disk (software module).

Note Do not use the **upgrade** command within the module software to install the image.

Prerequisites

- Hardware module—Be sure the TFTP server that you specify can transfer files up to 60 MB in size.

  Note This process can take approximately 15 minutes to complete, depending on your network and the size of the image.

- Software module—Copy the image to the ASA internal flash (disk0) before completing this procedure.

Note Before you download the IPS software to disk0, make sure at least 50% of the flash memory is free. When you install IPS, IPS reserves 50% of the internal flash memory for its file system.

Detailed Steps

	Command	Purpose

Step 1	For a hardware module (for example, the ASA 5585-X):
	`hw-module module 1 recover configure`
	For a software module (for example, the ASA 5545-X):
	`sw-module module ips recover configure image disk0:`*`file_path`*
	Example:
	`ciscoasa# hw-module module 1 recover configure`
	`Image URL [tftp://127.0.0.1/myimage]: tftp://10.1.1.1/ids-newimg`
	`Port IP Address [127.0.0.2]: 10.1.2.10`
	`Port Mask [255.255.255.254]: 255.255.255.0`
	`Gateway IP Address [1.1.2.10]: 10.1.2.254`
	`VLAN ID [0]: 100`
		Specifies the location of the new image.
		For a hardware module—This command prompts you for the URL for the TFTP server, the management interface IP address and netmask, gateway address, and VLAN ID (ASA 5505 only). These network parameters are configured in ROMMON; the network parameters you configured in the module application configuration are not available to ROMMON, so you must set them separately here.
		For a software module—Specify the location of the image on the local disk.
		You can view the recovery configuration using the **show module** {**1** | **ips**} **recover** command.
		In multiple context mode, enter this command in the system execution space.

Step 2	For a hardware module:
	`hw-module module 1 recover boot`
	For a software module:
	`sw-module module ips recover boot`
	Example:
	`ciscoasa# hw-module module 1 recover boot`
		Installs and boots the IPS module software.

Step 3	For a hardware module:
	`show module 1 details`
	For a software module:
	`show module ips details`
	Example:
	`ciscoasa# show module 1 details`
		Checks the progress of the image transfer and module restart process.
		The Status field in the output indicates the operational status of the module. A module operating normally shows a status of "Up." While the ASA transfers an application image to the module, the Status field in the output reads "Recover." When the ASA completes the image transfer and restarts the module, the newly transferred image is running.

Shutting Down the Module

Shutting down the module software prepares the module to be safely powered off without losing configuration data. **Note**: If you reload the ASA, the module is not automatically shut down, so we recommend shutting down the module before reloading the ASA. To gracefully shut down the module, perform the following steps at the ASA CLI.

Detailed Steps

Command	Purpose
For a hardware module (for example, the ASA 5585-X): **hw-module module 1 shutdown** For a software module (for example, the ASA 5545-X): **sw-module module ips shutdown** **Example:** `ciscoasa# hw-module module 1 shutdown`	Shuts down the module.

Uninstalling a Software Module Image

To uninstall a software module image and associated configuration, perform the following steps.

Detailed Steps

	Command	Purpose

Step 1	**sw-module module ips uninstall**
	Example:
	`ciscoasa# sw-module module ips uninstall`
	`Module ips will be uninstalled. This will completely remove the`
	`disk image associated with the sw-module including any configuration`
	`that existed within it.`
	`Uninstall module <id>? [confirm]`
		Permanently uninstalls the software module image and associated configuration.

Step 2	**reload** **Example:** `ciscoasa# reload`	Reloads the ASA. You must reload the ASA before you can install a new module type.

Resetting the Password

You can reset the module password to the default. For the user **cisco**, the default password is **cisco**. After resetting the password, you should change it to a unique value using the module application.

Resetting the module password causes the module to reboot. Services are not available while the module is rebooting.

To reset the module password to the default of cisco, perform the following steps.

Detailed Steps

Command	Purpose

For a hardware module (for example, the ASA 5585-X):
`hw-module module 1 password-reset`
For a software module (for example, the ASA 5545-X):
`sw-module module ips password-reset`
Example:
`ciscoasa# hw-module module 1 password-reset`
	Resets the module password to **cisco** for user **cisco**.

Reloading or Resetting the Module

To reload or reset the module, enter one of the following commands at the ASA CLI.

Detailed Steps

Command	Purpose
For a hardware module (for example, the ASA 5585-X): **hw-module module 1 reload** For a software module (for example, the ASA 5545-X): **sw-module module ips reload** **Example:** `ciscoasa# hw-module module 1 reload`	Reloads the module software.
For a hardware module: **hw-module module 1 reset** For a software module: **sw-module module ips reset** **Example:** `ciscoasa# hw-module module 1 reset`	Performs a reset, and then reloads the module.

Monitoring the ASA IPS module

To check the status of a module, enter one of the following commands:

Command	Purpose
show module	Displays the status.
show module {**1** | **ips**} **details**	Displays additional status information. Specify **1** for a hardware module and **ips** for a software module.
show module {**1** | **ips**} **recover**	Displays the network parameters for transferring an image to the module. Specify **1** for a hardware module and **ips** for a software module.

Examples

The following is sample output from the **show module details** command, which provides additional information for an ASA with an SSC installed:

```
ciscoasa# show module 1 details
Getting details from the Service Module, please wait...
ASA 5500 Series Security Services Card-5
Hardware version: 0.1
```

```
Serial Number: JAB11370240
Firmware version: 1.0(14)3
Software version: 6.2(1)E2
MAC Address Range: 001d.45c2.e832 to 001d.45c2.e832
App. Name: IPS
App. Status: Up
App. Status Desc: Not Applicable
App. Version: 6.2(1)E2
Data plane Status: Up
Status: Up
Mgmt IP Addr: 209.165.201.29
Mgmt Network Mask: 255.255.224.0
Mgmt Gateway: 209.165.201.30
Mgmt Access List: 209.165.201.31/32
                   209.165.202.158/32
                   209.165.200.254/24
Mgmt Vlan: 20
```

The following is sample output from the **show module ips** command for an ASA 5525-X with an IPS SSP software module installed:

```
ciscoasa# show module ips
Mod Card Type                                    Model      Serial No.
--- -------------------------------------------- -----------------------------
ips IPS 5525 Intrusion Protection System         IPS5525    FCH1504V03P

Mod MAC Address Range                 Hw Version   Fw Version   Sw Version
--- --------------------------------- ------------ ---------------------------
ips 503d.e59c.6f89 to 503d.e59c.6f89  N/A          N/A          7.1(1.160)E4

Mod SSM Application Name           Status           SSM Application Version
--- ------------------------------ ------------------------------------------
ips IPS                            Up               7.1(1.160)E4

Mod Status             Data Plane Status     Compatibility
--- ------------------ --------------------- -------------
ips Up                 Up

Mod License Name      License Status  Time Remaining
--- ----------------- --------------- ---------------
ips IPS Module        Enabled         7 days
```

Configuration Examples for the ASA IPS module

The following example diverts all IP traffic to the ASA IPS module in promiscuous mode, and blocks all IP traffic if the ASA IPS module card fails for any reason:

```
ciscoasa(config)# access-list IPS permit ip any any
ciscoasa(config)# class-map my-ips-class
ciscoasa(config-cmap)# match access-list IPS
ciscoasa(config-cmap)# policy-map my-ips-policy
ciscoasa(config-pmap)# class my-ips-class
ciscoasa(config-pmap-c)# ips promiscuous fail-close
ciscoasa(config-pmap-c)# service-policy my-ips-policy global
```

The following example diverts all IP traffic destined for the 10.1.1.0 network and the 10.2.1.0 network to the AIP SSM in inline mode, and allows all traffic through if the AIP SSM fails for any reason. For the my-ips-class traffic, sensor1 is used; for the my-ips-class2 traffic, sensor2 is used.

```
ciscoasa(config)# access-list my-ips-acl permit ip any 10.1.1.0 255.255.255.0
ciscoasa(config)# access-list my-ips-acl2 permit ip any 10.2.1.0 255.255.255.0
ciscoasa(config)# class-map my-ips-class
ciscoasa(config-cmap)# match access-list my-ips-acl
ciscoasa(config)# class-map my-ips-class2
ciscoasa(config-cmap)# match access-list my-ips-acl2
ciscoasa(config-cmap)# policy-map my-ips-policy
ciscoasa(config-pmap)# class my-ips-class
ciscoasa(config-pmap-c)# ips inline fail-open sensor sensor1
ciscoasa(config-pmap)# class my-ips-class2
ciscoasa(config-pmap-c)# ips inline fail-open sensor sensor2
ciscoasa(config-pmap-c)# service-policy my-ips-policy interface outside
```

Feature History for the ASA IPS module

Table 31-2 lists each feature change and the platform release in which it was implemented.

Table 31-2 Feature History for the ASA IPS module

Feature Name	Platform Releases	Feature Information
AIP SSM	7.0(1)	We introduced support for the AIP SSM for the ASA 5510, 5520, and 5540. The following command was introduced: **ips**.
Virtual sensors (ASA 5510 and higher)	8.0(2)	Virtual sensor support was introduced. Virtual sensors let you configure multiple security policies on the ASA IPS module. The following command was introduced: **allocate-ips**.
AIP SSC for the ASA 5505	8.2(1)	We introduced support for the AIP SSC for the ASA 5505. The following commands were introduced: **allow-ssc-mgmt**, **hw-module module ip**, and **hw-module module allow-ip**.
Support for the ASA IPS SSP-10, -20, -40, and -60 for the ASA 5585-X	8.2(5)/ 8.4(2)	We introduced support for the ASA IPS SSP-10, -20, -40, and -60 for the ASA 5585-X. You can only install the ASA IPS SSP with a matching-level SSP; for example, SSP-10 and ASA IPS SSP-10. **Note** The ASA 5585-X is not supported in Version 8.3.

Table 31-2 Feature History for the ASA IPS module (continued)

Feature Name	Platform Releases	Feature Information
Support for Dual SSPs for SSP-40 and SSP-60	8.4(2)	For SSP-40 and SSP-60, you can use two SSPs of the same level in the same chassis. Mixed-level SSPs are not supported (for example, an SSP-40 with an SSP-60 is not supported). Each SSP acts as an independent device, with separate configurations and management. You can use the two SSPs as a failover pair if desired. **Note** When using two SSPs in the chassis, VPN is not supported; note, however, that VPN has not been disabled. We modified the following commands: **show module**, **show inventory**, **show environment**.
Support for the ASA IPS SSP for the ASA 5512-X through ASA 5555-X	8.6(1)	We introduced support for the ASA IPS SSP software module for the ASA 5512-X, ASA 5515-X, ASA 5525-X, ASA 5545-X, and ASA 5555-X. We introduced or modified the following commands: **session**, **show module**, **sw-module**.

CHAPTER 32

Configuring the ASA CSC Module

This chapter describes how to configure the Content Security and Control (CSC) application that is installed in a CSC SSM in the ASA.

This chapter includes the following sections:

Information About the CSC SSM

Some ASA models support the CSC SSM, which runs Content Security and Control software. The CSC SSM provides protection against viruses, spyware, spam, and other unwanted traffic by scanning the FTP, HTTP/HTTPS, POP3, and SMTP packets that you configure the ASA to send to it.

For more information about the CSC SSM, see the following URL:

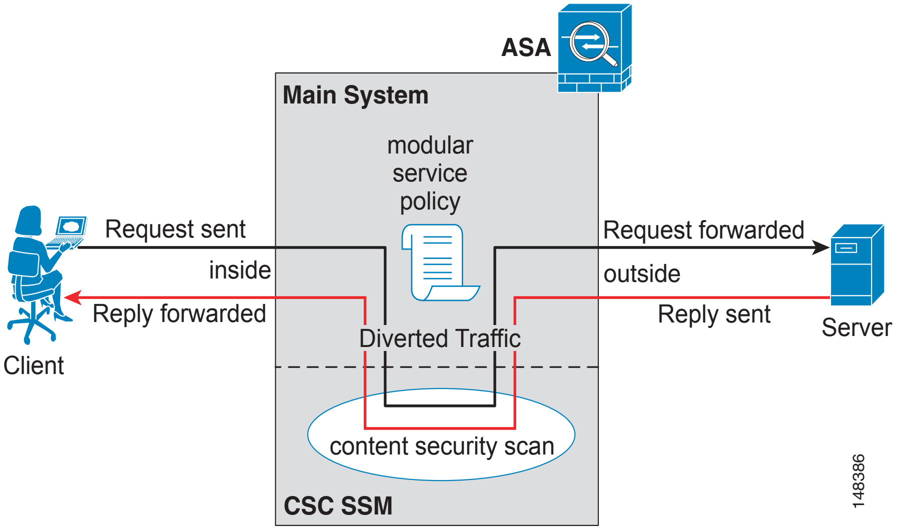
http://www.cisco.com/en/US/products/ps6823/index.html

Figure 32-1 shows the flow of traffic through an ASA that has the following:

- A CSC SSM installed and configured.
- A service policy that determines what traffic is diverted to the CSC SSM for scanning.

In this example, the client could be a network user who is accessing a website, downloading files from an FTP server, or retrieving mail from a POP3 server. SMTP scans differ in that you should configure the ASA to scan traffic sent from the outside to SMTP servers protected by the ASA.

Figure 32-1 Flow of Scanned Traffic with the CSC SSM



You use ASDM for system setup and monitoring of the CSC SSM. For advanced configuration of content security policies in the CSC SSM software, you access the web-based GUI for the CSC SSM by clicking links within ASDM. The CSC SSM GUI appears in a separate web browser window. To access the CSC SSM, you must enter the CSC SSM password. To use the CSC SSM GUI, see the *Cisco Content Security and Control SSM Administrator Guide*.

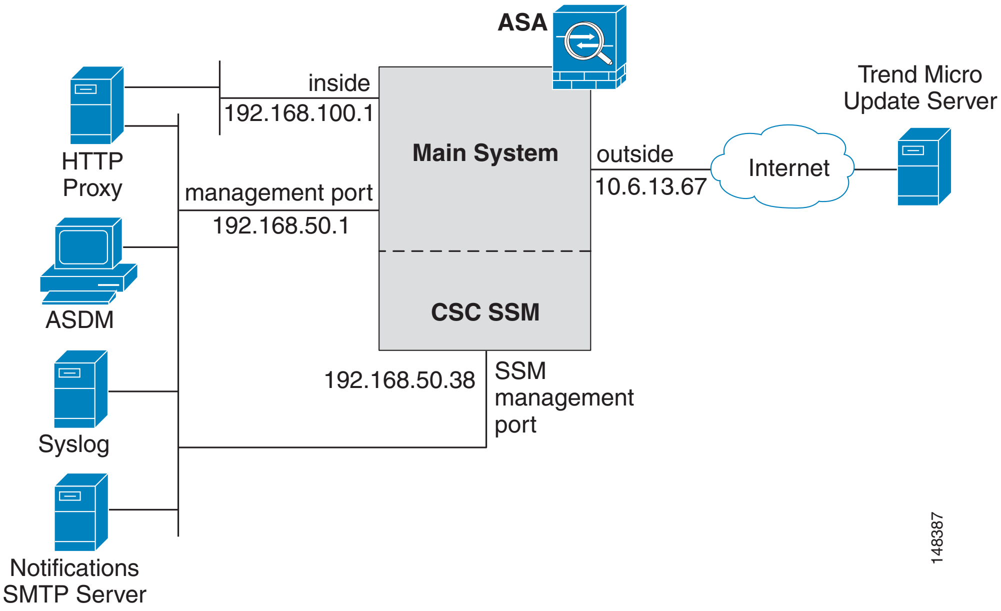
Note ASDM and the CSC SSM maintain separate passwords. You can configure their passwords to be identical; however, changing one of these two passwords does not affect the other password.

The connection between the host running ASDM and the ASA is made through a management port on the ASA. The connection to the CSC SSM GUI is made through the SSM management port. Because these two connections are required to manage the CSC SSM, any host running ASDM must be able to reach the IP address of both the ASA management port and the SSM management port.

Figure 32-2 shows an ASA with a CSC SSM that is connected to a dedicated management network. While use of a dedicated management network is not required, we recommend it. In this configuration, the following items are of particular interest:

- An HTTP proxy server is connected to the inside network and to the management network. This HTTP proxy server enables the CSC SSM to contact the Trend Micro Systems update server.
- The management port of the ASA is connected to the management network. To allow management of the ASA and the CSC SSM, hosts running ASDM must be connected to the management network.
- The management network includes an SMTP server for e-mail notifications for the CSC SSM and a syslog server to which the CSC SSM can send syslog messages.

Figure 32-2 CSC SSM Deployment with a Management Network



Determining What Traffic to Scan

The CSC SSM can scan FTP, HTTP/HTTPS, POP3, and SMTP traffic only when the destination port of the packet requesting the connection is the well-known port for the specified protocol. The CSC SSM can scan only the following connections:

- FTP connections opened to TCP port 21.
- HTTP connections opened to TCP port 80.
- HTTPS connections opened to TCP port 443.
- POP3 connections opened to TCP port 110.
- SMTP connections opened to TCP port 25.

You can choose to scan traffic for all of these protocols or any combination of them. For example, if you do not allow network users to receive POP3 e-mail, do not configure the ASA to divert POP3 traffic to the CSC SSM. Instead, block this traffic.

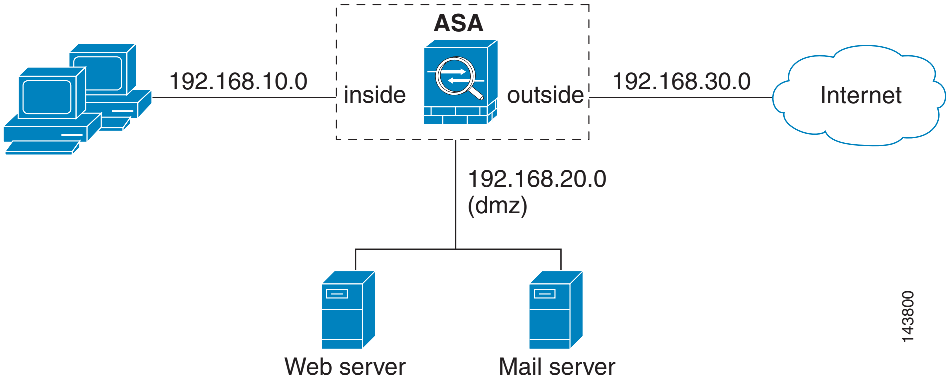
To maximize performance of the ASA and the CSC SSM, divert only the traffic to the CSC SSM that you want the CSC SSM to scan. Diverting traffic that you do not want scanned, such as traffic between a trusted source and destination, can adversely affect network performance.

Note When traffic is first classified for CSC inspection, it is flow-based. If traffic is part of a pre-existing connection, the traffic goes directly to the service policy set for that connection.

You can apply service policies that include CSC scanning globally or to specific interfaces; therefore, you can choose to enable CSC scans globally or for specific interfaces.

Based on the configuration shown in Figure 32-3, configure the ASA to divert to the CSC SSM only requests from clients on the inside network for HTTP, FTP, and POP3 connections to the outside network, and incoming SMTP connections from outside hosts to the mail server on the DMZ network. Exclude from scanning HTTP requests from the inside network to the web server on the DMZ network.

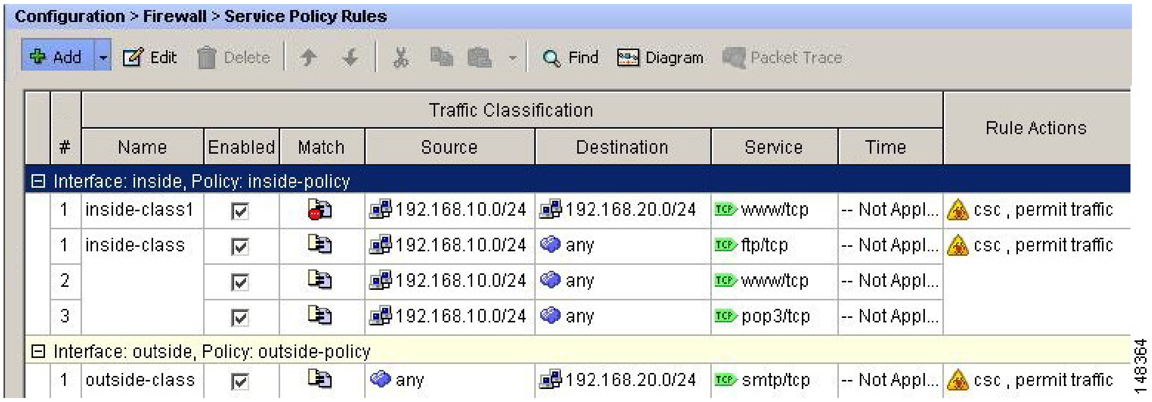
Figure 32-3 Common Network Configuration for CSC SSM Scanning



There are many ways you could configure the ASA to identify the traffic that you want to scan. One approach is to define two service policies: one on the inside interface and the other on the outside interface, each with ACLs that match traffic to be scanned.

Figure 32-4 shows service policy rules that select only the traffic that the ASA should scan.

Figure 32-4 Optimized Traffic Selection for CSC Scans



	Traffic Classification							Rule Actions
#	Name	Enabled	Match	Source	Destination	Service	Time	
⊟ Interface: inside, Policy: inside-policy								
1	inside-class1	☑	⊘	192.168.10.0/24	192.168.20.0/24	www/tcp	-- Not Appl...	csc , permit traffic
1	inside-class	☑	✓	192.168.10.0/24	any	ftp/tcp	-- Not Appl...	csc , permit traffic
2		☑	✓	192.168.10.0/24	any	www/tcp	-- Not Appl...	
3		☑	✓	192.168.10.0/24	any	pop3/tcp	-- Not Appl...	
⊟ Interface: outside, Policy: outside-policy								
1	outside-class	☑	✓	any	192.168.20.0/24	smtp/tcp	-- Not Appl...	csc , permit traffic

In the inside-policy, the first class, inside-class1, ensures that the ASA does not scan HTTP traffic between the inside network and the DMZ network. The Match column indicates this setting by displaying the "Do not match" icon. This setting does not mean the ASA blocks traffic sent from the 192.168.10.0 network to TCP port 80 on the 192.168.20.0 network. Instead, this setting exempts the traffic from being matched by the service policy applied to the inside interface, which prevents the ASA from sending the traffic to the CSC SSM.

The second class of the inside-policy, inside-class matches FTP, HTTP, and POP3 traffic between the inside network and any destination. HTTP connections to the DMZ network are exempted because of the inside-class1 setting. As previously mentioned, policies that apply CSC scanning to a specific interface affect both incoming and outgoing traffic, but by specifying 192.168.10.0 as the source network, inside-class1 matches only connections initiated by the hosts on the inside network.

In the outside-policy, outside-class matches SMTP traffic from any outside source to the DMZ network. This setting protects the SMTP server and inside users who download e-mail from the SMTP server on the DMZ network, without having to scan connections from SMTP clients to the server.

If the web server on the DMZ network receives files uploaded by HTTP from external hosts, you can add a rule to the outside policy that matches HTTP traffic from any source to the DMZ network. Because the policy is applied to the outside interface, the rule would only match connections from HTTP clients outside the ASA.

Licensing Requirements for the CSC SSM

Model	License Requirement
ASA 5510	• Base License—Supports SMTP virus scanning, POP3 virus scanning and content filtering, web mail virus scanning, HTTP file blocking, FTP virus scanning and file blocking, logging, and automatic updates. Supports two contexts. *Optional licenses: 5 contexts.* • Security Plus License—Supports the Base license features, plus SMTP anti-spam, SMTP content filtering, POP3 anti-spam, URL blocking, and URL filtering. Supports two contexts. *Optional license: 5 contexts.*
ASA 5520	Base License—Supports all features. Supports two contexts. *Optional licenses: 5, 10, or 20 contexts.*
ASA 5540	Base License—Supports all features. Supports two contexts. *Optional licenses: 5, 10, 20, or 50 contexts.*
All other models	No support.

Prerequisites for the CSC SSM

The CSC SSM has the following prerequisites:

- A CSC SSM card must be installed in the ASA.
- A Product Authorization Key (PAK) for use in registering the CSC SSM.
- Activation keys that you receive by e-mail after you register the CSC SSM.
- The management port of the CSC SSM must be connected to your network to allow management and automatic updates of the CSC SSM software.
- The CSC SSM management port IP address must be accessible by the hosts used to run ASDM.
- You must obtain the following information to use in configuring the CSC SSM:
  - The CSC SSM management port IP address, netmask, and gateway IP address.
  - DNS server IP address.
  - HTTP proxy server IP address (needed only if your security policies require the use of a proxy server for HTTP access to the Internet).

- Domain name and hostname for the CSC SSM.
- An e-mail address and an SMTP server IP address and port number for e-mail notifications.
- E-mail address(es) for product license renewal notifications.
- IP addresses of hosts or networks that are allowed to manage the CSC SSM. The IP addresses for the CSC SSM management port and the ASA management interface can be in different subnets.
- Password for the CSC SSM.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context modes.

Firewall Mode Guidelines

Supported in routed and transparent firewall modes.

Failover Guidelines

Does not support sessions in Stateful Failover. The CSC SSM does not maintain connection information, and therefore cannot provide the failover unit with the required information. The connections that a CSC SSM is scanning are dropped when the ASA in which the CSC SSM is installed fails. When the standby ASA becomes active, it forwards the scanned traffic to the CSC SSM and the connections are reset.

IPv6 Guidelines

Does not support IPv6.

Model Guidelines

Supported on the ASA 5510, ASA 5520, and ASA 5540 only. Not supported on the ASA 5580 and the ASA 5585-X.

Additional Guidelines

You cannot change the software type installed on the module; if you purchase a CSC module, you cannot later install IPS software on it.

Default Settings

Table 32-1 lists the default settings for the CSC SSM.

Table 32-1 Default CSC SSM Parameters

Parameter	Default
FTP inspection on the ASA	Enabled
All features included in the license(s) that you have purchased	Enabled

Configuring the CSC SSM

This section describes how to configure the CSC SSM and includes the following topics:

- Before Configuring the CSC SSM, page 32-7
- Connecting to the CSC SSM, page 32-8
- Diverting Traffic to the CSC SSM, page 32-10

Before Configuring the CSC SSM

Before configuring the ASA and the CSC SSM, perform the following steps:

Step 1 If the CSC SSM did not come preinstalled in a Cisco ASA, install it and connect a network cable to the management port of the SSM. For assistance with installation and connecting the SSM, see the *Cisco ASA 5500 Series Quick Start Guide*.

The management port of the CSC SSM must be connected to your network to allow management of and automatic updates to the CSC SSM software. Additionally, the CSC SSM uses the management port for e-mail notifications and syslog messages.

Step 2 You should have received a Product Authorization Key (PAK) with the CSC SSM. Use the PAK to register the CSC SSM at the following URL.

http://www.cisco.com/go/license

After you register, you receive activation keys by e-mail. The activation keys are required before you can complete Step 6.

Step 3 Obtain the following information for use in Step 6:

- Activation keys
- CSC SSM management port IP address, netmask, and gateway IP address
- DNS server IP address
- HTTP proxy server IP address (needed only if your security policies require the use of a proxy server for HTTP access to the Internet)
- Domain name and hostname for the CSC SSM
- An e-mail address, and SMTP server IP address and port number for e-mail notifications
- E-mail address(es) for product license renewal notifications
- IP addresses of hosts or networks that are allowed to manage the CSC SSM
- Password for the CSC SSM

Step 4 In a web browser, access ASDM for the ASA in which the CSC SSM is installed.

Note If you are accessing ASDM for the first time, see the "Additional References" section on page 32-18.

For more information about enabling ASDM access, see the "Configuring ASA Access for ASDM, Telnet, or SSH" section on page 45-1 in the general operations configuration guide.

Step 5 Verify time settings on the ASA. Time setting accuracy is important for logging of security events and for automatic updates of CSC SSM software. Do one of the following:

- If you manually control time settings, verify the clock settings, including time zone. Choose **Configuration > Properties > Device Administration > Clock**.
- If you are using NTP, verify the NTP configuration. Choose **Configuration > Properties > Device Administration > NTP**.

Step 6 Open ASDM.

Step 7 Connect to and log in to the CSC SSM. For instructions, see the "Connecting to the CSC SSM" section on page 32-8.

Step 8 Configure service policies to divert traffic that you want scanned to the CSC SSM. For instructions, see the "Diverting Traffic to the CSC SSM" section on page 32-10.

Step 9 Run the CSC Setup Wizard.

- To access the CSC Setup Wizard, choose **Configuration > Trend Micro Content Security > CSC Setup > Wizard Setup > Launch Setup Wizard**.
- If you are rerunning the CSC Setup Wizard, perform the same step listed in the previous bullet.

The CSC Setup Wizard appears.

Step 10 Complete the CSC Setup Wizard.

Note If you create a global service policy to divert traffic for CSC scans, all traffic (inbound and outbound) for the supported protocols is scanned. To maximize performance of the ASA and the CSC SSM, scan traffic only from untrusted sources.

Step 11 To reduce the load on the CSC SSM, configure the service policy rules that send packets to the CSC SSM to support only HTTP/HTTPS, SMTP, POP3, or FTP traffic.

Step 12 (Optional) Review the default content security policies in the CSC SSM GUI, which are suitable for most implementations. You review the content security policies by viewing the enabled features in the CSC SSM GUI. For the availability of features, see the "Licensing Requirements for the CSC SSM" section on page 32-5. For the default settings, see the "Default Settings" section on page 32-6.

What to Do Next

See the "Connecting to the CSC SSM" section on page 32-8.

Connecting to the CSC SSM

With each session you start in ASDM, the first time you access features related to the CSC SSM, you must specify the management IP address and provide the password for the CSC SSM. After you successfully connect to the CSC SSM, you are not prompted again for the management IP address and password. If you start a new ASDM session, the connection to the CSC SSM is reset and you must specify the IP address and the CSC SSM password again. The connection to the CSC SSM is also reset if you change the time zone on the ASA.

Note The CSC SSM has a password that is maintained separately from the ASDM password. You can configure the two passwords to be identical, but changing the CSC SSM password does not affect the ASDM password.

To connect to the CSC SSM, perform the following steps:

Step 1 In the ASDM main application window, click the **Content Security** tab.

Step 2 In the Connecting to CSC dialog box, click one of the following radio buttons:

- To connect to the IP address of the management port on the SSM, click **Management IP Address**. ASDM automatically detects the IP address for the SSM in the ASA. If this detection fails, you can specify the management IP address manually.
- To connect to an alternate IP address or hostname on the SSM, click **Other IP Address or Hostname**.

Step 3 Enter the port number in the Port field, and then click **Continue**.

Step 4 In the CSC Password field, type your CSC password, and then click **OK**.

Note If you have not completed the CSC Setup Wizard (choose **Configuration > Trend Micro Content Security > CSC Setup > Wizard Setup**), complete the configuration in the CSC Setup Wizard, which includes changing the default password, "cisco."

For ten minutes after you have entered the password, you do not need to reenter the CSC SSM password to access other parts of the CSC SSM GUI.

Step 5 To access the CSC SSM GUI, choose **Configuration > Trend Micro Content Security**, and then click one of the following tabs: **Web**, **Mail**, **File Transfer**, or **Updates**.

What to Do Next

See the "Diverting Traffic to the CSC SSM" section on page 32-10.

Diverting Traffic to the CSC SSM

You use Modular Policy Framework commands to configure the ASA to divert traffic to the CSC SSM.

Prerequisites

Before configuring the ASA to divert traffic to the CSC SSM, see Chapter 1, "Configuring a Service Policy Using the Modular Policy Framework," which introduces Modular Policy Framework concepts and common commands.

To configure the ASA to divert traffic to the CSC SSM, perform the following steps:

Detailed Steps

	Command	Purpose
Step 1	**access-list extended** **Example:** `ciscoasa(config)# access-list extended`	Creates an ACL that matches the traffic you want scanned by the CSC SSM. Create as many ACEs as are needed to match all the traffic. For example, to specify FTP, HTTP/HTTPS, POP3, and SMTP traffic, you need four ACEs. For guidance on identifying the traffic that you want to scan, see the "Determining What Traffic to Scan" section on page 32-3.
Step 2	**class-map** *class_map_name* **Example:** ciscoasa(config)# **class-map** *class_map_name*	Creates a class map to identify the traffic that should be diverted to the CSC SSM. The *class_map_name* argument is the name of the traffic class. When you enter the **class-map** command, the CLI enters class map configuration mode.
Step 3	**match access-list** *acl-name* **Example:** ciscoasa(config-cmap)# match access-list *acl-name*	Identifies the traffic to be scanned with the ACL that you created in Step 1. The *acl-name* argument is the name of the ACL.
Step 4	**policy-map** *policy_map_name* **Example:** ciscoasa(config-cmap)# policy-map *policy_map_name*	Creates a policy map or modify an existing policy map that you want to use to send traffic to the CSC SSM. The *policy_map_name* argument is the name of the policy map. When you enter the **policy-map** command, the CLI enters policy map configuration mode.
Step 5	**class** *class_map_name* **Example:** ciscoasa(config-pmap)# class *class_map_name*	Specifies the class map, created in Step 2, that identifies the traffic to be scanned. The *class_map_name* argument is the name of the class map that you created in Step 2. The CLI enters the policy map class configuration mode.

	Command	Purpose

Step 6	**`set connection per-client-max`** *`n`*
	Example:
	`ciscoasa(config-pmap-c)# set connection per-client-max 5`
		Lets you configure limits to thwart DoS attacks. The **per-client-max** parameter limits the maximum number of connections that individual clients can open. If a client uses more network resources simultaneously than is desired, you can enforce a per-client limit for simultaneous connections that the ASA diverts to the CSC SSM. The *n* argument is the maximum number of simultaneous connections that the ASA allows per client. This command prevents a single client from abusing the services of the CSC SSM or any server protected by the SSM, including prevention of attempts at DoS attacks on HTTP/HTTPS, FTP, POP3, or SMTP servers that the CSC SSM protects.

	Command	Purpose
Step 7	**csc** {**fail-close** | **fail-open**} **Example:** `ciscoasa(config-pmap-c)# csc {fail-close	fail-open}`	Enables traffic scanning with the CSC SSM and assigns the traffic identified by the class map as traffic to be sent to the CSC SSM. Must be part of a service policy, which can be applied globally or to specific interfaces. Ensures that all unencrypted connections through the ASA are scanned by the CSC SSM; however, this setting may mean that traffic from trusted sources is needlessly scanned. If enabled in interface-specific service policies, this command is bi-directional. Bi-directionality means that when the ASA opens a new connection, if this command is active on either the inbound or the outbound interface of the connection and the class map for the policy identifies traffic for scanning, the ASA diverts this traffic to the CSC SSM. However, bi-directionality also means that if you divert any of the supported traffic types that cross a given interface to the CSC SSM, it is probably performing unnecessary scans on traffic from your trusted inside networks. Therefore, to further limit the traffic selected by the class maps of CSC SSM service policies, we recommend using ACLs that match the following: • HTTP/HTTPS connections to outside networks. • FTP connections from clients inside the ASA to servers outside the ASA. • POP3 connections from clients inside the ASA to servers outside the ASA. • Incoming SMTP connections destined to inside mail servers. The **fail-close** and **fail-open** keywords control how the ASA handles traffic when the CSC SSM is unavailable. For more information about the operating modes and failure behavior, see the "Guidelines and Limitations" section on page 32-6.
Step 8	**service-policy** *policy_map_name* [**global** | **interface** *interface_ID*] **Example:** `ciscoasa(config-pmap-c)# service-policy` *`policy_map_name`* `[global	interface` *`interface_ID`*`]`	Applies the policy map globally or to a specific interface. The *policy_map_name* argument is the policy map that you configured in Step 4. To apply the policy map to traffic on all the interfaces, use the **global** keyword. To apply the policy map to traffic on a specific interface, use the **interface** *interface_ID* keyword and argument pair, where *interface_ID* is the name assigned to the interface with the **nameif** command. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

What to Do Next

See the "Monitoring the CSC SSM" section on page 32-13.

Monitoring the CSC SSM

To check the status of a module, enter one of the following commands:

Command	Purpose
show module	Displays the status.
show module 1 details	Displays additional status information.
show module 1 recover	Displays the network parameters for transferring an image to the module.

Examples

The following is sample output from the **show module** command for an ASA with a CSC SSM installed:

```
hostname# show module
Mod Card Type                                    Model              Serial No.
--- -------------------------------------------- ------------------ -----------
  0 ASA 5520 Adaptive Security Appliance         ASA5520            JMX1241L05S
  1 ASA 5500 Series Content Security Services Mo ASA-SSM-CSC-10     AF1234BQQL

Mod SSM Application Name           Status           SSM Application Version
--- ------------------------------ ---------------- --------------------------
  1 CSC SSM                        Down             6.2.1599.0
```

The following is sample output from the **show module details** command, which provides additional information about an ASA with a CSC SSM installed:

```
ciscoasa# show module 1 details
Getting details from the Service Module, please wait...
ASA 5500 Series Security Services Module-20
Model: ASA-SSM-20
Hardware version: 1.0
Serial Number: JAF10333331
Firmware version: 1.0(10)0
Software version: Trend Micro InterScan Security Module Version 6.2
App. name: Trend Micro InterScan Security Module
App. version: Version 6.2
Data plane Status: Up
Status: Up
HTTP Service: Up
Mail Service: Up
FTP Service: Up
Activated: Yes
Mgmt IP addr: 209.165.200.225
Mgmt web port: 8443
```

The following is sample output from the **show module recover** command, which includes recovery details for an ASA with a CSC SSM installed:

```
ciscoasa# show module 1 recover
Module 1 recover parameters. . .
Boot Recovery Image: Yes
Image URL: tftp://10.21.18.1/ids-oldimg
Port IP Address: 209.165.200.230
Port Mask: 255.255.224.0
Gateway IP Address: 209.165.200.254
```

Troubleshooting the CSC Module

This section includes procedures that help you recover or troubleshoot the module and includes the following topics:

- Installing an Image on the Module, page 32-14
- Resetting the Password, page 32-15
- Reloading or Resetting the Module, page 32-16
- Shutting Down the Module, page 32-17

Note This section covers all ASA module types; follow the steps appropriate for your module.

Installing an Image on the Module

If the module suffers a failure, and the module application image cannot run, you can reinstall a new image on the module from a TFTP server.

Note Do not use the **upgrade** command within the module software to install the image.

Prerequisites

Be sure the TFTP server that you specify can transfer files up to 60 MB in size.

Note This process can take approximately 15 minutes to complete, depending on your network and the size of the image.

Detailed Steps

	Command	Purpose

Step 1	**hw-module module 1 recover configure**
	Example:
	`ciscoasa# hw-module module 1 recover configure`
	`Image URL [tftp://127.0.0.1/myimage]: tftp://10.1.1.1/ids-newimg`
	`Port IP Address [127.0.0.2]: 10.1.2.10`
	`Port Mask [255.255.255.254]: 255.255.255.0`
	`Gateway IP Address [1.1.2.10]: 10.1.2.254`
	`VLAN ID [0]: 100`
		Specifies the location of the new image. This command prompts you for the URL for the TFTP server, the management interface IP address and netmask, gateway address, and VLAN ID (ASA 5505 only). These network parameters are configured in ROMMON; the network parameters you configured in the module application configuration are not available to ROMMON, so you must set them separately here.
		You can view the recovery configuration using the **show module 1 recover** command.
		In multiple context mode, enter this command in the system execution space.

Step 2	**hw-module module 1 recover boot** **Example:** `ciscoasa# hw-module module 1 recover boot`	Transfers the image from the TFTP server to the module and restarts the module.
Step 3	**show module 1 details** **Example:** `ciscoasa# show module 1 details`	Checks the progress of the image transfer and module restart process. The Status field in the output indicates the operational status of the module. A module operating normally shows a status of "Up." While the ASA transfers an application image to the module, the Status field in the output reads "Recover." When the ASA completes the image transfer and restarts the module, the newly transferred image is running.

Resetting the Password

You can reset the module password to the default. The default password is cisco. After resetting the password, you should change it to a unique value using the module application.

Resetting the module password causes the module to reboot. Services are not available while the module is rebooting.

To reset the module password to the default of cisco, perform the following steps.

Detailed Steps

Command	Purpose
hw-module module 1 password-reset **Example:** `ciscoasa# hw-module module 1 password-reset`	Resets the module password to cisco. The *1* is the specified slot number on the SSM hardware module. On the CSC SSM, entering this command resets web services on the hardware module after the password has been reset. You may lose connection to ASDM or be logged out of the hardware module. The CSC SSM supports this command in the most recent version of 6.3, dated January 2010, and in later versions. **Note** Make sure that the SSM hardware module is in the Up state and supports password reset.

Reloading or Resetting the Module

To reload or reset the module, enter one of the following commands at the ASA CLI.

Detailed Steps

Command	Purpose
hw-module module 1 reload **Example:** `ciscoasa# hw-module module 1 reload`	Reloads the module software.
hw-module module 1 reset **Example:** `ciscoasa# hw-module module 1 reset`	Performs a reset, then reloads the module.

Shutting Down the Module

If you restart the ASA, the module is not automatically restarted. To shut down the module, perform the following steps at the ASA CLI.

Detailed Steps

Command	Purpose
hw-module module 1 shutdown **Example:** `ciscoasa# hw-module module 1 shutdown`	Shuts down the module.

Configuration Examples for the CSC SSM

To identify the traffic that you want to scan, you can configure the ASA in different ways. One approach is to define two service policies, one on the inside interface and one on the outside interface, each with an ACL that matches traffic to be scanned. The following example is based on the network shown in Figure 32-3 and shows the creation of two service policies for a common CSC SSM scanning scenario:

- The first policy, csc_out_policy, is applied to the inside interface and uses the csc_out ACL to ensure that all outbound requests for FTP and POP3 are scanned. The csc_out ACL also ensures that HTTP connections from inside to networks on the outside interface are scanned, but it includes a deny ACE to exclude HTTP connections from inside to servers on the DMZ network.
- The second policy, csc_in_policy, is applied to the outside interface and uses the csc_in ACL to ensure that requests for SMTP and HTTP originating on the outside interface and destined for the DMZ network are scanned by the CSC SSM. Scanning HTTP requests protects the web server from HTTP file uploads.

```
ciscoasa(config)# access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 21
ciscoasa(config)# access-list csc_out deny tcp 192.168.10.0 255.255.255.0 192.168.20.0
255.255.255.0 eq 80
ciscoasa(config)# access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 80
ciscoasa(config)# access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 110

ciscoasa(config)# class-map csc_outbound_class
ciscoasa(config-cmap)# match access-list csc_out

ciscoasa(config-cmap)# policy-map csc_out_policy
ciscoasa(config-pmap)# class csc_outbound_class
ciscoasa(config-pmap-c)# csc fail-close

ciscoasa(config-pmap-c)# service-policy csc_out_policy interface inside

ciscoasa(config)# access-list csc_in permit tcp any 192.168.20.0 255.255.255.0 eq 25
ciscoasa(config)# access-list csc_in permit tcp any 192.168.20.0 255.255.255.0 eq 80

ciscoasa(config)# class-map csc_inbound_class
ciscoasa(config-cmap)# match access-list csc_in

ciscoasa(config-cmap)# policy-map csc_in_policy
ciscoasa(config-pmap)# class csc_inbound_class
ciscoasa(config-pmap-c)# csc fail-close

ciscoasa(config-pmap-c)# service-policy csc_in_policy interface outside
```

The following example shows how to use an ACL to exempt the traffic from being matched by the policy map and prevent the ASA from sending traffic to the CSC SSM:

```
ciscoasa(config)# access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 21
ciscoasa(config)# access-list csc_out deny tcp 192.168.10.0 255.255.255.0 192.168.20.0
255.255.255.0 eq 80
ciscoasa(config)# access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 80
ciscoasa(config)# access-list csc_out permit tcp 192.168.10.0 255.255.255.0 any eq 110
```

The following example shows how to add an ACE to the csc_out ACL to exclude HTTP connections between the trusted external web server and inside hosts from being scanned by the CSC SSM:

```
ciscoasa(config)# access-list csc_out deny tcp 192.168.10.0 255.255.255.0 209.165.201.7
255.255.255.255 eq 80
```

The following example shows how to use the ACL on the service policy applied to the outside interface:

```
ciscoasa(config)# access-list csc_in permit tcp any 192.168.20.0 255.255.255.0 eq 25
```

The following example shows how to add an ACE to the csc_in ACL to use the CSC SSM to protect the web server on a DMZ network from infected files uploaded by HTTP from external hosts:

```
ciscoasa(config)# access-list csc_in permit tcp any 192.168.20.0 255.255.255.0 eq 80
```

Additional References

For additional information related to implementing the CSC SSM, see the following documents:

Related Topic	Document Title
Instructions on use of the CSC SSM GUI. Additional licensing requirements of specific windows available in the CSC SSM GUI. Reviewing the default content security policies in the CSC SSM GUI before modifying them or entering advanced configuration settings.	*Cisco Content Security and Control SSM Administrator Guide*
Accessing ASDM for the first time and assistance with the Startup Wizard.	*Cisco ASA 5500 Series Quick Start Guide*
Assistance with SSM hardware installation and connection to the ASA.	hardware guide
Accessing ASDM for the first time and assistance with the Startup Wizard.	*Cisco ASA 5500 Series Quick Start Guide*

Related Topic	Document Title
Instructions on use of the CSC SSM GUI. Additional licensing requirements of specific windows available in the CSC SSM GUI. Reviewing the default content security policies in the CSC SSM GUI before modifying them or entering advanced configuration settings.	*Cisco Content Security and Control SSM Administrator Guide*
Technical Documentation, Marketing, and Support-related information.	See the following URL: http://www.cisco.com/en/US/products/ps6823/index.html.

Feature History for the CSC SSM

Table 32-2 lists each feature change and the platform release in which it was implemented.

Table 32-2 Feature History for the CSC SSM

Feature Name	Platform Releases	Feature Information
CSC SSM	7.0(1)	The CSC SSM runs Content Security and Control software, which provides protection against viruses, spyware, spam, and other unwanted traffic. We introduced the following commands: **csc {fail-close | fail-open}, hw-module module 1** [**recover** | **reload** | **reset** | **shutdown**], **session, show module** [**all** | *slot* [**details** | **recover**]].
Password reset	7.2(2)	We introduced the following command: **hw-module module password-reset**.
CSC SSM	8.1(1) and 8.1(2)	This feature is not supported on the ASA 5580.
CSC syslog format	8.3(1)	CSC syslog format is consistent with the ASA syslog format. Syslog message explanations have been added to the *Cisco Content Security and Control SSM Administrator Guide*. All syslog messages include predefined syslog priorities and cannot be configured through the CSC SSM GUI.

INDEX