

Cisco IronPort AsyncOS 7.6 for Email Advanced Configuration Guide

March 8, 2012

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

Text Part Number: OL-26343-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCDE, CCENT, CCSI, Cisco Eos, Cisco HealthPresence, Cisco IronPort, the Cisco logo, Cisco Nurse Connect, Cisco Pulse, Cisco SensorBase, Cisco StackPower, Cisco StadiumVision, Cisco TelePresence, Cisco Unified Computing System, Cisco WebEx, DCE, Flip Channels, Flip for Good, Flip Mino, Flipshare (Design), Flip Ultra, Flip Video, Flip Video (Design), Instant Broadband, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn, Cisco Capital, Cisco Capital (Design), Cisco:Financed (Stylized), Cisco Store, Flip Gift Card, and One Million Acts of Green are service marks; and Access Registrar, Aironet, AllTouch, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Lumin, Cisco Nexus, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, Continuum, EtherFast, EtherSwitch, Event Center, Explorer, Follow Me Browsing, GainMaker, iLYNX, IOS, iPhone, IronPort, the IronPort logo, Laser Link, LightStream, Linksys, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, PCNow, PIX, PowerKEY, PowerPanels, PowerTV, PowerTV (Design), PowerVu, Prisma, ProConnect, ROSA, SenderBase, SMARTnet, Spectrum Expert, StackWise, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0910R)

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 IronPort AsyncOS 7.6 for Email Advanced Configuration Guide
© 2011 Cisco Systems, Inc. All rights reserved.



CONTENTS

Preface xvii

Before you Read this Book xvii

Documentation Set xvii

How This Book Is Organized xvii

Typographic Conventions xix

Contacting Cisco IronPort Customer Support xix

Cisco IronPort Welcomes Your Comments xix

CHAPTER 1 **Customizing Listeners** 1-1

Listeners Overview 1-2

Configuring Listeners via the GUI 1-3

Global Settings for Listeners 1-4

Configuring Global Settings for Listeners 1-7

Creating Listeners 1-7

SMTP Address Parsing Options 1-8

Strict Mode 1-8

Loose Mode 1-9

Additional Options 1-9

Partial Domains, Default Domains, and Malformed MAIL FROMs 1-11

Advanced Configuration Options 1-11

LDAP Options 1-12

Accept Queries 1-12

Routing Queries 1-13

Masquerade Queries 1-13

Group Queries 1-13

Editing Listeners 1-14

Deleting Listeners 1-14

Configuring Listeners via the CLI 1-14

Advanced HAT Parameters 1-15

SenderBase Settings and HAT Mail Flow Policies 1-16

Timeouts for SenderBase Queries 1-17

HAT Significant Bits Feature 1-18

Encrypting SMTP Conversations Using TLS 1-22

Obtaining Certificates 1-22

Intermediate Certificates 1-23
Creating a Self-Signed Certificate 1-23
Importing a Certificate 1-24
Exporting a Certificate 1-25
Managing Lists of Certificate Authorities 1-25
Importing a Custom Certificate Authority List 1-26
Disabling the System Certificate Authority List 1-26
Exporting a Certificate Authorities List 1-26
Enabling TLS on a Listener's HAT 1-27
Assigning a Certificate 1-27
Logging 1-28
GUI Example 1-28
CLI Example 1-28
Enabling TLS and Certificate Verification on Delivery 1-29
Sending Alerts When a Required TLS Connection Fails 1-31
Logging 1-31
CLI Example 1-32
Enabling a Certificate for HTTPS 1-35

CHAPTER 2 **Configuring Routing and Delivery Features** 2-1
Routing Email for Local Domains 2-1
SMTP Routes Overview 2-2
Default SMTP Route 2-2
Defining an SMTP Route 2-2
SMTP Routes Limits 2-3
SMTP Routes and DNS 2-3
SMTP Routes and Alerts 2-3
SMTP Routes, Mail Delivery, and Message Splintering 2-3
SMTP Routes and Outbound SMTP Authentication. 2-4
Managing SMTP Routes via the GUI 2-4
Adding SMTP Routes 2-4
Editing SMTP Routes 2-5
Deleting SMTP Routes 2-5
Exporting SMTP Routes 2-5
Importing SMTP Routes 2-5
Rewriting Addresses 2-7
Creating Alias Tables 2-8
Configuring an Alias Table from the Command Line 2-8
Exporting and Importing an Alias Table 2-9

Deleting Entries from the Alias Table 2-9
Example Alias Table 2-10
Example aliasconfig Command 2-12
Configuring Masquerading 2-16
Masquerading and altsrchost 2-17
Configuring Static Masquerading Tables 2-17
Sample Masquerading Table for a Private Listener 2-19
Importing a Masquerading Table 2-19
Example Masquerading 2-19
The Domain Map Feature 2-28
Importing and Exporting a Domain Map Table 2-34
Directing Bounced Email 2-35
Handling Undeliverable Email 2-36
Notes on Soft and Hard Bounces 2-36
Bounce Profile Parameters 2-37
Hard Bounces and the status Command 2-38
Conversational Bounces and SMTP Routes Message Filter actions 2-38
Example Bounce Profiles 2-39
Delivery Status Notification Format 2-39
Delay Warning Messages 2-40
Delay Warning Messages and Hard Bounces 2-40
Creating a New Bounce Profile 2-40
Editing the Default Bounce Profile 2-41
Example of a Minimalist Bounce Profile 2-42
Applying Bounce Profiles to Listeners 2-42
Controlling Email Delivery 2-44
Determining Which Interface is Used for Mail Delivery 2-45
Default Delivery Limits 2-46
Working with Destination Controls 2-46
Controlling the Version of Internet Protocol Addresses 2-46
Controlling the Number of Connections, Messages, and Recipients to a Domain 2-46
Controlling TLS 2-47
Controlling Cisco IronPort Bounce Verification Tagging 2-48
Controlling Bounces 2-48
Adding a New Destination Control Entry 2-48
Editing Destination Control Entries 2-48
Deleting Destination Control Entries 2-49
Importing and Exporting Destination Control Configurations 2-49
Destination Controls and the CLI 2-53

Cisco IronPort Bounce Verification 2-53
Overview: Tagging and Cisco IronPort Bounce Verification 2-53
Handling Incoming Bounce Messages 2-54
Cisco IronPort Bounce Verification Address Tagging Keys 2-54
Cisco IronPort Bounce Verification and the HAT 2-55
Working with Cisco IronPort Bounce Verification 2-55
Configuring Bounce Verification Address Tagging Keys 2-56
Configuring Cisco IronPort Bounce Verification Settings 2-56
Cisco IronPort Bounce Verification and the CLI 2-56
Cisco IronPort Bounce Verification and Cluster Configuration 2-57
Set Email Delivery Parameters 2-57
Default Delivery IP Interface 2-57
Possible Delivery Feature 2-57
Default Maximum Concurrency 2-57
deliveryconfig Example 2-58
Using Virtual Gateway™ Technology 2-59
Overview 2-60
Setting Up Virtual Gateway Addresses 2-60
Creating New IP Interfaces for Use with Virtual Gateways 2-60
Mapping Messages to IP Interfaces for Delivery 2-63
Importing an altsrchost File 2-64
altsrchost Limits 2-64
Example Text File with Valid Mappings for the altsrchost Command 2-65
Adding an altsrchost Mapping through the CLI 2-65
Monitoring the Virtual Gateway Addresses 2-68
Managing Delivery Connections per Virtual Gateway Address 2-68
Using Global Unsubscribe 2-69
Adding a Global Unsubscribe Address Using The CLI 2-70
Exporting and Importing a Global Unsubscribe File 2-72
Review: Email Pipeline 2-73

CHAPTER 3 **LDAP Queries** 3-1
Overview 3-1
Understanding LDAP Queries 3-2
Understanding How LDAP Works with AsyncOS 3-3
Configuring AsyncOS to work with LDAP 3-4
Creating LDAP Server Profiles 3-4
Testing LDAP Servers 3-6
Working with LDAP, LDAP Queries, and Listeners 3-7

Configuring Global Settings 3-7
Example of Creating an LDAP Server Profile 3-7
Enabling LDAP Queries on a Public Listener 3-8
Enabling LDAP Queries on a Private Listener 3-9
Enhanced Support for Microsoft Exchange 5.5 3-9
Working with LDAP Queries 3-12
Types of LDAP Queries 3-12
Base Distinguishing Name (DN) 3-12
LDAP Query Syntax 3-13
Secure LDAP (SSL) 3-13
Routing Queries 3-14
Anonymous Queries 3-14
Notes for Active Directory Implementations 3-16
Testing LDAP Queries 3-17
Troubleshooting Connections to LDAP Servers 3-18
Acceptance (Recipient Validation) Queries 3-18
Sample Acceptance Queries 3-19
Configuring Acceptance Queries for Lotus Notes 3-19
Routing: Alias Expansion 3-20
Sample Routing Queries 3-20
Masquerading 3-20
Sample Masquerading Queries 3-21
Masquerading "Friendly Names" 3-21
Group LDAP Queries 3-22
Sample Group Queries 3-22
Configuring a Group Query 3-22
Example: Using a Group Query to Skip Spam and Virus Checking 3-25
Domain-based Queries 3-26
Creating a Domain-Based Query 3-26
Chain Queries 3-27
Creating a Chain Query 3-28
Using LDAP For Directory Harvest Attack Prevention 3-28
Directory Harvest Attack Prevention within the SMTP Conversation 3-29
Directory Harvest Attack Prevention within the Work Queue 3-30
Configuring Directory Harvest Prevention in the Work Queue 3-30
Configuring AsyncOS for SMTP Authentication 3-31
Configuring SMTP Authentication 3-32
Specifying a Password as Attribute 3-32
Configuring an SMTP Authentication Query 3-33

SMTP Authentication via Second SMTP Server (SMTP Auth with Forwarding) 3-34
SMTP Authentication with LDAP 3-35
Enabling SMTP Authentication on a Listener 3-36
Outgoing SMTP Authentication 3-39
Logging and SMTP Authentication 3-40
Configuring External Authentication for Users 3-40
User Accounts Query 3-41
Group Membership Queries 3-41
Spam Quarantine End-User Authentication Queries 3-43
Sample Active Directory End-User Authentication Settings 3-43
Sample OpenLDAP End-User Authentication Settings 3-44
Spam Quarantine Alias Consolidation Queries 3-44
Sample Active Directory Alias Consolidation Settings 3-45
Sample OpenLDAP Alias Consolidation Settings 3-45
User Distinguished Name Queries 3-45
Sample User Distinguished Name Settings 3-46
Sample OpenLDAP User Distinguished Name Settings 3-46
Configuring AsyncOS To Work With Multiple LDAP Servers 3-46
Testing Servers and Queries 3-47
Failover 3-47
Configuring the Cisco IronPort Appliance for LDAP Failover 3-47
Load Balancing 3-48
Configuring the Cisco IronPort Appliance for Load Balancing 3-48

CHAPTER 4 **Validating Recipients Using an SMTP Server** 4-1
SMTP Call-Ahead Recipient Validation: Overview 4-1
Configuring SMTP Call-Ahead Recipient Validation 4-3
Configuring the Call-Ahead Server Profile 4-3
SMTP Call-Ahead Server Profile Settings 4-4
Call Ahead Server Responses 4-5
Enabling the SMTP Call-Ahead Server Profile on a Public Listener 4-6
Configuring LDAP Routing Query Settings 4-7
SMTP Call-Ahead Query Routing 4-8
Bypassing SMTP Call-Ahead Validation 4-9

CHAPTER 5 **Email Authentication** 5-1
Email Authentication Overview 5-1
DomainKeys and DKIM Authentication: Overview 5-2

DomainKeys and DKIM Signing in AsyncOS 5-3
Configuring DomainKeys and DKIM Signing 5-4
Signing Keys 5-4
Exporting and Importing Signing Keys 5-4
Public Keys 5-5
Domain Profiles 5-5
Exporting and Importing Domain Profiles 5-6
Enabling Signing for Outgoing Mail 5-6
Enabling Signing for Bounce and Delay Messages 5-6
Configuring DomainKeys/DKIM Signing (GUI) 5-7
Creating Domain Profiles for DomainKeys Signing 5-8
Creating Domain Profiles for DKIM Signing 5-8
Creating New Signing Keys 5-11
Exporting Signing Keys 5-11
Importing or Entering Existing Signing Keys 5-12
Deleting Signing Keys 5-12
Generating a DNS Text Record 5-12
Testing Domain Profiles 5-13
Exporting Domain Profiles 5-14
Importing Domain Profiles 5-14
Deleting Domain Profiles 5-14
Searching Domain Profiles 5-14
Signing System-Generated Messages 5-15
Domain Keys and Logging 5-15
Configuring DKIM Verification 5-15
Managing DKIM Verification Profiles 5-16
Creating a DKIM Verification Profile 5-17
Exporting DKIM Verification Profiles 5-19
Importing DKIM Verification Profiles 5-19
Deleting DKIM Verification Profiles 5-19
Searching DKIM Verification Profiles 5-20
Configuring DKIM Verification on the Mail Flow Policy 5-20
DKIM Verification and Logging 5-20
Configuring an Action for DKIM Verified Mail 5-21
Overview of SPF and SIDF Verification 5-22
A Note About Valid SPF Records 5-22
Working with SPF on a Cisco IronPort Email Security Appliance 5-23
Enabling SPF and SIDF 5-24
Enabling SPF and SIDF via the CLI 5-25

The Received-SPF Header 5-30
Determining the Action to Take for SPF/SIDF Verified Mail 5-31
Verification Results 5-31
Using the spf-status Filter Rule in the CLI 5-32
spf-status Content Filter Rule in the GUI 5-33
Using the spf-passed Filter Rule 5-34
Testing the SPF/SIDF Results 5-34
Basic Granularity Test of SPF/SIDF Results 5-34
Greater Granularity Test of SPF/SIDF Results 5-35

CHAPTER 6 **Using Message Filters to Enforce Email Policies** 6-1
Overview 6-1
Components of a Message Filter 6-2
Message Filter Rules 6-2
Message Filter Actions 6-2
Message Filter Example Syntax 6-3
Message Filter Processing 6-3
Message Filter Order 6-4
Message Header Rules and Evaluation 6-4
Message Bodies vs. Message Attachments 6-5
Thresholds for Matches in Content Scanning 6-6
Threshold Scoring for Message Bodies and Attachments 6-6
Threshold Scoring Multipart/Alternative MIME Parts 6-7
Threshold Scoring for Content Dictionaries 6-7
AND Test and OR Tests in Message Filters 6-8
Message Filter Rules 6-9
Filter Rules Summary Table 6-9
Regular Expressions in Rules 6-15
Using Regular Expressions to Filter Messages 6-16
Guidelines for Using Regular Expressions 6-16
Regular Expression and Non-ASCII Character Sets 6-17
n Tests 6-17
Case-sensitivity 6-17
Writing Efficient Filters 6-18
PDFs and Regular Expressions 6-18
Smart Identifiers 6-19
Smart Identifier Syntax 6-19
Examples of Message Filter Rules 6-20
True Rule 6-20

Valid Rule 6-20
Subject Rule 6-21
Envelope Recipient Rule 6-21
Envelope Recipient in Group Rule 6-22
Envelope Sender Rule 6-22
Envelope Sender in Group Rule 6-23
Sender Group Rule 6-23
Body Size Rule 6-24
Remote IP Rule 6-24
Receiving Listener Rule 6-25
Receiving IP Interface Rule 6-25
Date Rule 6-25
Header Rule 6-26
Random Rule 6-27
Recipient Count Rule 6-28
Address Count Rule 6-28
Body Scanning Rule 6-28
Body Scanning 6-29
Encryption Detection Rule 6-29
Attachment Type Rule 6-30
Attachment Filename Rule 6-30
DNS List Rule 6-31
SenderBase Reputation Rule 6-32
Dictionary Rules 6-33
SPF-Status Rule 6-35
SPF-Passed Rule 6-36
Workqueue-count Rule 6-37
SMTP Authenticated User Match Rule 6-37
Signed Rule 6-39
Signed Certificate Rule 6-39
Message Filter Actions 6-42
Filter Actions Summary Table 6-42
Attachment Groups 6-45
Action Variables 6-48
Non-ASCII Character Sets and Message Filter Action Variables 6-49
Matched Content Visibility 6-50
Examples of Message Filter Actions 6-50
Skip Remaining Message Filters Action 6-50
Drop Action 6-51
Bounce Action 6-51

Encrypt Action 6-51
Notify and Notify-Copy Actions 6-52
Blind Carbon Copy Actions 6-54
Quarantine and Duplicate Actions 6-56
Alter Recipient Action 6-57
Alter Delivery Host Action 6-57
Alter Source Host (Virtual Gateway address) Action 6-58
Archive Action 6-59
Strip Header Action 6-60
Insert Header Action 6-60
Edit Header Text Action 6-61
Edit Body Text Action 6-61
HTML Convert Action 6-62
Bounce Profile Action 6-63
Bypass Anti-Spam System Action 6-63
Bypass Anti-Virus System Action 6-64
Bypass Outbreak Filter Scanning Action 6-64
Add Message Tag Action 6-64
Add Log Entry Action 6-65
Attachment Scanning 6-65
Message Filters for Scanning Attachments 6-66
Image Analysis 6-67
Configuring Scanning Values 6-68
Using the Image Analysis Message Filter 6-70
Using Image Analysis Content Filters 6-71
Notifications 6-72
Examples of Attachment Scanning Message Filters 6-72
Inserting Headers 6-72
Dropping Attachments by File Type 6-73
Dropping Attachments by Dictionary Matches 6-74
Quarantining Protected Attachments 6-74
Detecting Unprotected Attachments 6-75
Using the CLI to Manage Message Filters 6-75
Creating a New Message Filter 6-76
Deleting a Message Filter 6-77
Moving a Message Filter 6-77
Activating and Deactivating a Message Filter 6-77
Activating or Deactivating a Message Filter 6-80
Importing Message Filters 6-81
Exporting Message Filters 6-81

Viewing Non-ASCII Character Sets 6-81
Displaying a Message Filter List 6-82
Displaying Message Filter Details 6-82
Configuring Filter Log Subscriptions 6-82
Modifying Scanning Parameters 6-84
Using scanconfig 6-84
Changing Message Encoding 6-89
Creating Sample Message Filters 6-90
Message Filter Examples 6-97
Open-Relay Prevention Filter 6-97
Policy Enforcement Filters 6-97
Notify Based on Subject Filter 6-97
BCC and Scan Mail Sent to Competitors 6-98
Block Specific User Filter 6-98
Archive and Drop Messages Filter 6-98
Large "To:" Header Filter 6-98
Blank "From:" Filter 6-99
SRBS Filter 6-99
Alter SRBS Filter 6-100
Filename Regex Filter 6-100
Show SenderBase Reputation Score in Header Filter 6-100
Insert Policy into Header Filter 6-100
Too Many Recipients Bounce Filter 6-101
Routing and Domain Spoofing 6-101
Using Virtual Gateways Filter 6-101
Same Listener for Deliver and Injection Filter 6-101
Single Injector Filter 6-102
Drop Spoofed Domain Filter (Single Listener) 6-102
Drop Spoofed Domain Filter (Multiple Listeners) 6-102
Another Drop Spoofed Domain Filter 6-102
Detect Looping Filter 6-103

CHAPTER 7 **Advanced Network Configuration** 7-1
Media Settings on Ethernet Interfaces 7-1
Using etherconfig to Edit Media Settings on Ethernet Interfaces 7-1
Example of Editing Media Settings 7-2
Network Interface Card Pairing/Teaming 7-3
NIC Pairing and VLANs 7-4
NIC Pair Naming 7-4

Configuring and Testing NIC Pairing/Teaming 7-4
NIC Pairing and Existing Listeners 7-4
Enabling NIC Pairing via the etherconfig Command 7-5
Using the failover Subcommand for NIC Pairing 7-6
Verifying NIC Pairing 7-8
Virtual Local Area Networks (VLANs) 7-8
VLANs and Physical Ports 7-9
Managing VLANs 7-10
Creating a New VLAN via the etherconfig Command 7-10
Creating an IP Interface on a VLAN via the interfaceconfig Command 7-12
Direct Server Return 7-15
Enabling Direct Server Return 7-15
Enabling the Loopback Interface via the etherconfig Command 7-16
Creating an IP Interface on Loopback via the interfaceconfig Command 7-17
Creating a Listener on the New IP Interface 7-19

CHAPTER 8 **Centralized Management** 8-1
Cluster Requirements 8-1
Cluster Organization 8-2
Initial Configuration Settings 8-3
Creating and Joining a Cluster 8-4
The clusterconfig Command 8-4
Joining an Existing Cluster 8-6
Joining an Existing Cluster over SSH 8-6
Joining an Existing Cluster over CCS 8-8
Adding Groups 8-10
Managing Clusters 8-11
Administering a Cluster from the CLI 8-11
Copying and Moving Settings 8-11
Experimenting with New Configurations 8-12
Leaving a Cluster Permanently (Removal) 8-13
Upgrading Machines in a Cluster 8-13
Configuration File Commands 8-14
Resetting the Configuration 8-14
CLI Command Support 8-14
All Commands Are Cluster-aware 8-14
The commit and clearchanges Commands 8-14
New Operation Added 8-15
Restricted Commands 8-15

Administering a Cluster from the GUI 8-16
Cluster Communication 8-19
DNS and Hostname Resolution 8-19
Clustering, Fully Qualified Domain Names, and Upgrading 8-19
Cluster Communication Security 8-19
Cluster Consistency 8-20
Disconnect/Reconnect 8-21
Interdependent Settings 8-22
Best Practices and Frequently Asked Questions 8-23
Best Practices 8-23
Copy vs. Move 8-24
Good CM Design Practices 8-24
Procedures: Configuring an Example Cluster 8-25
Summary of GUI Options for Using CM Settings Other Than the Cluster Default 8-26
Setup and Configuration Questions 8-27
General Questions 8-28
Network Questions 8-28
Planning and Configuration 8-29

APPENDIX A **AsyncOS Quick Reference Guide** A-1

APPENDIX B **Accessing the Appliance** B-1
FTP Access B-1
Secure Copy (scp) Access B-4
Accessing via a Serial Connection B-5

INDEX



Preface

The *Cisco IronPort AsyncOS 7.6 for Email Advanced Configuration Guide* provides instructions for setting up, administering, and monitoring the Cisco IronPort Email Security appliance. These instructions are designed for an experienced system administrator with knowledge of networking and email administration.

Before you Read this Book

Read the *Quickstart Guide* and any product notes that were shipped with your appliance. This guide assumes that you have unpacked, physically installed into a rack cabinet, and powered-up the appliance.

Note If you have already cabled your appliance to your network, ensure that the default IP address for the Cisco IronPort appliance does not conflict with other IP addresses on your network. The IP address assigned to the Management port by the factory is `192.168.42.42`. See the "Setup and Installation" chapter of the *Cisco IronPort AsyncOS for Email Configuration Guide* and Appendix B, "Accessing the Appliance" for more information about assigning IP addresses to the Cisco IronPort appliance.

Documentation Set

The *AsyncOS* documentation set has been divided into four separate books — the *Cisco IronPort AsyncOS for Email Configuration Guide*, the *Cisco IronPort AsyncOS CLI Reference Guide*, the *Cisco IronPort AsyncOS for Email Daily Management Guide*, and this book, which contains information on advanced features and configuration. Occasionally, this book will refer to the other guides for additional information on topics.

How This Book Is Organized

Chapter 1, "Customizing Listeners" describes the process for tailoring the configuration of your Enterprise Email Gateway. This chapter discusses, in detail, advanced features available to you as you configure interfaces and listeners to handle email receiving through the gateway.

Chapter 2, "Configuring Routing and Delivery Features" explains the features that affect email routing and delivery of email traveling through the Cisco IronPort appliance.

Chapter 3, "LDAP Queries" described how your Cisco IronPort appliance can connect to your corporate Lightweight Directory Access Protocol (LDAP) servers and perform queries for the purposes of verifying recipients to accept (including group membership), mail routing and address rewriting. masquerading headers, and supporting for SMTP authentication.

Chapter 5, "Email Authentication" details the process of configuring and enabling email authentication on a Cisco IronPort appliance. Cisco IronPort AsyncOS supports several types of email authentication, including Sender Policy Framework (SPF), Sender ID Framework (SIDF), and DomainKeys Identified Mail (DKIM) verification of incoming mail, as well as DomainKeys and DKIM signing of outgoing mail.

Chapter 6, "Using Message Filters to Enforce Email Policies" describes how to use Message Filters to define rules for handling email, including the ability to modify the content of messages through the attachment filtering, image analysis, and content dictionary features.

Chapter 7, "Advanced Network Configuration" includes information about NIC pairing, virtual LANs and more.

Chapter 8, "Centralized Management" describes the centralized management feature, which allows you to manage and configure multiple appliances. The centralized management feature provides increased reliability, flexibility, and scalability within your network, allowing you to manage globally while complying with local policies.

Appendix A, "AsyncOS Quick Reference Guide" provides a quick reference for most commands in the CLI.

Appendix B, "Accessing the Appliance"describes how to access the Cisco IronPort appliance to send and retrieve files from Cisco IronPort appliance.

Typographic Conventions

Typeface or Symbol	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output.	`Please choose an IP interface for this Listener.` The `sethostname` command sets the name of the Cisco IronPort appliance.
AaBbCc123	What you type, when contrasted with on-screen computer output.	`mail3.example.com>` **`commit`** `Please enter some comments describing your changes:` `[]>` **`Changed the system hostname`**
AaBbCc123	Book titles, new words or terms, words to be emphasized. Command line variable; replace with a real name or value.	Read the *Cisco IronPort Quickstart Guide*. The Cisco IronPort appliance *must* be able to uniquely select an interface to send an outgoing packet. `Before you begin, please reset your password to a new value.` `Old password:` **`ironport`** `New password:` *`your_new_password`* `Retype new password:` *your_new_password*

Contacting Cisco IronPort Customer Support

You can request our support by phone, email or online 24 hours a day, 7 days a week.

During customer support hours (24 hours per day, Monday through Friday excluding U.S. holidays), an engineer will contact you within an hour of your request.

To report a critical issue that requires urgent assistance outside of our office hours, please contact Cisco IronPort using one of the following methods:

U.S. toll-free: 1(877) 641- 4766

International: http://cisco.com/web/ironport/contacts.html

Support Portal: http://cisco.com/web/ironport/index.html

Cisco IronPort Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

`docfeedback@ironport.com.`

Please include the following part number in the subject of your email: OL-25137-01.



CHAPTER 1

Customizing Listeners

In the *Cisco IronPort AsyncOS for Email Configuration Guide*, you learned how the Cisco IronPort AsyncOS operating system allows the Cisco IronPort appliance to function as the inbound email gateway for your enterprise, servicing SMTP connections from the Internet, accepting messages, and relaying messages to the appropriate systems by enabling *listeners* to service these connections.

A *listener* describes an email processing service that will be configured on a particular IP interface. Listeners only apply to email entering the Cisco IronPort appliance — either from the internal systems within your network or from the Internet. Cisco IronPort AsyncOS uses listeners to specify criteria that messages must meet in order to be accepted and relayed to recipient hosts. You can think of a listener as an "email injector" or even a "SMTP daemon" running on a specific port for each IP address you specify (including the initial addresses you configured with the System Setup Wizard or `systemsetup` command).

Note If you have completed the GUI's System Setup Wizard (or the Command Line Interface `systemsetup` command) as described in the "Setup and Installation" chapter of the *Cisco IronPort AsyncOS for Email Configuration Guide* and committed the changes, at least one listener should already be configured on your appliance.

This chapter describes how to use the Listeners page on the Network menu in the GUI or the `listenerconfig` CLI command to customize some of the advanced *receiving* properties of listeners configured on your Cisco IronPort appliance, including creating new listeners. The following chapter, Chapter 2, "Configuring Routing and Delivery Features" describes how to customize the delivery properties of listeners configured on the system.

The following topics are described:

- Listeners Overview, page 1-2
- Configuring Listeners via the GUI, page 1-3
- Configuring Listeners via the CLI, page 1-14
- SenderBase Settings and HAT Mail Flow Policies, page 1-16
  - HAT Significant Bits Feature, page 1-18
- Encrypting SMTP Conversations Using TLS, page 1-22

Listeners Overview

The Network > Listeners page and the `listenerconfig` command in the CLI allow you to create, edit, and delete a listener. Cisco IronPort AsyncOS requires that you specify criteria that messages must meet in order to be accepted and then relayed to recipient hosts — either internal to your network or to external recipients on the Internet.

These qualifying criteria are defined in listeners; collectively, they ultimately define and enforce your mail flow policies. Listeners also define how the Cisco IronPort appliance communicates with the system that is injecting email.

Each listener is composed of the criteria shown in Table 1-1.

Table 1-1 Criteria for Listeners

Name	Unique nickname you supply for the listener, for future reference. The names you define for listeners are case-sensitive. AsyncOS will not allow you to create two identical listener names.	
IP interface	Listeners are assigned to IP interfaces. The IP interface is defined by the `interfaceconfig` command. Any IP interfaces must be configured using the System Setup Wizard or the `systemsetup` command or the IP Interfaces page (or the `interfaceconfig` command) *before* you create and assign a listener to it. The version of the Internet Protocol address of the interface determines the type of traffic the listener accepts. If the IP interface has both an IPv4 and IPv6 address, then the listener can accept connections from both IPv4 and IPv6 addresses.	
Mail protocol	The mail protocol to used for email receiving: either SMTP or QMQP (only available via the `listenerconfig` command in the CLI).	
IP port	The specific IP port used for connections to the listener. By default, SMTP uses port 25 and QMQP uses port 628.	
Listener Type:	**Public**	Public and private listeners are used for most configurations. By convention, private listeners are intended to be used for private (internal) networks, while public listeners contain default characteristics for receiving email from the Internet.
	Private	
	Blackhole	"Blackhole" listeners can be used for testing or troubleshooting purposes. When you create a blackhole listener, you choose whether messages are written to disk or not before they are deleted. (See "Testing and Troubleshooting" in the *Cisco IronPort AsyncOS for Email Daily Management Guide* for more information.) Writing messages to disk before deleting them can help you measure the rate of receiving and the speed of the queue. A listener that doesn't write messages to disk can help you measure the pure rate of receiving from your message generation systems. This listener type is only available through the `listenerconfig` command in the CLI.

In addition to these criteria, you can also configure the following for each listener:

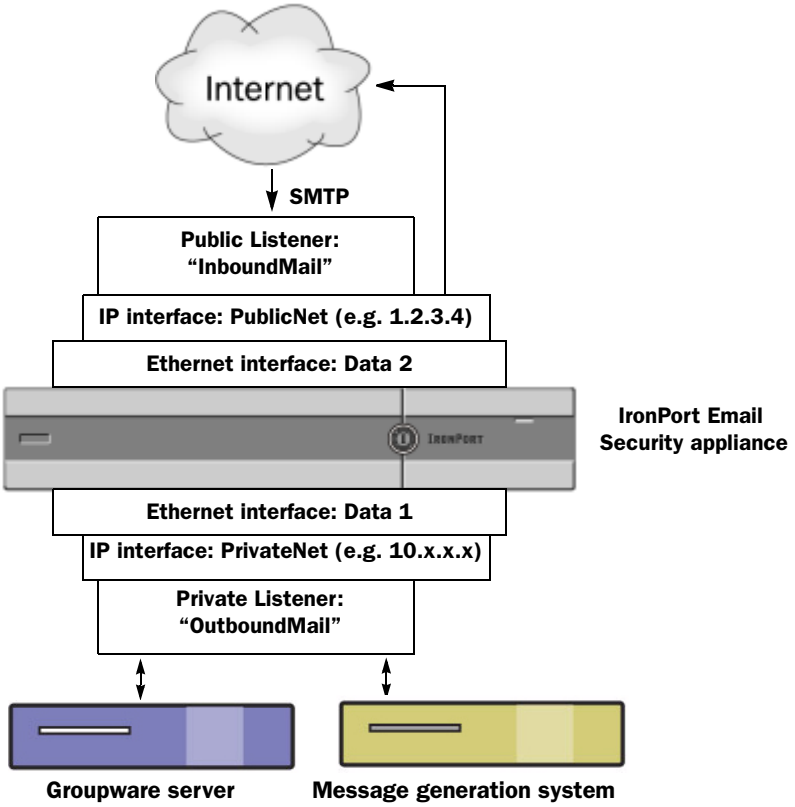
- SMTP Address Parsing Options (optional settings for controlling parsing in SMTP "MAIL FROM" and "RCPT TO," see SMTP Address Parsing Options, page 1-8)
- Advanced Configuration Options (optional settings for customizing the behavior of the Listener, see Advanced Configuration Options, page 1-11)
- LDAP Options (optional settings for controlling LDAP queries associated with this Listener, see LDAP Options, page 1-12)

Further, there are global settings that apply to all listeners, see Global Settings for Listeners, page 1-4 for more information.

When you create a listener, you specify the hosts that are allowed to connect to the listener through the Host Access Table (HAT). For public listeners, you also define all domains that the appliance will accept messages for using the Recipient Access Table (RAT). RATs apply *only* to public listeners. For more information about Host Access Table and Recipient Access Table entries, see the "Configuring the Gateway to Receive Mail" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

Figure 1-1 illustrates public and private listeners as they can be used with a Cisco IronPort appliance configured as an Enterprise Gateway. See also "Enterprise Gateway Configuration" in the *Cisco IronPort AsyncOS for Email Configuration Guide* for more information.

Figure 1-1 Public and Private Listeners in an Enterprise Gateway Configuration



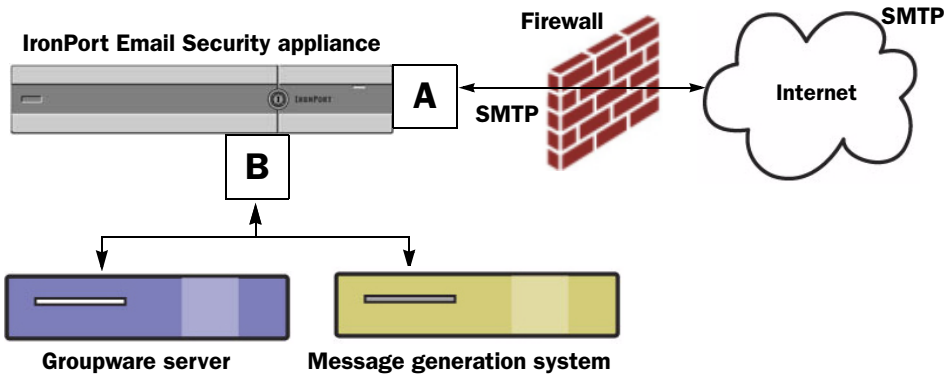
Configuring Listeners via the GUI

Use the Listeners page on the Network menu in the GUI to add additional listeners to your list of currently configured listeners.

Note If you have completed the GUI's System Setup Wizard (or the Command Line Interface `systemsetup` command) as described in the "Setup and Installation" chapter of the *Cisco IronPort AsyncOS for Email Configuration Guide* and committed the changes, at least one listener should already be configured on your appliance. (Refer to the settings you entered in the "Create a Listener" section of the GUI System Setup Wizard or the `systemsetup` command in the CLI.) The specific addresses to accept mail for were also entered at that time, as well as the first SMTP routes entry.

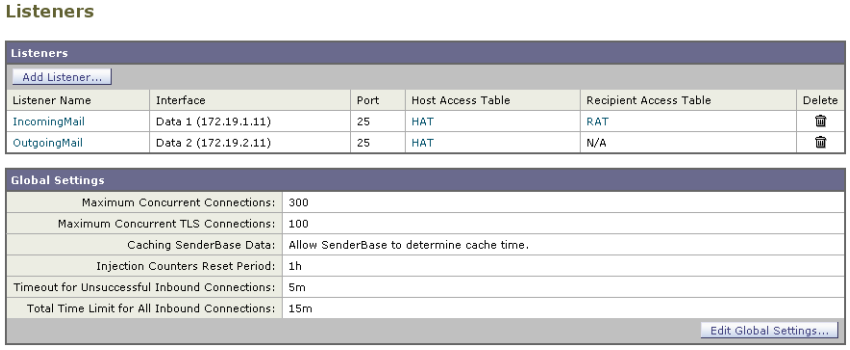
In Figure 1-2, listener A refers to the public listener named InboundMail created during System Setup. Listener B refers to the optional private listener that you also may have created.

Figure 1-2 Creating a New Private Listener



Use the Network > Listeners page to add, delete, or modify listeners. The Listeners page also provides access to the global settings for listeners.

Figure 1-3 Listeners Page



Global Settings for Listeners

Global settings for the listeners affect all of the listeners that are configured on the Cisco IronPort appliance. If the listener uses an interface that has both Internet Protocol version 4 (IPv4) and version 6 (IPv6) addresses, the listener settings apply to both IPv4 and IPv6 traffic.

Global settings include:

Table 1-2 Listener Global Settings

Global Setting	Description
Maximum Concurrent Connections	Set the maximum number of concurrent connections for listeners. The default value is 300. If the listener accepts both IPv4 and IPv6 connections, the number of connections is divided between the two. For example, if the maximum concurrent connections is 300, then the sum of IPv4 and IPv6 connections cannot exceed 300.
Maximum Concurrent TLS Connections	Set the maximum concurrent TLS connections across all listeners combined. The default value is 100. If the listener accepts both IPv4 and IPv6 TLS connections, the number of connections is divided between the two. For example, if the maximum concurrent connections is 100, then the sum of IPv4 and IPv6 TLS connections cannot exceed 100.
Caching SenderBase Data	You can allow the SenderBase information service to determine the cache time (which is recommended), or you can specify your own cache time. You can also disable caching.
Injection Counters Reset Period	Allows you to adjust when the injection control counters are reset. For very busy systems maintaining counters for a very large number of different IP addresses, configuring the counters to be reset more frequently (for example, every 15 minutes instead of every 60 minutes) will ensure that the data does not grow to an unmanageable size and impact system performance. The current default value is 1 hour. You can specify periods ranging from as little as 1 minute (60 seconds) to as long as 4 hours (14,400 seconds). See Injection Control Periodicity, page 1-19.
Timeout Period for Unsuccessful Inbound Connections	Set the length of time AsyncOS will allow an unsuccessful inbound connection to remain intact before closing it. An unsuccessful connection can be an SMTP conversation in which SMTP or ESMTP commands continue to be issued without a successful message injection occurring. When the specified timeout is reached, the behavior is to send an error and disconnect: “421 Timed out waiting for successful message injection, disconnecting.” A connection is considered unsuccessful until it successfully injects a message. Only available for SMTP connections on public listeners. The default value is 5 minutes.

Table 1-2 Listener Global Settings

Global Setting	Description
Total Time Limit for All Inbound Connections	Set the length of time AsyncOS will allow an inbound connection to remain intact before closing it. This setting is intended to preserve system resources by enforcing a maximum allowable connection time. Once about 80% of this maximum connection time is reached the following message is issued: "421 Exceeded allowable connection time, disconnecting." The appliance will attempt to disconnect when the connection exceeds 80% of the maximum connection time in order to prevent disconnecting mid-message. It is likely that a problem is occurring with the inbound connection if it is open long enough to reach 80% of the maximum connection time. Keep this threshold in mind when specifying the time limit. Only available for SMTP connections on public listeners. The default value is 15 minutes.
HAT delayed rejections	Configure whether to perform HAT rejection at the message recipient level.By default, HAT rejected connections will be closed with a banner message at the start of the SMTP conversation. When an email is rejected due to HAT "Reject" settings, AsyncOS can perform the rejection at the message recipient level (RCPT TO), rather than at the start of the SMTP conversation. Rejecting messages in this way delays the message rejection and bounces the message, allowing AsyncOS to retain more detailed information about the rejected messages. For example, you can see the mail from address and each recipient address of the message which is blocked. Delaying HAT rejections also makes it less likely that the sending MTA will perform multiple retries. When you enable HAT delayed rejection, the following behavior occurs: --The MAIL FROM command is accepted, but no message object is created. --All RCPT TO commands are rejected with text explaining that access to send e-mail is refused. --If the sending MTA authenticates with SMTP AUTH, they are granted a RELAY policy and are allowed to deliver mail as normal. NOTE: Only configurable from the CLI `listenerconfig --> setup` command.

Settings for messages containing multiple encodings: localeconfig

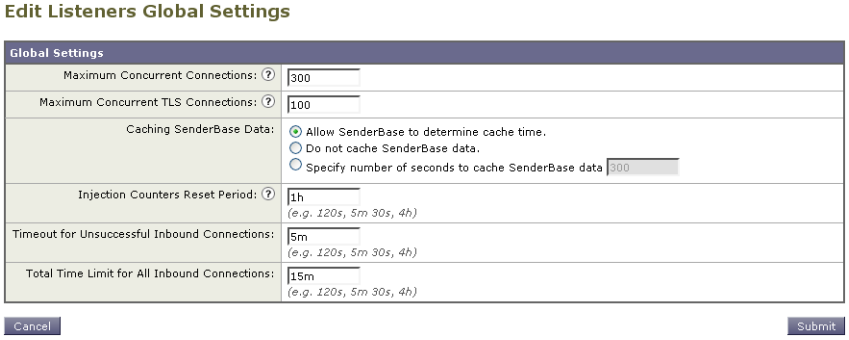
You can set the behavior of AsyncOS regarding modifying the encoding of message headings and footers during message processing. This setting is not configured via the GUI. Instead, it is configured via the `localeconfig` in the CLI.

Configuring Global Settings for Listeners

To edit global settings for listeners:

Step 1 On the Network > Listeners page, click **Edit Global Settings** The Edit Listeners Global Settings page is displayed:

Figure 1-4 Edit Listeners Global Settings Page



Step 2 Make changes to the settings and click **Submit**.

Step 3 The Listeners page is displayed, reflecting the changes made.

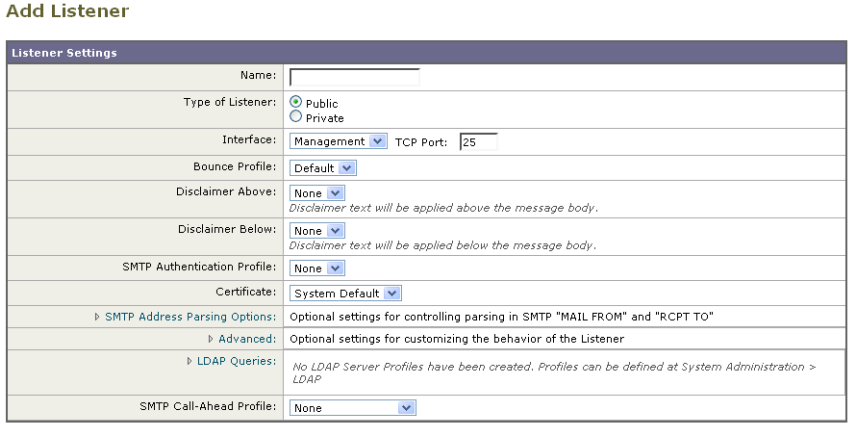
Step 4 Commit the changes.

Creating Listeners

To add a new listener:

Step 1 Click **Add Listener** on the Network > Listener page. The Add Listener page is displayed:

Figure 1-5 Add Listener Page



Step 2 Enter a name for the listener.

Step 3 Select the type of listener.

Step 4 Select the interface and TCP port on which to create the listener. Depending on the version of the IP address used by the interface, the listener accepts connections from IPv4 addresses, IPv6 addresses or from both versions.

Step 5 Select a bounce profile (bounce profiles created via the `bounceconfig` command in the CLI are available in the list, see Creating a New Bounce Profile, page 2-40).

Step 6 Select a disclaimer to attach above or below emails (disclaimers created via the Mail Policies > Text Resources page or the `textconfig` command in the CLI are available in the list, see the "Text Resources" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide*).

Step 7 Specify an SMTP Authentication profile.

Step 8 Specify a certificate for TLS connections to the listener (certificates added via the Network > Certificates page or the `certconfig` command in the CLI are available in the list, see Encrypting SMTP Conversations Using TLS, page 1-22).

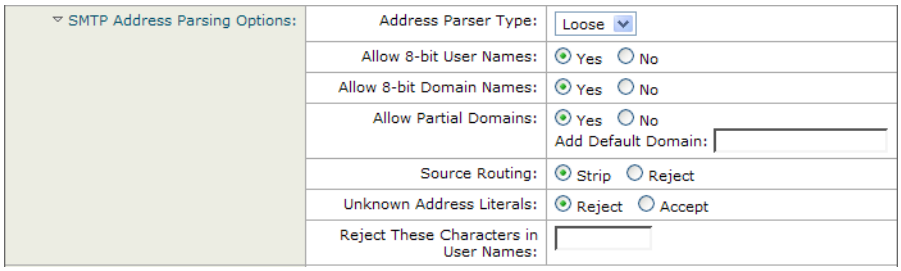
Step 9 Configure any optional SMTP Address Parsing, Advanced, and LDAP options (discussed in detail below).

Step 10 Submit and commit your changes.

SMTP Address Parsing Options

To access the SMTP address parsing options, expand the section by clicking on SMTP Address Parsing in the listing.

Figure 1-6 Listeners SMTP Address Parsing Options



SMTP address parsing controls how strict the AsyncOS address parser behaves for the SMTP "MAIL FROM" and "RCPT TO" commands. SMTP address parsing has two modes: Strict and Loose as well as several other parsing options (configured independently of the address parsing mode).

Selecting a parsing mode or type is deciding whether you want your appliance to strictly adhere to the standards of RFC2821.

Strict Mode

Strict mode tries to follow RFC 2821. In Strict mode, the address parser follows RFC 2821 rules with the following exceptions/enhancements:

- Space is allowed after the colon, as in "MAIL FROM: <joe@example.com>".
- Underscores are allowed in the domain name.
- "MAIL FROM" and "RCPT TO" commands are case-insensitive.
- Periods are not treated specially (for example, RFC 2821 does not allow a username of "J.D.").

Some of the additional options below may be enabled which technically would violate RFC 2821.

Loose Mode

The loose parser is basically the existing behavior from previous versions of AsyncOS. It does its best to "find" an email address and:

- Ignores comments. It supports nested comments (anything found in parenthesis) and ignores them.
- Does not require angle brackets around email addresses provided in "RCPT TO" and "MAIL FROM" commands.
- Allows multiple nested angle brackets (it searches for the email address in the deepest nested level).

Additional Options

In addition to the two parsing modes, you can specify behavior for the following additional items listed in Table 1-3.

Table 1-3 SMTP Address Parsing Additional Options

Option	Description	Default
Allow 8-bit username	If enabled, allow 8-bit characters in the username portion of the address without escaping.	on
Allow 8-bit domain	If enabled, allow 8-bit characters in the domain portion of the address.	on

Table 1-3 SMTP Address Parsing Additional Options

Option	Description	Default
Allow partial domain **Add Default Domain**	If enabled, will allow partial domains. Partial domains can be no domain at all, or a domain with no dots. The following addresses are examples of partial domains: – foo – foo@ – foo@bar This option *must* be enabled in order for the Default Domain feature to work properly. Add Default Domain: A default domain to use for email addresses without a fully qualified domain name. This option is disabled unless Allow Partial Domains is enabled in SMTP Address Parsing options (see SMTP Address Parsing Options, page 1-8). This affects how a listener modifies email that it relays by adding the "default sender domain" to sender and recipient addresses that do not contain fully-qualified domain names. (In other words, you can customize how a listener handles "bare" addresses). If you have a legacy system that sends email without adding (appending) your company's domain to the sender address, use this to add the default sender domain. For example, a legacy system may automatically create email that only enters the string "`joe`" as the sender of the email. Changing the default sender domain would append "`@yourdomain.com`" to "`joe`" to create a fully-qualified sender name of `joe@yourdomain.com`.	on
Source routing: reject, strip	Determines behavior if source routing is detected in the "MAIL FROM" and "RCPT TO" addresses. Source routing is a special form of an email address using multiple '@' characters to specify routing (for example: @one.dom@two.dom:joe@three.dom). If set to "reject," the address will be rejected. If "strip," the source routing portion of the address will be deleted, and the message will be injected normally.	Strip

Table 1-3 SMTP Address Parsing Additional Options

Option	Description	Default
Reject User Names containing These Characters:	Usernames that include characters (such as % or !, for example) entered here will be rejected.	%!:@
Unknown Address Literals (IPv6, etc.): reject, accept	Determines behavior for when an address literal is received that the system can not handle. Currently, this is everything except for IPv4. Thus, for example, for an IPv6 address literal, you can either reject it at the protocol level, or accept it and immediately hard bounce it. Recipient addresses containing literals will cause an immediate hard bounce. Sender addresses may get delivered. If the message cannot be delivered, then the hard bounce will hard bounce (double hard bounce). In the case of reject, both sender and recipient addresses will be rejected immediately at the protocol level.	reject

Partial Domains, Default Domains, and Malformed MAIL FROMs

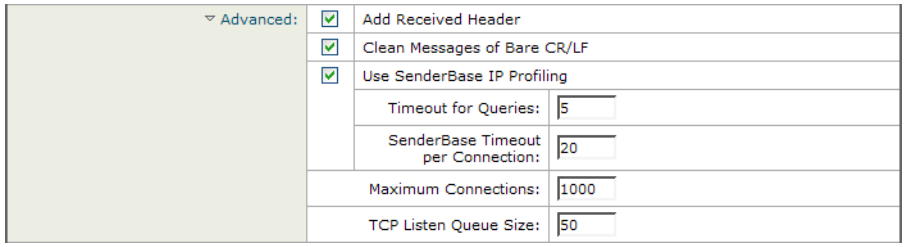
If you enable envelope sender verification or disable allowing partial domains in SMTP Address Parsing options for a listener, the default domain settings for that listener will no longer be used.

These features are mutually exclusive.

Advanced Configuration Options

To access the Advanced options, expand the section by clicking on Advanced in the listing.

Figure 1-7 Listeners Advanced Options



Advanced configuration options include:

- Add Received Header: Add a received header to all received email. A listener also modifies email that it relays by adding a Received: header on each message. If you do not want to include the Received: header, you can disable it using this option.

Note The Received: header is not added to the message within the work queue processing. Rather, it is added when the message is enqueued for delivery.

Disabling the received header is a way to ensure that your network's topology is not exposed by revealing the IP addresses or hostnames of internal servers on any messages travelling outside your infrastructure. Please use caution when disabling the received header.

- Change bare CR and LF characters to CRLF: New feature, converts bare CR and LF characters to CRLF characters
- Use SenderBase IP Profiling
  - Timeout for Queries
  - SenderBase Timeout per Connection
- Maximum Connections
- TCP Listen Queue Size (the backlog of connections that AsyncOS will manage before the SMTP server accepts them)

LDAP Options

To access the LDAP options, expand the section by clicking on LDAP Options in the listing.

The LDAP options settings for listeners are used to enable LDAP queries on the listener. You must create the LDAP query first, before using this option. Each type of query (Accept, Routing, Masquerade, Group) has a separate subsection. Click the type of query to expand the subsection.

For more information about creating LDAP queries, see LDAP Queries, page 3-1.

Accept Queries

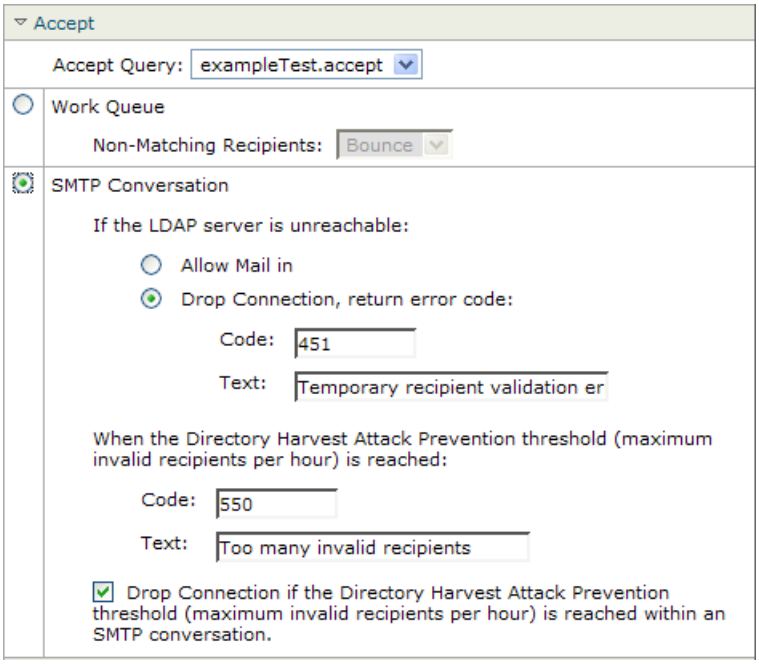
For Accept queries, select the query to use from the list. You can specify whether the LDAP Accept occurs during the work queue processing or during the SMTP conversation.

For LDAP Accept during the work queue processing, specify the behavior for non-matching recipients: bounce or drop.

For LDAP Accept during the SMTP conversation, specify how to handle mail if the LDAP server is unreachable. You can elect to allow messages or drop the connection with a code and custom response. Finally, select whether or not to drop connections if the Directory Harvest Attack Prevention (DHAP) threshold is reached during an SMTP conversation.

Performing recipient validation in the SMTP conversation can potentially reduce the latency between multiple LDAP queries. Therefore, you might notice an increased load on your directory server when you enable conversational LDAP Accept.

Figure 1-8 Listeners Accept Query Options



See Overview, page 3-1 for more information.

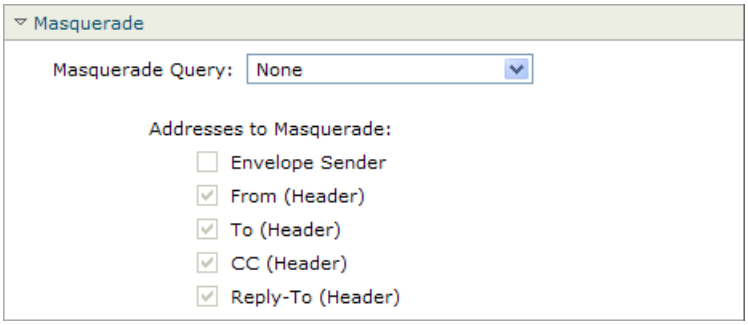
Routing Queries

For routing queries, select the query from the list. See Overview, page 3-1 for more information.

Masquerade Queries

For masquerade queries, select a query from the list, and select which address to masquerade.

Figure 1-9 Listeners Masquerade Query Options



See Overview, page 3-1 for more information.

Group Queries

For group queries, select the query from the list. See Overview, page 3-1 for more information.

Editing Listeners

To edit a listener:

Step 1 Click on the listener's name in the listing on the Network > Listeners page.

Step 2 Make changes to the listener.

Step 3 Submit and commit your changes.

Deleting Listeners

To delete a listener:

Step 1 Click the trash can icon in the Delete column for the corresponding listener on the Network > Listeners page.

Step 2 Confirm the deletion.

Step 3 Commit your changes.

Configuring Listeners via the CLI

Figure 1-9 lists some of the `listenerconfig` subcommands used in the tasks involved in creating and editing listeners.

Table 1-4 Tasks for Creating Listeners

Tasks for Creating Listeners	Command(s) and Subcommands	Reference
Create a new listener	`listenerconfig -> new`	
Edit global settings for listeners	`listenerconfig -> setup`	Global Settings for Listeners, page 1-4
Specify a bounce profile for the listener	`bounceconfig, listenerconfig -> edit -> bounceconfig`	Creating a New Bounce Profile, page 2-40
Associate a disclaimer with the listener	`textconfig, listenerconfig -> edit -> setup -> footer`	Covered in *Cisco IronPort AsyncOS for Email Configuration Guide*
Configure an SMTP Authentication	`smtpauthconfig, listenerconfig -> smtpauth`	
Configure SMTP address parsing	`textconfig, listenerconfig -> edit -> setup -> address`	
Configure a default domain for the listener	`listenerconfig -> edit -> setup -> defaultdomain`	
Add a received header to email	`listenerconfig -> edit -> setup -> received`	
Change bare CR and LF characters to CRLF	`listenerconfig -> edit -> setup -> cleansmtp`	
Modify the Host Access Table	`listenerconfig -> edit -> hostaccess`	Covered in *Cisco IronPort AsyncOS for Email Configuration Guide*

Table 1-4 Tasks for Creating Listeners

Accept email for local domains or specific users (RAT) (public listeners only)	`listenerconfig -> edit -> rcptaccess`	Covered in *Cisco IronPort AsyncOS for Email Configuration Guide*
Encrypt conversations on listeners (TLS)	`certconfig, settls, listenerconfig -> edit`	Encrypting SMTP Conversations Using TLS, page 1-22
Choose the certificate (TLS)	`listenerconfig -> edit -> certificate`	Encrypting SMTP Conversations Using TLS, page 1-22

See Chapter 2, "Configuring Routing and Delivery Features" for information about email routing and delivery configurations.

Advanced HAT Parameters

Table 1-5 defines the syntax of advanced HAT parameters. Note that for the values below which are numbers, you can add a trailing **k** to denote kilobytes or a trailing **M** to denote megabytes. Values with no letters are considered bytes. Parameters marked with an asterisk support the variable syntax shown in Table 1-5.

Table 1-5 Advanced HAT Parameter Syntax

Parameter	Syntax	Values	Example Values
Maximum messages per connection	`max_msgs_per_session`	Number	`1000`
Maximum recipients per message	`max_rcpts_per_msg`	Number	`10000` `1k`
Maximum message size	`max_message_size`	Number	`1048576` `20M`
Maximum concurrent connections allowed to this listener	`max_concurrency`	Number	`1000`
SMTP Banner Code	`smtp_banner_code`	Number	`220`
SMTP Banner Text (*)	`smtp_banner_text`	String	`Accepted`
SMTP Reject Banner Code	`smtp_banner_code`	Number	`550`
SMTP Reject Banner Text (*)	`smtp_banner_text`	String	`Rejected`
Override SMTP Banner Hostname	`use_override_hostname`	`on | off | default`	`default`
	`override_hostname`	String	`newhostname`
Use TLS	`tls`	`on | off | required`	`on`
Use anti-spam scanning	`spam_check`	`on | off`	`off`
Use virus scanning	`virus_check`	`on | off`	`off`

Table 1-5 Advanced HAT Parameter Syntax

Parameter	Syntax	Values	Example Values
Maximum Recipients per Hour	`max_rcpts_per_hour`	Number	`5k`
Maximum Recipients per Hour Error Code	`max_rcpts_per_hour_code`	Number	`452`
Maximum Recipients per Hour Text (*)	`max_rcpts_per_hour_text`	String	`Too many recipients`
Use SenderBase	`use_sb`	`on | off`	`on`
Define SenderBase Reputation Score	`sbrs[`*`value1`*`:`*`value2`*`]`	`-10.0- 10.0`	`sbrs[-10:-7.5]`
Directory Harvest Attack Prevention: Maximum Invalid Recipients Per Hour	`dhap_limit`	Number	`150`

SenderBase Settings and HAT Mail Flow Policies

In order to classify connections to the appliance and apply mail flow policies (which may or may not contain rate limiting), a listener's Host Access Table (HAT) uses the following methodology:

Classification -> Sender Group -> Mail Flow Policy -> Rate Limiting

For more information, refer to "Sender Groups Defined by Network Owners, Domains, and IP Addresses" in the "Configuring the Gateway to Receive Email" chapter of the *Cisco IronPort AsyncOS for Email Configuration Guide*.

The "Classification" stage uses the sending host's IP address to classify an inbound SMTP session (received on a public listener) into a Sender Group. The Mail Flow Policy associated with that Sender Group may have parameters for rate limiting enabled. (Rate limiting limits the maximum number of messages per session, the maximum number of recipients per message, the maximum message size, and/or the maximum number of concurrent connections you are willing to accept from a remote host.)

Normally, in this process, recipients are counted against each sender in the corresponding named sender group. If mail is received from several senders in the same hour, the total recipients for all senders is compared against the limit.

There are some exceptions to this counting methodology:

Step 1 If the classification is done by Network Owner, then the SenderBase Information Service will automatically divide a large block of addresses into smaller blocks.

Counting of recipients and recipient rate limiting is done separately for each of these smaller blocks (usually, but not always, the equivalent of a /24 CIDR block).

Step 2 If the HAT Significant Bits feature is used. In this case, a large block of addresses may be divided into smaller blocks by applying the significant bits parameter associated with the policy.

Note that this parameter relates to the **Mail Flow Policy -> Rate Limiting** phase. It is not the same as the "bits" field in the "network/bits" CIDR notation that may be used to classify IP addresses in a Sender Group.

By default, SenderBase Reputation Filters and IP Profiling support are *enabled* for public listeners and *disabled* for private listeners.

Timeouts for SenderBase Queries

The method by which queries to the SenderBase information service — for both SenderBase DNS queries and SenderBase Reputation Service Scores (SBRS Scores) — are configured has been improved beginning with the 4.0 release of AsyncOS. Previously, the configurable timeout value maximum of 5 seconds could cause a delay in mail processing for some Cisco IronPort appliances experiencing heavy load if the SenderBase information services were unreachable or unavailable.

The new timeout value can be configured first by issuing the `listenerconfig -> setup` command to change the global settings for caching SenderBase information service data. You can allow the SenderBase information service to determine the cache time (which is recommended), or you can specify your own cache time. You can also disable caching.

You enable "look ups" to the SenderBase Information Service in the `listenerconfig -> setup` command for listeners.

In this example, the feature is enabled and the default timeout values (for queries and for all queries per connection) are accepted:

```
Would you like to enable SenderBase Reputation Filters and IP Profiling

support? [Y]> y



Enter a timeout, in seconds, for SenderBase queries. Enter '0' to

disable SenderBase Reputation Filters and IP Profiling.

[5]>



Enter a timeout, in seconds, for all SenderBase queries per connection.

[20]>
```

Then, for each mail flow policy, you allow "look ups" to the SenderBase Information service on a per-mail flow policy basis using the `listenerconfig -> hostaccess -> edit` command:

```
Would you like to use SenderBase for flow control by default?  (Yes/No/Default) [Y]>
```

In the GUI:

Figure 1-10 Enabling SenderBase for Mail Flow Policies



HAT Significant Bits Feature

Beginning with the 3.8.3 release of AsyncOS, you can track and rate limit incoming mail on a per-IP address basis while managing sender group entries in a listener's Host Access Table (HAT) in large CIDR blocks. For example, if an incoming connection matched against the host "10.1.1.0/24," a counter could still be generated for each individual address within that range, rather than aggregating all traffic into one large counter.

Note In order for the significant bits HAT policy option to take effect, you *must* not enable "User SenderBase" in the Flow Control options for the HAT (or, for the CLI, answer **no** to the question for enabling the SenderBase Information Service in the `listenerconfig` -> setup command: "Would you like to enable SenderBase Reputation Filters and IP Profiling support?"). That is, the Hat Significant Bits feature and enabling SenderBase IP Profiling support are mutually exclusive.

In most cases, you can use this feature to define sender groups *broadly* — that is, large groups of IP addresses such as "10.1.1.0/24" or "10.1.0.0/16" — while applying mail flow rate limiting *narrowly* to smaller groups of IP addresses.

The HAT Significant Bits feature corresponds to these components of the system:

HAT Configuration

There are two parts of HAT configuration: sender groups and mail flow policies. Sender group configuration defines how a sender's IP address is "classified" (put in a sender group). Mail flow policy configuration defines how the SMTP session from that IP address is controlled. When using this feature, an IP address may be "classified in a CIDR block" (e.g. 10.1.1.0/24) sender group while being controlled as an individual host (/32). This is done via the "signficant_bits" policy configuration setting.

Significant Bits HAT Policy Option

The HAT syntax allows for the signficant_bits configuration option. When editing the default or a specific mail flow policy in a HAT (for example, when issuing the `listenerconfig -> edit -> hostaccess -> default` command) the following questions appear if:

- rate limiting is enabled, and
  - using SenderBase for flow control is disabled, or
  - Directory Harvest Attack Prevention (DHAP) is enabled for a mail flow policy (default or specific mail flow policy)

For example:

```
Do you want to enable rate limiting per host?  [N]> y


Enter the maximum number of recipients per hour from a remote host.

[]> 2345
```

```
Would you like to specify a custom SMTP limit exceeded response? [Y]> n

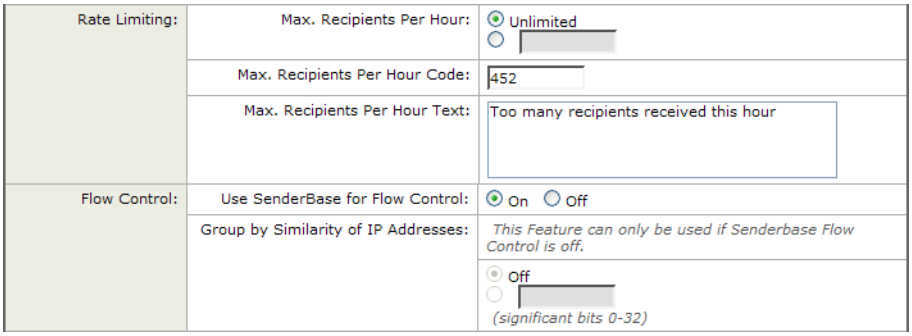
Would you like to use SenderBase for flow control by default?  [N]> n

Would you like to group hosts by the similarity of their IP addresses?  [N]> y

Enter the number of bits of IP address to treat as significant, from 0 to 32.
[24]>
```

This feature also appears in the GUI in the Mail Policies > Mail Flow Policies page.

Figure 1-11 Enable the HAT Significant Bits Feature



When the option to use SenderBase for flow control is set to "OFF" or Directory Harvest Attack Prevention is enabled, the "significant bits" value is applied to the connecting sender's IP address, and the resulting CIDR notation is used as the token for matching defined sender groups within the HAT. Any rightmost bits that are covered by the CIDR block are "zeroed out" when constructing the string. Thus, if a connection from the IP address 1.2.3.4 is made and matches on a policy with the significant_bits option set to 24, the resultant CIDR block would be 1.2.3.0/24. So by using this feature, the HAT sender group entry (for example, 10.1.1.0/24) can have a different number of network significant bits (24) from the significant bits entry in the policy assigned to that group (32, in the example above).

Injection Control Periodicity

A global configuration option exists to allow you to adjust when the injection control counters are reset. For very busy systems maintaining counters for a very large number of different IP addresses, configuring the counters to be reset more frequently (for example, every 15 minutes instead of every 60 minutes) will ensure that the data does not grow to an unmanageable size and impact system performance.

The current default value is 3600 seconds (1 hour).You can specify periods ranging from as little as 1 minute (60 seconds) to as long as 4 hours (14,400 seconds).

Adjust this period via the GUI, using the global settings (for more information, see Global Settings for Listeners, page 1-4).

You can also adjust this period using the `listenerconfig -> setup` command in the CLI.

```
mail3.example.com> listenerconfig


Currently configured listeners:

1. InboundMail (on PublicNet, 192.168.2.1) SMTP TCP Port 25 Public

2. OutboundMail (on PrivateNet, 192.168.1.1) SMTP TCP Port 25 Private


Choose the operation you want to perform:

- NEW - Create a new listener.

- EDIT - Modify a listener.

- DELETE - Remove a listener.

- SETUP - Change global settings.

[]> setup


Enter the global limit for concurrent connections to be allowed across all listeners.

[300]>


Enter the global limit for concurrent TLS connections to be allowed across all listeners.

[100]>


Enter the maximum number of message header lines.  0 indicates no limit.

[1000]>


1. Allow SenderBase to determine cache time (Recommended)

2. Don't cache SenderBase data.

3. Specify your own cache time.

[1]> 3
```

```
Enter the time, in seconds, to cache SenderBase data:

[300]>


Enter the rate at which injection control counters are reset.

[1h]> 15m


Enter the timeout for unsuccessful inbound connections.

[5m]>


Enter the maximum connection time for inbound connections.

[15m]>


What hostname should Received: headers be stamped with?
1. The hostname of the Virtual Gateway(tm) used for delivering the message
2. The hostname of the interface the message is received on

[2]>


The system will always add a Message-ID header to outgoing messages that don't already
have one.  Would you like to do the same for incoming messages? (Not recommended.) [N]>


By default connections with a HAT REJECT policy will be closed with a banner message at
the start of the SMTP conversation.  Would you like to do the rejection at the message
recipient level instead for more detailed logging of rejected mail? [N]>


Currently configured listeners:

1. InboundMail (on PublicNet, 192.168.2.1) SMTP TCP Port 25 Public

2. OutboundMail (on PrivateNet, 192.168.1.1) SMTP TCP Port 25 Private


[]>
```

Encrypting SMTP Conversations Using TLS

Enterprise Gateways (or Message Transfer Agents, i.e. MTAs) normally communicate "in the clear" over the Internet. That is, the communications are not encrypted. In several scenarios, malicious agents can intercept this communication without the knowledge of the sender or the receiver. Communications can be monitored and even altered by a third party.

Transport Layer Security (TLS) is an improved version of the Secure Socket Layer (SSL) technology. It is a widely used mechanism for encrypting SMTP conversations over the Internet. AsyncOS supports the STARTTLS extension to SMTP (Secure SMTP over TLS), described in RFC 3207 (which obsoletes RFC 2487).

The TLS implementation in AsyncOS provides privacy through encryption. It allows you to import an X.509 certificate and private key from a certificate authority service or create a self-signed certificate to use on the appliance. AsyncOS supports separate TLS certificates for public and private listeners, secure HTTP (HTTPS) management access on an interface, the LDAP interface, and all outgoing TLS connections.

To successfully configure TLS on the Cisco IronPort appliance, follow these steps:

Step 1 Obtain certificates.

Step 2 Install certificates on the Cisco IronPort appliance.

Step 3 Enable TLS on the system for receiving, delivery, or both.

Obtaining Certificates

To use TLS, the Cisco IronPort appliance must have an X.509 certificate and matching private key for receiving and delivery. You may use the same certificate for both SMTP receiving and delivery and different certificates for HTTPS services on an interface, the LDAP interface, and all outgoing TLS connections to destination domains, or use one certificate for all of them.

You may purchase certificates and private keys from a recognized certificate authority service. A certificate authority is a third-party organization or company that issues digital certificates used to verify identity and distributes public keys. This provides an additional level of assurance that the certificate is issued by a valid and trusted identity. Cisco IronPort does not recommend one service over another.

The Cisco IronPort appliance can create a self-signed certificate for your own use and generate a Certificate Signing Request (CSR) to submit to a certificate authority to obtain the public certificate. The certificate authority will return a trusted public certificate signed by a private key. Use the Network > Certificates page in the GUI or the `certconfig` command in the CLI to create the self-signed certificate, generate the CSR, and install the trusted public certificate.

If you are acquiring or creating a certificate for the first time, search the Internet for "certificate authority services SSL Server Certificates," and choose the service that best meets the needs of your organization. Follow the service's instructions for obtaining a certificate.

You can view the entire list of certificates on the Network > Certificates page in the GUI and in the CLI by using the `print` command after you configure the certificates using `certconfig`. Note that the `print` command does not display intermediate certificates.

Warning **Your Cisco IronPort appliance ships with a demonstration certificate to test the TLS and HTTPS functionality, but enabling either service with the demonstration certificate is not secure and is not recommended for general use. When you enable either service with the default demonstration certificate, a warning message is printed in the CLI.**

Intermediate Certificates

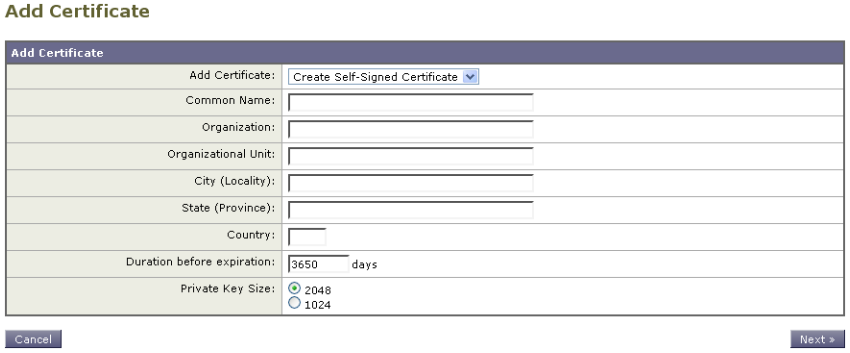
In addition to root certificate verification, AsyncOS supports the use of intermediate certificate verification. Intermediate certificates are certificates issued by a trusted root certificate authority which are then used to create additional certificates - effectively creating a chained line of trust. For example, a certificate may be issued by godaddy.com who, in turn, is granted the rights to issue certificates by a trusted root certificate authority. The certificate issued by godaddy.com must be validated against godaddy.com's private key as well as the trusted root certificate authority's private key.

Creating a Self-Signed Certificate

To create a self-signed certificate, begin by clicking Add Certificate on the Network > Certificates page in the GUI (or the `certconfig` command in the CLI). On the Add Certificate page, select Create Self-Signed Certificate.

Figure 1-12 shows the Add Certificate page with the Create Self-Signed Certificate option selected.

Figure 1-12 Add Certificate Page

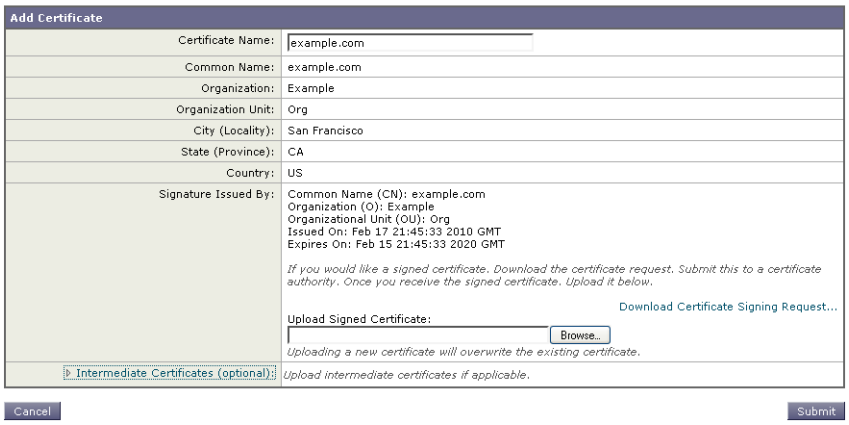


Enter the following information for the self-signed certificate:

Common Name	The fully qualified domain name.
Organization	The exact legal name of the organization.
Organizational Unit	Section of the organization.
City (Locality)	The city where the organization is legally located.
State (Province)	The state, county, or region where the organization is legally located.
Country	The two letter ISO abbreviation of the country where the organization is legally located.
Duration before expiration	The number of days before the certificate expires.
Private Key Size	Size of the private key to generate for the CSR. Only 2048-bit and 1024-bit are supported.

Click **Next** to view the certificate and signature information. Figure 1-13 shows an example of a self-signed certificate.

Figure 1-13 View Certificate Page



Enter a name for the certificate. AsyncOS assigns the common name by default.

If you want to submit a CSR for the self-signed certificate to a certificate authority, click **Download Certificate Signing Request** to save the CSR in PEM format to a local or network machine. Click **Submit** to save the certificate and commit your changes.

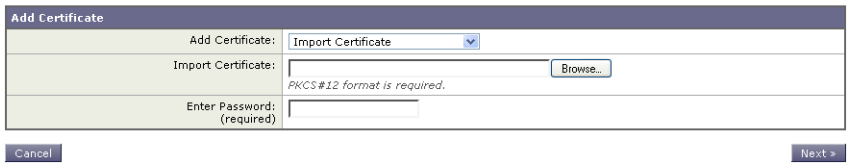
When the certificate authority returns the trusted public certificate signed by a private key, upload it by clicking on the certificate's name on the Certificates page and entering the path to the file on your local machine or network. Make sure that the trusted public certificate that you receive is in PEM format or a format that you can convert to PEM using before uploading to the appliance. (Tools for doing this are included with OpenSSL, free software from http://www.openssl.org.)

Uploading the certificate from the certificate authority overwrites the existing certificate. You can also upload an intermediate certificate related to the self-signed certificate. You can use the certificate with a public or private listener, an IP interface's HTTPS services, the LDAP interface, or all outgoing TLS connections to destination domains.

Importing a Certificate

AsyncOS also allows you to import certificates saved in the PKCS #12 format to use on your appliance. You can import the certificate either via the Network > Certificates page in the GUI or the `certconfig` command in the CLI.

Figure 1-14 The Add Certificate Page



To import a certificate via the GUI:

Step 1 Click **Add Certificate** the Network > Certificates page.

Step 2 Select the Import Certificate option.

Step 3 Enter the path to the certificate file on your network or local machine.

Step 4 Enter the password for the file.

Step 5 Click **Next** to view the certificate's information.

Step 6 Enter a name for the certificate. AsyncOS assigns the common name by default.

Step 7 Click **Submit** to save the certificate and commit your changes.

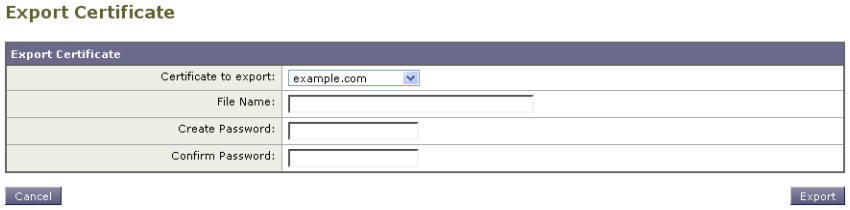
Exporting a Certificate

To export a certificate and save it in the PKCS #12 format via the GUI:

Step 1 Click **Export Certificate** on the Network > Certificates page.

The Export Certificate page is displayed.

Figure 1-15 The Export Certificate Page



Step 2 Select the certificate you want to export.

Step 3 Enter the file name for the certificate.

Step 4 Enter a password for the certificate file.

Step 5 Click **Export**.

Your web browser displays a dialog box asking you to save the file.

Step 6 Save the file to a local or network machine.

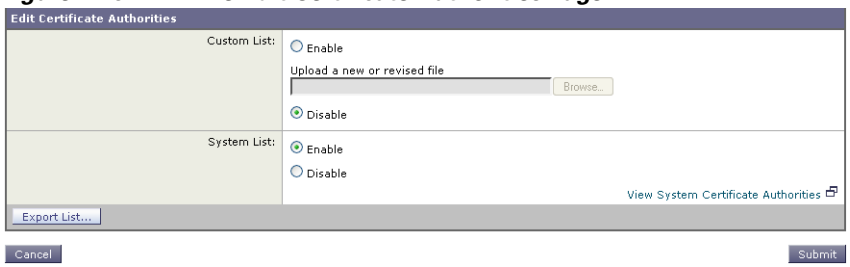
Step 7 You can export additional certificates or click **Cancel** to return to the Network > Certificates page.

Managing Lists of Certificate Authorities

The appliance has a pre-installed list of trusted certificate authorities that it uses to verify a certificate from a remote domain to establish the domain's credentials. You can import a custom list of trusted CAs for the appliance to use alongside the pre-installed system list or in place of the system list. You can manage the lists using the Network > Certificates > Edit Certificate Authorities page in the GUI or the `certconfig > certauthority` command in the CLI.

Figure 1-16 shows the Edit Certificate Authorities page in the GUI where you can manage the custom and system certificate authority lists.

Figure 1-16 The Edit Certificate Authorities Page



To see the trusted certificate authorities included in the system list, click **View System Certificate Authorities** on the Edit Certificate Authorities page.

Importing a Custom Certificate Authority List

You can create a custom of list trusted certificate authorities and import it onto the appliance. The file must be in the PEM format and include certificates for the certificate authorities that you want the appliance to trust. To import a list via the GUI, click **Enable** for the Custom List and enter the full path to the custom list on a local or network machine. Submit and commit your changes.

Disabling the System Certificate Authority List

The pre-installed system certificate authorities list cannot be removed from the appliance, but it can be disabled to allow the appliance to only use your custom list to verify certificates from remote hosts. To disable this list via the GUI, click Disable for the System List on the Edit Certificate Authorities page. Submit and commit your changes.

Exporting a Certificate Authorities List

If you want to use only a subset of the trusted certificate authorities in the system or edit an existing custom list, you can export the list to a .txt file and edit it to add or remove certificate authorities. After you have finished editing the list, import the file back onto the appliance as a custom list.

Figure 1-17 shows the Export Certificate Authority List page in the GUI where you can export the system and custom lists.

Figure 1-17 The Export Certificate Authority List Page



To export a list via the GUI, click **Export List** on the Edit Certificate Authorities page. AsyncOS displays the Export Certificate Authority List page. Select the list you want to export and enter a filename for the list. Click **Export**. AsyncOS displays a dialog box asking if want to open or save the list as a .txt file.

Enabling TLS on a Listener's HAT

You must enable TLS for any listeners where you require encryption. You may want to enable TLS on listeners facing the Internet (that is, public listeners), but not for listeners for internal systems (that is, private listeners). Or, you may want to enable encryption for all listeners.

You can specify 3 different settings for TLS on a listener. See Table 3-19.

Table 1-6 TLS Settings for a Listener

TLS Setting	Meaning
`1. No`	TLS is not allowed for incoming connections. No connections to the listener will require encrypted SMTP conversations. This is the default setting for all listeners you configure on the appliance.
`2. Preferred`	TLS is allowed for incoming connections to the listener from MTAs.
`3. Required`	TLS is allowed for incoming connections to the listener from MTAs, and until a `STARTTLS` command is received, the Cisco IronPort appliance responds with an error message to every command other than `NOOP`, `EHLO`, or `QUIT`. This behavior is specified by RFC 3207, which defines the SMTP Service Extension for Secure SMTP over Transport Layer Security. "Requiring" TLS means that email which the sender is not willing to encrypt with TLS will be refused by the Cisco IronPort appliance before it is sent, thereby preventing it from be transmitted in the clear.

By default, neither private nor public listeners allow TLS connections. You must enable TLS in a listener's HAT to enable TLS for either inbound (receiving) or outbound (sending) email. In addition, all default mail flow policy settings for private and public listeners have the `tls` setting set to "off."

You can assign a specific certificate for TLS connections to individual public listeners when creating a listener. For more information, see Creating Listeners, page 1-7.

Assigning a Certificate

You can assign a certificate to an individual public or private listener for TLS connections using either the Network > Listeners page or the `listenerconfig -> edit -> certificate` command in the CLI.

To assign a TLS certificate via the GUI, select the certificate you want in the Certificate section when creating or editing a listener and then submit and commit your changes.

Figure 1-18 Selecting a Certificate for a Listener



To assign a certificate to a listener via the CLI, follow these steps:

Step 1 Use the `listenerconfig -> edit` command to choose a listener you want to configure.

Step 2 Use the `certificate` command to see the available certificates.

Step 3 Choose the certificate you want to assign to the listener when prompted.

Step 4 When you are finished configuring the listener, issue the `commit` command to enable the change.

Logging

The Cisco IronPort appliance will note in the mail logs instances when TLS is required but could not be used by the listener. The mail logs will be updated when the following condition is met:

- TLS is set to “required” for a listener,
- the Cisco IronPort appliance has sent a "Must issue a STARTTLS command first" command, and
- the connection is closed without having received any successful recipients.

Information on why the TLS connection failed will be included in the mail logs.

GUI Example

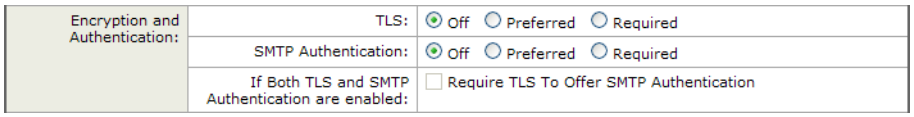
To change the TLS setting for a HAT mail flow policy for a listener via the GUI, follow these steps:

Step 1 From the Mail Flow Policies page, choose a listener whose policies you want to modify, and then click the link for the name of policy to edit. (You can also edit the Default Policy Parameters.)

The Edit Mail Flow Policies page is displayed.

Step 2 In the “Encryption and Authentication” section, for the “TLS:” field, choose the level of TLS you want for the listener.

Figure 1-19 Requiring TLS in a Listener's Mail Flow Policy Parameters



Step 3 Submit and commit your changes.

The mail flow policy for the listener is updated with the TLS setting you chose.

CLI Example

To change the default TLS setting for a listener via the CLI, follow these steps:

Step 1 Use the `listenerconfig -> edit` command to choose a listener you want to configure.

Step 2 Use the `hostaccess -> default` command to edit the listener’s default HAT settings.

Step 3 Change the TLS setting by entering one of the following choices when you are prompted with the following questions:

```
Do you want to allow encrypted TLS connections?

1. No

2. Preferred

3. Required

[1]> 3
```

```
You have chosen to enable TLS. Please use the 'certconfig' command to
ensure that there is a valid certificate configured.
```

Note that this example asks you to use the `certconfig` command to ensure that there is a valid certificate that can be used with the listener. If you have not created any certificates, the listener uses the demonstration certificate that is pre-installed on the appliance. You may enable TLS with the demonstration certificate for testing purposes, but it is not secure and is not recommended for general use. Use the `listenerconfig -> edit -> certificate` command to assign a certificate to the listener.

Once you have configured TLS, the setting will be reflected in the summary of the listener in the CLI:

```
Name: Inboundmail

Type: Public

Interface: PublicNet (192.168.2.1/24) TCP Port 25

Protocol: SMTP

Default Domain:

Max Concurrency: 1000 (TCP Queue: 50)

Domain map: disabled

TLS: Required
```

Step 4 Issue the `commit` command to enable the change.

Enabling TLS and Certificate Verification on Delivery

You can require that TLS is enabled for email delivery to specific domains using the Destination Controls page or the `destconfig` command.

In addition to TLS, you can require that the domain's server certificate is verified. This domain verification is based on a digital certificate used to establish the domain's credentials. The validation process involves two validation requirements:

- The chain of issuer certificates for the SMTP session ends in a certificate issued by a trusted certificate authority (CA).
- The Common Name (CN) listed on the certificate matches either the receiving machine's DNS name or the message's destination domain.

  - or -

  The message's destination domain matches one of the DNS names in the certificate's Subject Alternative Name (subjectAltName) extension, as described in RFC 2459. The matching supports wildcards as described in section 3.1 of RFC 2818.

A trusted CA is a third-party organization or company that issues digital certificates used to verify identity and distributes public keys. This provides an additional level of assurance that the certificate is issued by a valid and trusted identity.

You can configure your Cisco IronPort appliance to send messages to a domain over a TLS connection as an alternative to envelope encryption. See the "Cisco IronPort Email Encryption" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide* for more information.

You can specify a certificate for the appliance to use for all outgoing TLS connections. To specify the certificate, click **Edit Global Settings** on the Destination Controls page or use `destconfig -> setup` in the CLI. The certificate is a global setting, not a per-domain setting.

You can specify 5 different settings for TLS for a given domain when you include a domain using the Destination Controls page or the `destconfig` command. In addition to specifying whether exchanges with a domain are required or preferred to be TLS encoded, you can dictate whether validation of the domain is necessary. See Table 1-7 for an explanation of the settings.

Table 1-7 TLS Settings for Delivery

TLS Setting	Meaning
Default	The default TLS setting set using the Destination Controls page or the `destconfig -> default` subcommand used for outgoing connections from the listener to the MTA for the domain. The value "Default" is set if you answer "no" to the question: "Do you wish to apply a specific TLS setting for this domain?"
1. No	TLS is not negotiated for outgoing connections from the interface to the MTA for the domain.
2. Preferred	TLS is negotiated from the Cisco IronPort appliance interface to the MTA(s) for the domain. However, if the TLS negotiation fails (prior to receiving a 220 response), the SMTP transaction will continue "in the clear" (not encrypted). No attempt is made to verify if the certificate originates from a trusted certificate authority. If an error occurs after the 220 response is received the SMTP transaction does not fall back to clear text.
3. Required	TLS is negotiated from the Cisco IronPort appliance interface to MTA(s) for the domain. No attempt is made to verify the domain's certificate. If the negotiation fails, no email is sent through the connection. If the negotiation succeeds, the mail is delivered via an encrypted session.

Table 1-7 TLS Settings for Delivery

TLS Setting	Meaning
4. Preferred (Verify)	TLS is negotiated from the Cisco IronPort appliance to the MTA(s) for the domain. The appliance attempts to verify the domain's certificate. Three outcomes are possible: • TLS is negotiated and the certificate is verified. The mail is delivered via an encrypted session. • TLS is negotiated, but the certificate is not verified. The mail is delivered via an encrypted session. • No TLS connection is made and, subsequently the certificate is not verified. The email message is delivered in plain text.
5. Required (Verify)	TLS is negotiated from the Cisco IronPort appliance to the MTA(s) for the domain. Verification of the domain's certificate is required. Three outcomes are possible: • A TLS connection is negotiated and the certificate is verified. The email message is delivered via an encrypted session. • A TLS connection is negotiated but the certificate is not verified by a trusted CA. The mail is not delivered. • A TLS connection is not negotiated. The mail is not delivered.

If there is no specific entry for a given recipient domain in the good neighbor table, or if there is a specific entry but there is no specific TLS setting for the entry, then the behavior is whatever is set using the Destination Controls page or the `destconfig -> default` subcommand ("No," "Preferred," "Required," "Preferred (Verify)," or "Required (Verify)").

Sending Alerts When a Required TLS Connection Fails

You can specify whether the Cisco IronPort appliance sends an alert if the TLS negotiation fails when delivering messages to a domain that requires a TLS connection. The alert message contains name of the destination domain for the failed TLS negotiation. The Cisco IronPort appliance sends the alert message to all recipients set to receive Warning severity level alerts for System alert types. You can manage alert recipients via the System Administration > Alerts page in the GUI (or via the `alertconfig` command in the CLI).

To enable TLS connection alerts, click **Edit Global Settings** on the Destination Controls page or `destconfig -> setup` subcommand. This is a global setting, not a per-domain setting. For information on the messages that the appliance attempted to deliver, use the Monitor > Message Tracking page or the mail logs.

Logging

The Cisco IronPort appliance will note in the mail logs instances when TLS is required for a domain but could not be used. Information on why the TLS connection could not be used will be included. The mail logs will be updated when any of the following conditions are met:

- The remote MTA does not support ESMTP (for example, it did not understand the EHLO command from the Cisco IronPort appliance).
- The remote MTA supports ESMTP but "STARTTLS" was not in the list of extensions it advertised in its EHLO response.
- The remote MTA advertised the "STARTTLS" extension but responded with an error when the Cisco IronPort appliance sent the STARTTLS command.

CLI Example

In this example, the `destconfig` command is used to require TLS connections and encrypted conversations for the domain "partner.com." The list is then printed.

A certificate for example.com is used for outgoing TLS connections instead of the demonstration certificate that is pre-installed. You may enable TLS with the demonstration certificate for testing purposes, but it is not secure and is not recommended for general use.

```
mail3.example.com> destconfig


There is currently 1 entry configured.


Choose the operation you want to perform:

- SETUP - Change global settings.

- NEW - Create a new entry.

- DELETE - Remove an entry.

- DEFAULT - Change the default.

- LIST - Display a summary list of all entries.

- DETAIL - Display details for one destination or all entries.

- IMPORT - Import tables from a file.

- EXPORT - Export tables to a file.

[]> setup


The "Demo" certificate is currently configured. You may use "Demo", but this will not be
secure.


1. example.com

2. Demo
```

```
Please choose the certificate to apply:

[1]> 1


Do you want to send an alert when a required TLS connection fails? [N]>


There is currently 1 entry configured.


Choose the operation you want to perform:

- SETUP - Change global settings.

- NEW - Create a new entry.

- DELETE - Remove an entry.

- DEFAULT - Change the default.

- LIST - Display a summary list of all entries.

- DETAIL - Display details for one destination or all entries.

- IMPORT - Import tables from a file.

- EXPORT - Export tables to a file.

[]> new


Enter the domain you wish to limit.

[]> partner.com


Do you wish to configure a concurrency limit for partner.com?  [Y]> n


Do you wish to apply a messages-per-connection limit to this domain? [N]> n


Do you wish to apply a recipient limit to this domain?  [N]> n


Do you wish to apply a specific bounce profile to this domain?  [N]> n
```

```
Do you wish to apply a specific TLS setting for this domain?  [N]> y


Do you want to use TLS support?

1. No

2. Preferred

3. Required

4. Preferred (Verify)

5. Required (Verify)

[1]> 3


You have chosen to enable TLS. Please use the 'certconfig' command to ensure that there
is a valid certificate configured.

Do you wish to apply a specific bounce verification address tagging setting for this
domain? [N]> n


Do you wish to apply a specific bounce profile to this domain? [N]> n


There are currently 2 entries configured.


Choose the operation you want to perform:

- SETUP - Change global settings.

- NEW - Create a new entry.

- EDIT - Modify an entry.

- DELETE - Remove an entry.

- DEFAULT - Change the default.

- LIST - Display a summary list of all entries.

- DETAIL - Display details for one destination or all entries.

- CLEAR - Remove all entries.

- IMPORT - Import tables from a file.

- EXPORT - Export tables to a file.
```

```
[]> list


                Rate                Bounce        Bounce
Domain          Limiting  TLS       Verification  Profile
==========      ========  =======   ============  =========
partner.com     Default   Req       Default       Default
(Default)       On        Off       Off           (Default)


There are currently 2 entries configured.


Choose the operation you want to perform:
- SETUP - Change global settings.
- NEW - Create a new entry.
- EDIT - Modify an entry.
- DELETE - Remove an entry.
- DEFAULT - Change the default.
- LIST - Display a summary list of all entries.
- DETAIL - Display details for one destination or all entries.
- CLEAR - Remove all entries.
- IMPORT - Import tables from a file.
- EXPORT - Export tables to a file.
[]>
```

Enabling a Certificate for HTTPS

You can enable a certificate for HTTPS services on an IP interface using either the Network > IP Interfaces page in the GUI or the `interfaceconfig` command in the CLI. When adding an IP interface via the GUI, select a certificate that you want to use for the HTTPS service, check the **HTTPS** check box, and enter the port number.

In following example, the `interfaceconfig` command is used to edit the IP interface PublicNet to enable HTTPS services on port 443 (the default port). All other defaults for the interface are accepted. (Typing Enter at the prompt accepts the default value shown in brackets.)

Note that this example shows using the demonstration certificate that is pre-installed on the appliance. You may enable HTTPS services with the demonstration certificate for testing purposes, but it is not secure and is not recommended for general use.

Note You can enable HTTPS services using the System Setup Wizard in the GUI. Refer to "Define the Default Router (Gateway), Configure the DNS Settings, and Enabling Secure Web Access" in the "Setup and Installation" chapter of the *Cisco IronPort AsyncOS for Email Configuration Guide*.

After the changes from this command are committed, users can access the Graphical User Interface (GUI) using the URL for secure HTTPS: `https://192.168.2.1`

```
mail3.example.com> interfaceconfig


Currently configured interfaces:

1. Management (192.168.42.42/24: mail3.example.com)

2. PrivateNet (192.168.1.1/24: mail3.example.com)

3. PublicNet (192.168.2.1/24: mail3.example.com)


Choose the operation you want to perform:

- NEW - Create a new interface.

- EDIT - Modify an interface.

- GROUPS - Define interface groups.

- DELETE - Remove an interface.

[]> edit


Enter the number of the interface you wish to edit.

[]> 3


IP interface name (Ex: "InternalNet"):

[PublicNet]>


Would you like to configure an IPv4 address for this interface (y/n)? [Y]> y
```

```
IPv4 Address (Ex: 192.168.1.2):

[192.168.2.1]>


Netmask (Ex: "255.255.255.0" or "0xffffff00"):
[24]>


Would you like to configure an IPv6 address for this interface (y/n)? [N]>


Ethernet interface:

1. Data 1

2. Data 2

3. Management

[2]>


Hostname:

[mail3.example.com]>


Do you want to enable Telnet on this interface?  [N]>


Do you want to enable SSH on this interface?  [N]>


Do you want to enable FTP on this interface?  [N]>


Do you want to enable HTTP on this interface?  [Y]>


Which port do you want to use for HTTP?

[80]>


Do you want to enable HTTPS on this interface?  [N]> y
```

```
Which port do you want to use for HTTPS?

[443]> 443


Do you want to enable Spam Quarantine HTTP on this interface? [N]>


Do you want to enable Spam Quarantine HTTPS on this interface? [N]>


The "Demo" certificate is currently configured. You may use "Demo", but this will not be
secure. To assure privacy, run "certconfig" first.


Both HTTP and HTTPS are enabled for this interface, should HTTP requests redirect to the
secure service? [Y]>


Currently configured interfaces:

1. Management (192.168.42.42/24: mail3.example.com)

2. PrivateNet (192.168.1.1/24: mail3.example.com)

3. PublicNet (192.168.2.1/24: mail3.example.com)


Choose the operation you want to perform:

- NEW - Create a new interface.

- EDIT - Modify an interface.

- GROUPS - Define interface groups.

- DELETE - Remove an interface.

[]>
```



CHAPTER 2

Configuring Routing and Delivery Features

This chapter explains the features that affect routing and delivery of email traveling through the Cisco IronPort appliance. After you have configured the gateway to receive email with listeners, you can further tailor the configuration of the routing and delivery that the appliance performs, for both inbound (receiving email from the Internet) and outbound (sending email from your internal systems) processing.

This chapter contains the following sections:

- Routing Email for Local Domains, page 2-1 (SMTP Routes page and `smtproutes` command)
- Rewriting Addresses, page 2-7
- Creating Alias Tables, page 2-8 (`aliasconfig` command)
- Configuring Masquerading, page 2-16 (`masquerade` subcommand)
- The Domain Map Feature, page 2-28 (`domainmap` subcommand)
- Directing Bounced Email, page 2-35 (Bounce Profiles and the `bounceconfig` command)
- Controlling Email Delivery, page 2-44 (Destination Controls and the `destconfig` and `deliveryconfig` commands)
- Cisco IronPort Bounce Verification, page 2-53
- Set Email Delivery Parameters, page 2-57
- Using Virtual Gateway™ Technology, page 2-59 (`altsrchost` command)
- Using Global Unsubscribe, page 2-69 (`unsubscribe` command)

Routing Email for Local Domains

In Chapter 1, "Customizing Listeners" you customized private and public listeners to service SMTP connections for an Enterprise Gateway configuration. Those listeners were customized to handle specific connections (via HAT modification) and receive mail for specific domains (via RAT modification of public listeners).

The Cisco IronPort appliance routes mail to local domains to hosts specified via the Network > SMTP Routes page (or the `smtproutes` command). This feature is similar to the sendmail `mailertable` feature.

Note If you have completed the GUI's System Setup Wizard (or the Command Line Interface `systemsetup` command) as described in the "Setup and Installation" chapter of the *Cisco IronPort AsyncOS for Email Configuration Guide* and committed the changes, you defined the first SMTP route entries on the appliance for each RAT entry you entered at that time.

SMTP Routes Overview

SMTP Routes allow you to redirect all email for a particular domain to a different mail exchange (MX) host. For example, you could make a mapping from `example.com` to `groupware.example.com`. This mapping causes any email with `@example.com` in the Envelope Recipient address to go instead to `groupware.example.com`. The system performs an "MX" lookup on `groupware.example.com`, and then performs an "A" lookup on the host, just like a normal email delivery. This alternate MX host does not need to be listed in DNS MX records and it does not even need to be a member of the domain whose email is being redirected. The Cisco IronPort AsyncOS operating system allows up to forty thousand (40,000) SMTP Route mappings to be configured for your Cisco IronPort appliance. (See SMTP Routes Limits, page 2-3.)

This feature also allows host "globbing." If you specify a partial domain, such as `.example.com`, then any domain ending in `example.com` matches the entry. For instance, `fred@foo.example.com` and `wilma@bar.example.com` both match the mapping.

If a host is not found in the SMTP Routes table, an MX lookup is performed using DNS. The result is not re-checked against the SMTP Routes table. If the DNS MX entry for `foo.domain` is `bar.domain`, any email sent to `foo.domain` is delivered to the host `bar.domain`. If you create a mapping for `bar.domain` to some other host, email addressed to `foo.domain` is not affected.

In other words, recursive entries are not followed. If there is an entry for `a.domain` to redirect to `b.domain`, and a subsequent entry to redirect email for `b.domain` to `a.domain`, a mail loop will *not* be created. In this case, email addressed to `a.domain` will be delivered to the MX host specified by `b.domain`, and conversely email addressed to `b.domain` will be delivered to the MX host specified by `a.domain`.

The SMTP Routes table is read from the top down for every email delivery. The most specific entry that matches a mapping wins. For example, if there are mappings for both `host1.example.com` and `.example.com` in the SMTP Routes table, the entry for `host1.example.com` will be used because it is the more specific entry — even if it appears after the less specific `.example.com` entry. Otherwise, the system performs a regular MX lookup on the domain of the Envelope Recipient.

Default SMTP Route

You can also define a default SMTP route with the special keyword `ALL`. If a domain does not match a previous mapping in the SMTP Routes list, it defaults to being redirected to the MX host specified by the `ALL` entry.

When you print the SMTP Routes entries, the default SMTP route is listed as `ALL:`. You cannot delete the default SMTP route; you may only clear any values entered for it.

Configure the default SMTP route via the Network > SMTP Routes page or the `smtproutes` command.

Defining an SMTP Route

Use the Network > SMTP Routes page (or the `smtproutes` command) to construct routes. When you create a new route, you first specify the domain or partial domain for which you want to create a permanent route. You then specify destination hosts. Destination hosts can be entered as fully-qualified hostnames or as IP addresses. IP addresses can be either Internet Protocol version 4 (IPv4) or version 6 (IPv6).

For IPv6 addresses, AsyncOS supports the following formats:

- `2620:101:2004:4202::0-2620:101:2004:4202::ff`

- `2620:101:2004:4202::`
- `2620:101:2004:4202::23`
- `2620:101:2004:4202::/64`

You can also specify a a special destination host of `/dev/null` to drop the messages that match the entry. (So, in effect, specifying `/dev/null` for the default route is will ensure that no mail received by the appliance is ever delivered.)

A receiving domain can have multiple destination hosts, each assigned a priority number, much like an MX record. The destination host with the lowest number identifies as the primary destination host for the receiving domain. Other destination hosts listed will be used as backup.

Destinations with identical priority will be used in a "round-robin" fashion. The round-robin process is based on SMTP connections, and is not necessarily message-based. Also, if one or more of the destination hosts are not responding, messages will be delivered to one of the reachable hosts. If all the configured destination hosts are not responding, mail is queued for the receiving domain and delivery to the destination hosts is attempted later. (It does not fail over to using MX records).

When constructing routes using the `smtproutes` command in the CLI, you can prioritize each destination host by using `/pri=`, followed by an integer between `0` and `65535` to assign priority (`0` is the highest priority) after the hostname or IP address. For example, `host1.example.com/pri=0` has a higher priority than `host2.example.com/pri=10`. Separate multiple entries with commas.

SMTP Routes Limits

You can define up to 40,000 routes. The final default route of `ALL` is counted as a route against this limit. Therefore, you can define up to 39,999 custom routes and one route that uses the special keyword `ALL`.

SMTP Routes and DNS

Use the special keyword `USEDNS` to tell the appliance to do MX lookups to determine next hops for specific domains. This is useful when you need to route mail for subdomains to a specific host. For example, if mail to example.com is to be sent to the company's Exchange server, you might have something similar to the following SMTP route:

```
example.com exchange.example.com
```

However, for mail to various subdomains (foo.example.com), add an SMTP route that looks like this:

```
.example.com USEDNS
```

SMTP Routes and Alerts

Alerts sent from the appliance to addresses specified in the System Administration > Alerts page (or the `alertconfig` command) follow SMTP Routes defined for those destinations.

SMTP Routes, Mail Delivery, and Message Splintering

Incoming: if one message has 10 recipients and they are all on the same Exchange server, AsyncOS will open one TCP connection and present exactly one message to the mail store, not 10 separate messages.

Outgoing: works similarly, but if one message is going to 10 recipients in 10 different domains, AsyncOS will open 10 connections to 10 MTAs and deliver them one email each.

Splintering: if one incoming message has 10 recipients and they are each in separate Incoming Policy groups (10 groups), the message will splinter even if all 10 recipients are on the same Exchange server. Thus, 10 separate emails will be delivered over a single TCP connection.

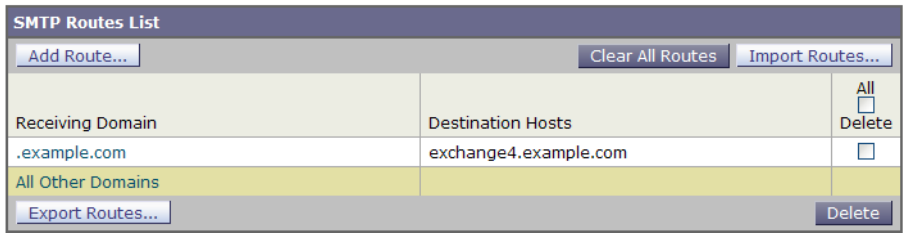
SMTP Routes and Outbound SMTP Authentication.

If an Outbound SMTP Authentication profile has been created, you can apply it to an SMTP Route. This allows authentication for outgoing mail in cases where the Cisco IronPort appliance sits behind a mail relay server that is at the edge of the network. For more information about Outbound SMTP Authentication, see Outgoing SMTP Authentication, page 3-39.

Managing SMTP Routes via the GUI

Use the Network > SMTP Routes page to manage SMTP Routes on your Cisco IronPort appliance. You can add, modify, and delete mappings in the table. You can export or import the SMTP Routes entries.

Figure 2-1 SMTP Routes Page

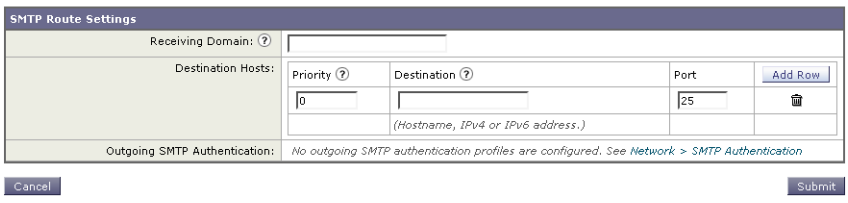


Adding SMTP Routes

To add an SMTP Route:

Step 1 Click **Add Route** on the Network > SMTP Routes page. The Add SMTP Route page is displayed:

Figure 2-2 Add SMTP Route Page



Step 2 Enter a receiving domain. This can be a hostname, domain, IPv4 address, or IPv6 address.

Step 3 Enter a destination host. This can be a hostname, IPv4 address, or IPv6 address. You can add multiple destination hosts by clicking **Add Row** and entering the next destination host in the new row.

Note You can specify a port number by adding ":<*port number*>" to the destination host: `example.com:25`.

Step 4 If you add multiple destination hosts, enter an integer between `0` and `65535` to assign priority to the hosts. `0` is the highest priority. See Defining an SMTP Route, page 2-2 for more information.

Step 5 Click **Submit**.The SMTP Routes page is displayed, reflecting your changes.

Step 6 Commit your changes.

Editing SMTP Routes

To edit an SMTP Route:

Step 1 Click the name of an existing SMTP Route in the SMTP Route listing. The Edit SMTP Route page is displayed.

Step 2 Edit the route.

Step 3 Click **Submit**.

Step 4 The SMTP Routes page is displayed, reflecting your changes.

Step 5 Commit your changes.

Deleting SMTP Routes

To delete SMTP Routes:

Step 1 Mark the checkbox(es) to the right of the SMTP Route(s) to delete.

Step 2 Click **Delete**.

To delete all of the SMTP Routes, mark the checkbox labeled "All" and click **Delete**.

Exporting SMTP Routes

Similar to the Host Access Table (HAT) and the Recipient Access Table (RAT), you can also modify SMTP routes mappings by exporting and importing a file. To export the SMTP Routes:

Step 1 Click **Export SMTP Routes** on the SMTP Routes page. The Export SMTP Routes page is displayed.

Step 2 Enter a name for the file and click **Submit**.

Importing SMTP Routes

Similar to the Host Access Table (HAT) and the Recipient Access Table (RAT), you can also modify SMTP routes mappings by exporting and importing a file. To import SMTP Routes:

Step 1 Click **Import SMTP Routes** on the SMTP Routes page. The Import SMTP Routes page is displayed.

Step 2 Select the file that contains the exported SMTP Routes.

Step 3 Click **Submit**. You are warned that importing will replace all existing SMTP Routes. All of the SMTP Routes in the text file are imported.

Step 4 Click **Import**.

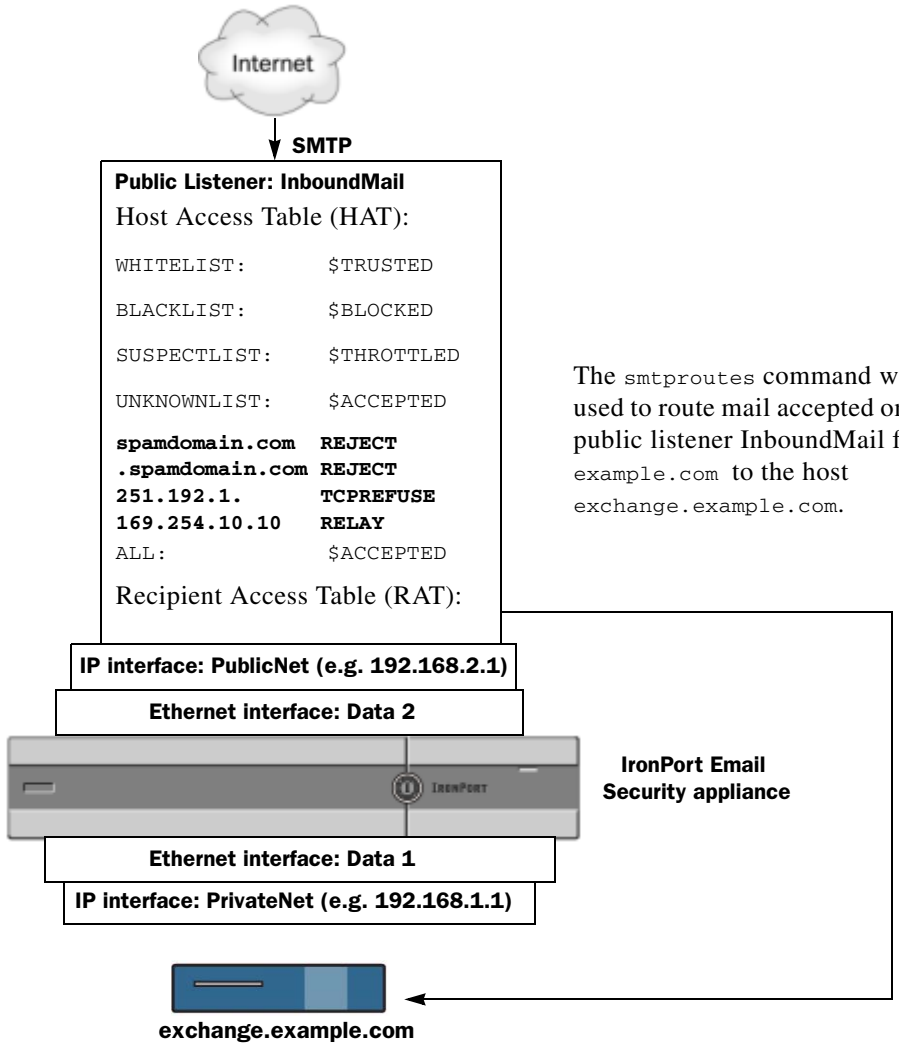
You can place "comments" in the file. Lines that begin with a '#' character are considered comments and are ignored by AsyncOS. For example:

```
# this is a comment, but the next line is not

ALL:
```

At this point, our Email Gateway configuration looks like this:

Figure 2-3 SMTP Routes Defined for a Public Listener



Rewriting Addresses

AsyncOS provides several methods for rewriting Envelope Sender and Recipient addresses in the email pipeline. Rewriting addresses can be used, for example, to redirect mail sent to a partner domain or to hide ("mask") your internal infrastructure.

Table 2-1 provides an overview of the various features used for rewriting sender and recipient email addresses.

Table 2-1 Methods for Rewriting Addresses

Original Address	Change to	Feature	Works on
`*@anydomain`	`user@domain`	Alias Tables (see Creating Alias Tables, page 2-8)	• Envelope Recipients only • Applied globally • Maps aliases to email addresses or other aliases
`*@olddomain`	`*@newdomain`	Domain Mapping (see The Domain Map Feature, page 2-28)	• Envelope Recipients only • Applied per listener
`*@olddomain`	`*@newdomain`	Masquerading (see Configuring Masquerading, page 2-16)	• Envelope Sender and the To:, From:, and/or CC: headers • Applied per listener

Creating Alias Tables

Alias tables provide a mechanism to redirect messages to one or more recipients. You can construct a mapping table of aliases to usernames and other aliases in a similar fashion to the `/etc/mail/aliases` feature of a sendmail configuration on some Unix systems.

When the Envelope Recipient (also known as the Envelope To, or `RCPT TO`) of an email accepted by a listener matches an alias as defined in an alias table, the Envelope Recipient address of the email will be rewritten.

Note A listener checks the alias table and modifies the recipients *after* checking the RAT and *before* message filters. Refer to "Understanding the Email Pipeline" in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

Note The Alias Table functionality actually rewrites the Envelope Recipient of the email. This is different than the `smtproutes` command (see Directing Bounced Email, page 2-35), which does not rewrite the Envelope Recipient of the email, but instead simply reroutes the email to specified domains.

Configuring an Alias Table from the Command Line

Alias tables are defined in sections as follows: each section is headed by a domain context, which is a list of domains that the section is relevant to, followed by a list of maps.

A domain context is a list of one or more domains or partial domains, separated by commas and enclosed in square brackets ('`[`' and '`]`'). A domain is a string containing letters, digits hyphens, and periods as defined in RFC 1035, section 2.3.1., "Preferred name syntax." A partial domain, such as `.example.com` is a domain that begins with a period. All domains that end with a substring matching the partial domain

are considered a match. For example, the domain context `.example.com` would match `mars.example.com` and `venus.example.com`. Below the domain context is a list of maps, which are aliases followed by a list of recipients. A map is constructed as follows:

Table 2-2 Alias Table Syntax

Left-hand Side (LHS)	Separator	Right-hand Side (RHS)
a list of one or more aliases to match	the colon character (":")	a list of one or more recipient addresses or aliases

An alias in the **left-hand side** can contain the following formats:

`username`	Specifies an alias to match. There must be a preceding "domains" attribute specified in the table. The lack of this parameter will produce an error.
`user@domain`	Specifies an exact email address to match on.

You can enter multiple aliases, separated by commas on a single left-hand side line.

Each recipient in the **right-hand side** can be a full `user@domain` email address, or another alias.

An alias file can contain "global" aliases (aliases that are applied globally instead of to a specific domain) with no implied domain, domain contexts within which aliases have one or more implied domains, or both.

"Chains" (or recursive entries) of aliases may be created, but they must end in a full email address.

A special destination of `/dev/null` is supported to drop the message in order to be compatible with context of a sendmail configuration. If a message is mapped to `/dev/null` via an alias table, the dropped counter is increased. (See "Managing and Monitoring via the CLI" in the *Cisco IronPort AsyncOS for Email Daily Management Guide*.) The recipient is accepted but not enqueued.

Exporting and Importing an Alias Table

To import an alias table, first see Appendix B, "Accessing the Appliance" to ensure that you can access the appliance.

Use the `export` subcommand of the `aliasconfig` command to save any existing alias table. A file (whose name you specify) will be written to the `/configuration` directory for the listener. You can modify this file outside of the CLI and then re-import it. (If you have malformed entries in the file, errors are printed when you try to import the file.)

Place the alias table file in the `/configuration` directory, and then use the `import` subcommand of the `aliasconfig` command to upload the file.

Comment out lines in the table using a number symbol (#) at the beginning of each line.

Remember to issue the `commit` command after you import an alias table file so that the configuration changes take effect.

Deleting Entries from the Alias Table

If you delete entries from the alias table from the command line interface (CLI), you are prompted to choose a domain group first. Choose the "ALL (any domain)" entry to see a numbered list of aliases that apply to all domains. Then choose the number(s) of the aliases you want to delete.

Example Alias Table



Note All entries in this example table have been commented out.

```
# sample Alias Table file
# copyright (c) 2001-2005, IronPort Systems, Inc.
#
# Incoming Envelope To addresses are evaluated against each
# entry in this file from top to bottom. The first entry that
# matches will be used, and the Envelope To will be rewritten.
#
# Separate multiple entries with commas.
#
# Global aliases should appear before the first domain
# context. For example:
#
#   admin@example.com: administrator@example.com
#   postmaster@example.net: administrator@example.net
#
# This alias has no implied domain because it appears
# before a domain context:
#
#   someaddr@somewhere.dom: specificperson@here.dom
#
# The following aliases apply to recipients @ironport.com and
# any subdomain within .example.com because the domain context
# is specified.
#
# Email to joe@ironport.com or joe@foo.example.com will
# be delivered to joseph@example.com.
```

```
#
# Similarly, email to fred@mx.example.com will be
# delivered to joseph@example.com
#
#   [ironport.com, .example.com]
#
#   joe, fred: joseph@example.com
#


# In this example, email to partygoers will be sent to
# three addresses:
#
#   partygoers: wilma@example.com, fred@example.com, barney@example.com
#
# In this example, mail to help@example.com will be delivered to
# customercare@otherhost.dom. Note that mail to help@ironport.com will
# NOT be processed by the alias table because the domain context
# overrides the previous domain context.
#
#   [example.com]
#
#   help: customercare@otherhost.dom
#
# In this example, mail to nobody@example.com is dropped.
#
#   nobody@example.com: /dev/null
#
# "Chains" may be created, but they must end in an email address.
# For example, email to "all" will be sent to 9 addresses:
```

```
#

#   [example.com]

#

#   all: sales, marketing, engineering

#   sales: joe@example.com, fred@example.com, mary@example.com

#   marketing:bob@example.com, advertising

#   engineering:betty@example.com, miles@example.com, chris@example.com

#   advertising:richard@example.com, karen@advertising.com
```

Example aliasconfig Command

In this example, the `aliasconfig` command is used to construct an alias table. First, the domain context of **`example.com`** is specified. Then, an alias of **`customercare`** is constructed so that any email sent to `customercare@example.com` is redirected to `bob@example.com`, `frank@example.com`, and `sally@example.com`. Next, a global alias of **`admin`** is constructed so that an email sent to `admin` is redirected to `administrator@example.com`. Finally, the alias table is printed to confirm.

Note that when the table is printed, the global alias for `admin` appears *before* the first domain context of `example.com`.

```
mail3.example.com> aliasconfig


No aliases in table.


Choose the operation you want to perform:

- NEW - Create a new entry.

- IMPORT - Import aliases from a file.

[]> new


How do you want your aliases to apply?


1. Globally

2. Add a new domain context

[1]> 2
```

```
Enter new domain context.

Separate multiple domains with commas.

Partial domains such as .example.com are allowed.

[]> example.com


Enter the alias(es) to match on.

Separate multiple aliases with commas.

Allowed aliases:

    - "user" - This user in this domain context.

    - "user@domain" - This email address.

[]> customercare


Enter address(es) for "customercare".

Separate multiple addresses with commas.

[]> bob@example.com, frank@example.com, sally@example.com


Adding alias customercare: bob@example.com,frank@example.com,sally@example.com

Do you want to add another alias?  [N]> n


There are currently 1 mappings defined.

Choose the operation you want to perform:

- NEW - Create a new entry.

- EDIT - Modify an entry.

- DELETE - Remove an entry.

- PRINT - Display the table.

- IMPORT - Import aliases from a file.

- EXPORT - Export table to a file.

- CLEAR - Clear the table.
```

```
[]> new

How do you want your aliases to apply?

1. Globally
2. Add a new domain context
3. example.com
[1]> 1

Enter the alias(es) to match on.
Separate multiple aliases with commas.
Allowed aliases:
    - "user@domain" - This email address.
    - "user" - This user for any domain
    - "@domain" - All users in this domain.
    - "@.partialdomain" - All users in this domain, or any of its sub domains.
[]> admin

Enter address(es) for "admin".
Separate multiple addresses with commas.
[]> administrator@example.com

Adding alias admin: administrator@example.com
Do you want to add another alias?  [N]> n

There are currently 2 mappings defined.

Choose the operation you want to perform:
- NEW - Create a new entry.
```

```
- EDIT - Modify an entry.

- DELETE - Remove an entry.

- PRINT - Display the table.

- IMPORT - Import aliases from a file.

- EXPORT - Export table to a file.

- CLEAR - Clear the table.

[]> print


admin: administrator@example.com


[ example.com ]

customercare: bob@example.com, frank@example.com, sally@example.com


There are currently 2 mappings defined.


Choose the operation you want to perform:

- NEW - Create a new entry.

- EDIT - Modify an entry.

- DELETE - Remove an entry.

- PRINT - Display the table.

- IMPORT - Import aliases from a file.

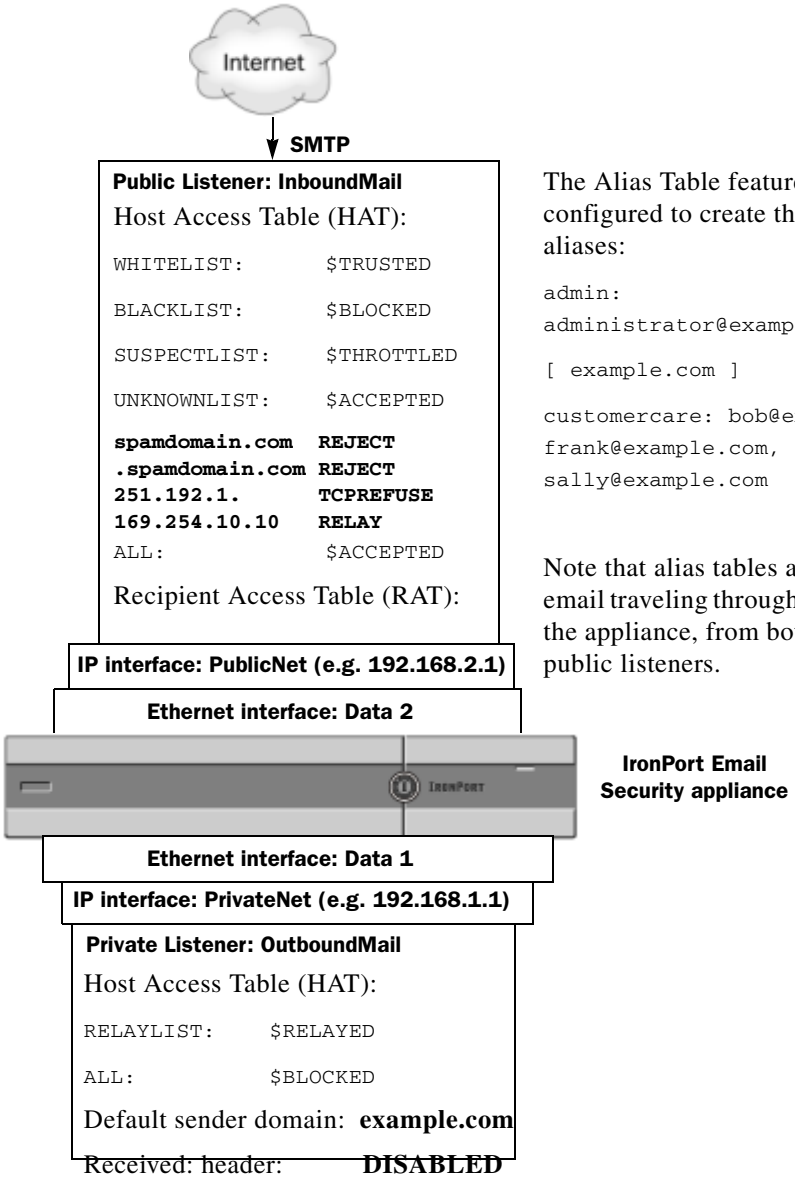
- EXPORT - Export table to a file.

- CLEAR - Clear the table.

[]>
```

At this point, our Email Gateway configuration looks like this:

Figure 2-4 Alias Tables Defined for the Appliance



The Alias Table feature was configured to create the following aliases:

```
admin:
administrator@example.com

[ example.com ]

customercare: bob@example.com,
frank@example.com,
sally@example.com
```

Note that alias tables apply to all email traveling through (received by) the appliance, from both private and public listeners.

Configuring Masquerading

Masquerading is a feature that rewrites the Envelope Sender (also known as the sender, or `MAIL FROM`) and the To:, From:, and/or CC: headers on email processed by a listener according to a table that you construct. A typical example implementation of this feature is "Virtual Domains," which allows you to host multiple domains from a single site. Another typical implementation is "hiding" your network infrastructure by "stripping" the subdomains from strings in email headers. The Masquerading feature is available for both private and public listeners.

Note The Masquerading feature is configured on a per-listener basis, as opposed to the Alias Tables functionality, which is configured for the entire system.

Note A listener checks the masquerading table for matches and modifies the recipients while the message is in the work queue, immediately after LDAP recipient acceptance queries and before LDAP routing queries. Refer to "Understanding the Email Pipeline" in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

The Masquerading feature actually rewrites addresses for the Envelope Sender and the To:, From:, and CC: fields of the email that has been received. You can specify different masquerading parameters for each listener you create in one of two ways:

Step 1 via a static table of mappings you create, or

Step 2 via an LDAP query.

This section discusses the static table method. The table format is forward-compatible with the `/etc/mail/genericstable` feature of a sendmail configuration on some Unix systems. See Chapter 3, "LDAP Queries" for more information on LDAP masquerading queries.

Masquerading and altsrchost

Generally, the masquerading feature rewrites the Envelope Sender, and any subsequent actions to be performed on the message will be "triggered" from the masqueraded address. However, when you run the `altscrchost` command from the CLI, the altsrchost mappings are triggered from the original address (and not the modified, masqueraded address).

For more information, see Using Virtual Gateway™ Technology, page 2-59 and Review: Email Pipeline, page 2-73.

Configuring Static Masquerading Tables

You configure the static masquerading table of mappings by using the `edit -> masquerade` subcommand of the `listenerconfig` command. Alternatively, you can import a file containing the mappings. See Importing a Masquerading Table, page 2-19. The subcommand creates and maintains a table that maps input addresses, usernames, and domains to new addresses and domains. See Chapter 3, "LDAP Queries" for more information on LDAP masquerading queries.

When messages are injected into the system, the table is consulted, and the message is rewritten if a match in the header is found.

A domain masquerading table is constructed as follows:

Table 2-3 Masquerading Table Syntax

Left-hand Side (LHS)	Separator	Right-hand Side (RHS)
a list of one or more usernames and/or domains to match	whitespace (space or tab character)	the rewritten username and/or domain

The following table lists valid entries in the masquerading table:

Left-hand Side (LHS)	Right-hand Side (RHS)
username	username@domain
This entry specifies a username to match. Incoming email messages matching a username on the left-hand side are matched and rewritten with the address on the right-hand size. The right-hand side must be a full address.	
user@domain	username@domain
The entry specifies an exact address to match. Incoming messages matching a full address on the left-hand side are rewritten with the address listed on the right-hand side. The right-hand side must be a full address.	
@domain	**@domain**
This entry specifies any address with the specified domain. The original domain on the left-hand side is replaced with the domain in the right-hand side, leaving the username intact.	
@.partialdomain	**@domain**
This entry specifies any address with the specified domain. The original domain on the left-hand side is replaced with the domain in the right-hand side, leaving the username intact.	
ALL	**@domain**
The ALL entry matches bare addresses and rewrites them with the address on the right-hand side. The right-hand side must be a domain preceded by an "@". This entry always has the lowest precedence regardless of its location in the table. **Note** You can use the ALL entry for private listeners only.	

- Rules are matched by the order in which they appear in the masquerading table.
- Addresses in the From:, To:, and CC: fields in the headers are matched and rewritten upon receiving by default. You can also configure the option to match and rewrite the Envelope Sender. Enable and disable the Envelope Sender and which headers to rewrite using the `config` subcommand.
- You can comment out lines in the table using a number symbol (#) at the beginning of each line. Everything following a # to the end of the line will be considered a comment and ignored.
- A masquerading table is limited to 400,000 entries, whether you create them via the `new` subcommand or import them from a file.

Sample Masquerading Table for a Private Listener

```
# sample Masquerading file

@.example.com @example.com # Hides local subdomains in the header

sales sales_team@success.com

@techsupport tech_support@biggie.com

user@localdomain user@company.com

ALL @bigsender.com
```

Importing a Masquerading Table

A traditional sendmail `/etc/mail/genericstable` file can be imported. To import a `genericstable` file, first see Appendix B, "Accessing the Appliance" to ensure that you can access the appliance.

Place the `genericstable` file in the configuration directory, and then use the `import` subcommand of the `masquerade` subcommand to upload the file. Use the commands in this order:

`listenerconfig` -> `edit` -> *injector_number* -> `masquerade` -> `import`

Alternatively, you can use the `export` subcommand to download the existing configuration. A file (whose name you specify) will be written to the configuration directory. You can modify this file outside of the CLI and then import it again.

When you use the `import` subcommand, ensure that the file contains only valid entries. If there is an invalid entry (for example, a left-hand side with no right-hand side), the CLI reports syntax errors when you import the file. If there is a syntax error during import, no mappings in the entire file are imported.

Remember to issue the `commit` command after you import a `genericstable` file so that the configuration changes for the listener take effect.

Example Masquerading

In this example, the `masquerade` subcommand of `listenerconfig` is used to construct a domain masquerading table for the private listener named "OutboundMail" on the PrivateNet interface.

First, the option to use LDAP for masquerading is declined. (For information on configuring LDAP masquerading queries, see See Chapter 3, "LDAP Queries" for more information on LDAP masquerading queries.)

Then, a partial domain notation of **`@.example.com`** is mapped to **`@example.com`** so that any email sent from any machine in the subdomain of `.example.com` will be mapped to `example.com`. Then, the username **`joe`** is mapped to the domain **`joe@example.com`**. The domain masquerading table is then printed to confirm both entries, and then exported to a file named `masquerade.txt`. The `config` subcommand is used to disable re-writing addresses in the CC: field, and finally, the changes are committed.

```
mail3.example.com> listenerconfig


Currently configured listeners:
```

```
1. InboundMail (on PublicNet, 192.168.2.1) SMTP TCP Port 25 Public

2. OutboundMail (on PrivateNet, 192.168.1.1) SMTP TCP Port 25 Private


Choose the operation you want to perform:

- NEW - Create a new listener.

- EDIT - Modify a listener.

- DELETE - Remove a listener.

- SETUP - Change global settings.

[]> edit


Enter the name or number of the listener you wish to edit.

[]> 2


Name: OutboundMail

Type: Private

Interface: PrivateNet (192.168.1.1/24) TCP Port 25

Protocol: SMTP

Default Domain:

Max Concurrency: 600 (TCP Queue: 50)

Domain Map: Disabled

TLS: No

SMTP Authentication: Disabled

Bounce Profile: Default

Footer: None

LDAP: Off


Choose the operation you want to perform:

- NAME - Change the name of the listener.

- INTERFACE - Change the interface.
```

```
- LIMITS - Change the injection limits.

- SETUP - Configure general options.

- HOSTACCESS - Modify the Host Access Table.

- BOUNCECONFIG - Choose the bounce profile to use for messages injected on this
listener.

- MASQUERADE - Configure the Domain Masquerading Table.

- DOMAINMAP - Configure domain mappings.

- LDAPACCEPT - Configure an LDAP query to determine whether a recipient address should
be accepted or bounced/dropped.

- LDAPROUTING - Configure an LDAP query to reroute messages.

- LDAPGROUP - Configure an LDAP query to determine whether a sender or

recipient is in a specified group.

- SMTPAUTH - Configure an SMTP authentication.

[]> masquerade


Do you want to use LDAP for masquerading?  [N]> n


Domain Masquerading Table


There are currently 0 entries.

Masqueraded headers: To, From, Cc


Choose the operation you want to perform:

- NEW - Create a new entry.

- DELETE - Remove an entry.

- PRINT - Display all entries.

- IMPORT - Import all entries from a file.

- EXPORT - Export all entries to a file.

- CONFIG - Configure masqueraded headers.

- CLEAR - Remove all entries.
```

```
[]> new

Enter the source address or domain to masquerade.
Usernames like "joe" are allowed.
Full addresses like "user@example.com" are allowed.
Full addresses with subdomain wildcards such as "username@.company.com" are allowed.
Domains like @example.com and @.example.com are allowed.
Hosts like @training and @.sales are allowed.
[]> @.example.com

Enter the masqueraded address or domain.
Domains like @example.com are allowed.
Full addresses such as user@example.com are allowed.
[]> @example.com

Entry mapping @.example.com to @example.com created.
Domain Masquerading Table

There are currently 1 entries.
Masqueraded headers: To, From, Cc

Choose the operation you want to perform:
- NEW - Create a new entry.
- DELETE - Remove an entry.
- PRINT - Display all entries.
- IMPORT - Import all entries from a file.
- EXPORT - Export all entries to a file.
- CONFIG - Configure masqueraded headers.
- CLEAR - Remove all entries.
```

```
[]> new

Enter the source address or domain to masquerade.
Usernames like "joe" are allowed.
Full addresses like "user@example.com" are allowed.
Full addresses with subdomain wildcards such as "username@.company.com" are allowed.
Domains like @example.com and @.example.com are allowed.
Hosts like @training and @.sales are allowed.
[]> joe

Enter the masqueraded address.
Only full addresses such as user@example.com are allowed.
[]> joe@example.com

Entry mapping joe to joe@example.com created.
Domain Masquerading Table

There are currently 2 entries.
Masqueraded headers: To, From, Cc

Choose the operation you want to perform:
- NEW - Create a new entry.
- DELETE - Remove an entry.
- PRINT - Display all entries.
- IMPORT - Import all entries from a file.
- EXPORT - Export all entries to a file.
- CONFIG - Configure masqueraded headers.
- CLEAR - Remove all entries.
[]> print
```

```
@.example.com    @example.com

joe    joe@example.com


Domain Masquerading Table


There are currently 2 entries.

Masqueraded headers: To, From, Cc


Choose the operation you want to perform:

- NEW - Create a new entry.

- DELETE - Remove an entry.

- PRINT - Display all entries.

- IMPORT - Import all entries from a file.

- EXPORT - Export all entries to a file.

- CONFIG - Configure masqueraded headers.

- CLEAR - Remove all entries.

[]> export


Enter a name for the exported file:

[]> masquerade.txt


Export completed.

Domain Masquerading Table


There are currently 2 entries.

Masqueraded headers: To, From, Cc


Choose the operation you want to perform:
```

```
- NEW - Create a new entry.

- DELETE - Remove an entry.

- PRINT - Display all entries.

- IMPORT - Import all entries from a file.

- EXPORT - Export all entries to a file.

- CONFIG - Configure masqueraded headers.

- CLEAR - Remove all entries.

[]> config


Do you wish to masquerade Envelope Sender?

[N]> y


Do you wish to masquerade From headers?

[Y]> y


Do you wish to masquerade To headers?

[Y]> y


Do you wish to masquerade CC headers?

[Y]> n


Do you wish to masquerade Reply-To headers?

[Y]> n


Domain Masquerading Table


There are currently 2 entries.


- NEW - Create a new entry.
```

```
- DELETE - Remove an entry.

- PRINT - Display all entries.

- IMPORT - Import all entries from a file.

- EXPORT - Export all entries to a file.

- CONFIG - Configure masqueraded headers.

- CLEAR - Remove all entries.

[]>


Name: OutboundMail

Type: Private

Interface: PrivateNet (192.168.1.1/24) TCP Port 25

Protocol: SMTP

Default Domain:

Max Concurrency: 600 (TCP Queue: 50)

Domain Map: Disabled

TLS: No

SMTP Authentication: Disabled

Bounce Profile: Default

Footer: None

LDAP: Off


Choose the operation you want to perform:

- NAME - Change the name of the listener.

- INTERFACE - Change the interface.

- LIMITS - Change the injection limits.

- SETUP - Configure general options.

- HOSTACCESS - Modify the Host Access Table.

- BOUNCECONFIG - Choose the bounce profile to use for messages injected on this
listener.
```

```
- MASQUERADE - Configure the Domain Masquerading Table.

- DOMAINMAP - Configure domain mappings.

- LDAPACCEPT - Configure an LDAP query to determine whether a recipient address should
be accepted or bounced/dropped.

- LDAPROUTING - Configure an LDAP query to reroute messages.

- LDAPGROUP - Configure an LDAP query to determine whether a sender or

recipient is in a specified group.

- SMTPAUTH - Configure an SMTP authentication.

[]>


Currently configured listeners:

1. InboundMail (on PublicNet, 192.168.2.1) SMTP TCP Port 25 Public

2. OutboundMail (on PrivateNet, 192.168.1.1) SMTP TCP Port 25 Private


Choose the operation you want to perform:

- NEW - Create a new listener.

- EDIT - Modify a listener.

- DELETE - Remove a listener.

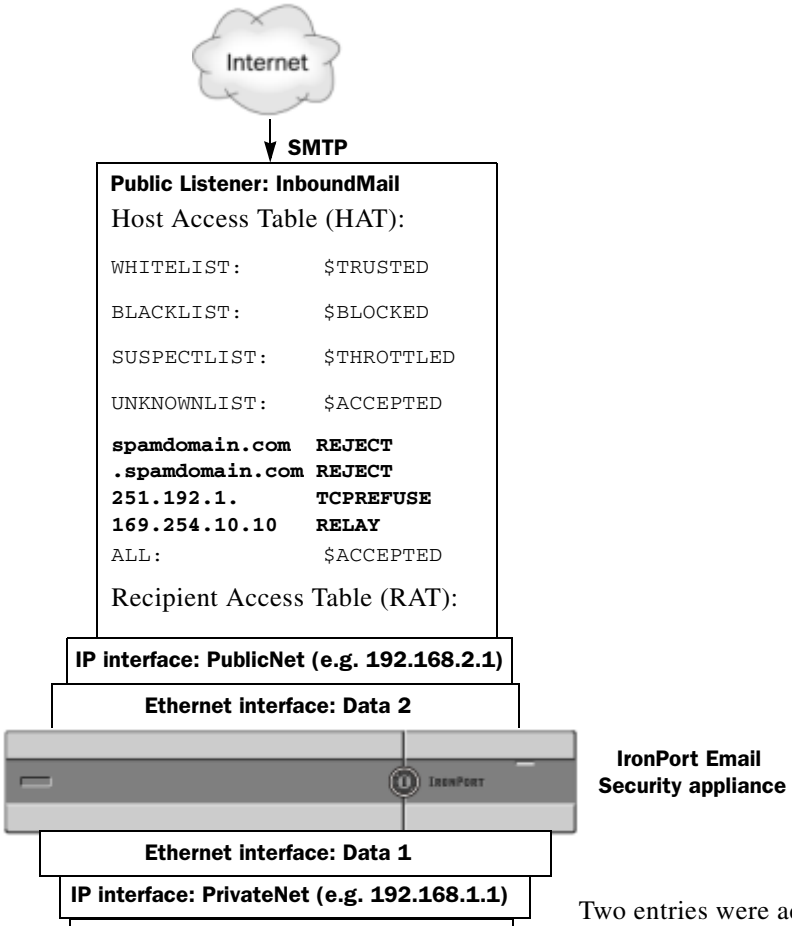
- SETUP - Change global settings.

[]>


mail3.example.com> commit
```

Our Enterprise Gateway configuration now looks like this:

Figure 2-5 Masquerading Defined for a Private Listener



Two entries were added to the masquerading table for this private listener. The first entry masks all subdomains of a network (e.g. `exchange.example.com` is rewritten as `example.com`). The second entry rewrites the bare address `joe` as `joe@example.com`.

The Domain Map Feature

You can configure a "domain map" for listeners. For each listener you configure, you can construct a domain map table which rewrites the Envelope Recipient for each recipient in a message that matches a domain in the domain map table. This feature is similar to the sendmail "Domain Table" or Postfix "Virtual Table" feature. Only the Envelope Recipient is affected; the "To:" headers are not re-written by this feature.

Note The processing of the domain map feature happens immediately before the RAT and right after Default Domain is evaluated. Refer to "Understanding the Email Pipeline" in the *Cisco IronPort AsyncOS for Email Configuration Guide.*

A common implementation of the domain map feature is to accept incoming mail for more than one legacy domain. For example, if your company has acquired another company, you could construct a domain map on the Cisco IronPort appliance to accept messages for the acquired domain and rewrite the Envelope Recipients to your company's current domain.

Note You can configure up to 20,000 separate, unique domain mappings.

Table 2-4 Domain Map Table Example Syntax

Left Side	Right Side	Comments
username@example.com	**username2@example.net**	Only complete address for the right side
user@.example.com	**user2@example.net**	
@example.com	**user@example.net** ***or*** **@example.net**	Complete address or fully-qualified domain name.
@.example.com	**user@example.net** ***or*** **@example.net**	

In the following example, the `domainmap` subcommand of the `listenerconfig` command is used to create a domain map for the public listener "InboundMail." Mail for the domain and any subdomain of `oldcompanyname.com` is mapped to the domain `example.com`. The mapping is then printed for confirmation. Contrast this example with the configuration of placing both domains in the listener's RAT: the domain map feature will actually rewrite the Envelope Recipient of `joe@oldcomapanyname.com` to `joe@example.com`, whereas placing the domain `oldcompanyname.com` in the listener's RAT will simply accept the message for `joe@oldcompanyname.com` and route it without rewriting the Envelope Recipient. Also, contrast this example with the alias table feature. Alias tables *must* resolve to an explicit address; they cannot be constructed to map "*any username*`@domain`" to "*the same username*`@newdomain`."

```
mail3.example.com> listenerconfig


Currently configured listeners:

1. Inboundmail (on PublicNet, 192.168.2.1) SMTP TCP Port 25 Public

2. Outboundmail (on PrivateNet, 192.168.1.1) SMTP TCP Port 25 Private


Choose the operation you want to perform:
```

```
- NEW - Create a new listener.

- EDIT - Modify a listener.

- DELETE - Remove a listener.

- SETUP - Change global settings.

[]> edit


Enter the name or number of the listener you wish to edit.

[]> 1


Name: InboundMail

Type: Public

Interface: PublicNet (192.168.2.1/24) TCP Port 25

Protocol: SMTP

Default Domain:

Max Concurrency: 1000 (TCP Queue: 50)

Domain Map: Disabled

TLS: No

SMTP Authentication: Disabled

Bounce Profile: Default

Use SenderBase For Reputation Filters and IP Profiling: Yes

Footer: None

LDAP: Off


Choose the operation you want to perform:

- NAME - Change the name of the listener.

- INTERFACE - Change the interface.

- LIMITS - Change the injection limits.

- SETUP - Configure general options.

- HOSTACCESS - Modify the Host Access Table.
```

```
- RCPTACCESS - Modify the Recipient Access Table.

- BOUNCECONFIG - Choose the bounce profile to use for messages injected on this
listener.

- MASQUERADE - Configure the Domain Masquerading Table.

- DOMAINMAP - Configure domain mappings.

[]> domainmap


Domain Map Table


There are currently 0 Domain Mappings.

Domain Mapping is: disabled


Choose the operation you want to perform:

- NEW - Create a new entry.

- IMPORT - Import domain mappings from a file.

[]> new


Enter the original domain for this entry.

Domains such as "@example.com" are allowed.

Partial hostnames such as "@.example.com" are allowed.

Email addresses such as "test@example.com" and "test@.example.com"

 are also allowed.

[]> @.oldcompanyname.com


Enter the new domain for this entry.

The new domain may be a fully qualified

such as "@example.domain.com" or a complete

email address such as "test@example.com"

[]> @example.com
```

```
Domain Map Table


There are currently 1 Domain Mappings.

Domain Mapping is: enabled


Choose the operation you want to perform:

- NEW - Create a new entry.

- EDIT - Modify an entry.

- DELETE - Remove an entry.

- PRINT - Display all domain mappings.

- IMPORT - Import domain mappings from a file.

- EXPORT - Export domain mappings to a file.

- CLEAR - Clear all domain mappings.

[]> print


@.oldcompanyname.com --> @example.com


Domain Map Table


There are currently 1 Domain Mappings.

Domain Mapping is: enabled


Choose the operation you want to perform:

- NEW - Create a new entry.

- EDIT - Modify an entry.

- DELETE - Remove an entry.

- PRINT - Display all domain mappings.

- IMPORT - Import domain mappings from a file.
```

```
- EXPORT - Export domain mappings to a file.

- CLEAR - Clear all domain mappings.

[]>


Name: InboundMail

Type: Public

Interface: PublicNet (192.168.2.1/24) TCP Port 25

Protocol: SMTP

Default Domain:

Max Concurrency: 1000 (TCP Queue: 50)

Domain Map: Enabled

TLS: No

SMTP Authentication: Disabled

Bounce Profile: Default

Use SenderBase For Reputation Filters and IP Profiling: Yes

Footer: None

LDAP: Off


Choose the operation you want to perform:

- NAME - Change the name of the listener.

- INTERFACE - Change the interface.

- LIMITS - Change the injection limits.

- SETUP - Configure general options.

- HOSTACCESS - Modify the Host Access Table.

- RCPTACCESS - Modify the Recipient Access Table.

- BOUNCECONFIG - Choose the bounce profile to use for messages injected on this
listener.

- MASQUERADE - Configure the Domain Masquerading Table.
```

```
- DOMAINMAP - Configure domain mappings.

[]>
```

Importing and Exporting a Domain Map Table

To import or export a domain map table, first see Appendix B, "Accessing the Appliance" to ensure that you can access the appliance.

Create a text file of entries of domains to map. Separate the entries with white space (either a tab character or spaces). Comment out lines in the table using a number symbol (#) at the beginning of each line.

Place the file in the configuration directory, and then use the `import` subcommand of the `domain` subcommand to upload the file. Use the commands in this order:

`listenerconfig` -> `edit` -> *inejctor_number* -> `domainmap` -> `import`

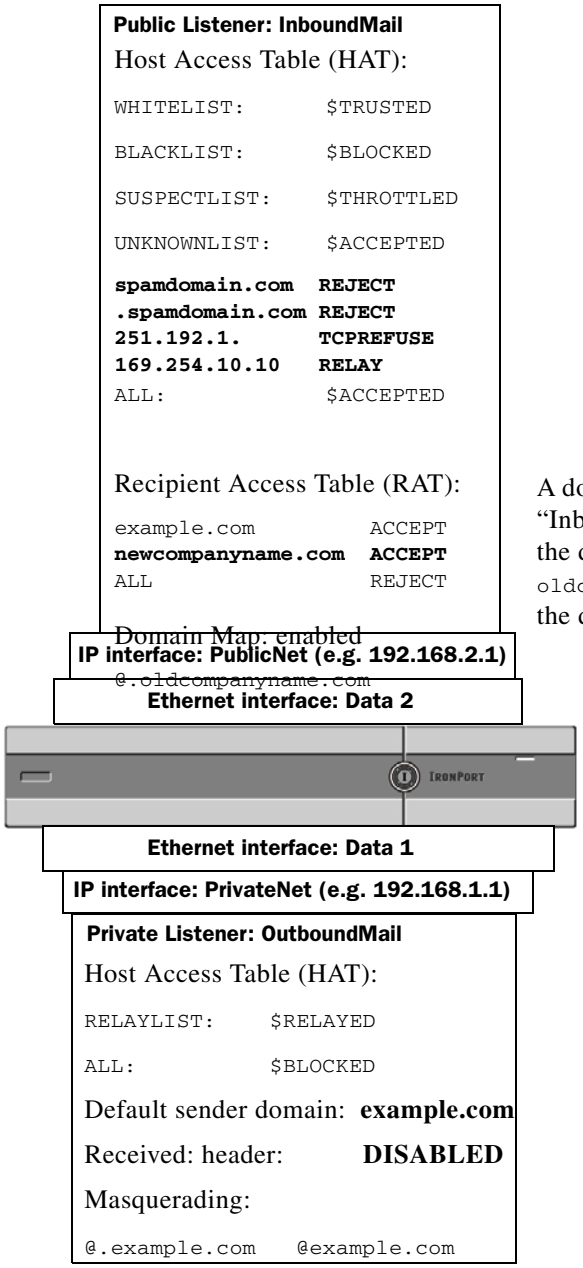
Alternatively, you can use the `export` subcommand to download the existing configuration. A file (whose name you specify) will be written to the configuration directory. You can modify this file outside of the CLI and then import it again.

When you use the `import` subcommand, ensure that the file contains only valid entries. If there is an invalid entry (for example, a left-hand side with no right-hand side), the CLI reports syntax errors when you import the file. If there is a syntax error during import, no mappings in the entire file are imported.

Remember to issue the `commit` command after you import a domain map table file so that the configuration changes for the listener take effect.

Our Enterprise Gateway configuration now looks like this:

Figure 2-6 Domain Map Defined for a Public Listener



A domain map for the public listener "InboundMail" was created. Mail for the domain and any subdomain of `oldcompanyname.com` is mapped to the domain `example.com`.

Directing Bounced Email

Bounced email is an inevitable part of any email delivery. Your Cisco IronPort appliance is able to process bounced email in a number of highly configurable ways.

Please note, this section describes how to control how your Cisco IronPort appliance generates outgoing bounces (based on incoming mail). To control how your Cisco IronPort appliance controls incoming bounces (based on outgoing mail) use Cisco IronPort Bounce Verification (see Cisco IronPort Bounce Verification, page 2-53).

Handling Undeliverable Email

The Cisco IronPort AsyncOS operating system classifies undeliverable email, or "bounced messages," into the following categories:

"Conversational" bounces: **The remote domain bounces the message during the initial SMTP conversation.**	
Soft bounces	A message that is temporarily undeliverable. For example, a user's mailbox may be full. These messages can be retried at a later time. (e.g. An SMTP 4XX error code.)
Hard bounces	A message that is permanently undeliverable. For example, the user no longer exists for that domain. These messages will not be retried. (e.g. An SMTP 5XX error code.)
"Delayed" (or "Non-conversational") bounces: **The remote domain accepts the message for delivery, only to bounce it at a later time.**	
Soft bounces	A message that is temporarily undeliverable. For example, a user's mailbox may be full. These messages can be retried at a later time. (e.g. An SMTP 4XX error code.)
Hard bounces	A message that is permanently undeliverable. For example, the user no longer exists for that domain. These messages will not be retried. (e.g. An SMTP 5XX error code.)

You use the Bounce Profiles page on the Network menu in the GUI (or the `bounceconfig` command) to configure how Cisco IronPort AsyncOS handles hard and soft conversational bounces for each listener you create. You create bounce profiles and then apply profiles to each listener via the Network > Listeners page (or the `listenerconfig` command). You can also assign bounce profiles to specific messages using message filters. (See Chapter 6, "Using Message Filters to Enforce Email Policies" for more information.)

Notes on Soft and Hard Bounces

- For conversational soft bounces, a soft bounce event is defined as each time a recipient delivery temporarily fails. A single recipient may incur several soft bounce events. You use the Bounce Profiles page or the `bounceconfig` command to configure parameters for each soft bounce event. (See Bounce Profile Parameters, page 2-37.)
- By default, the system generates a bounce message and sends it to the original sender for each hard bounced recipient. (The message is sent to the address defined in the Envelope Sender address of the message envelope. Envelope From is also commonly referred to as the Envelope Sender.) You can disable this feature and instead rely on log files for information about hard bounces. (See "Logging" in the *Cisco IronPort AsyncOS for Email Daily Management Guide*.)
- Soft bounces become hard bounces after the maximum time in queue or the maximum number of retries, whichever comes first.

Bounce Profile Parameters

When configuring a bounce profile, the following parameters control how conversational bounces are handled per message:

Table 2-5 Bounce Profile Parameters

Maximum number of retries	The number of times the system should try to reconnect to the recipient host to re-deliver the soft bounced message before treating it as a hard bounced message. The default is 100 retries.
Maximum number of seconds in queue	The amount of time the system should spend trying connect to the recipient host to re-deliver the soft bounced message before treating it as a hard bounced message. The default is 259,200 seconds (72 hours).
Initial number of seconds to wait before retrying a message	The amount of time the system should wait before the first attempt to re-deliver the soft bounced message. The default is 60 seconds. Set the initial retry time to a high value to reduce the frequency of soft bounce attempts. Conversely, to increase the frequency, lower the value.
Maximum number of seconds to wait before retrying a message	The maximum amount of time the system should wait before trying to re-deliver the soft bounced message. The default is 3,600 seconds (1 hour). This is not the interval between each subsequent try; rather, it is another parameter that can be used to control the number of retries. The initial retry interval is limited on the high end by the maximum retry interval. If the calculated retry interval period exceeds the maximum retry interval then the maximum retry interval is used instead.
Hard bounce message generation format	Specify whether hard bounce message generation is enabled or disabled. If it is enabled, you can choose the format of the message. By default, bounce messages generated use the DSN format (RFC 1894). You can select a custom notification template to use for bounce messages. For more information, see the "Text Resources" chapter of the *Cisco IronPort AsyncOS for Email Configuration Guide*. You can also choose whether or not to parse the DSN status field from the bounce response. If you choose "Yes," AsyncOS searches the bounce response for a DSN status code (RFC 3436) and uses the code in the Status field of the delivery status notification.
Send delay warning messages	Specify whether or not to send delay warnings. If enabled, specify the minimum interval between messages as well as the maximum number of retries to send. You can select a custom notification template to use for warning messages. For more information, see the "Text Resources" chapter of the *Cisco IronPort AsyncOS for Email Configuration Guide*.
Specify Recipient for Bounces	You can bounce messages to an alternate address rather than the default of the Envelope Sender address.
Use DomainKeys signing for bounce and delay messages	You can select a DomainKeys profile to use for signing bounce and delay messages. For information on DomainKeys, see DomainKeys and DKIM Authentication: Overview, page 5-2.
Global Settings	
Configure these settings via the Edit Global Settings link on the Bounce Profiles page or by editing the default bounce profile via the `bounceconfig` command in the CLI.	

Table 2-5 Bounce Profile Parameters (Continued)

Initial number of seconds to wait before retrying an unreachable host	The amount of time the system should wait before retrying a host that is unreachable. The default is 60 seconds.
Max interval allowed between retries to an unreachable host	The maximum amount of time the system should wait before retrying a host that is unreachable. The default is 3,600 seconds (1 hour). When the delivery initially fails due to the host being down, it will start with the minimum number of seconds retry value, and for each subsequent retry to the downed host, will increase the duration, up to this maximum number of seconds value.

Hard Bounces and the status Command

When hard bounce message generation is enabled, the following counters in the `status` and `status detail` commands increment each time the appliance generates a hard bounce message for delivery:

```
Counters:                          Reset          Uptime        Lifetime

    Receiving

    Messages Received              0              0             0

    Recipients Received            0              0             0

    Gen. Bounce Recipients         0              0             0
```

For more information, see "Monitoring and Managing via the CLI" in the *Cisco IronPort AsyncOS for Email Daily Management Guide*. When hard bounce message generation is disabled, none of these counters increments when a recipient hard bounces.

Note The Envelope Sender address of the message envelope is different than the From: in the message headers. Cisco IronPort AsyncOS can be configured to send hard bounce messages to an email address different than the Envelope Sender address.

Conversational Bounces and SMTP Routes Message Filter actions

Mappings for SMTP Routes and message filter actions are not applied to the routing of SMTP bounce messages generated by the appliance as a result of a conversational bounce. When an Cisco IronPort appliance receives a conversational bounce message, it generates an SMTP bounce message back to the Envelope Sender of the original message. In this case, the appliance is actually generating the message, so any SMTP Routes that apply to an injected message for relaying do not apply.

Example Bounce Profiles

Consider these two examples using different bounce profile parameters:

Table 2-6 Example 1: Bounce Profile Parameters

Parameter	Value
Max number of retries	2
Max number of seconds in queue	259,200 seconds (72 hours)
Initial number of seconds before retrying	60 seconds
Max number of seconds to wait before retrying	60 seconds

In Example 1, the first recipient delivery attempt is made at t=0, immediately after the message is injected into the Cisco IronPort appliance. With the default initial retry time of 60 seconds, the first retry attempt is made approximately one minute later at t=60. The retry interval is calculated and it is determined to use the maximum retry interval of 60 seconds. Thus, the second retry attempt is made at approximately t=120. Immediately after this retry attempt, the system generates a hard bounce message for that recipient because the maximum number of retries is two.

Table 2-7 Example 2: Bounce Profile Parameters

Parameter	Value
Max number of retries	100
Max number of seconds in queue	100 seconds
Initial number of seconds before retrying	60 seconds
Max number of seconds to wait before retrying	120 seconds

In Example 2, the first delivery attempt is made at t=0 and the first retry is made at t=60. The system hard bounces the message immediately before the next delivery attempt (scheduled to occur at t=120) because it has exceeded the maximum time in queue of 100 seconds.

Delivery Status Notification Format

Bounce messages generated by the system, by default, use the Delivery Status Notification (DSN) format for both hard and soft bounces. DSN is a format defined by RFC 1894 (see `http://www.faqs.org/rfcs/rfc1894.html`) that "defines a MIME content-type that may be used by a message transfer agent (MTA) or electronic mail gateway to report the result of an attempt to deliver a message to one or more recipients." By default, the delivery status notification includes an explanation of the delivery status and the original message if the message size is less than 10k. If the message size is greater than 10k, the delivery status notification includes the message headers only. If the message headers exceed 10k, the delivery status notification truncates the headers. If you want include messages (or message headers) that are greater than 10k in the DSN, you can use the `max_bounce_copy` parameter in the `bounceconfig` command (this parameter is only available from the CLI).

Delay Warning Messages

Time in Queue Messages (delay notification messages) generated by the system also use the DSN format. Change the default parameters by using the Bounce Profiles page on the Network menu (or the `bounceconfig` command) to edit existing or create new bounce profiles and change the default values for:

- The minimum interval between sending delay warning messages.
- The maximum number of delay warning messages to send per recipient.

Delay Warning Messages and Hard Bounces

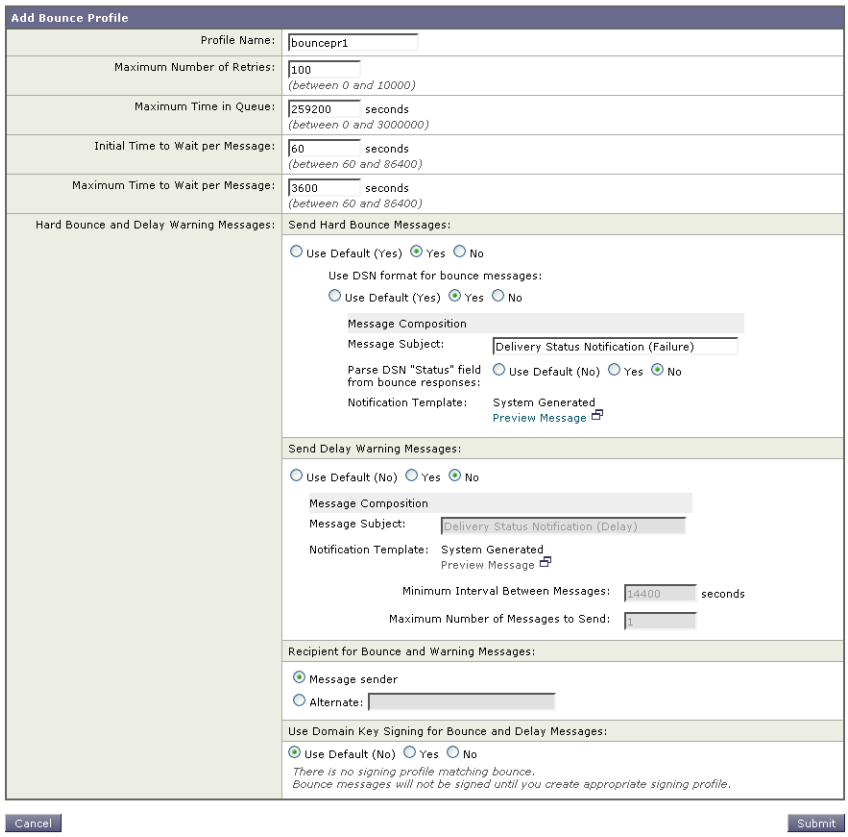
Note that it is possible to receive both a delay warning and a hard bounce for the same message *simultaneously*, if you have set a very small durations for both the "Maximum Time in Queue" setting and the minimum interval setting for "Send Delay Warning Messages." Cisco IronPort Systems recommends using the default values for these settings as a minimum if you choose to enable sending of delay warning messages.

Further, delay warning messages and bounce messages originated by the appliance may be delayed by as much as 15 minutes during processing.

Creating a New Bounce Profile

In the following example, a bounce profile named `bouncepr1` is created using the Bounce Profiles page. In this profile, all hard bounced messages are sent to the alternate address `bounce-mailbox@example.com`. Delay warnings messages are enabled. One warning message will be sent per recipient, and the default value of 4 hours (14400 seconds) between warning messages is accepted.

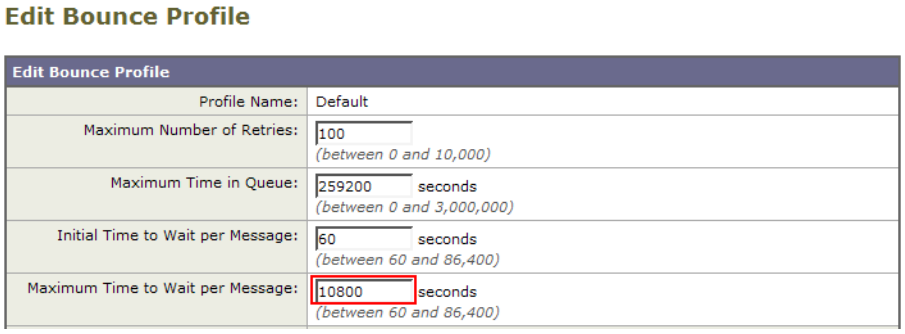
Figure 2-7 Creating a Bounce Profile



Editing the Default Bounce Profile

You can edit any bounce profile by clicking its name in the Bounce Profiles listing. You can also edit the default bounce profile. In this example, the default profile is edited to increase the `maximum number of seconds to wait before retrying unreachable hosts` from 3600 (one hour) to 10800 (three hours):

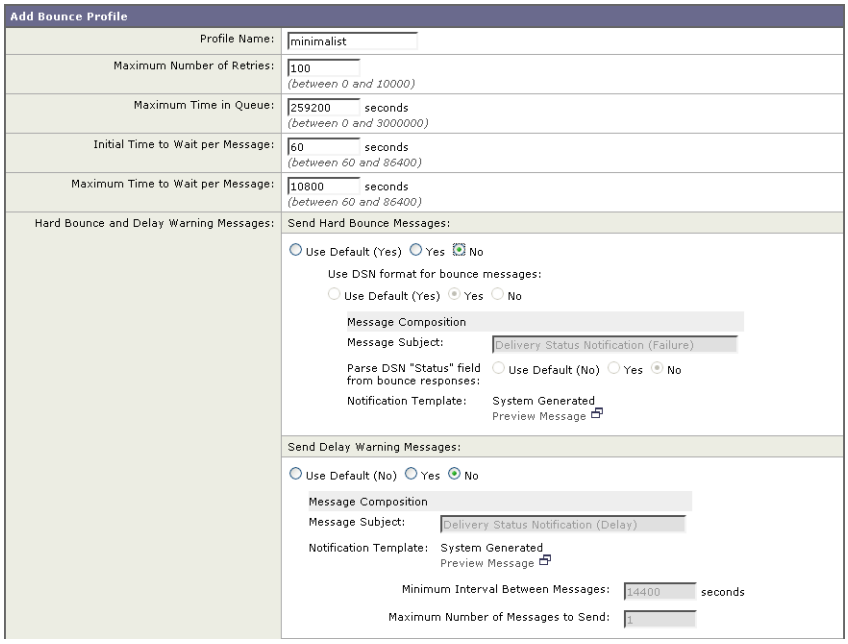
Figure 2-8 Editing the Default Bounce Profile



Example of a Minimalist Bounce Profile

In the following example, a bounce profile named `minimalist` is created. In this profile, messages are not retried when they bounce (zero maximum retries), and the maximum time to wait before retrying is specified. Hard bounce messages are disabled, and soft bounce warnings are not sent.

Figure 2-9 Creating a "Minimalist" Bounce Profile

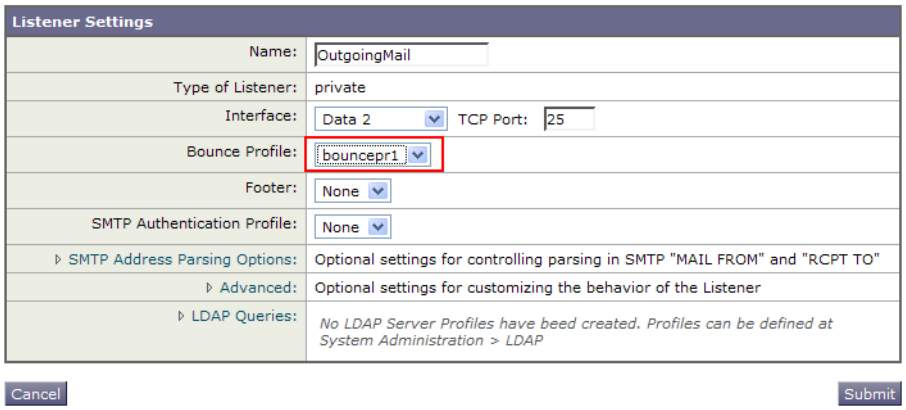


Applying Bounce Profiles to Listeners

Once you have created a bounce profile, you can apply that profile to a listener using the Network > Listeners page or the `listenerconfig` command.

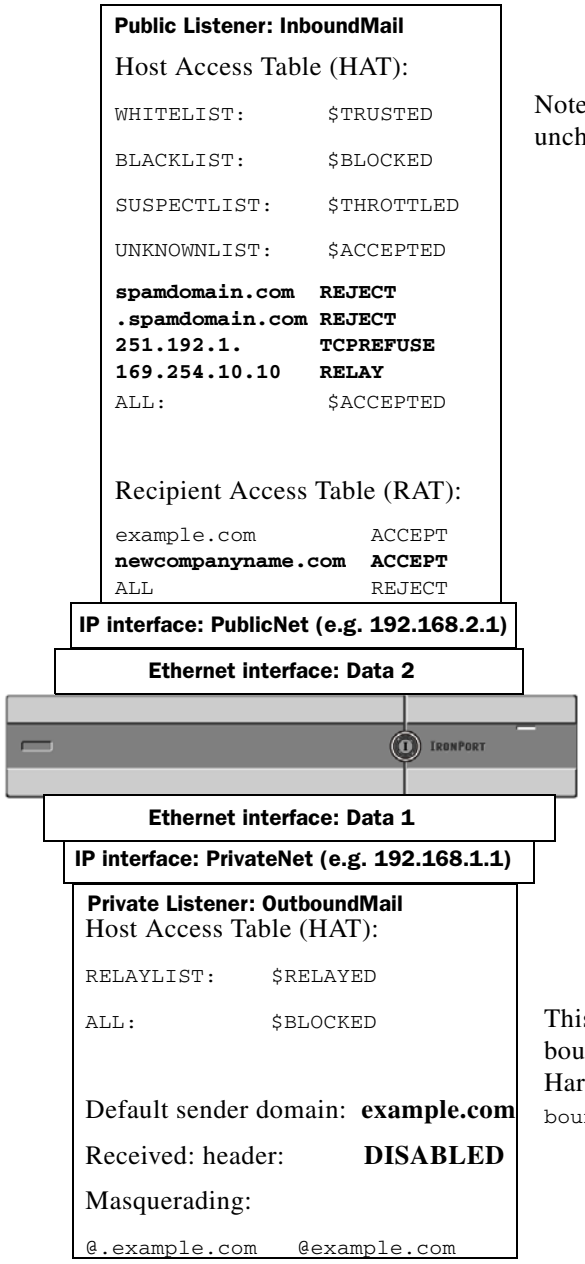
In the following example, the `bouncepr1` profile is applied to the OutgoingMail listener.

Figure 2-10 Creating a "Minimalist" Bounce Profile



At this point, our Email Gateway configuration looks like this:

Figure 2-11 Applying a Bounce Profile to a Private Listener



Note: This public listener remains unchanged.

This listener was modified to use a bounce profile named `bouncepr1`. Hard bounces are sent to the address: `bounce-mailbox@example.com`.

Controlling Email Delivery

Uncontrolled high-volume email delivery can overwhelm recipient domains. AsyncOS gives you full control of message delivery by defining the number of connections your appliance will open or the number of messages your appliance will send to each destination domain.

Using the Destination Controls feature (Mail Policies > Destination Controls in the GUI, or the `destconfig` command in the CLI), you can control:

Rate Limiting

- Concurrent Connections: number of simultaneous connections to remote hosts the appliance will attempt to open.
- Maximum Messages Per Connection: number of messages your appliance will send to a destination domain before the appliance initiates a new connection.
- Recipients: number of recipients the appliance will send to a given remote host in a given time period.
- Limits: how to apply the limits you have specified on a per-destination and per MGA hostname basis.

TLS

- Whether TLS connections to remote hosts will be accepted, allowed, or required (see Controlling TLS, page 2-47).
- Whether to send an alert when TLS negotiation fails when delivering a message to a remote host that requires a TLS connection. This is a global setting, not a per-domain setting.
- Assign a TLS certificate to use for all outbound TLS connections to remote hosts.

Bounce Verification

- Whether or not to perform address tagging via Cisco IronPort Bounce Verification (see Cisco IronPort Bounce Verification, page 2-53).

Bounce Profile

- Which bounce profile should be used by the appliance for a given remote host (the default bounce profile is set via the Network > Bounce Profiles page).

You can also control the default settings for unspecified domains.

Determining Which Interface is Used for Mail Delivery

Unless you specify the output interface via the `deliveryconfig` command or via a message filter (`alt-src-host`), or through the use of a virtual gateway, the output interface is selected by the AsyncOS routing table. Basically, selecting “auto” means to let AsyncOS decide.

In greater detail: local addresses are identified by applying the interface netmask to the interface IP address. Both of these are set via the Network > Interfaces page or by the `interfaceconfig` command (or during system setup). If the address space overlaps, the most specific netmask is used. If a destination is local, packets are sent via the appropriate local interface.

If the destination is not local, packets are sent to the default router (set via the Network > Routing page or with the `setgateway` command). The IP address of the default router is local. The output interface is determined by the rule for selecting the output interface for local addresses. For example, AsyncOS chooses the most specific IP address and netmask that include the default router's IP address.

The routing table is configured via the Network > Routing page (or via the `routeconfig` command). A matching entry in the routing table takes precedence over the default route. A more specific route take precedence over a less specific route.

Default Delivery Limits

Each outbound destination domain has its own outbound queue. Therefore, each domain has a separate set of concurrency limits as specified in the Destination Controls table. Further, each unique domain not listed specifically in the Destination Controls table uses another set of the "Default" limits as set in the table.

Working with Destination Controls

Use the Mail Policies > Destination Controls page in the GUI or the `destconfig` command in the CLI to create, edit, and delete Destination Control entries.

Controlling the Version of Internet Protocol Addresses

You can configure which version of Internet Protocol addresses to use for the connection to a domain. The Email Security appliance uses both Internet Protocol version 4 (IPv4) and Internet Protocol version (IPv6). You can configure a listener on the appliance to use one version of the protocol or both.

If the "Required" setting for either IPv4 or IPv6 is specified, the Cisco IronPort appliance will negotation a connection to the domain using an address of the specified version. If the domain doesn't use that IP address version, no email will be sent. If the "Preferred" setting for either IPv4 or IPv6 is specified, the Cisco IronPort appliance will first attempt to negotation a connection to the domain using an address of the specified version then fall back to the other if the first is not reachable.

Controlling the Number of Connections, Messages, and Recipients to a Domain

You may want to limit how your appliance will deliver email to avoid overwhelming remote hosts or your own internal groupware servers with email from your appliance.

For each domain, you can assign a maximum number of connections, outbound messages, and recipients that will never be exceeded by the system in a given time period. This "good neighbor" table is defined through the Destination Controls feature (Mail Policies > Destination Controls or the `destconfig` command — previously the `setgoodtable` command). You can specify the domain name using the following syntax:

```
domain.com
```

or

```
.domain.com
```

This syntax enables AsyncOS to specify destination controls for sub-domains such as sample.server.domain.com without entering each full subdomain address individually.

For connections, messages, and recipients, you set whether the limits you define are enforced for each Virtual Gateway address, or for the entire system. (Virtual Gateway address limits control the number of concurrent connections per IP interface. System-wide limits control the total number of connections the Cisco IronPort appliance will allow.)

You also set whether the limits you define are enforced for each MX record of the specified domain, or for the entire domain. (Many domains have multiple MX records defined for accepting email.)



Note The current system default is 500 connections per domain and 50 messages per connection.

These values are explained in Table 2-8.

Table 2-8 Values in the Destination Controls Table

Field	Description
Concurrent Connections	The maximum number of outbound connections that will be made by the Cisco IronPort appliance to a given host. (Note that the domain can include your internal groupware hosts.)
Maximum Messages Per Connection	The maximum number of messages allowed for a single outbound connection from the Cisco IronPort appliance to a given host before initiating a new connection.
Recipients	The maximum number of recipients allowed within the given period of time. "None" denotes that there is no recipient limit for the given domain. The minimum period of time — between 1 and 60 minutes — that the Cisco IronPort appliance will count the number of recipients. Specifying a time period of "0" disables the feature. **Note** If you change the recipient limit, AsyncOS resets the counters for all messages already in the queue. The appliance delivers the messages based on the new recipient limit.
Apply Limits	Specifies whether the limit will be applied (enforces) to the entire domain or to each mail exchange IP address specified for that domain. (Many domains have multiple MX records.) This setting applies to connection, message, and recipient limits. Specifies whether the limit will be applied system-wide or for each Virtual Gateway address. **Note** If you have configured groups of IP addresses, but you have not configured virtual gateways, do not configure apply limits per each virtual gateway. This setting is intended only for systems configured to use virtual gateways. For information on configuring virtual gateways, see Using Virtual Gateway™ Technology, page 2-59.

Note If limits are applied per each Virtual Gateway address, you can still effectively implement system-wide limits by setting the Virtual Gateway limit to the system-wide limit you want divided by the number of possible virtual gateways. For example, if you have four Virtual Gateway addresses configured, and you do not want to open more than 100 simultaneous connections to the domain `yahoo.com`, set the Virtual Gateway limit to 25 simultaneous connections.

Note The `delivernow` command, when acting on all domains, resets all counters tracked in the `destconfig` command.

Controlling TLS

You can also configure the TLS (Transport Layer Security) on a per-domain basis. If the "Required" setting is specified, a TLS connection will be negotiated from the Cisco IronPort appliance listener to MTA(s) for the domain. If the negotiation fails, no email will be sent through the connection. For more information, see Enabling TLS and Certificate Verification on Delivery, page 1-29.

You can specify whether the Cisco IronPort appliance sends an alert if the TLS negotiation fails when delivering messages to a domain that requires a TLS connection. The alert message contains name of the destination domain for the failed TLS negotiation. The Cisco IronPort appliance sends the alert message to all recipients set to receive Warning severity level alerts for System alert types. You can manage alert recipients via the System Administration > Alerts page in the GUI (or via the `alertconfig` command in the CLI).

To enable TLS connection alerts, click **Edit Global Settings** on the Destination Controls page or `destconfig -> setup` subcommand. This is a global setting, not a per-domain setting. For information on the messages that the appliance attempted to deliver, use the Monitor > Message Tracking page or the mail logs.

You must specify a certificate to use for all outgoing TLS connections. Use the **Edit Global Settings** on the Destination Controls page or `destconfig -> setup` subcommand to specify the certificate. For information on obtaining a certificate, see Obtaining Certificates, page 1-22.

For more information on alerts, see the "System Administration" chapter of the *Cisco IronPort AsyncOS for Email Configuration Guide*.

Controlling Cisco IronPort Bounce Verification Tagging

You can specify whether or not mail sent is tagged for bounce verification. You can specify this for the default, as well as specific destinations. Cisco IronPort suggests enabling bounce verification for the default, and then creating new destinations for specific exclusions. See Cisco IronPort Bounce Verification, page 2-53 for more information.

Controlling Bounces

In addition to controlling the number of connections and recipients will deliver to a remote host, you can also specify a bounce profile to be used for that domain. If specified, the bounce profile appears in the fifth column of the `destconfig` command. If you do not specify a bounce profile, the default bounce profile will be used. For more information, see Creating a New Bounce Profile, page 2-40.

Adding a New Destination Control Entry

To add a new Destination Control entry:

Step 1 Click **Add Destination**:

Step 2 Configure the entry.

Step 3 Submit and commit your changes.

Editing Destination Control Entries

To edit a Destination Control entry,

Step 1 Click the domain name in the Domain column on the Destination Control page.

Step 2 Make your changes.

Step 3 Submit and commit your changes.

Deleting Destination Control Entries

To delete one or more Destination Control entries,

Step 1 Select the entry or entries by marking the corresponding checkbox in the left column.

Step 2 Click **Delete**.

Step 3 Confirm the deletion.

Note that you cannot delete the default Destination Control entry.

Importing and Exporting Destination Control Configurations

If you are managing multiple domains, you can create a single configuration file to define Destination Control entries for all of the domains and import it onto the appliance. The format of the configuration file is similar to a Windows INI configuration file. The parameters for a domain are grouped in a section with the domain name as the section name. For example, use the section name `[example.com]` to group the parameters for the domain example.com. Any parameter that is not defined will be inherited from the default Destination Control entry. You can define the parameters for the default Destination Control entry by including a `[DEFAULT]` section in the configuration file.

Importing the configuration file overwrites all of appliance's Destination Control entries, except for the default entry unless the configuration file includes the `[DEFAULT]` section. All other existing Destination Control entries will be deleted.

You can define any of the following parameters for a domain in the configuration file. All parameters are required for the `[DEFAULT]` section except for the `bounce_profile` parameter:

Table 2-9 Destination Control Configuration File Parameters

Parameter Name	Description
ip_sort_pref	Specifies the Internet Protocol version for the domain. Enter one of the following values: • `PREFER_V6` for "IPv6 Preferred" • `REQUIRE_v6` for "IPv6 Required" • `PREFER_V4` for "IPv4 Preferred" • `REQUIRE_v4` for "IPv4 Required"
`max_host_concurrency`	The maximum number of outbound connections that will be made by the Cisco IronPort appliance to a given host. If you define this parameter for a domain, the `limit_type` and `limit_apply` parameters must also be defined.
`max_messages_per_connection`	The maximum number of messages allowed for a single outbound connection from the Cisco IronPort appliance to a given host before initiating a new connection.
`recipient_minutes`	The period of time — between 1 and 60 minutes — that the Cisco IronPort appliance will count the number of recipients. Leave undefined if no recipient limit should be applied.

Table 2-9 Destination Control Configuration File Parameters

Parameter Name	Description
`recipient_limit`	The maximum number of recipients allowed within the given period of time. Leave undefined if no recipient limit should be applied. If you define this parameter for a domain, the `recipient_minutes`, `limit_type`, and `limit_apply` parameters must also be defined.
`limit_type`	Specifies whether the limit will be applied to the entire domain or to each mail exchange IP address specified for that domain. Enter one of the following values: • `0` (or `host`) for the domain • `1` (or `MXIP`) for the mail exchange IP address
`limit_apply`	Specifies whether the limit will be applied system-wide or for each Virtual Gateway address. Enter one of the following values: • `0` (or `system`) for system-wide • `1` (or `VG`) for Virtual Gateway
`bounce_validation`	Specifies whether to turn on bounce validation address tagging. Enter one of the following values: • `0` (or `off`) • `1` (or `on`)
`table_tls`	Specifies the TLS setting for the domain. See Enabling TLS and Certificate Verification on Delivery, page 1-29 for more information. Enter one of the following values: • `0` (or `off`) • `1` (or `on`) for “Preferred” • 2 (or `required`) for “Required” • 3 (or `on_verify`) for “Preferred (Verify)” • 4 (or `require_verify`) for “Required (Verify)” Strings are not case sensitive.
`bounce_profile`	Name of the bounce profile to use. This cannot be used in the `[DEFAULT]` destination control entry.

Table 2-9 Destination Control Configuration File Parameters

Parameter Name	Description
`send_tls_req_alert`	Whether to send an alert if the required TLS connection fails. Enter one of the following values: • `0` (or `off`) • `1` (or `on`) This is a global setting and can only be used in the `[DEFAULT]` destination control entry.
certificate	Certificate used for outgoing TLS connections. This is a global setting and can only be used in the `[DEFAULT]` destination control entry. **Note** If you do not specify a certificate, AsyncOS assigns the demonstration certificate, but using the demonstration certificate is not secure and not recommended for general use.

The following example shows a configuration file for the domains example1.com and example2.com along with the default Destination Control entry:

```
[DEFAULT]

ip_sort_pref = PREFER_V6

max_host_concurrency = 500

max_messages_per_connection = 50

recipient_minutes = 60

recipient_limit = 300

limit_type = host

limit_apply = VG

table_tls = off

bounce_validation = 0

send_tls_req_alert = 0

certificate = example.com



[example1.com]

ip_sort_pref = PREFER_V6

recipient_minutes = 60
```

```
recipient_limit = 100

table_tls = require_verify

limit_apply = VG

bounce_profile = tls_failed

limit_type = host


[example2.com]

table_tls = on

bounce_profile = tls_failed
```

The above example results in the following Destination Control entries for example1.com and example2.com:

```
example1.com

    IP Address Preference: IPv6 Preferred

    Maximum messages per connection: 50

    Rate Limiting:

        500 concurrent connections

        100 recipients per 60 minutes

        Limits applied to entire domain, across all virtual gateways

    TLS: Required (Verify)

    Bounce Profile: tls_failed


example2.com

    IP Address Preference: IPv6 Preferred

    Maximum messages per connection: Default

    Rate Limiting: Default

    TLS: Preferred

    Bounce Profile: tls_failed
```

Use the **Import Table** button on the Destination Controls page or the `destconfig -> import` command to import a configuration file.You can also export your Destination Control entries to an INI file using the **Export Table** button on the Destination Controls page or the `destconfig -> export` command. AsyncOS includes the `[Default]` domain control entry in the exported INI file.

Destination Controls and the CLI

You can use the `destconfig` command in the CLI to configure Destination Control entries. This command is discussed in the *Cisco IronPort AsyncOS CLI Reference Guide*.

Cisco IronPort Bounce Verification

A "bounce" message is a new message that is sent by a receiving MTA, using the Envelope Sender of the original email as the new Envelope Recipient. This bounce is sent back to the Envelope Recipient (usually) with a blank Envelope Sender (MAIL FROM: < >) when the original message is undeliverable (typically due to a non-existent recipient address).

Increasingly, spammers are attacking email infrastructure via misdirected bounce attacks. These attacks consist of a flood of bounce messages, sent by unknowing, legitimate mail servers. Basically, the process spammers use is to send email via open relays and "zombie" networks to multiple, potentially invalid addresses (Envelope Recipients) at various domains. In these messages, the Envelope Sender is forged so that the spam appears to be coming from a legitimate domain (this is known as a "Joe job").

In turn, for each incoming email with an invalid Envelope Recipient, the receiving mail servers generate a new email — a bounce message — and send it along to the Envelope Sender at the innocent domain (the one whose Envelope Sender address was forged). As a result, this target domain receives a flood of "misdirected" bounces — potentially millions of messages. This type of distributed denial of service attack can bring down email infrastructure and render it impossible for the target to send or receive legitimate email.

To combat these misdirected bounce attacks, AsyncOS includes Cisco IronPort Bounce Verification. When enabled, Cisco IronPort Bounce Verification tags the Envelope Sender address for messages sent via your Cisco IronPort appliance. The Envelope Recipient for any bounce message received by the Cisco IronPort appliance is then checked for the presence of this tag. Legitimate bounces (which should contain this tag) are untagged and delivered. Bounce messages that do not contain the tag can be handled separately.

Note that you can use Cisco IronPort Bounce Verification to manage incoming bounce messages based on your outgoing mail. To control how your Cisco IronPort appliance generates outgoing bounces (based on incoming mail), see Directing Bounced Email, page 2-35.

Overview: Tagging and Cisco IronPort Bounce Verification

When sending email with bounce verification enabled, your Cisco IronPort appliance will rewrite the Envelope Sender address in the message. For example, MAIL FROM: `joe@example.com` becomes MAIL FROM: `prvs=joe=123ABCDEFG@example.com`. The `123...` string in the example is the "bounce verification tag" that gets added to the Envelope Sender as it is sent by your Cisco IronPort appliance. The tag is generated using a key defined in the Bounce Verification settings (see Cisco IronPort Bounce Verification Address Tagging Keys, page 2-54 for more information about specifying a key). If this message bounces, the Envelope Recipient address in the bounce will typically include this bounce verification tag.

You can enable or disable bounce verification tagging system-wide as a default. You can also enable or disable bounce verification tagging for specific domains. In most situations, you would enable it by default, and then list specific domains to exclude in the Destination Controls table (see Working with Destination Controls, page 2-46).

If a message already contains a tagged address, AsyncOS does not add another tag (in the case of an Cisco IronPort appliance delivering a bounce message to an Cisco IronPort appliance inside the DMZ).

Handling Incoming Bounce Messages

Bounces that include a valid tag are delivered. The tag is removed and the Envelope Recipient is restored. This occurs immediately after the Domain Map step in the email pipeline. You can define how your Cisco IronPort appliances handle untagged or invalidly tagged bounces — reject them or add a custom header. See Configuring Cisco IronPort Bounce Verification Settings, page 2-56 for more information.

If the bounce verification tag is not present, or if the key used to generate the tag has changed, or if the message is more than seven days old, the message is treated as per the settings defined for Cisco IronPort Bounce Verification.

For example, the following mail log shows a bounced message rejected by the Cisco IronPort appliance:

```
Fri Jul 21 16:02:19 2006 Info: Start MID 26603 ICID 125192

Fri Jul 21 16:02:19 2006 Info: MID 26603 ICID 125192 From: <>

Fri Jul 21 16:02:40 2006 Info: MID 26603 ICID 125192 invalid bounce, rcpt address
<bob@example.com> rejected by bounce verification.

Fri Jul 21 16:03:51 2006 Info: Message aborted MID 26603 Receiving aborted by sender

Fri Jul 21 16:03:51 2006 Info: Message finished MID 26603 aborted
```

Note When delivering non-bounce mail to your own internal mail server (Exchange, etc.), you should disable Cisco IronPort Bounce Verification tagging for that internal domain.

AsyncOS considers bounces as mail with a null Mail From address (<>). For non-bounce messages that might contain a tagged Envelope Recipient, AsyncOS applies a more lenient policy. In such cases, AsyncOS ignores the seven-day key expiration and tries to find a match with older keys as well.

Cisco IronPort Bounce Verification Address Tagging Keys

The tagging key is a text string your Cisco IronPort appliance uses when generating the bounce verification tag. Ideally, you would use the same key across all of your Cisco IronPort appliances so that all mail leaving your domain is tagged consistently. That way, if one Cisco IronPort appliance tags the Envelope Sender on an outgoing message an incoming bounce will be verified and delivered even if the bounce is received by a different Cisco IronPort appliance.

There is a seven day grace period for tags. For example, you may choose to change your tagging key multiple times within a seven-day period. In such a case, your Cisco IronPort appliance will try to verify tagged messages using all previous keys that are less than seven days old.

Cisco IronPort Bounce Verification and the HAT

AsyncOS also includes a HAT setting related to Cisco IronPort Bounce Verification for considering whether untagged bounces are valid. The default setting is "No," which means that untagged bounces are considered invalid and the appliance either rejects the message or applies a customer header, depending on the action selected on the Mail Policies > Bounce Verification page. If you select "Yes," the appliance considers untagged bounces to be valid and accepts them. This may be used in the following scenario:

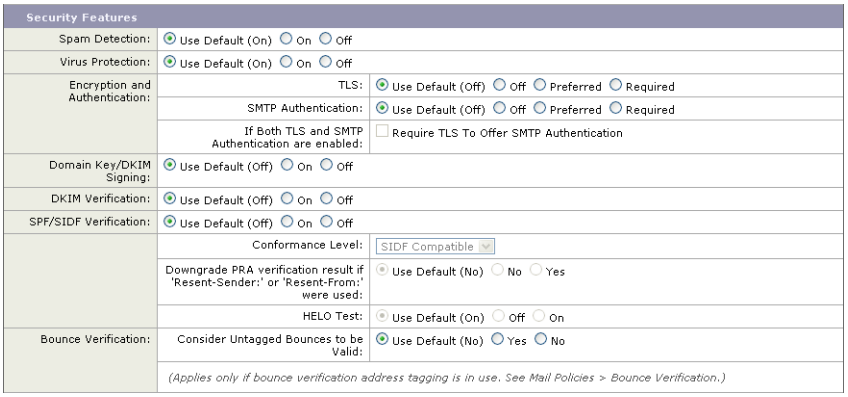
Suppose you have a user that wants to send email to a mailing list. However, the mailing list accepts messages only from a fixed set of Envelope Senders. In such a case, tagged messages from your user will not be accepted (as the tag changes regularly).

To help that user, follow these steps:

Step 1 Add the domain to which the user is trying to send mail to the Destination Controls table and disable tagging for that domain. At this point, the user can send mail without problems.

Step 2 However, to properly support receiving bounces from that domain (since they will not be tagged) you can create a sender group for that domain and enable the Consider Untagged Bounces to be Valid parameter in an "Accept" mail flow policy:

Figure 2-12 The Consider Untagged Bounces to be Valid HAT Parameter



Working with Cisco IronPort Bounce Verification

When configuring Cisco IronPort Bounce Verification, follow these steps:

Step 1 Enter a tagging key (see Configuring Bounce Verification Address Tagging Keys, page 2-56).

Step 2 Edit the bounce verification settings (see Configuring Cisco IronPort Bounce Verification Settings, page 2-56).

Step 3 Enable bounce verification via Destination Controls (see Working with Destination Controls, page 2-46).

Figure 2-13 IronPort Bounce Verification Page



Configuring Bounce Verification Address Tagging Keys

The Bounce Verification Address Tagging Keys listing shows your current key and any unpurged keys you have used in the past. To add a new key:

Step 1 On the Mail Policies > Bounce Verification page, click **New Key.**

Step 2 Enter a text string and click **Submit**.

Step 3 **Commit** your changes.

Purging Keys

You can purge your old address tagging keys by selecting a rule for purging from the pull-down menu and clicking **Purge**.

Configuring Cisco IronPort Bounce Verification Settings

The bounce verification settings determine which action to take when an invalid bounce is received. To configure bounce verification settings:

Step 1 Click **Edit Settings**. The Edit Bounce Verification Settings page is displayed.

Step 2 Select whether to reject invalid bounces, or to add a custom header to the message. If you want to add a header, enter the header name and value.

Step 3 Optionally, enable smart exceptions. This setting allows incoming mail messages, and bounce messages generated by internal mail servers, to be automatically exempted from bounce verification processing (even when a single listener is used for both incoming and outgoing mail).

Step 4 Submit and commit your changes.

Cisco IronPort Bounce Verification and the CLI

You can use the `bvconfig` and `destconfig` commands in the CLI to configure bounce verification. These commands are discussed in the *Cisco IronPort AsyncOS CLI Reference Guide*.

Cisco IronPort Bounce Verification and Cluster Configuration

Bounce verification works in a cluster configuration as long as both Cisco IronPort appliances use the same "bounce key." When you use the same key, either systems should be able to accept a legitimate bounce back. The modified header tag/key is not specific to each Cisco IronPort appliance.

Set Email Delivery Parameters

The `deliveryconfig` command sets parameters to be used when delivering email from the Cisco IronPort appliance.

The Cisco IronPort appliance accepts email using multiple mail protocols: SMTP and QMQP. However, all outgoing email is delivered using SMTP, which is why the `deliveryconfig` command does not require that the protocol be specified.

Note Several of the features or commands described in this section will affect, or be affected by routing precedence. Please see Appendix B, "Assigning Network and IP Addresses" in the *Cisco IronPort AsyncOS for Email Configuration Guide* for more information.

Default Delivery IP Interface

By default, the system uses an IP interface or IP interface group for email delivery. Any currently configured IP interface or IP interface group can be set. If no specific interface is identified, AsyncOS will use the hostname associated with the default delivery interface in the `SMTP HELO` command when communicating with recipient hosts. To configure IP interfaces, use the `interfaceconfig` command.

These are the rules for using Auto selection of email delivery interfaces:

- If the remote email server is on the same subnet as one of the configured interfaces, then traffic will go out on the matching interface.
- When set to auto-select, static routes you have configured using `routeconfig` take effect.
- Otherwise, the interface that is on the same subnet as the default gateway will be used. If all of the IP addresses have an equivalent route to the destination, then the system uses the most efficient interface available.

Possible Delivery Feature

When the Possible Delivery feature is enabled, AsyncOS treats any message that times-out after the body of the message is delivered, but before recipient host acknowledges receipt of the message, as a "possible delivery." This functionality prevents recipients from receiving multiple copies of a message if continuous errors at their recipient host prevent acknowledgement of receipt. AsyncOS logs this recipient as a possible delivery in the mail logs and counts the message as completed. It is recommended that the Possible Delivery feature remains enabled.

Default Maximum Concurrency

You also specify the default maximum number of concurrent connections the appliance makes for outbound message delivery. (The system-wide default is 10,000 connections to separate domains.) The limit is monitored in conjunction with the per-listener maximum outbound message delivery

concurrency (the default per listener is 600 connections for private listeners and 1000 connections for public listeners). Setting the value lower than the default prevents the Cisco IronPort gateway from dominating weaker networks. For example, certain firewalls do not support large numbers of connections, and the Cisco IronPort could induce Denial of Service (DoS) warnings in these environments.

deliveryconfig Example

In the following example, the `deliveryconfig` command is used to set the default interface to "Auto" with "Possible Delivery" enabled. The system-wide maximum outbound message delivery is set to 9000 connections.

```
mail3.example.com> deliveryconfig


Choose the operation you want to perform:

- SETUP - Configure mail delivery.

[]> setup


Choose the default interface to deliver mail.

1. Auto

2. PublicNet2 (192.168.3.1/24: mail4.example.com)

3. Management (192.168.42.42/24: mail3.example.com)

4. PrivateNet (192.168.1.1/24: mail3.example.com)

5. PublicNet (192.168.2.1/24: mail3.example.com)

[1]> 1


Enable "Possible Delivery" (recommended)?  [Y]> y


Please enter the default system wide maximum outbound message delivery

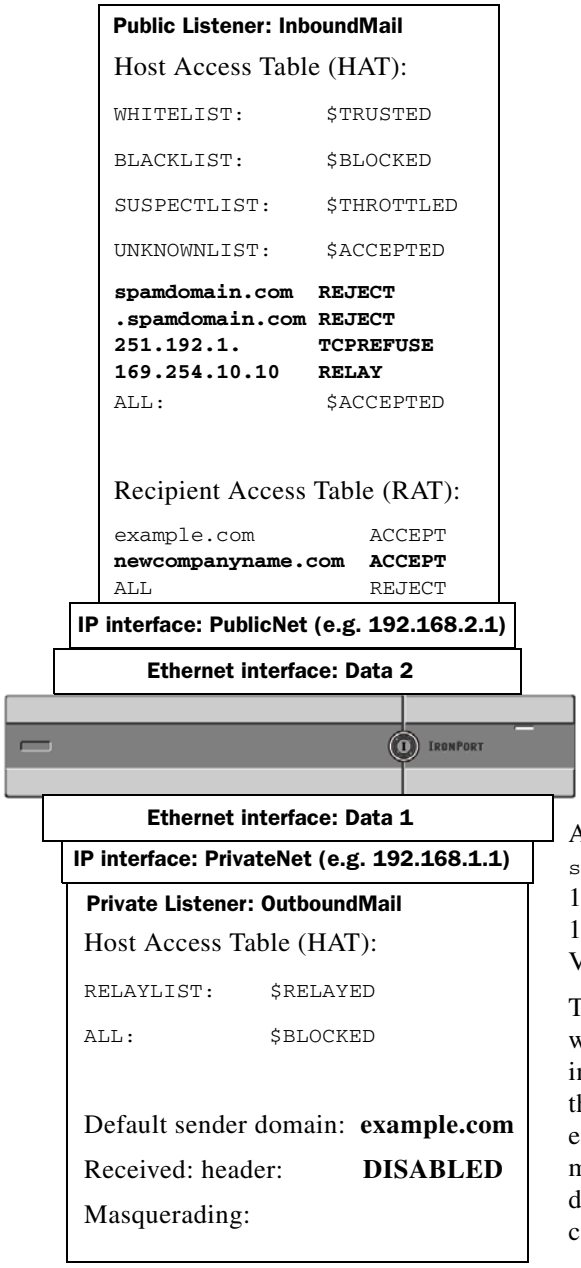
concurrency

[10000]> 9000


mail3.example.com>
```

Our Email Gateway configuration now looks like this:

Figure 2-14 Setting Destination and Delivery Parameters



A `destconfig` entry for the host `small-isp.net` was used to limit 100 simultaneous connections, or 10 simultaneous connections using Virtual Gateway addresses.

The `deliveryconfig` command was used to use Auto-selection of interfaces for email delivery and the Possible Delivery feature was enabled. The system-wide maximum outbound message delivery was set to 9000 total concurrent connections.

Using Virtual Gateway™ Technology

This section describes Cisco IronPort Virtual Gateway™ technology and its benefits, how to set up a Virtual Gateway address, and how to monitor and manage Virtual Gateway addresses.

The Cisco IronPort Virtual Gateway technology allows you to configure enterprise mail gateways for all domains you host — with distinct IP addresses, hostname and domains — and create separate corporate email policy enforcement and anti-spam strategies for those domains, while hosted within the same physical appliance.

Note The number of Virtual Gateway addresses available to you depends on the model of your Cisco IronPort appliance. Some appliance models can be upgraded to support more Virtual Gateway addressed via a feature key. Contact your Cisco IronPort sales representative for more information about upgrading the number of Virtual Gateway addresses on your appliance.

Overview

Cisco has developed a unique Virtual Gateway technology designed to help ensure that corporations can reliably communicate with their customers via email. Virtual Gateway technology enables users to separate the Cisco IronPort appliance into multiple Virtual Gateway addresses from which to send and receive email. Each Virtual Gateway address is given a distinct IP address, hostname and domain, and email queue.

Assigning a distinct IP address and hostname to each Virtual Gateway address ensures that email delivered through the gateway will be properly identified by the recipient host and prevents critical email from being blocked as spam. The Cisco IronPort appliance has the intelligence to give the correct hostname in the `SMTP HELO` command for each of the Virtual Gateway addresses. This ensures that if a receiving Internet Service Provider (ISP) performs a reverse DNS look-up, the Cisco IronPort appliance will match the IP address of the email sent through that Virtual Gateway address. This feature is extremely valuable, because many ISPs use a reverse DNS lookup to detect unsolicited email. If the IP address in the reverse DNS look-up does not match the IP address of the sending host, the ISP may assume the sender is illegitimate and will frequently discard the email. The Cisco IronPort Virtual Gateway technology ensures that reverse DNS look-ups will always match the sending IP address, preventing messages from being blocked accidentally.

Messages in each Virtual Gateway address are also assigned to a separate message queue. If a certain recipient host is blocking email from one Virtual Gateway address, messages intended for that host will remain in the queue and eventually timeout. But messages intended for the same domain in a different Virtual Gateway queue that is not being blocked will be delivered normally. While these queues are treated separately for delivery purposes, the system administration, logging and reporting capability still provide a holistic view into all Virtual Gateway queues as if they were one.

Setting Up Virtual Gateway Addresses

Before setting up the Cisco IronPort Virtual Gateway addresses, you must allocate a set of IP addresses that will be used to send email from. (For more information, see “Assigning Network and IP Addresses” in the *Cisco IronPort AsyncOS for Email Configuration Guide*.) You should also ensure proper configuration of your DNS servers so that the IP address resolves to a valid hostname. Proper configuration of DNS servers ensures that if the recipient host performs a reverse DNS lookup, it will resolve to valid IP/hostname pairs.

Creating New IP Interfaces for Use with Virtual Gateways

After the IP addresses and hostnames have been established, the first step in configuring the Virtual Gateway addresses is to create new IP interfaces with the IP/hostname pairs using the Network > IP Interfaces page in the GUI or the `interfaceconfig` command in the CLI.

Once the IP interfaces have been configured, you have the option to combine multiple IP interfaces into interface groups; these groups can then be assigned to specific Virtual Gateways addresses which the system cycles through in a “round robin” fashion when delivering email.

After creating the required IP interfaces, you have two options for setting up the Virtual Gateway addresses and defining which email campaign will be sent from each IP interface or interface group:

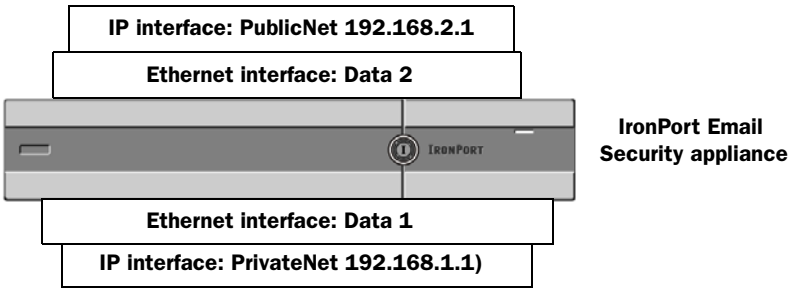
Step 1 You can use the `altsrchost` command to map email from specific sender IP addresses or Envelope Sender address information to a host IP interface (Virtual Gateway address) or interface group for delivery.

Step 2 Using message filters, you can set up specific filters to deliver flagged messages using a specific host IP interface (Virtual Gateway address) or interface group. See Alter Source Host (Virtual Gateway address) Action, page 6-58. (This method is more flexible and powerful than the one above.)

For more information about creating IP interfaces, see the "Accessing the Appliance" appendix in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

So far, we have been using an Email Gateway configuration with the following interfaces defined as shown in Figure 2-15.

Figure 2-15 Example Public and Private Interfaces



In the following example, the IP Interfaces page confirms that these two interfaces (PrivateNet and PublicNet) have been configured, in addition to the Management interface.

Figure 2-16 IP Interfaces Page

IP Interfaces

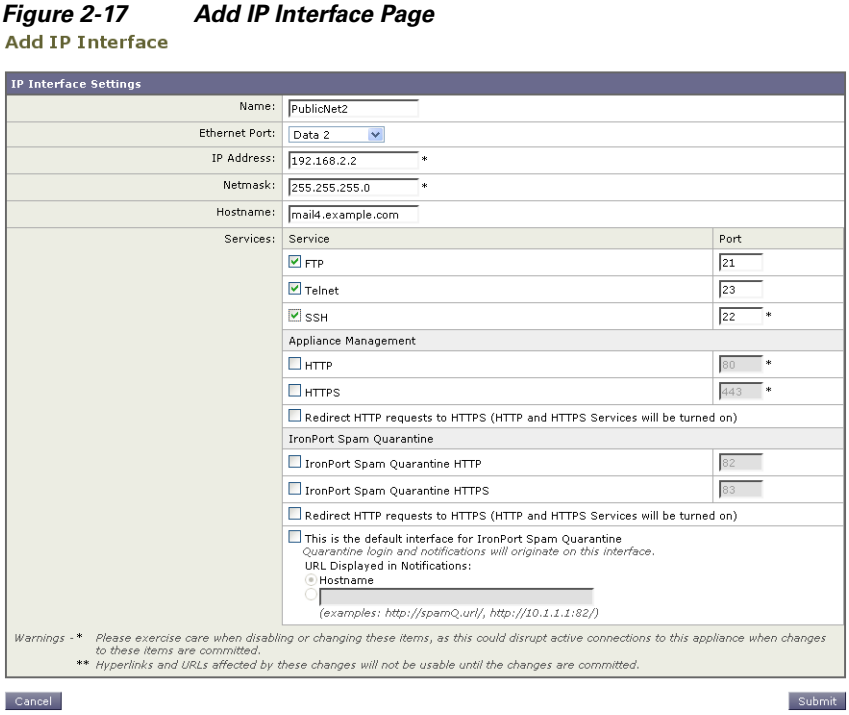
Network Interfaces and IP Addresses

Add IP Interface...

Name	IP Address	Hostname	Delete
Management	192.168.42.42/24	mail3.example.com	
PrivateNet	192.168.1.1/24	mail3.example.com	
PublicNet	192.168.2.1/24	mail3.example.com	

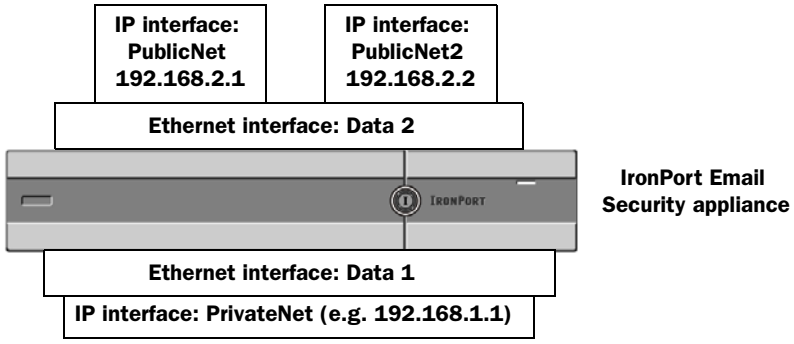
Next, the Add IP Interface page is used to create a new interface named PublicNet2 on the Data2 Ethernet interface. The IP address of 192.168.2.2 is used, and the hostname of `mail4.example.com` is specified. The services for FTP (port 21), Telnet (port 23), and SSH (port 22) are then enabled.

Figure 2-17 Add IP Interface Page



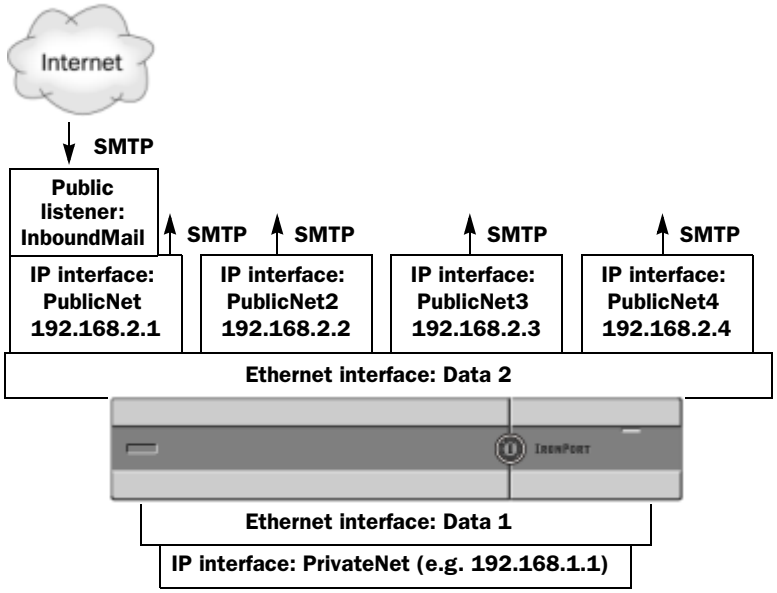
Our Email Gateway configuration now looks like this:

Figure 2-18 Adding Another Public Interface



Using Virtual Gateway addresses, a configuration like the one shown in Figure 2-19 is also possible.

Figure 2-19 Four Virtual Gateway Addresses on One Ethernet Interface



Note that four separate IP interfaces can be used to deliver mail, where only one public listener is configured to accept messages from the Internet.

Mapping Messages to IP Interfaces for Delivery

The `altsrchost` command provides the simplest and most straightforward method to segment each Cisco IronPort appliance into multiple IP interfaces (Virtual Gateway addresses) from which to deliver email. However, users requiring more power and flexibility in mapping messages to particular Virtual Gateways should investigate the use of message filters. See Chapter 6, "Using Message Filters to Enforce Email Policies" for more information.

The `altsrchost` command allows you to control which IP interface or interface group to use during email delivery based on one of the following:

- the sender's IP address
- the Envelope Sender address

To specify which IP interface or interface group the system will deliver email from, you create mapping keys that pair either the sender's IP address or the Envelope Sender address to an IP interface or interface group (specified by interface name or group name).

Cisco IronPort AsyncOS will compare both the IP address and Envelope Sender address to the mapping keys. If either the IP address or Envelope Sender address matches one of the keys, the corresponding IP interface is used for the outbound delivery. If there is no match, the default outbound interface will be used.

The system can match any of the following keys and take preference in the following order:

Sender's IP address	The IP address of the sender must match exactly. Example: `192.168.1.5`
Fully-formed Envelope Sender	The Envelope Sender must match the entire address exactly. Example: `username@example.com`

Username	The system will match username syntax against the Envelope Sender address up to the `@` sign. The `@` sign must be included. Example: `username@`
Domain	The system will match domain name syntax against the Envelope Sender address starting with the `@` sign. The `@` sign must be included. Example: `@example.com`

Note A listener checks the information in the `altsrchost` table and directs the email to a particular interface *after* checking the masquerading information and *before* message filters are checked.

Use these subcommands within the `altsrchost` command to create mappings in the Virtual Gateways via the CLI:

Syntax	Description
`new`	Create a new mapping manually.
`print`	Display the current list of mappings.
`delete`	Remove one of the mappings from the table.

Importing an altsrchost File

Like the HAT, the RAT, `smtproutes`, and masquerading and alias tables, you can modify `altsrchost` entries by exporting and importing a file. Follow these steps:

Step 1 Use the `export` subcommand of the `altsrchost` command to export the existing entries to a file (whose name you specify).

Step 2 Outside of the CLI, get the file. (See Appendix B, "Accessing the Appliance" for more information.)

Step 3 With a text editor, create new entries in the file. The order that rules appear in the `altsrchost` table is important.

Step 4 Save the file and place it in the "altsrchost" directory for the interface so that it can be imported. (See Appendix B, "Accessing the Appliance" for more information.)

Step 5 Use the `import` subcommand of `altsrchost` to import the edited file.

altsrchost Limits

You can define up to 1,000 `altsrchost` entries.

Example Text File with Valid Mappings for the altsrchost Command

```
# Comments to describe the file

@example.com DemoInterface

paul@ PublicInterface

joe@ PublicInterface

192.168.1.5, DemoInterface

steve@example.com PublicNet
```

The `import` and `export` subcommands operate on a line-by-line basis and map either the sender IP address or the Envelope Sender address line to the interface name. The key must be the first block of non-space characters followed by the interface name in the second block of non-space characters, separated by a comma (`,`) or space ( ). Comment lines start with a number sign (`#`) and will be ignored.

Adding an altsrchost Mapping through the CLI

In the following example, the `altsrchost` table is printed to show that there are no existing mappings. Two entries are then created:

- Mail from the groupware server host named `@exchange.example.com` is mapped to the PublicNet interface.
- Mail from the sender IP address of 192.168.35.35 (for example, the marketing campaign messaging system) is mapped to the PublicNe2t interface.

Finally, the `altsrchost` mappings are printed to confirm and the changes are committed.

```
mail3.example.com> altsrchost



There are currently no mappings configured.



Choose the operation you want to perform:

- NEW - Create a new mapping.

- IMPORT - Load new mappings from a file.

[]> new



Enter the Envelope From address or client IP address for which you want to set up a
Virtual Gateway mapping.  Partial addresses such as "@example.com" or "user@" are
allowed.

[]> @exchange.example.com
```

```
Which interface do you want to send messages for @exchange.example.com from?

1. PublicNet2 (192.168.2.2/24: mail4.example.com)

2. Management (192.168.42.42/24: mail3.example.com)

3. PrivateNet (192.168.1.1/24: mail3.example.com)

4. PublicNet (192.168.2.1/24: mail4.example.com)

[1]> 4


Mapping for @exchange.example.com on interface PublicNet created.


Choose the operation you want to perform:

- NEW - Create a new mapping.

- EDIT - Modify a mapping.

- DELETE - Remove a mapping.

- IMPORT - Load new mappings from a file.

- EXPORT - Export all mappings to a file.

- PRINT - Display all mappings.

- CLEAR - Remove all mappings.

[]> new


Enter the Envelope From address or client IP address for which you want to set up a
Virtual Gateway mapping.  Partial addresses such as "@example.com" or "user@" are
allowed.

[]> 192.168.35.35


Which interface do you want to send messages for 192.168.35.35 from?

1. PublicNet2 (192.168.2.2/24: mail4.example.com)

2. Management (192.168.42.42/24: mail3.example.com)

3. PrivateNet (192.168.1.1/24: mail3.example.com)

4. PublicNet (192.168.2.1/24: mail4.example.com)
```

```
[1]> 1

Mapping for 192.168.35.35 on interface PublicNet2 created.

Choose the operation you want to perform:
- NEW - Create a new mapping.
- EDIT - Modify a mapping.
- DELETE - Remove a mapping.
- IMPORT - Load new mappings from a file.
- EXPORT - Export all mappings to a file.
- PRINT - Display all mappings.
- CLEAR - Remove all mappings.
[]> print

1. 192.168.35.35 -> PublicNet2
2. @exchange.example.com -> PublicNet

Choose the operation you want to perform:
- NEW - Create a new mapping.
- EDIT - Modify a mapping.
- DELETE - Remove a mapping.
- IMPORT - Load new mappings from a file.
- EXPORT - Export all mappings to a file.
- PRINT - Display all mappings.
- CLEAR - Remove all mappings.
[]>

mail3.example.com> commit
```

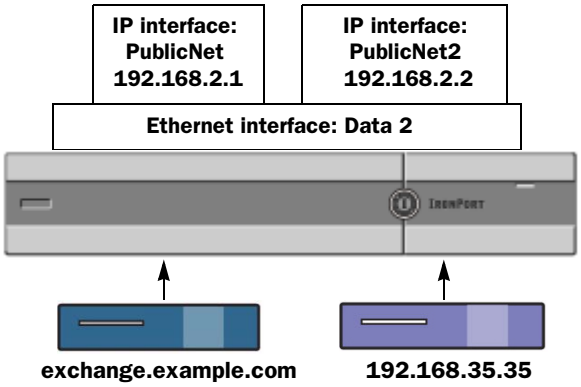
```
Please enter some comments describing your changes:

[]> Added 2 altsrchost mappings

Changes committed: Thu Mar 27 14:57:56 2003
```

An illustration of the configuration change in this example is shown in Figure 2-20:

Figure 2-20 Example: Selecting an IP Interface or Interface Group to Use



The `altsrchost` table was modified to create these mappings. Messages from `@exchange.example.com` use the interface PublicNet, and messages from 192.168.35.35 use the interface PublicNet2.

Monitoring the Virtual Gateway Addresses

While each Virtual Gateway address has its own email queue for delivery purposes, the system administration, logging, and reporting capabilities still provide a holistic view into all Virtual Gateway queues as if they were one. To monitor the recipient host status for each Virtual Gateway queue, use the `hoststatus` and `hostrate` command. See "Reading the Available Components of Monitoring" in the *Cisco IronPort AsyncOS for Email Daily Management Guide*.

The `hoststatus` command returns monitoring information about email operations relating to a specific recipient host.

If you are using Virtual Gateway technology, information about each Virtual Gateway address is also displayed. The command requires you to input the domain of the host information to be returned. DNS information stored in the AsyncOS cache and the last error returned from the recipient host is also given. Data returned is cumulative since the last `resetcounters` command.

The statistics returned are grouped into two categories: counters and gauges. In addition, other data returned include: last activity, MX records, and last 5XX error.

Managing Delivery Connections per Virtual Gateway Address

Certain system parameters require settings at the system and Virtual Gateway address levels.

For example, some recipient ISPs limit the number of connections they allow for each client host. Therefore, it is important to manage relationships with the ISPs, especially when email is being delivered over multiple Virtual Gateway addresses.

See Controlling Email Delivery, page 2-44 for information about the `destconfig` command and how Virtual Gateway addresses are affected.

When you create a "group," of Virtual Gateway addresses, the good neighbor table settings for Virtual Gateway are applied to the group, even if the group consists of 254 IP addresses.

For example, suppose you have created group of 254 outbound IP addresses set up as a group to cycle through in a "round-robin" fashion, and suppose the good neighbor table for `small-isp.com` is 100 simultaneous connections for the system and 10 connections for Virtual Gateway addresses. This configuration will *never* open more than 10 connections total for all 254 IP addresses in that group; the group is treated as a single Virtual Gateway address.

Using Global Unsubscribe

To ensure that specific recipients, recipient domains, or IP addresses never receive messages from the Cisco IronPort appliance, use the Cisco IronPort AsyncOS Global Unsubscribe feature. The `unsubscribe` command allows you to add and delete addresses to a global unsubscribe list, as well as enable and disable the feature. AsyncOS checks all recipient addresses against a list of "globally unsubscribed" users, domains, email addresses, and IP addresses. If a recipient matches an address in the list, the recipient is either dropped or hard bounced, and the Global Unsubscribe (GUS) counter is incremented. (Log files will note whether a matching recipient was dropped or hard bounced.) The GUS check occurs immediately before an attempt to send email to a recipient, thus inspecting all messages sent by the system.



Note Global Unsubscribe is not intended to replace the removal of names and general maintenance of mailing lists. The feature is intended to act as a fail-safe mechanism to ensure email does not get delivered to inappropriate entities.

The global unsubscribe feature applies to private and public listeners.

Global Unsubscribe has a maximum limit of 10,000 addresses. To increase this limit, contact your Cisco IronPort sales representative. Global Unsubscribe addresses can be in one of four forms:

Table 2-10 Global Unsubscribe Syntax

`username@example.com`	Fully-formed email address This syntax is used to block a specific recipient at a specific domain.
`username@`	Username The username syntax will block all recipients with the specified username at all domains. The syntax is the username followed by an at sign (`@`).
`@example.com`	Domain The domain syntax is used to block all recipients destined for a particular domain. The syntax is the specific domain, preceded by an at sign (`@`).

Table 2-10 Global Unsubscribe Syntax (Continued)

@.example.com	Partial Domain The partial domain syntax is used to block all recipients destined for a particular domain and all its subdomains.
10.1.28.12	IP address The IP address syntax is used to block all recipients destined for a particular IP address. This syntax can be useful if a single IP address is hosting multiple domains. The syntax consists of a common dotted octet IP address.

Adding a Global Unsubscribe Address Using The CLI

In this example, the address `user@example.net` is added to the Global Unsubscribe list, and the feature is configured to hard bounce messages. Messages sent to this address will be bounced; the appliance will bounce the message immediately prior to delivery.

```
mail3.example.com> unsubscribe


Global Unsubscribe is enabled. Action: drop.


Choose the operation you want to perform:

- NEW - Create a new entry.

- IMPORT - Import entries from a file.

- SETUP - Configure general settings.

[]> new


Enter the unsubscribe key to add.  Partial addresses such as

"@example.com" or "user@" are allowed, as are IP addresses. Partial hostnames such as
"@.example.com" are allowed.

[]> user@example.net


Email Address 'user@example.net' added.

Global Unsubscribe is enabled.


Choose the operation you want to perform:
```

```
- NEW - Create a new entry.
- DELETE - Remove an entry.
- PRINT - Display all entries.
- IMPORT - Import entries from a file.
- EXPORT - Export all entries to a file.
- SETUP - Configure general settings.
- CLEAR - Remove all entries.
[]> setup

Do you want to enable the Global Unsubscribe feature? [Y]> y

Would you like matching messages to be dropped or bounced?
1. Drop
2. Bounce
[1]> 2

Global Unsubscribe is enabled. Action: bounce.

Choose the operation you want to perform:
- NEW - Create a new entry.
- DELETE - Remove an entry.
- PRINT - Display all entries.
- IMPORT - Import entries from a file.
- EXPORT - Export all entries to a file.
- SETUP - Configure general settings.
- CLEAR - Remove all entries.
[]>

mail3.example.com> commit
```

```
Please enter some comments describing your changes:

[]> Added username "user@example.net" to global unsubscribe


Changes committed: Thu Mar 27 14:57:56 2003
```

Exporting and Importing a Global Unsubscribe File

Like the HAT, the RAT, `smtproutes`, static masquerading tables, alias tables, domain map tables, and `altsrchost` entries, you can modify global unsubscribe entries by exporting and importing a file. Follow these steps:

Step 1 Use the `export` subcommand of the `unsubscribe` command to export the existing entries to a file (whose name you specify).

Step 2 Outside of the CLI, get the file. (See Appendix B, "Accessing the Appliance" for more information.)

Step 3 With a text editor, create new entries in the file.

Separate entries in the file by new lines. Return representations from all standard operating systems are acceptable (<CR>, <LF>, or <CR><LF>). Comment lines start with a number sign (`#`) and are ignored. For example, the following file excludes a single recipient email address (`test@example.com`), all recipients at a particular domain (`@testdomain.com`), all users with the same name at multiple domains (`testuser@`), and any recipients at a specific IP address (`11.12.13.14`).

```
# this is an example of the global_unsubscribe.txt file

test@example.com

@testdomain.com

testuser@

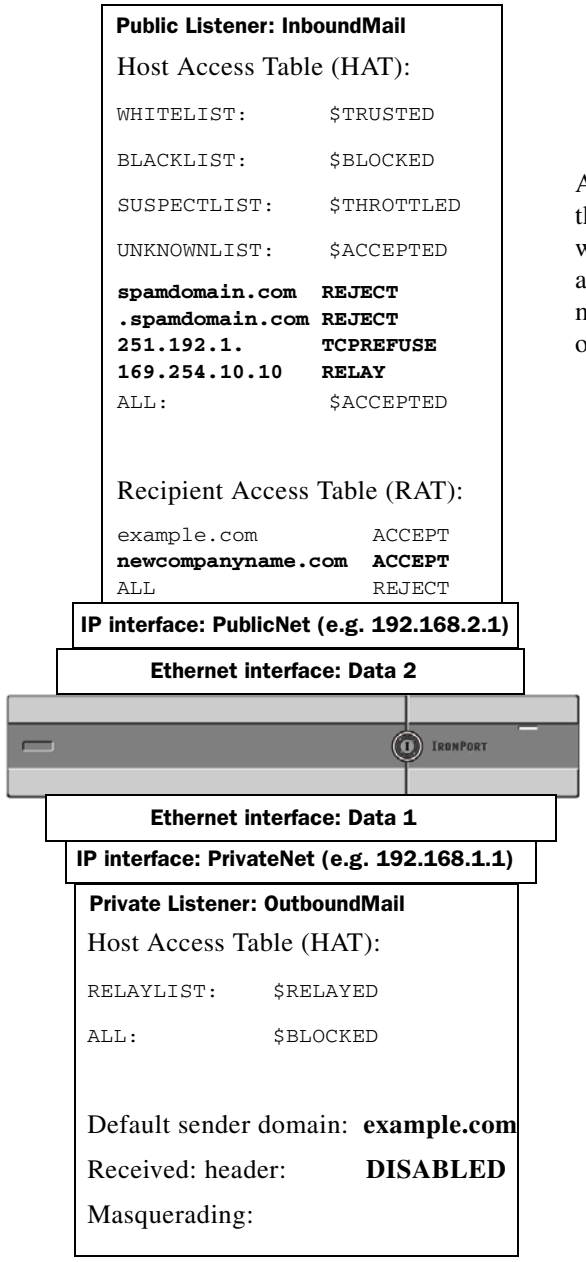
11.12.13.14
```

Step 4 Save the file and place it in the configuration directory for the interface so that it can be imported. (See Appendix B, "Accessing the Appliance" for more information.)

Step 5 Use the `import` subcommand of `unsubscribe` to import the edited file.

Our Email Gateway configuration now looks like this:

Figure 2-21 Global Unsubscribe Example



A global unsubscribe entry for the address *user@example.net* was created. Messages to this address will be bounced from mail accepted by either private or public listeners.

Review: Email Pipeline

Table 2-11 and Table 2-12 provide an overview of how email is routed through the system, from reception to routing to deliver. Each feature is processed in order (from top to bottom) and is briefly summarized. Shaded areas in Figure 2-21 represent processing that occurs in the Work Queue.

You can test most of the configurations of features in this pipeline using the `trace` command. For more information, see "Debugging Mail Flow Using Test Messages: Trace" in the *Cisco IronPort AsyncOS for Email Daily Management Guide*.

Note For outgoing mail, RSA Email Data Loss Prevention scanning takes place after the Outbreak Filters stage.

Table 2-11 Email Pipeline for the Cisco IronPort Appliance: Receiving Email Features

Feature	Description
Host Access Table (HAT) **Host DNS Sender Verification** **Sender Groups** **Envelope Sender Verification** **Sender Verification Exception Table** **Mail Flow Policies**	ACCEPT, REJECT, RELAY, or TCPREFUSE connections Maximum outbound connections Maximum concurrent inbound connections per IP address Maximum message size and messages per connection Maximum recipients per message and per hour TCP listen queue size TLS: no/preferred/required SMTP AUTH: no/preferred/required Drop email with malformed FROM headers Always accept or reject mail from entries in the Sender Verification Exception Table. SenderBase on/off (IP profiling/flow control)
Received Header	Adds a received header to accepted email: on/off.
Default Domain	Adds default domain for “bare” user addresses.
Bounce Verification	Used to verify incoming bounce messages as legitimate.
Domain Map	Rewrites the Envelope Recipient for each recipient in a message that matches a domain in the domain map table.
Recipient Access Table (RAT)	(Public listeners only) ACCEPT or REJECT recipients in `RCPT TO` plus Custom SMTP Response. Allow special recipients to bypass throttling.
Alias tables	Rewrites the Envelope Recipient. (Configured system-wide. `aliasconfig` is not a subcommand of `listenerconfig`.)
LDAP Recipient Acceptance	LDAP validation for recipient acceptance occurs within the SMTP conversation. If the recipient is not found in the LDAP directory, the message is dropped or bounced. LDAP validation can be configured to occur within the work queue instead.

Table 2-12 Email Pipeline for the Cisco IronPort Appliance: Routing and Delivery Features

Work Queue	**LDAP Recipient Acceptance**		LDAP validation for recipient acceptance occurs within the work queue. If the recipient is not found in the LDAP directory, the message is dropped or bounced. LDAP validation can be configured to occur within the SMTP conversation instead.
	Masquerading **or LDAP Masquerading**		Masquerading occurs in the work queue; it rewrites the Envelope Sender, To:, From:, and/or CC: headers, from a static table or via an LDAP query.
	LDAP Routing		LDAP queries are performed for message routing or address rewriting. Group LDAP queries work in conjunction with message filter rules `mail-from-group` and `rcpt-to-group`.
	Message Filters*		Message Filters are applied prior to message "splintering." * Can send messages to quarantines.
	Anti-Spam**	Per Recipient Scanning	Anti-spam scanning engine examines messages and returns a verdict for further processing.
	Anti-Virus*		Anti-Virus scanning examines messages for viruses. Messages are scanned and optionally repaired, if possible. * Can send messages to quarantines.
	Content Filters*		Content Filters are applied. * Can send messages to quarantines.
	Outbreak Filters*		The Outbreak Filters feature helps protect against virus outbreaks. * Can send messages to quarantines.
	Virtual gateways		Sends mail over particular IP interfaces or groups of IP interfaces.
	Delivery limits		1. Sets the default delivery interface. 2. Sets the total maximum number of outbound connections.
	Domain-based Limits		Defines, per-domain: maximum outbound connections for each virtual gateway and for the entire system; the bounce profile to use; the TLS preference for delivery: no/preferred/required
	Domain-based routing		Routes mail based on domain without rewriting Envelope Recipient.

Table 2-12 Email Pipeline for the Cisco IronPort Appliance: Routing and Delivery Features (Continued)

Global unsubscribe	Drops recipients according to specific list (configured system-wide).
Bounce profiles	Undeliverable message handling. Configurable per listener, per Destination Controls entry, and via message filters.

* These features can send messages to special queues called Quarantines.



CHAPTER 3

LDAP Queries

If you store user information within LDAP directories in your network infrastructure — for example, in Microsoft Active Directory, SunONE Directory Server, or OpenLDAP directories — you can configure the Cisco IronPort appliance to query your LDAP servers to accept, route, and authenticate messages. You can configure the Cisco IronPort appliance to work with one or multiple LDAP servers.

This chapter covers the following topics:

- Overview, page 3-1
- Creating LDAP Server Profiles, page 3-4
- Working with LDAP Queries, page 3-12
- Acceptance (Recipient Validation) Queries, page 3-18
- Routing: Alias Expansion, page 3-20
- Masquerading, page 3-20
- Group LDAP Queries, page 3-22
- Domain-based Queries, page 3-26
- Chain Queries, page 3-27
- Using LDAP For Directory Harvest Attack Prevention, page 3-28
- Configuring AsyncOS for SMTP Authentication, page 3-31
- Configuring External Authentication for Users, page 3-40
- Spam Quarantine End-User Authentication Queries, page 3-43
- Spam Quarantine Alias Consolidation Queries, page 3-44
- User Distinguished Name Queries, page 3-45
- Configuring AsyncOS To Work With Multiple LDAP Servers, page 3-46

Overview

The following section provides an overview on the types of LDAP queries you can perform; how LDAP works with the Cisco IronPort appliance to authenticate, accept, and route messages; and how to configure your Cisco IronPort appliance to work with LDAP.

Understanding LDAP Queries

If you store user information within LDAP directories in your network infrastructure, you can configure the Cisco IronPort appliance to query your LDAP server for the following purposes:

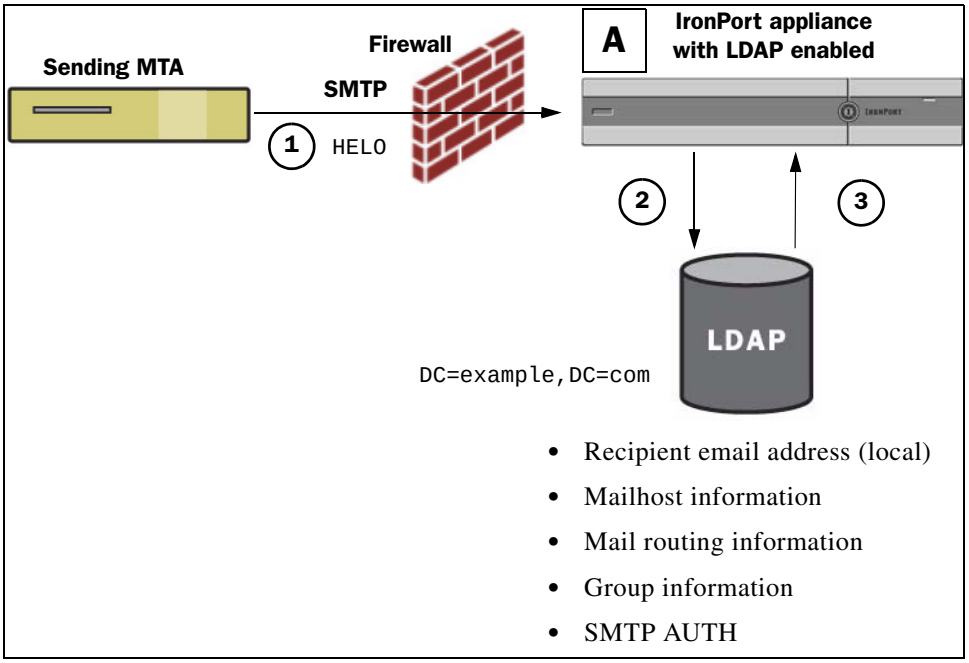
- **Acceptance Queries**. You can use your existing LDAP infrastructure to define how the recipient email address of incoming messages (on a public listener) should be handled. For more information, see Acceptance (Recipient Validation) Queries, page 3-18.
- **Routing (Aliasing)** You can configure the appliance to route messages to the appropriate address and/or mail host based upon the information available in LDAP directories on your network. For more information, see Routing: Alias Expansion, page 3-20.
- **Masquerading**. You can masquerade Envelope Senders (for outgoing mail) and message headers (for incoming mail, such as To:, Reply To:, From: or CC:). For more information about masquerading, see Masquerading, page 3-20.
- **Group Queries**. You can configure the Cisco IronPort appliance to perform actions on messages based on the groups in the LDAP directory. You do this by associating a group query with a message filter. You can perform any message action available for message filters on messages that match the defined LDAP group. For more information, see Group LDAP Queries, page 3-22.
- **Domain-based Queries**. You can create domain-based queries to allow the Cisco IronPort appliance to perform different queries for different domains on a single listener. When the Email Security Appliance runs the domain-based queries, it determines the query to use based on the domain, and it queries the LDAP server associated with that domain.
- **Chain Queries.** You can create a chain query to enable the Cisco IronPort appliance to perform a series of queries in sequence. When you configure a chain query, the Cisco IronPort appliance runs each query in sequence until the LDAP appliance returns a positive result.
- **Directory Harvest Prevention**. You can configure the Cisco IronPort appliance to combat directory harvest attacks using your LDAP directories. You can configure directory harvest prevention during the SMTP conversation or within the work queue. If the recipient is not found in the LDAP directory, you can configure the system to perform a delayed bounce or drop the message entirely. Consequently, spammers are not able to differentiate between valid and invalid email addresses. See Using LDAP For Directory Harvest Attack Prevention, page 3-28.
- **SMTP Authentication**. AsyncOS provides support for SMTP authentication. SMTP Auth is a mechanism for authenticating clients connected to an SMTP server. You can use this functionality to enable users at your organization to send mail using your mail servers even if they are connecting remotely (e.g. from home or while traveling). For more information, see Configuring AsyncOS for SMTP Authentication, page 3-31.
- **External Authentication**. You can configure your Cisco IronPort appliance to use your LDAP directory to authenticate users logging in to the Cisco IronPort appliance. For more information, see Configuring External Authentication for Users, page 3-40.
- **Spam Quarantine End-User Authentication**. You can configure your appliance to validate users when they log in to the end-user quarantine. For more information, see Spam Quarantine End-User Authentication Queries, page 3-43.
- **Spam Quarantine Alias Consolidation**. If you use email notifications for spam, this query consolidates the end-user aliases so that end-users do not receive quarantine notices for each aliased email address. For more information, see Spam Quarantine Alias Consolidation Queries, page 3-44.
- **User Distinguished Name.** If you use RSA Enterprise Manager for data loss prevention (DLP), this query retrieves the distinguished name for senders of messages that may contain DLP violations. The Email Security appliance includes the distinguished name when it sends DLP incident data to Enterprise Manager. For more information, see User Distinguished Name Queries, page 3-45.

Understanding How LDAP Works with AsyncOS

When you work with LDAP directories, the Cisco IronPort appliance can be used in conjunction with an LDAP directory server to accept recipients, route messages, and/or masquerade headers. LDAP group queries can also be used in conjunction with message filters to create rules for handling messages as they are received by the Cisco IronPort appliance.

Figure 3-1 demonstrates how the Cisco IronPort appliance works with LDAP:

Figure 3-1 LDAP Configuration



Step 1 The sending MTA sends a message to the public listener "A" via SMTP.

Step 2 The Cisco IronPort appliance queries the LDAP server defined via the System Administration > LDAP page (or by the global `ldapconfig` command).

Step 3 Data is received from the LDAP directory, and, depending on the queries defined on the System Administration > LDAP page (or in the `ldapconfig` command) that are used by the listener:

- the message is routed to the new recipient address, or dropped or bounced
- the message is routed to the appropriate mailhost for the new recipient
- From:, To:, and CC: message headers are re-written based upon the query
- further actions as defined by `rcpt-to-group` or `mail-from-group` message filter rules (used in conjunction with configured group queries).

Note You can configure your Cisco IronPort appliance to connect to multiple LDAP servers. When you do this, you can configure the LDAP profile settings for load-balancing or failover. For more information about working with multiple LDAP servers, see Configuring AsyncOS To Work With Multiple LDAP Servers, page 3-46.

Configuring AsyncOS to work with LDAP

When you configure your Cisco IronPort appliance to work with an LDAP directory, you must complete the following steps to configure your AsyncOS appliance for acceptance, routing, aliasing, and masquerading:

Step 1 **Configure LDAP server profiles**. The server profile contains information to enable AsyncOS to connect to the LDAP server (or servers), such as:

- the name of the server (s) and port to send queries,
- the base DN, and
- the authentication requirements for binding to the server

For more information about configuring a server profile, see Creating LDAP Server Profiles, page 3-4.

When you configure the LDAP server profile, you can configure AsyncOS to connect to one or multiple LDAP servers.

For information about configuring AsyncOS to connect to multiple servers, see Configuring AsyncOS To Work With Multiple LDAP Servers, page 3-46.

Step 2 **Configure the LDAP query**. You configure the LDAP queries on the LDAP server profile. The query you configure should be tailored to your particular LDAP implementation and schema.

For information on the types of LDAP queries you can create, see Understanding LDAP Queries, page 3-2.

For information on writing queries, see Working with LDAP Queries, page 3-12.

Step 3 **Enable the LDAP server profile on a public listener or on a private listener**. You must enable the LDAP server profile on a listener to instruct the listener to run the LDAP query when accepting, routing, or sending a message.

For more information, see Working with LDAP, LDAP Queries, and Listeners, page 3-7.



Note When you configure a group query, you need to take additional steps to configure AsyncOS to work with the LDAP server. For information on configuring a group query, see Group LDAP Queries, page 3-22. When you configure an end-user authentication or spam notification consolidation query, you must enable LDAP end-user access to the Cisco IronPort Spam Quarantine. For more information on the Cisco IronPort Spam Quarantine, see "Configuring the Cisco IronPort Spam Quarantines Feature" in the *Cisco IronPort AsyncOS for Email Daily Management Guide*.

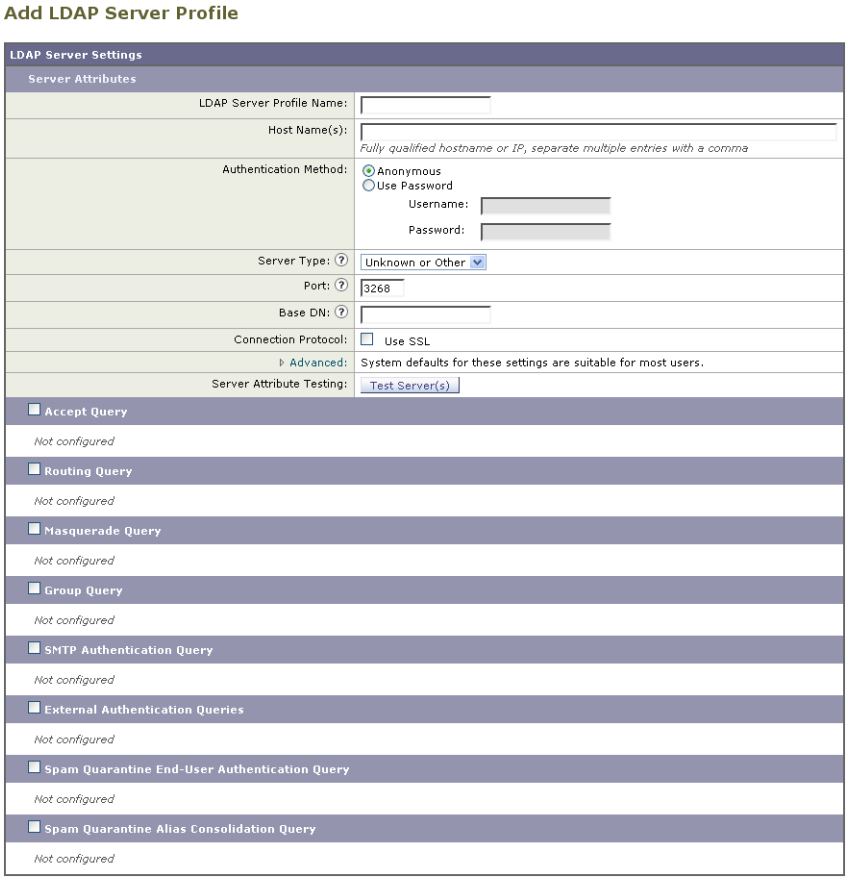
Creating LDAP Server Profiles

When you configure AsyncOS to use LDAP directories, you create an LDAP server profile to store the information about the LDAP server.

To create an LDAP server profile,

Step 1 On the System Administration > LDAP page, click **Add LDAP Server Profile** The Add LDAP Server Profile page is displayed:

Figure 3-2 Configuring an LDAP Server Profile



Step 2 Enter a name for the server profile.

Step 3 Enter the host name for the LDAP server.

You can enter multiple host names to configure the LDAP servers for failover or load-balancing. Separate multiple entries with commas. For more information, see Configuring AsyncOS To Work With Multiple LDAP Servers, page 3-46.

Step 4 Select an authentication method. You can use anonymous authentication or specify a username and password.

Step 5 Select the LDAP server type: Active Directory, OpenLDAP, or Unknown or Other.

Step 6 Enter a port number.

The default port is 3268. This is the default port for Active Directory that enables it to access the global catalog in a multi-server environment.

Step 7 Enter a Base DN (distinguishing name) for the LDAP server.

If you authenticate with a username and a password, the username must include the full DN to the entry that contains the password. For example, a user is a member of the marketing group with an email address of joe@example.com. The entry for this user would look like the following entry:

```
uid=joe, ou=marketing, dc=example dc=com
```

Step 8 Select whether to use SSL when communicating with the LDAP server.

Step 9 Under Advanced, enter cache time-to-live. This value represents the amount of time to retain caches.

Step 10 Enter the maximum number of retained cache entries.

Step 11 Enter a maximum number of simultaneous connections.

If you configure the LDAP server profile for load balancing, these connections are distributed among the listed LDAP servers. For example, if you configure 10 simultaneous connections and load balance the connections over three servers, AsyncOS creates 10 connections to each server, for a total of 30 connections.

Note The maximum number of simultaneous connections includes LDAP connections used for LDAP queries. However, the appliance may open more connections if you use LDAP authentication for the Cisco IronPort Spam Quarantine.

Step 12 Test the connection to the server by clicking the **Test Server(s)** button. If you specified multiple LDAP servers, they are all tested. The results of the test appear in the Connection Status field. For more information, see Testing LDAP Servers, page 3-6.

Step 13 Create queries by marking the checkbox and completing the fields. You can select Accept, Routing, Masquerade, Group, SMTP Authentication, External Authentication, Spam Quarantine End-User Authentication, and Spam Quarantine Alias Consolidation.

Note To allow the Cisco IronPort appliance to run LDAP queries when you receive or send messages, you must enable the LDAP query on the appropriate listener. For more information, see Working with LDAP, LDAP Queries, and Listeners, page 3-7.

Step 14 Test a query by clicking the **Test Query** button.

Enter the test parameters and click **Run Test**. The results of the test appear in the Connection Status field. If you make any changes to the query definition or attributes, click **Update**. For more information, see Testing LDAP Queries, page 3-17.

Note If you have configured the LDAP server to allow binds with empty passwords, the query can pass the test with an empty password field.

Step 15 Submit and commit your changes.

Note Although the number of server configurations is unlimited, you can configure only one recipient acceptance, one routing, one masquerading, and one group query per server.

Testing LDAP Servers

Use the **Test Server(s)** button on the Add/Edit LDAP Server Profile page (or the `test` subcommand of the `ldapconfig` command in the CLI) to test the connection to the LDAP server. AsyncOS displays a message stating whether the connection to the server port succeeded or failed. If you configured multiple LDAP servers, AsyncOS tests each server and displays individual results.

Working with LDAP, LDAP Queries, and Listeners

To allow the Cisco IronPort appliance to run LDAP queries when you receive or send messages, you must enable the LDAP query on the appropriate listener.

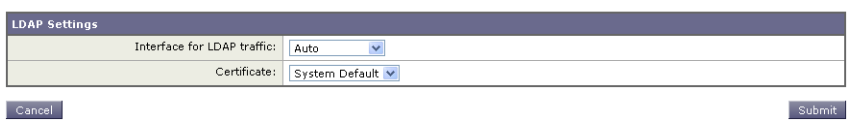
Configuring Global Settings

The LDAP global settings define how the appliance handles all LDAP traffic. To configure global settings for LDAP:

Step 1 On the System Administration > LDAP page, click **Edit Settings**.

The Edit LDAP Settings page is displayed:

Figure 3-3 Edit LDAP Settings Page



Step 2 Select the IP interface to use for LDAP traffic. The appliance automatically chooses an interface by default.

Step 3 Select the TLS certificate to use for the LDAP interface (TLS certificates added via the Network > Certificates page or the `certconfig` command in the CLI are available in the list, see Encrypting SMTP Conversations Using TLS, page 1-22).

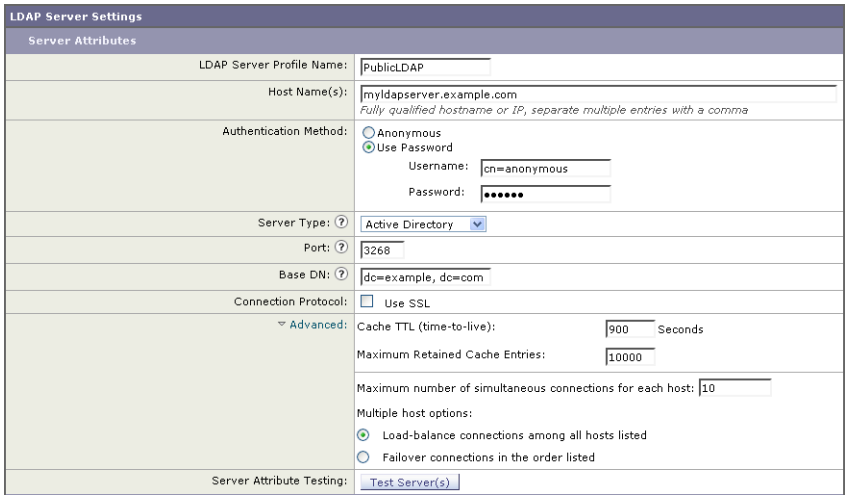
Step 4 Submit and commit your changes.

Example of Creating an LDAP Server Profile

In the following example, the System Administration > LDAP page is used to define an LDAP server for the appliance to bind to, and queries for recipient acceptance, routing, and masquerading are configured.

Note There is a 60 second connection attempt time-out for LDAP connections (which covers the DNS lookup, the connection itself, and, if applicable, the authentication bind for the appliance itself). After the first failure, AsyncOS immediately starts trying other hosts in the same server (if you specified more than one in the comma separated list). If you only have one host in the server, AsyncOS continues attempting to connect to it.

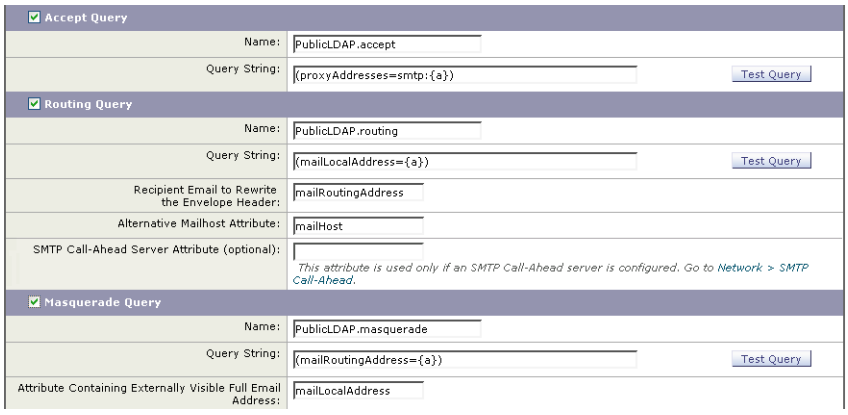
Figure 3-4 Configuring an LDAP Server Profile (1 of 2)



First, the nickname of "PublicLDAP" is given for the `myldapserver.example.com` LDAP server. The number of connections is set to 10 (the default), and the multiple LDAP server (hosts) load balance option is left as the default. You can specify multiple hosts here by providing a comma separated list of names. Queries are directed to port 3268 (the default). SSL is not enabled as the connection protocol for this host. The base DN of `example.com` is defined (`dc=example,dc=com`). The cache time-to-live is set to 900 seconds, the maximum number of cache entries is 10000, and the authentication method is set to password.

Queries for recipient acceptance, mail routing, and masquerading are defined. Remember that query names are case-sensitive and must match exactly in order to return the proper results.

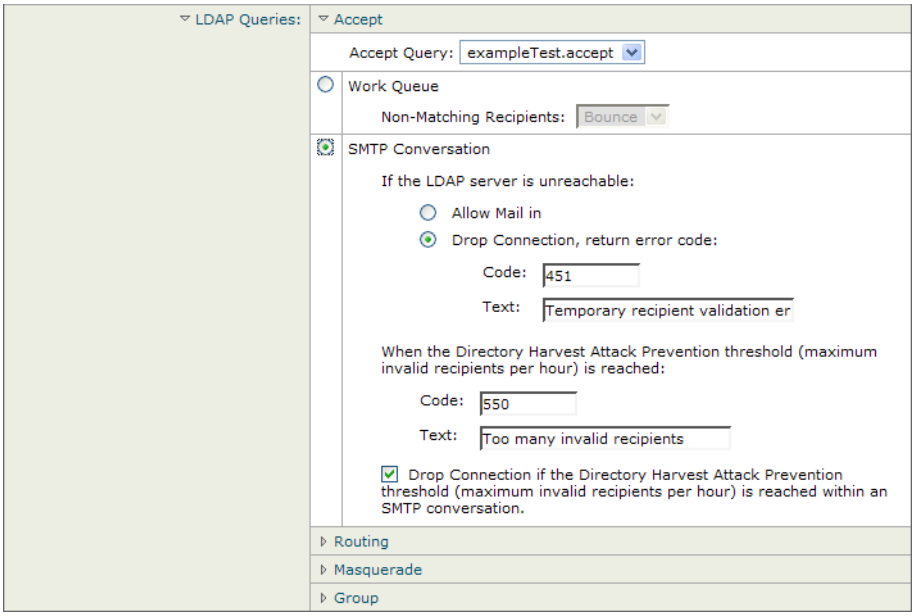
Figure 3-5 Configuring an LDAP Server Profile (2 of 2)



Enabling LDAP Queries on a Public Listener

In this example, the public listener "InboundMail" is updated to use LDAP queries for recipient acceptance. Further, recipient acceptance is configured to happen during the SMTP conversation (for more information, see Acceptance (Recipient Validation) Queries, page 3-18 for more information).

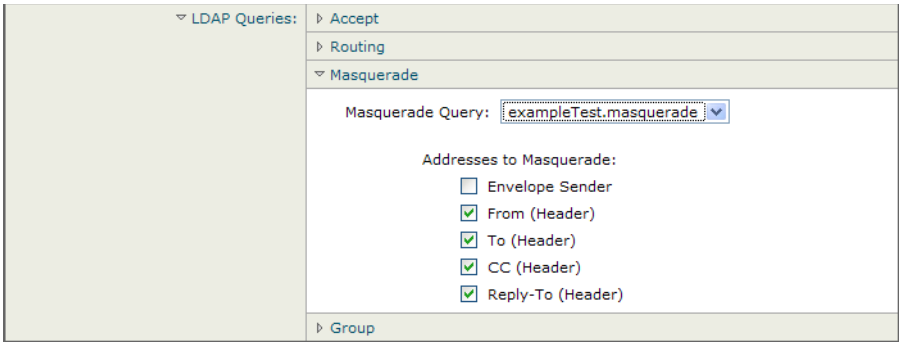
Figure 3-6 Enabling Acceptance and Routing Queries on a Listener



Enabling LDAP Queries on a Private Listener

In this example, the private listener "OutboundMail" is updated to use LDAP queries for masquerading. The masqueraded fields include: From, To, CC, and Reply-To.

Figure 3-7 Enabling a Masquerading Query on a Listener



Enhanced Support for Microsoft Exchange 5.5

AsyncOS includes a configuration option to provide support for Microsoft Exchange 5.5. If you use a later version of Microsoft Exchange, you do not need to enable this option. When configuring an LDAP server, you can elect to enable Microsoft Exchange 5.5 support by answering "y" when prompted in the `ldapconfig -> edit -> server -> compatibility` subcommand (this is only available via the CLI):

```
mail3.example.com> ldapconfig


Current LDAP server configurations:
```

```
1. PublicLDAP: (ldapexample.com:389)


Choose the operation you want to perform:

- NEW - Create a new server configuration.

- EDIT - Modify a server configuration.

- DELETE - Remove a server configuration.

[]> edit


Enter the name or number of the server configuration you wish to edit.

[]> 1


Name: PublicLDAP

Hostname: ldapexample.com Port 389

Authentication Type: anonymous

Base: dc=ldapexample,dc=com


Choose the operation you want to perform:

- SERVER - Change the server for the query.

- LDAPACCEPT - Configure whether a recipient address should be accepted or
bounced/dropped.

- LDAPROUTING - Configure message routing.

- MASQUERADE - Configure domain masquerading.

- LDAPGROUP - Configure whether a sender or recipient is in a specified group.

- SMTPAUTH - Configure SMTP authentication.

[]> server


Name: PublicLDAP

Hostname: ldapexample.com Port 389

Authentication Type: anonymous
```

```
Base: dc=ldapexample,dc=com

Microsoft Exchange 5.5 Compatibility Mode: Disabled


Choose the operation you want to perform:

- NAME - Change the name of this configuration.

- HOSTNAME - Change the hostname used for this query.

- PORT - Configure the port.

- AUTHTYPE - Choose the authentication type.

- BASE - Configure the query base.

- COMPATIBILITY - Set LDAP protocol compatibility options.

[]> compatibility


Would you like to enable Microsoft Exchange 5.5 LDAP compatibility mode?  (This is not
recommended for versions of Microsoft Exchange later than 5.5, or other LDAP servers.)
[N]> y


Do you want to configure advanced LDAP compatibility settings? (Typically not required)
[N]>


Name: PublicLDAP

Hostname: ldapexample.com Port 389

Authentication Type: anonymous

Base: dc=ldapexample,dc=com

Microsoft Exchange 5.5 Compatibility Mode: Enabled (attribute "objectClass")


Choose the operation you want to perform:

- NAME - Change the name of this configuration.

- HOSTNAME - Change the hostname used for this query.

- PORT - Configure the port.

- AUTHTYPE - Choose the authentication type.
```

```
- BASE - Configure the query base.

- COMPATIBILITY - Set LDAP protocol compatibility options.

[]>
```

Working with LDAP Queries

You create an entry in the LDAP server profile for each type of LDAP query you want to perform. When you create LDAP queries, you must enter the query syntax for your LDAP server. Please note that the queries you construct should be tailored and specific to your particular implementation of LDAP directory services, particularly if you have extended your directory with new object classes and attributes to accommodate the unique needs of your directory.

Types of LDAP Queries

The following sections provide sample queries and configuration details for each type of query:

- **Acceptance queries**. For more information, see Acceptance (Recipient Validation) Queries, page 3-18.
- **Routing queries**. For more information, see Routing: Alias Expansion, page 3-20.
- **Masquerading queries**. For more information, see Masquerading, page 3-20.
- **Group queries**. For more information, see Group LDAP Queries, page 3-22.
- **Domain-based queries**. For more information, see Domain-based Queries, page 3-26.
- **Chain queries**. For more information, see Chain Queries, page 3-27.

You can also configure queries for the following purposes:

- **Directory harvest prevention**. For more information, see Understanding LDAP Queries, page 3-2.
- **SMTP authentication.** For more information, see Configuring AsyncOS for SMTP Authentication, page 3-31.
- **External authentication.** For more information, Configuring External Authentication for Users, page 3-40.
- **Spam quarantine end-user authentication query.** For more information, see Spam Quarantine End-User Authentication Queries, page 3-43.
- **Spam quarantine alias consolidation query.** For more information, see Spam Quarantine Alias Consolidation Queries, page 3-44.

The search queries you specify are available to all listeners you configure on the system.

Base Distinguishing Name (DN)

The root level of the directory is called the base. The name of the base is the DN (distinguishing name). The base DN format for Active Directory (and the standard as per RFC 2247) has the DNS domain translated into domain components (dc=). For example, example.com's base DN would be: dc=example, dc=com. Note that each portion of the DNS name is represented in order. This may or may not reflect the LDAP settings for your configuration.

If your directory contains multiple domains you may find it inconvenient to enter a single BASE for your queries. In this case, when configuring the LDAP server settings, set the base to NONE. This will, however, make your searches inefficient.

LDAP Query Syntax

Spaces are allowed in LDAP paths, and they do not need to be quoted. The CN and DC syntax is not case-sensitive.

```
Cn=First Last,oU=user,dc=domain,DC=COM
```

The variable names you enter for queries are *case-sensitive* and must match your LDAP implementation in order to work correctly. For example, entering **`mailLocalAddress`** at a prompt performs a different query than entering **`maillocaladdress`**.

Tokens:

You can use the following tokens in your LDAP queries:

- {a} username@domainname
- {d} domainname
- {dn} distinguished name
- {g} groupname
- {u} username
- {f} MAIL FROM: address



Note The {f} token is valid in acceptance queries only.

For example, you might use the following query to accept mail for an Active Directory LDAP server:

(|(mail={a})(proxyAddresses=smtp:{a}))



Note Cisco Systems strongly recommends using the Test feature of the LDAP page (or the `test` subcommand of the `ldapconfig` command) to test all queries you construct and ensure that expected results are returned *before* you enable LDAP functionality on a listener. See Testing LDAP Queries, page 3-17 for more information.

Secure LDAP (SSL)

You can use instruct AsyncOS to use SSL when communicating with the LDAP server. If you configure your LDAP server profile to use SSL:

- AsyncOS will use the LDAPS certificate configured via `certconfig` in the CLI (see Creating a Self-Signed Certificate, page 1-23).

  You may have to configure your LDAP server to support using the LDAPS certificate.
- If an LDAPS certificate has not been configured, AsyncOS will use the demo certificate.

Routing Queries

There is no recursion limit for LDAP routing queries; the routing is completely data driven. However, AsyncOS does check for circular reference data to prevent the routing from looping infinitely.

Anonymous Queries

You may need to configure your LDAP directory server to allow for anonymous queries. (That is, clients can bind to the server anonymously and perform queries.) For specific instructions on configuring Active Directory to allow anonymous queries, see the "Microsoft Knowledge Base Article - 320528" at the following URL:

```
http://support.microsoft.com/default.aspx?scid=kb%3Ben-us%3B320528
```

Alternately, you can configure one "user" dedicated solely for the purposes of authenticating and performing queries instead of opening up your LDAP directory server for anonymous queries from any client.

A summary of the steps is included here, specifically:

- How to set up Microsoft Exchange 2000 server to allow "anonymous" authentication.
- How to set up Microsoft Exchange 2000 server to allow "anonymous bind."
- How to set up Cisco IronPort AsyncOS to retrieve LDAP data from a Microsoft Exchange 2000 server using both "anonymous bind" and "anonymous" authentication.

Specific permissions must be made to a Microsoft Exchange 2000 server in order to allow "anonymous" or "anonymous bind" authentication for the purpose of querying user email addresses. This can be very useful when an LDAP query is used to determine the validity of an income email message to the SMTP gateway.

Anonymous Authentication Setup

The following setup instructions allow you to make specific data available to unauthenticated queries of Active Directory and Exchange 2000 servers in the Microsoft Windows Active Directory. If you wish to allow "anonymous bind" to the Active Directory, see Anonymous Bind Setup for Active Directory, page 3-15.

Step 1 Determine required Active Directory permissions.

Using the ADSI Edit snap-in or the LDP utility, you must modify the permissions to the attributes of the following Active Directory objects:

- The root of the domain naming context for the domain against which you want to make queries.
- All OU and CN objects that contain users against which you wish to query email information.

The following table shows the required permissions to be applied to all of the needed containers.

User Object	Permissions	Inheritance	Permission Type
Everyone	List Contents	Container Objects	Object
Everyone	List Contents	Organizational Unit Objects	Object

User Object	Permissions	Inheritance	Permission Type
Everyone	Read Public Information	User Objects	Property
Everyone	Read Phone and Mail Options	User Objects	Property

Step 2 Set Active Directory Permissions

- Open ADSIEdit form the Windows 2000 Support Tools.
- Locate the **Domain Naming Context** folder. This folder has the LDAP path of your domain.
- Right click the **Domain Naming Context** folder, and then click **Properties**.
- Click **Security**.
- Click **Advanced**.
- Click **Add**.
- Click the **User Object** Everyone, and then click **OK**.
- Click the **Permission Type** tab.
- Click **Inheritance** from the **Apply onto** box.
- Click to select the Allow check box for the **Permission** permission.

Step 3 Configure the Cisco IronPort Messaging Gateway

Use `ldapconfig` on the Command Line Interface (CLI) to create an LDAP server entry with the following information.

- Hostname of an Active Directory or Exchange server
- Port 3268
- Base DN matching the root naming context of the domain
- Authentication type Anonymous

Anonymous Bind Setup for Active Directory

The following setup instructions allow you to make specific data available to anonymous bind queries of Active Directory and Exchange 2000 servers in the Microsoft Windows Active Directory. Anonymous bind of an Active Directory server will send the username `anonymous` with a blank password.

Note If a password is sent to an Active Directory server while attempting anonymous bind, authentication may fail.

Step 1 Determine required Active Directory permissions.

Using the ADSI Edit snap-in or the LDP utility, you must modify the permissions to the attributes of the following Active Directory objects.

- The root of the domain naming context for the domain against which you want to make queries.
- All OU and CN objects that contain users against which you wish to query email information.

The following table shows the required permissions to be applied to all of the needed containers.

User Object	Permissions	Inheritance	Permission Type
ANONYMOUS LOGON	List Contents	Container Objects	Object
ANONYMOUS LOGON	List Contents	Organizational Unit Objects	Object
ANONYMOUS LOGON	Read Public Information	User Objects	Property
ANONYMOUS LOGON	Read Phone and Mail Options	User Objects	Property

Step 2 Set Active Directory Permissions

- Open ADSIEdit form the Windows 2000 Support Tools.
- Locate the **Domain Naming Context** folder. This folder has the LDAP path of your domain.
- Right click the **Domain Naming Context** folder, and then click **Properties**.
- Click **Security**.
- Click **Advanced**.
- Click **Add**.
- Click the **User Object** ANONYMOUS LOGON, and then click **OK**.
- Click the **Permission Type** tab.
- Click **Inheritance** from the **Apply** onto box.
- Click to select the **Allow** check box for the **Permission** permission.

Step 3 Configure the Cisco IronPort Messaging Gateway

Use the System Administration > LDAP page (or `ldapconfig` in the CLI) to create an LDAP server entry with the following information.

- Hostname of an Active Directory or Exchange server
- Port 3268
- Base DN matching the root naming context of the domain
- Authentication type password based using `cn=anonymous` as the user with a blank password

Notes for Active Directory Implementations

- Active Directory servers accept LDAP connections on ports 3268 and 389. The default port for accessing the global catalog is port 3268.
- Active Directory servers accept LDAPS connections on ports 636 and 3269. Microsoft supports LDAPS on Windows Server 2003 and higher.
- The Cisco IronPort appliance should connect to a domain controller that is also a global catalog so that you can perform queries to different bases using the same server.
- Within Active Directory, you may need to grant read permissions to the group “Everyone” to directory objects to yield successful queries. This includes the root of the domain naming context.
- Generally, the value of the `mail` attribute entry in many Active Directory implementations has a matching value “ProxyAddresses” attribute entry.

- Microsoft Exchange environments that are aware of each other within the infrastructure can usually route mail between each other without involving a route back to the originating MTA.

Testing LDAP Queries

Use the Test Query button on the Add/Edit LDAP Server Profile page (or the `test` subcommand in the CLI) of each query type to test the query to the LDAP server you configured. In addition to displaying the result, AsyncOS also displays the details on each stage of the query connection test. You can test each of the query types.

The `ldaptest` command is available as a batch command, for example:

```
ldaptest LDAP.ldapaccept foo@ironport.com
```

If you entered multiple hosts in the Host Name field of the LDAP server attributes, the Cisco IronPort appliance tests the query on each LDAP server.

Table 3-1 summarizes the testing results. (You can also use the `ldaptest` command.)

Table 3-1 Testing LDAP Queries

Query type	If a recipient matches (PASS)...	If a recipient does not match (FAIL)...
Recipient Acceptance (Accept, `ldapaccept`)	Accept the message.	Invalid Recipient: Conversation or delayed bounce or drop the message per listener settings. DHAP: Drop.
Routing (Routing, `ldaprouting`)	Route based on the query settings.	Continue processing the message.
Masquerade (Masquerade, `masquerade`)	Alter the headers with the variable mappings defined by the query.	Continue processing the message.
Group Membership (Group, `ldapgroup`)	Return “true” for message filter rules.	Return “false” for message filter rules.
SMTP Auth (SMTP Authentication, `smtpauth`)	A password is returned from the LDAP server and is used for authentication; SMTP Authentication occurs.	No password match can occur; SMTP Authentication attempts fail.
External Authentication (`externalauth`)	Individually returns a “match positive” for the bind, the user record, and the user’s group membership.	Individually returns a “match negative” for the bind, the user record, and the user’s group membership.
Spam Quarantine End-User Authentication (`isqauth`)	Returns a “match positive” for the end-user account.	No password match can occur; End-User Authentication attempts fail.
Spam Quarantine Alias Consolidation (`isqalias`)	Returns the email address that the consolidated spam notifications will be sent to.	No consolidation of spam notifications can occur.

Note The variable names you enter for queries are *case-sensitive* and must match your LDAP implementation in order to work correctly. For example, entering **`mailLocalAddress`** at a prompt performs a different query than entering **`maillocaladdress`**. Cisco IronPort Systems strongly recommends using the `test` subcommand of the `ldapconfig` command to test all queries you construct and ensure the proper results are returned.

Troubleshooting Connections to LDAP Servers

If the LDAP server is unreachable by the appliance, one of the following errors will be shown:

- `Error: LDAP authentication failed: <LDAP Error "invalidCredentials" [0x31]>`
- `Error: Server unreachable: unable to connect`
- `Error: Server unreachable: DNS lookup failure`

Note that a server may be unreachable because the wrong port was entered in the server configuration, or the port is not opened in the firewall. LDAP servers typically communicate over port 3268 or 389. Active Directory uses port 3268 to access the global catalog used in multi-server environments (See "Firewall Information" in the *Cisco IronPort AsyncOS for Email Configuration Guide* for more information.) In AsyncOS 4.0, the ability to communicate to the LDAP server via SSL (usually over port 636) was added. For more information, see Secure LDAP (SSL), page 3-13.

A server may also be unreachable because the hostname you entered cannot be resolved.

You can use the **Test Server(s)** on the Add/Edit LDAP Server Profile page (or the `test` subcommand of the `ldapconfig` command in the CLI) to test the connection to the LDAP server. For more information, see Testing LDAP Servers, page 3-6.

If the LDAP server is unreachable:

- If LDAP Accept or Masquerading or Routing is enabled on the work queue, mail will remain within the work queue.
- If LDAP Accept is not enabled but other queries (group policy checks, etc.) are used in filters, the filters evaluate to false.

Acceptance (Recipient Validation) Queries

You can use your existing LDAP infrastructure to define how the recipient email address of incoming messages (on an public listener) should be handled. Changes to user data in your directories are updated the next time the Cisco IronPort appliance queries the directory server. You can specify the size of the caches and the amount of time the Cisco IronPort appliance stores the data it retrieves.

Note You may wish to bypass LDAP acceptance queries for special recipients (such as `administrator@example.com`). You can configure this setting from the Recipient Access Table (RAT). For information about configuring this setting, see "Configuring the Gateway to Receive Email" in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

Sample Acceptance Queries

Table 3-2 shows sample acceptance queries.

Table 3-2 Example LDAP Query Strings for Common LDAP Implementations: Acceptance

Query for:	Recipient validation
OpenLDAP	`(mailLocalAddress={a})` `(mail={a})` `(mailAlternateAddress={a})`
Microsoft Active Directory Address Book **Microsoft Exchange**	`(|(mail={a})(proxyAddresses=smtp:{a}))`
SunONE Directory Server	`(mail={a})` `(mailAlternateAddress={a})` `(mailEquivalentAddress={a})` `(mailForwardingAddress={a})` `(mailRoutingAddress={a})`
Lotus Notes **Lotus Domino**	`(|(|(mail={a})(uid={u}))(cn={u}))` `(|(ShortName={u})(InternetAddress={a})(FullName={u}))`

You can also validate on the username (Left Hand Side). This is useful if your directory does not contain all the domains you accept mail for. Set the Accept query to (uid={u}).

Configuring Acceptance Queries for Lotus Notes

Note that there is a potential complication with LDAPACCEPT and Lotus Notes. If Notes LDAP contains a person with attributes like these:

```
mail=juser@example.com

cn=Joe User

uid=juser

cn=123456

location=New Jersey
```

Lotus accepts email for this person for various different forms of email addresses, other than what is specified, such as “Joe_User@example.com” — which do not exist in the LDAP directory. So AsyncOS may not be able to find all of the valid user email addresses for that user.

One possible solution is to try to publish the other forms of addresses. Please contact your Lotus Notes administrator for more details.

Routing: Alias Expansion

AsyncOS supports alias expansion (LDAP routing with multiple target addresses). AsyncOS replaces the original email message with a new, separate message for each alias target (for example, recipient@yoursite.com might be replaced with new separate messages to newrecipient1@hotmail.com and recipient2@internal.yourcompany.com, etc.). Routing queries are sometimes known as aliasing queries on other mail processing systems.

Sample Routing Queries

Table 3-3 Example LDAP Query Strings for Common LDAP Implementations: Routing

Query for:	Route to another mailhost
OpenLDAP	`(mailLocalAddress={a})`
Microsoft Active Directory Address Book **Microsoft Exchange**	May not be applicable[a]
SunONE Directory Server	`(mail={a})` `(mailForwardingAddress={a})` `(mailEquivalentAddress={a})` `(mailRoutingAddress={a})` `(otherMailbox={a})` `(rfc822Mailbox={a})`

a.Active Directory implementations can have multiple entries for the `proxyAddresses` attribute, but because AD formats this attribute value as `smtp:user@domain.com`, that data cannot be used for LDAP routing/alias expansion. Each target address must be in a separate `attribute:value` pair. Microsoft Exchange environments that are aware of each other within the infrastructure can usually route mail between each other without involving a route back to the originating MTA.

Routing: MAILHOST and MAILROUTINGADDRESS

For Routing queries, the value of MAILHOST cannot be an IP address; it must be a resolvable hostname. This usually requires the use of an Internal DNSconfig.

MAILHOST is optional for the routing query. MAILROUTINGADDRESS is mandatory if MAILHOST is not set.

Masquerading

Masquerading is a feature that rewrites the Envelope Sender (also known as the sender, or MAIL FROM) and the To:, From:, and/or CC: headers on email based on queries you construct. A typical example implementation of this feature is “Virtual Domains,” which allows you to host multiple domains from a single site. Another typical implementation is “hiding” your network infrastructure by “stripping” the subdomains from strings in email headers.

Sample Masquerading Queries

Table 3-4 Example LDAP Query Strings for Common LDAP Implementation: Masquerading

Query for:	Masquerade
OpenLDAP	`(mailRoutingAddress={a})`
Microsoft Active Directory Address Book	`(proxyaddresses=smtp:{a})`
SunONE Directory Server	`(mail={a})` `(mailAlternateAddress={a})` `(mailEquivalentAddress={a})` `(mailForwardingAddress={a})` `(mailRoutingAddress={a})`

Masquerading "Friendly Names"

In some user environments, an LDAP directory server schema may store a "friendly name" in addition to a mail routing address or a local mail address. AsyncOS allows you to masquerade Envelope Senders (for outgoing mail) and message headers (for incoming mail, such as To:, Reply To:, From: or CC:) with this "friendly address" — even if the friendly address contains special characters that are not normally permitted in a valid email address (for example, quotation marks, spaces, and commas).

When using masquerading of headers via an LDAP query, you now have the option to configure whether to replace the entire friendly email string with the results from the LDAP server. Note that even with this behavior enabled, only the user@domain portion will be used for the Envelope Sender (the friendly name is illegal).

As with the normal LDAP masquerading, if empty results (zero length or entire white space) are returned from the LDAP query, no masquerading occurs.

To enable this feature, answer "y" to the following question when configuring an LDAP-based masquerading query for a listener (LDAP page or `ldapconfig` command):

```
Do you want the results of the returned attribute to replace the entire
friendly portion of the original recipient? [N]
```

For example, consider the following example LDAP entry:

Attribute	Value
mailRoutingAddress	admin\@example.com
mailLocalAddress	joe.smith\@example.com
mailFriendlyAddress	"Administrator for example.com," <joe.smith\@example.com>

If this feature is enabled, an LDAP query of (mailRoutingAddress={a}) and a masquerading attribute of (mailLocalAddress) would result in the following substitutions:

Original Address (From, To, CC, Reply-to)	Masqueraded Headers	Masqueraded Envelope Sender
admin@example.com	From: "Administrator for example.com," <joe.smith@example.com>	MAIL FROM: <joe.smith@example.com>

Group LDAP Queries

You can define a query to your LDAP servers to determine if a recipient is a member of a group as defined by your LDAP directory.

Configuring LDAP group queries involves three steps:

Step 1 Create a message filter that uses a `rcpt-to-group` or `mail-from-group` rule to act upon the message.

Step 2 Then, use the System Administration > LDAP page (or the `ldapconfig` command) to define the LDAP server for the appliance to bind to and configure a query for a group membership.

Step 3 Use the Network > Listeners page (or the `listenerconfig -> edit -> ldapgroup` subcommand) to enable the group query for the listener.

Sample Group Queries

Table 3-5 Example LDAP Query Strings for Common LDAP Implementation: Group

Query for:	Group
OpenLDAP	OpenLDAP does not support the `memberOf` attribute by default. Your LDAP Administrator may add this attribute or a similar attribute to the schema.
Microsoft Active Directory	`(&(memberOf={g})(proxyAddresses=smtp:{a}))`
SunONE Directory Server	`(&(memberOf={g})(mailLocalAddress={a}))`

For example, suppose that your LDAP directory classifies members of the "Marketing" group as `ou=Marketing`. You can use this classification to treat messages sent to or from members of this group in a special way. Step 1 creates a message filter to act upon the message, and Steps 2 and 3 enable the LDAP lookup mechanism.

Configuring a Group Query

In the following example, mail from members of the Marketing group (as defined by the LDAP group "Marketing") will be delivered to the alternate delivery host `marketingfolks.example.com`.

Step 1 First, a message filter is created to act upon messages that match positively for group membership. In this example, a filter is created that uses the `mail-from-group` rule. All messages whose Envelope Sender is found to be in the LDAP group "marketing-group1" will be delivered with an alternate delivery host (the filters `alt-mailhost` action).

The group membership field variable (groupName) will be defined in step 2. The group attribute "groupName" is defined with the value `marketing-group1`.

```
mail3.example.com> filters


Choose the operation you want to perform:

- NEW - Create a new filter.
```

```
- IMPORT - Import a filter script from a file.

[]> new


Enter filter script.  Enter '.' on its own line to end.

MarketingGroupfilter:

  if (mail-from-group == "marketing-group1") {

      alt-mailhost ('marketingfolks.example.com');}

.

1 filters added.


Choose the operation you want to perform:

- NEW - Create a new filter.

- DELETE - Remove a filter.

- IMPORT - Import a filter script from a file.

- EXPORT - Export filters to a file

- MOVE - Move a filter to a different position.

- SET - Set a filter attribute.

- LIST - List the filters.

- DETAIL - Get detailed information on the filters.

- LOGCONFIG - Configure log subscriptions used by filters.

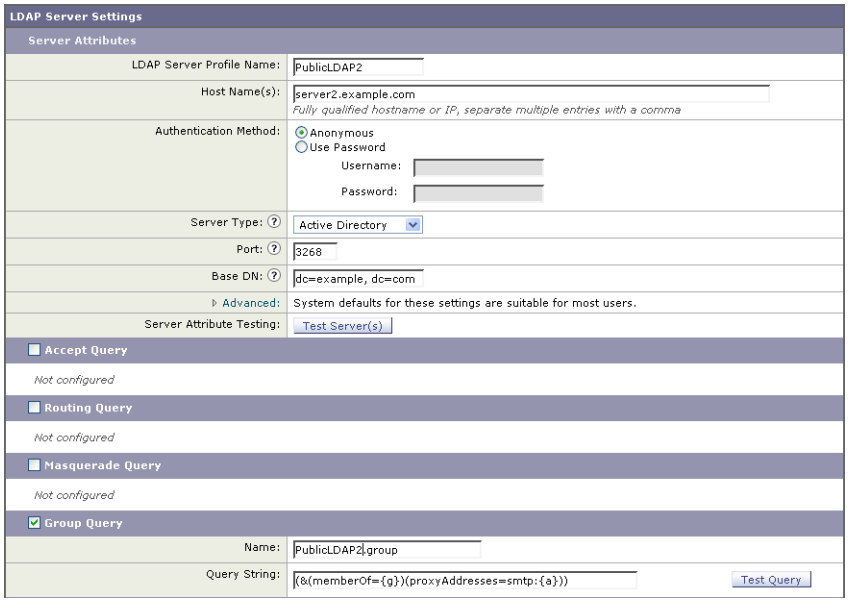
- ROLLOVERNOW - Roll over a filter log file.

[]>
```

For more information on the `mail-from-group` and `rcpt-to-group` message filter rules, see Message Filter Rules, page 6-2.

Step 2 Next, the Add LDAP Server Profile page is used to define an LDAP server for the appliance to bind to, and an initial query for a group membership is configured.

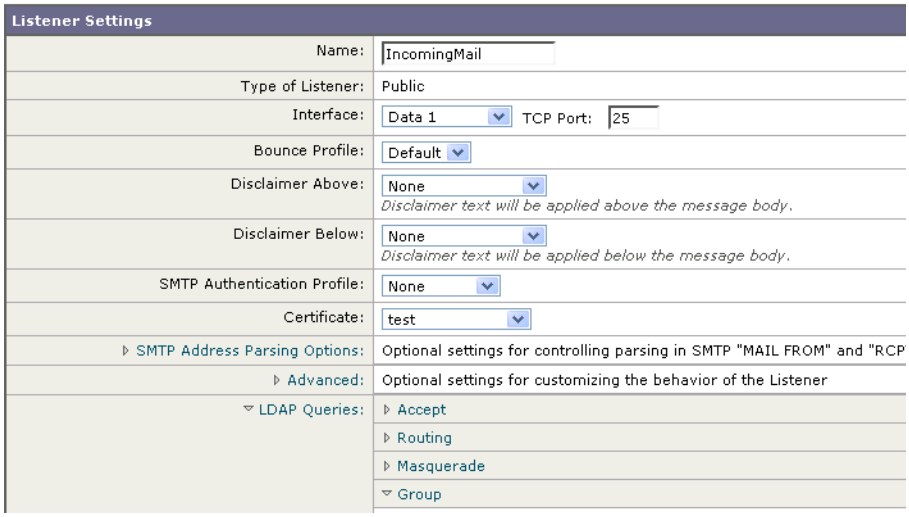
Figure 3-8 Adding a New LDAP Profile and Group Query



Step 3 Next, the public listener "InboundMail" is updated to use LDAP queries for group routing. The Edit Listener page is used to enable the LDAP query specified above.

As a result of this query, messages accepted by the listener trigger a query to the LDAP server to determine group membership. The PublicLDAP2.group query was defined previously via the System Administration > LDAP page.

Figure 3-9 Specifying a Group Query on a Listener



Note that in this example, a commit must be issued for the changes to take effect.

Example: Using a Group Query to Skip Spam and Virus Checking

Because message filters occurs early in the pipeline, you can use a group query to skip virus and spam checking for specified groups. For example, you want your IT group to receive all messages and to skip spam and virus checking. In your LDAP record, you create a group entry that uses the DN as the group name. The group name consists of the following DN entry:

cn=IT, ou=groups, o=sample.com

You create an LDAP server profile with the following group query:

(&(memberOf={g})(proxyAddresses=smtp:{a}))

You then enable this query on a listener so that when a message is received by the listener, the group query is triggered.

To skip virus and spam filtering for members of the IT group, you create the following message filter to check incoming messages against LDAP groups.

```
[]> - NEW - Create a new filter.

- IMPORT - Import a filter script from a file.

[]> new


Enter filter script.  Enter '.' on its own line to end.

IT_Group_Filter:

if (rcpt-to-group == "cn=IT, ou=groups, o=sample.com"){

skip-spamcheck();

skip-viruscheck();

deliver();

}

.

1 filters added.
```

Note The rcpt-to-group in this message filter reflects the DN entered as the group name: cn=IT, ou=groups, o=sample.com. Verify that you use the correct group name in the message filter to ensure that your filter matches the name in your LDAP directory.

Messages accepted by the listener trigger a query to the LDAP server to determine group membership. If the message recipient is a member of the IT group, the message filter skips both virus and spam checking and delivers the message to the recipient. To enable the filter to check the results of the LDAP query, you must create the LDAP query on the LDAP server and enable the LDAP query on a listener.

Domain-based Queries

Domain-based queries are LDAP queries grouped by type, associated with a domain, and assigned to a particular listener. You might want to use domain-based queries if you have different LDAP servers associated with different domains but you want to run queries for all your LDAP servers on the same listener. For example, the company "Bigfish" purchases company "Redfish" and company "Bluefish." Bigfish maintains its domain, Bigfish.com as well as domains for Redfish.com and Bluefish.com, and it maintains a different LDAP server for employees associated with each domain. To accept mail for all three of these domains, Bigfish creates domain-based queries. This allows Bigfish to accept emails for Bigfish.com, Redfish.com, and Bluefish.com on the same listener.

To configure domain-based queries, complete the following steps:

Step 1 Create a server profile for each of the domains you want to use in the domain-based queries. For each of the server profiles, configure the queries you want to use for a domain-based query (acceptance, routing, etc.). For more information, see Creating LDAP Server Profiles, page 3-4.

Step 2 Create the domain-based query. When you create the domain-based query, you select queries from each server profile, and enable the Cisco IronPort appliance to determine which query to run based on the domain in the Envelope To field. For more information about creating the query, see Creating a Domain-Based Query, page 3-26.

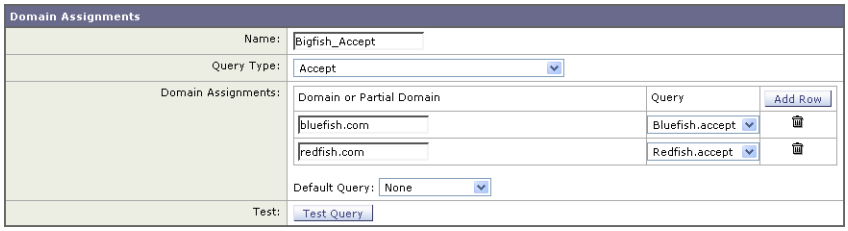
Step 3 Enable the domain-based query on the public or private listener. For more information about configuring listeners, see "Configuring the Gateway to Receive Mail" in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

Note You can also enable domain-based queries for LDAP end-user access or spam notifications for the Cisco IronPort Spam Quarantine. For more information, see "Configuring the Cisco IronPort Spam Quarantines Feature" in the *Cisco IronPort AsyncOS for Email Daily Management Guide*.

Creating a Domain-Based Query

You create a domain-based query from the System Administration > LDAP > LDAP Server Profiles page.

Figure 3-10 Configuring a domain-based query



Step 1 From the LDAP Server Profiles page, click **Advanced**.

Step 2 Click **Add Domain Assignments.**

Step 3 The Domain Assignments page opens.

Step 4 Enter a name for the domain-based query.

Step 5 Select the query type.

Note When you create domain-based queries, you cannot select different types of queries. Once you select a query type, the Cisco IronPort appliance populates the query field with queries of that type from the available server profiles.

Step 6 In the Domain Assignments field, enter a domain.

Step 7 Select a query to associate with the domain.

Step 8 Continue to add rows until you have added all the domains to your query.

Step 9 You can enter a default query to run if all other queries fail. If you do not want to enter a default query, select **None**.

Step 10 Test the query by clicking the **Test Query** button and entering a user login and password or an email address to test in the Test Parameters fields. The results appear in the Connection Status field.

Step 11 Optionally, if you use the {f} token in an acceptance query, you can add an envelope sender address to the test query.

Note Once you create the domain-based query, you need to associate it with a public or private listener.

Step 12 Submit and commit your changes.

Chain Queries

A chain query is a series of LDAP queries that the Cisco IronPort appliance attempts to run in succession. The Cisco IronPort appliance attempts to run each query in the "chain" until the LDAP server returns a positive response (or the final query in the "chain" returns a negative response or fails). Chain queries can be useful if entries in your LDAP directory use different attributes to store similar (or the same) values. For example, you might have used the attributes `maillocaladdress` and `mail` to store user email addresses. To ensure that your queries run against both these attributes, you can use chain queries.

To configure chain queries, complete the following steps:

Step 1 Create server profiles for each of the queries you want to use in the chain queries. For each of the server profiles, configure the queries you want to use for a chain query. For more information, see Creating LDAP Server Profiles, page 3-4.

Step 2 Create the chain query. For more information, see Creating a Chain Query, page 3-28.

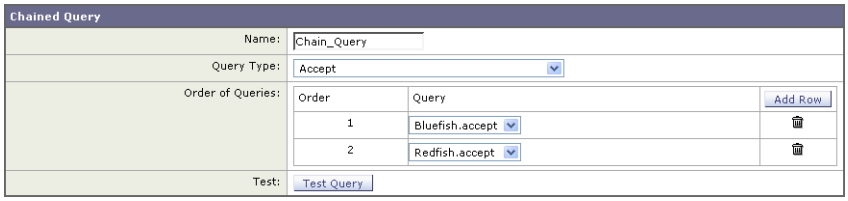
Step 3 Enable the chain query on the public or private listener. For more information about configuring listeners, see "Configuring the Gateway to Receive Mail" in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

Note You can also enable domain-based queries for LDAP end-user access or spam notifications for the Cisco IronPort Spam Quarantine. For more information, see "Configuring the Cisco IronPort Spam Quarantines Feature" in the *Cisco IronPort AsyncOS for Email Daily Management Guide*.

Creating a Chain Query

You create a chain query from the System Administration > LDAP > LDAP Server Profiles page.

Figure 3-11 Configuring a Chain Query



Step 1 From the LDAP Server Profiles page, click **Advanced**.

Step 2 Click **Add Chain Query**.

The Chain query page opens.

Step 3 Add a name for the chain query.

Step 4 Select the query type.

When you create chain queries, you cannot select different types of queries. Once you select a query type, the Cisco IronPort appliance populates the query field with queries of that type from available server profiles.

Step 5 Select a query to add to the chain query.

The Cisco IronPort appliance runs the queries in the order you configure them. Therefore, if you add multiple queries to the chain query, you might want to order the queries so that more specific queries are followed by more general queries.

Step 6 Test the query by clicking the **Test Query** button and entering a user login and password or an email address to test in the Test Parameters fields. The results appear in the Connection Status field.

Step 7 Optionally, if you use the {f} token in an acceptance query, you can add an envelope sender address to the test query.

Note Once you create the chain query, you need to associate it with a public or private listener.

Step 8 Submit and commit your changes.

Using LDAP For Directory Harvest Attack Prevention

Directory Harvest Attacks occur when a malicious sender attempts to send messages to recipients with common names, and the email gateway responds by verifying that a recipient has a valid mailbox at that location. When performed on a large scale, malicious senders can determine who to send mail to by "harvesting" these valid addresses for spamming.

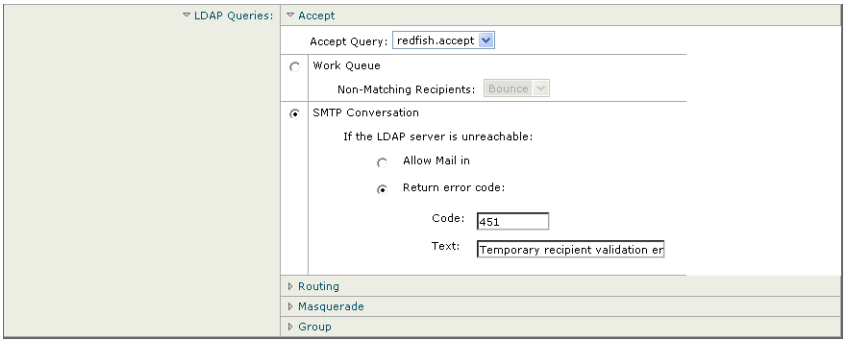
The Cisco IronPort Email Security appliance can detect and prevent Directory Harvest Attack (DHA) when using LDAP acceptance validation queries. You can configure LDAP acceptance to prevent directory harvest attacks within the SMTP conversation or within the work queue.

Directory Harvest Attack Prevention within the SMTP Conversation

You can prevent DHAs by entering only domains in the Recipient Access Table (RAT), and performing the LDAP acceptance validation in the SMTP conversation.

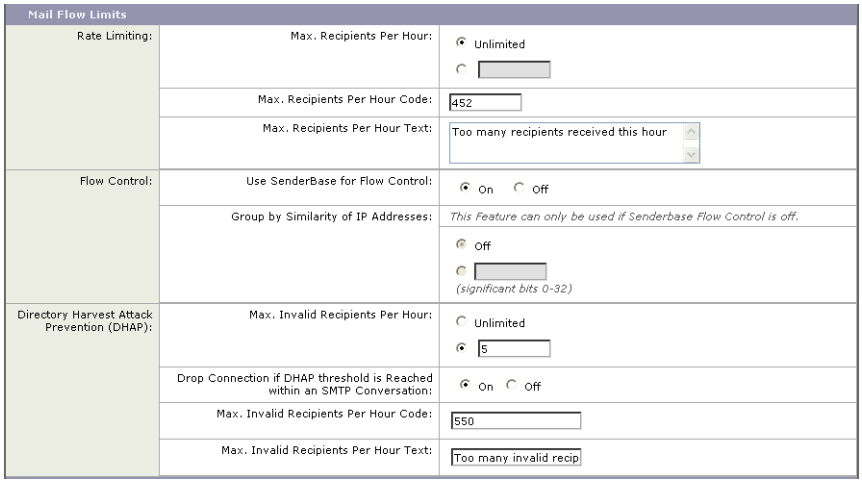
To drop messages during the SMTP conversation, configure an LDAP server profile for LDAP acceptance. Then, configure the listener to perform an LDAP accept query during the SMTP conversation.

Figure 3-12 Configuring the Acceptance Query in the SMTP Conversation



Once you configure LDAP acceptance queries for the listener, you must configure DHAP settings in the mail flow policy associated with the listener.

Figure 3-13 Configuring the Mail Flow Policy to Drop Connections in the SMTP Conversation



In the mail flow policy associated with the listener, configure the following Directory Harvest Attack Prevention settings:

- **Max. Invalid Recipients Per hour**. The maximum number of invalid recipients per hour this listener will receive from a remote host. This threshold represents the total number of RAT rejections combined with the total number of messages to invalid LDAP recipients dropped in the SMTP conversation or bounced in the work queue. For example, you configure the threshold as five, and the counter detects two RAT rejections and three dropped messages to invalid LDAP recipients. At this point, the Cisco IronPort appliance determines that the threshold is reached, and the connection is dropped. By default, the maximum number of recipients per hour for a public listener is 25. For a private listener, the maximum number of recipients per hour is unlimited by default. Setting it to “Unlimited” means that DHAP is not enabled for that mail flow policy.

- **Drop Connection if DHAP Threshold is reached within an SMTP conversation**. Configure the Cisco IronPort appliance to drop the connection if the Directory Harvest Attack Prevention threshold is reached.
- **Max. Recipients Per Hour Code**. Specify the code to use when dropping connections. The default code is 550.
- **Max. Recipients Per Hour Text**. Specify the text to use for dropped connections. The default text is "Too many invalid recipients."

If the threshold is reached, the Envelope Sender of the message does not receive a bounce message when a recipient is invalid.

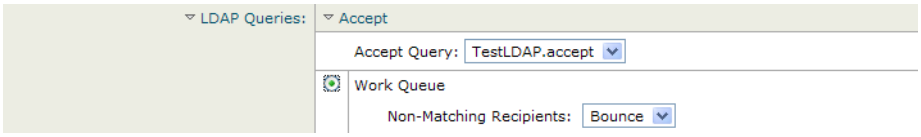
Directory Harvest Attack Prevention within the Work Queue

You can prevent most DHAs by entering only domains in the Recipient Access Table (RAT), and performing the LDAP acceptance validation within the work queue. This technique prevents the malicious senders from knowing if the recipient is valid during the SMTP conversation. (When acceptance queries are configured, the system accepts the message and performs the LDAP acceptance validation within the work queue.) However, the Envelope Sender of the message will still receive a bounce message if a recipient is not valid.

Configuring Directory Harvest Prevention in the Work Queue

To prevent Directory Harvest Attacks, you first configure an LDAP server profile, and enable LDAP Accept. Once you have enabled LDAP acceptance queries, configure the listener to use the accept query, and to bounce mail for non-matching recipients:

Figure 3-14 Configuring the Acceptance Query to Bounce Messages for Non-Matching Recipients



Next, configure the Mail Flow Policy to define the number of invalid recipient addresses the system will allow per sending IP address for a specific period of time. When this number is exceeded, the system will identify this condition as a DHA and send an alert message. The alert message will contain the following information:

```
LDAP: Potential Directory Harvest Attack from host=('IP-address', 'domain_name'),
dhap_limit=n, sender_group=sender_group,

listener=listener_name, reverse_dns=(reverse_IP_address, 'domain_name', 1),
sender=envelope_sender, rcpt=envelope_recipients
```

The system will bounce the messages up to the threshold you specified in the mail flow policy and then it will silently accept and drop the rest, thereby informing legitimate senders that an address is bad, but preventing malicious senders from determining which receipts are accepted.

This invalid recipients counter functions similarly to the way Rate Limiting is currently available in AsyncOS: you enable the feature and define the limit as part of the mail flow policy in a public listener's HAT (including the default mail flow policy for the HAT).

For example, you are prompted with these questions when creating or editing a mail flow policy in a public listener's HAT in the CLI — the `listenerconfig -> edit -> hostaccess -> default | new` commands:

```
Do you want to enable Directory Harvest Attack Prevention per host?   [Y]> y


Enter the maximum number of invalid recipients per hour from a remote host.

[25]>
```

This feature is also displayed when editing any mail flow policy in the GUI, providing that LDAP queries have been configured on the corresponding listener:

Figure 3-15 DHAP Prevention Feature in GUI



Entering a number of invalid recipients per hour enables DHAP for that mail flow policy. By default, 25 invalid recipients per hour are allowed for public listeners. For private listeners, the maximum invalid recipients per hour is unlimited by default. Setting it to "Unlimited" means that DHAP is not enabled for that mail flow policy.

Configuring AsyncOS for SMTP Authentication

AsyncOS provides support for SMTP authentication. SMTP Auth is a mechanism for authenticating clients connected to an SMTP server.

The practical use of this mechanism is that users at a given organization are able to send mail using that entity's mail servers even if they are connecting remotely (e.g. from home or while traveling). Mail User Agents (MUAs) can issue an authentication request (challenge/response) when attempting to send a piece of mail.

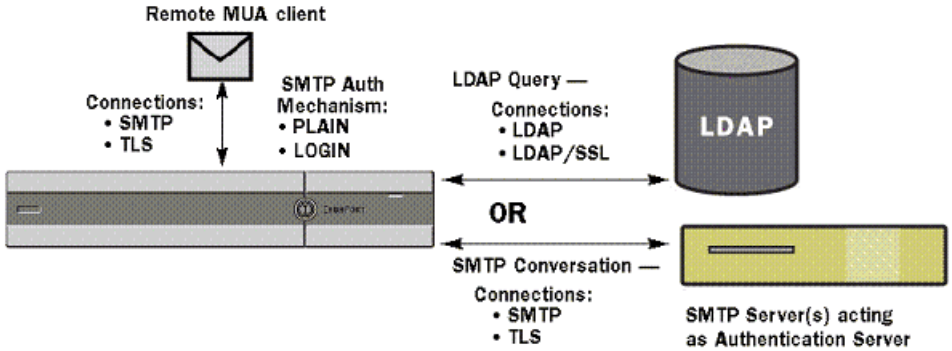
Users can also use SMTP authentication for outgoing mail relays. This allows the Cisco IronPort appliance to make a secure connection to a relay server in configurations where the appliance is not at the edge of the network.

AsyncOS complies with RFC 2554 which defines how an authentication command may be given in an SMTP conversation, the responses to the negotiation, and any error codes that may need to be generated.

AsyncOS supports two methods to authenticate user credentials:

- You can use an LDAP directory.
- You can use a different SMTP server (SMTP Auth forwarding and SMTP Auth outgoing).

Figure 3-16 SMTP Auth Support: LDAP Directory Store or SMTP Server



Configured SMTP Authentication methods are then used to create SMTP Auth profiles via the `smtpauthconfig` command for use within HAT mail flow policies (see Enabling SMTP Authentication on a Listener, page 3-36).

Configuring SMTP Authentication

If you are going to authenticate with an LDAP server, select the SMTPAUTH query type on the Add or Edit LDAP Server Profile pages (or in the `ldapconfig` command) to create an SMTP Authentication query. For each LDAP server you configure, you can configure a SMTPAUTH query to be used as an SMTP Authentication profile.

There are two kinds of SMTP authentication queries: LDAP bind and Password as attribute. When you use password as attribute, the Cisco IronPort appliance will fetch the password field in the LDAP directory. The password may be stored in plain text, encrypted, or hashed.When you use LDAP bind, the Cisco IronPort appliance attempts to log into the LDAP server using the credentials supplied by the client.

Specifying a Password as Attribute

The convention in OpenLDAP, based on RFC 2307, is that the type of coding is prefixed in curly braces to the encoded password (for example, "{SHA}5en6G6MezRroT3XKqkdPOmY/BfQ="). In this example, the password portion is a base64 encoding of a plain text password after application of SHA.

The Cisco IronPort appliance negotiates the SASL mechanism with the MUA before getting the password, and the appliance and the MUA decide on what method (LOGIN, PLAIN, MD5, SHA, SSHA, and CRYPT SASL mechanisms are supported). Then, the appliance queries the LDAP database to fetch a password. In LDAP, the password can have a prefix in braces.

- If there is no prefix, the appliance assumes that the password was stored in LDAP in plaintext.
- If there is a prefix, the appliance will fetch the hashed password, perform the hash on the username and/or password supplied by the MUA, and compare the hashed versions. The Cisco IronPort appliance supports SHA1 and MD5 hash types based on the RFC 2307 convention of prepending the hash mechanism type to the hashed password in the password field.
- Some LDAP servers, like the OpenWave LDAP server, do not prefix the encrypted password with the encryption type; instead, they store the encryption type as a separate LDAP attribute. In these cases, you can specify a default SMTP AUTH encryption method the appliance will assume when comparing the password with the password obtained in the SMTP conversation.

The Cisco IronPort appliance takes an arbitrary username from the SMTP Auth exchange and converts that to an LDAP query that fetches the clear or hashed password field. It will then perform any necessary hashing on the password supplied in the SMTP Auth credentials and compare the results with what it has retrieved from LDAP (with the hash type tag, if any, removed). A match means that the SMTP Auth conversation shall proceed. A failure to match will result in an error code.

Configuring an SMTP Authentication Query

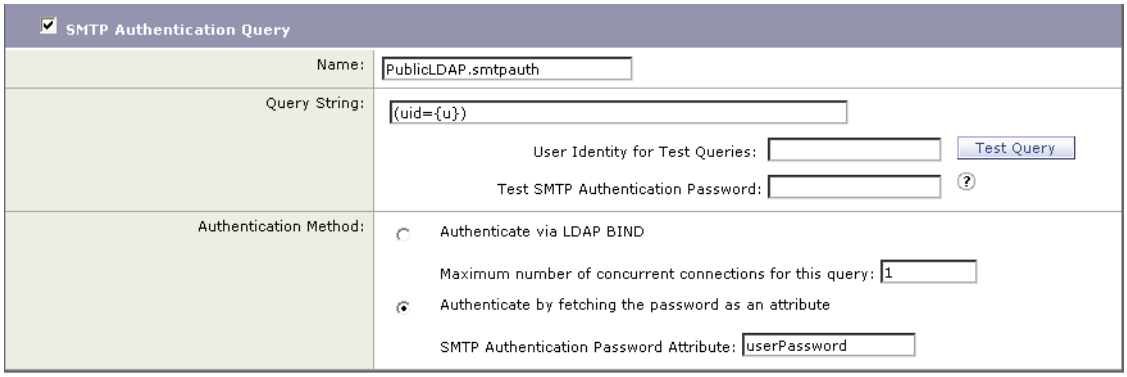
When configuring an SMTP Authentication query, you specify the following information:

Table 3-6 SMTP Auth LDAP Query Fields

Name	A name for the query.
Query String	You can select whether to authenticate via LDAP bind or by fetching the password as an attribute. **Bind**: Attempt to log into the LDAP server using the credentials supplied by the client (this is called an LDAP bind). Specify the maximum number of concurrent connections to be used by the SMTP Auth query. This number should not exceed the number specified in the LDAP server attributes above. Note, to avoid large number of session time-outs for bind authentication, increase the maximum number of concurrent connections here (typically nearly all of the connections can be assigned to SMTP Auth). A new connection is used for each bind authentication. The remainder of the connections are shared by the other LDAP query types. **Password as Attribute**: To authenticate by fetching passwords, specify the password in the SMTP Auth password attribute field below. Specify the LDAP query to use for either kind of authentication. Active Directory example query: `(&(samaccountname={u})(objectCategory=person)(objectClass=user))`
SMTP Auth Password Attribute	If you have selected "Authenticate by fetching the password as an attribute," you can specify the password attribute here.

In the following example, the System Administration > LDAP page is used to edit the LDAP configuration named "PublicLDAP" to include an SMTPAUTH query. The query string (`uid={u}`) is constructed to match against userPassword attribute.

Figure 3-17 SMTP Authentication Query



When an SMTPAUTH profile has been configured, you can specify that the listener uses that query for SMTP authentication.

SMTP Authentication via Second SMTP Server (SMTP Auth with Forwarding)

You can configure the appliance to verify the username and password that have been provided to another SMTP authenticated conversation with a different SMTP server.

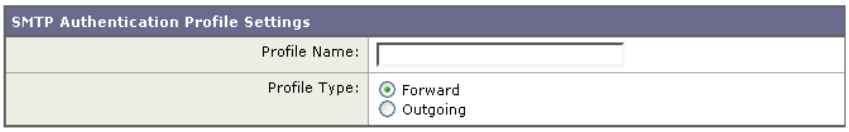
The authenticating server is not the server that transfers mail; rather, it only responds to SMTP Authentication requests. When authentication has succeeded, the SMTP transfer of mail with the dedicated mail server can proceed. This feature is sometimes referred to as "SMTP Authentication with forwarding" because only the credentials are forwarded (or "proxied") to another SMTP server for authentication.

To create an SMTP Authentication Forwarding profile:

Step 1 Click the Network > SMTP Authentication link. The SMTP Authentication page is displayed.

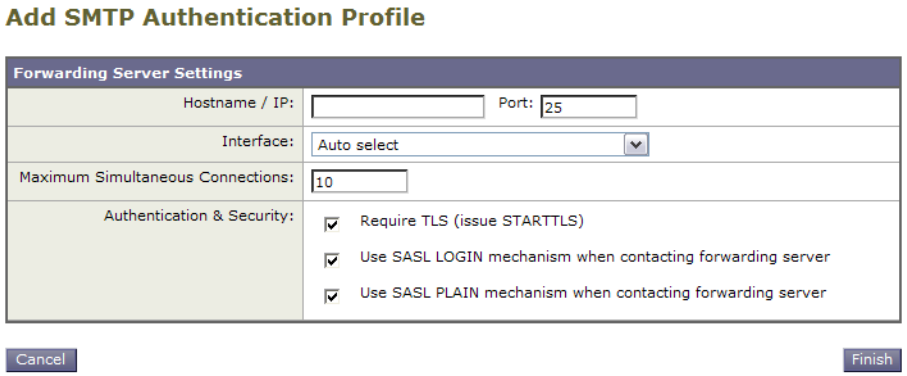
Step 2 Click the **Add Profile** link. The Add SMTP Authentication Profile: SMTP Authentication Profile Settings page is displayed. Enter a unique name for the SMTP authentication profile. Select 'Forwarding' for the Profile Type.

Figure 3-18 Selecting a Forwarding SMTP Authentication Profile



Step 3 Click the **Next** button. The Add SMTP Authentication Profile: Forwarding Server Settings page is displayed.

Figure 3-19 Adding Forwarding Server Settings



Enter the hostname/IP address and port of the forwarding server. Select a forwarding interface to use for forwarding authentication requests. Specify the number of maximum simultaneous connections. Then, you can configure whether TLS is required for connections from the appliance to the forwarding server You can also select a SASL method to use (PLAIN or LOGIN), if available. This selection is configured for each forwarding server.

Step 4 Submit and commit your changes.

After creating the authentication profile, you can enable the profile on a listener. See Enabling SMTP Authentication on a Listener, page 3-36 for more information.

SMTP Authentication with LDAP

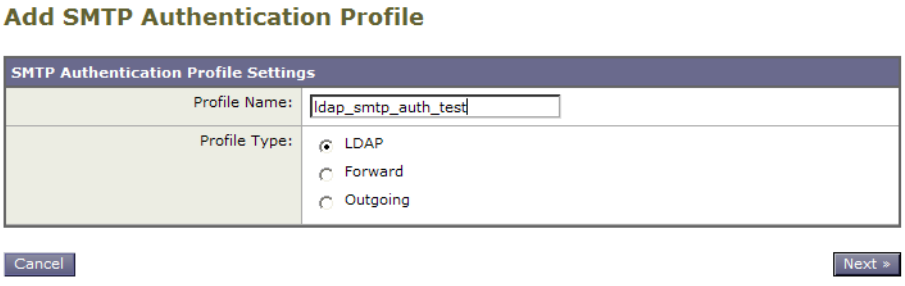
To create an LDAP-based SMTP Authentication profile, you must have previously created an SMTP Authentication query in conjunction with an LDAP server profile using the System Administration > LDAP page. You can then use this profile to create an SMTP Authentication profile. For more information about creating an LDAP profile, see Understanding LDAP Queries, page 3-2.

To configure an SMTP Authentication profile using LDAP:

Step 1 Click the Network > SMTP Authentication link. The SMTP Authentication page is displayed.

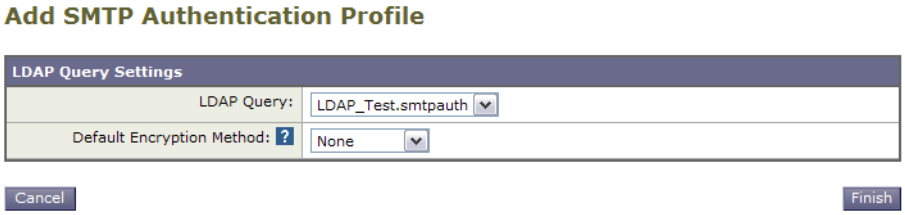
Step 2 Click the **Add Profile** link. The Add SMTP Authentication Profile: SMTP Authentication Profile Settings page is displayed. Enter a unique name for the SMTP authentication profile. Select ‘LDAP’ for the Profile Type.

Figure 3-20 Selecting a LDAP SMTP Authentication Profile



Step 3 Click the **Next** button. The Add SMTP Authentication Profile: LDAP Query Settings page is displayed.

Figure 3-21 Configuring the LDAP Query Settings for an LDAP SMTP Authentication Profile



Step 4 Select the LDAP query you would like to use for this authentication profile. Select a default encryption method from the drop-down menu. You can select from SHA, Salted SHA, Crypt, Plain, or MD5. If your LDAP servers prefix an encrypted password with the encryption type, leave 'None' selected. If your LDAP server saves the encryption type as a separate entity (OpenWave LDAP servers, for example), then select an encryption method from the menu. The default encryption setting will not be used if the LDAP query is using bind.

Step 5 Click the **Finish** button.

Step 6 Click the **Commit Changes** button, add an optional comment if necessary, and then click **Commit Changes** to finish adding the LDAP SMTP Authentication profile.

After creating the authentication profile, you can enable the profile on a listener. See Enabling SMTP Authentication on a Listener, page 3-36 for more information.

Enabling SMTP Authentication on a Listener

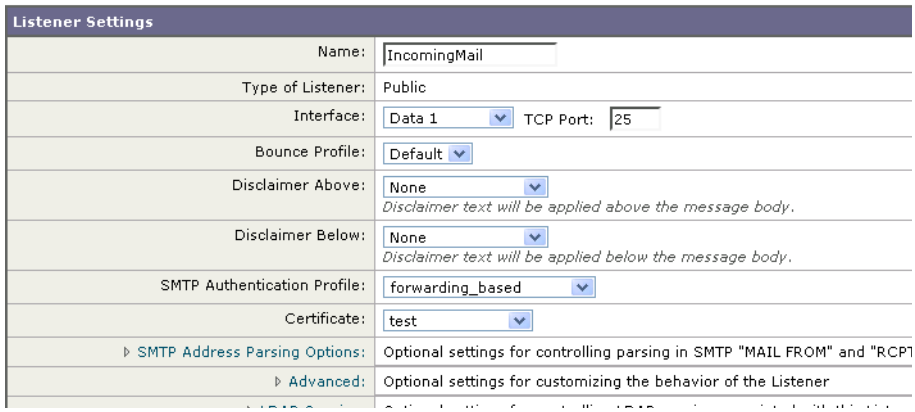
After using the Network > SMTP Authentication page to create an SMTP authentication "profile" that specifies the type of SMTP authentication you want to perform (LDAP-based or SMTP forwarding-based), you must associate that profile with a listener using the Network > Listeners page (or the `listenerconfig` command).

Note An authenticated user is granted RELAY connection behavior within their current Mail Flow Policy.

Note You may specify more than one forwarding server in a profile. SASL mechanisms CRAM-MD5 and DIGEST-MD5 are not supported between the Cisco IronPort appliance and a forwarding server.

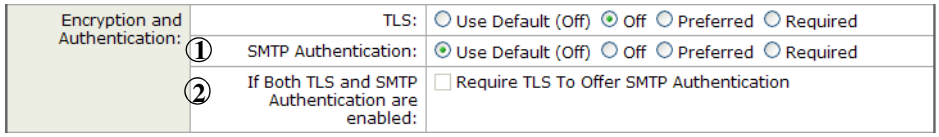
In the following example, the listener "InboundMail" is edited to use the SMTPAUTH profile configured via the Edit Listener page:

Figure 3-22 Selecting an SMTP Authentication Profile via the Edit Listener page



Once a listener is configured to use the profile, the Host Access Table default settings can be changed so that the listener allows, disallows, or requires SMTP Authentication:

Figure 3-23 Enabling SMTP Authentication on a Mail Flow Policy



Number	Description
1.	The SMTP Authentication field provides listener-level control for SMTP authentication. If you select “No,” authentication will not be enabled on the listener, regardless of any other SMTP authentication settings you configure.
2.	If “Required” is selected in the second prompt (SMTP Authentication:), no AUTH keyword will be issued until TLS is negotiated (after the client issues a second EHLO command).

SMTP Authentication and HAT Policy Settings

Because senders are grouped into the appropriate sender group before the SMTP Authentication negotiation begins, Host Access Table (HAT) settings, are not affected. When a remote mail host connects, the appliance first determines which sender group applies and imposes the Mail Policy for that sender group. For example, if a remote MTA “suspicious.com” is in your SUSPECTLIST sender group, the THROTTLE policy will be applied, regardless of the results of “suspicious.com’s” SMTPAUTH negotiation.

However, senders that do authenticate using SMTPAUTH are treated differently than “normal” senders. The connection behavior for successful SMTPAUTH sessions changes to “RELAY,” effectively bypassing the Recipient Access Table (RAT) and LDAPACCEPT. This allows the sender to relay messages through the Cisco IronPort appliance. As stated, any Rate Limiting or throttling that applies will remain in effect.

HAT Delayed Rejection

When HAT Delayed Rejection is configured, connections that would get dropped based on the HAT Sender Group and Mail Flow Policy configuration can still authenticate successfully and get the RELAY mail flow policy granted.

You can configure delayed rejection using the `listenerconfig --> setup` CLI command. This behavior is disabled by default.

The following table shows how to configure delayed rejection for HAT.

```
example.com> listenerconfig


Currently configured listeners:


1. listener1 (on main, 172.22.138.17) QMQP TCP Port 628 Private

2. listener2 (on main, 172.22.138.17) SMTP TCP Port 25 Private


Choose the operation you want to perform:

- NEW - Create a new listener.

- EDIT - Modify a listener.

- DELETE - Remove a listener.

- SETUP - Change global settings.

[]> setup


Enter the global limit for concurrent connections to be allowed across all listeners.

[300]>


[...]


By default HAT rejected connections will be closed with a banner

message at the start of the SMTP conversation. Would you like to do the rejection at the
message recipient level instead for more detailed logging of rejected mail?

[N]> y


Do you want to modify the SMTP RCPT TO reject response in this case?

[N]> y
```

```
Enter the SMTP code to use in the response. 550 is the standard code.

[550]> 551


Enter your custom SMTP response. Press Enter on a blank line to finish.


Sender rejected due to local mail policy.

Contact your mail admin for assistance.
```

Outgoing SMTP Authentication

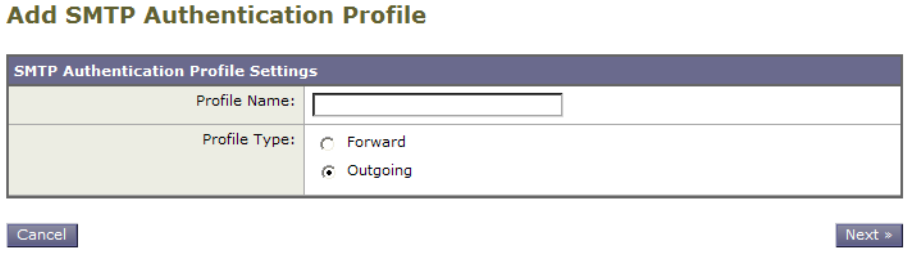
SMTP Authentication can also be used to provide validation for an outbound mail relay, using a username and password. Create an 'outgoing' SMTP authentication profile and then attach the profile to an SMTP route for the ALL domain. On each mail delivery attempt, the Cisco IronPort appliance will log on to the upstream mail relay with the necessary credentials. Only a PLAIN SASL formatted login is supported.

To use SMTP Authentication for all outgoing mail:

Step 1 Click the Network > SMTP Authentication link. The SMTP Authentication page is displayed.

Step 2 Click the **Add Profile** link. The Add SMTP Authentication Profile: SMTP Authentication Profile Settings page is displayed. Enter a unique name for the SMTP authentication profile. Select 'Outgoing' for the Profile Type. Click the **Next** button.

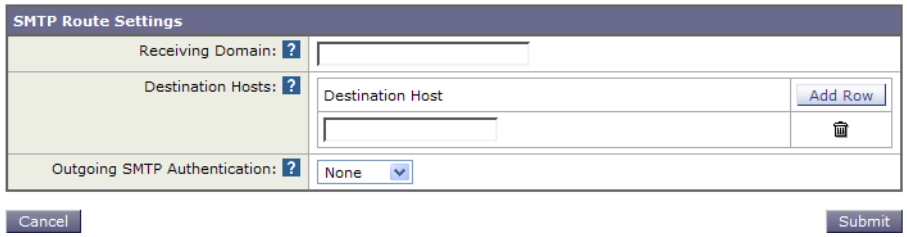
Figure 3-24 Adding a Outgoing SMTP Authentication Profile



Enter an authentication username and password for the authentication profile. Click the **Finish** button. The SMTP Authentication Profiles page is displayed with the new outgoing profile.

Step 3 Click the Network > SMTP Routes link. The SMTP Routes page is displayed.

Figure 3-25 Adding an Outgoing SMTP Route



Step 4 Click the All Other Domains link. The Edit SMTP Route page is displayed. Enter the name of the Destination Host for the SMTP route. This is the hostname of your external mail relay used to deliver outgoing mail.

Step 5 Select the outgoing SMTP authentication profile from the drop-down menu. Click the **Submit** button

Step 6 Commit your changes.

Logging and SMTP Authentication

The following events will be logged in the Cisco IronPort mail logs when the SMTP Authentication mechanism (either LDAP-based, SMTP forwarding server based, or SMTP outgoing) is configured on the appliance:

- [Informational] Successful SMTP Authentication attempts — including the user authenticated and the mechanism used. (No plaintext passwords will be logged.)
- [Informational] Unsuccessful SMTP Authentication attempts — including the user authenticated and the mechanism used.
- [Warning] Inability to connect to the authentication server — including the server name and the mechanism.
- [Warning] A time-out event when the forwarding server (talking to an upstream, injecting Cisco IronPort appliance) times out while waiting for an authentication request.

Configuring External Authentication for Users

You can configure the Cisco IronPort appliance to use an LDAP directory on your network to authenticate users by allowing them to log in with their LDAP usernames and passwords. After you configure the authentication queries for the LDAP server, enable the appliance to use external authentication on the System Administration > Users page in the GUI (or use the `userconfig` command in the CLI).

To configure external authentication for users, complete the following steps:

Step 1 **Create a query to find user accounts.** In an LDAP server profile, create a query to search for user accounts in the LDAP directory.

Step 2 **Create group membership queries.** Create a query to determine if a user is a member of a directory group.

Step 3 **Set up external authentication to use the LDAP server.** Enable the appliance to use the LDAP server for user authentication and assign user roles to the groups in the LDAP directory. For more information, see “Adding Users” in the *Cisco IronPort AsyncOS for Email Daily Management Guide*.

Note Use the Test Query button on the LDAP page (or the `ldaptest` command) to verify that your queries return the expected results. For more information, see Testing LDAP Queries, page 3-17.

User Accounts Query

To authenticate external users, AsyncOS uses a query to search for the user record in the LDAP directory and the attribute that contains the user's full name. Depending on the server type you select, AsyncOS enters a default query and a default attribute. You can choose to have your appliance deny users with expired accounts if you have attributes defined in RFC 2307 in your LDAP user records (`shadowLastChange`, `shadowMax`, and `shadowExpire`). The base DN is required for the domain level where user records reside.

Table 3-7 shows the default query string and full username attribute that AsyncOS uses when it searches for a user account on an Active Directory server.

Table 3-7 Default User Account Query String and Attribute: Active Directory

Server Type	Active Directory
Base DN	[blank] (You need to use a specific base DN to find the user records.)
Query String	`(&(objectClass=user)(sAMAccountName={u}))`
Attribute containing the user's full name	`displayName`

Table 3-8 shows the default query string and full username attribute that AsyncOS uses when it searches for a user account on an OpenLDAP server.

Table 3-8 Default User Account Query String and Attribute: OpenLDAP

Server Type	OpenLDAP
Base DN	[blank] (You need to use a specific base DN to find the user records.)
Query String	`(&(objectClass=posixAccount)(uid={u}))`
Attribute containing the user's full name	`gecos`

Group Membership Queries

AsyncOS also uses a query to determine if a user is a member of a directory group. Membership in a directory group membership determines the user's permissions within the system. When you enable external authentication on the System Administration > Users page in the GUI (or `userconfig` in the CLI), you assign user roles to the groups in your LDAP directory. User roles determine the permissions that users have in the system, and for externally authenticated users, the roles are assigned to directory groups instead of individual users. For example, you can assign users in the IT directory group the Administrator role and users in the Support directory group to the Help Desk User role.

If a user belongs to multiple LDAP groups with different user roles, AsyncOS grants the user the permissions for the most restrictive role. For example, if a user belongs to a group with Operator permissions and a group with Help Desk User permissions, AsyncOS grants the user the permissions for the Help Desk User role.

When you configure the LDAP profile to query for group membership, enter the base DN for the directory level where group records can be found, the attribute that holds the group member's username, and the attribute that contains the group name. Based on the server type that you select for your LDAP server profile, AysncOS enters default values for the username and group name attributes, as well default query strings.



Note For Active Directory servers, the default query string to determine if a user is a member of a group is `(&(objectClass=group)(member={u}))`. However, if your LDAP schema uses distinguished names in the "memberof" list instead of usernames, you can use `{dn}` instead of `{u}`.

Table 3-9 shows the default query strings and attributes that AsyncOS uses when it searches for group membership information on an Active Directory server.

Table 3-9 Default Group Membership Query Strings and Attribute: Active Directory

Server Type	Active Directory
Base DN	[blank] (You need to use a specific base DN to find the group records.)
Query string to determine if a user is a member of a group	`(&(objectClass=group)(member={u}))` **Note** If your LDAP schema uses distinguished names in the `memberOf` list instead of usernames, you can replace `{u}` with `{dn}`.
Attribute that holds each member's username (or a DN for the user's record)	`member`
Attribute that contains the group name	`cn`

Table 3-10 shows the default query strings and attributes that AsyncOS uses when it searches for group membership information on an OpenLDAP server.

Table 3-10 Default Group Membership Query Strings and Attributes: OpenLDAP

Server Type	OpenLDAP
Base DN	[blank] (You need to use a specific base DN to find the group records.)
Query string to determine if a user is a member of a group	`(&(objectClass=posixGroup)(memberUid={u}))`
Attribute that holds each member's username (or a DN for the user's record)	`memberUid`
Attribute that contains the group name	`cn`

Spam Quarantine End-User Authentication Queries

Spam quarantine end-user authentication queries validate users when they log in to the Cisco IronPort Spam Quarantine. The token {u} specifies the user (it represents the user's login name). The token {a} specifies the user's email address. The LDAP query does not strip "SMTP:" from the email address; AsyncOS strips that portion of the address.

If you want the Cisco IronPort Spam Quarantine to use an LDAP query for end-user access, check the "Designate as the active query" check box. If there is an existing active query, it is disabled. When you open the System Administration > LDAP page, an asterix (*) is displayed next to the active queries.

Based on the server type, AsyncOS uses one of the following default query strings for the end-user authentication query:

- **Active Directory:** `(sAMAccountName={u})`
- **OpenLDAP:** `(uid={u})`
- **Unknown or Other:** [Blank]

By default, the primary email attribute is `proxyAddresses` for Active Directory servers and `mail` for OpenLDAP servers. You can enter your own query and email attributes. To create the query from the CLI, use the `isqauth` subcommand of the `ldapconfig` command.

Note If you want users to log in with their full email address, use `(mail=smtp:{a})` for the Query String.

For information on enabling end-user authentication for spam quarantines, see "Configuring the Cisco IronPort Spam Quarantines Feature" in the *Cisco IronPort AsyncOS for Email Daily Management Guide*.

Sample Active Directory End-User Authentication Settings

This section shows sample settings for an Active Directory server and the end-user authentication query. This example uses password authentication for the Active Directory server, the `mail` and `proxyAddresses` email attributes, and the default query string for end-user authentication for Active Directory servers.

Table 3-11 Example LDAP Server and Spam Quarantine End-User Authentication Settings: Active Directory

Authentication Method	Use Password (Need to create a low-privilege user to bind for searching, or configure anonymous searching.)
Server Type	Active Directory
Port	3268
Base DN	[Blank]
Connection Protocol	[Blank]
Query String	`(sAMAccountName={u})`
Email Attribute(s)	`mail,proxyAddresses`

Sample OpenLDAP End-User Authentication Settings

This section shows sample settings for an OpenLDAP server and the end-user authentication query. This example uses anonymous authentication for the OpenLDAP server, the `mail` and `mailLocalAddress` email attributes, and the default query string for end-user authentication for OpenLDAP servers.

Table 3-12 Example LDAP Server and Spam Quarantine End-User Authentication Settings: OpenLDAP

Authentication Method	Anonymous
Server Type	OpenLDAP
Port	389
Base DN	[Blank] (Some older schemas will want to use a specific Base DN.)
Connection Protocol	[Blank]
Query String	`(uid={u})`
Email Attribute(s)	`mail,mailLocalAddress`

Spam Quarantine Alias Consolidation Queries

If you use spam notifications, the spam quarantine alias consolidation query consolidates the email aliases so that recipients do not receive quarantine notices for each alias. For example, a recipient might receive mail for the following email addresses: john@example.com, jsmith@example.com, and john.smith@example.com. When you use alias consolidation, the recipient receives a single spam notification at a chosen primary email address for messages sent to all of the user's aliases.

To consolidate messages to a primary email address, create a query to search for a recipient's alternate email aliases, and then enter the attribute for the recipient's primary email address in the Email Attribute field.

If you want the Cisco IronPort Spam Quarantine to use an LDAP query for spam notifications, check the "Designate as the active query" check box. If there is an existing active query, it is disabled. When you open the System Administration > LDAP page, an asterix (*) is displayed next to the active queries.

For Active Directory servers, the default query string is `(|(proxyAddresses={a})(proxyAddresses=smtp:{a}))` and the default email attribute is `mail`. For OpenLDAP servers, the default query string is `(mail={a})` and the default email attribute is `mail`. You can define your own query and email attributes, including multiple attributes separated by commas. If you enter more than one email attribute, Cisco IronPort recommends entering a unique attribute that uses a single value, such as `mail`, as the first email attribute instead of an attribute with multiple values that can change, such as `proxyAddresses`.

To create the query in the CLI, use the `isqalias` subcommand of the `ldapconfig` command.

Sample Active Directory Alias Consolidation Settings

This section shows sample settings for an Active Directory server and the alias consolidation query. This example uses anonymous authentication for the Active Directory server, a query string for alias consolidation for Active Directory servers, and the `mail` email attribute.

Table 3-13 Example LDAP Server and Spam Quarantine Alias Consolidation Settings: Active Directory

Authentication Method	Anonymous
Server Type	Active Directory
Port	3268
Base DN	[Blank]
Connection Protocol	Use SSL
Query String	`(|(mail={a})(mail=smtp:{a}))`
Email Attribute	`mail`

Sample OpenLDAP Alias Consolidation Settings

This section shows sample settings for an OpenLDAP server and the alias consolidation query. This example uses anonymous authentication for the OpenLDAP server, a query string for alias consolidation for OpenLDAP servers, and the `mail` email attribute.

Table 3-14 Example LDAP Server and Spam Quarantine Alias Consolidation Settings: OpenLDAP

Authentication Method	Anonymous
Server Type	OpenLDAP
Port	389
Base DN	[Blank] (Some older schemas will want to use a specific Base DN.)
Connection Protocol	Use SSL
Query String	`(mail={a})`
Email Attribute	`mail`

the Email Security appliance must include the complete distinguished names for the message senders when it sends DLP incident data to Enterprise Manager. To acquire the sender name for Enterprise Manager, create a user distinguished name query for your LDAP server and add the query to the listeners that send outgoing messages on your Email Security appliance. The Email Security appliance only uses this query when RSA Enterprise Manager is enabled for DLP. Otherwise, it does not appear as an option for the server profile.

Configuring AsyncOS To Work With Multiple LDAP Servers

When you configure an LDAP profile, you can configure the Cisco IronPort appliance to connect to a list of multiple LDAP servers. To use multiple LDAP servers, you must configure LDAP servers to contain the same information, use the same structure, and use the same authentication information. (third party products exist that can consolidate the records).

When you configure the Cisco IronPort appliance to connect to redundant LDAP servers, you can configure the LDAP configuration for failover or load balancing.

You can use multiple LDAP servers to achieve the following results:

- **Failover**. When you configure the LDAP profile for failover, the Cisco IronPort appliance fails over to the next LDAP server in the list if it cannot connect to the first LDAP server.
- **Load Balancing**. When you configure the LDAP profile for load balancing, the Cisco IronPort appliance distributes connections across the list of LDAP servers when it performs LDAP queries.

You can configure redundant LDAP servers from the System Administration > LDAP page or from the CLI `ldapconfig` command.

Testing Servers and Queries

Use the **Test Server(s)** button on the Add (or Edit) LDAP Server Profile page (or the `test` subcommand in the CLI) to test the connection to an LDAP server. If you use multiple LDAP servers, AsyncOS tests each server and displays individual results for each server. AsyncOS will also test the query on each LDAP server and display the individual results.

Failover

To ensure that LDAP queries are resolved, you can configure your LDAP profile for failover.

The appliance attempts to connect to the first server in the list of LDAP servers for a specified period of time. If the Cisco IronPort appliance cannot connect to the first LDAP server in the list, the appliance attempts to connect to the next LDAP server in the list. By default, the appliance always attempts to connect to the first server in the list, and it attempts to connect to each subsequent server in the order they are listed. To ensure that the Cisco IronPort appliance connects to your primary LDAP server by default, ensure that you enter it as the first server in your list of LDAP servers.

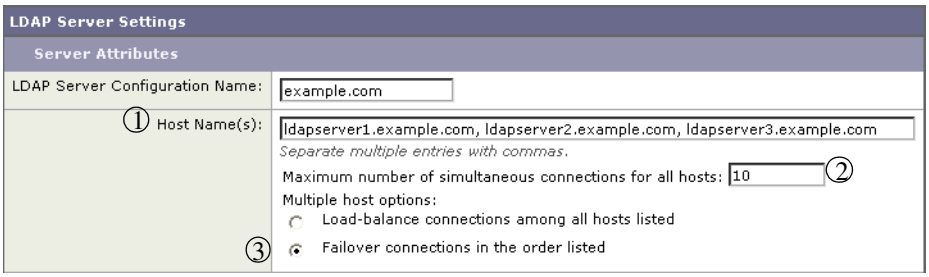
If the Cisco IronPort appliance connects to a second or subsequent LDAP server, it remains connected to that server until it reaches a timeout period. After it reaches the timeout, it attempts to reconnect to the first server in the list.

Configuring the Cisco IronPort Appliance for LDAP Failover

To configure the Cisco IronPort appliance for LDAP failover, complete the following steps in the GUI:

Step 1 From System Administration > LDAP, select the LDAP server profile you want to edit.

Step 2 From the LDAP server profile, configure the following settings:



Number	Description
1	List LDAP Servers.
2	Configure Maximum Connections.
3	Select Failover Mode.

Step 3 Configure other LDAP settings and commit the changes.

Load Balancing

To distribute LDAP connections among a group of LDAP servers, you can configure your LDAP profile for load balancing.

When you configure your LDAP profile for load balancing, the Cisco IronPort appliance distributes connections among the LDAP servers listed. If a connection fails or times out, the Cisco IronPort appliance determines which LDAP servers are available and reconnects to available servers. The Cisco IronPort appliance determines the number of simultaneous connections to establish based on the maximum number of connections you configure.

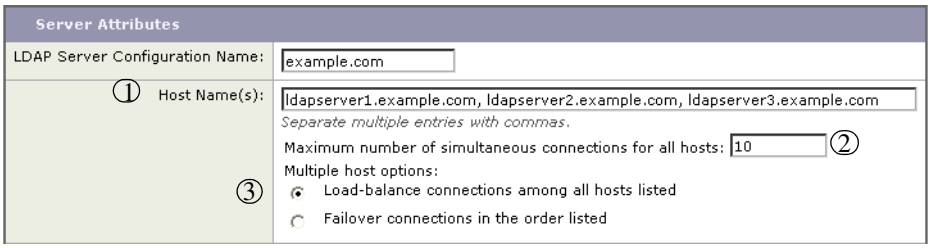
If one of the listed LDAP servers does not respond, the Cisco IronPort appliance distributes the connection load among the remaining LDAP servers.

Configuring the Cisco IronPort Appliance for Load Balancing

To configure the Cisco IronPort appliance for LDAP load balancing, complete the following steps in the GUI:

Step 1 From System Administration > LDAP, select the LDAP server profile you want to edit.

Step 2 From the LDAP server profile, configure the following settings:



Number	Description
1	List LDAP Servers
2	Configure Maximum Connections
3	Select Load Balancing Mode

Step 3 Configure other LDAP settings and commit the changes.



CHAPTER 4

Validating Recipients Using an SMTP Server

This chapter explains how to validate recipients using an SMTP server.

This chapter contains the following sections:

- SMTP Call-Ahead Recipient Validation: Overview, page 4-1
- Configuring SMTP Call-Ahead Recipient Validation, page 4-3
- Enabling the SMTP Call-Ahead Server Profile on a Public Listener, page 4-6
- Configuring LDAP Routing Query Settings, page 4-7
- SMTP Call-Ahead Query Routing, page 4-8
- Bypassing SMTP Call-Ahead Validation, page 4-9

SMTP Call-Ahead Recipient Validation: Overview

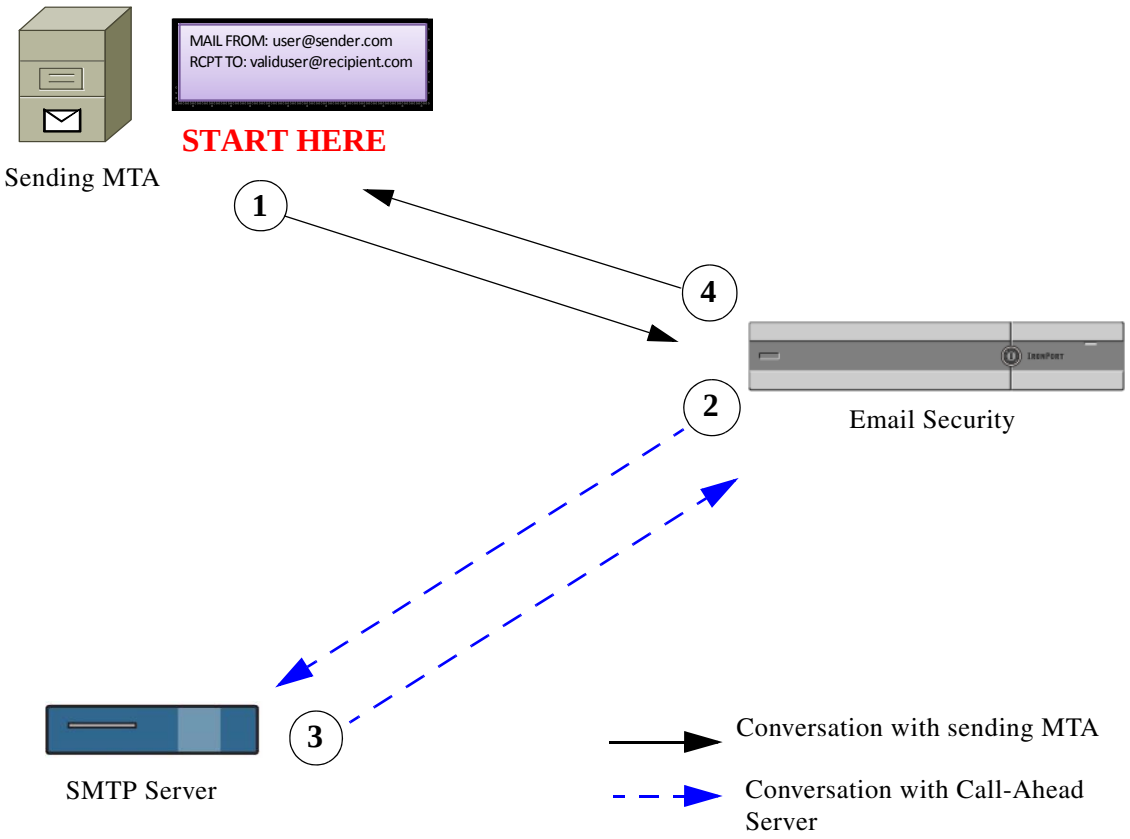
SMTP call-ahead recipient validation allows you to perform recipient validation by querying an external SMTP server prior to accepting incoming mail for the recipient. SMTP call-ahead recipient validation is useful in cases where you might want to validate users but cannot use LDAP Accept or the Recipient Access Table (RAT) for recipient validation. For example, a customer hosts mail for a large number of different mailboxes, each using a separate domain. Because of the LDAP infrastructure, there is no way to query the infrastructure to validate each of the customers in their separate domains. In this case, the customer could set up SMTP call-ahead recipient validation to allow the Email Security appliance to query the SMTP server and validate the recipient before continuing the SMTP conversation.

SMTP call-ahead recipient validation allows the Email Security appliance to save a significant amount of processing on messages for invalid recipients. In normal processing, a message for an invalid recipient must be processed through the work queue phases of the email pipeline before it can be dropped. Using the SMTP call-ahead recipient validation feature, an invalid message can be dropped or bounced during the incoming/receiving part of the email pipeline without requiring additional processing.

When you configure your Email Security appliance for SMTP call-ahead recipient validation, the Email Security appliance suspends the SMTP conversation with the sending MTA while it “calls ahead” to the SMTP server to verify the recipient. When the Cisco IronPort appliance queries the SMTP server, it returns the SMTP server’s response to the Email Security appliance, and depending on the settings you have configured, you can accept the mail or drop the connection with a code and custom response.

Figure 4-1 shows the basic workflow of the SMTP call-head validation conversation.

Figure 4-1 SMTP Call Ahead Server Conversation Workflow



1. The sending MTA initiates an SMTP conversation.
2. The Email Security appliance suspends the SMTP conversation while it sends a query to the SMTP server to verify the recipient, *validuser@recipient.com*.

   Note If SMTP routes or LDAP routing queries are configured, these routes will be used to query the SMTP server.

3. The SMTP Server returns a query response to the Email Security appliance.
4. The Email Security appliance resumes the SMTP conversation and sends a response to the sending MTA, allowing the conversation to continue or dropping the connection based on the SMTP server response (and settings you configure in the SMTP Call-Ahead profile).

Due to the order of processes in the email pipeline, if the message for a given recipient is rejected by the RAT, then the SMTP call-ahead recipient validation will not occur. For example, if you specified in the RAT that only mail for *example.com* is accepted, then mail for *recipient@domain2.com* is rejected before SMTP call-ahead recipient validation can occur.

Note If you have configured Directory Harvest Attack Prevention (DHAP) in the HAT, be aware that SMTP call-ahead server rejections are part of the number of rejections included in the maximum invalid recipients per hour that you specify. You may need to adjust this number to account for additional SMTP server rejections. For more information about DHAP, see "Configuring the Gateway to Receive Email" in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

Configuring SMTP Call-Ahead Recipient Validation

You configure SMTP call-ahead recipient validation by creating an STMP call-ahead profile and enabling that profile on a public listener. The profile defines the behavior of the SMTP call-ahead recipient validation feature — how you connect to the SMTP server, and what actions to take based on the responses of the SMTP server. You assign this profile to a public listener to enable messages received by the specified listener to be processed using SMTP call-ahead recipient validation.

To configure SMTP call-ahead recipient validation, complete the following steps:

1. **Configure a Call-Ahead Server Profile.** The Call-Ahead Server Profile specifies how to connect to the call-ahead server and how to handle the call-ahead server responses. For more information, see Configuring the Call-Ahead Server Profile, page 4-3.
2. **Enable the Call-Ahead Server Profile on a Public Listener**. Enabling the Call-Ahead Server Profile on a public listener allows the Email Security appliance to process incoming mail on that listener using SMTP call-ahead recipient validation. For more information, see Enabling the SMTP Call-Ahead Server Profile on a Public Listener, page 4-6.
3. **Configure your LDAP routing query settings**. If you use an LDAP routing query to direct mail to different hosts, you can configure a query to return SMTP call ahead server values for the SMTP call-ahead query. See Configuring LDAP Routing Query Settings, page 4-7.

Configuring the Call-Ahead Server Profile

When you configure the SMTP Call-Ahead Server Profile, you specify the settings that determine how the Email Security appliance connects with the SMTP server and how it interprets the responses sent back from the SMTP server.

To configure the SMTP Call-Ahead Server Profile:

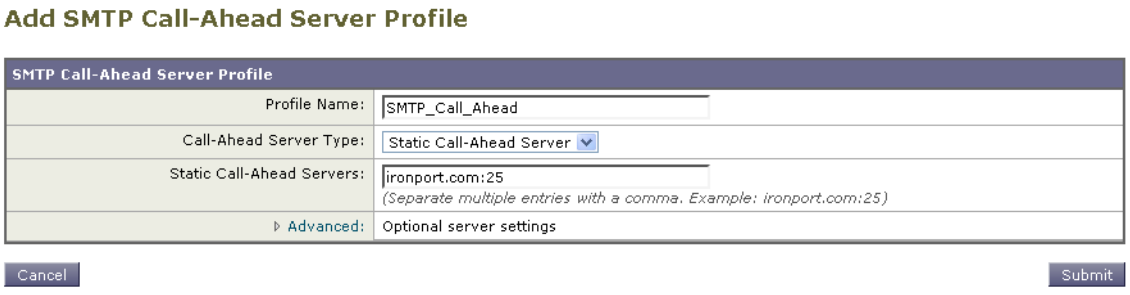
Step 1 Click **Network > SMTP Call-Ahead**.

The SMTP Call-Ahead Server Profile page opens.

Step 2 Click **Add Profile**.

The Add SMTP Call-Ahead Server Profile page opens.

Figure 4-2 Add SMTP Call-Ahead Server Profile page



Step 3 Enter the settings for profile. For more information, see Table 4-1SMTP Call-Ahead Server Profile Settings, page 4-4.

Step 4 Configure the advanced settings for the profile. For more information, see Table 4-2SMTP Call-Ahead Server Profile Advanced Settings, page 4-5.

Step 5 Submit and commit your changes.

SMTP Call-Ahead Server Profile Settings

When you configure the SMTP Call-Ahead Server Profile, you need to configure settings that determine how the Email Security appliance connects with the SMTP server.

Table 4-1 describes the SMTP Call-Ahead Server Profile basic settings.

Table 4-1 SMTP Call-Ahead Server Profile Settings

Setting	Description
Profile Name	Name of the call-ahead server profile.
Call-Ahead Server Type	Choose from one of the following methods for connecting to the call-ahead server: • **Use Delivery Host**. Select this option to specify that the host for the delivery email address is used for the SMTP call-ahead query. For example, if the mail recipient address is *recipient@example.com*, the SMTP query is executed against the SMTP server associated with *example.com*. If you have configured SMTP routes or LDAP routing queries, these routes are used to determine the SMTP server to query. For details about configuring LDAP routing queries, see Configuring LDAP Routing Query Settings, page 4-7. • **Static Call-Ahead Server**. Use this option to create a static list of call-ahead servers to query. You may want to use this option if you do not expect the names and locations of the call-ahead servers to change often. When you use this option, the Email Security appliance queries the hosts in a round-robin fashion, starting with the first static call-ahead server listed. **Note** Note that when you choose the static call-ahead server type, no SMTP routes are applied to the query. Instead an MX lookup is performed, and then an A lookup is performed on the hosts to obtain the call-ahead IP addresses for the static servers.
Static Call-Ahead Servers	If you choose to use the static call-ahead server type, enter a list of host and port combinations in this field. List the server and port using the following syntax: `ironport.com:25` Separate multiple entries with a comma.

Table 4-2 describes the SMTP Call-Ahead Server Profile advanced settings:

Table 4-2 SMTP Call-Ahead Server Profile Advanced Settings

Setting	Description
Interface	The interface used to initiate the SMTP conversation with the SMTP server. Choose to use the Management interface or Auto. When you select Auto, the Email Security appliance attempts to automatically detect an interface to use. The Cisco IronPort interface attempts to connect to the SMTP server in the following ways: • If the call-ahead server is on the same subnet as one of the configured interfaces, then the connection is initiated by the matching interface. • Any configured SMTP routes are used to route the query. • Otherwise, the interface that is on the same subnet as the default gateway are used.
MAIL FROM Address	The MAIL FROM: address to be used for the SMTP conversation with the SMTP server.
Validation Request Timeout	The number of seconds to wait for a result from the SMTP server. This timeout value is for a single recipient validation request which may involve contacting multiple call-ahead servers. See Call Ahead Server Responses, page 4-5.
Validation Failure Action	The action to be taken when a recipient validation request fails (due to a timeout, server failure, network issue, or unknown response). You can configure how you want the Email Security appliance to handle the different responses. See Call Ahead Server Responses, page 4-5.
Temporary Failure Action	The action to be taken when a recipient validation request temporarily fails (and a 4xx response is returned from the remote SMTP server). This can occur when the mailbox is full, the mailbox is not available, or the service is not available). See Call Ahead Server Responses, page 4-5.
Max. Recipients per Session	Maximum number of recipients to be validated in a single SMTP session. Specify between 1 - 25,000 sessions.
Max. Connections per Server	Maximum number of connections to a single call-ahead SMTP server. Specify between 1-100 connections.
Cache	Size of the cache for SMTP responses. Specify between 100-1,000,000 entries
Cache TTL	Time-to-live value for entries in the cache. This field defaults to 900 seconds. Specify between 60 - 86400 seconds.

Call Ahead Server Responses

The SMTP server may return the following responses:

- **2xx**: When an SMTP code starting with 2 is received from the call-ahead server, the recipient is accepted. For example, a response of 250 allows the mailing action to continue.

- **4xx**: An SMTP code starting with a 4 means that a temporary failure has occurred in processing the SMTP request. A retry may later be processed successfully. For example, a response of 451 means the requested action was aborted or there was a local error in processing.
- **5xx**: An SMTP code starting with 5 means a permanent failure in processing the SMTP request occurred. For example, a response of 550 means the requested action was not taken or the mailbox was unavailable.
- **Timeout.** If no response is returned from the call-ahead server, you can configure how long to attempt to retry before a timeout occurs.
- **Connection error**. If a connection to the call-ahead server fails, you can configure whether to accept or reject a connection for the recipient address.

Enabling the SMTP Call-Ahead Server Profile on a Public Listener

Once you create the SMTP Call-Ahead Server Profile, you need to enable it on a listener to allow the listener to validate incoming mail via the SMTP server. SMTP call-ahead functionality is only available on public listeners, as recipient validation is not necessary for private listeners.

To enable an SMTP Call-Ahead Server Profile on a listener:

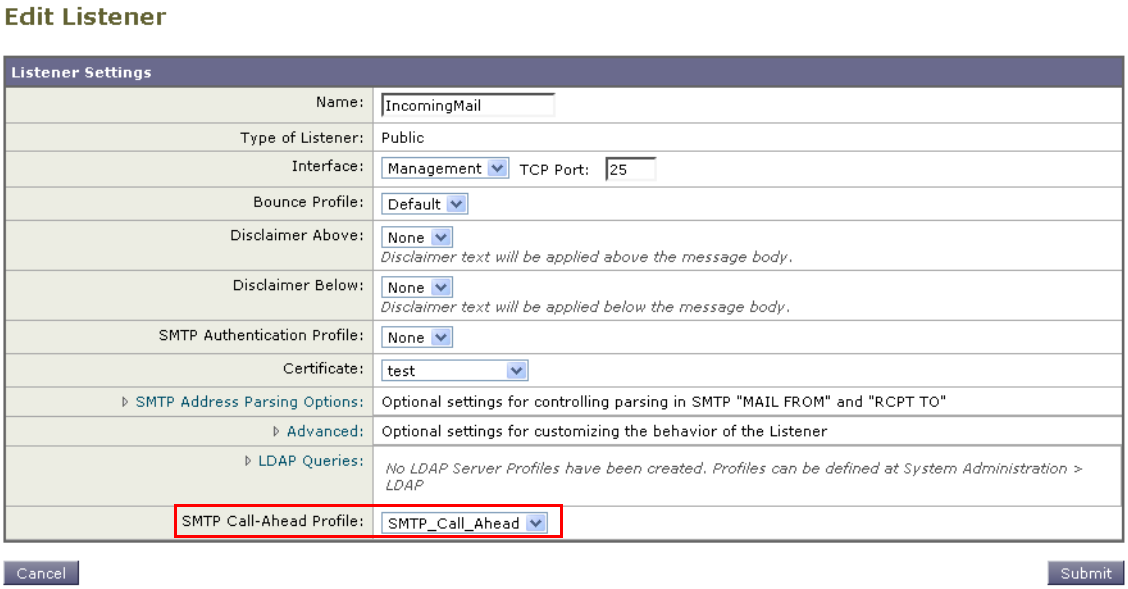
Step 1 Go to **Network > Listeners**.

Step 2 Click the name of the listener where you want to enable SMTP call-ahead functionality.

The **Edit Listener** page opens.

Step 3 In the **SMTP Call Ahead Profile** field, select the SMTP Call-Ahead profile you want to enable.

Figure 4-3 Edit Listener Page with SMTP Call-Ahead Server Profile Enabled



Step 4 Submit and commit your changes.

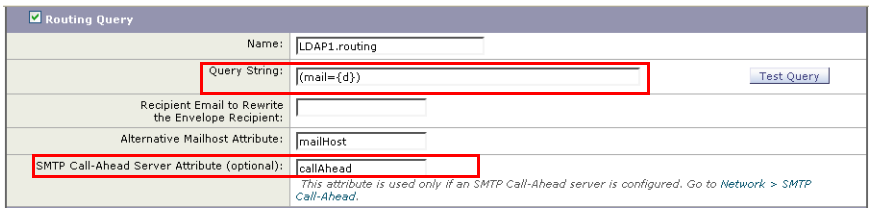
Configuring LDAP Routing Query Settings

If you use an LDAP routing query to route mail to a different mail host, AsyncOS uses the Alternate Mailhost Attribute to determine the SMTP server to query. However, there are cases where you may not want that to occur. For example, in the following schema, note that the mail host attribute (`mailHost`) has a different SMTP address than the servers listed in the call-ahead SMTP server attribute (`callAhead`):

```
dn: mail=cisco.com, ou=domains
mail: cisco.com
mailHost: smtp.mydomain.com
policy: ASAV
callAhead: smtp2.mydomain.com,smtp3.mydomain.com:9025
```

In this case, you can use the **SMTP Call-Ahead** field to create a routing query that directs the SMTP call-ahead query to the servers listed in the `callAhead` attribute. For example, you might create a routing query with the following attributes:

Figure 4-4 LDAP Routing Query Configured for SMTP Call-Ahead:



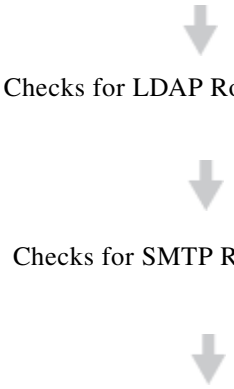
In this query, the `{d}` represents the domain part of the recipient address, and the SMTP Call-Ahead Server Attribute returns the values for the call-ahead servers and the port that should be used for the query: smtp2.mydomain.com, smtp3.mydomain.com on port 9025.

Note This example shows just one way to configure a query that enables you to use the LDAP routing query to direct SMTP call-ahead queries to the correct SMTP servers. You are not required to use the query string or specific LDAP attributes described in this example.

SMTP Call-Ahead Query Routing

When routing an SMTP call-ahead query, AsyncOS checks for information in the following order:

Figure 4-5 SMTP Call Ahead Query Routing Workflow



If there is no LDAP routing query or no SMTP Routes configured for the domain, the result of preceding state is passed to next stage. In any case where there is no SMTP Route present, a DNS lookup is performed.

When you use an LDAP Routing query for an SMTP call-ahead query and you also have SMTP routes configured, the routing behavior depends upon the values returned by the routing query.

- If the LDAP routing query returns a single hostname without a port, the SMTP call-ahead query applies SMTP routes. If the SMTP routes only lists the destination host as the hostname, a DNS lookup is performed to obtain the IP address of the SMTP server.
- If the LDAP routing query returns a single hostname with a port, the SMTP route is used, but the port returned by the LDAP query is used over any ports specified in SMTP routes. If the SMTP routes only lists the destination host as the hostname, a DNS lookup is performed to obtain the IP address of the SMTP server.
- If the LDAP routing query returns multiple hosts with or without ports, SMTP routes are applied, but the ports returned by the LDAP routing query are used over those present in SMTP routes. If the SMTP routes only lists the destination host as the hostname, a DNS lookup is performed to obtain the IP address of the SMTP server.

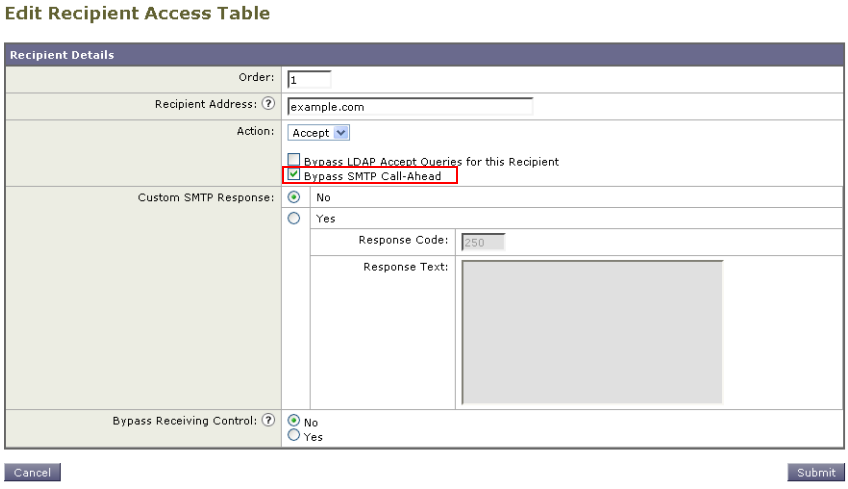
Bypassing SMTP Call-Ahead Validation

You may want to enable SMTP call-ahead validation on a listener but skip the SMTP call-ahead validation for certain users or groups of users.

You may want to skip SMTP call-ahead validation for recipients for whom mail should not be delayed during SMTP call-ahead queries. For example, you could add a RAT entry for a customer service alias that you know is valid and will likely need immediate attention.

To configure bypassing SMTP call-ahead validation via the GUI, select **Bypass SMTP Call-Ahead** when you add or edit the RAT entry.

Figure 4-6 Bypassing SMTP Call-Ahead:





CHAPTER 5

Email Authentication

Cisco IronPort AsyncOS supports several forms of email authentication, including Sender Policy Framework (SPF), Sender ID Framework (SIDF), DomainKeys and DomainKeys Identified Mail (DKIM).

DomainKeys and DKIM verify the authenticity of email based on a signing key used by the sender. SPF and SIDF are methods for verifying the authenticity of email based on DNS TXT records. SPF and SIDF allow the owner of an Internet domain to use a special format of DNS records to designate which machines are authorized to send email for that domain.

This chapter contains the following sections:

- Email Authentication Overview, page 5-1
- DomainKeys and DKIM Authentication: Overview, page 5-2
- Configuring DomainKeys and DKIM Signing, page 5-4
- Configuring DKIM Verification, page 5-15
- Overview of SPF and SIDF Verification, page 5-22
- Working with SPF on a Cisco IronPort Email Security Appliance, page 5-23
- Enabling SPF and SIDF, page 5-24
- Determining the Action to Take for SPF/SIDF Verified Mail, page 5-31
- Testing the SPF/SIDF Results, page 5-34

Email Authentication Overview

Cisco IronPort AsyncOS supports several forms of email authentication to prevent email forgery. To verify incoming mail, AsyncOS supports Sender Policy Framework (SPF), Sender ID Framework (SIDF), and DomainKeys Identified Mail (DKIM). To sign outgoing mail, AsyncOS supports DomainKeys and DKIM.

With DomainKeys or DKIM email authentication, the sender signs the email using public key cryptography. The verified domain can then be used to detect forgeries by comparing it with the domain in the From: (or Sender:) header of the email. The current version of AsyncOS supports email signing for DomainKeys, and it supports both email signing and verification for DKIM. For more information about DomainKeys and DKIM, see DomainKeys and DKIM Authentication: Overview, page 5-2.

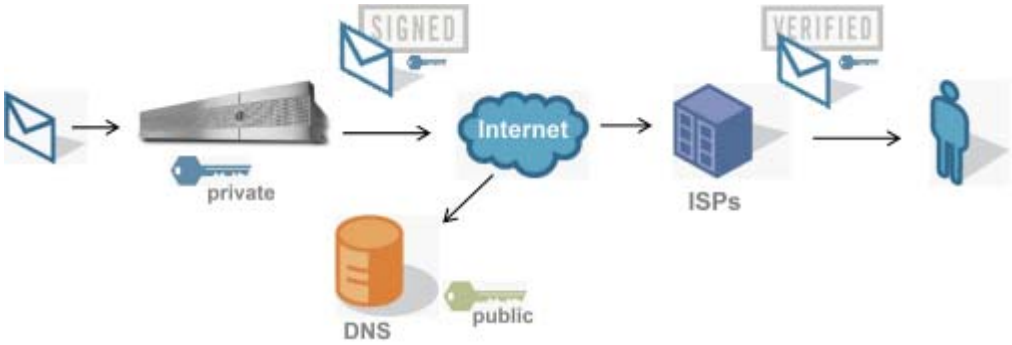
SPF and SIDF email authentication allow the owners of Internet domains to use a special format of DNS TXT records to specify which machines are authorized to transmit email for their domains. Compliant mail receivers then use the published SPF records to test the authorization of the sending Mail Transfer Agent's identity during a mail transaction. For more information about SPF and SIDF, see Overview of SPF and SIDF Verification, page 5-22.

DomainKeys and DKIM Authentication: Overview

AsyncOS supports DomainKeys and DKIM authentication to prevent email forgery. DomainKeys and DKIM are mechanisms used to verify that the source of the email and the contents of the message were not altered during transit. DKIM is an enhanced protocol that combines DomainKeys specification with aspects of Identified Internet Mail to create an enhanced protocol called DomainKeys Identified Mail (DKIM). DomainKeys and DKIM consist of two main parts: signing and verification. The current version of AsyncOS supports the "signing" half of the process for DomainKeys, and it supports both signing and verification for DKIM. You can also enable bounce and delay messages to use DomainKeys and DKIM signing.

When you use DomainKeys or DKIM authentication, the sender signs the email using public key cryptography. The verified domain can then be used to detect forgeries by comparing it with the domain in the From: (or Sender:) header of the email.

Figure 5-1 Authentication Work Flow



Step 1 Administrator (domain owner) publishes a public key into the DNS name space.

Step 2 Administrator loads a private key in the outbound Mail Transfer Agent (MTA).

Step 3 Email submitted by an authorized user of that domain is digitally signed with the respective private key. The signature is inserted in the email as a DomainKey or DKIM signature header and the email is transmitted.

Step 4 Receiving MTA extracts the DomainKeys or DKIM signature from the header and the claimed sending domain (via the Sender: or From: header) from the email. The public key is retrieved from the claimed signing domain which is extracted from DomainKeys or DKIM signature header fields.

Step 5 The public key is used to determine whether the DomainKeys or DKIM signature was generated with the appropriate private key.

To test your outgoing DomainKeys signatures, you can use a Yahoo! or Gmail address, as these services are free and provide validation on incoming messages that are DomainKeys signed.

DomainKeys and DKIM Signing in AsyncOS

DomainKeys and DKIM signing in AsyncOS is implemented via domain profiles and enabled via a mail flow policy (typically, the outgoing "relay" policy). For more information, see the "Configuring the Gateway to Receive Mail" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide*. Signing the message is the last action performed by the appliance before the message is sent.

Domain profiles associate a domain with domain key information (signing key and related information). As email is sent via a mail flow policy on the Cisco IronPort appliance, sender email addresses that match any domain profile are DomainKeys signed with the signing key specified in the domain profile. If you enable both DKIM and DomainKeys signing, the DKIM signature is used. You implement DomainKeys and DKIM profiles via the `domainkeysconfig` CLI command or via the Mail Policies > Domain Profiles and the Mail Policies > Signing Keys pages in the GUI.

DomainKeys and DKIM signing works like this: a domain owner generates two keys — a public key stored in the public DNS (a DNS TXT record associated with that domain) and a private key that is stored on the appliance is used to sign mail that is sent (mail that originates) from that domain.

As messages are received on a listener used to send messages (outbound), the Cisco IronPort appliance checks to see if any domain profiles exist. If there are domain profiles created on the appliance (and implemented for the mail flow policy), the message is scanned for a valid Sender: or From: address. If both are present, the Sender: is used for DomainKeys. The From: address is always used for DKIM signing. Otherwise, the first From: address is used. If a valid address is not found, the message is not signed and the event is logged in the mail_logs.



Note If you create both a DomainKey and DKIM profile (and enable signing on a mail flow policy), AsyncOS signs outgoing messages with both a DomainKeys and DKIM signature.

If a valid sending address is found, the sending address is matched against the existing domain profiles. If a match is found, the message is signed. If not, the message is sent without signing. If the message has an existing DomainKeys (a "DomainKey-Signature:" header) the message is only signed if a new sender address has been added after the original signing. If a message has an existing DKIM signature, a new DKIM signature is added to the message.

AsyncOS provides a mechanism for signing email based on domain as well as a way to manage (create new or input existing) signing keys.

The configuration descriptions in this document represent the most common uses for signing and verification. You can also enable DomainKeys and DKIM signing on a mail flow policy for inbound email, or enable DKIM verification on a mail flow policy for outbound email.



Note When you configure domain profiles and signing keys in a clustered environment, note that the Domain Key Profile settings and Signing Key settings are linked. Therefore, if you copy, move or delete a signing key, the same action is taken on the related profile.

Configuring DomainKeys and DKIM Signing

Signing Keys

A signing key is the private key stored on the Cisco IronPort appliance. When creating a signing key, you specify a key size. Larger key sizes are more secure; however, larger keys also can impact performance. The Cisco IronPort appliance supports keys from 512 bits up to 2048 bits. The 768 - 1024 bit key sizes are considered secure and used by most senders today. Keys based on larger key sizes can impact performance and are not supported above 2048 bits. For more information about creating signing keys, see Creating New Signing Keys, page 5-11.

If you are entering an existing key, simply paste it into the form. Another way to use existing signing keys is to import the key as a text file. For more information about adding existing signing keys, see Importing or Entering Existing Signing Keys, page 5-12.

Once a key is entered, it is available for use in domain profiles, and will appear in the Signing Key list in the domain profile:

Figure 5-2 Add Domain Profile Page (DomainKeys)— Signing Keys



Exporting and Importing Signing Keys

You can export your signing keys to a text file on the Cisco IronPort appliance. When you export keys, all of the keys currently existing on the appliance are put into a text file. For more information about exporting keys, see Exporting Signing Keys, page 5-11.

You can import keys that have been exported as well.

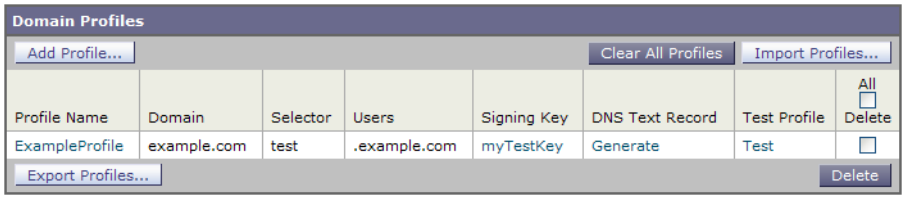
Note Importing keys causes all of the current keys on the appliance to be replaced.

For more information, see Importing or Entering Existing Signing Keys, page 5-12.

Public Keys

Once you have associated a signing key with a domain profile, you can create DNS text record which contains your public key. You do this via the Generate link in the DNS Text Record column in the domain profile listing (or via `domainkeysconfig -> profiles -> dnstxt` in the CLI):

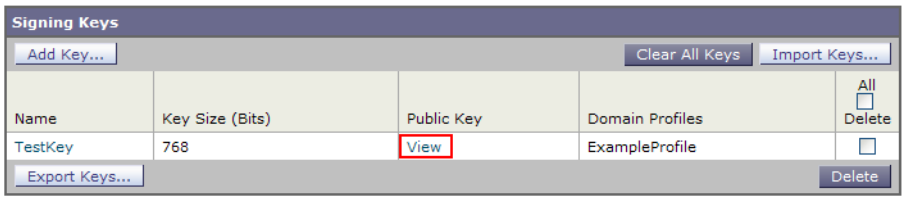
Figure 5-3 Generate DNS Text Record Link on Domain Profiles Page



For more information about generating a DNS Text Record, see Generating a DNS Text Record, page 5-12.

You can also view the public key via the View link on the Signing Keys page:

Figure 5-4 View Public Key Link on Signing Keys Page



Domain Profiles

A domain profile associates a sender domain with a signing key, along with some other information needed for signing. A domain profile consists of the following information:

- A name for the domain profile.
- A domain name (the domain to be included in the "d=" header).
- A selector (a selector is used to form the query for the public key. In the DNS query type, this value is prepended to the "_domainkey." namespace of the sending domain).
- A canonicalization method (the method by which the headers and content are prepared for presentation to the signing algorithm). AsyncOS supports both "simple" and "nofws" for DomainKeys and "relaxed" and "simple" for DKIM.
- A signing key (see Signing Keys, page 5-4 for more information).
- A list of headers and the body length to sign (DKIM only).
- A list of tags you want to include in the signature's header (DKIM only). These tags store the following information:
  - The identity of the user or agent (e.g., a mailing list manager) on whose behalf the message is signed.
  - A comma-separated list of query methods used to retrieve the public key.
  - The timestamp of when the signature was created.

  - The expiration time of the signature, in seconds.
  - A vertical bar-separated (i.e., |) list of header fields present when the message was signed.
- The tags you want to include in the signature (DKIM only).
- A list of Profile Users (addresses allowed to use the domain profile for signing).

Note The domain in the addresses specified in the profile users must match the domain specified in the Domain field.

You can search through all of your existing domain profiles for a specific term. See Searching Domain Profiles, page 5-14 for more information.

You can also choose whether to sign system-generated messages with DKIM signatures. See Signing System-Generated Messages, page 5-15 for more information.

Exporting and Importing Domain Profiles

You can export your existing domain profiles to a text file on the Cisco IronPort appliance. When you export the domain profiles, all of the profiles existing on the appliance are put into a single text file. See Exporting Domain Profiles, page 5-14.

You can import domain profiles that you have previously exported. Importing domain profiles causes all of the current domain profiles on the machine to be replaced. See Importing Domain Profiles, page 5-14.

Enabling Signing for Outgoing Mail

DomainKeys and DKIM signing is enabled on mail flow policies for outbound mail. For more information, see the "Configuring the Gateway to Receive Mail" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

To enable signing on an outgoing mail flow policy:

Step 1 On the Mail Flow Policies page (from the Mail Policies menu), click on the RELAYED mail flow policy (outgoing).

Step 2 From the Security Features section, enable DomainKeys/DKIM Signing by selecting On.

Figure 5-5 Enabling DomainKeys/DKIM Signing



Step 3 Submit and commit your changes.

Enabling Signing for Bounce and Delay Messages

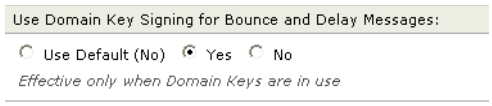
In addition to signing outbound messages, you may want to sign bounce and delay messages. You may want to do this to alert recipients that the bounce and delay messages they receive from your company are legitimate. To enable DomainKeys and DKIM signing for bounce and delay messages, you enable DomainKeys/DKIM signing for the bounce profile associated with the public listener.

To enable signing for bounce and delay messages:

Step 1 On the bounce profile associated with the public listener where you will send signed outbound messages, go to Hard Bounce and Delay Warning Messages.

Step 2 Enable "Use Domain Key Signing for Bounce and Delay Messages":

Figure 5-6 Enabling Signing For Bounce and Warning Messages



Note You must have completed all steps listed in Configuring DomainKeys/DKIM Signing (GUI), page 5-7 to sign bounced and delay messages.

Note The From: address in the domain profile must match the address used for the bounce return address. To ensure these addresses match, you can configure a return address for the bounce profile (System Administration > Return Addresses), and then use the same name in the Profile Users list in the domain profile. For example, you would configure a return address of MAILER-DAEMON@example.com for the bounce return address, and add MAILER-DAEMON@example.com as a profile user in the domain profile.

Configuring DomainKeys/DKIM Signing (GUI)

The basic workflow for DomainKeys/DKIM signing in AsyncOS is:

Step 1 Create a new or import an existing private key. For information on creating or importing signing keys, see Signing Keys, page 5-4.

Step 2 Create a domain profile and associate the key with the domain profile. For information on creating a domain profile, see Domain Profiles, page 5-5.

Step 3 Create the DNS text record. For information about creating the DNS text record, see Generating a DNS Text Record, page 5-12.

Step 4 If you have not already done so, enable DomainKeys/DKIM signing on a mail flow policy for outbound mail (see Enabling Signing for Outgoing Mail, page 5-6).

Step 5 Optionally, enable DomainKeys/DKIM signing for bounced and delay messages. For information about enabling signing for bounce and delay messages, see Enabling Signing for Bounce and Delay Messages, page 5-6.

Step 6 Send email. Mail sent from a domain that matches a domain profile will be DomainKeys/DKIM signed. In addition, bounce or delay messages will be signed if you configured signing for bounce and delay messages.

Note If you create both a DomainKey and DKIM profile (and enable signing on a mail flow policy), AsyncOS signs outgoing messages with both a DomainKeys and DKIM signature.

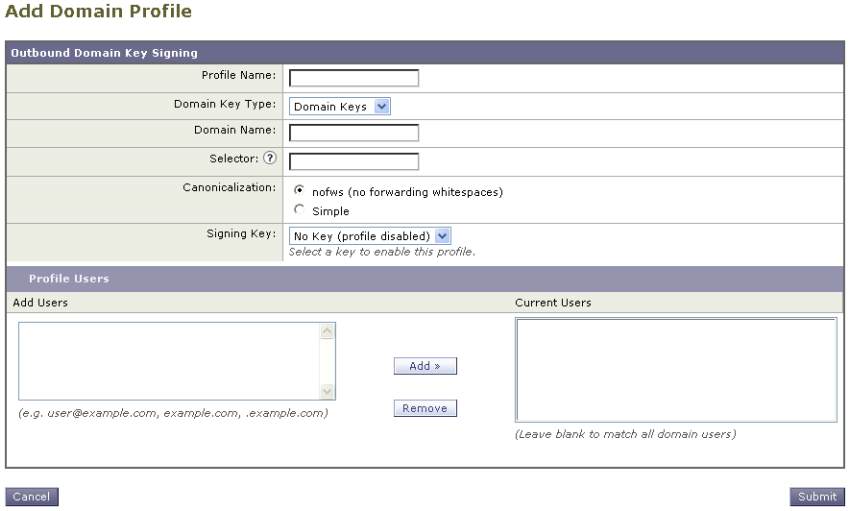
Creating Domain Profiles for DomainKeys Signing

To create a new domain profile for DomainKeys signing:

Step 1 Click **Add Profile** on the Domain Profiles page.

Step 2 Enter a name for the profile, and the Domain Key type (DomainKeys). After you select the Domain Keys type, the Add Domain Profile page is displayed.

Figure 5-7 Add Domain Profile Page (DomainKeys)



Step 3 Enter the domain name.

Step 4 Enter a selector. Selectors are arbitrary names prepended to the "_domainkey" namespace, used to help support multiple concurrent public keys per sending domain. A selector value and length must be legal in the DNS namespace and in email headers with the additional provision that they cannot contain a semicolon.

Step 5 Select the canonicalization (no forwarding whitespaces or simple).

Step 6 Select a signing key (If you have already created a signing key, otherwise, skip to the next step). You must create (or import) at least one signing key in order to have signing keys to choose from in the list. See Creating New Signing Keys, page 5-11.

Step 7 Enter users (email addresses, hosts, etc.) that will use the domain profile for signing.

Step 8 Click **Submit**.

Step 9 Click the **Commit Changes** button, add an optional comment if necessary, and then click **Commit Changes** to finish adding the new domain profile.

Step 10 At this point (if you have not already) you should enable DomainKeys/DKIM signing on an outgoing mail flow policy (see Enabling Signing for Outgoing Mail, page 5-6).

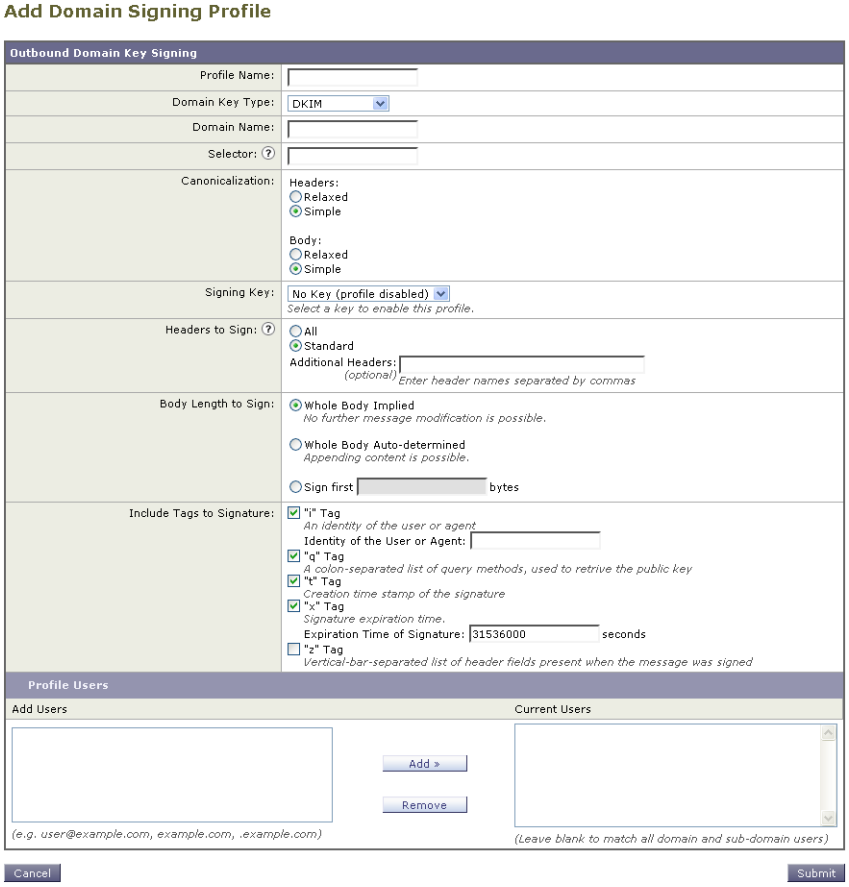
Creating Domain Profiles for DKIM Signing

To create a new domain profile for DKIM signing:

Step 1 Click **Add Profile** on the Domain Profiles page.

Step 2 Enter a name for the profile, and the Domain Key type (DKIM). After you select the Domain Key type, the Add Domain Profile page is displayed.

Figure 5-8 Add Domain Profile Page (DKIM)



Step 3 Enter the domain name.

Step 4 Enter a selector. Selectors are arbitrary names prepended to the "_domainkey." namespace, used to help support multiple concurrent public keys per sending domain. A selector value and length must be legal in the DNS namespace and in email headers with the additional provision that they cannot contain a semicolon.

Step 5 Select the canonicalization for the header. Choose from the following options:

- **Relaxed**. The “relaxed” header canonicalization algorithm performs the following: header names are changed to lowercase, headers are unfolded, linear white spaces are reduced to a single space, leading and trailing spaces are stripped.
- **Simple**. No changes to headers are made.

Step 6 Select the canonicalization for the body. Choose from the following options:

- **Relaxed**. The “relaxed” header canonicalization algorithm performs the following: empty lines are stripped at the end of the body, white spaces are reduced to a single space within lines, and trailing white spaces are stripped in lines.
- **Simple.** Empty lines at the end of the body are stripped.

Step 7 Select a signing key (If you have already created a signing key, otherwise, skip to the next step). You must create (or import) at least one signing key in order to have signing keys to choose from in the list. See Creating New Signing Keys, page 5-11.

Step 8 Select the list of headers to sign. You can select from the following headers:

- **All**. AsyncOS signs all the headers present at the time of signature. You may want to sign all headers if you do not expect headers to be added or removed in transit.
- **Standard**. You may want to select the standard headers if you expect that headers may be added or removed in transit. AsyncOS signs only the following standard headers (if the header is not present in the message, the DKIM signature indicates a null value for the header):
  - From
  - Sender, Reply To-
  - Subject
  - Date, Message-ID
  - To, Cc
  - MIME-Version
  - Content-Type, Content-Transfer-Encoding, Content-ID, Content-Description
  - Resent-Date, Resent-From, Resent-Sender, Resent-To, Resent-cc, Resent-Message-ID
  - In-Reply-To, References
  - List-Id, List-Help, List-Unsubscribe, LIst-Subscribe, List-Post, List-Owner, List-Archive

Note When you select “Standard”, you can add additional headers to sign.

Step 9 Specify how to sign the message body. You can choose to sign the message body, and/or how many bytes to sign. Select one of the following options:

- **Whole Body Implied**. Do not use the “l=” tag to determine body length. The entire message is signed and no changes are allowed.
- **Whole Body Auto-determined**. The entire message body is signed, and appending some additional data to the end of body is allowed during transit.
- **Sign first _ bytes.** Sign the message body up to the specified number of bytes.

Step 10 Select the tags you want to include in the message signature’s header field. The information stored in these tags are used for message signature verification. Select one or more of the following options:

- **“i” Tag**. The identity of the user or agent (e.g., a mailing list manager) on behalf of which this message is signed. Enter the domain name prepended with the `@` symbol, such as the domain `@example.com`.
- **“q” Tag**. A colon-separated list of query methods used to retrieve the public key. Currently, the only valid value is dns/txt.
- **“t” Tag**. A timestamp for when the signature was created.
- **“x” Tag**. The absolute date and time when the signature expires. Specify an expiration time (in seconds) for the signature. The default is `31536000` seconds.
- **“z” Tag**. A vertical bar-separated (i.e., `|`) list of header fields present when the message was signed. This includes the names of the header fields and their values. For example:

```
z=From:admin@example.come|To:joe@example.com|
Subject:test%20message|Date:Date:August%2026,%202011%205:30:02%20PM%20-0700
```

Step 11 Enter users (email addresses, hosts, etc.) that will use the domain profile for signing.

Note When you create domain profiles, be aware that a hierarchy is used in determining the profile to associate with a particular user. For example, you create a profile for example.com and another profile for joe@example.com. When mail is sent from joe@example.com, the profile for joe@example.com is used. However, when mail is sent from adam@example.com, the profile for example.com is used.

Step 12 Submit and commit your changes.

Step 13 At this point (if you have not already) you should enable DomainKeys/DKIM signing on an outgoing mail flow policy (see Enabling Signing for Outgoing Mail, page 5-6).

Note If you create both a DomainKeys and DKIM profile, AsyncOS performs both DomainKeys and DKIM signing on outgoing mail.

Creating New Signing Keys

To create a new signing key:

Step 1 Click **Add Key** on the Mail Policies > Signing Keys page. The Add Key page is displayed.

Step 2 Enter a name for the key.

Step 3 Click **Generate** and Select a key size.

Larger key sizes are more secure; however, larger keys can have an impact on performance. Cisco recommends a key size of 768 bits, which should provide a good balance between security and performance.

Step 4 Click **Submit**. The key is generated.

Step 5 Click the **Commit Changes** button, add an optional comment if necessary, and then click **Commit Changes** to finish adding the new signing key.

Note If you have not done so already, you may need to edit your domain profile to assign the key.

Exporting Signing Keys

When you export signing keys, all of the keys currently existing on your Cisco IronPort appliance are exported together in a single text file. To export signing keys:

Step 1 Click **Export Keys** on the Signing Keys page. The Export Signing Keys page is displayed:

Figure 5-9 Export Signing Keys Page



Step 2 Enter a name for the file and click **Submit**.

Importing or Entering Existing Signing Keys

To enter an existing key:

Step 1 Click **Add Key** on the Mail Policies > Signing Keys page. The Add Key page is displayed.

Step 2 Paste the key into the Paste Key field (must be PEM-formatted and must be RSA keys only).

Step 3 Submit and commit your changes.

To import keys from an existing export file (see Exporting Signing Keys, page 5-11):

Step 1 Click **Import Keys** on the Mail Policies > Signing Keys page. The Import Key page is displayed.

Step 2 Select the file that contains the exported signing keys.

Step 3 Click **Submit**. You are warned that importing will replace all existing signing keys. All of the keys in the text file are imported.

Step 4 Click **Import**.

Deleting Signing Keys

Step 1 To remove specific keys from the list of signing keys:

Step 2 On the Signing Keys page, mark the checkbox to the right of each signing key to remove.

Step 3 Click **Delete**.

Step 4 Confirm the deletion.

To remove all of the currently configured signing keys:

Step 1 Click **Clear All Keys** on the Signing Keys page.

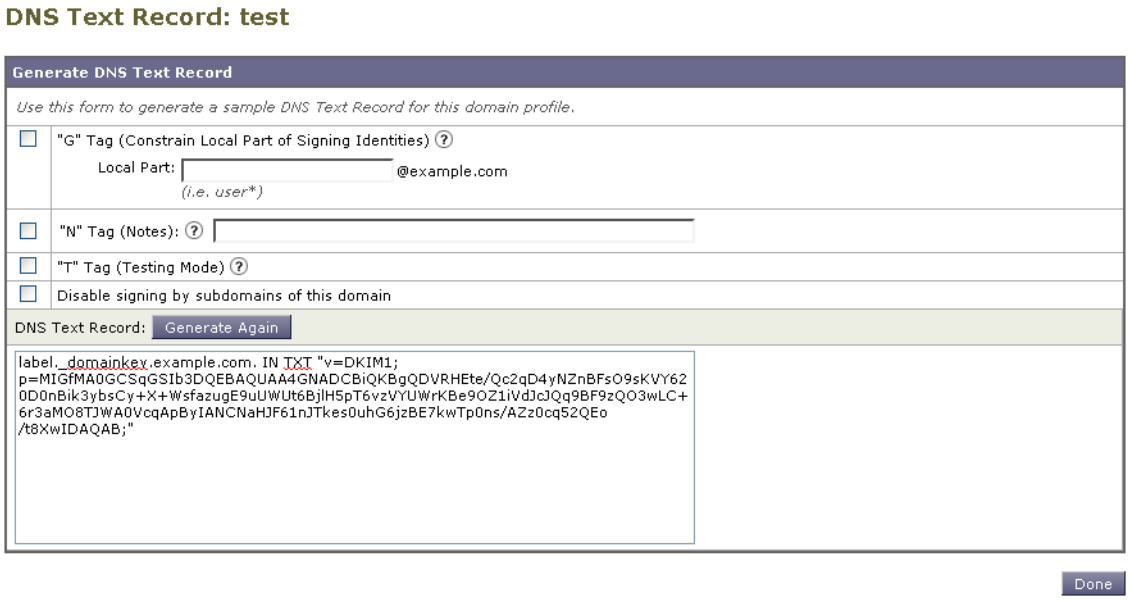
Step 2 You are prompted to confirm the deletion.

Generating a DNS Text Record

To generate a DNS text record:

Step 1 Click the **Generate** link in the DNS Text Record column for the corresponding domain profile. The Generate DNS Text Record page is displayed:

Figure 5-10 The DNS Text Record Page



Step 2 Mark the checkbox for the attributes you wish to include in the DNS text record.

Step 3 Click **Generate Again** to re-generate the key with any changes you have made.

Step 4 The DNS text record is displayed in the text field (where you can now copy it).

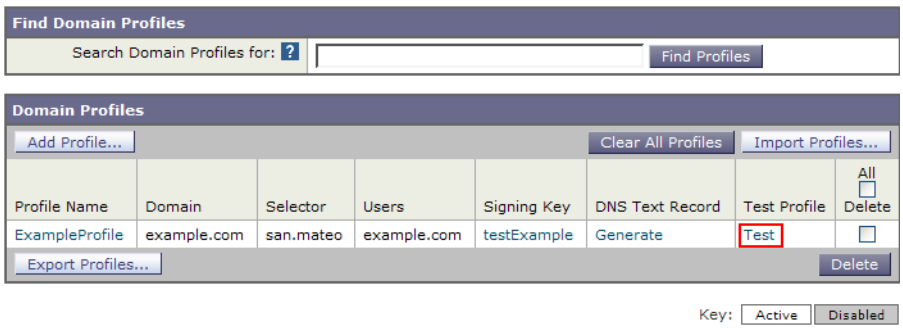
Step 5 Click **Done**.

Testing Domain Profiles

Once you have created a signing key, associated it with a domain profile, and generated and inserted the DNS text into your authorized DNS, you can test your domain profile. To do so:

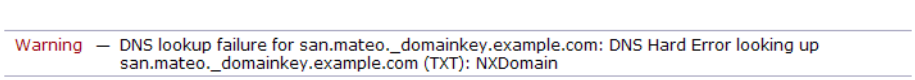
Step 1 Click Test on the Domain Profiles page:

Figure 5-11 Testing a Domain Profile



Step 2 A message is displayed at the top of the page, indicating success or failure. If the test fails, a warning message in displayed, including the error text:

Figure 5-12 An Unsuccessful Domain Profile Test



Exporting Domain Profiles

When you export domain profiles, all of the domain profiles currently existing on your Cisco IronPort appliance are exported together in a single text file. To export domain profiles:

Step 1 Click **Export Domain Profiles** on the Domain Profiles page. The Export Domain profiles page is displayed.

Step 2 Enter a name for the file and click **Submit**.

Importing Domain Profiles

To import domain profiles from an existing export file:

Step 1 Click **Import Domain Profiles** on the Mail Policies > Domain Profiles page. The Import Domain Profiles page is displayed.

Step 2 Select the file that contains the exported domain profiles.

Step 3 Click **Submit**. You are warned that importing will replace all existing domain profiles. All of the domain profiles in the text file are imported.

Step 4 Click **Import**.

Deleting Domain Profiles

To remove specific domain profiles from the list of domain profiles:

Step 1 On the Domain Profiles page, mark the checkbox to the right of each domain profile to remove.

Step 2 Click **Delete**.

Step 3 Confirm the deletion.

To remove all of the currently configured domain profiles:

Step 1 Click **Clear All Profiles** on the Domain Profiles page.

Step 2 You are prompted to confirm the deletion.

Searching Domain Profiles

To search all domain profiles for a specific term (typically a username or host name):

Step 1 Specify the search term in the Find Domain Profiles field on the Domain Profiles page.

Step 2 Click **Find Profiles**.

Step 3 The search scans the following fields for each domain profile: email, domain, selector, and signing key name.

Note If you do not enter search terms, the search engine returns all domain profiles.

Signing System-Generated Messages

You can choose whether to sign system-generated messages with a DKIM signature. The types of system-generated message that the Email Security appliance will sign include the following:

- Cisco IronPort Spam Quarantine notifications
- Content filter-generated notifications
- Configuration messages
- Support requests

To enable the appliance to sign system-generated messages:

Step 1 Click **Edit Settings** under DKIM Signing of System Generated Messages on the Domain Signing Profiles page. The DKIM Signing of System Generated Messages.

Step 2 Select **On**.

Step 3 Submit and commit your changes.

Domain Keys and Logging

Lines such as the following are added to the mail logs upon DomainKeys signing:

```
Tue Aug 28 15:29:30 2007 Info: MID 371 DomainKeys: signing with dk-profile - matches
user123@example.com
Tue Aug 28 15:34:15 2007 Info: MID 373 DomainKeys: cannot sign - no profile matches
user12@example.com
```

Lines such as these are added to the mail logs upon DKIM signing:

```
Tue Aug 28 15:29:54 2007 Info: MID 372 DKIM: signing with dkim-profile - matches
user@example.com
Tue Aug 28 15:34:15 2007 Info: MID 373 DKIM: cannot sign - no profile matches
user2@example.com
```

Configuring DKIM Verification

In addition to signing outgoing mail, you can use DKIM to verify incoming mail.

To configure DKIM verification, you need to:

- Create a DKIM verification profile for your mail flow policies.
- Enable DKIM verification on a mail flow policy for inbound mail.

- Optionally, configure a content filter to perform an action for DKIM verified emails using the DKIM authentication condition.

When you configure an AsyncOS appliance for DKIM verification, the following checks are performed:

Step 1 AsyncOS checks for the DKIM-Signature field in incoming mail, the syntax of the signature header, valid tag values, and required tags. If the signature fails any of these checks, AsyncOS returns a *permfail*.

Step 2 After the signature check is performed, the public key is retrieved from the public DNS record, and the TXT record is validated. If errors are encountered during this process, AsyncOS returns a *permfail*. A *tempfail* occurs if the DNS query for the public key fails to get a response.

Step 3 After retrieving the public key, AsyncOS checks the hashed values and verifies the signature. If any failures occur during this step, AsyncOS returns a *permfail*.

Step 4 If the checks all pass, AsyncOS returns a *pass*.

Note When the message body is greater than the specified length, AsyncOS returns the following verdict:

```
dkim = pass (partially verified [x bytes])
```

where *X* represents the number of bytes verified.

The final verification result is entered as an *Authentication-Results* header. For example, you might get a header that looks like one of the following:

```
Authentication-Results: example1.com
    header.from=From:user123@example.com; dkim=pass (signature verified)
Authentication-Results: example1.com
    header.from=From:user123@example.com; dkim=pass (partially verified [1000 bytes])
Authentication-Results: example1.com
    header.from=From:user123@example.com; dkim=permfail (body hash did not verify)
```

Note Current DKIM verification stops at the first valid signature. It is not possible to verify using the last signature encountered. This functionality may be available in a later release.

Managing DKIM Verification Profiles

A DKIM verification profile is a list of parameters that the Email Security appliance's mail flow policies use for verifying DKIM signatures. For example, you can create two verification profiles, one that allows 30 seconds before a query times out and a second that allows only 3 seconds before a query times out. You can assign the second verification profile to the Throttled mail flow policy to prevent connection starvation in case of a DDoS. A verification profile consists of the following information:

- A name for the verification profile.
- The smallest and largest acceptable public key size. The default key sizes are 512 and 2048, respectively.
- The maximum number of signatures in the message to verify. If a message has more signatures than the maximum amount you defined, the appliance skips verification of the remaining signatures and continues to process the message. The default is 5 signatures.

- The maximum allowed difference in time (in seconds) between the sender's system time and verifier's. For example, if the message signature expires at 05:00:00 and the verifier's system time is 05:00:30, the message signature is still valid if the allowed difference in time is 60 seconds but it is invalid if the allowed difference is 10 seconds. The default is 60 seconds.
- An option whether to use a body length parameter.
- The SMTP action to take in case of a temporary failure.
- The SMTP action to take in case of a permanent failure.

You can search through all of your existing verification profiles by the profile name.

You can export your DKIM verification profiles as a text file in your Cisco IronPort appliance's configure directory. When you export the verification profiles, all of the profiles existing on the appliance are put into a single text file. See Exporting DKIM Verification Profiles, page 5-19 for more information.

You can import DKIM verification profiles that you previously exported. Importing DKIM verification profiles causes all of the current DKIM verification profiles on the machine to be replaced. See Importing DKIM Verification Profiles, page 5-19 for more information.

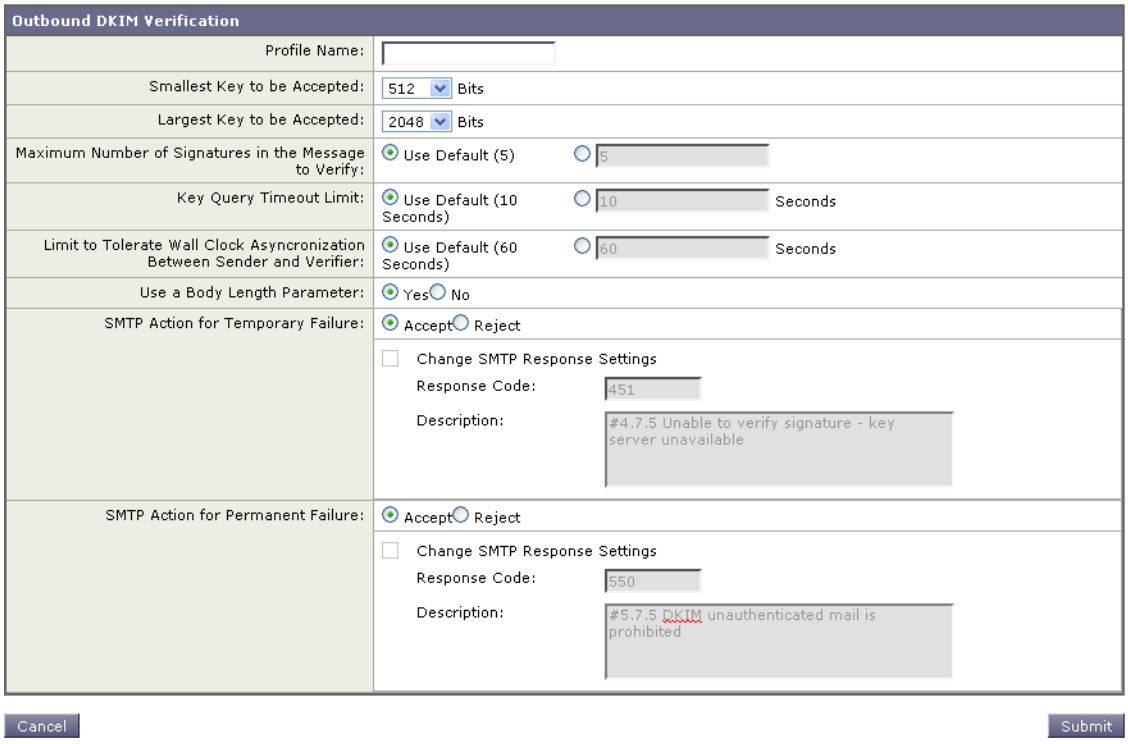
Creating a DKIM Verification Profile

To create a DKIM verification profile:

Step 1 Click **Mail Policies > Verification Profiles**.

Step 2 Click **Add Profile**. The Add DKIM Verification Profiles page is displayed:

Figure 5-13 Add DKIM Verification Profiles Page



Step 3 Enter the name of the profile.

Step 4 Select the minimum key size you want the appliance to accept for signing keys.

Step 5 Select the maximum key size you want the appliance to accept for signing keys.

Step 6 Select the maximum number of signatures to verify in a single message. The default is five signatures.

Step 7 Select the number of seconds before the key query times out. The default is 10 seconds.

Step 8 Select maximum allowed difference in time (in seconds) between the sender's system time and verifier's. The default is 60 seconds.

Step 9 Select whether to use the body-length parameter in the signature to verify the message.

Step 10 Select whether the Email Security appliance accepts or rejects the message if there is a temporary failure when verifying its signature. If you want the appliance to reject the message, you can choose to have it send the default 451 SMTP response code or another SMTP response code and text.

Step 11 Select whether the Email Security appliance accepts or rejects the message if there is a permanent failure when verifying its signature. If you want the appliance to reject the message, you can choose to have it send the default 451 SMTP response code or another SMTP response code and text.

Step 12 Submit your changes.

The new profile appears in the DKIM Verification Profiles table.

Step 13 Commit your changes.

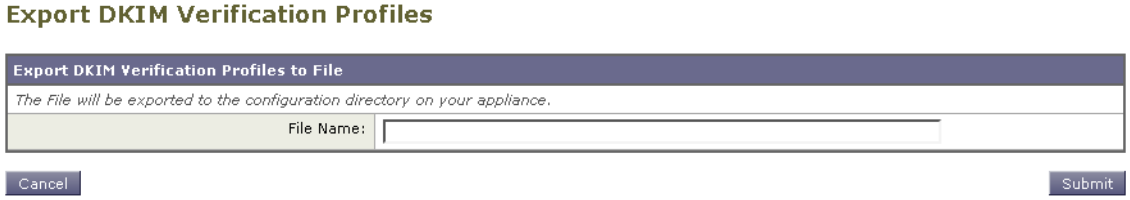
Step 14 At this point you should enable DKIM verification on an incoming mail flow policy and select the verification profile you want to use.

Exporting DKIM Verification Profiles

When you export DKIM verification profiles, all of the DKIM verification profiles currently existing on your Cisco IronPort appliance are exported as a single text file and saved in the configuration directory on the appliance. To export DKIM verification profiles:

Step 1 Click **Export Profiles** on the **Mail Policies > Verification Profiles** page. The Export DKIM Verification Profile page is displayed:

Figure 5-14 Export DKIM Verification Profiles Page



Step 2 Enter a name for the file and click **Submit**.

Importing DKIM Verification Profiles

To import DKIM verification profiles from an existing file:

Step 1 Click Import Profiles on the **Mail Policies > Verification Profiles** page. The Import DKIM Verification Profiles page is displayed.

Step 2 Select the file that contains the DKIM verification profiles.

Step 3 Click **Submit**. You are warned that importing will replace all existing DKIM verification profiles.

Step 4 Click **Import**.

Deleting DKIM Verification Profiles

To remove specific DKIM verification profiles from the list of DKIM verification profiles:

Step 1 On the **Mail Policies > Verification Profiles** page, mark the checkbox to the right of each DKIM verification profile you want to delete.

Step 2 Click **Delete**.

Step 3 Confirm the deletion.

To remove all of the currently configured DKIM verification profiles:

Step 1 Click **Clear All Profiles** on the **Mail Policies > Verification Profiles** page.

Step 2 Confirm the deletion.

Searching DKIM Verification Profiles

To search all DKIM verification profiles for a specific term in the profile name:

Step 1 Specify the search term in the **Search DKIM Verification Profiles** for field on the DKIM Verification Profiles page.

Step 2 Click **Find Profiles**.

The search scans the profile name for each DKIM verification profile.

If you do not enter search terms, the search engine returns all DKIM verification profiles.

Configuring DKIM Verification on the Mail Flow Policy

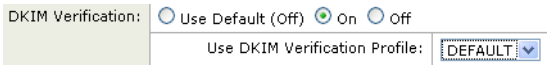
DKIM verification is enabled on mail flow policies for incoming email.

To enable verification on an incoming mail flow policy:

Step 1 On the Mail Flow Policies page (from the Mail Policies menu), click on the incoming mail policy for the listener where you want to perform verification.

Step 2 In the Security Features section of the mail flow policy, enable DKIM Verification by selecting On.

Figure 5-15 Enabling DKIM Verification



Step 3 Select the DKIM verification profile that you want to use for the policy.

Step 4 Commit your changes.

DKIM Verification and Logging

Lines such as the following are added to the mail logs upon DKIM verification:

```
mail.current:Mon Aug  6 13:35:38 2007 Info: MID 17 DKIM: no signature


mail.current:Mon Aug  6 15:00:37 2007 Info: MID 18 DKIM: verified pass
```

Configuring an Action for DKIM Verified Mail

When you verify DKIM mail, an *Authentication-Results* header is added to the mail, but the mail is accepted regardless of the authentication result. To configure actions based on these authentication results, you can create a content filter to perform actions on the DKIM-verified mail. For example, if DKIM verification fails, you may want configure the mail to be delivered, bounced, dropped, or sent to a quarantine. To do this, you must configure an action using a content filter.

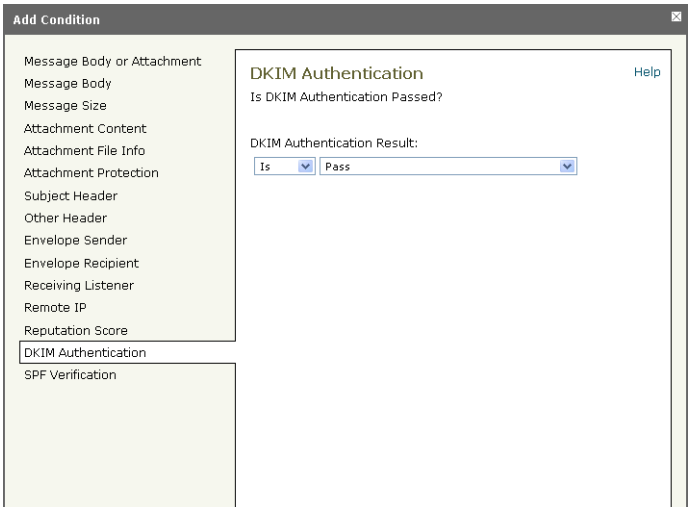
To add an action from the GUI, complete the following steps:

Step 1 From Mail Policies > Incoming Filters, click **Add Filter**

Step 2 In the Conditions section, click **Add Condition**

Step 3 Select DKIM Authentication.

Figure 5-16 DKIM Authentication Content Filter Condition



Step 4 Choose a DKIM condition. Select one of the following options:

- **Pass. The messaged passed the authentication tests.**
- **Neutral**. The message was not signed.
- **Temperror**. A recoverable error occurred.
- **Permerror**. An unrecoverable error occurred.
- **Hardfail**. The authentication tests failed.
- **None**. Authentication was not performed.

Step 5 Select an action to associate with the condition. For example, if the DKIM verification fails, you may want to notify the recipient and bounce the message. Or, if DKIM verification passes, you may want to deliver the message immediately without further processing.

Step 6 Submit the new content filter.

Step 7 Enable the content filter on the appropriate incoming mail policy.

Step 8 Commit your changes.

Overview of SPF and SIDF Verification

Cisco IronPort AsyncOS supports Sender Policy Framework (SPF) and Sender ID Framework (SIDF) verification. SPF and SIDF are methods for verifying authenticity of email based on DNS records. SPF and SIDF allow the owner of an Internet domain to use a special format of DNS TXT records to specify which machines are authorized to transmit email for that domain.

When you use SPF/SIDF authentication, the senders publish SPF records specifying which hosts are permitted to use their names, and compliant mail receivers use the published SPF records to test the authorization of the sending Mail Transfer Agent's identity during a mail transaction.

Note Because SPF checks require parsing and evaluation, AsyncOS performance may be impacted. In addition, be aware that SPF checks increase the load on your DNS infrastructure.

When you work with SPF and SIDF, note that SIDF is similar to SPF, but it has some differences. To get a full description of the differences between SIDF and SPF, see RFC 4406. For the purposes of this documentation, the two terms are discussed together except in the cases where only one type of verification applies.

Note AsyncOS does not support SPF for incoming relays, and AsyncOS does not support SPF for IPv6.

A Note About Valid SPF Records

To use SPF and SIDF with a Cisco IronPort appliance, publish the SPF record according to the RFCs 4406 and 4408. Review RFC 4407 for a definition of how the PRA identity is determined. You may also want to refer to the following website to view common mistakes made when creating SPF and SIDF records:

```
http://www.openspf.org/FAQ/Common_mistakes
```

Valid SPF Records

To pass the SPF HELO check, ensure that you include a "v=spf1 a –all" SPF record for each sending MTA (separate from the domain). If you do not include this record, the HELO check will likely result in a None verdict for the HELO identity. If you notice that SPF senders to your domain return a high number of None verdicts, these senders may not have included a "v=spf1 a –all" SPF record for each sending MTA.

Valid SIDF Records

To support the SIDF framework, you need to publish both "v=spf1" and "spf2.0" records. For example, your DNS record may look like the following example:

```
example.com. TXT "v=spf1 +mx a:colo.example.com/28 -all"

smtp-out.example.com TXT "v=spf1 a -all"

example.com. TXT "spf2.0/mfrom,pra +mx a:colo.example.com/28 -all"
```

SIDF does not verify the HELO identity, so in this case, you do not need to publish SPF v2.0 records for each sending MTA.

Note If you choose not to support SIDF, publish an "spf2.0/pra ~all" record.

Testing Your SPF Records

In addition to reviewing the RFCs, it is a good idea to test your SPF records before you implement SPF verification on a Cisco IronPort appliance. There are several testing tools available on the openspf.org website:

`http://www.openspf.org/Tools`

You can use the following tool to determine why an email failed an SPF record check:

`http://www.openspf.org/Why`

In addition, you can enable SPF on a test listener and use Cisco's `trace` CLI command (or perform trace from the GUI) to view the SPF results. Using trace, you can easily test different sending IPs.

Working with SPF on a Cisco IronPort Email Security Appliance

To use SPF/SIDF on a Cisco IronPort appliance, complete the following steps:

Step 1 **Enable SPF/SIDF.** You enable SPF/SIDF on an incoming listener from the default mail flow policy, or you can enable it for different incoming mail flow policies. For more information, see Enabling SPF and SIDF, page 5-24.

Step 2 **Configure actions to take on SPF/SIDF-verified mail.** You can use message or content filters to determine actions to take for SPF-verified mail. For more information, see Determining the Action to Take for SPF/SIDF Verified Mail, page 5-31.

Step 3 **Test the SPF/SIDF results.** Because organizations use different email authorization methods, and each organization may use SPF/SIDF differently (for example, the SPF or SIDF policy may conform to different standards), you need to test the SPF/SIDF results to ensure that you do not bounce or drop emails from authorized senders. You can test the SPF/SIDF results by using a combination of content filters, message filters, and the Content Filters report. For more information about testing the SPF/SIDF results, see Testing the SPF/SIDF Results, page 5-34.

Warning **Although Cisco strongly endorses email authentication globally, at this point in the industry's adoption, Cisco suggests a cautious disposition for SPF/SIDF authentication failures. Until more organizations gain greater control of their authorized mail sending infrastructure, Cisco urges customers to avoid bouncing emails and instead quarantine emails that fail SPF/SIDF verification.**

The AysncOS command line interface (CLI) provides more control settings for SPF level than the web interface. Based on the SPF verdict, the appliance can accept or reject a message, in SMTP conversation, on a per listener basis. You can modify the SPF settings when editing the default settings for a listener's Host Access Table using the `listenerconfig` command. See the Enabling SPF and SIDF via the CLI, page 5-25 for more information on the settings.

Enabling SPF and SIDF

To use SPF/SIDF, you must enable SPF/SIDF for a mail flow policy on an incoming listener. You can enable SPF/SIDF on the listener from the default mail flow policy, or you can enable it for particular incoming mail flow policies.

To enable SPF/SIDF on the default mailflow policy via the GUI:

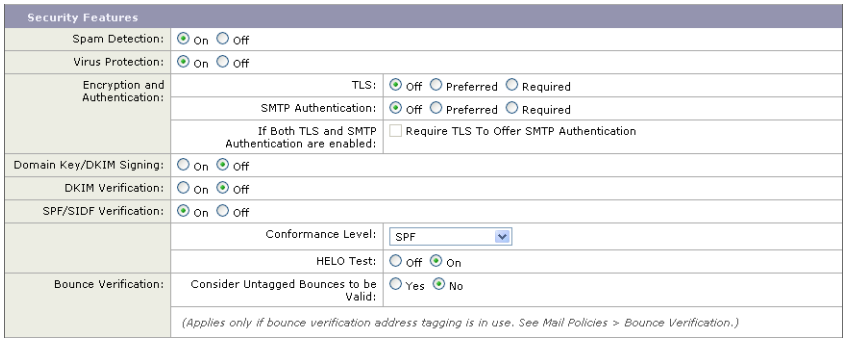
Step 1 Click Mail Policies > Mail Flow Policy.

Step 2 Click Default Policy Parameters.

Step 3 In the default policy parameters, view the Security Features section.

Step 4 In the SPF/SIDF Verification section, click Yes.

Figure 5-17 Enabling SPF/SIDF in the Mail Flow Policy



Step 5 Set the level of conformance (the default is SIDF-compatible). This option allows you to determine which standard of SPF or SIDF verification to use. In addition to SIDF conformance, you can choose SIDF-compatible, which combines SPF and SIDF.

Table 5-1 SPF/SIDF Conformance Levels

Conformance Level	Description
SPF	The SPF/SIDF verification behaves according to RFC4408. - No purported responsible address (PRA) identity verification takes place. **NOTE**: Select this conformance option to test against the HELO identity.

Table 5-1 SPF/SIDF Conformance Levels

Conformance Level	Description
SIDF	The SPF/SIDF verification behaves according to RFC4406. -The PRA Identity is determined with full conformance to the standard. - SPF v1.0 records are treated as spf2.0/mfrom,pra. - For a nonexistent domain or a malformed identity, a verdict of Fail is returned.
SIDF Compatible	The SPF/SIDF verification behaves according to RFC4406 *except for* the following differences: - SPF v1.0 records are treated as spf2.0/mfrom. - For a nonexistent domain or a malformed identity, a verdict of None is returned. **NOTE**: This conformance option was introduced at the request of the OpenSPF community (www.openspf.org).

Note More settings are available via the CLI. See Enabling SPF and SIDF via the CLI, page 5-25 for more information.

Step 6 If you choose a conformance level of SIDF-compatible, configure whether the verification downgrades a Pass result of the PRA identity to None if there are Resent-Sender: or Resent-From: headers present in the message. You might choose this option for security purposes.

Step 7 If you choose a conformance level of SPF, configure whether to perform a test against the HELO identity. You might use this option to improve performance by disabling the HELO check. This can be useful because the `spf-passed` filter rule checks the PRA or the MAIL FROM Identities first. The appliance only performs the HELO check for the SPF conformance level.

Enabling SPF and SIDF via the CLI

The AsyncOS CLI supports more control settings for each SPF/SIDF conformance level. When configuring the default settings for a listener's Host Access Table, you can choose the listener's SPF/SIDF conformance level and the SMTP actions (ACCEPT or REJECT) that the appliance performs, based on the SPF/SIDF verification results. You can also define the SMTP response that the appliance sends when it rejects a message.

Depending on the conformance level, the appliance performs a check against the HELO identity, MAIL FROM identity, or PRA identity. You can specify whether the appliance proceeds with the session (ACCEPT) or terminates the session (REJECT) for each of the following SPF/SIDF verification results for each identity check:

- **None.** No verification can be performed due to the lack of information.
- **Neutral.** The domain owner does not assert whether the client is authorized to use the given identity.
- **SoftFail.** The domain owner believes the host is not authorized to use the given identity but is not willing to make a definitive statement.
- **Fail.** The client is not authorized to send mail with the given identity.
- **TempError.** A transient error occurred during verification.

- **PermError.** A permanent error occurred during verification.

The appliance accepts the message for a Pass result unless you configure the SIDF Compatible conformance level to downgrade a Pass result of the PRA identity to None if there are Resent-Sender: or Resent-From: headers present in the message. The appliance then takes the SMTP action specified for when the PRA check returns None.

If you choose not to define the SMTP actions for an identity check, the appliance automatically accepts all verification results, including Fail.

The appliance terminates the session if the identity verification result matches a REJECT action for any of the enabled identity checks. For example, an administrator configures a listener to accept messages based on all HELO identity check results, including Fail, but also configures it to reject messages for a Fail result from the MAIL FROM identity check. If a message fails the HELO identity check, the session proceeds because the appliance accepts that result. If the message then fails the MAIL FROM identity check, the listener terminates the session and then returns the STMP response for the REJECT action.

The SMTP response is a code number and message that the appliance returns when it rejects a message based on the SPF/SIDF verification result. The TempError result returns a different SMTP response from the other verification results. For TempError, the default response code is `451` and the default message text is `#4.4.3 Temporary error occurred during SPF verification`. For all other verification results, the default response code is `550` and the default message text is `#5.7.1 SPF unauthorized mail is prohibited`. You can specify your own response code and message text for TempError and the other verification results.

Optionally, you can configure the appliance to return a third-party response from the SPF publisher domain if the REJECT action is taken for Neutral, SoftFail, or Fail verification result. By default, the appliance returns the following response:

```
550-#5.7.1 SPF unauthorized mail is prohibited.

550-The domain example.com explains:

550 <Response text from SPF domain publisher>
```

To enable these SPF/SIDF settings, use the `listenerconfig -> edit` subcommand and select a listener. Then use the `hostaccess -> default` subcommand to edit the Host Access Table's default settings. Answer `yes` to the following prompts to configure the SPF controls:

```
Would you like to change SPF/SIDF settings?  [N]> yes

Would you like to perform SPF/SIDF Verification?  [Y]> yes
```

The following SPF control settings are available for the Host Access Table:

Table 5-2 SPF Control Settings via the CLI

Conformance Level	Available SPF Control Settings
SPF Only	• whether to perform HELO identity check • SMTP actions taken based on the results of the following identity checks: – HELO identity (if enabled) – MAIL FROM Identity • SMTP response code and text returned for the REJECT action • verification time out (in seconds)
SIDF Compatible	• whether to perform a HELO identity check • whether the verification downgrades a Pass result of the PRA identity to None if the Resent-Sender: or Resent-From: headers are present in the message • SMTP actions taken based on the results of the following identity checks: – HELO identity (if enabled) – MAIL FROM Identity – PRA Identity • SMTP response code and text returned for the REJECT action • verification timeout (in seconds)
SIDF Strict	• SMTP actions taken based on the results of the following identity checks: – MAIL FROM Identity – PRA Identity • SMTP response code and text returned in case of SPF REJECT action • verification timeout (in seconds)

The following example shows a user configuring the SPF/SIDF verification using the SPF Only conformance level. The appliance performs the HELO identity check and accepts the None and Neutral verification results and rejects the others. The CLI prompts for the SMTP actions are the same for all identity types. The user does not define the SMTP actions for the MAIL FROM identity. The appliance automatically accepts all verification results for the identity. The appliance uses the default reject code and text for all REJECT results.

```
Would you like to change SPF/SIDF settings?  [N]> yes


Would you like to perform SPF/SIDF Verification?  [N]> yes
```

```
What Conformance Level would you like to use?

1. SPF only

2. SIDF compatible

3. SIDF strict

[2]> 1


Would you like to have the HELO check performed? [Y]> y


Would you like to change SMTP actions taken as result of the SPF verification? [N]> y


Would you like to change SMTP actions taken for the HELO identity? [N]> y


What SMTP action should be taken if HELO check returns None?

1. Accept

2. Reject

[1]> 1


What SMTP action should be taken if HELO check returns Neutral?

1. Accept

2. Reject

[1]> 1


What SMTP action should be taken if HELO check returns SoftFail?

1. Accept

2. Reject

[1]> 2


What SMTP action should be taken if HELO check returns Fail?
```

```
1. Accept

2. Reject

    [1]> 2


    What SMTP action should be taken if HELO check returns TempError?

    1. Accept

2. Reject

[1]> 2


What SMTP action should be taken if HELO check returns PermError?

1. Accept

2. Reject

[1]> 2


Would you like to change SMTP actions taken for the MAIL FROM identity? [N]> n


Would you like to change SMTP response settings for the REJECT action? [N]> n


Verification timeout (seconds)

[40]>
```

The following shows how the SPF/SIDF settings are displayed for the listener's Default Policy Parameters.

```
SPF/SIDF Verification Enabled: Yes

  Conformance Level: SPF only

  Do HELO test: Yes

  SMTP actions:

    For HELO Identity:

      None, Neutral: Accept
```

```
      SoftFail, Fail, TempError, PermError: Reject

    For MAIL FROM Identity: Accept

  SMTP Response Settings:

    Reject code: 550

    Reject text: #5.7.1 SPF unauthorized mail is prohibited.

    Get reject response text from publisher: Yes

    Defer code: 451

    Defer text: #4.4.3 Temporary error occurred during SPF verification.

  Verification timeout: 40
```

See the *Cisco IronPort AsyncOS CLI Reference Guide* for more information on the `listenerconfig` command.

The Received-SPF Header

When you configure AsyncOS for SPF/SIDF verification, it places an SPF/SIDF verification header (`Received-SPF`) in the email. The `Received-SPF` header contains the following information:

- **verification result** - the SPF verification result (see Verification Results, page 5-31).
- **identity** - the identity that SPF verification checked: HELO, MAIL FROM, or PRA.
- **receiver** - the verifying host name (which performs the check).
- **client IP address** - the IP address of the SMTP client.
- **ENVELOPE FROM** - the envelope sender mailbox. (Note that this may be different from the MAIL FROM identity, as the MAIL FROM identity cannot be empty.)
- **x-sender** - the value of the HELO, MAIL FROM, or PRA identity.
- **x-conformance** - the level of conformance (see Table 5-1SPF/SIDF Conformance Levels, page 5-24) and whether a downgrade of the PRA check was performed.

The following example shows a header added for a message that passed the SPF/SIDF check:

```
Received-SPF: Pass identity=pra; receiver=box.example.com;

    client-ip=1.2.3.4; envelope-from="alice@fooo.com";

    x-sender="alice@company.com"; x-conformance=sidf_compatible
```



Note The `spf-status` and `spf-passed` filter rules use the received-SPF header to determine the status of the SPF/SIDF verification.

Determining the Action to Take for SPF/SIDF Verified Mail

When you receive SPF/SIDF verified mail, you may want to take different actions depending on the results of the SPF/SIDF verification. You can use the following message and content filter rules to determine the status of SPF/SIDF verified mail and perform actions on the messages based on the verification results:

- `spf-status`. This filter rule determines actions based on the SPF/SIDF status. You can enter a different action for each valid SPF/SIDF return value.
- `spf-passed`. This filter rule generalizes the SPF/SIDF results as a Boolean value.

Note The `spf-passed` filter rule is only available in message filters.

You can use the `spf-status` rule when you want to address more granular results, and use the `spf-passed` rule when you want to create a simple Boolean.

Verification Results

If you use the `spf-status` filter rule, you can check against the SPF/SIDF verification results using the following syntax:

```
if (spf-status == "Pass")
```

If you want a single condition to check against multiple status verdicts, you can use the following syntax:

```
if (spf-status == "PermError, TempError")
```

You can also check the verification results against the HELO, MAIL FROM, and PRA identities using the following syntax:

```
if (spf-status("pra") == "Fail")
```

Note You can only use the `spf-status` message filter rule to check results against HELO, MAIL FROM, and PRA identities. You cannot use the `spf-status` content filter rule to check against identities.

You can receive any of the following verification results:

- None - no verification can be performed due to the lack of information.
- Pass - the client is authorized to send mail with the given identity.
- Neutral - the domain owner does not assert whether the client is authorized to use the given identity.
- SoftFail - the domain owner believes the host is not authorized to use the given identity but is not willing to make a definitive statement.
- Fail - the client is not authorized to send mail with the given identity.
- TempError - a transient error occurred during verification.

- PermError - a permanent error occurred during verification.

Using the spf-status Filter Rule in the CLI

The following example shows the `spf-status` message filter in use:

```
skip-spam-check-for-verified-senders:

     if (sendergroup == "TRUSTED" and spf-status == "Pass"){

         skip-spamcheck();

     }


quarantine-spf-failed-mail:

     if (spf-status("pra") == "Fail") {

         if (spf-status("mailfrom") == "Fail"){

             # completely malicious mail

             quarantine("Policy");

         } else {

           if(spf-status("mailfrom") == "SoftFail") {

             # malicious mail, but tempting

             quarantine("Policy");

           }

         }

     } else {

       if(spf-status("pra") == "SoftFail"){

         if (spf-status("mailfrom") == "Fail"

                  or spf-status("mailfrom") == "SoftFail"){

             # malicious mail, but tempting

             quarantine("Policy");

         }

       }

     }
```

```
stamp-mail-with-spf-verification-error:

     if (spf-status("pra") == "PermError, TempError"

             or spf-status("mailfrom") == "PermError, TempError"

             or spf-status("helo") == "PermError, TempError"){

          # permanent error - stamp message subject

          strip-header("Subject");

          insert-header("Subject", "[POTENTIAL PHISHING] $Subject"); }
```

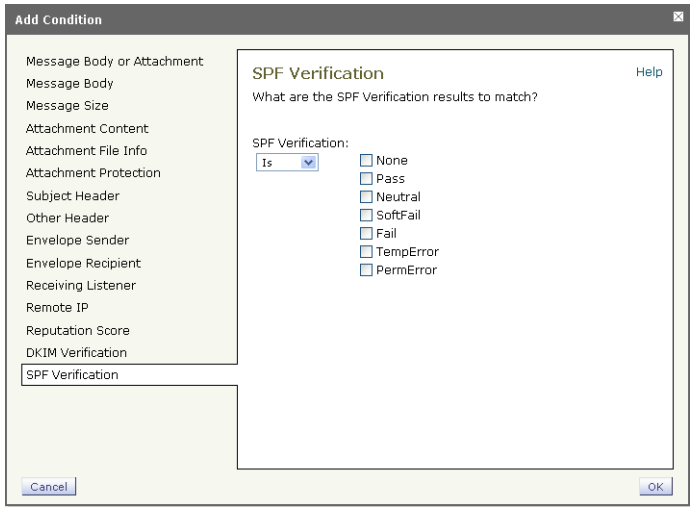
.

spf-status Content Filter Rule in the GUI

You can also enable the `spf-status` rule from the content filters in the GUI. However, you cannot check results against HELO, MAIL FROM, and PRA identities when using the `spf-status` content filter rule.

To add the `spf-status` content filter rule from the GUI, click Mail Policies > Incoming Content Filters. Then add the SPF Verification filter rule from the Add Condition dialog box. Specify one or more verification results for the condition.

Figure 5-18 Using the spf-status Content Filter Rule



After you add the SPF Verification condition, specify an action to perform based on the SPF status. For example, if the SPF status is SoftFail, you might quarantine the message.

Using the spf-passed Filter Rule

The `spf-passed` rule shows the results of SPF verification as a Boolean value. The following example shows an `spf-passed` rule used to quarantine emails that are not marked as spf-passed:

```
quarantine-spf-unauthorized-mail:

    if (not spf-passed) {

        quarantine("Policy");

    }
```

Note Unlike the `spf-status` rule, the `spf-passed` rule reduces the SPF/SIDF verification values to a simple Boolean. The following verification results are treated as not passed in the `spf-passed` rule: None, Neutral, Softfail, TempError, PermError, and Fail. To perform actions on messages based on more granular results, use the `spf-status` rule.

Testing the SPF/SIDF Results

Test the results of SPF/SIDF verification and use these results to determine how to treat SPF/SIDF failures because different organizations implement SPF/SIDF in different ways. Use a combination of content filters, message filters, and the Email Security Monitor - Content Filters report to test the results of the SPF/SIDF verification.

Your degree of dependence on SPF/SIDF verification determines the level of granularity at which you test SPF/SIDF results.

Basic Granularity Test of SPF/SIDF Results

To get a basic measure of the SPF/SIDF verification results for incoming mail, you can use content filters and the Email Security Monitor - Content Filters page. This test provides a view of the number of messages received for each type of SPF/SIDF verification result.

To perform a basic SPF/SIDF verification test:

Step 1 Enable SPF/SIDF verification for a mail flow policy on an incoming listener, and use a content filter to configure an action to take. For information on enabling SPF/SIDF, see Enabling SPF and SIDF, page 5-24.

Step 2 Create an `spf-status` content filter for each type of SPF/SIDF verification. Use a naming convention to indicate the type of verification. For example, use “SPF-Passed” for messages that pass SPF/SIDF verification, or “SPF-TempErr” for messages that weren’t passed due to a transient error during verification. For information about creating an `spf-status` content filter, see spf-status Content Filter Rule in the GUI, page 5-33.

Step 3 After you have processed a number of SPF/SIDF verified messages, click Monitor > Content Filters to see how many messages triggered each of the SPF/SIDF verified content filters.

Greater Granularity Test of SPF/SIDF Results

For more comprehensive information about SPF/SIDF verification results, only enable SPF/SIDF verification for specific groups of senders, and review the results for those specific senders. Then, create a mail policy for that particular group and enable SPF/SIDF verification on the mail policy. Create content filters and review the Content Filters report as explained in Basic Granularity Test of SPF/SIDF Results, page 5-34. If you find that the verification is effective, then you can use SPF/SIDF verification as a basis for deciding whether to drop or bounce emails for this specified group of senders.

To perform a granular SPF/SIDF verification test:

Step 1 Create a mail flow policy for SPF/SIDF verification. Enable SPF/SIDF verification for the mail flow policy on an incoming listener. For information about enabling SPF/SIDF, see Enabling SPF and SIDF, page 5-24.

Step 2 Create a sender group for SPF/SIDF verification and use a naming convention to indicate SPF/SIDF verification. For information about creating sender groups, see the "Configuring the Gateway to Receive Mail" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

Step 3 Create an `spf-status` content filter for each type of SPF/SIDF verification. Use a naming convention to indicate the type of verification. For example, use "SPF-Passed" for messages that pass SPF/SIDF verification, or "SPF-TempErr" for messages that weren't passed due to a transient error during verification. For information about creating an `spf-status` content filter, see spf-status Content Filter Rule in the GUI, page 5-33.

Step 4 After you process a number of SPF/SIDF-verified messages, click Monitor > Content Filters to see how many messages triggered each of the SPF/SIDF-verified content filters.



CHAPTER 6

Using Message Filters to Enforce Email Policies

The Cisco IronPort appliance contains extensive content scanning and message filtering technology that allows you to enforce corporate policies and act on specific messages as they enter or leave your corporate networks.

This chapter contains information about the powerful combinations of features available for policy enforcement: a content scanning engine, message filters, attachment filters, and content dictionaries.

This chapter contains the following sections:

- Overview, page 6-1
- Components of a Message Filter, page 6-2
- Message Filter Processing, page 6-3
- Message Filter Rules, page 6-9
- Message Filter Actions, page 6-42
- Attachment Scanning, page 6-65
- Using the CLI to Manage Message Filters, page 6-75
- Message Filter Examples, page 6-97

Overview

Message filters allow you to create special rules describing how to handle messages as they are received by the Cisco IronPort appliance. A message filter specifies that a certain kind of email message should be given special treatment. Cisco IronPort message filters also allow you to enforce corporate email policy by scanning the content of messages for words you specify. This chapter contains the following sections:

- **Components of a message filter**. Message filters allow you to create special rules describing how to handle messages as they are received. Filter rules identify messages based on message or attachment content, information about the network, message envelope, message headers, or message body. Filter actions generate notifications or allow messages to be dropped, bounced, archived, blind carbon copied, or altered. For more information, see Components of a Message Filter, page 6-2.
- **Processing Message Filters**. When AsyncOS processes message filters, the content that AsyncOS scans, the order of the processing, and the actions taken are based on several factors, including the message filter order, any prior processing that may have altered the message content, the MIME structure of the message, the threshold score configured for content matching, and structure of the query. For more information, see Message Filter Processing, page 6-3.

- **Message Filter Rules**. Each filter has a rule that defines the collection of messages that the filter can act upon. You define those rules when you create a message filter. For more information, see Message Filter Rules, page 6-9.
- **Message Filter Actions**. Each filter has an action that is performed on a message if the rule evaluates to `true`. There are two types of actions that can be performed: final actions (such as delivering, dropping, or bouncing a message), or non-final actions (such as stripping or inserting a header) which permit the message to be further processed. For more information, see Message Filter Actions, page 6-42.
- **Attachment Scanning Message Filters**. Attachment scanning message filters allow you to strip attachments from messages that are inconsistent with your corporate policies, while still retaining the ability to deliver the original message. You can filter attachments based on their specific file type, fingerprint, or content. You can also scan image attachments using an image analyzer. The image analyzer creates algorithms to measure skin color, body size and curvature to determine the probability that the graphic contains inappropriate content. For more information, see Attachment Scanning, page 6-65.
- **Using the CLI to Manage Message Filters**. The CLI accepts commands for working with message filters. For example, you might want to display, reorder, import or export a list of message filters. For more information, see Using the CLI to Manage Message Filters, page 6-75.
- **Message Filter Examples**. This section contains some real world examples of filters with a brief discussion of each. For more information, see Message Filter Examples, page 6-97.

Components of a Message Filter

Message filters allow you to create special rules describing how to handle messages as they are received. A message filter is comprised of message filter rules and message filter actions.

Message Filter Rules

Message filter rules determine the messages that a filter will act on. Rules may be combined using the logical connectors AND, OR, and NOT to create more complex tests. Rule expressions may also be grouped using parentheses.

Message Filter Actions

The purpose of message filters is to perform actions on selected messages.

The two types of actions are:

- *Final* actions — such as `deliver`, `drop`, and `bounce` — end the processing of a message, and permit no further processing through subsequent filters.
- *Non-final* actions perform an action which permits the message to be processed further.



Note Non-final message filter actions are cumulative. If a message matches multiple filters where each filter specifies a different action, then all actions are accumulated and enforced. However, if a message matches multiple filters specifying the same action, the prior actions are overridden and the final filter action is enforced.

Message Filter Example Syntax

The intuitive meaning of a filter specification is:

if the message matches the rule, *then* apply the actions in sequence. If the `else` clause is present, the actions within the `else` clause are executed in the event the message does *not match* the rule.

The name of the filter you specify makes it easier to manage filters when you are activating, deactivating, or deleting them.

Message filters use the following syntax:

Example Syntax	Purpose
expedite:	filter name
if (recv-listener == 'InboundMail' or recv-int == 'notmain')	rule specification
`{` `  alt-src-host('outbound1');` `skip_filters();` `}`	action specification
`else` `{` `  alt-src-host('outbound2');` `}`	optional alternative action specification

Note that you can omit any alternative actions:

Example Syntax	Purpose
expedite2:	filter name
`if ((not (recv-listener == 'InboundMail')) and` **(not (recv-int == 'notmain')))**	rule specification
`{` `  alt-src-host('outbound2');` `skip_filters();` **}**	action specification

You can combine several filters in sequence within a single text file, one following the other.

You must enclose the values in filters in either single or double quotation marks. Single or double quotation marks must be equally paired on each side of the value; for example, the expressions `notify('customercare@example.com')` and `notify("customercare@example.com")` are both legal, but the expression `notify("customercare@example.com')` causes a syntax error.

Lines beginning with a '`#`' character are considered comments and are ignored; however, they are not preserved by AsyncOS as can be verified by viewing a filter via `filters -> detail`.

Message Filter Processing

When AsyncOS processes message filters, the content that AsyncOS scans, the order of the processing, and the actions taken are based on several factors:

- **Message filter order.** Message filters are maintained in an ordered list. When a message is processed, AsyncOS applies each message filter in the order it appears in the list. If a final action occurs, no further action is taken on the message. For more information, see Message Filter Order, page 6-4.
- **Prior processing**. Actions performed on AsyncOS messages may add or remove headers before the message filter is evaluated. AsyncOS processes the message filter process on the headers that are present in the message at the time of processing. For more information, see Message Header Rules and Evaluation, page 6-4.
- **The MIME structure of the message**. The MIME structure of the message determines which part of the message is treated as "body," and which part of the message is treated as an "attachment". Many message filters are configured to act on just the body or just the attachment part of the message. For more information, see Message Bodies vs. Message Attachments, page 6-5.
- **The threshold score configured for the regular expression**. When you match a regular expression, you configure a "score" to tally up the number of times a match must occur before a filter action is taken. This allows you to "weight" the responses to different terms. For more information, see Thresholds for Matches in Content Scanning, page 6-6.
- **The structure of the query**. When evaluating AND or OR tests within message filters, AsyncOS does not evaluate unneeded tests. In addition, it is important to note that the system does not evaluate the tests from left to right. Instead, when AND and OR tests are evaluated, the least expensive test is evaluated first. For more information, see AND Test and OR Tests in Message Filters, page 6-8.

Message Filter Order

Message filters are kept in an ordered list and numbered by their position in the list. When a message is processed, the message filters are applied in the associated numeric order. Therefore, filter number 30 will not have a chance to alter the source host of a message if filter number 9 has already executed a final action on (for example, bounced) the message. The position of a filter in the list can be changed via the system user interfaces. Filters imported via a file are ordered based on their relative order in the imported file.

After a final action, no further actions may be taken on the message.

Although a message may match a filter rule, the filter may not act upon that message for any of the following reasons:

- The filter is inactive.
- The filter is invalid.
- The filter has been superseded by an earlier filter that executed a final action for the message.

Message Header Rules and Evaluation

Filters evaluate "processed" headers rather than the original message headers when applying header rules. Thus:

- If a header was added by a previous processing action, it can now be matched by any subsequent header rule.
- If a header was stripped by a previous processing action, it can no longer be matched by any subsequent header rule.

- If a header was modified by a previous processing action, any subsequent header rule will evaluate the modified header and not the original message header.

This behavior is common to both message filters and content filters.

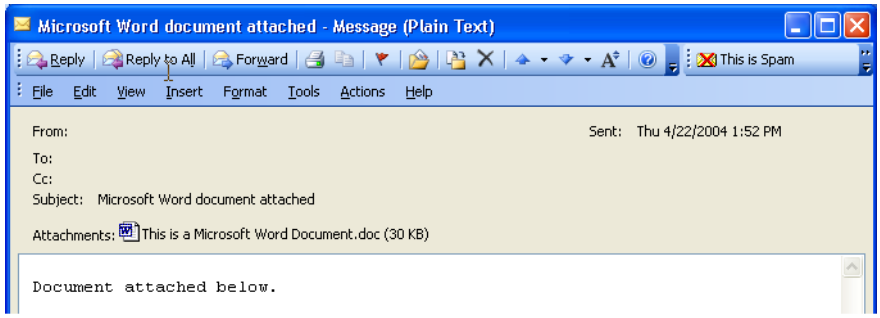
Message Bodies vs. Message Attachments

An email message is composed of multiple parts. Although RFCs define everything that comes after a message's headers as a multipart "message body," many users still conceptualize a message's "body" and its "attachment" differently. When you use any of the Cisco IronPort message filters named `body-`*variable* or `attachment-`*variable*, the Cisco IronPort appliance attempts to distinguish the parts that most users consider to be the "body" and the "attachment" in the same way that many MUAs attempt to render these parts differently.

For the purposes of writing `body-`*variable* or `attachment-`*variable* message filter rules, everything after the message headers is considered the message *body*, whose content is considered the first text part of the MIME parts that are within the body. Anything after the content, (that is, any additional MIME parts) is considered an *attachment*. AsyncOS evaluates the different MIME parts of the message, and identifies the parts of the file that is treated as an attachment.

For example, Figure 6-1 shows a message in the Microsoft Outlook MUA where the words "`Document attached below.`" appear as a plain text message *body* and the document "`This is a Microsoft Word document.doc`" appears as an *attachment*. Because many users conceptualize email this way (rather than as a multipart message whose first part is plain text and whose second part is a binary file), the Cisco IronPort uses the term "attachment" in message filters to create rules to differentiate and act on the .doc file part (in essence, the second MIME part) as opposed to the "body" of the message (the first, plain text part) — although, according to the language used in RFCSs 1521 and 1522, a message's *body* is comprised of all MIME parts.

Figure 6-1 Message with "Attachment"



Because the Cisco IronPort appliance makes this distinction between the *body* and the *attachment* in multipart messages, there are several cases you should be aware of when using the `body-`*variable* or `attachment-`*variable* message filter rules in order to achieve the expected behavior:

- If you have a message with a single text part—that is, a message containing a header of "Content-Type: text/plain" or "Content-Type: text/html" — the Cisco IronPort appliance will consider the entire message as the body. If the content type is anything different, the Cisco IronPort appliance considers it to be a single attachment.
- Some encoded files (uuencoded, for example) are included in the body of the email message. When this occurs, the encoded file is treated as an attachment, and it is extracted and scanned, while the remaining text is considered to be the body of the text.
- A single, non-text part is always considered an *attachment*. For example, a message consisting of only a.zip file is considered an attachment.

Thresholds for Matches in Content Scanning

When you add filter rules that search for patterns in the message body or attachments, you can specify the minimum threshold for the number of times the pattern must be found. When AsyncOS scans the message, it totals the "score" for the number of matches it finds in the message and attachments. If the minimum threshold is not met, the regular expression does not evaluate to true. You can specify this threshold for the following filter rules:

- body-contains
- only-body-contains
- attachment-contains
- every-attachment-contains
- dictionary-match
- attachment-dictionary-match

You can also specify a threshold value for the `drop-attachments-where-contains` action.



Note You cannot specify thresholds for filter rules that scan headers or envelope recipients and senders.

Threshold Syntax

To specify a threshold for the minimum number of occurrences, specify the pattern and the minimum number of matches required to evaluate to true:

```
if(<filter rule>('<pattern>',<minimum threshold>)){
```

For example, to specify that the `body-contains` filter rule must find the value "Company Confidential" at least two times, use the following syntax:

```
if(body-contains('Company Confidential',2)){
```

By defeat, when AsyncOS saves a content scanning filter, it compiles the filter and assigns a threshold value of 1, if you have not assigned a value.

You can also specify a minimum number of pattern matches for values in a content dictionary. For more information about content dictionaries, see the "Text Resources" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

Threshold Scoring for Message Bodies and Attachments

An email message may be composed of multiple parts. When you specify threshold values for filter rules that search for patterns in the message body or attachments, AsyncOS counts the number of matches in the message parts and attachments to determine the threshold "score." Unless the message filter specifies a specific MIME part (such as the `attachment-contains` filter rule), AsyncOS will total the matches found in all parts of the message to determine if the matches total the threshold value. For example, you have a `body-contains` message filter with a threshold of 2. You receive a message in which the body contains one match, and the attachment contains one match. When AsyncOS scores this message, it totals the two matches and determines that the threshold score has been met.

Similarly, if you have multiple attachments, AsyncOS totals the scores for each attachment to determine the score for matches. For example, you have an `attachment-contains` filter rule with a threshold of 3. You receive a message with two attachments, and each attachment contains two matches. AsyncOS would score this message with four matches and determine that the threshold score has been met.

Threshold Scoring Multipart/Alternative MIME Parts

To avoid duplicate counting, if there are two representatives of the same content (plain text and HTML), AsyncOS does not total the matches from the duplicate parts. Instead, it compares the matches in each part and selects the highest value. AsyncOS would then add this value to the scores from other parts of the multipart message to create a total score.

For example, you configure a `body-contains` filter rule and set the threshold to 4. You then receive a message that contains both plain text, HTML and two attachments. The message would use the following structure:

```
multipart/mixed

        multipart/alternative

                text/plain

                text/html

        application/octet-stream

        application/octet-stream
```

The `body-contains` filter rule would determine the score for this message by first scoring the text/plain and text/html parts of the message. It would then compare the results of these scores and select the highest score from the results. Next, it would add this result to the score from each of the attachments to determine the final score. Suppose the message has the following number of matches:

```
multipart/mixed

        multipart/alternative

                text/plain (2 matches)

                text/html (2 matches)

        application/octet-stream (1 match)

        application/octet-stream
```

Because AsyncOS compares the matches for the text/plain and text/html parts, it returns a score of 3, which does not meet the minimum threshold to trigger the filter rule.

Threshold Scoring for Content Dictionaries

When you use a content dictionary, you can “weight” terms so that certain terms trigger filter actions more easily. For example, you may want not want to trigger a message filter for the term, “bank.” However, if the term, “bank” is combined with the term, “account,” and accompanied with an ABA

routing number, you may want to trigger a filter action. To accomplish this, you can use a weighted dictionary to give increased importance to certain terms or a combination of terms. When a message filter that uses a content dictionary scores the matches for filter rule, it uses these weights to determine the final score. For example, suppose you create a content dictionary with the following contents and weights:

Table 6-1 Sample Content Dictionary

Term/Smart Identifier	Weight
ABA Routing Number	3
Account	2
Bank	1

When you associate this content dictionary with a `dictionary-match` or `attachment-dictionary-match` message filter rule, AsyncOS would add the weight for the term to the total "score" for each instance of the matching term found in the message. For example, if the message contains three instances of the term, "account" in the message body, AsyncOS would add a value of 6 to the total score. If you set the threshold value for the message filter to 6, AsyncOS would determine that the threshold score has been met. Or, if the message contained one instance of each term, the total value would be 6, and this score would trigger the filter action.

AND Test and OR Tests in Message Filters

When evaluating AND or OR tests within message filters, AsyncOS does not evaluate unneeded tests. So, for example, if one side of an AND test is false, the system will not evaluate the other side. It is important to note that the system does not evaluate the tests from left to right. Instead, when AND and OR tests are evaluated, the least expensive test is evaluated first. For example, in the following filter, the `remote-ip` test will always be processed first because it has a lower cost than the `rcpt-to-group` test (generally LDAP tests are more expensive):

```
andTestFilter:

if (remote-ip == "192.168.100.100" AND rcpt-to-group == "GROUP")

  { ... }
```

Because the least expensive test is performed first, switching the order of the items in the test will have no effect. If you want to guarantee the order in which tests are performed, use nested `if` statements. This is also the best way to ensure that an expensive test is avoided whenever possible:

```
expensiveAvoid:

if (<simple tests>)

  { if (<expensive test>)

    { <action> }

  }
```

In a somewhat more complicated example, consider:

```
if (test1 AND test2 AND test3) { ... }
```

The system groups the expression from left to right, so this becomes:

```
if ((test1 AND test2) AND test3) { ... }
```

This means the first thing the system does is compare the cost of `(test1 AND test2)` against the cost of `test3`, evaluating the second AND first. If all three tests have the same cost, then `test3` will be performed first because `(test1 AND test2)` would be twice as expensive.

Message Filter Rules

Each message filter contains a rule that defines the collection of messages that a filter can act upon. You define the filter rules, and then you define a filter action for messages that return `true`.

Filter Rules Summary Table

Table 6-2 summarizes the rules you can use in message filters.

Table 6-2 Message Filter Rules

Rule	Syntax	Description
Subject Header	`subject`	Does the subject header match a certain pattern? See Subject Rule, page 6-21.
Body Size	`body-size`	Is the body size within some range? See Body Size Rule, page 6-24.
Envelope Sender	`mail-from`	Does the Envelope Sender (i.e., the Envelope From, <MAIL FROM>) match a given pattern? See Envelope Sender Rule, page 6-22.
Envelope Sender in Group	`mail-from-group`	Is the Envelope Sender (i.e., the Envelope From, <MAIL FROM>) in a given LDAP group? See Envelope Sender in Group Rule, page 6-23.
Sender Group	`sendergroup`	Which sender group was matched in a listener's Host Access Table (HAT)? See Sender Group Rule, page 6-23.
Envelope Recipient	`rcpt-to`	Does the Envelope Recipient, (i.e. the Envelope To, <RCPT TO>) match a given pattern? See Envelope Recipient Rule, page 6-21. **Note:** The `rcpt-to` rule is message-based. If a message has multiple recipients, only one recipient has to match the rule for the specified action to affect the message to all recipients.

Table 6-2 Message Filter Rules

Rule	Syntax	Description
Envelope Recipient in Group	`rcpt-to-group`	Is the Envelope Recipient, (i.e. the Envelope To, <RCPT TO>) in a given LDAP group? See Envelope Recipient in Group Rule, page 6-22. **Note:** The `rcpt-to-group` rule is message-based. If a message has multiple recipients, only one recipient has to be found in a group for the specified action to affect the message to all recipients.
Remote IP	`remote-ip`	Was the message sent from a remote host that matches a given IP address or IP block? See Remote IP Rule, page 6-24.
Receiving Interface	`recv-int`	Did the message arrive via the named receiving interface? See Receiving IP Interface Rule, page 6-25.
Receiving Listener	`recv-listener`	Did the message arrive via the named listener? See Receiving Listener Rule, page 6-25.
Date	`date`	Is current time before or after a specific time and date? See Date Rule, page 6-25.
Header	`header(<string>)`	Does the message contain a specific header? Does the value of that header match a certain pattern? See Header Rule, page 6-26.
Random	`random(<integer>)`	Is a random number in some range? See Random Rule, page 6-27.
Recipient Count	`rcpt-count`	How many recipients is this email going to? See Recipient Count Rule, page 6-28.
Address Count	`addr-count()`	What is the cumulative number of recipients? This filter differs from the rcpt-count filter rule in that it operates on the message body headers instead of the envelope recipients. See Address Count Rule, page 6-28.
SPF Status	`spf-status`	What was the SPF verification status? This filter rule allows you to query for different SPF verification results. You can enter a different action for each valid SPF/SIDF return value. See SPF-Status Rule, page 6-35.
SPF Passed	`spf-passed`	Did the SPF/SIDF verification pass? This filter rule generalizes the SPF/SIDF results as a Boolean value. See SPF-Passed Rule, page 6-36.
Image verdict	`image-verdict`	What was the image scanning verdict? This filter rule allows you to query for different image analysis verdicts. See Image Analysis, page 6-67.

Table 6-2 Message Filter Rules

Rule	Syntax	Description
Workqueue count	`workqueue-count`	Is the work queue count equal to, less than, or greater than the specified value? See Workqueue-count Rule, page 6-37.
Body Scanning	`body-contains(<regular expression>)`	Does the message contain text or an attachment that matches a specified pattern? Does the pattern occur the minimum number of times you specified for the threshold value? The engine scans delivery-status parts and associated attachments. See Body Scanning Rule, page 6-28.
Body Scanning	`only-body-contains(<regular expression>)`	Does the message body contain text that matches a specified pattern? Does the pattern occur the minimum number of times you specified for the threshold value? Attachments are not scanned. See Body Scanning, page 6-29.
Encryption Detection	`encrypted`	Is the message encrypted? See Encryption Detection Rule, page 6-29.
Attachment Filename[a]	`attachment-filename`	Does the message contain an attachment with a filename that matches a specific pattern? See Attachment Filename Rule, page 6-30.
Attachment Type[a]	`attachment-type`	Does the message contain an attachment of a particular MIME type? See Attachment Type Rule, page 6-30.
Attachment File Type[a]	`attachment-filetype`	Does the message contain an attachment of a file type that matches a specific pattern based on its fingerprint (similar to a UNIX `file` command)? If the attachment is an Excel or Word document, you can also search for the following embedded file types: .exe , .dll, .bmp, .tiff, .pcx, .gif, .jpeg, png, and Photoshop images. You must enclose the file type in quotes to create a valid filter. You can use single or double quotes. For example, to search for .exe attachments, use the following syntax: `if (attachment-filetype == "exe")` For more information, see Examples of Attachment Scanning Message Filters, page 6-72.

Table 6-2 Message Filter Rules

Rule	Syntax	Description
Attachment MIME Type[a]	`attachment-mimetype`	Does the message contain an attachment of a specific MIME type? This rule is similar to the `attachment-type` rule, except only the MIME type given by the MIME attachment is evaluated. (The appliance does not try to "guess" the type of the file by its extension if there is no explicit type given.) See Examples of Attachment Scanning Message Filters, page 6-72.
Attachment Protected	`attachment-protected`	Does the message contain an attachment that is password protected? See Quarantining Protected Attachments, page 6-74.
Attachment Unprotected	`attachment-unprotected`	The attachment-unprotected filter condition returns true if the scanning engine detects an attachment that is unprotected. A file is considered unprotected if the scanning engine was able to read the attachment. A zip file is considered to be unprotected if any of its members is unprotected. **Note —** The attachment-unprotected filter condition is not mutually exclusive of the attachment-protected filter condition. It is possible for both filter conditions to return true when scanning the same attachment. This can occur, for example, if a zip file contains both protected and unprotected members. See Detecting Unprotected Attachments, page 6-75.
Attachment Scanning [a]	`attachment-contains(<regular expression>)`	Does the message contain an attachment that contains text or another attachment that matches a specific pattern? Does the pattern occur the minimum number of times you specified for the threshold value? This rule is similar to the `body-contains()` rule, but it attempts to avoid scanning the entire "body" of the message. That is, it attempts to scan only that which the user would view as being an attachment. See Examples of Attachment Scanning Message Filters, page 6-72.
Attachment Scanning	`attachment-binary-contains(<regular expression>)`	Does the message contain an attachment with binary data that matches a specific pattern? This rule is like the `attachment-contains ()` rule, but it searches specifically for patterns in the binary data.

Table 6-2 Message Filter Rules

Rule	Syntax	Description
Attachment Scanning	`every-attachment-contains(<regular expression>)`	Do all of the attachments in this message contain text that matches a specific pattern? The text must exist in all attachments and the action performed is, in effect, a logical AND operation of an '`attachment-contains()`' for each attachment. The body is not scanned. Does the pattern occur the minimum number of times you specified for the threshold value? See Examples of Attachment Scanning Message Filters, page 6-72.
Attachment Size[a]	`attachment-size`	Does the message contain an attachment whose size is within some range? This rule is similar to the `body-size` rule, but it attempts to avoid scanning the entire "body" of the message. That is, it attempts to scan only that which the user would view as being an attachment. The size is evaluated prior to any decoding. See Examples of Attachment Scanning Message Filters, page 6-72.
Public Blacklists	`dnslist(<query server>)`	Does the sender's IP address appear on a public blacklist server (RBL)? See DNS List Rule, page 6-31.
SenderBase Reputation	`reputation`	What is the sender's SenderBase Reputation Score? See SenderBase Reputation Rule, page 6-32.
No SenderBase Reputation	`no-reputation`	Used to test if SenderBase Reputation Score is "None." See SenderBase Reputation Rule, page 6-32.
Dictionary[b]	`dictionary-match(<dictionary_name>)`	Does the message body contain any of the regular expressions or terms in the content dictionary named *dictionary_name*? Does the pattern occur the minimum number of times you specified for the threshold value? See Dictionary Rules, page 6-33.
Attachment Dictionary Match	`attachment-dictionary-match(<dictionary_name>)`	Does the attachment contain any of the regular expressions in the content dictionary named *dictionary_name*? Does the pattern occur the minimum number of times you specified for the threshold value? See Dictionary Rules, page 6-33.
Subject Dictionary Match	`subject-dictionary-match(<dictionary_name>)`	Does the Subject header contain any of the regular expressions or terms in the content dictionary named *dictionary name*? See Dictionary Rules, page 6-33.

Table 6-2 Message Filter Rules

Rule	Syntax	Description
Header Dictionary Match	`header-dictionary-match(<dictionary_name>, <header>)`	Does the specified header (case insensitive) contain any of the regular expressions or terms in the content dictionary named *dictionary name*? See Dictionary Rules, page 6-33.
Body Dictionary Match	`body-dictionary-match(<dictionary_name>)`	This filter condition returns true if the dictionary term matches content in the body of the message only. The filter searches for terms within the MIME parts not considered to be an attachment. and it returns true if the user-defined threshold is met (the default threshold value is one). See Dictionary Rules, page 6-33.
Envelope Recipient Dictionary Match	`rcpt-to-dictionary-match(<dictionary_name>)`	Does the envelope recipient contain any of the regular expressions or terms in the content dictionary named *dictionary name*? See Dictionary Rules, page 6-33.
Envelope Sender Dictionary Match	`mail-from-dictionary-match(<dictionary_name>)`	Does the envelope sender contain any of the regular expressions or terms in the content dictionary named *dictionary name*? See Dictionary Rules, page 6-33.
SMTP Authenticated User Match	`smtp-auth-id-matches(<target> [, <sieve-char>])`	Does the address of the Envelope Sender and the address in message header match the authenticated SMTP user ID of the sender? See SMTP Authenticated User Match Rule, page 6-37.
True	`true`	Matches all messages. See True Rule, page 6-20.
Valid	`valid`	Returns false if the message contains unparsable/invalid MIME parts and true otherwise. See Valid Rule, page 6-20.
Signed	`signed`	Is the message is signed? See Signed Rule, page 6-39.
Signed Certificate	`signed-certificate(<field> [<operator> <regular expression>])`	Does the message signer or X.509 certificate issuer match a certain pattern? See Signed Certificate Rule, page 6-39.

a.Attachment filtering is discussed in detail in the section Attachment Scanning, page 6-65.
b.Content Dictionaries are discussed in the detail in the "Text Resources" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

Each message injected into the Cisco IronPort appliance is processed through all message filters in order, unless you specify a final action, which stops the message from being processed further. (See Message Filter Actions, page 6-2.) Filters may also apply to all messages, and rules may also be combined using logical connectors (AND, OR, NOT).

Regular Expressions in Rules

Several of the atomic tests used to define rules use *regular expression matching*. Regular expressions can become complex. Use the following table as a guide for the applying of regular expressions within message filter rules:

Table 6-3 Regular Expression in Rules

Regular expression (`abc`)	Regular expressions in filter rules match a string if the sequence of directives in the regular expression match any part of the string. For example, the regular expression `Georg` matches the string `George Of The Jungle`, the string `Georgy Porgy`, the string `La Meson Georgette` as well as `Georg`.
Carat (`^`) **Dollar sign ($)**	Rules containing the dollar sign character ($) only match the end of the string, and rules containing the caret symbol (`^`) only match the beginning of the string. For example, the regular expression `^Georg$` only matches the string `Georg`. Searching for an empty header would look like this: `"^$"`
Letters, white space and the at sign (@) character	Rules containing characters, white space, and the at sign character (`@`) only match themselves explicitly. For example, the regular expression `^George@admin$` only matches the string `George@admin`.
Period character (`.`)	Rules containing a period character (`.`) match any character (except a new line). For example, the regular expression `^...admin$` matches the string `macadmin` as well as the string `sunadmin` but not `win32admin`.
Asterisk (`*`) directive	Rules containing an asterisk (`*`) match "zero or more matches of the previous directive." In particular, the sequence of a period and an asterisk (`.*`) matches any sequence of characters (not containing a new line). For example, the regular expression `^P.*Piper$` matches all of these strings: `PPiper`, `Peter Piper`, `P.Piper`, and `Penelope Penny Piper`.
Backslash special characters (`\`)	The backslash character *escapes* special characters. Thus the sequence `\.` only matches a literal period, the sequence `\$` only matches a literal dollar sign, and the sequence `\^` only matches a literal caret symbol. For example, the regular expression `^ik\.ac\.uk$` only matches the string `ik.ac.uk`. **Important Note:** The backslash is also a special escape character for the parser. As a result, if you want to include backslash in your regular expression, you must use *two* backslashes — so that after parsing, only one "real" backslash remains, which is then passed to the regular expression system. So, if you wanted to match the example domain above, you would enter `^ik\\.ac\\.uk$`.

Table 6-3 Regular Expression in Rules

Case-insensitivity (`(?i)`)	The token `(?i)` that indicates the rest of the regular expression should be treated in case-insensitive mode. Placing this token at the beginning of a case-sensitive regular expression results in a completely insensitive match. For example, the regular expression "`(?i)viagra`" matches `Viagra`, `vIaGrA`, and `VIAGRA`.
Number of repetitions `{min,max}`	The regular expression notation that indicates the number of repetitions of the previous token is supported. For example, the expression "`fo{2,3}`" matches `foo` and `fooo` but not `fo` or `fofo`. This statement: `if(header('To') == "^.{500,}")` looks for a "To" header that has 500 or more characters in it.
Or (`|`)	Alternation, or the "or" operator. If A and B are regular expressions, the expression "`A|B`" will match any string that matches either "A" or "B." For example, the expression "`foo|bar`" will match either `foo` or `bar`, but not `foobar`.

Using Regular Expressions to Filter Messages

You can use filters to search for strings and patterns in non-ASCII encoded message content (both headers and bodies). Specifically, the system supports regular expression (regex) searching for non-ASCII character sets within:

- Message headers
- MIME attachment filename strings
- Message body:
  - Bodies without MIME headers (i.e. traditional email)
  - Bodies with MIME headers indicating encoding but no MIME parts
  - Multi-part MIME messages with encoding indicated
  - All of the above without the encoding specified in a MIME header

You can use regular expressions (regexes) to match on any part of the message or body, including matching attachments. The various attachment types include text, HTML, MS Word, Excel, and others. Examples of character sets of interest include gb2312, HZ, EUC, JIS, Shift-JIS, Big5, and Unicode. Message filter rules with regular expressions can be created through the content filter GUI (see "Email Security Manager" in the *Cisco IronPort AsyncOS for Email Configuration Guide*), or using a text editor to generate a file that is then imported into the system. For more information, see Using the CLI to Manage Message Filters, page 6-75 and Modifying Scanning Parameters, page 6-84.

Guidelines for Using Regular Expressions

It is important to begin a regular expression with a caret (`^`) and end it with a dollar sign (`$`) whenever you want to exactly match a string and not a prefix.

Note When matching an empty string, do not use `""` as that actually matches *all* strings. Instead use `"^$"`. For an example, see the second example in Subject Rule, page 6-21.

It is also important to remember that if you want to match a literal period, you must use an escaped period in the regular expression. For example, the regular expression `sun.com` matches the string `thegodsunocommando`, but the regular expression `^sun\.com$` only matched the string `sun.com`.

Technically, the style of regular expressions used are **Python re Module** style regular expressions. For a more detailed discussion of Python style regular expressions, consult the Python Regular Expression HOWTO, accessible from:
`http://www.python.org/doc/howto/`

Regular Expression and Non-ASCII Character Sets

In some languages, the concepts of a word or word boundary, or case do not exist.

Complex regular expressions that depend on concepts like what is or is not a character that would compose a word (represented as "`\w`" in regex syntax) cause problems when the locale is unknown or if the encoding is not known for certain.

n Tests

Regular expressions can be tested for matching using the sequence `==` and for non-matching using the sequence `!=`. For example:

```
rcpt-to == "^goober@dev\\.null\\....$" (matching)


rcpt-to != "^goober@dev\\.null\\....$" (non-matching)
```

Case-sensitivity

Unless otherwise noted, regular expressions are case-sensitive. Thus, if your regular expression is searching for `foo`, it does not match the pattern `FOO` or even `Foo`.

Writing Efficient Filters

This example shows two filters that do the same thing, but the first one takes much more CPU. The second filter uses a regular expression that is more efficient.

```
attachment-filter: if ((recv-listener == "Inbound") AND
((((((((((((((((((((((((((((((((((((((((((((((((((((((attachment-filename ==

"\\.386$") OR (attachment-filename == "\\.exe$")) OR (attachment-filename == "\\.ad$"))
OR (attachment-filename == "\\.ade$")) OR (attachment-filename == "\\.adp$")) OR
(attachment-filename == "\\.asp$")) OR (attachment-filename == "\\.bas$")) OR
(attachment-filename == "\\.bat$")) OR (attachment-filename == "\\.chm$")) OR
(attachment-filename == "\\.cmd$")) OR (attachment-filename == "\\.com$")) OR
(attachment-filename == "\\.cpl$")) OR (attachment-filename == "\\.crt$")) OR
(attachment-filename == "\\.exe$")) OR (attachment-filename == "\\.hlp$")) OR
(attachment-filename == "\\.hta$")) OR (attachment-filename == "\\.inf$")) OR
(attachment-filename == "\\.ins$")) OR (attachment- filename == "\\.isp$")) OR
(attachment-filename == "\\.js$")) OR (attachment-filename == "\\.jse$")) OR
(attachment- filename == "\\.lnk$")) OR (attachment-filename == "\\.mdb$")) OR
(attachment-filename == "\\.mde$")) OR (attachment-filename == "\\.msc$")) OR
(attachment-filename == "\\.msi$")) OR (attachment-filename == "\\.msp$")) OR
(attachment-filename == "\\.mst$")) OR (attachment-filename == "\\.pcd$")) OR
(attachment-filename == "\\.pif$")) OR (attachment-filename == "\\.reg$")) OR
(attachment-filename == "\\.scr$")) OR (attachment-filename == "\\.sct$")) OR
(attachment-filename == "\\.shb$")) OR (attachment-filename == "\\.shs$")) OR
(attachment-filename == "\\.url$")) OR (attachment-filename == "\\.vb$")) OR
(attachment-filename == "\\.vbe$")) OR (attachment-filename == "\\.vbs$")) OR
(attachment-filename == "\\.vss$")) OR (attachment-filename == "\\.vst$")) OR
(attachment-filename == "\\.vsw$")) OR (attachment-filename == "\\.ws$")) OR
(attachment-filename == "\\.wsc$")) OR (attachment-filename == "\\.wsf$")) OR
(attachment-filename == "\\.wsh$"))) { bounce(); }
```

In this instance, AsyncOS will have to start the regular expression engine 30 times, once for each attachment type and the recv-listener.

Instead, write the filter to look like this:

```
attachment-filter: if (recv-listener == "Inbound") AND (attachment-filename ==
"\\.(386|exe|ad|ade|adp|asp|bas|bat|chm|cmd|com|cpl|crt|exe|hlp|hta|inf|ins|isp|js|jse|l
nk|mdb|mde|msc|msi|msp|mst|pcd|pif|reg|scr|sct|shb|shs|url|vb|vbe|vbs|vss|vst|vsw|ws|wsc
|wsf|wsh)$") {

    bounce();

  }
```

The regular expression engine only has to start twice and the filter is arguably easier to maintain as you do not have to worry about adding “()”, spelling errors. In contrast to the above, this should show a decrease in CPU overhead.

PDFs and Regular Expressions

Depending on how a PDF is generated, it may contain no spaces or line breaks. When this occurs, the scanning engine attempts to insert logical spaces and line breaks based on the location of the words on the page. For example, when a word is constructed using multiple fonts or font sizes, the PDF code is

rendered in a way that makes it difficult for the scanning engine to determine word and line breaks. When you attempt to match a regular expression against a PDF file constructed in this way, the scanning engine may return unexpected results.

For example, you enter a word in a PowerPoint document that uses different fonts and different font sizes for each letter in the word. When a scanning engine reads a PDF generated from this application, it inserts logical spaces and line breaks. Because of the construction of the PDF, it may interpret the word "callout" as "call out" or "c a l lout." Attempting to match either of these renderings against the regular expression, "callout," would result in no matches.

Smart Identifiers

When you use message rules that scan message content, you can use smart identifiers to detect certain patterns in the data.

Smart identifiers can detect the following patterns in data:

- Credit card numbers
- U.S. Social Security numbers
- Committee on Uniform Security Identification Procedures (CUSIP) numbers
- American Banking Association (ABA) routing numbers

To use smart identifiers in a filter, enter the following keywords in a filter rule that scans body or attachment content:

Table 6-4 Smart Identifiers in Message Filters

Key Word	Smart Identifier	Description
*credit	Credit card number	Identifies 14-, 15-, and 16- digit credit card numbers. NOTE: The smart identifier does not identify enRoute or JCB cards.
*aba	ABA routing number	Identifies ABA routing numbers.
*ssn	Social security number	Identifies U.S. social security numbers. The *ssn smart identifier identifies social security numbers with dashes, periods and spaces.
*cusip	CUSIP number	Identifies CUSIP numbers.

Smart Identifier Syntax

When you use a smart identifier in a filter rule, enter the smart-identifier keyword in quotes within a filter rule that scans the body or attachment file, as in the example below:

```
ID_Credit_Cards:

if(body-contains("*credit")){
```

```
notify("legaldept@example.com");

}
.
```

You can also use smart identifiers in content filters and as a part of content dictionaries.

Note You cannot combine a smart identifier key word with a normal regular expression or another key word. For example the pattern `*credit|*ssn` would not be valid.

Note To minimize on false positives using the *SSN smart identifier, it may be helpful to use the *ssn smart identifier along with other filter criteria. One example filter that can be used is the "only-body-contains" filter condition. This will only evaluate the expression to be true if the search string is present in all of the message body mime parts. For example, you could create the following filter:

```
SSN-nohtml: if only-body-contains("*ssn") { duplicate-quarantine("Policy");}
```

Examples of Message Filter Rules

The following section shows examples of message filter rules in use.

True Rule

The `true` rule matches all messages. For example, the following rule changes the IP interface to external for all messages it tests.

```
externalFilter:

   if (true)

   {

        alt-src-host('external');

   }
```

Valid Rule

The `valid` rule returns false if the message contains unparsable/invalid MIME parts and true otherwise. For example, the following rule drops all unparsable messages it tests.

```
not-valid-mime:

   if not valid

   {
```

```
        drop();

    }
```

Subject Rule

The `subject` rule selects those messages where the value of the subject header matches the given regular expression.

For example, the following filter discards all messages with subjects that start with the phrase `Make Money...`

```
scamFilter:

    if (subject == '^Make Money')

    {

        drop();

    }
```

You can specify non-ASCII characters to search for in the value of the header.

When working with headers, remember that the current value of the header includes changes made during processing (such as with filter actions that add, remove, or modify message headings). See Message Header Rules and Evaluation, page 6-4 for more information.

The following filter returns true if the headers are empty or if the headers are missing from the message:

```
EmptySubject_To_filter:

if (header('Subject') != ".") OR

    (header('To') != ".") {

    drop();

}
```

Note This filter returns true for empty Subject and To headers, but it also returns true for missing headers. If the message does not contain the specified headers, the filter still returns true.

Envelope Recipient Rule

The `rcpt-to` rule selects those messages where any Envelope Recipient matches the given regular expression. For example, the following filter drops all messages sent with an email address containing the string “scarface.”

Note The regular expression for the `rcpt-to` rule is case *insensitive*.

```
scarfaceFilter:

   if (rcpt-to == 'scarface')

   {

        drop();

   }
```

Note The `rcpt-to` rule is message-based. If a message has multiple recipients, only one recipient has to match the rule for the specified action to affect the message to all recipients.

Envelope Recipient in Group Rule

The `rcpt-to-group` rule selects those messages where any Envelope Recipient is found to be a member of the LDAP group given. For example, the following filter drops all messages sent with an email address within the LDAP group "ExpiredAccounts."

```
expiredFilter:

   if (rcpt-to-group == 'ExpiredAccounts')

   {

        drop();

   }
```

Note The `rcpt-to-group` rule is message-based. If a message has multiple recipients, only one recipient has to match the rule for the specified action to affect the message to all recipients.

Envelope Sender Rule

The `mail-from` rule selects those messages where the Envelope Sender matches the given regular expression. For example, the following filter immediately delivers any message sent by `admin@yourdomain.com`.

Note The regular expression for the `mail-from` rule is case *insensitive*. Note that the period character is escaped in the following example.

```
kremFilter:

   if (mail-from == '^admin@yourdomain\\.com$')

   {

 skip_filters();

   }
```

Envelope Sender in Group Rule

The `mail-from-group` rule selects those messages where the Envelope Sender is found to be in an LDAP group given on the right side of the operator (or, in the case of inequality, where the sender's email address is *not* in the given LDAP group). For example, the following filter immediately delivers any message sent by someone whose email address is in the LDAP group "KnownSenders."

```
SenderLDAPGroupFilter:

   if (mail-from-group == 'KnownSenders')

   {

 skip_filters();

   }
```

Sender Group Rule

The `sendergroup` message filter selects a message based on which sender group was matched in a listener's Host Access Table (HAT). This rule uses '==' (for matching) or '!=' (for not matching) to test for matching a given regular expression (the right side of the expression). For example, the following message filter rule evaluates to `true` if the sender group of the message matches the regular expression Internal, and, if so, sends the message to an alternate mail host.

```
senderGroupFilter:

      if (sendergroup == "Internal")

      {

        alt-mailhost("[172.17.0.1]");

      }
```

Body Size Rule

Body size refers to the size of the message, including both headers and attachments. The `body-size` rule selects those messages where the body size compares as directed to a given number. For example, the following filter bounces any message where the body size is greater than 5 megabytes.

```
BigFilter:

   if (body-size > 5M)

   {

        bounce();

   }
```

The `body-size` can be compared in the following ways:

Example	Comparison Type
`body-size < 10M`	Less than
`body-size <= 10M`	Less than or equal
`body-size > 10M`	Greater than
`body-size >= 10M`	Greater than or equal
`body-size == 10M`	Equal
`body-size != 10M`	Not equal

As a convenience, the size measurement may be specified with a suffix:

Quantity	Description
`10b`	ten bytes (same as 10)
`13k`	thirteen kilobytes
`5M`	five megabytes
`40G`	40 gigabytes (Note: The Cisco IronPort appliance cannot accept messages larger than 100 megabytes.)

Remote IP Rule

The `remote-ip` rule tests to see if the IP address of the host that sent that message matches a certain pattern. The IP address can be either Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6). The IP address pattern is specified using the **allowed hosts** notation described in "Sender Group Syntax" in the *Cisco IronPort AsyncOS for Email Configuration Guide*, except for the `SBO`, `SBRS`, `dnslist` notations and the special keyword `ALL`.

The allowed hosts notation can only identify sequences and numeric ranges of IP addresses (not hostnames). For example, the following filter bounces any message *not* injected from IP addresses of form `10.1.1.x` where `x` is `50`, `51`, `52`, `53`, `54`, or `55`.

```
notMineFilter:

   if (remote-ip != '10.1.1.50-55')

   {

        bounce();

   }
```

Receiving Listener Rule

The `recv-listener` rule selects those messages received on the named listener. The listener name must be the nickname of one of the listeners currently configured on the system. For example, the following filter immediately delivers any message arriving from the listener named `expedite`.

```
expediteFilter:

   if (recv-listener == 'expedite')

   {

 skip_filters();

   }
```

Receiving IP Interface Rule

The `recv-int` rule selects those messages received via the named interface. The interface name must be the nickname of one of the interfaces currently configured for the system. For example, the following filter bounces any message arriving from the interface named `outside`.

```
outsideFilter:

   if (recv-int == 'outside')

   {

        bounce();

   }
```

Date Rule

The `date` rule checks the current time and date against a time and date you specify. The date rule is compares against a string containing a timestamp of the format *MM/DD/YYYY hh:mm:ss*. This is useful to specify actions to be performed before or after certain times

in US format. (Note that there may be an issue if you are searching messages with non-US date formats.) the following filter bounces all messages from `campaign1@yourdomain.com` that are injected after 1:00pm on July 28th, 2003:

```
TimeOutFilter:

   if ((date > '07/28/2003 13:00:00') and (mail-from ==

      'campaign1@yourdomain\\.com'))

   {

        bounce();

   }
```



Note Do not confuse the `date` rule with the `$Date` message filter action variable.

Header Rule

The `header()` rule checks the message headers for a specific header, which must be specified quoted in parentheses ("*header name*"). This rule may be compared to a regular expression, much like the `subject` rule, or may be used without any comparison, in which case it will be "true" if the header is found in the message, and "false" if it is not found. For example, the following example checks to see if the header `X-Sample` is found, and if its value contains the string "`sample text`". If a match is made, the message is bounced.

```
FooHeaderFilter:

   if (header('X-Sample') == 'sample text')

   {

        bounce();

   }
```

You can specify non-ASCII characters to search for in the value of the header.

The following example demonstrates the header rule without a comparison. In this case, if the header `X-DeleteMe` is found, it is removed from the message.

```
DeleteMeHeaderFilter:

   if header('X-DeleteMe')

   {
```

```
        strip-header('X-DeleteMe');

    }
```

When working with headers, remember that the current value of the header includes changes made during processing (such as with filter actions that add, remove, or modify message headings). See Message Header Rules and Evaluation, page 6-4 for more information.

Random Rule

The `random` rule generates a random number from zero to N-1, where N is the integer value supplied in parenthesis after the rule. Like the `header()` rule, this rule may be used in a comparison, or may be used alone in a "unary" form. The rule evaluates to `true` in the unary form if the random number generated is non-zero. For example, both of the following filters are effectively equal, choosing Virtual Gateway address A half the time, and Virtual Gateway address B the other half of the time:

```
load_balance_a:

    if (random(10) < 5) {

        alt-src-host('interface_a');

    } else {

        alt-src-host('interface_b');

    }



load_balance_b:

    if (random(2)) {

        alt-src-host('interface_a');

    } else {

        alt-src-host('interface_b');

    }
```

Recipient Count Rule

The `rcpt-count` rule compares the number of recipients of a message against an integer value, in a similar way to the `body-size` rule. This can be useful for preventing users from sending email to large numbers of recipients at once, or for ensuring that such large mailing campaigns go out over a certain Virtual Gateway address. The following example sends any email with more than 100 recipients over a specific Virtual Gateway address:

```
large_list_filter:

   if (rcpt-count > 100) {

        alt-src-host('mass_mailing_interface');

   }
```

Address Count Rule

The `addr-count()` message filter rule takes one or more header strings, counts the number of recipients in each line and reports the cumulative number of recipients. This filter differs from the `rcpt-count` filter rule in that it operates on the message body headers instead of the envelope recipients. The following example shows the filter rule used to replace a long list of recipients with the alias, “undisclosed-recipients”:

```
count: if (addr-count("To", "Cc") > 30) {

            strip-header("To");

            strip-header("Cc");

            insert-header("To", "undisclosed-recipients");

           }
```

Body Scanning Rule

The `body-contains()` rule scans the incoming email and all its attachments for a particular pattern defined by its parameter. This includes delivery-status parts and associated attachments. The `body-contains()` rule does not perform multi-line matching. The scanning logic can be modified using the `scanconfig` command in the CLI to define which MIME types should or should not be scanned. You can also specify a minimum number of matches that the scanning engine must find in order for the scan to evaluate to true.

By default, the system scans all attachments *except* for those with a MIME type of `video/*`, `audio/*`, `image/*`. The system scans archive attachments — `.zip`, `.bzip`, `.compress`, `.tar`, or `.gzip` attachments containing multiple files. You can set the number of “nested” archived attachments to scan (for example, a `.zip` contained within a `.zip`).

For more information, including an example of how to use the `scanconfig` command to set the attachment scanning behavior, see Modifying Scanning Parameters, page 6-84.

Body Scanning

When AsyncOS performs body scanning, it scans the body text and attachments for the regular expression. You can assign a minimum threshold value for the expression, and if the scanning engine encounters the regular expression the minimum number of times, the expression evaluates to `true`.

AsyncOS evaluates the different MIME parts of the message, and it scans any MIME part that is textual. AsyncOS identifies the text parts if the MIME type specifies text in the first part. AsyncOS determines the encoding based on the encoding specified in the message, and it converts the text to Unicode. It then searches for the regular expression in Unicode space. If no encoding is specified in the message, AsyncOS uses the encoding you specify in the `scanconfig` command.

For more information about how AsyncOS evaluates MIME parts when scanning messages, see Message Bodies vs. Message Attachments, page 6-5.

If the MIME part is not textual, AsyncOS extract files from a .zip or .tar archive or decompresses compressed files. After extracting the data, a scanning engine identifies the encoding for the file and returns the data from the file in Unicode. AsyncOS then searches for the regular expression in Unicode space.

The following example searches the body text and attachment for the phrase “Company Confidential.” The example specifies a minimum threshold of two instances, so if the scanning engine finds two or more instances of the phrase, it bounces any matching messages, and notifies the legal department of the attempt:

```
ConfidentialFilter:

   if (body-contains('Company Confidential',2)) {

        notify ('legaldept@example.domain');

        bounce();

   }
```

To scan only the body of the message, use `only-body-contains`:

```
disclaimer:

    if (not only-body-contains('[dD]isclaimer',1) ) {

        notify('hresource@example.com');

    }
```

Encryption Detection Rule

The `encrypted` rule examines the contents of a message for encrypted data. It does not attempt to decode the encrypted data, but merely examines the contents of the message for the existence of encrypted data. This can be useful for preventing users from sending encrypted email.

Note The encrypted rule can only detect encrypted data in the content of messages. It does not detect encrypted attachments.

The `encrypted` rule is similar to the `true` rule in that it takes no parameters and cannot be compared. This rule returns `true` if encrypted data is found and `false` if no encrypted data is found. Because this function requires the message to be scanned, it uses the scanning settings you define in the `scanconfig` command. For more information about configuring these options, see Modifying Scanning Parameters, page 6-84.

The following filter checks all email sent through the listener, and if a message contains encrypted data, the message is blind-carbon-copied to the legal department and then bounced:

```
prevent_encrypted_data:

    if (encrypted) {

         bcc ('legaldept@example.domain');

         bounce();

      }
```

Attachment Type Rule

The `attachment-type` rule checks the MIME types of each attachment in a message to see if it matches the given pattern. The pattern must be of the same form used in the `scanconfig` command as described in Modifying Scanning Parameters, page 6-84, and so may have either side of the slash (`/`) replaced by an asterisk as a wildcard. If the message contains an attachment that matches this specified MIME type, this rule returns “true.”

Because this function requires the message to be scanned, it obeys all of the options defined by the `scanconfig` command as described in Modifying Scanning Parameters, page 6-84.

See Attachment Scanning, page 6-65 for more information on message filter rules you can use to manipulate attachments to messages.

The following filter checks all email sent through the listener, and if a message contains an attachment with a MIME type of `video/*`, the message is bounced:

```
bounce_video_clips:

   if (attachment-type == 'video/*') {

        bounce();

   }
```

Attachment Filename Rule

The `attachment-filename` rule checks the filenames of each attachment in a message to see if it matches the given regular expression. This comparison is case-sensitive. The comparison is, however sensitive to whitespace so if the filename has encoded whitespace at the end, the filter will skip the attachment. If one of the message’s attachments matches the filename, this rule returns “true.”

Please note the following points:

- Each attachment’s filename is captured from the MIME headers. The filename in the MIME header may contain trailing spaces.

- If an attachment is an archive, the Cisco IronPort appliance will harvest the filenames from inside the archive and apply `scanconfig` rules (see Modifying Scanning Parameters, page 6-84) accordingly.
  - If the attachment is a single compressed file (despite the file extension), it is not considered an archive and the filename of the compressed file is not harvested. This means that the file is not processed by the `attachment-filename` rule. An example of this type of file is an executable file (.exe) compressed with `gzip`.
  - For attachments consisting of a single compressed file, such as foo.exe.gz, use regular expression to search for specific file types within compressed files. See Attachment Filenames and Single Compressed Files within Archive Files, page 6-31.

See Attachment Scanning, page 6-65 for more information on message filter rules you can use to manipulate attachments to messages.

The following filter checks all email sent through the listener, and if a message contains an attachment with a filename `*.mp3`, the message is bounced:

```
block_mp3s:

   if (attachment-filename == '(?i)\\.mp3$') {

        bounce();

   }
```

Attachment Filenames and Single Compressed Files within Archive Files

This example shows how to match single compressed files in archives such as those created by `gzip`:

```
quarantine_gzipped_exe_or_pif:

if (attachment-filename == '(?i)\\.(exe|pif)($|.gz$)') {

  quarantine("Policy");

}
```

DNS List Rule

The `dnslist()` rule queries a public DNS List server that uses the DNSBL method (sometimes called "ip4r lookups") of querying. The IP address of the incoming connection is reversed (so an IP of 1.2.3.4 becomes 4.3.2.1) and then added as a prefix to the server name in the parenthesis (a period to separate the two is added if the server name does not start with one). A DNS query is made, and the system is returned with either a DNS failure response (indicating the connection's IP address was not found in the server's list) or an IP address (indicating that the address was found). The IP address returned is *usually* of the form `127.0.0.`*x* where `x` can be almost any number from 0 to 255 (IP address ranges are not allowed). Some servers actually return different numbers based on the reason for the listing, while others return the same result for all matches.

Like the `header()` rule, `dnslist()` can be used in either a unary or binary comparison. By itself, it simply evaluates to `true` if a response is received and `false` if no response is received (for example, if the DNS server is unreachable).

the following filter immediately delivers a message if the sender has been bonded with the Cisco IronPort Bonded Sender information services program:

```
whitelist_bondedsender:

   if (dnslist('query.bondedsender.org')) {

 skip_filters();

   }
```

Optionally, you can compare the result to a string using the equality (==) or inequality (!=) expressions.

The following filter drops a message that results in a "127.0.0.2" response from the server. If the response is anything else, the rule returns "false" and the filter is ignored.

```
blacklist:

   if (dnslist('dnsbl.example.domain') == '127.0.0.2') {

        drop();

   }
```

SenderBase Reputation Rule

The `reputation` rule checks the SenderBase Reputation Score against another value. All the comparison operators are allowed, such as >, ==, <=, and so forth. If the message does not have a SenderBase Reputation Score at all (because one was never checked for it, or because the system failed to get a response from the SenderBase Reputation Service query server), any comparison against a reputation fails (the number will not be greater than, less than, equal to, or not equal to any value). You can check for a SBRS score of "none" using the `no-reputation` rule described below. The following example adjusts the "Subject:" line of a message to be prefixed by "`*** BadRep ***`" if the reputation score returned from the SenderBase Reputation Service is below a threshold of -7.5..

```
note_bad_reps:

   if (reputation < -7.5) {

        strip-header ('Subject');

        insert-header ('Subject', '*** BadRep $Reputation *** $Subject');

   }
```

For more information, see "Reputation Filtering" and "SenderBase Reputation Score (SBRS)" in the *Cisco IronPort AsyncOS for Email Configuration Guide*. See also Bypass Anti-Spam System Action, page 6-63

Values for the SenderBase Reputation rule are -10 through 10, but the value `NONE` may also be returned. To check specifically for the value `NONE`, use the `no-reputation` rule.

```
none_rep:

   if (no-reputation) {

       strip-header ('Subject');

        insert-header ('Subject', '*** Reputation = NONE *** $Subject');

   }
```

Dictionary Rules

The `dictionary-match(<dictonary_name>)` rule evaluates to `true` if the message body contains any of the regular expressions or terms in the content dictionary named "*dictonary_name*." If the dictionary does not exist, the rule evaluates to `false`. For more information on defining dictionaries (including their case sensitivity and word boundary settings), see the "Text Resources" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

The following filter blind carbon copies the administrator when the Cisco IronPort scans a message that contains any words within the dictionary named "secret_words."

```
copy_codenames:

   if (dictionary-match ('secret_words')) {

       bcc('administrator@example.com');

       }
```

The following example sends the message to the Policy quarantine if the message body contains any words within the dictionary named "secret_words." Unlike the `only-body-contains` condition, the `body-dictionary-match` condition does not require that all the content parts individually match the dictionary. The scores of each content part (taking into account multipart/alternative parts) are added together.

```
quarantine_data_loss_prevention:

   if (body-dictionary-match ('secret_words'))

       {

       quarantine('Policy');

       }
```

In the following filter, a subject that matches a term in the specified dictionary is quarantined:

```
quarantine_policy_subject:

    if (subject-dictionary-match ('gTest'))
```

```
        {
        quarantine('Policy');
        }
```

This example matches an email address in the “to” header and blind copies an administrator:

```
headerTest:
    if (header-dictionary-match ('competitorsList', 'to'))
        {
        bcc('administrator@example.com');
        }
```

The `attachment-dictionary-match(<dictonary_name>)` rule works like the `dictionary-match` rule above, except that it looks for matches in the attachment.

The following filter sends the message to the Policy quarantine if the message attachment contains any words found within the dictionary named “secret_words.”

```
quarantine_codenames_attachment:
   if (attachment-dictionary-match ('secret_words'))
       {
       quarantine('Policy');
       }
```

The `header-dictionary-match(<dictonary_name>, <header>)` rule works like the dictionary-match rule above, except that it looks for matches in the header specified in <*header*>. The header name is case insensitive, so, for example, “subject” and “Subject” both work.

The following filter sends the message to the Policy quarantine if the message’s “cc” header contains any words found within the dictionary named “ex_employees.”

```
quarantine_codenames_attachment:
   if (header-dictionary-match ('ex_employees', 'cc'))
       {
       quarantine('Policy');
       }
```

You can use wild cards within the dictionary terms. You do not have to escape the period in email addresses.

SPF-Status Rule

When you receive SPF/SIDF verified mail, you may want to take different actions depending on the results of the SPF/SIDF verification. The spf-status rule checks against different SPF verification results. For more information, see Verification Results, page 5-31.

You can check against the SPF/SIDF verification results using the following syntax:

```
if (spf-status == "Pass")
```

If you want a single condition to check against multiple status verdicts, you can use the following syntax:

```
if (spf-status == "PermError, TempError")
```

You can also check the verification results against the HELO, MAIL FROM, and PRA identities using the following syntax:

```
if (spf-status("pra") == "Fail")
```

The following example shows the `spf-status` filter in use:

```
skip-spam-check-for-verified-senders:

     if (sendergroup == "TRUSTED" and spf-status == "Pass"){

         skip-spamcheck();

     }



quarantine-spf-failed-mail:

     if (spf-status("pra") == "Fail") {

         if (spf-status("mailfrom") == "Fail"){

             # completely malicious mail

             quarantine("Policy");

         } else {

           if(spf-status("mailfrom") == "SoftFail") {

             # malicious mail, but tempting

             quarantine("Policy");

           }

         }
```

```
    } else {
      if(spf-status("pra") == "SoftFail"){
        if (spf-status("mailfrom") == "Fail"
                or spf-status("mailfrom") == "SoftFail"){
            # malicious mail, but tempting
            quarantine("Policy");
        }
      }
    }


stamp-mail-with-spf-verification-error:
    if (spf-status("pra") == "PermError, TempError"
            or spf-status("mailfrom") == "PermError, TempError"
            or spf-status("helo") == "PermError, TempError"){
        # permanent error - stamp message subject
        strip-header("Subject");
        insert-header("Subject", "[POTENTIAL PHISHING] $Subject"); }
.
```

SPF-Passed Rule

The following example shows an `spf-passed` rule used to quarantine emails that are not marked as spf-passed:

```
quarantine-spf-unauthorized-mail:
    if (not spf-passed) {
        quarantine("Policy");
    }
```

Note Unlike the `spf-status` rule, the `spf-passed` rule reduces the SPF/SIDF verification values to a simple Boolean. The following verification results are treated as not passed in the `spf-passed` rule: None, Neutral, Softfail, TempError, PermError, and Fail. To perform actions on messages based on more granular results, use the `spf-status` rule.

Workqueue-count Rule

The `workqueue-count` rule checks the workqueue-count against a specified value. All the comparison operators are allowed, such as `>`, `==`, `<=`, and so forth.

The following filter checks the workqueue count, and skips spamcheck if the queue is greater than the specified number.

```
 wqfull:

if (workqueue-count > 1000) {

 skip-spamcheck();

}
```

For more information on SPF/SIDF, see Overview of SPF and SIDF Verification, page 5-22.

SMTP Authenticated User Match Rule

If your Cisco IronPort appliance uses SMTP authentication to send messages, the `smtp-auth-id-matches` (*<target> [, <sieve-char>]*)rule can check a message's headers and Envelope Sender against the sender's SMTP authenticated user ID to identify outgoing messages with spoofed headers. This filter allows the system to quarantine or block potentially spoofed messages.

The `smtp-auth-id-matches` rule compares the SMTP authenticated ID against the following targets:

Target	Description
*EnvelopeFrom	Compares the address of the Envelope Sender (also known as MAIL FROM) in the SMTP conversation
*FromAddress	Compares the addresses parsed out of the From header. Since multiple addresses are permitted in the From: header, only one has to match.
*Sender	Compares the address specified in the Sender header.
*Any	Matches messages that were created during an authenticated SMTP session regardless of identity.
*None	Matches messages that were not created during an authenticated SMTP session. This is useful when authentication is optional (preferred).

The filter performs matches loosely. It is not case-sensitive. If the optional *sieve-char* parameter is supplied, the last portion of an address that follows the specified character will be ignored for the purposes of comparison. For example, if the `+` character is included as a parameter, the filter ignores the portion of the address `joe+folder@example.com` that follows the `+` character. If the address was

joe+smith+folder@example.com, only the +folder portion is ignored. If the SMTP authenticated user ID string is a simple username and not a fully-qualified e-mail address, only the username portion of the target will be examined to determine a match. The domain must be verified in a separate rule.

Also, you can use the $SMTPAuthID variable to insert the STMP authenticated user ID into headers.

The following table shows examples of comparisons between the SMTP authenticated ID and email addresses and whether they would match using the smtp-auth-id-matches filter rule:

SMTP Auth ID	Sieve Char	Comparison Address	Matches?
someuser		otheruser@example.com	No
someuser		someuser@example.com	Yes
someuser		someuser@another.com	Yes
SomeUser		someuser@example.com	Yes
someuser		someuser+folder@example.com	No
someuser	+	someuser+folder@example.com	Yes
someuser@example.com		someuser@forged.com	No
someuser@example.com		someuser@example.com	Yes
SomeUser@example.com		someuser@example.com	Yes

The following filter checks all messages created during an authenticated SMTP session to verify that the addresses in the From header and the Envelope Sender match the SMTP authenticated user ID. If the addresses and the ID match, the filter verifies the domain. If they do not match, the appliance quarantines the message.

```
Msg_Authentication:

if (smtp-auth-id-matches("*Any"))

{

    # Always include the original authentication credentials in a

    # special header.

    insert-header("X-Auth-ID","$SMTPAuthID");



    if (smtp-auth-id-matches("*FromAddress", "+") and

        smtp-auth-id-matches("*EnvelopeFrom", "+"))

    {

        # Username matches.  Verify the domain

        if header('from') != "(?i)@(?:example\\.com|alternate\\.com)" or

           mail-from != "(?i)@(?:example\\.com|alternate\\.com)"

        {
```

```
            # User has specified a domain which cannot be authenticated
            quarantine("forged");
        }
    } else {
        # User claims to be an completely different user
        quarantine("forged");
    }
}
```

Signed Rule

The `signed` rule checks messages for a signature. The rule returns a boolean value to indicate if the message is signed or not. This rule evaluates whether the signature is encoded according to ASN.1 DER encoding rules and that it conforms to the CMS SignedData Type structure (RFC 3852, Section 5.1.). It does not aim to validate whether the signature matches the content, nor does it check the validity of the certificate.

The following example shows a `signed` rule used to insert headers into a signed message:

```
signedcheck: if signed { insert-header("X-Signed", "True"); }
```

The following example shows a `signed` rule used to drop attachments from unsigned messages from a certain sender group:

```
Signed: if ((sendergroup == "NOTTRUSTED") AND NOT signed) {
 html-convert();
    if (attachment_size > 0)
    {
        drop_attachments("");
    }
}
```

Signed Certificate Rule

The `signed-certificate` rule selects those S/MIME messages where the X.509 certificate issuer or message signer matches the given regular expression. This rule only supports X.509 certificates.

The rule's syntax is `signed-certificate (<field> [<operator> <regular expression>])`, where:

- `<field>` is either the quoted string `"issuer"` or `"signer"`,
- `<operator>` is either `==` or `!=`,

- and `<regular expression>` is the value for matching the “issuer” or “signer.”

If the message is signed using multiple signatures, the rule returns true if any of the issuers or signers match the regular expression. The short form of this rule, `signed-certificate("issuer")` and `signed-certificate("signer")`, returns true if the S/MIME message contains an issuer or signer.

Signer

For message signers, the rule extracts the sequence of `rfc822Name` names from the X.509 certificate’s `subjectAltName` extension. If there is no `subjectAltName` field in the signing certificate, or this field does not have any `rfc822Name` names, the `signed-certificate("signer")` rule evaluates to false. In the rare cases of multiple `rfc822Name` names, the rule tries to match all of the names to the regular expression and evaluates as true on the first match.

Issuer

The issuer is a non-empty distinguished name in the X.509 certificate. AsyncOS extracts the issuer from the certificate and converts it to an LDAP-UTF8 Unicode string. For example:

- C=US,S=CA,O=IronPort
- C=US,CN=Bob Smith

Since X.509 certificates require the issuer field, `signed-certificate("issuer")` evaluates whether the S/MIME message contains an X.509 certificate.

Escaping in Regular Expressions

LDAP-UTF8 defines a mechanism for escaping that you can use in your regular expressions. For a detailed discussion on escaping characters in LDAP-UTF8, consult Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names, accessible from `http://www.ietf.org/rfc/rfc4514.txt`.

The escaping rules for the `signed-certificate` rule’s regular expressions differ from the escaping rules defined in LDAP-UTF8 by limiting escaping to only the characters that require escaping. LDAP-UTF8 allows optional escaping for characters that can be represented without escaping. For example, the following two strings are considered correct for “Example, Inc.” using the LDAP-UTF8 escaping rules:

- `Example\, Inc.`
- `Example\,\ Inc\.`

However, the `signed-certificate` rule only matches `Example\, Inc.` The regular expression does not allow escaping the space and period for matching because these characters do not require escaping, even though it is permitted in LDAP-UTF8. When creating a regular expression for the `signed-certificate` rule, do not escape a character if it can be represented without escaping.

$CertificateSigners Action Variable

The action variable `$CertificateSigners` is a comma separated list of signers obtained from the `subjectAltName` element of the signing certificate. Multiple email addresses of a single signer will be included in the list with duplicates removed.

For example, Alice signs a message with her two certificates. Bob signs the message with his single certificate. All certificates are issued by a single corporate authority. After the message passes the S/MIME scan, the extracted data contain three items:

```
[
  {
    'issuer': 'CN=Auth,O=Example\, Inc.',
    'signer': ['alice@example.com', 'al@private.example.com']
  },
  {
    'issuer': 'CN=Auth,O=Example\, Inc.',
    'signer': ['alice@example.com', 'al@private.example.com']
  },
  {
    'issuer': 'CN=Auth,O=Example\, Inc.',
    'signer': ['bob@example.com', 'bob@private.example.com']
  }
]
```

The `$CertificateSigners` variable expands to:

```
"alice@example.com, al@private.example.com, bob@example.com, bob@private.example.com"
```

Examples

The following example inserts a new header if the certificate issuer is from the US:

```
Issuer: if signed-certificate("issuer") == "(?i)C=US" {
    insert-header("X-Test", "US issuer");
}
```

The following example notifies an administrator if the signer is not from example.com:

```
NotOurSigners: if signed-certificate("signer") AND
      signed-certificate("signer") != "example\\.com$" {
```

```
    notify("admin@example.com");

}
```

The following example adds a header if the message has an X.509 certificate:

```
AnyX509: if signed-certificate ("issuer") {

    insert-header("X-Test", "X.509 present");

}
```

The following example adds a header if the message's certificate does not have a signer:

```
NoSigner: if not signed-certificate ("signer") {

    insert-header("X-Test", "Old X.509?");

}
```

Message Filter Actions

The purpose of message filters is to perform actions on selected messages.

The two types of actions are:

- *Final* actions — such as `deliver`, `drop`, and `bounce` — end the processing of a message, and permit no further processing through subsequent filters.
- *Non-final* actions perform an action which permits the message to be processed further.

Non-final message filter actions are cumulative. If a message matches multiple filters where each filter specifies a different action, then all actions are accumulated and enforced. However, if a message matches multiple filters specifying the same action, the prior actions are overridden and the final filter action is enforced.

Filter Actions Summary Table

Message filters can apply the following actions shown in Table 6-5 to an email message.

Table 6-5 Message Filter Actions

Action	Syntax	Description
Alter source host	`alt-src-host`	Change the source hostname and IP interface (Virtual Gateway address) to send the message. See Alter Source Host (Virtual Gateway address) Action, page 6-58.
Alter recipient	`alt-rcpt-to`	Change a recipient of the message. See Alter Recipient Action, page 6-57.
Alter mailhost	`alt-mailhost`	Change the destination mail host for the message. See Alter Delivery Host Action, page 6-57.

Table 6-5 Message Filter Actions

Action	Syntax	Description
Notify	`notify`	Report this message to another recipient. See Notify and Notify-Copy Actions, page 6-52.
Notify Copy	`notify-copy`	Perform just like the `notify` action, but also sends a copy as with the `bcc-scan` action. See Notify and Notify-Copy Actions, page 6-52.
Blind carbon copy	`bcc`	Copy this message (message replication) anonymously to another recipient. See Blind Carbon Copy Actions, page 6-54.
Blind carbon copy with scan	`bcc-scan`	Copy this message anonymously to another recipient, and process that message through the work queue as if it were a new message. See Blind Carbon Copy Actions, page 6-54.
Archive	`archive`	Archive this message into an mbox-format file. See Archive Action, page 6-59.
Quarantine	`quarantine (`*`quarantine_name`*`)`	Flag this message to be sent to the quarantine named *quarantine_name*. See Quarantine and Duplicate Actions, page 6-56.
Duplicate (Quarantine)	`duplicate-quarantine(`*`quarantine_name`*`)`	Send a copy of the message to the specified quarantine. See Quarantine and Duplicate Actions, page 6-56.
Remove headers	`strip-header`	Remove specified headers from the message before delivering. See Strip Header Action, page 6-60.
Insert headers	`insert-header`	Insert a header and value pair into the message before delivering. See Insert Header Action, page 6-60.
Edit header text	`edit-header-text`	Replace specified header text with a text string you specify in the filter condition. See Edit Header Text Action, page 6-61.
Edit body text	`edit-body-text()`	Strip a regular expression from a message body and replaces it with text that you specify. You might want to use this filter if you want to remove and replace specific content, such as a URL within a message body. See Edit Body Text Action, page 6-61.
Convert HTML	`html-convert()`	Strip HTML tags from message bodies and leaves the plain text content of the message. You might want to use this filter if you want to convert all HTML text in a message to plain text. HTML Convert Action, page 6-62.
Assign bounce profile	`bounce-profile`	Assign a specific bounce profile to the message. See Bounce Profile Action, page 6-63.
Bypass Anti-Spam System	`skip-spamcheck`	Ensure that the anti-spam systems in the Cisco IronPort system are *not* applied to this message. See Bypass Anti-Spam System Action, page 6-63.
Bypass Anti-Virus System	`skip-viruscheck`	Ensure that the anti-virus systems in the Cisco IronPort system are *not* applied to this message. See Bypass Anti-Virus System Action, page 6-64.

Table 6-5 Message Filter Actions

Action	Syntax	Description
Skip Outbreak Filter Scanning	`skip-vofcheck`	Ensure that this message is not processed by the Outbreak Filters scanning. See Bypass Anti-Virus System Action, page 6-64.
Drop Attachments by Name	`drop-attachments-by-name`	Drop all attachments on messages that have a filename that match the given regular expression. Archive file attachments (zip, tar) will be dropped if they contain a file that matches. See Examples of Attachment Scanning Message Filters, page 6-72.
Drop Attachments by Type	`drop-attachments-by-type`	Drop all attachments on messages that have a MIME type, determined by either the given MIME type or the file extension. Archive file attachments (zip, tar) will be dropped if they contain a file that matches. See Examples of Attachment Scanning Message Filters, page 6-72.
Drop Attachments by File Type	`drop-attachments-by-filetype`	Drop all attachments on messages that match the given "fingerprint" of the file. Archive file attachments (zip, tar) will be dropped if they contain a file that matches. For more information, see Attachment Scanning, page 6-65.
Drop Attachments by MIME Type	`drop-attachments-by-mimetype`	Drop all attachments on messages that have a given MIME type. This action does not attempt to ascertain the MIME type by file extension and so it also does not examine the contents of archives. See Examples of Attachment Scanning Message Filters, page 6-72.
Drop Attachments by Size	`drop-attachments-by-size`	Drop all attachments on the message that, in raw encoded form, are equal to or greater than the size (in bytes) given. Note that for archive or compressed files, this action does not examine the uncompressed size, but rather the size of the actual attachment prior to any decoding. See Examples of Attachment Scanning Message Filters, page 6-72.
Drop Attachments by Content	`drop-attachments-where-contains`	Drop all attachments on message that contain the regular expression. Does the pattern occur the minimum number of times you specified for the threshold value? Archive files (zip, tar) will be dropped if any of the files they contain match the regular expression pattern. See Examples of Attachment Scanning Message Filters, page 6-72. The optional comment serves as the means to modify the text used to replace the attachment that was dropped. Attachment footers simply append to the message.
Drop Attachments by Dictionary Matches	`drop-attachments-where-dictionary-match`	Strip attachments based on matches to dictionary terms. If the terms in the MIME parts considered to be an attachment match a dictionary term (and the user-defined threshold is met), the attachment is stripped from the email. See Examples of Attachment Scanning Message Filters, page 6-72.

Table 6-5 Message Filter Actions

Action	Syntax	Description
Add Footer	`add-footer(`*`footer-name`*`)`	Add a footer to the message. See "Message Disclaimer Stamping" in the "Text Resources" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide* for more information.
Encrypt on Delivery	`encrypt-deferred`	Encrypt message on delivery, which means that the message continues to the next stage of processing, and when all processing is complete, the message is encrypted and delivered.
Add Message Tag	`tag-message(tag-name)`	Add a custom term into the message to use with RSA Email DLP policy filtering. You can configure a RSA Email DLP policy to limit scanning to messages with the message tag. The message tag is not visible to recipients. See Add Message Tag Action, page 6-64 and the "Data Loss Prevention" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide*.
Add Log Entry	`log-entry`	Adds customized text into the IronPort Text Mail logs at the `INFO` level. The text can include action variables. The log entry appears in message tracking. See Add Log Entry Action, page 6-65.
***Skip Remaining Message Filters**	`skip-filters`	Ensure that this message is not processed by any other message filters and continues through the email pipeline. See Skip Remaining Message Filters Action, page 6-50.
***Drop message**	`drop`	Drop and discard the message. See Drop Action, page 6-51.
***Bounce message**	`bounce`	Send the message back to the sender. See Bounce Action, page 6-51.
***Encrypt and Deliver Now**	`encrypt`	Use Cisco IronPort Email Encryption to encrypt outgoing messages. See Encrypt Action, page 6-51.

**** Final Actions***

Attachment Groups

You can specify a particular file type ("exe" files for example) or common groups of attachments in the `attachment-filetype` and `drop-attachments-by-filetype rules`. AsyncOS divides the attachments into the groups listed in Table 6-6.

If you create a message filter that uses the `!=` operator to match a message that does not contain an attachment with a specific file type, the filter will not perform any action on the message if there is at least one attachment with the file type you want to filter out. For example, the following filter drops any message with an attachment that is not an *`.exe`* file type:

```
exe_check: if (attachment-filetype != "exe") {

                drop();

           }
```

If a message has multiple attachments, the Email Security appliance does not drop the message if at least one of the attachments is an `.exe` file, even if the other attachments not `.exe` files.

Table 6-6 Attachment Groups

Attachment Group Name	Scanned File Types
Document	• doc, docx • mdb • mpp • ole • pdf • ppt, pptx • rtf • wps • x-wmf • xls, xlsx
Executable	• exe • java • msi • pif **Note** Filtering the Executable group will also scan `.dll` and `.scr` files, but you cannot filter these file types individually.
Compressed	• ace (ACE Archiver compressed file) • arc (SQUASH Compressed archive) • arj (Robert Jung ARJ compressed archive) • binhex • bz (Bzip compressed file) • bz2 (Bzip compressed file) • cab (Microsoft cabinet file) • gzip* (Compressed file - UNIX gzip) • lha (Compressed Archive [LHA/LHARC/LHZ]) • sit (Compressed archive - Macintosh file [Stuffit]) • tar* (Compressed archive) • unix (UNIX compress file) • zip* (Compressed archive - Windows) • zoo (ZOO Compressed Archive File) * These file types can be “body-scanned”

Table 6-6 Attachment Groups (Continued)

Attachment Group Name	Scanned File Types
Text	• txt • html • xml
Image	• bmp • cur • gif • ico • jpeg • pcx • png • psd • psp • tga • tiff
Media	• aac • aiff • asf • avi • flash • midi • mov • mp3 • mpeg • ogg • ram • snd • wav • wma • wmv

Action Variables

The `bcc()`, `bcc-scan()`, `notify()`, `notify-copy()`, `add-footer()`, and `insert-headers()` actions have parameters that may use certain variables that will be automatically replaced with information from the original message when the action is executed. These special variables are called *action variables.* Your Cisco IronPort appliance supports the following set of action variables:

Table 6-7 Message Filter Action Variables

Variable	Syntax	Description
All Headers	`$AllHeaders`	Returns the message headers.
Body Size	`$BodySize`	Returns the size, in bytes, of the message.
Certificate Signers	`$CertificateSigners`	Returns the signers from the `subjectAltName` element of a signing certificate. See $CertificateSigners Action Variable, page 6-40 for more information.
Date	`$Date`	Returns the current date, using the format MM/DD/YYYY.
Dropped File Name	`$dropped_filename`	Returns only the most recently dropped filename.
Dropped File Names	`$dropped_filenames`	Displays list of dropped files (similar to `$filenames`.
Dropped File Types	`$dropped_filetypes`	Displays list of dropped file types (similar to `$filetypes)`.
Envelope Sender	`$EnvelopeFrom`	Returns the Envelope Sender (Envelope From, <MAIL FROM>) of the message.
Envelope Recipients	`$EnvelopeRecipients`	Returns all Envelope Recipients (Envelope To, <RCPT TO>) of the message.
File Names	`$filenames`	Returns a comma-separated list of the message's attachments' filenames.
File Sizes	`$filesizes`	Returns a comma-separated list of the message's attachments file sizes.
File Types	`$filetypes`	Returns a comma-separated list of the message's attachments' file types.
Filter Name	`$FilterName`	Returns the name of the filter being processed.
GMTimeStamp	`$GMTimeStamp`	Returns the current time and date, as would be found in the Received: line of an email message, using GMT.
HAT Group Name	`$Group`	Returns the name of the sender group the sender matched on when injecting the message. If the sender group had no name, the string "`>Unknown<`" is inserted.
Matched Content	`$MatchedContent`	Returns the content that triggered a scanning filter rule (including filter rules such as `body-contains` and content dictionaries).

Table 6-7 Message Filter Action Variables (Continued)

Variable	Syntax	Description
Mail Flow Policy	`$Policy`	Returns the name of the HAT policy applied to the sender when injecting the message. If no predefined policy name was used, the string "`>Unknown<`" is inserted.
Header	`$Header['string']`	Returns the value of the quoted header, if the original message contains a matching header. Note that double quotes may also be used.
Hostname	`$Hostname`	Returns the hostname of the Cisco IronPort appliance.
Internal Message ID	`$MID`	Returns the Message ID, or "MID" used internally to identify the message. Not to be confused with the RFC822 "Message-Id" value (use $Header to retrieve that).
Receiving Listener	`$RecvListener`	Replaced by the nickname of the listener that received the message.
Receiving Interface	`$RecvInt`	Returns the nickname of the interface that received the message.
Remote IP Address	`$RemoteIP`	Returns the IP address of the system that sent the message to the Cisco IronPort appliance.
Remote Host Address	`$remotehost`	Returns the hostname of the system that sent the message to the Cisco IronPort appliance.
SenderBase Reputation Score	`$Reputation`	Returns the SenderBase Reputation score of the sender. If there is no reputation score, it is replaced with "`None`".
Subject	`$Subject`	Returns the subject of the message.
Time	`$Time`	Returns the current time, in the local time zone.
Timestamp	`$Timestamp`	Returns the current time and date, as would be found in the Received: line of an email message, in the local time zone.

Non-ASCII Character Sets and Message Filter Action Variables

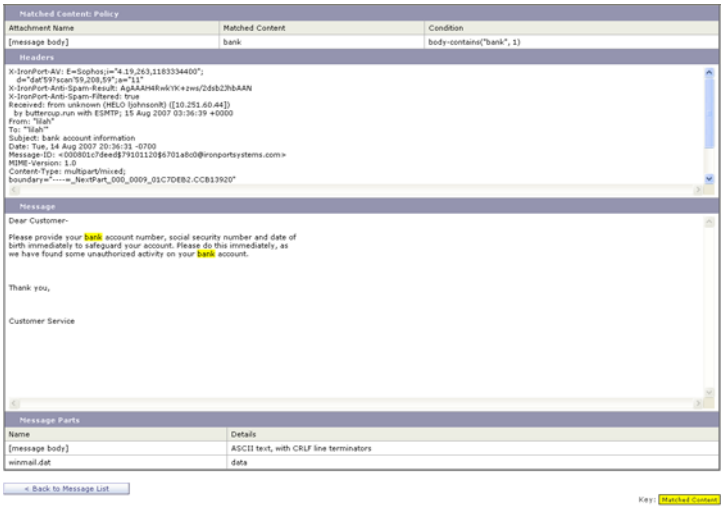
The system supports the expansion of action variables that contain ISO-2022 style character codings (the style of encoding used in header values) and also supports international text in the notification. These will be merged together to generate a notification that will then be sent as a UTF-8, quoted printable message.

Matched Content Visibility

When you configure a quarantine action for messages that match Attachment Content conditions, Message Body or Attachment conditions, Message body conditions, or the Attachment content conditions, you can view the matched content in the quarantined message. When you display the message body, the matched content is highlighted in yellow. You can also use the $MatchedContent action variable to include the matched content in the message subject.

When you view messages in the local quarantine that have triggered message or content filter rules, the GUI may display content that did not actually trigger the filter action (along with content that triggered the filter action). The GUI display should be used as a guideline for locating content matches, but does not necessarily reflect an exact list of content matches. This occurs because the GUI uses less strict content matching logic than is used in the filters. This issue applies only to the highlighting in the message body. The table that lists the matched strings in each part of the message along with the associated filter rule is correct.

Figure 6-2 Matched Content Viewed in the Policy Quarantine



Examples of Message Filter Actions

Skip Remaining Message Filters Action

The `skip-filters` action ensures that the message skips any further processing from message filters and continues through the email pipeline. The message that incurs the `skip-filters` action will be subject to anti-spam scanning and anti-virus scanning, if it is available on the appliance. The `skip-filters` action is the default final action for message filters.

The following filter notifies `customercare@example.com` and then immediately delivers any message addressed to `boss@admin`.

```
bossFilter:

   if(rcpt-to == 'boss@admin$')

   {
```

```
        notify('customercare@example.com');

 skip-filters();

   }
```

Drop Action

The `drop` action discards a message without any delivery. The message is not returned to the sender, not sent to the intended recipient, nor processed further in any way.

the following filter first notifies `george@whitehouse.gov` and then discards any message where the subject begins with `SPAM`.

```
spamFilter:

   if(subject == '^SPAM.*')

   {

        notify('george@whitehouse.gov');

        drop();

   }
```

Bounce Action

The `bounce` action sends the message back to the sender (Envelope Sender) without further processing.

the following filter returns (bounces) any message from an email address that ends in `@yahoo\\.com`.

```
yahooFilter:

   if(mail-from == '@yahoo\\.com$')

   {

        bounce();

   }
```

Encrypt Action

The encrypt action uses the configured encryption profile to deliver encrypted messages to email recipients.

The following filter encrypts messages if they contain the term [encrypt] in the subject:

```
Encrypt_Filter:

  if ( subject == '\\[encrypt\\]' )
```

```
    {
        encrypt('My_Encryption_Profile');
    }
```

Note You must have a Cisco IronPort Encryption Appliance in your network or a hosted key service configured to use this filter action. You must also have configured an encryption profile to use this filter action.

Notify and Notify-Copy Actions

The `notify` and `notify-copy` actions send an email summary of the message to the specified email address. The `notify-copy` action also sends a copy of the original message, similar to the `bcc-scan` action. The notification summary contains:

- The contents of the Envelope Sender and Envelope Recipient (`MAIL FROM` and `RCPT TO`) directives from the mail transfer protocol conversation for the message.
- The message headers of the message.
- The name of the message filter that matched the message.

You can specify the recipient, subject line, from address, and notification template. the following filter selects messages with sizes larger than 4 megabytes, sends a notification email of each matching message to `admin@example.com`, and finally discards the message:

```
bigFilter:
    if(body-size >= 4M)
    {
        notify('admin@example.com');
        drop();
    }
```

Or

```
bigFilterCopy:
    if(body-size >= 4M)
    {
        notify-copy('admin@example.com');
        drop();
    }
```

The Envelope Recipient parameter may be any valid email address (for example, `admin@example.com` in the example above), or alternatively, may be the action variable `$EnvelopeRecipients` (see Action Variables, page 6-48), which specifies all Envelope Recipients of the message:

```
bigFilter:

   if(body-size >= 4M)

   {

        notify('$EnvelopeRecipients');

        drop();

   }
```

The `notify` action also supports up to three additional, optional arguments that allow you to specify the subject header, the Envelope Sender, and a pre-defined text resource to use for the notification message. These parameters must appear in order, so a subject must be provided if the Envelope Sender is to be set or a notification template specified.

The subject parameter may contain action variables (see Action Variables, page 6-48) that will be replaced with data from the original message. By default, the subject is set to `Message Notification`.

The Envelope Sender parameter may be any valid email address, or alternatively, may be the action variable `$EnvelopeFrom`, which will set the return path of the message to the same as the original message

The notification template parameter is the name of an existing notification template. For more information, see Notifications, page 6-72.

This example extends the previous one, but changes the subject to look like `[bigFilter] Message too large`, sets the return path to be the original sender, and uses the “message.too.large” template:

```
bigFilter:

   if (body-size >= 4M)

   {

        notify('admin@example.com', '[$FilterName] Message too large',

               '$EnvelopeFrom', 'message.too.large');

        drop();

   }
```

You can also use the $MatchedContent action variable to notify senders or administrators that a content filter was triggered. The $MatchedContent action variable displays the content that triggered the filter. For example, the following filter sends a notification to an administrator if the email contains ABA account information.

```
ABA_filter:

if (body-contains ('*aba')){

notify('admin@example.com','[$MatchedContent]Account Information Displayed');

}
```

Notification Template

You can use the Text Resources page or the `textconfig` CLI command to configure custom notification templates as text resources for use with the `notify()` and `notify-copy()` actions. If you do not create a custom notification template, a default template is used. The default template includes message headers, but the custom notification template does not include message headers by default. To include message headers in the custom notification, include the `$AllHeaders` action variable.

For more information, see the "Text Resources" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

in the following example, when a large message triggers the filter shown below, an email is sent to the intended recipients explaining that the message was too large:

```
bigFilter:

   if (body-size >= 4M)

   {

        notify('$EnvelopeRecipients', '[$FilterName] Message too large',

               '$EnvelopeFrom', 'message.too.large');

        drop();

   }
```

Blind Carbon Copy Actions

The `bcc` action sends an anonymous copy of the message to a specified recipient. This is sometimes referred to as message replication. Because no mention of the copy is made in the original message and the anonymous copy will never successfully bounce back to the recipient, the original sender and recipients of the message will not necessarily know that the copy was sent.

the following filter sends a blind carbon copy to `mom@home.org` for each message addressed to `sue` from `johnny`:

```
momFilter:

   if ((mail-from == '^johnny$') and (rcpt-to == '^sue$'))
```

```
   {
        bcc('mom@home.org');
   }
```

The `bcc` action also supports up to three additional, optional arguments that allow you to specify the subject header and Envelope Sender to use on the copied message, as well as an alt-mailhost. These parameters must appear in order, so a subject must be provided if the Envelope Sender is to be set.

The subject parameter may contain action variables (see Action Variables, page 6-48) that will be replaced with data from the original message. By default, this is set to the subject of the original message (the equivalent of `$Subject`).

The Envelope Sender parameter may be any valid email address, or alternatively, may be the action variable `$EnvelopeFrom`, which will set the return path of the message to the same as the original message.

This example expands the previous one by setting the subject to be `[Bcc] <original subject>`, and the return path set to `badbounce@home.org`:

```
momFilter:
   if ((mail-from == '^johnny$') and (rcpt-to == '^sue$'))
   {
        bcc('mom@home.org', '[Bcc] $Subject', 'badbounce@home.org');
   }
```

The alt-mailhost is the fourth parameter:

```
momFilterAltM:
   if ((mail-from == '^johnny$') and (rcpt-to == '^sue$'))
   {
      bcc('mom@home.org', '[Bcc] $Subject', '$EnvelopeFrom',
'momaltmailserver.example.com');
   }
```

Warning **The `Bcc()`, `notify()`, and `bounce()` filter actions can allow viruses through your network. The blind carbon copy filter action creates a new message which is a full copy of the original message. The notify filter action creates a new message that contains the headers of the original message. While it is rare, headers can contain viruses. The bounce filter action creates a new message which contains the first 10k of the original message. In all three cases, the new message will not be processed by anti-virus or anti-spam scanning.**

To send to multiple hosts, you can call the `bcc()` action multiple times:

```
multiplealthosts:

   if (recv-listener == "IncomingMail")

   {

    insert-header('X-ORIGINAL-IP', '$remote_ip');

    bcc ('$EnvelopeRecipients', '$Subject', '$EnvelopeFrom', '10.2.3.4');

    bcc ('$EnvelopeRecipients', '$Subject', '$EnvelopeFrom', '10.2.3.5');

    bcc ('$EnvelopeRecipients', '$Subject', '$EnvelopeFrom', '10.2.3.6');

   }
```

The bcc-scan() Action

The `bcc-scan` action functions similarly to the bcc action, except that the message that is sent is treated as a brand new message and is therefore sent through the entire email pipeline.

```
momFilter:

   if ((mail-from == '^johnny$') and (rcpt-to == '^sue$'))

   {

        bcc-scan('mom@home.org');

   }
```

Quarantine and Duplicate Actions

The `quarantine('quarantine_name')` action flags a message for inclusion into a queue called a quarantine. For more information about quarantines, see "Quarantines" in the *Cisco IronPort AsyncOS for Email Daily Management Guide*. The `duplicate-quarantine('quarantine_name')` action immediately places a copy of the message into the specified quarantine and the original message continues through the email pipeline. The quarantine name is case sensitive.

When flagged for quarantine, the message continues through the rest of the email pipeline. When the message reaches the end of the pipeline, if the message has been flagged for one or more quarantines then it enters those queues. Otherwise, it is delivered. Note that if the message does not reach the end of the pipeline, it is not placed in a quarantine.

Accordingly, if a message filter contains a `quarantine()` action followed by a `bounce()` or `drop()` action, the message will not enter the quarantine, since the final action prevents the message from reaching the end of the pipeline. The same is true if a message filter includes a quarantine action, but the message is later dropped by anti-spam or anti-virus scanning, or a content filter. The `skip_filters()` final action causes the message to skip any remaining message filters, but content filters may still apply. For example, if a message filter flags a message for quarantine and also includes the `skip_filters()` final action, the message skips all remaining message filters and will be quarantined, unless another action in the email pipeline causes the message to be dropped.

In the following example, the message is sent to the Policy quarantine if the message contains any words within the dictionary named “secret_word.”

```
quarantine_codenames:

   if (dictionary-match ('secret_words'))

       {

       quarantine('Policy');

       }
```

In the following example, suppose a company has an official policy to drop all .mp3 file attachments. If an inbound message has a .mp3 attachment, the attachment is stripped and the remaining message (original body and remaining attachments) is sent to the original recipient. Another copy of the original message with all attachments will be quarantined (sent to the Policy quarantine). If it is necessary to receive the blocked attachment(s), the original recipient would then request that the message be released from the quarantine.

```
strip_all_mp3s:

  if (attachment-filename == '(?i)\\.mp3$') {

        duplicate-quarantine('Policy');

        drop-attachments-by-name('(?i)\\.mp3$');

   }
```

Alter Recipient Action

The `alt-rcpt-to` action changes all recipients of the message to the specified recipient upon delivery.

The following filter sends all messages with an Envelope Recipient address that contain `.freelist.com` and changes all recipients for the message to `system-lists@myhost.com`:

```
freelistFilter:

   if(rcpt-to == '\\.freelist\\.com$')

   {

        alt-rcpt-to('system-lists@myhost.com');

   }
```

Alter Delivery Host Action

The `alt-mailhost` action changes the IP address for all recipients of the selected message to the numeric IP address or hostname given.

Note The `alt-mailhost` action prevents a message classified as spam by an anti-spam scanning engine from being quarantined. The `alt-mailhost` action overrides the quarantine action and sends it to the specified mail host.

The following filter redirects recipient addresses to the host `example.com` for all messages.

```
localRedirectFilter:

   if(true)

   {

        alt-mailhost('example.com');

   }
```

Thus, a message directed to `joe@anywhere.com` is delivered to the mailhost at `example.com` with the Envelope To address `joe@anywhere.com`. Note that any additional routing information specified by the `smtproutes` command still affects the routing of the message. (See Routing Email for Local Domains, page 2-1.)

Note The `alt-mailhost` action does not support specifying a port number. To do this, add an SMTP route instead.

The following filter redirects all messages to `192.168.12.5`:

```
local2Filter:

   if(true)

   {

        alt-mailhost('192.168.12.5');

   }
```

Alter Source Host (Virtual Gateway address) Action

The `alt-src-host` action changes the source host for the message to the source specified. The source host consists of the IP interface or group of IP interfaces that the messages should be delivered from. If a group of IP interfaces is selected, the system round-robins through all of the IP interfaces within the group as the source interface when delivering email. In essence, this allows multiple Virtual Gateway addresses to be created on a single Cisco IronPort Email Security appliance. For more information, see Using Virtual Gateway™ Technology, page 2-59.

The IP interface may only be changed to an IP interface or interface group currently configured in the system. the following filter creates a Virtual Gateway using the outbound (delivery) IP interface `outbound2` for all messages received from a remote host with the IP address `1.2.3.4`.

```
externalFilter:

   if(remote-ip == '1.2.3.4')

   {

        alt-src-host('outbound2');

   }
```

the following filter uses the IP interface group `Group1` for all messages received from a remote host with the IP address `1.2.3.4`.

```
groupFilter:

   if(remote-ip == '1.2.3.4')

   {

        alt-src-host('Group1');

   }
```

Archive Action

The `archive` action saves a copy of the original message, including all message headers and recipients into an mbox-format file on the appliance. The action takes a parameter that is the name of the log file in which to save the message. The system automatically creates a log subscription with the specified filename when you create the filter, or you can also specify an existing filter log file. After the filter and the filter log file are created, the filter log options may then be edited with the `filters -> logconfig` subcommand.

Note The `logconfig` command is a subcommand of `filters`. See Using the CLI to Manage Message Filters, page 6-75 for a full description of how to use this subcommand.

The mbox format is a standard UNIX mailbox format, and there are many utilities available to make viewing the messages easier. Most UNIX systems allow you to type "`mail -f` *mbox.filename*" to view the files. The mbox format is in plain text, so you can use a simple text editor to view the contents of the messages.

In the following example, a copy of the message is saved to a log named `joesmith` if the Envelope Sender matches `joesmith@yourdomain.com`:

```
logJoeSmithFilter:

   if(mail-from == '^joesmith@yourdomain\\.com$')

   {
```

```
        archive('joesmith');

    }
```

Strip Header Action

The `strip-header` action examines the message for a particular header and removes those lines from the message before delivering it. When there are multiple headers, all instances of the header are removed (for example, the “Received:” header.)

In the following example, all messages have the header `X-DeleteMe` removed before transmission:

```
stripXDeleteMeFilter:

    if (true)

    {

        strip-header('X-DeleteMe');

    }
```

When working with headers, remember that the current value of the header includes changes made during processing (such as with filter actions that add, remove, or modify message headings). See Message Header Rules and Evaluation, page 6-4 for more information.

Insert Header Action

The `insert-header` action inserts a new header into a message. AsyncOS does not verify the compliance to standards of the header you insert; you are responsible for ensuring that the resulting message complies with Internet standards for email.

The following example inserts a header named `X-Company` with the value set to `My Company Name` if the header is not already found in the message:

```
addXCompanyFilter:

    if (not header('X-Company'))

    {

        insert-header('X-Company', 'My Company Name');

    }
```

The insert-header() action allows the use of non-ASCII characters in the text of the header, while restricting the header name to be ASCII (to comply with standards). The transport encoding will be quoted-printable to maximize the readability.

Note The `strip-headers` and `insert-header` actions can be used in combination to rewrite any message headers in the original message. In some case, it is valid to have multiple instances of the same header (for example, `Received:`) where in other cases, multiple instances of the same header could confuse a MUA (for example, multiple Subject: headers.)

When working with headers, remember that the current value of the header includes changes made during processing (such as with filter actions that add, remove, or modify message headings). See Message Header Rules and Evaluation, page 6-4 for more information.

Edit Header Text Action

The `edit-header-text` action allows you to rewrite specified header text using the regular expression substitution function. The filter matches the regular expression within the header and replaces it with a regular expression you specify.

For example, an email contains the following subject header:

```
Subject: SCAN Marketing Messages
```

The following filter removes the “SCAN” text, and leaves the text, “Marketing Messages”, in the header:

```
Remove_SCAN: if true

   {

      edit-header-text (‘Subject’, ‘^SCAN\\s*’,’’);

}
```

After the filter processes the message, it returns the following header:

```
Subject: Marketing Messages
```

Edit Body Text Action

The `edit-body-text()` message filter is similar to the `Edit-Header-Text()` filter, but it operates across the body of the message instead of one of the headers.

The `edit-body-text()` message filter uses the following syntax where the first parameter is the regular expression to search for and the second parameter is the replacement text:

```
Example: if true {

edit-body-text("parameter 1",

"parameter 2");

       }
```

The `edit-body-text()` message filter only works on the message body parts. For more information about whether a given MIME part is considered a message “body” or a message “attachment”, see Message Bodies vs. Message Attachments, page 6-5.

The following example shows a URL removed from a message and replaced with the text, ‘URL REMOVED’:

```
URL_Replaced: if true {



 edit-body-text("(?i)(?:https?|ftp)://[^\\s\">]+", "URL REMOVED");

       }
```

The following example shows a social security number removed from the body of a message and replaced with the text, “XXX-XX-XXXX’:

```
ssn: if true {

  edit-body-text("(?!000)(?:[0-6]\\d{2}|7(?:[0-6]\\d|7[012]))([
-]?)(?!00)\\d\\d\\1(?!0000)\\d{4}",

"XXX-XX-XXXX");

}
```

Note You cannot use smart identifiers with the `edit-body-text()` filter at this time.

HTML Convert Action

While RFC 2822 defines a text format for email messages, there are extensions (such as MIME) to provide the transport of other content within an RFC 2822 message. AsyncOS can now use the `html-convert()` message filter to convert HTML to plain text using the following syntax:

```
Convert_HTML_Filter:

if (true)

{

html-convert();

}
```

The Cisco IronPort message filters make a determination on whether a given MIME part is considered a message “body” or a message “attachment”. The `html-convert()` filter only works on the message body parts. For more information about message bodies and attachments, see Message Bodies vs. Message Attachments, page 6-5.

Depending on the format, the `html-convert()` filter uses different methods to strip the HTML from within the documents.

If the message is plain text (text/plain), the message passes through the filter unchanged. If the message is a simple HTML message (text/html), all the HTML tags are stripped out of the message and the resulting body replaces the HTML message. The lines are not reformatted, and the HTML is not rendered in plain text. If the structure is MIME (with a multipart/alternative structure) and it contains both a text/plain part and text/html part with the same content, the filter removes the text/html part of the message and leaves the text/plain part of the message. For all other MIME types (such as multipart/mixed), all HTML body parts are stripped of their tags and reinserted into the message.

When encountered in a message filter, the `html-convert()` filter action only tags the message to be processed but does not immediately make a change to the message structure. The changes to the message only take effect after all processing is complete. This allows the other filter actions to process the original message body prior to modification.

Bounce Profile Action

The `bounce-profile` action assigns a previously-configured bounce profile to the message. (See Directing Bounced Email, page 2-35.) If the message is undeliverable, the bounce options configured via the bounce profile are used. Using this feature overrides the bounce profile assigned to the message from the listener's configuration (if one is assigned).

The following filter example assigns the bounce profile "fastbounce" to all email sent with the header `X-Bounce-Profile: fastbounce`:

```
fastbounce:

   if (header ('X-Bounce-Profile') == 'fastbounce') {

        bounce-profile ('fastbounce');

   }
```

Bypass Anti-Spam System Action

The `skip-spamcheck` action instructs the system to allow the message to bypass any content-based anti-spam filtering configured on the system. This action does nothing to the message if no content-based anti-spam filtering is configured, or if the message was never flagged to be scanned for spam in the first place.

The following example allows messages that have a high SenderBase Reputation Score to bypass the content-based anti-spam filtering feature:

```
whitelist_on_reputation:

   if (reputation > 7.5)

   {

        skip-spamcheck();

   }
```

Bypass Anti-Virus System Action

The `skip-viruscheck` action instructs the system to allow the message to bypass any virus protection system configured on the system. This action does nothing to the message if there is no anti-virus system configured, or if the message was never flagged to be scanned for viruses in the first place.

The following example specifies that messages received on the listener "private_listener" should bypass the anti-spam and the anti-virus systems.

```
internal_mail_is_safe:

   if (recv-listener == 'private_listener')

   {

        skip-spamcheck();

        skip-viruscheck();

   }
```

Bypass Outbreak Filter Scanning Action

The `skip-vofcheck` action instructs the system to allow the message to bypass the Outbreak Filters scanning. This action does nothing to the message if Outbreak Filters scanning is not enabled.

The following example specifies that messages received on the listener "private_listener" should bypass Outbreak Filter scanning.

```
internal_mail_is_safe:

   if (recv-listener == 'private_listener') Outbreak Filters

   {

        skip-vofcheck();

   }
```

Add Message Tag Action

The `tag-message` action inserts a custom term into an outgoing message to use with RSA Email DLP policy filtering. You can configure a RSA Email DLP policy to limit scanning to messages with the message tag. The message tag is not visible to recipients. The tag name can contain any combination of characters from the set `[a-zA-Z0-9_-.]`.

For information on configuring a DLP policy to filter messages, see the "Data Loss Prevention" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide*.

The following example inserts a message tag into a message with "[Encrypt]" in the subject. You can then create a DLP policy that will encrypt messages with this message tag before delivering them if Cisco IronPort Email Encryption is available:

```
Tag_Message:

   if (subject == '^\\[Encrypt\\]')

   {

        tag-message('Encrypt-And-Deliver');

   }
```

Add Log Entry Action

The `log-entry` action inserts customized text into the IronPort Text Mail logs at the `INFO` level. The text can include action variables. You can use this action to insert useful text for debugging purposes and information on why a message filter performed a certain action. The log entry also appears in message tracking.

The following example inserts a log entry explaining that message was bounced because it possibly contained confidential company information:

```
CompanyConfidential:

   if (body-contains('Company Confidential'))

   {

        log-entry('Message may have contained confidential information.');

        bounce();

   }
```

Attachment Scanning

AsyncOS can strip attachments from messages that are inconsistent with your corporate policies, while still retaining the ability to deliver the original message.

You can filter attachments based on their specific file type, fingerprint, or based on the content of the attachment. Using the fingerprint to determine the exact type of attachment prevents users from renaming a malicious attachment extension (for example, `.exe`) to a more commonly used extension (for example, `.doc`) in the hope that the renamed file would bypass attachment filters.

When you scan attachments for content, the Stellent attachment scanning engine extracts data from attachment files to search for the regular expression. It examines both data and metadata in the attachment file. If you scan an Excel or Word document, the attachment scanning engine can also detect the following types of embedded files: .exe, .dll, .bmp, .tiff, .pcx, .gif, .jpeg, .png, and Photoshop images.

Message Filters for Scanning Attachments

The message filter actions described in Table 6-8 are *non-final* actions. (Attachments are dropped and the message processing continues.)

The optional comment is text that is added to the message, much like a footer, and it can contain Message Filter Action Variables (see Examples of Attachment Scanning Message Filters, page 6-72).

Table 6-8 Message Filter Actions for Attachment Filtering

Action	Syntax	Description
Drop Attachments by Name	`drop-attachments-by-name (<regular expression>[, <optional comment>])`	Drops all attachments on messages that have a filename that matches the given regular expression. Archive file attachments (zip, tar) will be dropped if they contain a file that matches. See Examples of Attachment Scanning Message Filters, page 6-72.
Drop Attachments by Type	`drop-attachments-by-type (<MIME type>[, <optional comment>])`	Drops all attachments on messages that have a MIME type, determined by either the given MIME type or the file extension. Archive file attachments (zip, tar) will be dropped if they contain a file that matches.
Drop Attachments by File Type	`drop-attachments-by-filetype (<fingerprint name>[, <optional comment>])`	Drops all attachments on messages that match the given "fingerprint" of the file. Archive file attachments (zip, tar) will be dropped if they contain a file that matches. For more information, see Table 6-6Attachment Groups, page 6-46.
Drop Attachments by MIME Type	`drop-attachments-by-mimetype (<MIME type>[, <optional comment>])`	Drops all attachments on messages that have a given MIME type. This action does not attempt to ascertain the MIME type by file extension and so it also does not examine the contents of archives.
Drop Attachments by Size	`drop-attachments-by-size (<number>[, <optional comment>])`	Drops all attachments on the message that, in raw encoded form, are equal to or greater than the size (in bytes) given. Note that for archive or compressed files, this action does not examine the uncompressed size, but rather the size of the actual attachment itself.
Attachment Scanning	`drop-attachments-where-contains (<regular expression>[, <optional comment>])`	Drops all attachments on message that contain the regular expression. Archive files (zip, tar) will be dropped if any of the files they contain match the regular expression pattern.
Drop Attachments by Dictionary Matches	`drop-attachments-where-dictionary-match(<dictionary name>)`	This filter action strips attachments based on matches to dictionary terms. If the terms in the MIME parts considered to be an attachment match a dictionary term (and the user-defined threshold is met), the attachment is stripped from the email. See Examples of Attachment Scanning Message Filters, page 6-72.

Image Analysis

Some messages contain images that you may wish to scan for inappropriate content. You can use the image analysis engine to search for inappropriate content in email. Image analysis is not designed to supplement or replace your anti-virus and anti-spam scanning engines. Its purpose is to enforce acceptable use by identifying inappropriate content in email. Use the image analysis scanning engine to quarantine and analyze mail and to detect trends.

After you configure AsyncOS for image analysis, you can use image analysis filter rules to perform actions on suspect or inappropriate emails. Image scanning allows you to scan the following types of attached files: JPEG, BMP, PNG, TIFF, GIF, TGA, ICO, and PCX. The image analyzer uses algorithms that measure skin color, body size and curvature to determine the probability that the graphic contains inappropriate content. When you scan image attachments, Cisco IronPort fingerprinting determines the file type, and the image analyzer uses algorithms to analyze the image content. If the image is embedded in another file, the Stellent scanning engine extracts the file. The Stellent scanning engine can extract images from many file types, including Word, Excel, and PowerPoint documents. The image analysis verdict is computed on the message as a whole. If the message does not include any images, the message receives a score of “0” which maps to a “clean” verdict. Therefore, a message without any images will receive a "clean" verdict.

Note Images cannot be extracted from PDF files.

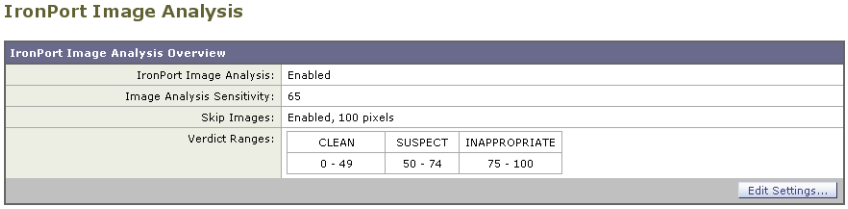
To enable image analysis from the GUI:

Step 1 Go to Security Services > IronPort Image Analysis.

Step 2 Click Enable.

A success message displays, and the verdict settings display.

Figure 6-3 Cisco IronPort Image Analysis Overview



The image analysis filter rule allows you to determine the actions to take based on the following verdicts:

- **Clean:** The image is free of inappropriate content. The image analysis verdict is computed on the message as a whole, so a message without any images will receive a "clean" verdict if scanned.
- **Suspect**: The image may contain inappropriate content.
- **Inappropriate**: The image contains inappropriate content.

These verdicts represent a numeric value assigned by the image analyzer algorithm to determine probability of inappropriate content.

The following values are recommended:

- Clean: 0 to 49
- Suspect: 50 to 74
- Inappropriate: 75 to 100

You can fine-tune image scanning by configuring the sensitivity setting, which helps reduce the number of false positives. For example, if you find that you are getting false positives, you can decrease the sensitivity setting. Or, conversely, if you find that the image scanning is missing inappropriate content, you may want to set the sensitivity higher. The sensitivity setting is a value between 0 (no sensitivity) and 100 (highly sensitive). The default sensitivity setting of 65 is recommended.

Configuring Scanning Values

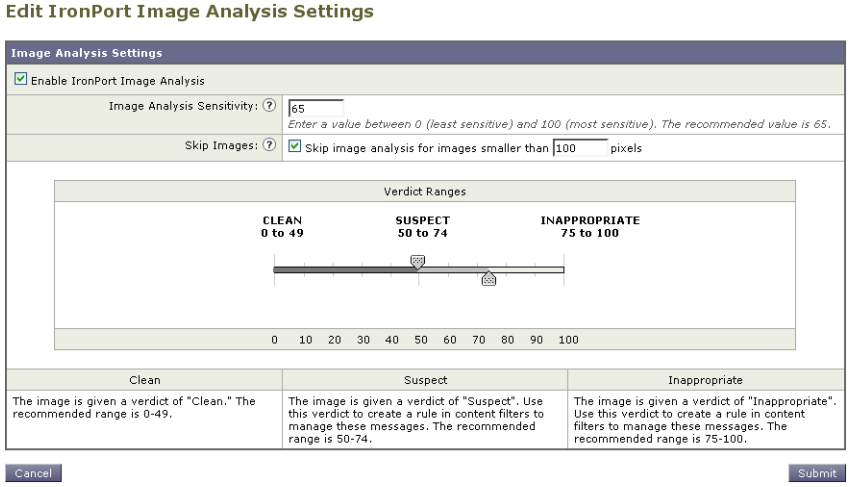
To configure scanning values:

Step 1 Go to Security Services > IronPort Image Analysis.

Step 2 Click Edit Settings

The Edit IronPort Image Analysis Settings page opens:

Figure 6-4 Edit IronPort Image Analysis Settings



Step 3 Configure the settings for image analysis sensitivity. The default sensitivity setting of 65 is recommended.

Step 4 Configure the settings for Clean, Suspect, and Inappropriate verdicts.

When you configure the value ranges, ensure that you do not overlap values and that you use whole integers.

Step 5 Optionally, configure AsyncOS to bypass scanning images that do not meet a minimum size requirement (recommended). By default, this setting is configured for 100 pixels. Scanning images that are smaller than 100 pixels can sometimes result in false positives.

You can also enable image analysis settings from the CLI via the `imageanalysisconfig` command:

```
test.com> imageanalysisconfig


IronPort Image Analysis: Enabled

Image Analysis Sensitivity: 65

Verdict Ranges: Clean (0-49), Suspect(50-74), Inappropriate (75+)
```

```
Skip small images with size less than 100 pixels (width or height)


Choose the operation you want to perform:

- SETUP - Configure IronPort Image Analysis.

[]> setup


IronPort Image Analysis: Enabled

Would you like to use IronPort Image Analysis? [Y]>


Define the image analysis sensitivity. Enter a value between 0 (least

sensitive) and 100 (most sensitive). As sensitivity increases, so does the false positive
rate. The default setting of 65 is recommended.

[65]>


Define the range for a CLEAN verdict. Enter the upper bound of the CLEAN range by
entering a value between 0 and 98. The default setting of 49 is recommended.

[49]>


Define the range for a SUSPECT verdict. Enter the upper bound of the SUSPECT range by
entering a value between 50 and 99. The default setting of 74 is recommended.

[74]>


Would you like to skip scanning of images smaller than a specific size? [Y]>


Please enter minimum image size to scan in pixels, representing either height or width of
a given image.

[100]>
```

Viewing Verdict Results

To see the verdict score for a particular message, you can view the mail logs. The mail logs display the image name or file name, the score for a particular message attachment. In addition, the log displays information about whether the images in a file were scannable or unscannable. Note that information in the log describes the result for each message attachment, rather than each image. For example, if the

message had a zip attachment that contained a JPEG image, the log entry would contain the name of the zip file rather than the name of the JPEG. Also, if the zip file included multiple images then the log entry would include the maximum score of all the images. The unscannable notation indicates whether any of the images were unscannable.

The log does not contain information about how the scores translate to a particular verdict (clean, suspect or inappropriate). However, because you can use mail logs to track the delivery of specific messages, you can determine by the actions performed on the messages whether the mail contained inappropriate or suspect images.

For example, the following mail log shows attachments dropped by message filter rules as a result of Image Analysis scanning:

```
Thu Apr  3 08:17:56 2009 Debug: MID 154 IronPort Image Analysis: image 'Unscannable.jpg'
is unscannable.

Thu Apr  3 08:17:56 2009 Info: MID 154 IronPort Image Analysis: attachment
'Unscannable.jpg' score 0 unscannable



Thu Apr  3 08:17:56 2009 Info: MID 6 rewritten to MID 7 by
drop-attachments-where-image-verdict filter 'f-001'



Thu Apr  3 08:17:56 2009 Info: Message finished MID 6 done
```

Using the Image Analysis Message Filter

Once you enable image analysis, you must create a message filter to perform different actions for different message verdicts. For example, you may wish to deliver messages with a clean verdict, but quarantine messages that are determined to have inappropriate content.



Note Cisco recommends you do not drop or bounce messages with inappropriate or suspect verdicts. Instead, send copies of violations to a quarantine for later review and better understanding of trend analysis.

The following filter shows messages tagged if the content is inappropriate or suspect:

```
image_analysis: if image-verdict == "inappropriate" {

strip-header("Subject");

insert-header("Subject", "[inappropriate image] $Subject");

}

 else {

if image-verdict == "suspect" {

strip-header("Subject");

insert-header("Subject", "[suspect image] $Subject");
```

```
}

}
```

Using Image Analysis Content Filters

After you enable image analysis, you can create a content filter to strip attachments based on image analysis verdicts, or you can configure a filter to perform different actions for different message verdicts. For example, you might decide to quarantine messages that contain inappropriate content.

To strip attachments based on image analysis verdicts:

Step 1 Click Mail Policies > Incoming Content Filters.

Step 2 Click Add Filter.

Step 3 Enter a name for the content filter.

Step 4 Under Actions, click **Add Action.**

Step 5 Under Strip Attachment by File Info, click **Image Analysis Verdict is**:

Step 6 Select from the following image analysis verdicts:

- Suspect
- Inappropriate
- Suspect or Inappropriate
- Unscannable
- Clean

To configure an action based on image analysis verdicts:

Step 1 Click Mail Policies > Incoming Content Filters.

Step 2 Click Add Filter.

Step 3 Enter a name for the content filter.

Step 4 Under Conditions, click **Add Condition**.

Step 5 Under Attachment File Info, click **Image Analysis Verdict**.

Step 6 Choose from one of the following verdicts:

- Suspect
- Inappropriate
- Suspect or Inappropriate
- Unscannable
- Clean

Step 7 Click **Add Action**.

Step 8 Select an action to perform on messages based on the image analysis verdict.

Step 9 Submit and commit your changes.

Notifications

Using the Text Resources page in the GUI or the `textconfig` CLI command to configure custom notification templates as text resources is another useful tool when used in conjunction with attachment filtering rules. The notification template supports non-ASCII characters (you are prompted to choose an encoding while creating the template).

In the following example, the `textconfig` command was first used to create a notification template named `strip.mp3` that will be inserted into to the body of the notification message. Then, an attachment filtering rule is created so that when an .mp3 file has been stripped from a message, a notification email is sent to the intended recipients explaining that the .mp3 file has been deleted.

```
drop-mp3s:

if (attachment-type == '*/mp3')

{ drop-attachments-by-filetype('Media');

  notify ('$EnvelopeRecipients', 'Your mp3 has been removed', '$EnvelopeFrom',
'strip.mp3');

}
```

For more information, see Notify and Notify-Copy Actions, page 6-52.

Examples of Attachment Scanning Message Filters

The following examples shows actions performed on attachments.

Inserting Headers

In these examples, AsyncOS inserts headers when the attachments contain specified content.

In the following example, all of the attachments on the message are scanned for a keyword. If the keyword is present in all of the attachments, a custom X-Header is inserted:

```
attach_disclaim:

    if (every-attachment-contains('[dD]isclaimer') ) {

        insert-header("X-Example-Approval", "AttachOK");

    }
```

In the following example, the attachment is scanned for a pattern in the binary data. The filter uses the `attachment-binary-contains` filter rule to search for a pattern that indicates that the PDF document is encrypted. If the pattern is present in the binary data, a custom header is inserted:

```
match_PDF_Encrypt:

if (attachment-filetype == 'pdf' AND

attachment-binary-contains('/Encrypt')){

strip-header (‘Subject’);

insert-header (‘Subject’, ‘[Encrypted] $Subject’);

}
```

Dropping Attachments by File Type

In the following example, the “executable” group of attachments (`.exe`, `.dll`, and `.scr`) is stripped from messages and text is added to the message, listing the filenames of the dropped files (via the `$dropped_filename` action variable). Note that the `drop-attachments-by-filetype` action examines attachments and strips them based on the fingerprint of the file, and not just the three-letter filename extension. Note also that you can specify a single filetype (“mpeg”) or you can refer to all of the members of the filetype (“Media”):

```
strip_all_exes: if (true) {

                    drop-attachments-by-filetype ('Executable', “Removed attachment:
$dropped_filename”);

                }
```

In the following example, the same “executable” group of attachments (`.exe`, `.dll`, and `.scr`) are stripped from messages whose Envelope Sender is not within the domain `example.com`.

```
strip_inbound_exes: if (mail-from != "@example\\.com$") {

                        drop-attachments-by-filetype ('Executable');

                    }
```

In the following example, a specific member of a file type (“wmf”) as well as a the same “executable” group of attachments (`.exe`, `.dll`, and `.scr`) are stripped from messages whose Envelope Sender is not within the domain `example.com`.

```
strip_inbound_exes_and_wmf: if (mail-from != "@example\\.com$") {

                        drop-attachments-by-filetype ('Executable');

                        drop-attachments-by-filetype ('x-wmf');

                    }
```

In the following example, the "executable" pre-defined group of attachments is extended to include more attachment names. (Note that this action will *not* examine the attachments' file type.)

```
strip_all_dangerous: if (true) {

                    drop-attachments-by-filetype ('Executable');

                      drop-attachments-by-name('(?i)\\.(cmd|pif|bat)$');

                     }
```

The `drop-attachments-by-name` action supports non-ASCII characters.

Note The `drop-attachments-by-name` action matches the regular expression against the filename captured from the MIME header. The filename captured from the MIME header may contain trailing spaces.

In the following example, a message is dropped if the attachment is not an .exe executable filetype. However, the filter will not perform any action on the message if there is at least one attachment with the file type you want to filter out. For example, the following filter drops any message with an attachment that is not an *.exe* file type:

```
exe_check: if (attachment-filetype != "exe") {

                    drop();

                }
```

If a message has multiple attachments, the Email Security appliance does not drop the message if at least one of the attachments is an `.exe` file, even if the other attachments not `.exe` files.

Dropping Attachments by Dictionary Matches

This `drop-attachments-where-dictionary-match` action strips attachments based on matches to dictionary terms. If the terms in the MIME parts considered to be an attachment match a dictionary term (and the user-defined threshold is met), the attachment is stripped from the email. The following example shows attachment drops if words in the "secret_words" dictionary are detected in the attachment. Note that the threshold for the matches is set to one:

```
Data_Loss_Prevention: if (true) {

drop-attachments-where-dictionary-match("secret_words", 1);

}
```

Quarantining Protected Attachments

The `attachment-protected` filter tests whether any attachment in the message is password protected. You might use this filter on incoming mail to ensure that the attachments are scannable. According to this definition, a zip file containing one encrypted member along with unencrypted members will be

considered protected. Similarly, PDF file that has no open password will not be considered protected, even though it may restrict copying or printing with a password. The following example shows protected attachments sent to a policy quarantine:

```
quarantine_protected:

if attachment-protected

{

quarantine("Policy");

}
```

Detecting Unprotected Attachments

The `attachment-unprotected` filter tests whether any attachment in the message is *not* password protected. This message filter complements the `attachment-protected` filter. You might use this filter on outgoing mail to detect outgoing mail that is unprotected. The following example shows AsyncOS detecting unprotected attachments on an outgoing listener and quarantining the messages:

```
quarantine_unprotected:

if attachment-unprotected

{

quarantine("Policy");

}
```

Using the CLI to Manage Message Filters

You can use the CLI to add, delete, activate and de-activate, import and export, and set logging options for message filters. The table below shows a summary of the commands and subcommands.

Table 6-9 Message Filters Subcommands

Syntax	Description
`filters`	The main command. This command is interactive; it asks you for more information (for example, `new`, `delete`, `import`).
`new`	Creates a new filter. If no location is given, it is appended to the current sequence. Otherwise, the filter will be inserted into the specific place in the sequence. For more information, see Creating a New Message Filter, page 6-76.
`delete`	Deletes a filter by name or by sequence number. For more information, see Deleting a Message Filter, page 6-77.
`move`	Rearranges the existing filters. For more information, see Moving a Message Filter, page 6-77.
`set`	Sets filter to active or inactive state. For more information, see Activating and Deactivating a Message Filter, page 6-77.

Table 6-9 Message Filters Subcommands (Continued)

Syntax	Description
`import`	Replaces the current set of filters with a new set stored in a file (in the /configuration directory of the appliance). For more information, see Importing Message Filters, page 6-81.
`export`	Exports the current set of filters to a file (in the /configuration directory of the appliance). For more information, see Exporting Message Filters, page 6-81.
list	Lists information about a filter or filters. For more information, see Displaying a Message Filter List, page 6-82.
detail	Prints detailed information about a specific filter, including the body of the filter rule itself. For more information, see Displaying Message Filter Details, page 6-82.
logconfig	Enters the logconfig submenu of filters, allowing you to edit the log subscriptions from `archive()` filter actions. For more information, see Configuring Filter Log Subscriptions, page 6-82.

Note You must issue the `commit` command for filters to take effect.

Three types of parameters are:

Table 6-10 Filter Management Parameters

seqnum	An integer representing a filter based on its position in the list of filters. A *seqnum* of `2` represents the second filter in the list, for example.
filtname	The colloquial name of a filter.
range	A range may be used to represent more than one filter, and appears in the form of *X*-*Y*, where X and Y are the first and last *seqnums* that identify the extent. For example, `2-4` represents filters in the second, third, and fourth positions. Either X or Y may be left off to represent an open-ended list. For example, `-4` represents the first four filters, and `2-` represents all filters except the first. You can also use the keyword `all` to represents all the filters in the filter list.

Creating a New Message Filter

```
new [seqnum|filtname|last]
```

Specifies the position at which to insert the new filter(s). If omitted, or given the keyword `last`, the filters entered in are appended to the list of filters. No gaps in the sequence numbers are allowed; you are not allowed to enter a *seqnum* outside the boundaries of the current list. If you enter an unknown *filtname*, you are prompted to enter a valid *filtname*, *seqnum*, or `last`.

After a filter has been entered, you may manually enter the filter script. When you are finished typing, end the entry by typing a period (`.`) on a line by itself.

The following conditions can cause errors:

- Sequence number beyond the current range of sequence numbers.
- Filter with a non-unique *filtname*.
- Filter with a *filtname* that is a reserved word.

- Filter with a syntax error.
- Filter with actions referring to non-existent system resources such as interfaces.

Deleting a Message Filter

```
delete [seqnum|filtname|range]
```

Deletes the filter(s) identified.

The following conditions can cause errors:

- No filter with a given filter name.
- No filter with a given sequence number.

Moving a Message Filter

```
move [seqnum|filtname|range seqnum|last]
```

Moves the filters identified by the first parameter to the position identified by the second parameter. If the second parameter is the keyword `last`, the filters are moved to the end of the list of filters. If more than one filter is being moved, their ordering remains the same in relation to one another.

The following conditions can cause errors:

- No filter with a given filter name.
- No filter with a given sequence number.
- Sequence number beyond the current range of sequence numbers.
- Movement would result in no change of sequence.

Activating and Deactivating a Message Filter

A given message filter is either *active* or *inactive* and it is also either *valid* or *invalid*. A message filter is only used for processing if it is both *active* and *valid*. You change an existing filter from active to inactive (and back again) via the CLI. A filter is invalid if it refers to a listener or interface which does not exist (or has been removed).



Note You can determine if a filter is inactive by its syntax; AsyncOS changes the colon after the filter name to an exclamation point for inactive filters. If you use this syntax when entering or importing a filter, AsyncOS marks the filter as inactive.

For example, the following benign filter named “filterstatus” is entered. It is then made inactive using the `filter -> set` subcommand. Note that when the details of the filter are shown, the colon has been changed to an exclamation point (and is bold in the following example).

```
mail3.example.com> filters


Choose the operation you want to perform:
```

```
- NEW - Create a new filter.

- IMPORT - Import a filter script from a file.

[]> new


Enter filter script.  Enter '.' on its own line to end.

filterstatus: if true{skip_filters();}

.

1 filters added.


Choose the operation you want to perform:

- NEW - Create a new filter.

- DELETE - Remove a filter.

- IMPORT - Import a filter script from a file.

- EXPORT - Export filters to a file

- MOVE - Move a filter to a different position.

- SET - Set a filter attribute.

- LIST - List the filters.

- DETAIL - Get detailed information on the filters.

- LOGCONFIG - Configure log subscriptions used by filters.

- ROLLOVERNOW - Roll over a filter log file.

[]> list


Num Active Valid Name

  1   Y      Y   filterstatus


Choose the operation you want to perform:

- NEW - Create a new filter.

- DELETE - Remove a filter.

- IMPORT - Import a filter script from a file.
```

```
- EXPORT - Export filters to a file
- MOVE - Move a filter to a different position.
- SET - Set a filter attribute.
- LIST - List the filters.
- DETAIL - Get detailed information on the filters.
- LOGCONFIG - Configure log subscriptions used by filters.
- ROLLOVERNOW - Roll over a filter log file.
[]> set

Enter the filter name, number, or range:
[all]> all

Enter the attribute to set:
[active]> inactive

1 filters updated.

Choose the operation you want to perform:
- NEW - Create a new filter.
- DELETE - Remove a filter.
- IMPORT - Import a filter script from a file.
- EXPORT - Export filters to a file
- MOVE - Move a filter to a different position.
- SET - Set a filter attribute.
- LIST - List the filters.
- DETAIL - Get detailed information on the filters.
- LOGCONFIG - Configure log subscriptions used by filters.
- ROLLOVERNOW - Roll over a filter log file.
[]> detail
```

```
Enter the filter name, number, or range:
[]> all



Num Active Valid Name
  1   N      Y   filterstatus
filterstatus! if (true) {
 skip_filters();
              }



Choose the operation you want to perform:
- NEW - Create a new filter.
- DELETE - Remove a filter.
- IMPORT - Import a filter script from a file.
- EXPORT - Export filters to a file
- MOVE - Move a filter to a different position.
- SET - Set a filter attribute.
- LIST - List the filters.
- DETAIL - Get detailed information on the filters.
- LOGCONFIG - Configure log subscriptions used by filters.
- ROLLOVERNOW - Roll over a filter log file.
[]>
```

Activating or Deactivating a Message Filter

`set [`*`seqnum`*`|`*`filtname`*`|`*`range`*`] active|inactive`

Sets the filters identified to have the given state. Legal states are:

- active: Set the state of the selected filters to be active.
- inactive: Set the state of the selected filters to be inactive.

The following conditions can cause errors:

- No filter with a given *filtname*.
- No filter with a given sequence number.

Note A filter which is inactive may also be noted in its syntax; the colon after the label (name of the filter) is changed to an exclamation point (`!`). A filter entered manually from the CLI, or imported, that contains this syntax, will automatically be marked inactive. For example, `mailfrompm!` instead of `mailfrompm:` is displayed.

Importing Message Filters

```
import filename
```

The name of the file containing filters to be processed. This file must reside in the configuration directory of the FTP/SCP root directory on the appliance, if you enabled FTP/SCP access for the interface with the `interfaceconfig` command. It is ingested and parsed, and any errors are reported. The filters imported replace all filters existing in the current filter set. See Appendix B, "Accessing the Appliance" for more information. Consider exporting the current filter list (see Exporting Message Filters, page 6-81) and then editing that file before importing.

When importing message filters, you are prompted to select the encoding used.

The following conditions can cause errors:

- File does not exist.
- Filter with a non-unique filter name.
- Filter with a *filtname* that is a reserved word.
- Filter with a syntax error.
- Filter with actions referring to non-existent system resources such as interfaces.

Exporting Message Filters

```
export filename [seqnum|filtname|range]
```

Output a formatted version of the existing filter set to a file in the configuration directory of the FTP/SCP root directory on the appliance. See Appendix B, "Accessing the Appliance" for more information.

When exporting message filters, you are prompted to select the encoding used.

The following conditions can cause errors:

- No filter with a given filter name.
- No filter with a given sequence number.

Viewing Non-ASCII Character Sets

The system displays filters containing non-ASCII characters in the CLI in UTF-8. If your terminal/display does not support UTF-8, the filter will unreadable.

The best way to manage non-ASCII characters in filters is to edit the filter in a text file and then import that text file (see Importing Message Filters, page 6-81) into the appliance.

Displaying a Message Filter List

```
list [seqnum|filtname|range]
```

Shows summarized information about the identified filters in a tabular form without printing the filter body. The information displayed includes:

- Filter name
- Filter sequence number
- Filter's active/inactive state
- Filter's valid/invalid state

The following conditions can cause errors:

- Illegal range format.

Displaying Message Filter Details

```
detail [seqnum|filtname|range]
```

Provides full information about the identified filters, including the body of the filter and any additional state information.

Configuring Filter Log Subscriptions

```
logconfig
```

Enters a submenu that allows you to configure the filter log options for the mailbox files generated by the `archive()` action. These options are very similar to those used by the regular `logconfig` command, but the logs may only be created or deleted by adding or removing filters that reference them.

Each filter log subscription has the following default values, which can be modified using the `logconfig` subcommand:

- Retrieval method - FTP Poll
- File size - 10MB
- Max number of files - 10

For more information, see "Logging" in the *Cisco IronPort AsyncOS for Email Daily Management Guide*.

```
mail3.example.com> filters


Choose the operation you want to perform:

- NEW - Create a new filter.
```

```
- DELETE - Remove a filter.

- IMPORT - Import a filter script from a file.

- EXPORT - Export filters to a file

- MOVE - Move a filter to a different position.

- SET - Set a filter attribute.

- LIST - List the filters.

- DETAIL - Get detailed information on the filters.

- LOGCONFIG - Configure log subscriptions used by filters.

- ROLLOVERNOW - Roll over a filter log file.

[]> logconfig


Currently configured logs:

1. "joesmith" Type: "Filter Logs" Retrieval: FTP Poll


Choose the operation you want to perform:

- EDIT - Modify a log setting.

[]> edit


Enter the number of the log you wish to edit.

[]> 1


Choose the method to retrieve the logs.

1. FTP Poll

2. FTP Push

3. SCP Push

[1]> 1


Please enter the filename for the log:

[joesmith.mbox]>
```

```
Please enter the maximum file size:

[10485760]>


Please enter the maximum number of files:

[10]>


Currently configured logs:

1. "joesmith" Type: "Filter Logs" Retrieval: FTP Poll


Enter "EDIT" to modify or press Enter to go back.

[]>
```

Modifying Scanning Parameters

The `scanconfig` command controls the behavior of body and attachment scanning, such as which types should be skipped when scanning.

Note If you want to scan a MIME type that may be included in a zip or compressed file, you must include list 'compressed' or 'zip' or 'application/zip' in the scan list.

Using scanconfig

In the following example, the `scanconfig` command sets the following parameters:

- MIME types of `video/*`, `audio/*`, `image/*` are not scanned for content.
- Nested (recursive) archive attachments up to 10 levels are scanned. (The default is 5 levels.)
- The maximum size for attachments to be scanned is 25 megabytes; anything larger will be skipped. (The default is 5 megabytes.)
- The attachment is enabled for metadata scanning. When the scanning engine scans attachments, it scans the metadata for the regular expression. This is the default setting.
- The attachment timeout scanning is configured for 60 seconds. The default is 30 seconds.
- Attachments that were not scanned are assumed to not match the search pattern. (This is the default behavior.)
- The `application/(x-)pkcs7-mime` (opaque-signed) parts of a message are converted to `multipart/signed` (clear-signed) to provide the message's content for processing. The default is not to convert opaque-signed messages.

Note When setting the `assume the attachment matches the search pattern` to Y, messages that cannot be scanned will cause the message filter rule to evaluate to true. This could result in unexpected behavior, such as the quarantining of messages that do not match a dictionary, but were quarantined because their content could not be correctly scanned.

```
mail3.example.com> scanconfig

There are currently 5 attachment type mappings configured to be SKIPPED.

Choose the operation you want to perform:
- NEW - Add a new entry.
- DELETE - Remove an entry.
- SETUP - Configure scanning behavior.
- IMPORT - Load mappings from a file.
- EXPORT - Save mappings to a file.
- PRINT - Display the list.
- CLEAR - Remove all entries.
- SMIME - Configure S/MIME unpacking.
[]> setup

1. Scan only attachments with MIME types or fingerprints in the list.
2. Skip attachments with MIME types or fingerprints in the list.
Choose one:
[2]> 2

Enter the maximum depth of attachment recursion to scan:
[5]> 10

Enter the maximum size of attachment to scan:
[5242880]> 10m
```

```
Do you want to scan attachment metadata? [Y]> Y


Enter the attachment scanning timeout (in seconds):

[30]> 60


If a message has attachments that were not scanned for any reason (e.g. because of size,
depth limits, or scanning timeout), assume the attachment matches the search pattern?
[N]>


If a message could not be deconstructed into its component parts in order to remove
specified attachments, the system should:

1. Deliver

2. Bounce

3. Drop

[1]> 1


Configure encoding to use when none is specified for plain body text or anything with
MIME type plain/text or plain/html.

1. US-ASCII

2. Unicode (UTF-8)

3. Unicode (UTF-16)

4. Western European/Latin-1 (ISO 8859-1)

5. Western European/Latin-1 (Windows CP1252)

6. Traditional Chinese (Big 5)

7. Simplified Chinese (GB 2312)

8. Simplified Chinese (HZ GB 2312)

9. Korean (ISO 2022-KR)

10. Korean (KS-C-5601/EUC-KR)

11. Japanese (Shift-JIS (X0123))

12. Japanese (ISO-2022-JP)
```

```
13. Japanese (EUC)

[1]>


Scan behavior changed.


There are currently 5 attachment type mappings configured to be SKIPPED.


Choose the operation you want to perform:

- NEW - Add a new entry.

- DELETE - Remove an entry.

- SETUP - Configure scanning behavior.

- IMPORT - Load mappings from a file.

- EXPORT - Save mappings to a file.

- PRINT - Display the list.

- CLEAR - Remove all entries.

- SMIME - Configure S/MIME unpacking.

[]> SMIME


Do you want to convert opaque-signed messages to clear-signed? This will provide the
clear text content for various blades to process. [N]> Y




There are currently 5 attachment type mappings configured to be SKIPPED.


Choose the operation you want to perform:

- NEW - Add a new entry.

- DELETE - Remove an entry.

- SETUP - Configure scanning behavior.

- IMPORT - Load mappings from a file.
```

```
- EXPORT - Save mappings to a file.
- PRINT - Display the list.
- CLEAR - Remove all entries.
- SMIME - Configure S/MIME unpacking.
[]> print


1. Fingerprint   Image
2. Fingerprint   Media
3. MIME Type     audio/*
4. MIME Type     image/*
5. MIME Type     video/*



There are currently 5 attachment type mappings configured to be SKIPPED.


Choose the operation you want to perform:
- NEW - Add a new entry.
- DELETE - Remove an entry.
- SETUP - Configure scanning behavior.
- IMPORT - Load mappings from a file.
- EXPORT - Save mappings to a file.
- PRINT - Display the list.
- CLEAR - Remove all entries.
- SMIME - Configure S/MIME unpacking.
[]>
```

Changing Message Encoding

You can use the `localeconfig` command to set the behavior of AsyncOS regarding modifying the encoding of message headings and footers during message processing:

```
example.com> localeconfig


Behavior when modifying headers: Use encoding of message body

Behavior for untagged non-ASCII headers: Impose encoding of message body

Behavior for mismatched footer or heading encoding: Only try encoding from

message body


Choose the operation you want to perform:

- SETUP - Configure multi-lingual settings.

[]> setup


If a header is modified, encode the new header in the same encoding as

the message body? (Some MUAs incorrectly handle headers encoded in a

different encoding than the body. However, encoding a modified header

in the same encoding as the message body may cause certain characters in the modified
header to be lost.) [Y]>


If a non-ASCII header is not properly tagged with a character set and

is being used or modified, impose the encoding of the body on the

header during processing and final representation of the message?

(Many MUAs create non-RFC-compliant headers that are then handled in

an undefined way. Some MUAs handle headers encoded in character sets

that differ from that of the main body in an incorrect way. Imposing the encoding of the
body on the header may encode

the header more precisely. This will be used to interpret the content of headers for
processing, it will not modify or rewrite the header

 unless that is done explicitly as part of the processing.) [Y]>
```

```
Footers or headings are added in-line with the message body whenever
possible. However, if the footer or heading is encoded differently
than the message body,and if imposing a single encoding will cause
loss of characters, it will be added as an attachment. The system will
always try to use the message body's encoding for the footer or
heading. If that fails, and if the message body's encoding is US-
ASCII, the system can try to edit the message body to use the footer's
or heading's encoding. Should the system try to impose the footer's
or headings's encoding on the message body? [N]> y

Behavior when modifying headers: Use encoding of message body
Behavior for untagged non-ASCII headers: Impose encoding of message
body. Behavior for mismatched footer or heading encoding: Try both
body and footer or heading encodings

Choose the operation you want to perform:
- SETUP - Configure multi-lingual settings.
```

The first prompt determines whether or not a message header's encoding should be changed to match that of the message body if the header is changed (via a filter, for example).

The second prompt controls whether or not the appliance should impose the encoding of the message body on the header if the header is not properly tagged with a character set.

The third prompt is used to configure how disclaimer stamping (and multiple encodings) in the message body works. Please see "Disclaimer Stamping and Multiple Encodings" in the "Text Resources" chapter in the *Cisco IronPort AsyncOS for Email Configuration Guide* for more information.

Creating Sample Message Filters

In the following example, the `filter` command is used to create three new filters:

- The first filter is named **`big_messages`**. It uses the `body-size` rule to drop messages larger than 10 megabytes.
- The second filter is named **`no_mp3s`**. It uses the `attachment-filename` rule to drop messages that contain attachments with the filename extension of `.mp3`.

- The third filter is named **mailfrompm**. It uses `mail-from` rule examines all mail from `postmaster@example.com` and blind-carbon copies `administrator@example.com`.

Using the `filter -> list` subcommand, the filters are listed to confirm that they are active and valid, and then the first and last filters are switched in position using the `move` subcommand. Finally, the changes are committed so that the filters take effect.

```
mail3.example.com> filters


Choose the operation you want to perform:

- NEW - Create a new filter.

- IMPORT - Import a filter script from a file.

[]> new


Enter filter script.  Enter '.' on its own line to end.

big_messages:

    if (body-size >= 10M) {

       drop();

    }

.

1 filters added.


Choose the operation you want to perform:

- NEW - Create a new filter.

- DELETE - Remove a filter.

- IMPORT - Import a filter script from a file.

- EXPORT - Export filters to a file

- MOVE - Move a filter to a different position.

- SET - Set a filter attribute.

- LIST - List the filters.

- DETAIL - Get detailed information on the filters.

- LOGCONFIG - Configure log subscriptions used by filters.
```

```
- ROLLOVERNOW - Roll over a filter log file.

[]> new


Enter filter script.  Enter '.' on its own line to end.

no_mp3s:

    if (attachment-filename == '(?i)\\.mp3$') {

        drop();

     }

.

1 filters added.


Choose the operation you want to perform:

- NEW - Create a new filter.

- DELETE - Remove a filter.

- IMPORT - Import a filter script from a file.

- EXPORT - Export filters to a file

- MOVE - Move a filter to a different position.

- SET - Set a filter attribute.

- LIST - List the filters.

- DETAIL - Get detailed information on the filters.

- LOGCONFIG - Configure log subscriptions used by filters.

- ROLLOVERNOW - Roll over a filter log file.

[]> new


Enter filter script.  Enter '.' on its own line to end.

mailfrompm:

    if (mail-from == "^postmaster$")

     { bcc ("administrator@example.com");}

.
```

```
1 filters added.


Choose the operation you want to perform:
- NEW - Create a new filter.
- DELETE - Remove a filter.
- IMPORT - Import a filter script from a file.
- EXPORT - Export filters to a file
- MOVE - Move a filter to a different position.
- SET - Set a filter attribute.
- LIST - List the filters.
- DETAIL - Get detailed information on the filters.
- LOGCONFIG - Configure log subscriptions used by filters.
- ROLLOVERNOW - Roll over a filter log file.
[]> list


Num Active Valid Name
  1   Y      Y   big_messages
  2   Y      Y   no_mp3s
  3   Y      Y   mailfrompm


Choose the operation you want to perform:
- NEW - Create a new filter.
- DELETE - Remove a filter.
- IMPORT - Import a filter script from a file.
- EXPORT - Export filters to a file
- MOVE - Move a filter to a different position.
- SET - Set a filter attribute.
- LIST - List the filters.
- DETAIL - Get detailed information on the filters.
```

```
- LOGCONFIG - Configure log subscriptions used by filters.
- ROLLOVERNOW - Roll over a filter log file.
[]> move

Enter the filter name, number, or range to move:
[]> 1

Enter the target filter position number or name:
[]> last

1 filters moved.

Choose the operation you want to perform:
- NEW - Create a new filter.
- DELETE - Remove a filter.
- IMPORT - Import a filter script from a file.
- EXPORT - Export filters to a file
- MOVE - Move a filter to a different position.
- SET - Set a filter attribute.
- LIST - List the filters.
- DETAIL - Get detailed information on the filters.
- LOGCONFIG - Configure log subscriptions used by filters.
- ROLLOVERNOW - Roll over a filter log file.
[]> list

Num Active Valid Name
  1   Y      Y   no_mp3s
  2   Y      Y   mailfrompm
  3   Y      Y   big_messages
```

```
Choose the operation you want to perform:
- NEW - Create a new filter.
- DELETE - Remove a filter.
- IMPORT - Import a filter script from a file.
- EXPORT - Export filters to a file
- MOVE - Move a filter to a different position.
- SET - Set a filter attribute.
- LIST - List the filters.
- DETAIL - Get detailed information on the filters.
- LOGCONFIG - Configure log subscriptions used by filters.
- ROLLOVERNOW - Roll over a filter log file.
[]> move

Enter the filter name, number, or range to move:
[]> 2

Enter the target filter position number or name:
[]> 1

1 filters moved.

Choose the operation you want to perform:
- NEW - Create a new filter.
- DELETE - Remove a filter.
- IMPORT - Import a filter script from a file.
- EXPORT - Export filters to a file
- MOVE - Move a filter to a different position.
- SET - Set a filter attribute.
- LIST - List the filters.
```

```
- DETAIL - Get detailed information on the filters.
- LOGCONFIG - Configure log subscriptions used by filters.
- ROLLOVERNOW - Roll over a filter log file.
[]> list

Num Active Valid Name
  1   Y      Y   mailfrompm
  2   Y      Y   no_mp3s
  3   Y      Y   big_messages

Choose the operation you want to perform:
- NEW - Create a new filter.
- DELETE - Remove a filter.
- IMPORT - Import a filter script from a file.
- EXPORT - Export filters to a file
- MOVE - Move a filter to a different position.
- SET - Set a filter attribute.
- LIST - List the filters.
- DETAIL - Get detailed information on the filters.
- LOGCONFIG - Configure log subscriptions used by filters.
- ROLLOVERNOW - Roll over a filter log file.
[]>

mail3.example.com> commit

Please enter some comments describing your changes:
[]> entered and enabled 3 filters: no_mp3s, mailfrompm, big_messages
```

Message Filter Examples

This section contains some real world examples of filters with a brief discussion of each.

Open-Relay Prevention Filter

This filter bounces messages with addresses using `%`, extra `@`, and `!` characters in email addresses:

- `user%otherdomain@validdomain`
- `user@otherdomain@validdomain:`
- `domain!user@validdomain`

```
sourceRouted:

if (rcpt-to == "(%|@|!)(.*)@") {

  bounce();

}
```

Cisco IronPort appliances are not susceptible to these third party relay hacks that are often used to exploit traditional Sendmail/Qmail systems. As many of these symbols (for example `%`) can be part of a perfectly legal email address, Cisco IronPort appliances will accept these as valid addresses, verify them against the configured recipient lists, and pass them on to the next internal server. Cisco IronPort appliances do not relay these messages to the world.

These filters are put in place to protect users who may have open-source MTAs that are misconfigured to allow relay of these types of messages.

Note You can also configure a listener to handle these types of addresses. See SMTP Address Parsing Options, page 1-8 for more information.

Policy Enforcement Filters

Notify Based on Subject Filter

This filter sends notification based on whether the subject contains specific words:

```
search_for_sensitive_content:

if (Subject == "(?i)plaintiff|lawsuit|judge" ) {

    notify ("admin@company.com");

}
```

BCC and Scan Mail Sent to Competitors

This filter scans and blind copies messages that are sent to competitors. Note that you could use a dictionary and the header-dictionary-match() rule to specify a more flexible list of competitors (see Dictionary Rules, page 6-33):

```
competitorFilter:

if (rcpt-to == '@competitor1.com|@competitor2.com') {

    bcc-scan('legal@example.com');

}
```

Block Specific User Filter

Use this filter to block email from a specific address:

```
block_harrasing_user:

if (mail-from == "ex-employee@hotmail\\.com") {

    notify ("admin@company.com");

    drop ();

}
```

Archive and Drop Messages Filter

Log and drop only the messages that have matching filetypes:

```
drop_attachments:

if (mail-from != "user@example.com") AND (attachment-filename ==

'(?i)\\.(asp|bas|bat|cmd|cpl|exe|hta|ins|isp|js)$')

{

  archive("Drop_Attachments");

  insert-header("X-Filter", "Dropped by: $FilterName MID: $MID");
drop-attachments-by-name("\\.(asp|bas|bat|cmd|cpl|exe|hta|ins|isp|js)$");

}
```

Large “To:” Header Filter

Find messages with very large “To” headers.

Use the `archive()` line for verification of proper action, with drop() enabled or disabled for extra safety:

```
toTooBig:

if(header('To') == "^.{500,}") {

   archive('tooTooBigdropped');

   drop();

}
```

Blank “From:” Filter

Identify blank “From” headers,

This filter can alleviate various forms of blank “from” addresses:

```
blank_mail_from_stop:

if (recv-listener == "InboundMail" AND header("From") == "^$|<\\s*>") {

  drop ();

}
```

If you also want to drop messages with a blank envelope from, use this filter:

```
blank_mail_from_stop:

if (recv-listener == "InboundMail" AND (mail-from == "^$|<\\s*>" OR header ("From") ==
"^$|<\\s*>"))

{

  drop ();

}
```

SRBS Filter

SenderBase Reputation filter:

```
note_bad_reps:

if (reputation < -2)  {

  strip-header ('Subject');

  insert-header ('Subject', '***BadRep $Reputation *** $Subject');

}
```

Alter SRBS Filter

Alter the (SenderBase Reputation Score) SBRS threshold for certain domains:

```
mod_sbrs:

if ( (rcpt-count == 1) AND (rcpt-to == "@domain\\.com$") AND (reputation < -2) ) {

    drop ();

}
```

Filename Regex Filter

This filter specifies a range of size for the body of the message, and looks for an attachment that matches the regular expression (this matches files named “readme.zip”, “readme.exe”, “attach.exe”, and so forth.):

```
filename_filter:

if ((body-size >= 9k) AND (body-size <= 20k)) {

   if (body-contains ("(?i)(readme|attach|information)\\.(zip|exe)$")) {

      drop ();

   }

}
```

Show SenderBase Reputation Score in Header Filter

Remember to log the headers (see “Logging” in the *Cisco IronPort AsyncOS for Email Daily Management Guide*) so they appear in the mail log:

```
Check_SBRS:

if (true) {

  insert-header('X-SBRS', '$Reputation');

}
```

Insert Policy into Header Filter

Show which mail flow policy accepted the connection:

```
Policy_Tracker:

if (true) {
```

```
  insert-header ('X-HAT', 'Sender Group $Group, Policy $Policy applied.');

}
```

Too Many Recipients Bounce Filter

Bounce all outbound email messages with more than 50 recipients from more than two unique domains:

```
bounce_high_rcpt_count:

if ( (rcpt-count > 49) AND (rcpt-to != "@example\\.com$") ) {

  bounce-profile ("too_many_rcpt_bounce"); bounce ();

}
```

Routing and Domain Spoofing

Using Virtual Gateways Filter

Segment traffic using virtual gateways. Assuming you have two Interfaces on the system, 'public1' and 'public2', and the default delivery interface is 'public1'. This would force all of your outbound traffic over the second interface; since bounces and other similar types of mail do not go through filters, they will be delivered from public1:

```
virtual_gateways:

if (recv-listener == "OutboundMail") {

    alt-src-host ("public2");

}
```

Same Listener for Deliver and Injection Filter

Use the same listener for delivery and receiving. This filter will allow you to send any messages received on the public listener “listener1” out the interface “listener1” (you will have to set up a unique filter for each public injector configured):

```
same_listener:

if (recv-inj == 'listener1') {

  alt-src-host('listener1');

}
```

Single Injector Filter

Make the filter work on a single listener. For example, specify a specific listener for message filter processing instead of being performed system wide.

```
textfilter-new:

if (recv-inj == 'inbound' and body-contains("some spammy message")) {

  alt-rcpt-to ("spam.quarantine@spam.example.com");

}
```

Drop Spoofed Domain Filter (Single Listener)

Drop email with a spoofed domain (pretending to be from an internal address; works with a single listener). IP addresses below represent fictional domain for `mycompany.com`:

```
DomainSpoofed:

if (mail-from == "mycompany\\.com$") {

  if ((remote-ip != "1.2.") AND (remote-ip != "3.4.")) {

    drop();

  }

}
```

Drop Spoofed Domain Filter (Multiple Listeners)

As above, but works with multiple listeners:

```
domain_spoof:

if ((recv-listener == "Inbound") and (mail-from == "@mycompany\\.com")) {

archive('domain_spoof');

drop ();

}
```

Another Drop Spoofed Domain Filter

Summary: Anti domain spoof filter:

```
reject_domain_spoof:

if (recv-listener == "MailListener") {
```

```
  insert-header("X-Group", "$Group");

  if ((mail-from == "@test\\.mycompany\\.com") AND (header("X-Group") != "RELAYLIST")) {

    notify("me@here.com");

    drop();

    strip-header("X-Group");

}
```

Detect Looping Filter

This filter is used to detect, stop, and determine what is causing, a mail loop. This filter can help determine a configuration issue on the Exchange server or elsewhere.

```
External_Loop_Count:

if (header("X-ExtLoop1")) {

  if (header("X-ExtLoopCount2")) {

    if (header("X-ExtLoopCount3")) {

      if (header("X-ExtLoopCount4")) {

        if (header("X-ExtLoopCount5")) {

          if (header("X-ExtLoopCount6")) {

            if (header("X-ExtLoopCount7")) {

              if (header("X-ExtLoopCount8")) {

                if (header("X-ExtLoopCount9")) {

                   notify ('joe@example.com');

                   drop();

                }

              else {insert-header("X-ExtLoopCount9", "from
                   $RemoteIP");}}

            else {insert-header("X-ExtLoopCount8", "from $RemoteIP");}}

          else {insert-header("X-ExtLoopCount7", "from $RemoteIP");}}

        else {insert-header("X-ExtLoopCount6", "from $RemoteIP");}}

      else {insert-header("X-ExtLoopCount5", "from $RemoteIP");}}

    else {insert-header("X-ExtLoopCount4", "from $RemoteIP");}}
```

```
  else {insert-header("X-ExtLoopCount3", "from $RemoteIP");}}
else {insert-header("X-ExtLoopCount2", "from $RemoteIP");}}
else {insert-header("X-ExtLoop1", "1"); }
```



Note By default, AsyncOS automatically detects mail loops and will drop messages after 100 loops.



CHAPTER 7

Advanced Network Configuration

This chapter includes information about advanced network configuration generally available via the `etherconfig` command, such as NIC pairing, VLANs, Direct Server Return, and more. This chapter contains the following sections:

- Media Settings on Ethernet Interfaces, page 7-1
- Network Interface Card Pairing/Teaming, page 7-3
- Virtual Local Area Networks (VLANs), page 7-8
- Direct Server Return, page 7-15

Media Settings on Ethernet Interfaces

Media settings for the ethernet interfaces can be accessed via the use of the `etherconfig` command. Each ethernet interface is listed with its current setting. By selecting the interface, the possible media settings are displayed. See Example of Editing Media Settings, page 7-2 for an example.

Using etherconfig to Edit Media Settings on Ethernet Interfaces

The `etherconfig` command can be used to set the duplex settings (full/half) as well as the speed (10/100/1000 Mbps) of ethernet interfaces. By default, interfaces automatically select the media settings; however, in some cases you may wish to override this setting.

Note If you have completed the GUI's System Setup Wizard (or the Command Line Interface `systemsetup` command) as described in the "Setup and Installation" chapter of the *Cisco IronPort AsyncOS for Email Configuration Guide* and committed the changes, the default ethernet interface settings should already be configured on your appliance.

Note Some Cisco IronPort C3x, C6x, and X10x appliances contain a fiber optic network interface option. If available, you will see two additional ethernet interfaces (Data 3 and Data 4) in the list of available interfaces on these appliances. These gigabit fiber optic interfaces can be paired with the copper (Data 1, Data 2, and Management) interfaces in a heterogeneous configuration. See Network Interface Card Pairing/Teaming, page 7-3.

Example of Editing Media Settings

```
mail3.example.com> etherconfig


Choose the operation you want to perform:

- MEDIA - View and edit ethernet media settings.

- PAIRING - View and configure NIC Pairing.

- VLAN - View and configure VLANs.

- LOOPBACK - View and configure Loopback.

[]> media


Ethernet interfaces:

1. Data 1 (Autoselect: <100baseTX full-duplex>) 00:06:5b:f3:ba:6d

2. Data 2 (Autoselect: <100baseTX full-duplex>) 00:06:5b:f3:ba:6e

3. Management (Autoselect: <100baseTX full-duplex>) 00:02:b3:c7:a2:da


Choose the operation you want to perform:

- EDIT - Edit an ethernet interface.

[]> edit


Enter the name or number of the ethernet interface you wish to edit.

[]> 2


Please choose the Ethernet media options for the Data 2 interface.

1. Autoselect

2. 10baseT/UTP half-duplex

3. 10baseT/UTP full-duplex

4. 100baseTX half-duplex

5. 100baseTX full-duplex

6. 1000baseTX half-duplex
```

```
7. 1000baseTX full-duplex

[1]> 5


Ethernet interfaces:

1. Data 1 (Autoselect: <100baseTX full-duplex>) 00:06:5b:f3:ba:6d

2. Data 2 (100baseTX full-duplex: <100baseTX full-duplex>) 00:06:5b:f3:ba:6e

3. Management (Autoselect: <100baseTX full-duplex>) 00:02:b3:c7:a2:da


Choose the operation you want to perform:

- EDIT - Edit an ethernet interface.

[]>


Choose the operation you want to perform:

- MEDIA - View and edit ethernet media settings.

- PAIRING - View and configure NIC Pairing.

- VLAN - View and configure VLANs.

- LOOPBACK - View and configure Loopback.

[]>
```

Network Interface Card Pairing/Teaming

NIC pairing allows you to combine any two physical data ports in order to provide a backup Ethernet interface if the data path from the NIC to the upstream Ethernet port should fail. Basically, pairing configures the Ethernet interfaces so that there is a primary interface and a backup interface. If the primary interface fails (i.e. if the carrier between the NIC and the upstream node is disrupted), the backup interface becomes active and an alert is sent. Within Cisco IronPort documentation, NIC pairing is synonymous with NIC teaming.

You can create more than one NIC pair, providing you have enough data ports. When creating pairs, you can combine any two data ports. For example:

- Data 1 and Data 2
- Data 3 and Data 4
- Data 2 and Data 3
- etc.

NIC pairing is not available on C1x appliances or M-Series appliances. Some C3x, C6x, and X10x appliances contain a fiber optic network interface option. If available, you will see two additional ethernet interfaces (Data 3 and Data 4) in the list of available interfaces on these appliances. These gigabit fiber optic interfaces can be paired with the copper (Data 1, Data 2, and Management) interfaces in a heterogeneous configuration.

NIC Pairing and VLANs

VLANs (see Virtual Local Area Networks (VLANs), page 7-8) are only allowed on the primary interface.

NIC Pair Naming

When creating NIC pairs, you must specify a name to use to refer to the pair. NIC pairs created in versions of AsyncOS prior to version 4.5 will automatically receive the default name of 'Pair 1' following an upgrade.

Any alerts generated regarding NIC pairing will reference the specific NIC pair by name.

Configuring and Testing NIC Pairing/Teaming

Once you have verified your ethernet media setting, use the `etherconfig` command to configure NIC pairing. You will be prompted for a name to use to refer to the pair.

The `failover` sub-command switches the active interface. The system will not automatically switch back to the primary NIC when it comes back on line and the backup interface will remain active in that case until you explicitly switch the system back over to the primary NIC (by using the `failover` command) or unless the backup NIC has a failure. See Using the failover Subcommand for NIC Pairing, page 7-6.

Use the `delete` subcommand to remove NIC pairs.

When configuring NIC pairing, keep in mind that all configuration changes require a commit, except for `failover`. The `failover` command force a failover during the next polling interval which is every 15 seconds once NIC pairing configuration has been committed.

NIC Pairing and Existing Listeners

If you enable NIC pairing on an interface that has listeners assigned to it, you are prompted to either delete, reassign, or disable all listeners assigned to the backup interface.

Enabling NIC Pairing via the etherconfig Command

```
mail3.example.com> etherconfig


Choose the operation you want to perform:

- MEDIA - View and edit ethernet media settings.

- PAIRING - View and configure NIC Pairing.

- VLAN - View and configure VLANs.

- LOOPBACK - View and configure Loopback.

[]> pairing


Paired interfaces:


Choose the operation you want to perform:

- NEW - Create a new pairing.

[]> new


Please enter a name for this pair (Ex: "Pair 1"):

[]> Pair 1


Warning: The backup (Data 2) for the NIC Pair is currently configured with one or more
IP addresses.  If you continue, the Data 2 interface will be deleted.

Do you want to continue? [N]> y


The interface you are deleting is currently used by listener "OutgoingMail".

What would you like to do?

1. Delete: Remove the listener and all its settings.

2. Change: Choose a new interface.

3. Ignore: Leave the listener configured for interface "Data 2" (the listener will be
disabled until you add a new interface named "Data 2" or edit the listener's settings).

[1]>
```

```
Injector OutgoingMail deleted for mail3.example.com.

Interface Data 2 deleted.

Paired interfaces:

1. Pair 1:

        Primary (Data 1) Active, Link is up

        Backup (Data 2) Standby, Link is up


Choose the operation you want to perform:

- FAILOVER - Manually failover to other port.

- DELETE - Delete a pairing.

- STATUS - Refresh status.

[]>


mail3.example.com> commit
```

Note Be sure to test the NIC pair now that you have created it. See Verifying NIC Pairing, page 7-8 for more information.

Using the failover Subcommand for NIC Pairing

In this example, a manual failover is issued, forcing the Data 2 interface to become the primary interface. Note that you must issue the `status` sub-command to see the change in the CLI:

```
mail3.example.com> etherconfig


Choose the operation you want to perform:

- MEDIA - View and edit ethernet media settings.

- PAIRING - View and configure NIC Pairing.

- VLAN - View and configure VLANs.

- LOOPBACK - View and configure Loopback.
```

```
[]> pairing


Paired interfaces:

1. Pair 1:

        Primary (Data 1) Active, Link is up

        Backup (Data 2) Standby, Link is up


Choose the operation you want to perform:

- FAILOVER - Manually failover to other port.

- DELETE - Delete a pairing.

- STATUS - Refresh status.

[]> failover


Paired interfaces:

1. Pair 1:

        Primary (Data 1) Active, Link is up

        Backup (Data 2) Standby, Link is up


Choose the operation you want to perform:

- FAILOVER - Manually failover to other port.

- DELETE - Delete a pairing.

- STATUS - Refresh status.

[]> status


Paired interfaces:

1. Pair 1:

        Primary (Data 1) Standby, Link is up

        Backup (Data 2) Active, Link is up
```

```
Choose the operation you want to perform:

- FAILOVER - Manually failover to other port.

- DELETE - Delete a pairing.

- STATUS - Refresh status.

[]>


Choose the operation you want to perform:

- MEDIA - View and edit ethernet media settings.

- PAIRING - View and configure NIC Pairing.

- VLAN - View and configure VLANs.

- LOOPBACK - View and configure Loopback.

[]>
```

Verifying NIC Pairing

You should verify that NIC pairing is working correctly. To do so:

Step 1 Use the `ping` command in the CLI to test your paired interface by "pinging" an IP address on the same subnet as the NIC pair that has been confirmed to return a ping by an independent source.:

```
mail3.example.com> ping x.x.x.x
```

Step 2 Issue a failover command (`etherconfig -> pairing -> failover`). Wait 15 seconds.

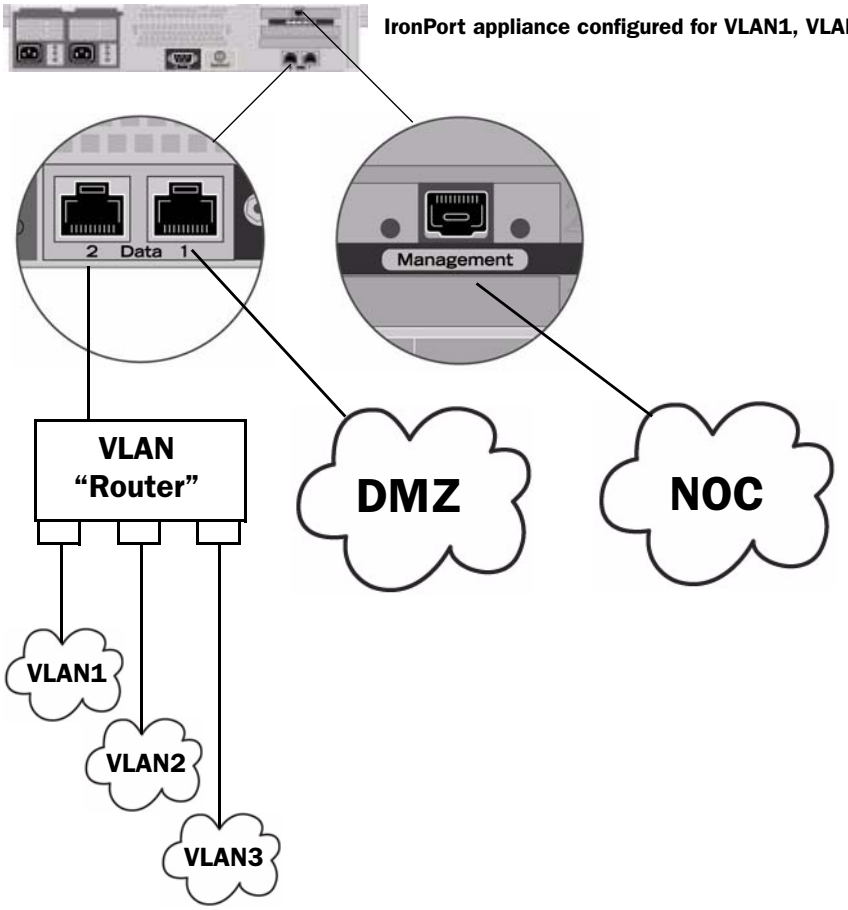
Step 3 Use the `ping` command in the CLI to test your paired interface again with the backup NIC as the active interface.

Step 4 Finally, return the NIC pair to its default state by issuing one more failover so that the primary interface is now active.

Virtual Local Area Networks (VLANs)

VLANs are virtual local area networks bound to physical data ports. You can configure VLANs to increase the number of networks the Cisco IronPort appliance can connect to beyond the number of physical interfaces included. For example, a Cisco IronPort C6x appliance has three interfaces: Data 1, Data 2, and Management. VLANs allow more networks to be defined on separate "ports" on existing listeners. (See Appendix B, "Accessing the Appliance" for more information.) You can configure multiple VLANs on any physical network port. Figure 7-1 provides an example of configuring several VLANs on the Data 2 interface.

Figure 7-1 Using VLANs to increase the number of networks available on the appliance



VLANs can be used to segment networks for security purposes, to ease administration, or increase bandwidth. VLANs appear as dynamic "Data Ports" labeled in the format of: "VLAN DDDD" where the "DDDD" is the ID and is an integer up to 4 digits long (VLAN 2, or VLAN 4094 for example). AsyncOS supports up to 30 VLANs. Duplicate VLAN IDs are not allowed on an Cisco IronPort appliance.

VLANs and Physical Ports

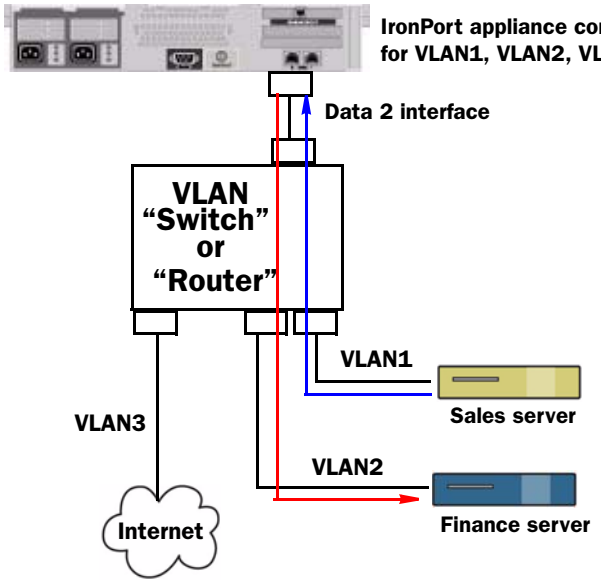
A physical port does not need an IP address configured in order to be in a VLAN. The physical port on which a VLAN is created can have an IP that will receive non-VLAN traffic, so you can have both VLAN and non-VLAN traffic on the same interface.

VLANs can be created on all "Data" and "Management" ports, including fiber optic data ports available on some Cisco IronPort X10x, C3x, and C6x appliances.

VLANs can be used with NIC pairing (available on paired NICs) and with Direct Server Return (DSR).

Figure 7-2 illustrates a use case showing how two mail servers unable to communicate directly due to VLAN limitations can send mail through the Cisco IronPort appliance. The blue line shows mail coming from the sales network (VLAN1) to the appliance. The appliance will process the mail as normal and then, upon delivery, tag the packets with the destination VLAN information (red line).

Figure 7-2 Using VLANs to Facilitate Communication Between Appliances



Managing VLANs

You can create, edit and delete VLANs via the `etherconfig` command. Once created, a VLAN can be configured via the Network -> Interfaces page or the `interfaceconfig` command in the CLI. Remember to commit all changes.

Creating a New VLAN via the etherconfig Command

In this example, two VLANs are created (named VLAN 31 and VLAN 34) on the Data 1 port:

```
mail3.example.com> etherconfig

Choose the operation you want to perform:

- MEDIA - View and edit ethernet media settings.

- PAIRING - View and configure NIC Pairing.

- VLAN - View and configure VLANs.

- LOOPBACK - View and configure Loopback.

[]> vlan


VLAN interfaces:


Choose the operation you want to perform:
```

```
- NEW - Create a new VLAN.

[]> new


VLAN ID for the interface (Ex: "34"):

[]> 34


Enter the name or number of the ethernet interface you wish bind to:

1. Data 1

2. Data 2

3. Management

[1]> 1


VLAN interfaces:

1. VLAN   34 (Data 1)


Choose the operation you want to perform:

- NEW - Create a new VLAN.

- EDIT - Edit a VLAN.

- DELETE - Delete a VLAN.

[]> new


VLAN ID for the interface (Ex: "34"):

[]> 31


Enter the name or number of the ethernet interface you wish bind to:

1. Data 1

2. Data 2

3. Management

[1]> 1
```

```
VLAN interfaces:

1. VLAN   31 (Data 1)

2. VLAN   34 (Data 1)


Choose the operation you want to perform:

- NEW - Create a new VLAN.

- EDIT - Edit a VLAN.

- DELETE - Delete a VLAN.

[]>


Choose the operation you want to perform:

- MEDIA - View and edit ethernet media settings.

- PAIRING - View and configure NIC Pairing.

- VLAN - View and configure VLANs.

- LOOPBACK - View and configure Loopback.

[]>
```

Creating an IP Interface on a VLAN via the interfaceconfig Command

In this example, a new IP interface is created on the VLAN 31 ethernet interface.

Note Making changes to an interface may close your connection to the appliance.

```
mail3.example.com> interfaceconfig


Currently configured interfaces:

1. Data 1 (10.10.1.10/24: example.com)

2. Management (10.10.0.10/24: example.com)


Choose the operation you want to perform:
```

```
- NEW - Create a new interface.

- EDIT - Modify an interface.

- GROUPS - Define interface groups.

- DELETE - Remove an interface.

[]> new


Please enter a name for this IP interface (Ex: "InternalNet"):

[]> InternalVLAN31


Would you like to configure an IPv4 address for this interface (y/n)? [Y]>


IPv4 Address (Ex: 10.10.10.10):

[]> 10.10.31.10


Netmask (Ex: "255.255.255.0" or "0xffffff00"):
[255.255.255.0]>


Would you like to configure an IPv6 address for this interface (y/n)? [N]>


Ethernet interface:

1. Data 1

2. Data 2

3. Management

4. VLAN   31

5. VLAN   34

[1]> 4


Hostname:

[]> mail31.example.com
```

```
Do you want to enable Telnet on this interface? [N]>

Do you want to enable SSH on this interface? [N]>

Do you want to enable FTP on this interface? [N]>

Do you want to enable HTTP on this interface? [N]>

Do you want to enable HTTPS on this interface? [N]>



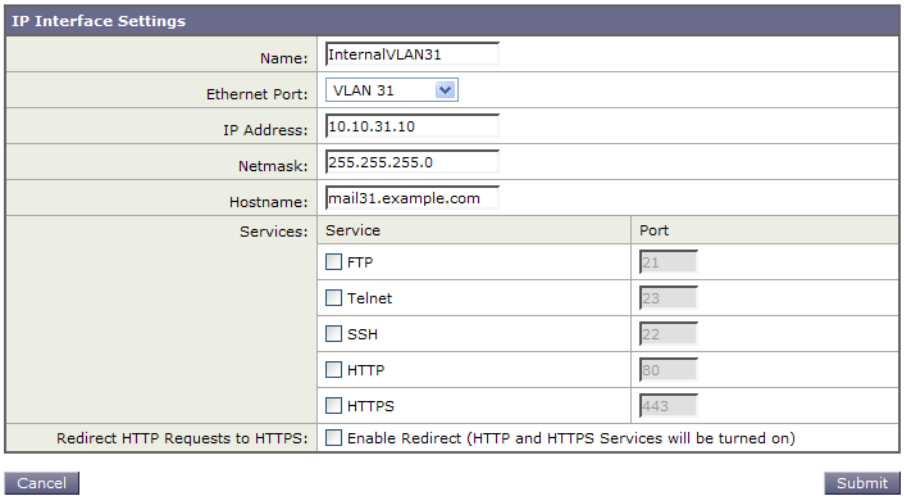
Currently configured interfaces:
1. Data 1 (10.10.1.10/24: example.com)
2. InternalVLAN31 (10.10.31.10/24: mail31.example.com)
3. Management (10.10.0.10/24: example.com)

Choose the operation you want to perform:
- NEW - Create a new interface.
- EDIT - Modify an interface.
- GROUPS - Define interface groups.
- DELETE - Remove an interface.
[]>

mail3.example.com> commit
```

You can also configure VLANs via the Network -> Listeners page:

Figure 7-3 Using a VLAN when Creating a New IP Interface via the GUI



Direct Server Return

Direct Server Return (DSR) is a way of providing support for a light-weight load balancing mechanism to load balance between multiple Cisco IronPort appliances sharing the same Virtual IP (VIP).

DSR is implemented via an IP interface created on the “loopback” ethernet interface on the Cisco IronPort appliance.

Note Configuring load balancing for Cisco IronPort appliances is beyond the scope of this document.

Enabling Direct Server Return

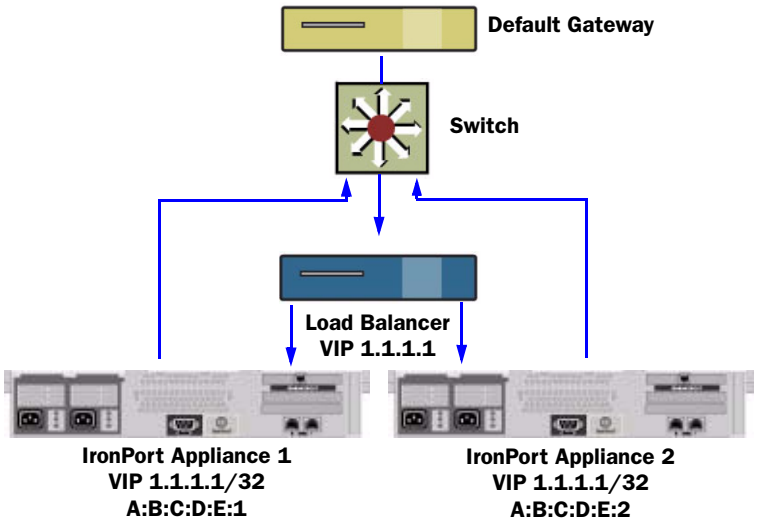
Enable DSR by enabling the “loopback” ethernet interface on each participating appliance. Next, create an IP interface on the loopback interface with a virtual IP (VIP) via the `interfaceconfig` command in the CLI or via the Network -> Interfaces page in the GUI. Finally, create a listener on the new IP interface via the `listenerconfig` command in the CLI or via the Network -> Listeners page in the GUI. Remember to commit all changes.

Note Using the loopback interface prevents the appliance from issuing ARP replies for that specific interface.

When enabling DSR, the following rules apply:

- All systems use the same Virtual IP (VIP) address
- All systems must be on the same switch and subnet as the load balancer

Figure 7-4 Using DSR to Load Balance Between Multiple Cisco IronPort Appliances on a Switch



Enabling the Loopback Interface via the etherconfig Command

Once enabled, the loopback interface is treated like any other interface (e.g. Data 1):

```
mail3.example.com> etherconfig


Choose the operation you want to perform:

- MEDIA - View and edit ethernet media settings.

- PAIRING - View and configure NIC Pairing.

- VLAN - View and configure VLANs.

- LOOPBACK - View and configure Loopback.

[]> loopback


Currently configured loopback interface:


Choose the operation you want to perform:

- ENABLE - Enable Loopback Interface.

[]> enable


Currently configured loopback interface:
```

```
1. Loopback


Choose the operation you want to perform:

- DISABLE - Disable Loopback Interface.

[]>


Choose the operation you want to perform:

- MEDIA - View and edit ethernet media settings.

- PAIRING - View and configure NIC Pairing.

- VLAN - View and configure VLANs.

- LOOPBACK - View and configure Loopback.

[]>
```

Creating an IP Interface on Loopback via the interfaceconfig Command

Create an IP interface on the loopback interface:

```
mail3.example.com> interfaceconfig


Currently configured interfaces:

1. Data 1 (10.10.1.10/24: example.com)

2. InternalV1 (10.10.31.10/24: mail31.example.com)

3. Management (10.10.0.10/24: example.com)


Choose the operation you want to perform:

- NEW - Create a new interface.

- EDIT - Modify an interface.

- GROUPS - Define interface groups.

- DELETE - Remove an interface.

[]> new
```

```
Please enter a name for this IP interface (Ex: "InternalNet"):

[]> LoopVIP


Would you like to configure an IPv4 address for this interface (y/n)? [Y]>


IPv4 Address (Ex: 10.10.10.10):

[]> 10.10.1.11


Netmask (Ex: "255.255.255.0" or "0xffffff00"):

[255.255.255.0]> 255.255.255.255


Would you like to configure an IPv6 address for this interface (y/n)? [N]>


Ethernet interface:

1. Data 1

2. Data 2

3. Loopback

4. Management

5. VLAN   31

6. VLAN   34

[1]> 3


Hostname:

[]> example.com


Do you want to enable Telnet on this interface? [N]>


Do you want to enable SSH on this interface? [N]>
```

```
Do you want to enable FTP on this interface? [N]>


Do you want to enable HTTP on this interface? [N]>


Do you want to enable HTTPS on this interface? [N]>


Currently configured interfaces:

1. Data 1 (10.10.1.10/24: example.com)

2. InternalV1 (10.10.31.10/24: mail31.example.com)

3. LoopVIP (10.10.1.11/24: example.com)

4. Management (10.10.0.10/24: example.com)


Choose the operation you want to perform:

- NEW - Create a new interface.

- EDIT - Modify an interface.

- GROUPS - Define interface groups.

- DELETE - Remove an interface.

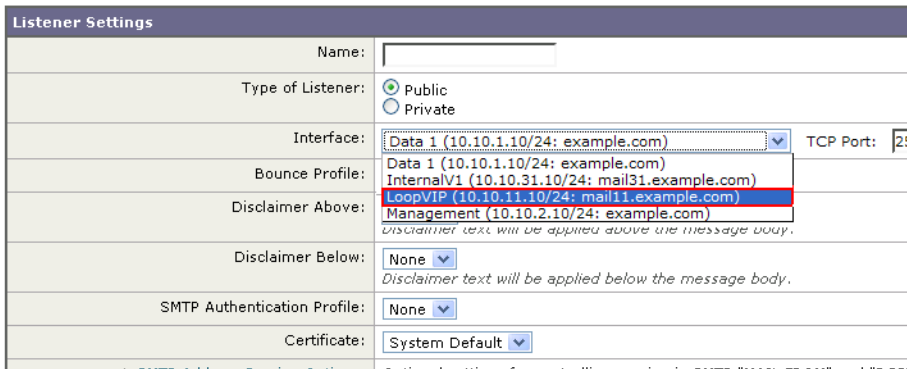
[]>


mail3.example.com> commit
```

Creating a Listener on the New IP Interface

Create a listener on the new IP interface via the GUI or the CLI. For example, Figure 7-5 shows the newly created IP interface available in the Add Listener page in the GUI.

Figure 7-5 Creating a Listener on the New Loopback IP Interface





CHAPTER 8

Centralized Management

The Cisco IronPort centralized management feature (available via feature key) allows you to manage and configure multiple appliances at the same time, reducing administration time and ensuring a consistent configuration across your network. You do not need to purchase additional hardware for managing multiple appliances. The centralized management feature provides increased reliability, flexibility, and scalability within your network, allowing you to manage globally while complying with local policies.

A *cluster* is defined as a set of machines that share configuration information. Within the cluster, machines (Cisco IronPort appliances) are divided into *groups*; every cluster will contain at least one group. A given machine is a member of one and only one group. An administrator user can configure different elements of the system on a cluster-wide, group-wide, or per-machine basis, enabling the segmentation of Cisco IronPort appliances based on network, geography, business unit, or other logical relationships.

Clusters are implemented as a *peer-to-peer* architecture; there is no master/slave relationship within a cluster. You may log into any machine to control and administer the cluster. (Some configuration commands, however, are limited. See Restricted Commands, page 8-15.)

The user database is shared across all machines in the cluster. That is, there will be only one set of users and one administrator user (with the associated passwords) for an entire cluster. All machines that join a cluster will share a single administrator password which is referred to as the *admin password* of the cluster.

Topics discussed in this chapter include:

- Cluster Requirements, page 8-1
- Cluster Organization, page 8-2
- Creating and Joining a Cluster, page 8-4
- Managing Clusters, page 8-11
- Administering a Cluster from the GUI, page 8-16
- Cluster Communication, page 8-19
- Best Practices and Frequently Asked Questions, page 8-23

Cluster Requirements

- Machines in a cluster must have resolvable hostnames in DNS. Alternatively, you can use IP addresses instead, but you may not mix the two.

  See DNS and Hostname Resolution, page 8-19. Cluster communication is normally initiated using the DNS hostnames of the machines.

- A cluster must consist entirely of machines in the same series (X-Series and C-Series are compatible).

  For example, Cisco IronPort X1000, C60, C600, C30, C300, and C10 appliances can be in the same cluster; however, C60 and A60 appliances cannot be in the same cluster. If you attempt to add an incompatible appliance to an existing cluster, an error message explaining why that appliance cannot be added to the cluster will be displayed.

- A cluster must consist entirely of machines running the same version of AsyncOS.

  See Upgrading Machines in a Cluster, page 8-13 for how to upgrade members of a cluster.

- Machines can either join the cluster via SSH (typically on port 22) *or* via the Cluster Communication Service (CCS).

  See Cluster Communication, page 8-19.

- Once machines have joined the cluster, they can communicate via SSH or via Cluster Communication Service. The port used in configurable. SSH is typically enabled on port 22, and by default CCS is on port 2222, but you can configure either of these services on a different port.

  In addition to the normal firewall ports that must be opened for the appliance, clustered machines communicating via CCS must be able to connect with each other via the CCS port. See Cluster Communication, page 8-19.

- You must use the Command Line Interface (CLI) command `clusterconfig` to create, join, or configure clusters of machines.

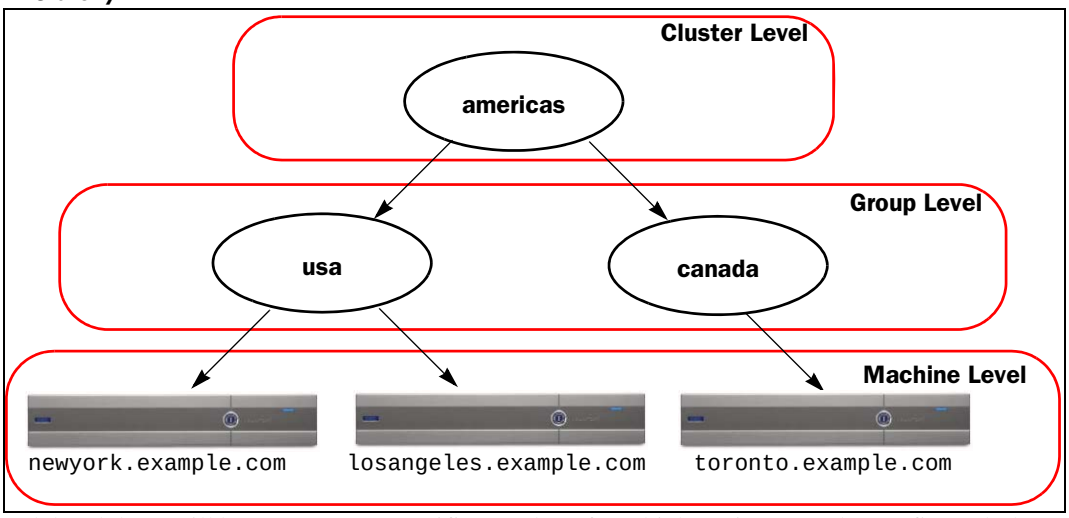
  Once you have created a cluster, you can manage non-cluster configuration settings from either the GUI or the CLI.

  See Creating and Joining a Cluster, page 8-4 and Administering a Cluster from the GUI, page 8-16.

Cluster Organization

Within a cluster, configuration information is divided into 3 groupings or *levels*. The top level describes cluster settings; the middle level describes group settings; and the lowest level describes machine-specific settings.

Figure 8-1 Cluster Level Hierarchy



Within each level there will be one or more specific members for which settings may be configured; these are referred to as *modes*. A mode refers to a named member at a specified level. For example, the group "usa" represents one of two group modes in the diagram. While levels are a general term, modes are specific; modes are always referred to by name. The cluster depicted in Figure 8-1 has six modes.

Although settings are configured at a given level, they are always configured **for** a specific mode. It is not necessary to configure settings for all modes within a level. The cluster mode is a special case. Because there can only be one cluster, all settings configured for the cluster mode can be said to be configured at the cluster level.

You should normally configure most settings at the cluster level. However, settings that have been specifically configured at lower levels will *override* settings configured at higher levels. Thus, you can override cluster-mode settings with group-mode or machine-mode settings.

For example, you might start by configuring the Good Neighbor Table in cluster mode; all machines in the cluster would use that configuration. Then, you might also configure this table in machine mode for machine `newyork`. In this case, all other machines in the cluster will still use the good neighbor table defined at the cluster level, but the machine `newyork` will override the cluster settings with its individual machine mode settings.

The ability to override cluster settings for specific groups or machines gives you a lot of flexibility. However, if you find yourself configuring many settings individually in machine mode, you will lose much of the ease of administration that clusters were intended to provide.

Initial Configuration Settings

For most features, when you begin to configure settings for a new mode, those settings will initially be empty by default. There is a distinction between empty settings and having no settings in a mode. As an example, consider a very simple cluster composed of one group and one machine. Imagine that you have an LDAP query configured at the cluster level. There are no settings configured at the group or machine levels:

`Cluster`	`(ldap queries: a, b, c)`
`Group`	
`Machine`	

Now, imagine that you create new LDAP query settings for the group. The result will be something like this:

Cluster	(ldap queries: a, b, c)
Group	(ldap queries: None)
Machine	

The group-level settings now override the cluster-level setting; however, the new group settings are initially empty. The group mode does not actually have any LDAP queries of its own configured. Note that a machine within this group will inherit this "empty" set of LDAP queries from the group.

Next, you can add an LDAP query to the group, for example:

Cluster	(ldap queries: a, b, c)
Group	(ldap queries: d)
Machine	

Now the cluster level has one set of queries configured while the group has another set of queries. The machine will inherit its queries from the group.

Creating and Joining a Cluster

You cannot create or join a cluster from the Graphical User Interface (GUI). You must use the Command Line Interface (CLI) to create, join, or configure clusters of machines. Once you have created a cluster, you can change configuration settings from either the GUI or the CLI.

Be sure to enable your centralized management feature key *before* you attempt to create a cluster.

Note Your Cisco IronPort appliance does not ship with an evaluation key for the centralized management feature. You must request a 30-day evaluation, or purchase a key, before you can enable the centralized management feature. Use the `featurekey` command in the CLI or the System Administration > Feature Keys page to enable your key.

The clusterconfig Command

A machine can create or join a cluster only via the `clusterconfig` command.

- When a new cluster is *created*, all of that cluster's initial settings will be inherited from the machine that creates the cluster. If a machine was previously configured in "standalone" mode, its standalone settings are used when creating the cluster.
- When a machine *joins* a cluster, all of that machine's clusterable settings will be inherited from the cluster level. In other words, everything except certain machine-specific settings (IP addresses, etc) will be lost and will be replaced with the settings from the cluster and/or the group selected for that machine to join. If a machine was previously configured in "standalone" mode, its standalone settings are used when creating the cluster, and no settings at the machine level are maintained.

If the current machine is not already part of a cluster, issuing the `clusterconfig` command presents the option to join an existing cluster or create a new one.

```
newyork.example.com> clusterconfig


Do you want to join or create a cluster?

1. No, configure as standalone.

2. Create a new cluster.

3. Join an existing cluster over SSH.

4. Join an existing cluster over CCS.

[1]> 2


Enter the name of the new cluster.

[]> americas


New cluster committed: Wed Jun 22 10:02:04 2005 PDT

Creating a cluster takes effect immediately, there is no need to commit.


Cluster americas


Choose the operation you want to perform:

- ADDGROUP - Add a cluster group.

- SETGROUP - Set the group that machines are a member of.

- RENAMEGROUP - Rename a cluster group.

- DELETEGROUP - Remove a cluster group.

- REMOVEMACHINE - Remove a machine from the cluster.

- SETNAME - Set the cluster name.

- LIST - List the machines in the cluster.

- LISTDETAIL - List the machines in the cluster with detail.

- DISCONNECT - Temporarily detach machines from the cluster.
```

```
- RECONNECT - Restore connections with machines that were previously detached.

- PREPJOIN - Prepare the addition of a new machine over CCS.

[]>
```

At this point you can add machines to the new cluster. Those machines can communicate via SSH or CCS.

Joining an Existing Cluster

From the host you want to add to the cluster, issue the `clusterconfig` command to join the existing cluster. You can choose to join the cluster over SSH or over CCS (cluster communication service).

In order to join a host to an existing cluster, you must:

- be able to validate the SSH host key of a machine in the cluster
- know the IP address of a machine in the cluster and be able to connect to this machine in the cluster (for example, via SSH or CCS)
- know the administrator password for the admin user on a machine belonging to the cluster

Note All machines you intend to add to the cluster must have the centralized management feature key installed on them before they can be added to the cluster. It is also possible to join an existing cluster within the `systemsetup` command if the feature key for centralized management has been installed on the system prior to running the CLI system setup wizard and if a cluster exists. After changing the administrator password, setting the hostname of the appliance, and configuring network interfaces and IP addresses, the systemsetup will prompt you to create or join a cluster.

Joining an Existing Cluster over SSH

The following table demonstrates adding the machine `losangeles.example.com` to the cluster using the SSH option.

```
losangeles.example.com> clusterconfig


Do you want to join or create a cluster?

1. No, configure as standalone.

2. Create a new cluster.

3. Join an existing cluster over SSH.

4. Join an existing cluster over CCS.

[1]> 3
```

```
While joining a cluster, you will need to validate the SSH host key of the remote
machine to which you are joining.  To get the public host key

fingerprint of the remote host, connect to the cluster and run: logconfig ->
hostkeyconfig -> fingerprint.



WARNING: All non-network settings will be lost. System will inherit the values set at
the group or cluster mode for the non-network settings.  Ensure that the cluster
settings are compatible with your network settings (e.g. dnsconfig settings)



Do you want to enable the Cluster Communication Service on

losangeles.example.com? [N]> n



Enter the IP address of a machine in the cluster.

[]> IP address is entered



Enter the remote port to connect to. The must be the normal admin ssh

port, not the CCS port.

[22]> 22



Enter the admin password for the cluster.

The administrator password for the clustered machine is entered



Please verify the SSH host key for IP address:

Public host key fingerprint: xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx

Is this a valid key for this host? [Y]> y



Joining cluster group Main_Group.

Joining a cluster takes effect immediately, there is no need to commit.

Cluster americas



Choose the operation you want to perform:
```

```
- ADDGROUP - Add a cluster group.

- SETGROUP - Set the group that machines are a member of.

- RENAMEGROUP - Rename a cluster group.

- DELETEGROUP - Remove a cluster group.

- REMOVEMACHINE - Remove a machine from the cluster.

- SETNAME - Set the cluster name.

- LIST - List the machines in the cluster.

- LISTDETAIL - List the machines in the cluster with detail.

- DISCONNECT - Temporarily detach machines from the cluster.

- RECONNECT - Restore connections with machines that were previously detached.

- PREPJOIN - Prepare the addition of a new machine over CCS.

[]>

(Cluster americas)>
```

Joining an Existing Cluster over CCS

Use CCS instead of SSH if you cannot use SSH. The only advantage of CCS is that only cluster communication happens over that port (no user logins, SCP, etc). To add another machine to an existing cluster via CCS, use the `prepjoin` subcommand of `clusterconfig` to prepare the machine to be added to the cluster. In this example, the `prepjoin` command is issued on the machine `newyork` to prepare the machine `losangeles` to be added to the cluster.

The `prepjoin` command involves obtaining the user key of the host you want to add to the cluster by typing `clusterconfig prepjoin print` in the CLI of that host, and then copying the key into the command line of the host that is currently in the cluster.

```
Choose the operation you want to perform:

- ADDGROUP - Add a cluster group.

- SETGROUP - Set the group that machines are a member of.

- RENAMEGROUP - Rename a cluster group.

- DELETEGROUP - Remove a cluster group.

- REMOVEMACHINE - Remove a machine from the cluster.

- SETNAME - Set the cluster name.

- LIST - List the machines in the cluster.

- LISTDETAIL - List the machines in the cluster with detail.
```

```
- DISCONNECT - Temporarily detach machines from the cluster.

- RECONNECT - Restore connections with machines that were previously detached.

- PREPJOIN - Prepare the addition of a new machine over CCS.

[]> prepjoin


Prepare Cluster Join Over CCS


No host entries waiting to be added to the cluster.


Choose the operation you want to perform:

- NEW - Add a new host that will join the cluster.

[]> new


Enter the hostname of the system you want to add.

[]> losangeles.example.com


Enter the serial number of the host mail3.example.com.

[]> unique serial number is added


Enter the user key of the host losangeles.example.com. This can be obtained by typing
"clusterconfig prepjoin print" in the CLI on mail3.example.com. Press enter on a blank
line to finish.

unique user key from output of prepjoin print is pasted


Host losangeles.example.com added.

Prepare Cluster Join Over CCS


1. losangeles.example.com (serial-number)


Choose the operation you want to perform:
```

```
- NEW - Add a new host that will join the cluster.

- DELETE - Remove a host from the pending join list.

[]>


(Cluster americas)> commit
```

Once a machine is already part of a cluster, the `clusterconfig` command allows you to configure various settings for the cluster.

```
(Cluster Americas)> clusterconfig


Cluster americas


Choose the operation you want to perform:

- ADDGROUP - Add a cluster group.

- SETGROUP - Set the group that machines are a member of.

- RENAMEGROUP - Rename a cluster group.

- DELETEGROUP - Remove a cluster group.

- REMOVEMACHINE - Remove a machine from the cluster.

- SETNAME - Set the cluster name.

- LIST - List the machines in the cluster.

- LISTDETAIL - List the machines in the cluster with detail.

- DISCONNECT - Temporarily detach machines from the cluster.

- RECONNECT - Restore connections with machines that were previously detached.

- PREPJOIN - Prepare the addition of a new machine over CCS.

[]>
```

Adding Groups

All clusters must contain at least one group. When you create a new cluster, a default group called Main_Group is created automatically. However, you may decide to create additional groups within your cluster. This example shows how to create additional groups within an existing cluster and assign machines to the new group(s).

Step 1 Issue the `clusterconfig` command.

Step 2 Choose the `addgroup` subcommand and enter the name of the new group.

Step 3 Use the `setgroup` subcommand to choose machines for the new group.

Managing Clusters

Administering a Cluster from the CLI

For machines that are part of a cluster, the CLI can be switched into different *modes*. Recall that a mode refers to a specific, named, member of a level.

The CLI mode determines precisely where a configuration setting will be modified. The default is "machine" mode for the machine the user logged into, the "login host."

Use the `clustermode` command to switch between different modes. For example:

Table 8-1 Administering Clusters

Command Example	Description
`clustermode`	Prompt to switch cluster mode
`clustermode group northamerica`	Switch to group mode for the group "northamerica"
`clustermode machine losangeles.example.com`	Switch to machine mode for the machine "losangeles"

The prompt in the CLI changes to indicate your current mode, e.g.

```
(Cluster Americas)>
```

or

```
(Machine losangeles.example.com)>
```

In machine mode, the prompt will include the fully qualified domain name of the machine.

Copying and Moving Settings

All non-restricted (see Restricted Commands, page 8-15) commands have new operations: `CLUSTERSHOW` and `CLUSTERSET`. `CLUSTERSHOW` is used to show in which modes a command is configured (see New Operation Added, page 8-15). The `CLUSTERSET` operation allows you to move or copy the current settings (configurable with the current command) from one mode to another or between levels (e.g. from a machine to a group).

A *copy* retains the settings for the current mode. A *move* resets (clears) the configuration of the current mode; i.e., following a move, no settings will be configured for the current mode.

For example, if you have configured Good Neighbor Table settings (the `destconfig` command) for group `northamerica`, and you decide that you want the entire cluster to have these settings, you can use the `clusterset` operation from within the `destconfig` command to copy (or move) the current settings to the cluster mode. (See Experimenting with New Configurations, page 8-12.)

Warning **Exercise caution when moving or copying configuration settings to avoid inconsistent dependencies. For example, if you move or copy listeners with disclaimer stamping configured to another machine, and that new machine does not have the same disclaimers configured, disclaimer stamping will not be enabled on the new machine.**

Experimenting with New Configurations

One of the most advantageous ways to use clusters is to experiment with new configuration settings. First you make changes at the machine mode, in an isolated environment. Then, when you are satisfied with your configuration, you move those configuration changes up to the cluster mode to make them available on all machines.

The following example shows the steps to change a listener setting on one machine and then publish the setting to the rest of the cluster when ready. Because listeners are normally configured at the cluster level, the example starts by pulling the configuration down to machine mode on one machine before making and testing the changes. You should test experimental changes of this type on one machine before making the change to the other machines in the cluster.

Step 1 Use the `clustermode cluster` command to change to the cluster mode.

Remember: the `clustermode` command is the CLI command you use to change modes to the cluster, group, and machine levels.

Step 2 Type `listenerconfig` to see the listener settings configured for the cluster.

Step 3 Choose the machine you want to experiment with, then use the `clusterset` command to copy settings from the cluster “down” to machine mode.

Step 4 Use the `clustermode` command to navigate to machine mode for the experimental machine, e.g.:

```
clustermode machine newyork.example.com
```

Step 5 In machine mode, on the experimental machine, issue the `listenerconfig` command to make changes specifically for the experimental machine.

Step 6 Commit the changes.

Step 7 Continue to experiment with the configuration changes on the experimental machine, remembering to commit the changes.

Step 8 When you are ready to apply your new settings to all the other machines, use the `clusterset` command to move the settings up to the cluster mode.

Step 9 Commit the changes.

Leaving a Cluster Permanently (Removal)

You use the `REMOVEMACHINE` operation of `clusterconfig` to remove a machine permanently from a cluster. When a machine is permanently removed from a cluster, its configuration is "flattened" such that it will work the same as it did when it was part of the cluster. For example, if there is only a cluster-mode Global Unsubscribe table, the Global Unsubscribe table data will be copied to the machine's local configuration when the machine is removed from the cluster.

Upgrading Machines in a Cluster

A cluster does not allow the connected machines to have different versions of AsyncOS. Before you perform an upgrade of AsyncOS, you need to disconnect each machine in the cluster will via the `clusterconfig` command. After you upgrade all the machines, the cluster can be reconnected via the `clusterconfig` command. You can have two separate clusters running while you upgrade machines to the same version. You can also upgrade clustered machines on the GUI Upgrades page.

Note If you use the upgrade command before disconnecting the individual machine from the cluster, AsyncOS disconnects all the machines in the cluster. Cisco Systems recommends that you disconnect each machine from the cluster before upgrading it. Then, other machines can continue working as a cluster until each is disconnected and upgraded.

To upgrade the machines in a cluster via the CLI:

Step 1 On a machine in the cluster, use the `disconnect` operation of `clusterconfig`. For example, to disconnect the machine `losangeles.example.com`, type `clusterconfig disconnect losangeles.example.com`. No `commit` is necessary.

Step 2 Optionally, use the `suspendlistener` command to halt acceptance of new connections and messages during the upgrade process.

Step 3 Issue the `upgrade` command to upgrade AsyncOS to a newer version.

Note Disregard any warnings or confirmation prompts about disconnecting all of the machines in the cluster. Because you have disconnected the machine, AsyncOS does not disconnect the other machines in the cluster at this point.

Step 4 Select the version of AsyncOS for the machine. The machine will reboot after the upgrade is complete.

Step 5 Use the `resume` command on the upgraded machine to begin accepting new messages.

Step 6 Repeat steps 1 - 5 for each machine in the cluster.

Note After you disconnect a machine from the cluster, you cannot use it to change the configurations of other machines. Although you can still modify the cluster configuration, do not change it while machines are disconnected because settings can become unsynchronized.

Step 7 After you have upgraded all the machines, use the reconnect operation of clusterconfig for each upgraded machine to reconnect it. For example, to reconnect the machine `losangeles.example.com`, type `clusterconfig reconnect losangeles.example.com`. Note that you can only connect a machine to a cluster that is running the same version of AsyncOS.

Configuration File Commands

Configuration information may be saved for any individual system in the cluster. If you are in machine mode and you export a configuration file (using the System Administration > Configuration File page or the `exportconfig` command), the file will be exported onto the local disk of the machine you are currently configuring. If you are in cluster mode or group mode, then the file will be saved on the machine you are currently logged into. You will be notified which machine the file was exported to.



Note Saving the configuration of an *entire* cluster (or a clustered machine) prior to restoring that configuration onto a set of machines (either the same machines or a different set) via the System Administration > Configuration File page or the `loadconfig` command is not supported.

Resetting the Configuration

If the configuration is reset (via the System Administration > Configuration File page or the `resetconfig` command) on a machine (restricted to local machine mode) that is part of a cluster, then that machine will return to the default factory settings. If that machine was previously part of a cluster, resetting the configuration will also automatically remove it from the cluster.

CLI Command Support

All Commands Are Cluster-aware

All CLI commands in AsyncOS are now cluster-aware. The behavior of some commands will change slightly when issued in a cluster mode. For example, the behavior of the following commands changes when issued on a machine that is part of a cluster:

The commit and clearchanges Commands

commit

The `commit` command commits all changes for all three levels of the cluster, regardless of which mode you are currently in.

commitdetail

The `commitdetail` command provides details about configuration changes as they are propagated to all machines within a cluster.

clearchanges

The `clearchanges` (`clear`) command clears all changes for all three levels of the cluster, regardless of which mode you are currently in.

New Operation Added

CLUSTERSHOW

Within each command, there is now a `CLUSTERSHOW` operation that allows you to see in which modes a command is configured.

When you enter a CLI command to perform an action that will be overridden by existing settings at a lower level, you will be presented with a notification. For example, if you are in cluster mode and enter a command, you may see a notification like this:

```
Note: Changes to these settings will not affect the following groups and machines
because they are overriding the cluster-wide settings:

East_Coast, West_Coast

facilities_A, facilities_B, receiving_A
```

A similar message would be printed if you are editing settings for a group mode.

Restricted Commands

Most CLI commands and their corresponding GUI pages can be run in any mode (cluster, group, or machine). However, some commands and pages are restricted to one mode only.

The system interface (either the GUI and the CLI) will always will make it clear that a command is restricted and how it is restricted. It is easy to switch to the appropriate mode for configuring the command.

- In the GUI, use the "Change Mode" menu or the "Settings for this features are currently defined at:" links to switch modes.
- In the CLI, use the `clustermode` command to switch modes.

The following commands are restricted to *cluster mode*:

Table 8-2 Commands Restricted to Cluster Mode

clusterconfig	**sshconfig**
clustercheck	**userconfig**
passwd	

If a you try to run one of these commands in group or machine mode, you will be given a warning message and the opportunity to switch to the appropriate mode.

Note The `passwd` command is a special case because it needs to be usable by guest users. If a guest user issues the `passwd` command on a machine in a cluster, it will not print the warning message but will instead just silently operate on the cluster level data without changing the user's mode. All other users will get the above written behavior (consistent with the other restricted configuration commands).

The following commands are restricted to *machine mode*:

antispamstatus	etherconfig	resume	suspenddel
antispamupdate	featurekey	resumedel	suspendlistener
antivirusstatus	hostrate	resumelistener	techsupport
antivirusupdate	hoststatus	rollovernow	tophosts
bouncerecipients	interfaceconfig	routeconfig	topin
deleterecipients	ldapflush	sbstatus	trace
delivernow	ldaptest	setgateway	version
diagnostic	nslookup	sethostname	vofflush
dnsflush	quarantineconfig	settime	vofstatus
dnslistflush	rate	shutdown	workqueue
dnslisttest	reboot	status	
dnsstatus	resetcounters	suspend	

If a you try to run one of the commands above in cluster or group mode, you will be given a warning message and the opportunity to switch to an appropriate mode.

The following commands are further restricted to the *login host* (i.e., the specific machine you are logged into). These commands require access to the local file system.

Table 8-3 Commands Restricted to Login Host Mode

last	resetconfig	tail	upgrade
ping	supportrequest	telnet	who

Administering a Cluster from the GUI

Although you cannot create or join clusters or administer cluster specific settings from the GUI (the equivalent of the `clusterconfig` command), you can browse machines in the cluster, create, delete, copy, and move settings among the cluster, groups, and machines (that is, perform the equivalent of the `clustermode` and `clusterset` commands) from within the GUI.

When you first log into the GUI, you are shown the Incoming Mail Overview page. Presuming that you have configured the current machine to be a member of a cluster, you are also notified that the centralized management feature has been enabled in the GUI.

The Incoming Mail Overview page is an example of a command that is restricted to the login host, because the Mail Flow Monitoring data you are viewing is stored on the local machine. To view the Incoming Mail Overview reports for another machine, you must log into the GUI for that machine.

Note the URL in the browser's address field when clustering has been enabled on an appliance. The URL will contain the word `machine`, `group`, or `cluster` as appropriate. For example, when you first log in, the URL of the Incoming Mail Overview page will appear as:

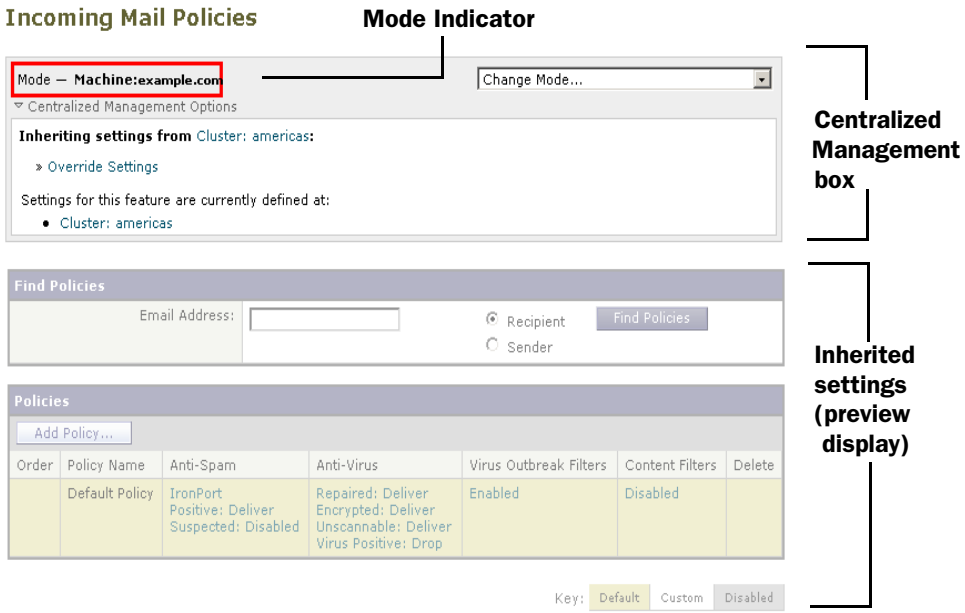
`https://`*hostname*`/`**machine/***serial_number*`/monitor/incoming_mail_overview`



Note The Incoming Mail Overview and Incoming Mail Details pages on the Monitor menu are restricted to the login machine.

The Mail Policies, Security Services, Network, and System Administration tabs contain pages that are not restricted to the local machine. If you click the Mail Policies tab, the centralized management information in the GUI changes.

Figure 8-2 Centralized Management Feature in the GUI: No Settings Defined

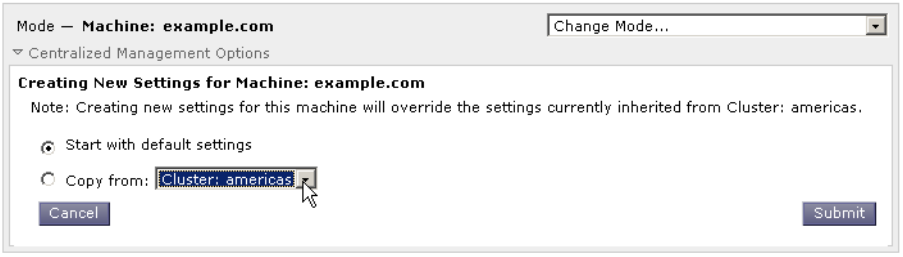


In Figure 8-2, the machine is inheriting all of its configuration settings for the current feature from the cluster mode. The settings being inherited in a light grey (preview).You can retain these settings or change them, overriding the cluster level settings for this machine.

Note The inherited settings (preview display) will always show the settings inherited from the cluster. Use caution when enabling or disabling dependent services among group and cluster levels. For more information, see Copying and Moving Settings, page 8-11.

If you click the Override Settings link, you are taken to a new page for that feature. This page allows you to create new configuration settings for machine mode. You may begin with the default settings, or, if you've already configured settings in another mode, you can copy those settings to this machine.

Figure 8-3 Centralized Management Feature in the GUI: Create New Settings



Alternatively, as shown in Figure 8-2, you can also navigate to modes where this configuration setting is already defined. The modes are listed in the lower half of the centralized management box, under "Settings for this feature are currently defined at:". Only those modes where the settings are actually defined will be listed here. When you view a page for settings that are defined in (and inherited from) another mode, the page will display those settings for you.

If you click on one of the listed modes (for example, the Cluster: Americas link as shown in Figure 8-2), you will be taken to a new page that allows you to view and manage the settings for that mode.

Figure 8-4 Centralized Management Feature in GUI: Settings Defined



When settings are defined for a given mode, the centralized management box is displayed on every page in a minimized state. Click the "Centralized Management Options" link to expand the box to show a list of options available for the current mode with respect to the current page. Clicking the "Manage Settings" button allows you to copy or move the current settings to a different mode or to delete those settings completely.

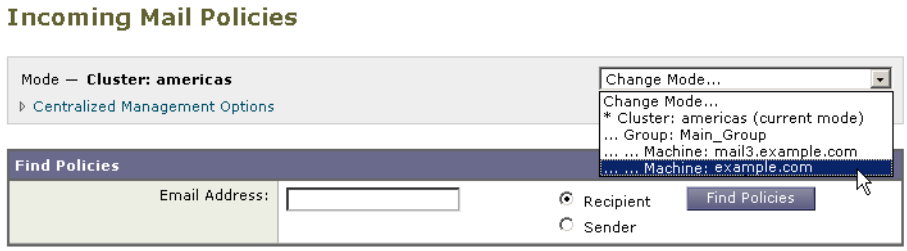
For example, in Figure 8-5, the Centralized Management Options link has been clicked to present the available options.

Figure 8-5 Centralized Management Feature in GUI: Manage Settings



On the right side of the box is the "Change Mode" menu. This menu displays your current mode and provides the ability to navigate to any other mode (cluster, group, or machine) at any time.

Figure 8-6 The Change Mode Menu



When you navigate to a page that represents a different mode, the "Mode —" text on the left side of the centralized management box will flash yellow, briefly, to alert you that your mode has changed.

Some pages within certain tabs are restricted to machine mode. However, unlike the Incoming Mail Overview page (which is restricted to the current login host), these pages can be used for any machine in the cluster.

Figure 8-7 Centralized Management Feature: Machine Restricted



Choose which machine to administer from the Change Mode menu. You will see a brief flashing of the text to remind you that you have changed modes.

Cluster Communication

Machines within a cluster communicate with each other using a *mesh network*. By default, all machines connect to all other machines. If one link goes down, other machines will not be prevented from receiving updates.

By default, all intra-cluster communication is secured with SSH. Each machine keeps an in-memory copy of the route table and makes in-memory changes as necessary if links go down or up. Each machine also performs a periodic "ping" (every 1 minute) of every other machine in the cluster. This ensures up-to-date link status and maintains the connections in case a router or NAT has a timeout.

Note The connection between two clustered appliances may be dropped if one of the appliances attempts to open more than the maximum number of SSH connections allowed. The appliances automatically rejoin the cluster within seconds and no manual configuration is needed.

DNS and Hostname Resolution

DNS is required to connect a machine to the cluster. Cluster communication is normally initiated using the DNS hostnames of the machines (not the hostname of an interface on the machine). A machine with an unresolvable hostname would be unable to actually communicate with any other machines in the cluster, even though it is technically part of the cluster.

Your DNS must be configured to have the hostname point to the correct IP interface on the appliance that has SSH or CCS enabled. This is very important. If DNS points to another IP address that does not have SSH or CCS enabled it will not find the host. Note that centralized management uses the "main hostname," as set with the `sethostname` command, not the per-interface hostname.

If you use an IP address to connect to another machine in the cluster, the machine you connect to must be able to make a reverse look up of the connecting IP address. If the reverse look up times out because the IP address isn't in the DNS, the machine cannot connect to the cluster.

Clustering, Fully Qualified Domain Names, and Upgrading

DNS changes can cause a loss of connectivity after upgrading AsyncOS. Please note that if you need to change the fully qualified domain name of a machine in the cluster (not the hostname of an interface on a machine in the cluster), you must change the hostname settings via `sethostname` and update the DNS record for that machine *prior* to upgrading AsyncOS.

Cluster Communication Security

Cluster Communication Security (CCS) is a secure shell service similar to a regular SSH service. Cisco implemented CCS in response to concerns regarding using regular SSH for cluster communication. SSH communication between two machines opens regular logins (admin, etc.) on the same port. Many administrators prefer not to open regular logins on their clustered machines.

Tip: never enable Cluster Communication Services, even though it is the default, unless you have firewalls blocking port 22 between some of your clustered machines. Clustering uses a full mesh of SSH tunnels (on port 22) between all machines. If you have already answered Yes to enabling CCS on any machine, remove all machines from the cluster and start again. Removing the last machine in the cluster removes the cluster.

CCS provides an enhancement where the administrator can open up cluster communication, but not CLI logins. By default, the service is disabled. If the centralized management feature is enabled on the appliance, then you will be prompted to enable CCS from the `interfaceconfig` command when you are prompted to enable other services. For example:

```
Do you want to enable SSH on this interface?  [Y]>


Which port do you want to use for SSH?

[22]>


Do you want to enable Cluster Communication Service on this interface?

[N]> y


Which port do you want to use for Cluster Communication Service?

[2222]>
```

The default port number for CCS is 2222. You may change this to another open, unused, port number if you prefer. After the join is complete and the joining machine has all the configuration data from the cluster, the following question is presented:

```
Do you want to enable Cluster Communication Service on this interface? [N]> y


Which port do you want to use for Cluster Communication Service?

[2222]>
```

Cluster Consistency

When centralized management is enabled, the machines that are "cluster aware" will continually verify network connections to other machines within the cluster. This verification is done by periodic "pings" sent to other machines in the cluster.

If all attempts to communicate with a particular machine fail, then the machine that has been trying to communicate will log a message saying that the remote host has disconnected. The system will send an alert to the administrator that the remote host went down.

Even if a machine is down, the verification pings will continue to be sent. When a machine rejoins the cluster network, a synchronization command will be issued so that any previously offline machines can download any updates. The synchronization command will also determine if there have been any changes on one side but not the other. If so, then the previously down machine will silently download the updates.

Disconnect/Reconnect

A machine may be disconnected from a cluster. Occasionally, you may intend to deliberately disconnect the machine, for example, because you are upgrading the machine. A disconnect could also occur by accident, for example, due to a power failure or other software or hardware error. A disconnect can also occur if one appliance attempts to open more than the maximum number of SSH connections allowed in a session. A machine that is disconnected from a cluster can still be accessed directly and configured; however, any changes made will not be propagated to other machines within the cluster until the disconnected machine becomes reconnected.

When a machine reconnects to the cluster, it tries to reconnect to all machines at once.

In theory, two machines in a cluster that are disconnected could commit a similar change to their local databases at the same time. When the machines are reconnected to the cluster, an attempt will be made to synchronize these changes. If there is a conflict, the most recent change is recorded (supersedes any other changes).

During a commit, the appliance checks every variable that is being changed. The commit data includes version information, sequence identification numbers, and other information that can be compared. If the data you are about to change is found to be in conflict with previous changes, you will be given the option to discard your changes. For example, you might see something like this:

```
(Machine mail3.example.com)> clustercheck



This command is restricted to "cluster" mode.  Would you like to switch to "cluster"
mode? [Y]> y



Checking Listeners (including HAT, RAT, bounce profiles)...

Inconsistency found!

  Listeners (including HAT, RAT, bounce profiles) at Cluster enterprise:

    mail3.example.com was updated Mon Sep 12 10:59:17 2005 PDT by 'admin' on
mail3.example.com

    test.example.com was updated Mon Sep 12 10:59:17 2005 PDT by 'admin' on
mail3.example.com



How do you want to resolve this inconsistency?

1. Force entire cluster to use test.example.com version.

2. Force entire cluster to use mail3.example.com version.

3. Ignore.

[1]>
```

If you choose not to discard your changes, they are still intact (but uncommitted). You can review your changes against the current settings and decide how to proceed.

You can also use the `clustercheck` command at any time to verify that the cluster is operating correctly.

```
losangeles> clustercheck


Do you want to check the config consistency across all machines in the cluster? [Y]> y


Checking losangeles...

Checking newyork...

No inconsistencies found.
```

Interdependent Settings

In a centrally managed environment, some interdependent settings are configured in different modes. The flexibility of the configuration model allows you to configure settings at multiple modes, and the laws of inheritance govern which settings will be used on a per-machine basis. However, some settings have dependencies on other settings, and the availability of the dependent settings' configuration is not limited to settings at the same mode. Thus, it is possible to configure a setting for one level that references a setting that is configured for a specific machine at a different level.

The most common example of an interdependent setting involves a select field on a page that pulls data from a different cluster section. For example, the following features can be configured in different modes:

- using LDAP queries
- using dictionaries or text resources
- using bounce or SMTP authentication profiles.

Within centralized management, there are restricted and non-restricted commands. (See Restricted Commands, page 8-15.) Non-restricted commands are generally configuration commands that can be shared across the cluster.

The `listenerconfig` command is an example of a command that can be configured for all machines in a cluster. Non-restricted commands represent commands that can be mirrored on all machines in a cluster, and do not require machine-specific data to be modified.

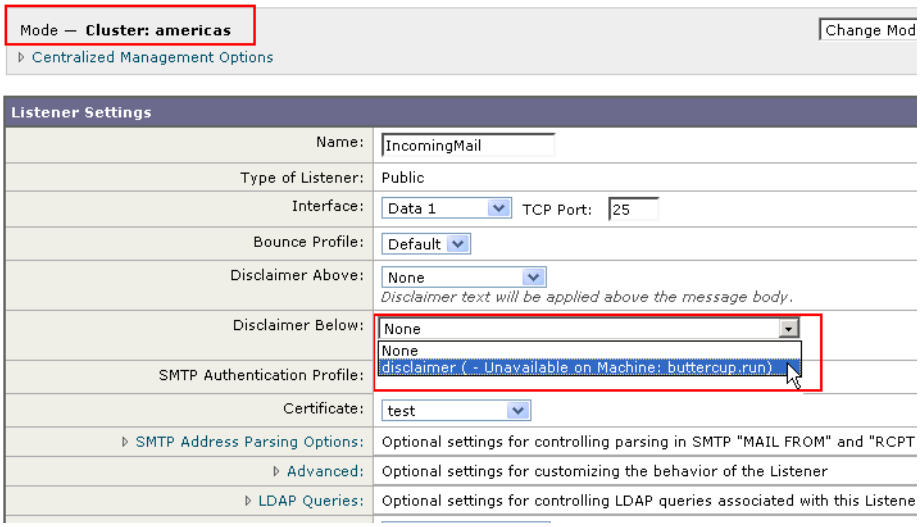
Restricted commands, on the other hand, are commands that only apply to a specific mode. For example, users cannot be configured for specific machines — there must be only one user set across the whole cluster. (Otherwise, it would be impossible to login to remote machines with the same login.) Likewise, since the Mail Flow Monitor data, System Overview counters, and log files are only maintained on a per-machine basis, these commands and pages must be restricted to a machine.

You will notice that while Scheduled Reports may be configured identically across the whole cluster, the viewing of reports is machine-specific. Therefore, within a single Scheduled Reports page in the GUI, configuration must be performed at the cluster mode, but viewing of reports must be done at the machine mode.

The System Time pages encompass the `settz`, `ntpconfig`, and `settime` commands, and thus represents a mixture of restricted and non-restricted commands. In this case, `settime` must be restricted to machine-only modes (since time settings are specific for machine), while `settz` and `ntpconfig` may be configured at cluster or group modes.

Consider the following example:

Figure 8-8 Example of Interdependent Settings



In this representation, the listener "IncomingMail" is referencing a footer named "disclaimer" that has been configured at the machine level only. The drop-down list of available footer resources shows that the footer is not available on the machine "buttercup.run" which is also available in the cluster. There are two solutions to this dilemma:

- promote the footer "disclaimer" from the machine level to the cluster level
- demote the listener to the machine level to remove the interdependency

In order to fully maximize the features of a centrally managed system, the former solution is preferred. Be aware of interdependencies among settings as you tailor the configuration of your clustered machines.

Best Practices and Frequently Asked Questions

Best Practices

When you create the cluster, the machine you happen to be logged into is automatically added to the cluster as the first machine, and also added to the Main_Group. Its machine level settings effectively get moved to the cluster level as much as possible. There are no settings at the group level, and the only settings left at the machine level are those which do not make sense at the cluster level, and cannot be clustered. Examples are IP addresses, featurekeys, etc.

Leave as many settings at the cluster level as possible. If only one machine in the cluster needs a different setting, copy that cluster setting to the machine level for that machine. Do not move that setting. If you move a setting which has no factory default (e.g. HAT table, SMTPROUTES table, LDAP server profile, etc.), the systems inheriting the cluster settings will have blank tables and will probably not process email.

To have that machine re-inherit the cluster setting, manage the CM settings and delete the machine setting. You will only know if a machine is overriding the cluster setting when you see this display:

```
Settings are defined:

To inherit settings from a higher level: Delete Settings for this feature at this mode.

You can also Manage Settings.

Settings for this feature are also defined at:

Cluster: xxx
```

Or this display:

```
Delete settings from:

    Cluster: xxx

    Machine: yyyy.domain.com
```

Copy vs. Move

When to copy: when you want the cluster to have a setting, and a group or machine to also have no settings or to have different settings.

When to move: when you want the cluster to have no setting at all, and for the group or machine to have the settings.

Good CM Design Practices

When you LIST your CM machines, you want to see something like this:

```
cluster = CompanyName

Group Main_Group:

    Machine lab1.example.com (Serial #: XXXXXXXXXXXX-XXXXXXX)

    Machine lab2.example.com (Serial #: XXXXXXXXXXXX-XXXXXXX)

Group Paris:

    Machine lab3.example.com (Serial #: XXXXXXXXXXXX-XXXXXXX)

    Machine lab4.example.com (Serial #: XXXXXXXXXXXX-XXXXXXX)

Group Rome:

    Machine lab5.example.com (Serial #: XXXXXXXXXXXX-XXXXXXX)

    Machine lab6.example.com (Serial #: XXXXXXXXXXXX-XXXXXXX)
```

Be careful not to lose track of the level at which you are making changes. For example, if you have changed the name of your Main_Group (using RENAMEGROUP) to London, it will look like this:

```
cluster = CompanyName
```

```
Group London:

    Machine lab1.cable.nu (Serial #: 000F1FF7B3F0-CF2SX51)

    ...
```

However, this configuration tends to confuse many administrators, because they begin making changes to the London systems at the group level, and they stop using the Cluster level as the normal configuration level for basic settings.

Tip: it is not a good practice to have a group with the same name as the cluster, e.g. cluster London, group London. If you are using site names for group names, it is not good practice to have a cluster name that refers to a location.

The correct method, as explained above, is to leave as many settings at the cluster level as possible. In most cases you should leave your primary site or main collection of machines in the Main_Group, and use groups for your additional sites. This is true even if you consider that both sites are "equal." Remember, CM has no primary/secondary or master/slave servers — all clustered machines are peers.

Tip: if you will be using extra groups you can easily prepare the groups before those extra machines are joined to the cluster.

Procedures: Configuring an Example Cluster

To configure this example cluster, log out of all GUIs on all machines before running `clusterconfig`. Run `clusterconfig` on any one of the primary site machines. You will then join to this cluster only the other local and remote machines that need the maximum possible shared settings (allowing for the machine only-settings like IP address). The `clusterconfig` command cannot be used to join a remote machine to the cluster — you must use the CLI on the remote machine and run `clusterconfig` ("join an existing cluster").

In our example above we log in to lab1, run `clusterconfig` and create a cluster called CompanyName. We have only one machine with identical requirements, so we log in to lab2, and `saveconfig` the existing configuration (it will be drastically altered when it inherits most of lab1 settings.) On lab2 we can then use `clusterconfig` to join an existing cluster. Repeat if you have additional machines at this site needing similar policies and settings.

Run CONNSTATUS to confirm that DNS resolves correctly. As machines are joined to the cluster, the new machines inherit almost all of their settings from lab1 and their older settings are lost. If they are production machines you will need to anticipate if mail will still be processed using the new configuration instead of their previous configuration. If you remove them from the cluster, they will not revert to their old, private configs.

Next, we count the number of exceptional machines. If there is only one, it should receive a few extra machine level settings and you will not need to create an extra group for it. Join it to the cluster and begin copying settings down to the machine level. If this machine is an existing production machine you must back up the configuration and consider the changes to mail processing as above.

If there are two or more, as in our example, decide if those two will share any settings with each other that are not shared with the cluster. In that case, you will be creating one or more groups for them. Otherwise, you will make machine level settings for each, and do not need to have extra groups.

In our case we want to run `clusterconfig` from the CLI on any of the machines already in the cluster, and select ADDGROUP. We will do this twice, once for Paris and once for Rome.

Now you can begin using the GUI and CLI to build configuration settings for the cluster and for ALL the groups, even if the groups have no machines in them yet. You will only be able to create machine specific settings for machines *after* they have joined the cluster.

The best way to create your override or exceptional settings is to copy the settings from the higher (e.g. cluster) level down to a lower (e.g. group) level.

For example, after creating the cluster our `dnsconfig` settings initially looked like this:

```
Configured at mode:

Cluster: Yes

Group Main_Group: No

Group Paris: No

Group Rome: No

Machine lab2.cable.nu: No
```

If we "Copy to Group" the DNS settings, it will look like this:

```
Configured at mode:

Cluster: Yes

Group Main_Group: No

Group Paris: Yes

Group Rome: No

Machine lab2.cable.nu: No
```

Now you can edit the Paris group-level DNS settings, and other machines in the Paris group will inherit them. Non-Paris machines will inherit the cluster settings, unless they have machine-specific settings. Besides DNS settings, it is common to create group level settings for SMTPROUTES.

Tip: when using the CLI CLUSTERSET function in various menus, you can use a special option to copy settings to All Groups, which is not available through the GUI.

Tip: complete listeners will be automatically inherited from the group or cluster, and you normally only create these on the first system in the cluster. This reduces administration considerably. However, for this to work *you must name the Interfaces identically throughout your group or cluster.*

Once the settings are defined correctly at the group level, you can join machines to the cluster and make them part of this group. This requires two steps:

First, to join our remaining 4 systems to the cluster, we run `clusterconfig` on each. The larger and more complex the cluster, the longer it takes to join, and this can take several minutes. You can monitor the joining progress with the LIST and CONNSTATUS sub-commands. After the joins are complete you can use SETGROUP to move the machines from the Main_Group into Paris and Rome. There is no way to avoid the fact that initially, all machines added to the cluster inherit the Main_Group settings, not the Paris and Rome settings. This could affect mail flow traffic if the new systems are already in production.

Tip: do not make your lab machines part of the same cluster as your production machines. Use a new cluster name for lab systems. This provides an added layer of protection against unexpected changes (someone changing a lab system and accidently losing production mail, for example).

Summary of GUI Options for Using CM Settings Other Than the Cluster Default

Override settings, and start with default settings. For example, the default settings for the SMTPROUTES configuration is a blank table, which you can then build from scratch.

Override settings, but start with a copy of the settings currently inherited from Cluster xxx, or group yyy. For example, you may want to a new copy of the SMTPROUTES table at the group level which is initially identical to the cluster table. All Cisco IronPort appliances that are contained in that same group (SETGROUP) will get this table. Machines not in the group will still use the cluster level settings. Changing the SMTPROUTES on this independent copy of the table will not affect other groups, machines inheriting the cluster settings, or machines where the setting is defined at the individual machine level. This is the most common selection.

Manage settings, a sub-menu of Centralized Management Options. From this menu you can copy as above, but you can also move or delete settings. If you move the SMTPROUTES to a group or machine level, then the routes table will be blank at the cluster level but will exist at the more specific level.

Manage settings. Continuing our SMTPROUTES example, using the delete option will also result in a blank SMTPROUTES table for the cluster. This is fine if you previously configured definitions for SMTPROUTES at the group level or machine levels. It is not a best practice to delete the cluster level settings and rely only on group or machine settings. The cluster-wide settings are useful as defaults on newly added machines, and keeping them reduces the number or group or site settings you have to maintain by one.

Setup and Configuration Questions

Q. How do I receive a Centralized Management feature key?

A. All Cisco IronPort appliances must have a unique feature key for Centralized Management installed before they can be joined together in a cluster. Contact Cisco IronPort Customer Support to obtain keys. Use the System Administration > Feature Keys page (GUI) or the `featurekey` command (CLI) to install each key.

Q. I have a standalone appliance that has been fully configured and receiving mail with listeners, users, etc If I apply the Centralized Management feature key and create a new cluster, what happens to my settings?

A. If an appliance was previously configured in “standalone” mode, its standalone settings will be used when creating the cluster. That is, when you create a new cluster, using the `clusterconfig -> create cluster` command, all configurations are initially set at the cluster level. The next machine to join the cluster will receive all of these settings.

Q. I have a previously configured standalone machine and I join an existing cluster. What happens to my settings?

A. When a machine joins a cluster, all of that machine's clusterable settings will be inherited from the cluster level. Upon joining a cluster, all locally configured non-network settings will be lost, overwritten with the settings of the cluster and any associated groups. (This includes the user/password table; passwords and users are shared within a cluster).

Q. I have a clustered machine and I remove it (permanently) from the cluster. What happens to my settings?

A. When a machine is permanently removed from a cluster, its configuration hierarchy is “flattened” such that the machine will continue to work the same as it did when it was part of the cluster. All settings that the machine has been inheriting will be applied to the machine in the standalone setting.

For example, if there is only a cluster-mode Global Unsubscribe table, that Global Unsubscribe table data will be copied to the machine's local configuration when the machine is removed from the cluster.

General Questions

Q. Are log files aggregated within centrally managed machines?

A. No. Log files are still retained for each individual machines. The Security Management appliance can be used to aggregate mail logs from multiple machines for the purposes of tracking and reporting.

Q. How does User Access work?

A. The Cisco IronPort appliances share one database for the entire cluster. In particular, there is only `admin` account (and password) for the entire cluster.

Q. How should I cluster a data center?

A. Ideally, a data center would be a "group" within a cluster, not its own cluster. However, if the data centers do not share much between themselves, you may have better results with separate clusters for each data center.

Q. What happens if systems are offline and they reconnect?

A. Systems attempt to synchronize upon reconnecting to the cluster.

Network Questions

Q. Is the centralized management feature a "peer-to-peer" architecture or a "master/slave" architecture?

A. Because every machine has all of the data for all of the machines (including all machine-specific settings that it will never use), the centralized management feature can be considered a peer-to-peer architecture.

Q. How do I set up a box so it is not a peer? I want a "slave" system.

A. Creating a true "slave" machine is not possible with this architecture. However, you can disable the HTTP (GUI) and SSH/Telnet (CLI) access at the machine level. In this manner, a machine without GUI or CLI access *only* be configured by clusterconfig commands (that is, it can never be a login host). This is similar to having a slave, but the configuration can be defeated by turning on login access again.

Q. Can I create multiple, segmented clusters?

A. Isolated "islands" of clusters are possible; in fact, there may be situations where creating them may be beneficial, for example, for performance reasons.

Q. I would like to reconfigure the IP address and hostname on one of my clustered appliances. If I do this, will I lose my GUI/CLI session before being able to run the reboot command?

Follow these steps:

a. Add the new IP address

b. Move the listener onto the new address

c. Leave the cluster

d. Change the hostname

e. Make sure that oldmachinename does not appear in the `clusterconfig` connections list when viewed from any machine

f. Make sure that all GUI sessions are logged out

g. Make sure that CCS is not enabled on any interface (check via `interfaceconfig` or Network > Listeners)

h. Add the machine back into the cluster

Q. Can the Destination Controls function be applied at the cluster level, or is it local machine level only?

It may be set at a cluster level; however, the limits are on a per-machine basis. So if you limit to 50 connections, that is the limit set for each machine in the cluster.

Planning and Configuration

Q. What can I do to maximize efficiency and minimize problems when setting up a cluster?

1. Initial Planning
   - Try to configure as many things as possible at the cluster level.
   - Manage by machines only for the exceptions.
   - If you have multiple data centers, for example, use groups to share traits that are neither cluster-wide nor necessarily machine-specific.
   - Use the same name for Interfaces and Listeners on each of the appliances.
2. Be aware of restricted commands.
3. Pay attention to interdependencies among settings.

   For example, the `listenerconfig` command (even at the cluster level) depends on interfaces that only exist at a machine level. If the interface does not exist at the machine level on all machines in the cluster, that listener will be disabled.

   Note that deleting an interface would also affect `listenerconfig`.
4. Pay attention to your settings!

   Remember that previously-configured machines will lose their independent settings upon joining a cluster. If you want to re-apply some of these previously configured settings at the machine level, be sure to take note of all settings before joining the cluster.

   Remember that a “disconnected” machine is still part of the cluster. When it is reconnected, any changes you made while it was offline will be synchronized with the rest of the cluster.

   Remember that if you permanently remove a machine from a cluster, it will retain all of the settings it had as part of that cluster. However, if you change your mind and re-join the cluster, the machine will lose all standalone settings. This is unlikely to restore the configuration to the state you intended.

   Use the `saveconfig` command to keep records of settings.



APPENDIX A

AsyncOS Quick Reference Guide

Use the quick reference guide in this appendix to locate the appropriate CLI command and its purpose.

Table A-1 CLI Commands (No commit required)

antispamstatus	Display Anti-Spam status
antispamupdate	Manually update spam definitions
antivirusstatus	Display Anti-Virus status
antivirusupdate	Manually update virus definitions
bouncerecipients	Bounce messages from the queue
clearchanges *or* **clear**	Clear changes
commit	Commit changes
commitdetail	Display detailed information about the last commit
diagnostic	Hardware and software troubleshooting utilities
deleterecipients	Delete messages from the queue
delivernow	Reschedule messages for immediate delivery
dnsflush	Clear all entries from the DNS cache
dnslistflush	Flush the current DNS List cache
dnslisttest	Test a DNS lookup for a DNS-based list service
dnsstatus	Display DNS statistics
encryptionstatus	Shows the version of the PXE Engine and Domain Mappings file
encryptionupdate	Requests an update to the PXE Engine
featurekey	Administer system feature keys
help *or* **h** *or* **?**	Help
hostrate	Monitor activity for a particular host
hoststatus	Get the status of the given hostname
last	Display who has recently logged into the system
ldapflush	Flush any cached LDAP results
ldaptest	Perform a single LDAP query test
mailconfig	Mail the current configuration to an email address

Table A-1 CLI Commands (No commit required) (Continued)

netstat	Displays network connections, routing tables, and a number of network interface statistics.
nslookup	Query a nameserver
outbreakflush	Clear the cached Outbreak Rules
outbreakstatus	Display current Outbreak Rules
outbreakupdate	Update Outbreak Filters rules
packetcapture	Intercept and display packets being transmitted or received over the network
ping	Ping a network host
quit *or* **q** *or* **exit**	Quit
rate	Monitor message throughput
reboot	Restart the system
redirectrecipients	Redirect all messages to another relay host
resetconfig	Restore the factory configuration defaults
resetcounters	Reset all of the counters in the system
resume	Resume receiving and deliveries
resumedel	Resume deliveries
resumelistener	Resume receiving
rollovernow	Roll over a log file
saveconfig	Saves the configuration to disk
sbstatus	Display status of SenderBase queries
settime	Manually set the system clock
showconfig	Display all configuration values
showmessage	Display message
showrecipients	Show messages from the queue
shutdown	Shut down the system to power off
status	System status
supportrequest	Send a message to Cisco IronPort Customer Support
suspend	Suspend receiving and deliveries
suspenddel	Suspend deliveries
suspendlistener	Suspend receiving
systemsetup	First time system setup
tail	Continuously display the end of a log file.
techsupport	Allow Cisco IronPort customer service to access your system
telnet	Connect to a remote host
tlsverify	Establish an outbound TLS connection to a remote host and debug any TLS connection issues
tophosts	Display the top hosts by queue size

Table A-1 CLI Commands (No commit required) (Continued)

topin	Display the top hosts by number of incoming connections
trace	Trace the flow of a message through the system
traceroute	Display the network route to a remote host
tzupdate	Update timezone rules
updatenow	Update all components
upgrade	Install an upgrade
version	View system version information
who	List who is logged in
whoami	Display your current user id
workqueue	Display and/or alter work queue pause status

The commands in Table A-2 require you to issue the `commit` command in order to take effect

Table A-2 CLI Commands (commit required)

addressconfig	Configure From: addresses for system generated mail
adminaccessconfig	Configure network access list and banner login
alertconfig	Configure email alerts
aliasconfig	Configure email aliases
altsrchost	Configure Virtual Gateway(tm) mappings
antispamconfig	Configure Anti-Spam policy
antivirusconfig	Configure Anti-Virus policy
bounceconfig	Configure the behavior of bounces
bvconfig	Configure Cisco IronPort Bounce Verification
certconfig	Configure security certificates and keys
clusterconfig	Configure cluster related settings
deliveryconfig	Configure mail delivery
destconfig	Configure Destination Controls
dictionaryconfig	Configure content dictionaries
dnsconfig	Configure DNS setup
dnslistconfig	Configure DNS List services support
domainkeysconfig	Configure DomainKeys support
etherconfig	Configure Ethernet settings
exceptionconfig	Configure domain exception table
filters	Configure message processing options
incomingrelayconfig	Configure Incoming Relays
interfaceconfig	Configure Ethernet IP addresses
listenerconfig	Configure mail listeners
ldapconfig	Configure LDAP servers

Table A-2 CLI Commands (commit required) (Continued)

loadconfig	Load a configuration file
localeconfig	Configure multi-lingual settings
logconfig	Configure access to log files
ntpconfig	Configure NTP time server
outbreakconfig	Configure Outbreak Filters
password or passwd	Change your password
policyconfig	Configure per recipient or sender based policies
quarantineconfig	Configure system quarantines
routeconfig	Configure IP routing table
scanconfig	Configure attachment scanning policy
senderbaseconfig	Configure SenderBase connection settings
setgateway	Set the default gateway (router)
destconfig	Set outbound host limits and delivery settings
sethostname	Set the name of the machine
settz	Set the local time zone
sievechar	Configure characters for Sieve Email Filtering
smtpauthconfig	Configure SMTP Auth profiles
smtproutes	Set up permanent domain redirections
snmpconfig	Configure SNMP
sshconfig	Configure SSH keys
sslconfig	Configure SSL settings
stripheaders	Set message headers to remove
textconfig	Configure text resources
unsubscribe	Update the global unsubscribe list
updateconfig	Configure system update parameters
userconfig	Manage user accounts and connections to external authentication sources



APPENDIX B

Accessing the Appliance

You can access any interface you create on the appliance through a variety of services.

By default, the following services are either enabled or disabled on each interface:

Table B-1 Services Enabled by Default on Interfaces

		Enabled by default?	
Service	**Default port**	**Management interface**[a]	**New interfaces you create**
FTP	21	No	No
Telnet	23	Yes	No
SSH	22	Yes	No
HTTP	80	Yes	No
HTTPS	443	Yes	No

a.The "Management Interface" settings shown here are also the default settings for the Data 1 Interface on Cisco IronPort C10 appliances.

- If you need to access the appliance via the graphical user interface (GUI), you must enable HTTP and/or HTTPS on an interface.
- If you need to access the appliance for the purposes of uploading or downloading configuration files, you must enable FTP or Telnet on an interface.
- You can also upload or download files using secure copy (`scp`).

FTP Access

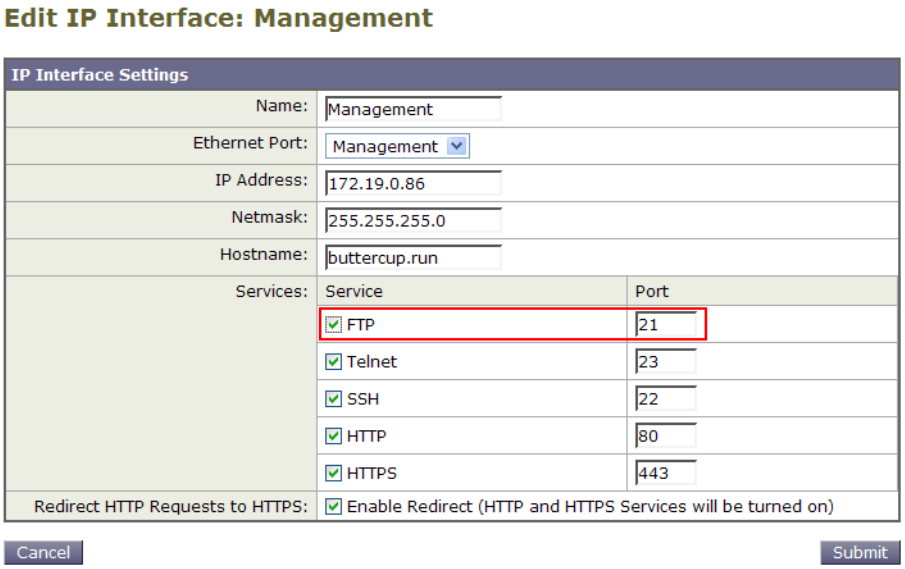
To access the appliance via FTP, follow these steps:

Step 1 Use the Network > IP Interfaces page or the `interfaceconfig` command to enable FTP access for the interface.

WARNING: By disabling services via the `interfaceconfig` command, you have the potential to disconnect yourself from the CLI, depending on how you are connected to the appliance. Do not disable services with this command if you are not able to reconnect to the appliance using another protocol, the Serial interface, or the default settings on the Management port.

In this example, the Management interface is edited to enable FTP access on port 21 (the default port):

Figure B-1 Edit IP Interface Page



Note Remember to `commit` your changes before moving on to the next step.

Step 2 Access the interface via FTP. Ensure you are using the correct IP address for the interface. For example:

```
$ ftp 192.168.42.42
```

Note Many browsers also allow you to access interfaces via FTP.

Step 3 Browse to the directory for the specific task you are trying to accomplish. After you have accessed an interface via FTP, you can browse the following directories to copy and add ("GET" and "PUT") files. See the following table.

Directory Name	Description
/configuration	The directory where data from the following commands is exported to and/or imported (saved) from: • Virtual Gateway mappings (`altsrchost`) • configuration data in XML format (`saveconfig, loadconfig`) • Host Access Table (HAT) (`hostaccess`) • Recipient Access Table (RAT) (`rcptaccess`) • SMTP routes entries (`smtproutes`) • alias tables (`aliasconfig`) • masquerading tables (`masquerade`) • message **filters** (`filters`) • global unsubscribe data (`unsubscribe`) • test messages for the `trace` command • Safelist/Blocklist backup file, saved in the following format: *slbl<timestamp><serial number>.csv*
/antivirus	The directory where the Anti-Virus engine log files are kept. You can inspect the log files this directory to manually check for the last successful download of the virus definition file (`scan.dat`).
/configuration /system_logs /cli_logs /status /reportd_logs reportqueryd_logs /ftpd_logs /mail_logs /asarchive /bounces /error_logs /avarchive /gui_logs /sntpd_logs /RAID.output /euq_logs /scanning /antispam /antivirus /euqgui_logs /ipmitool.output	Created automatically for **logging** via the `logconfig` and `rollovernow` commands. See "Logging" in the *Cisco IronPort AsyncOS for Email Daily Management Guide* for a detailed description of each log. See "Log File Type Comparison" for the differences between each log file type.

Step 4 Use your FTP program to upload and download files to and from the appropriate directory.

Secure Copy (scp) Access

If your client operating system supports a secure copy (`scp`) command, you can copy files to and from the directories listed in the previous table. For example, in the following example, the file `/tmp/test.txt` is copied from the client machine to the configuration directory of the appliance with the hostname of `mail3.example.com`.

Note that the command prompts for the password for the user (`admin`). This example is shown for reference only; your particular operating system's implementation of secure copy may vary.

```
% scp /tmp/test.txt admin@mail3.example.com:configuration

The authenticity of host 'mail3.example.com (192.168.42.42)' can't be established.

DSA key fingerprint is 69:02:01:1d:9b:eb:eb:80:0c:a1:f5:a6:61:da:c8:db.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'mail3.example.com ' (DSA) to the list of known hosts.

admin@mail3.example.com's password: (type the password)

test.txt             100% |***************************|  1007       00:00

%
```

In this example, the same file is copied from the appliance to the client machine:

```
% scp admin@mail3.example.com:configuration/text.txt .

admin@mail3.example.com's password: (type the password)

test.txt             100% |***************************|  1007       00:00

%
```

You can use secure copy (`scp`) as an alternative to FTP to transfer files to and from the Cisco IronPort appliance.

Note Only users in the operators and administrators group can use secure copy (`scp`) to access the appliance. For more information, see "Adding Users" in the "Common Administrative Tasks" chapter in the *Cisco IronPort AsyncOS for Email Daily Management Guide*.

Accessing via a Serial Connection

If you are connecting to the appliance via a serial connection (see "Connecting to the Appliance" in the *Cisco IronPort AsyncOS for Email Configuration Guide*), Figure B-2 illustrates the pin numbers for the serial port connector, and Table B-2 defines the pin assignments and interface signals for the serial port connector.

Figure B-2 Pin Numbers for the Serial Port

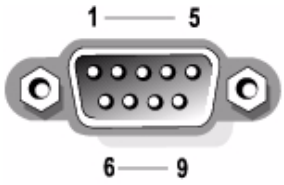


Table B-2 Serial Port Pin Assignments

Pin	Signal	I/O	Definition
1	DCD	I	Data carrier detect
2	SIN	I	Serial input
3	SOUT	O	Serial output
4	DTR	O	Data terminal ready
5	GND	n/a	Signal ground
6	DSR	I	Data set ready
7	RTS	I	Request to send
8	CTS	O	Clear to send
9	RI	I	Ring indicator
Shell	n/a	n/a	Chassis ground



INDEX

Symbols

`/dev/null`, in alias tables **2-3, 2-9**

`/etc/mail/aliases` 2-8

`/etc/mail/genericstable` 2-17

Numerics

4XX error code **2-36**

5XX error code **2-36**

A

Active Directory **3-20**

address literals **1-11**

address rewriting **2-8**

address tagging key

- purging **2-56**

`aliasconfig` command **2-9, 2-12**

Alias tables

- virtusertable **2-8**

alias tables

- `aliasconfig` command **2-9**
- commenting **2-9**
- configuring via CLI **2-8**
- definition **2-8**
- multiple entries **2-9**

ALL entry

- in masquerading **2-18**

alternate MX host **2-2**

`altsrchost` command **2-17, 2-61**

anti-spam

- HAT parameter **1-15**

appending domains **1-10**

auto delivery feature **2-57**

auto-select **2-57**

B

bare addresses **1-10**

Base DN **3-12**

blackhole listener **1-2**

body scanning **6-28**

`bounceconfig` Command **2-40**

bounce profile **2-41**

bounces

- conversational **2-36**
- non-conversational **2-36**

Bounce Verification **2-53**

bypassing

- anti-spam **6-63**

C

caching SenderBase data **1-5**

call-ahead SMTP server **4-1**

- routing **4-8**

canonicalization **5-5**

case-sensitivity

- in LDAP queries **3-13, 3-18**
- in message filters **6-17**

Centralized Management

- and Destination Controls **8-29**

certificate

- certificate authority **1-22**

certificates

adding 1-23
certificate authorities list 1-25
exporting 1-25
generating and signing your own 1-22
generating a request 1-24
importing 1-22
intermediate certificates 1-23
Certificate Signing Request 1-22
chain query
creating 3-28
LDAP 3-27
chains, of aliases 2-9
closing unsuccessful or unproductive inbound connections 1-5
command quick reference A-1
comments 2-6
comments in imported files 2-6
conformance level
SPF/SIDF verification 5-24
conversational bounces 2-36
CRAM-MD5 3-36

D

default
sender domain 1-10
delayed bounces 2-36
delivering mail 2-44
controlling 2-44
controlling mail to destination domain 2-44
message time out 2-57
possible delivery 2-57
delivery 2-1
encrypting 1-22
`deliveryconfig` command 2-58
demo certificate 1-23
demonstration certificate 1-29
`destconfig` command 1-32, 2-46
Destination Controls 2-46
and Centralized Management 8-29
importing and exporting configurations 2-49
Directory Harvest Attack (DHA) 3-28
Direct Server Return (DSR) 7-15
DKIM
DNS TXT record 5-5
domain profile 5-3
enabled on a mail flow policy 5-3
signing 5-3
DKIM verification 5-21
Authentication-Results header 5-21
DNSBL 6-31
DNS list 6-31
DNS TXT record 5-3
domain
adding a default domain 1-10
domain context
in alias tables 2-8, 2-12
Domain Keys 5-2
canonicalization 5-5
DNS text record 5-12
DNS TXT record 5-5
domain profile 5-3
enabled on a mail flow policy 5-3
exporting domain profiles 5-14
importing domain profiles 5-14
importing signing keys 5-12
selector 5-5
signing 5-3
signing key size 5-4
testing a domain profile 5-13
verification 5-2
verifying signatures 5-2
DomainKey-Signature header 5-3
domain map
commenting 2-34
importing and exporting 2-34
importing invalid entries 2-34
limits 2-29

overview 2-28
domain profile
deleting all existing profiles 5-14
exporting 5-14
importing 5-14
removing domain profiles 5-14
testing 5-13
domain table 2-28
drop-attachments-where-dictionary-match 6-74
DSN (delay notification messages) 2-40
DSR 7-15
load balancing 7-15
loopback interface 7-15
Virtual IP (VIP) 7-15
dual DKIM and DomainKey signing 5-7
duplex settings, editing 7-1

E

email
rewriting addresses 2-8
email address
source routing 1-10
enabling DomainKeys and DKIM signing on a mail flow policy 5-3
encryption 1-15, 1-22
Envelope Recipient 2-8, 6-22
Envelope Recipient, rewriting 2-8
Envelope Sender 6-22
Envelope Sender, rewriting 2-16
Envelope To 2-8
Envelope To, rewriting in alias tables 2-8
external authentication 3-40

F

filtering unparsable messages 6-20
filters 6-1
comment character 6-3
matching dictionary terms 6-13, 6-33
matching empty headers 6-21
regular express and Python 6-17
scannable archive file types 6-28
unparsable messages 6-20
FTP B-1
FTP Access B-1

G

`genericstable` file 2-19
global alias 2-9
global unsubscribe
adding 2-70
commenting 2-72
importing and exporting 2-72
maximum entries 2-69
overview 2-69
syntax 2-69
globbing 2-2
good neightbor table 1-31

H

HAT
delayed rejections 1-6
HAT delayed rejections 1-6
headers 2-8, 2-16, 2-18
headers, stripping with message filters 6-60
hiding network topology 1-11, 2-16
Host Access Table (HAT)
definition 1-3
HTTP B-1
HTTPS B-1
certificate for 1-35

I

image analysis 6-67
image scanning 6-67
image verdicts 6-67
importing domain profiles 5-14
importing signing keys 5-12
inbound connections
  closing unusccessful or unproductive connections 1-5
inbound connection timeout 1-5
inbound email gateway 1-1
injection control counter reset 1-19
injection control periodicity 1-19
injection counters reset period 1-5
`interface` command 2-57
IP interfaces
  defining in `listenerconfig` command 1-2
IP port
  defining in `listenerconfig` command 1-2
IronPort Spam Quarantine
  stripping "SMTP:" in the LDAP query 3-43

K

key size 5-4

L

LDAP
  Accept query 1-12
  alias consolidation query 3-44
  alias expansion 3-20
  anonymous queries 3-14
  base DN 3-12
  chain queries 3-27
  connection pooling 3-33
  connections 3-16
  domain-based queries 3-26
  end-user authentication query 3-43
  external authentication 3-40
  failover 3-46
  group queries 6-22, 6-23
  LDAPS certificate 3-13
  load-balancing 3-46
  Microsoft Exchange 5.5 support 3-9
  multiple servers 3-46
  OpenLDAP queries 3-19
  query tokens 3-13
  recursive queries 3-14
  SSL 3-13
  SunONE queries 3-19
  testing queries 3-12, 3-17
  testing servers 3-6
  test servers 3-6
LDAP Accept during SMTP conversation 1-12
LDAP accept query 1-12
LDAP errors 3-18
LDAP routing query
  with SMTP call-ahead recipient validation 4-7
LDAPS certificate 3-13
limits
  `altsrchost` 2-64
  SMTP Routes 2-3
link aggregation 7-3
listener
  add a default domain 1-10
  adding received header 1-11
  add listener 1-7
  caching SenderBase data 1-5
  definition 1-1
  deleting 1-14
  edit global settings 1-7
  editing 1-14
  encrypting 1-22
  encryption on 1-15
  injection counters reset period 1-5
  LDAP accept query 1-12

loose SMTP address parsing 1-9
malformed MAIL FROM and default domain 1-11
maximum concurrent connections 1-5
strict SMTP address parsing 1-8
timeout for unsuccessful inbound connections 1-5
Total Time Limit for All Inbound Connections 1-6
`listenerconfig` command 1-2
loopback interface 7-15

M

mailertable feature 2-1
mail flow policies
`listenerconfig` Command 1-2
MAIL FROM 2-16, 6-9
mail loops, detecting 6-103
mail protocol
defining in `listenerconfig` command 1-2
malformed entries, in alias tables 2-9
mapping domains 2-2
`masquerade` subcommand 2-19
masquerading
and `altsrchost command` 2-17
commenting 2-18
configuring via CLI 2-17
definition 2-16
importing and exporting 2-19
importing invalid entries 2-19
limits 2-18
table syntax 2-17
via an LDAP query 2-17
via a static table 2-17
matching empty headers 6-21
maximum
message size in HAT 1-15
messages per connection in HAT 1-15
recipients per message in HAT 1-15
maximum concurrent connections 1-5
maximum connections for listeners 2-58
maximum recipients per hour 1-16
mbox format 6-59
message body scanning 6-29
message encoding 1-6
modifying 1-6, 6-89
setting for headings and footers 1-6
message filter
filter actions 6-42
message filters
adding 6-76
attachment-protected 6-12
attachment-unprotected 6-12
body-dictionary-match 6-33
combining 6-3, 6-14
deleting 6-77
encryption 6-29
exporting 6-81
importing 6-81
making (in)active 6-77
MIME types 6-28
moving 6-77
ordering 6-4
overview 6-1
random numbers in 6-27
rules 6-2
SenderBase Reputation Score 6-32
status 6-77
syntax 6-3
time and date 6-25
variables 6-48
message headers 6-26
message headers, inserting with message filters 6-60
message replication 6-43, 6-54
Microsoft Exchange, LDAP queries for 3-20
monitoring Virtual Gateway addresses 2-68
MTA 1-1, 1-22
multiple IP interfaces 2-63

N

NIC pairing 7-3
  alerts 7-4
  named on upgrade 7-4
NIC teaming 7-3
non-conversationsal bounces 2-36
numbers 1-15

P

partial domain
  in alias tables 2-8
  in Masquerading 2-18
PEM format, for certificates 1-24
possible delivery 2-57, 2-58
private key 1-22
prototcol
  see *mail protocol*
public blacklist 6-31
purging address tagging keys 2-56

Q

queries
  acceptance 3-18
  chain queries 3-27
  domain-based 3-26
  external authentication 3-40
  group 3-22
  masquerading 3-20
  routing 3-20
  SMTP authentication 3-32
  spam quarantine alias consolidation 3-44
  spam quarantine end-user authentication 3-43
queue 1-2

R

RBL 6-13
RCPT TO 6-9, 6-10
RCPT TO command 2-8
Received: Header, disabling 1-11
received header 1-11
Recipient Access Table (RAT)
  definition 1-3
recipients, counting in message filters 6-28
recipient validation 4-1
recursive entries
  in alias tbales 2-9
  in SMTP Routes 2-2
recursive queries, LDAP 3-14
redirecting email 2-2
relaying messages 1-1
required TLS 1-28
reverse DNS lookup 2-60
revertive switching 7-4
rewriting email addresses 2-8
RFC
  1035 2-8
  2487 1-22
  2821 1-8
round robin Virtual Gateways 2-60
routing 2-1
  SMTP call-ahead server 4-8

S

SBRS
  none 6-33
scanconfig
  scanning recursive levels of attachments 6-84
`scanconfig`
  setting max size for files to be scanned 6-84
  skipping attachment types 6-84
scannable archive file types 6-28

scanning images 6-67
scp command B-4
secure copy B-4
secure HTTP (https) 1-22
Secure LDAP 3-13
Secure Socket Layer (SSL) 1-22
SenderBase 1-16
  timeout per connection 1-12
  using IP Profiling 1-12
serial connection pinouts B-5
services for interfaces B-1
SIDF records
  testing 5-23
  valid 5-22
SIDF verification 6-10
  configuring 5-22
  conformance level 5-24
  enabling 5-24
  results 5-31
  testing 5-34
signing
  DKIM 5-3
  Domain Keys 5-3
  dual Domain Keys and DKIM 5-3
signing key
  size 5-4
signing keys
  deleting all existing keys 5-12
  removing specific keys 5-12
SMTP Address Parsing
  Loose mode 1-8, 1-9
  Strict mode 1-8
SMTP Auth 3-2, 3-31
  DIGEST-MD5 3-36
  MD5 3-32
  SHA 3-32
  suported authentication mechanisms 3-32
  TLS 3-37
SMTP authenticated user match filter rule 6-37
SMTP Authentication profile 3-36
SMTP Authentication with forwarding
  defined 3-34
SMTP call-ahead recipient validation 4-1
  bypassing 4-9
  conversation workflow 4-2
  SMTP Call-Ahead Server Profile 4-4
  SMTP server responses 4-5
  with LDAP routing query 4-7
SMTP CAll-Ahead Server Profile
  creating 4-3
  enabling on a listener 4-6
  settings 4-4
SMTP conversation
  SMTP call-ahead server 4-2
SMTP query workflow 4-8
SMTP Routes 2-1
  deleting all 2-5
  limits 2-3
  mail delivery and splintering 2-3
  multiple host entries 2-3
  recursive entries in 2-2
  USEDNS 2-3
SMTP Routes, maximum 2-2
SMTP Routes and DNS 2-3
source routing 1-10
spf-passed filter rule 5-34, 6-10
SPF records
  testing 5-23
  valid 5-22
spf-status filter rule 5-32, 6-10
SPF verification
  configuring 5-22
  conformance level 5-24
  enabling 5-24
  received SPF header 5-30
  results 5-31
  testing 5-34
SPFverification 6-10

SSL 3-13
STARTTLS
definition 1-22
static routes 2-57
strip-header filter action 6-60
strip headers 6-60
stripping subdomains 2-16
systemsetup command 1-3

T

TCP listen queue 1-12
Telnet B-1
TLS
certificates 1-22
default 1-30
preferred 1-30
required 1-30

U

unary form, in message filters 6-27
unparsable messages 6-20
uuencoded attachments 6-5

V

verification
SIDF 5-22
SPF 5-22
Virtual Domains 2-16
Virtual Gateway™ technology 2-59
Virtual Gateway addresses 2-63, 6-58
Virtual Gateway queue 2-60
Virtual IP (VIP) 7-15
virtual table 2-28
virususerstable. See alias tables
VLAN
defined 7-8
labels 7-9

W

white space 6-15
wizard
for listener 1-2

X

X.509 certificate 1-22