Configuring Routing and Delivery Features

This chapter contains the following sections:

Routing Email for Local Domains

In Configuring the Gateway to Receive Email 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 email gateway 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 and committed the changes, you defined the first SMTP route entries on the email gateway for each RAT entry you entered at that time.

Related Topics

- SMTP Routes Overview, on page 2
- Default SMTP Route, on page 2

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 AsyncOS operating system allows up to forty thousand (40,000) SMTP Route mappings to be configured for your email gateway. (See SMTP Routes Limits, on page 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 email gateway 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 email gateway 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 email gateway 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 email gateway 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.

Managing SMTP Routes to Send Outbound Email Using the GUI

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

Related Topics

- Adding SMTP Routes, on page 4
- Exporting SMTP Routes, on page 5
- Importing SMTP Routes, on page 5

Adding SMTP Routes

Procedure

Step 1 Click **Add Route** on the Network > SMTP Routes 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, on page 3for more information.

Step 5 Submit and commit your changes.

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:

Procedure

Step 1 Click **Export SMTP Routes** on the SMTP Routes page.

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:

Procedure

Step 1 Click **Import SMTP Routes** on the SMTP Routes page.

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**.

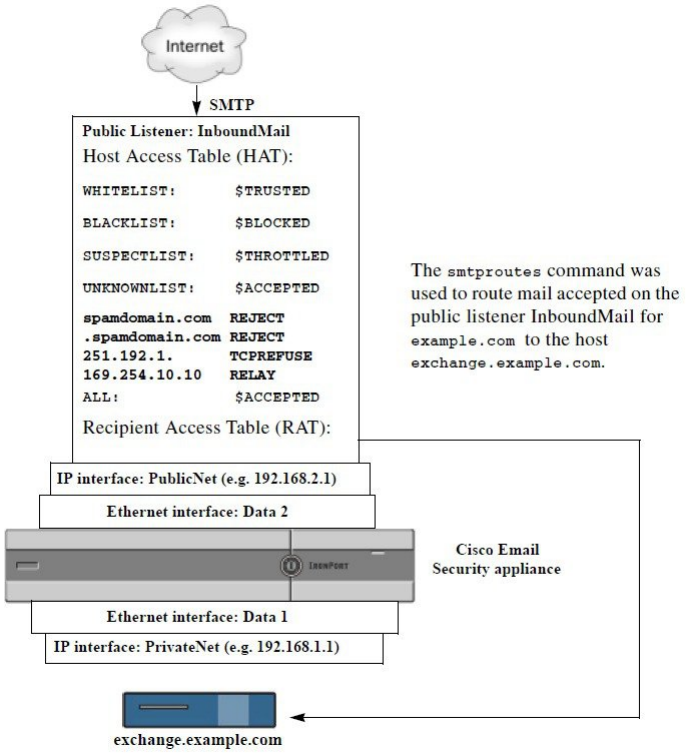
You can place “comments” in the file. Lines that begin with a ‘#’ character are considered comments and are ignored by AsyncOS. For example:

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

ALL:
```

What to do next

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

Figure 1: SMTP Routes Defined for a Public Listene



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.

The following table provides an overview of the various features used for rewriting sender and recipient email addresses.

Table 1: Methods for Rewriting Addresses

Original Address	Change to	Feature	Works on
*@anydomain	user@domain	Alias Tables (see Creating Alias Tables, on page 7)	• Envelope Recipients only • Applied globally • Maps aliases to email addresses or other aliases
*@olddomain	*@newdomain	Domain Mapping (see The Domain Map Feature, on page 24)	• Envelope Recipients only • Applied per listener

Original Address	Change to	Feature	Works on
*@olddomain	*@newdomain	Masquerading (see Configuring Masquerading, on page 14)	• 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. See the “Understanding the Email Pipeline” chapter.

Note The Alias Table functionality actually rewrites the Envelope Recipient of the email. This is different than the `smtproutes` command (see Directing Bounced Email, on page 30), which does not rewrite the Envelope Recipient of the email, but instead simply reroutes the email to specified domains.

Related Topics

- Configuring an Alias Table from the Command Line, on page 7
- Exporting and Importing an Alias Table , on page 8
- Deleting Entries from the Alias Table, on page 8

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: 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 the “Managing and Monitoring via the CLI” chapter.) The recipient is accepted but not enqueued.

Related Topics

- Example Alias Table, on page 9
- Example aliasconfig Command, on page 11

Exporting and Importing an Alias Table

To import an alias table, first see FTP, SSH, and SCP Accessto ensure that you can access the email gateway.

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`.

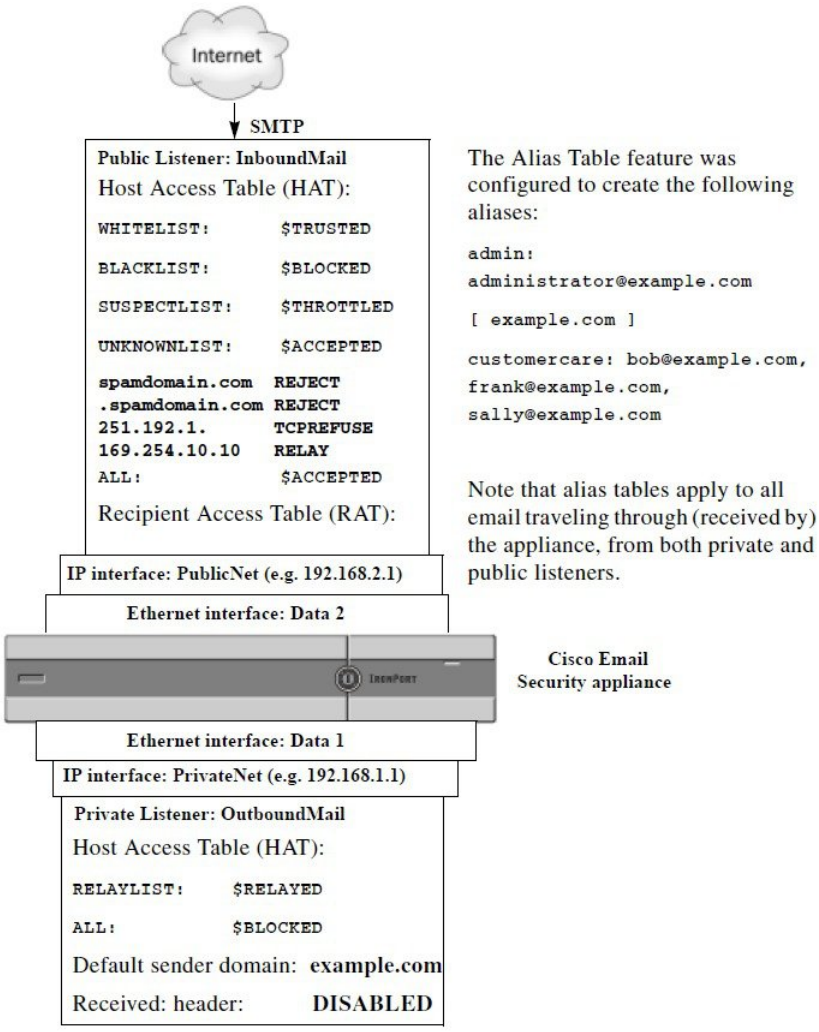
```
mail3.example.com> aliasconfig

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: Alias Tables Defined for the Email Gateway



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.

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. See the "Understanding the Email Pipeline" chapter.

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:

- via a static table of mappings you create
- 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 LDAP Queriesfor more information on LDAP masquerading queries.

Related Topics

- Masquerading and altsrchost, on page 15

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 Configuring Mail Gateways for all Hosted Domains Using Virtual Gateway™ Technology, on page 53and Review: Email Pipeline, on page 65.

Related Topics

- Configuring Static Masquerading Tables, on page 15
- Sample Masquerading Table for a Private Listener, on page 16
- Importing a Masquerading Table , on page 17
- Example Masquerading , on page 17

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 , on page 17. The subcommand creates and maintains a table that maps input addresses, usernames, and domains to new addresses and domains. See LDAP Queriesfor 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 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 FTP, SSH, and SCP Accessto ensure that you can access the email gateway.

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 -> listener_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 LDAP Queriesfor 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
```

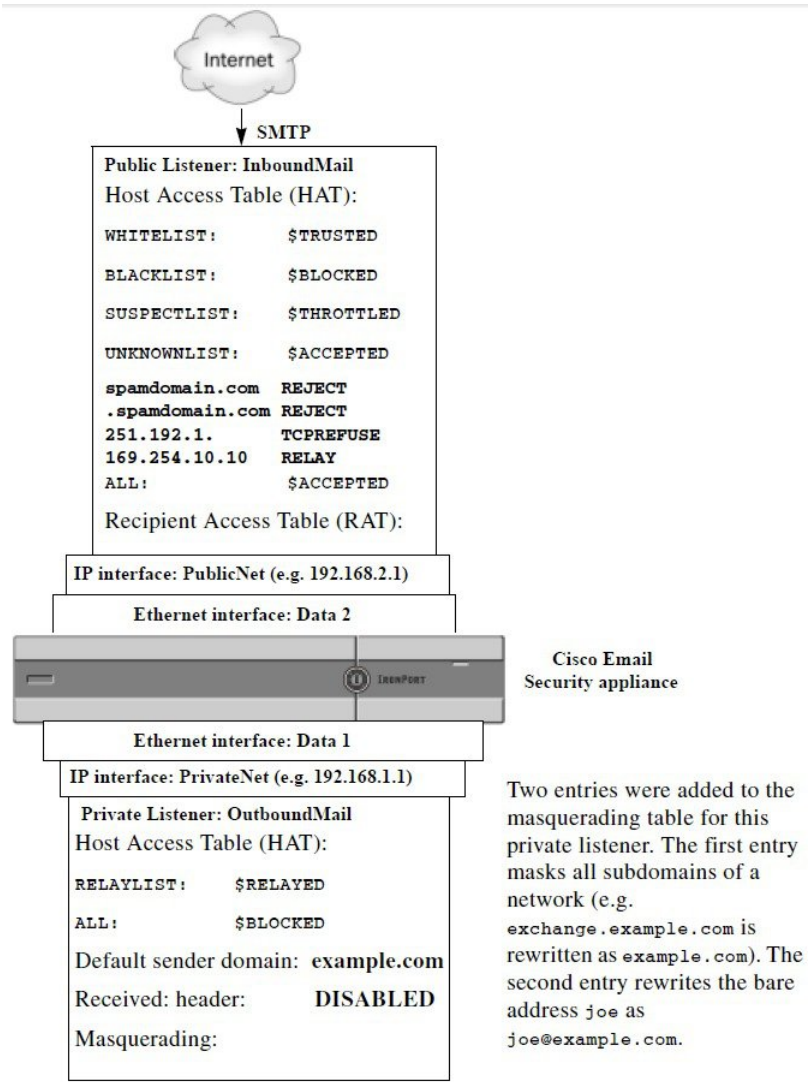
```
There are currently 2 entries.
Masqueraded headers: To, From, Cc
Choose the operation you want to perform:
- NEW - Create a new entry.
- DELETE - Remove an entry.
- PRINT - Display all entries.
- IMPORT - Import all entries from a file.
- EXPORT - Export all entries to a file.
- CONFIG - Configure 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.


[]>
```

Our Enterprise Gateway configuration now looks like this:

Figure 3: Masquerading Defined for a Private Listene



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. See the “Understanding the Email Pipeline” chapter.

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 email gateway 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 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.

[]>
```

Related Topics

- Importing and Exporting a Domain Map Table , on page 29

Importing and Exporting a Domain Map Table

To import or export a domain map table, first see FTP, SSH, and SCP Accessto ensure that you can access the email gateway.

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
```

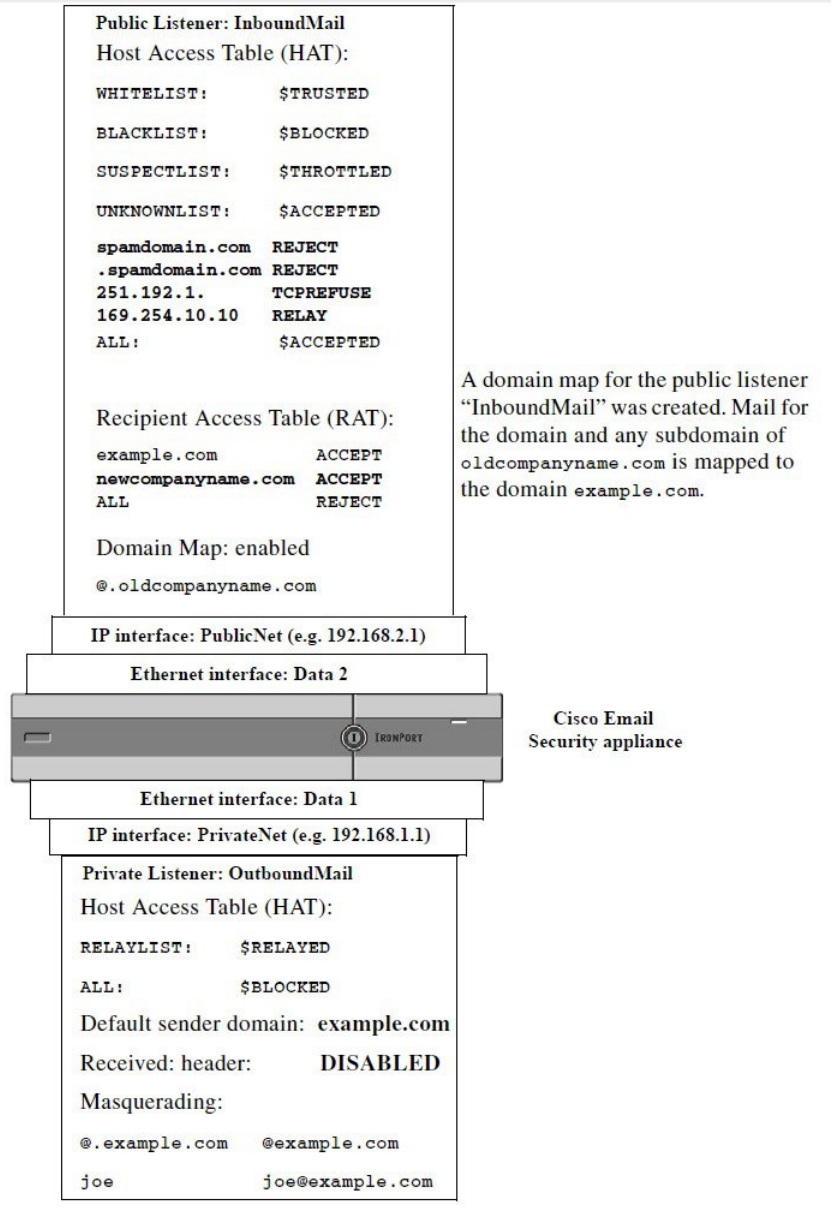
Alternatively, you can use the `export` subcommand to download the existing configuration. A file (whose name you specify) will be written to the configuration directory. You can modify this file outside of the CLI and then import it again.

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 4: Domain Map Defined for a Public Listener



Directing Bounced Email

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

Please note, this section describes how to control how your email gateway generates outgoing bounces (based on incoming mail). To control how your email gateway controls incoming bounces (based on outgoing mail) use Bounce Verification (see Bounce Verification, on page 39).

Related Topics

- Handling Undeliverable Email, on page 31
- Creating a New Bounce Profile, on page 37
- Applying Bounce Profiles to Listeners, on page 37

Handling Undeliverable Email

The 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 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 Using Message Filters to Enforce Email Policiesfor more information.)

Related Topics

- Notes on Soft and Hard Bounces, on page 31
- Bounce Profile Parameters, on page 32
- Hard Bounces and the status Command, on page 35
- Conversational Bounces and SMTP Routes Message Filter actions, on page 35
- Example Bounce Profiles, on page 35
- Delivery Status Notification Format, on page 36
- Delay Warning Messages, on page 36
- Delay Warning Messages and Hard Bounces, on page 36

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, on page 32.)

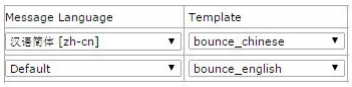
- By default, the system generates a bounce message and sends it to the original sender for each hard bounced recipient. (The message is sent to the address defined in the Envelope Sender address of the message envelope. Envelope From is also commonly referred to as the Envelope Sender.) You can disable this feature and instead rely on log files for information about hard bounces. (See the "Logging" chapter.)
- Soft bounces become hard bounces after the maximum time in queue or the maximum number of retries, whichever comes first.

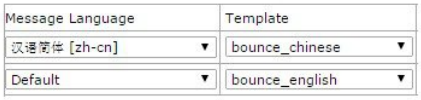
Bounce Profile Parameters

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

Table 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.

Send Hard Bounce Messages	Specify whether to send bounce message for hard bounce. If this option is enabled, you can choose the format of the bounce message. By default, bounce messages use the DSN format (RFC 1894). You can also send customized bounce messages based on the language of the original message (subject and body). For example, you may want to send bounce messages in Chinese for messages in Chinese and bounce messages in English for all the messages in other languages. Under **Notification Template**, click **Add Row** and choose the message language and the template that you want to use. **Note** Make sure that you do not delete the default entry (**Message Language** set to **Default**). You can change the bounce notification template for the default entry. The language of a message is considered Default in the following scenarios: • If the language of the message is different from the language selected in the other Notification Template entries. • If the language of the message is not supported by the email gateway. • If the email gateway is unable to detect the language of the message. • If the content (subject and body) in the message is less than 50 bytes. While configuring the above example (send bounce messages in Chinese for messages in Chinese and bounce messages in English for all the messages in other languages,) the Notification Template table will look like this:  You can also choose whether to parse the DSN status field from the bounce response. If you choose “Yes,” the email gateway 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 to send warning message for delayed delivery. If this option is enabled, you can configure custom delay warning messages based on the language of the original message (subject and body). For example, you may want to send delay warning messages in Chinese for the messages in Chinese and delay warning messages in English for all the messages in other languages. Under **Notification Template**, click **Add Row** and choose the message language and the template that you want to use. **Note** Make sure that you do not delete the default entry (**Message Language** set to **Default**). You can change the bounce notification template for the default entry. The language of a message is considered Default in the following scenarios: • If the language of the message is different from the language selected in the other Notification Template entries. • If the language of the message is not supported by the email gateway. • If the email gateway is unable to detect the language of the message. • If the content (subject and body) in the message is less than 50 bytes. While configuring the above example (send delay warning messages in Chinese for the messages in Chinese and delay warning messages in English for all the messages in other languages,) the Notification Template table will look like this:  You can also specify the minimum interval between messages as well as the maximum number of retries to send.
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.
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.	
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 email gateway 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 the "Monitoring and Managing via the CLI" chapter. 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. 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 email gateway as a result of a conversational bounce. When an email gateway receives a conversational bounce message, it generates an SMTP bounce message back to the Envelope Sender of the original message. In this case, the email gateway 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 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 email gateway. 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 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 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 email gateway 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.

Related Topics

- Editing the Default Bounce Profile, on page 37
- Example of a Minimalist Bounce Profile, on page 37

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):

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.

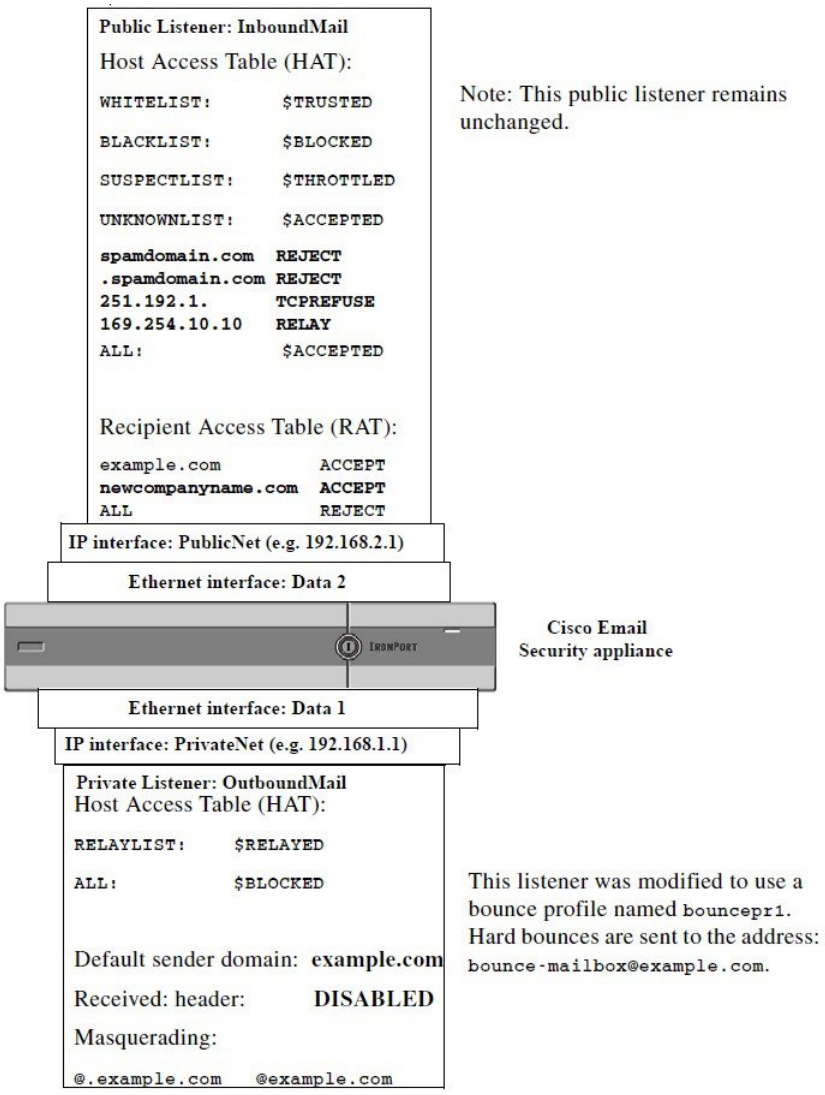
Applying Bounce Profiles to Listeners

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.

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

Figure 5: Applying a Bounce Profile to a Private Listener



Controlling Email Delivery Using Destination Controls

Uncontrolled high-volume email delivery can overwhelm recipient domains. AsyncOS gives you full control of message delivery by defining the number of connections your email gateway will open or the number of messages your email gateway 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, on page 39
- TLS, on page 39
- Bounce Verification, on page 39
- Bounce Profile, on page 39

Rate Limiting

- Concurrent Connections: number of simultaneous connections to remote hosts the email gateway will attempt to open.
- Maximum Messages Per Connection: number of messages your email gateway will send to a destination domain before the email gateway initiates a new connection.
- Recipients: number of recipients the email gateway 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, on page 42).
- 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 Bounce Verification (see Bounce Verification, on page 47).

Bounce Profile

- Which bounce profile should be used by the email gateway 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.

Related Topics

- Determining Which Interface is Used for Mail Delivery, on page 39
- Default Delivery Limits, on page 40
- Working with Destination Controls, on page 40

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.

Related Topics

- Controlling the Version of Internet Protocol Addresses, on page 40
- Controlling the Number of Connections, Messages, and Recipients to a Domain, on page 40
- Controlling TLS, on page 42
- Controlling Bounce Verification Tagging, on page 43
- Controlling Bounces, on page 43
- Adding a New Destination Control Entry, on page 43
- Importing and Exporting Destination Control Configurations, on page 43
- Destination Controls and the CLI, on page 46

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 gateway uses both Internet Protocol version 4 (IPv4) and Internet Protocol version (IPv6). You can configure a listener on the email gateway to use one version of the protocol or both.

If the “Required” setting for either IPv4 or IPv6 is specified, the email gateway will negotiation a connection to the domain using an address of the specified version. If the domain does not use that IP address version, no email will be sent. If the “Preferred” setting for either IPv4 or IPv6 is specified, the email gateway will first attempt to negotiation 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 email gateway.

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 email gateway will allow.)

You also set whether the limits you define are enforced for the entire domain.

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

These values are explained in the following table.

Table 8: Values in the Destination Controls Table

Field	Description
Concurrent Connections	The maximum number of outbound connections that will be made by the email gateway 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 email gateway 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 email gateway 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 email gateway delivers the messages based on the new recipient limit.
Apply Limits	Specifies whether the limit will be applied (enforces) to the entire domain. 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 Configuring Mail Gateways for all Hosted Domains Using Virtual Gateway™ Technology, on page 53.

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.

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 email gateway 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.

You can specify whether the email gateway 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 email gateway 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 email gateway 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. You can edit the ‘Default' destination controls entry in the web interface or use the `destconfig` > `setup` subcommand in the CLI to specify the certificate to use for all destination controls.. For information on obtaining a certificate, see Working with Certificates.

You can also choose a different certificate other than the certificate configured in the ‘Default' destination control entry for specific domains.

You can choose a different certificate in any one of the following ways:

- Edit the corresponding destination control entry and select a different certificate using the **TLS certificate** option in the web interface.
- Use the `destconfig` > `new` or `edit` sub commands to select a certificate when you create or edit a destination control entry.

Note You can create a maximum of 100 destination control entries with a different certificate than the certificate configured in the 'Default' destination control entry.

For more information on alerts, see the “System Administration” chapter.

Controlling 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 suggests enabling bounce verification for the default, and then creating new destinations for specific exclusions. See Bounce Verification, on page 47for 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, on page 37.

Adding a New Destination Control Entry

Procedure

Step 1 Click **Add Destination**:

Step 2 Configure the entry.

Step 3 Submit and commit your changes.

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 email gateway. 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 email gateway'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 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”

Parameter Name	Description
max_host_concurrency	The maximum number of outbound connections that will be made by the email gateway 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 email gateway to a given host before initiating a new connection.
recipient_minutes	The period of time — between 1 and 60 minutes — that the email gateway will count the number of recipients. Leave undefined if no recipient limit should be applied.
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 Deliveryfor 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.

Parameter Name	Description
`bounce_profile`	Name of the bounce profile to use. This cannot be used in the `[DEFAULT]` destination control entry.
`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. If you do not specify a certificate, then the certificate in the `[DEFAULT]` destination control entry is used. **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]

certificate = 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)

TLS Certificate: example.com

Bounce Profile: tls_failed


example2.com


IP Address Preference: IPv6 Preferred

Maximum messages per connection: Default

Rate Limiting: Default

TLS: Preferred

TLS Certificate: example2.com

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 CLI Reference Guide for AsyncOS for Cisco Secure Email Gateway.

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 Bounce Verification. When enabled, Bounce Verification tags the Envelope Sender address for messages sent via your email gateway. The Envelope Recipient for any bounce message received by the email gateway 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 Bounce Verification to manage incoming bounce messages based on your outgoing mail. To control how your email gateway generates outgoing bounces (based on incoming mail), see Directing Bounced Email, on page 30.

Related Topics

- Overview: Tagging and Bounce Verification, on page 47
- Preventing a Bounced Message Storm Using Bounce Verification, on page 49
- Accepting Legitimate Untagged Bounced Messages, on page 48

Overview: Tagging and Bounce Verification

When sending email with bounce verification enabled, your email gateway 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 email gateway. The tag is generated using a key defined in the Bounce Verification settings (see Bounce Verification Address Tagging Keys, on page 48for 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, on page 40).

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

Related Topics

- Handling Incoming Bounce Messages, on page 48
- Bounce Verification Address Tagging Keys, on page 48

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 email gateways handle untagged or invalidly tagged bounces — reject them or add a custom header. See Configuring Bounce Verification Settings, on page 50 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 Bounce Verification.

For example, the following mail log shows a bounced message rejected by the email gateway:

```
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 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.

Bounce Verification Address Tagging Keys

The tagging key is a text string your email gateway uses when generating the bounce verification tag. Ideally, you would use the same key across all of your email gateways so that all mail leaving your domain is tagged consistently. That way, if one email gateway 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 email gateway.

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 email gateway will try to verify tagged messages using all previous keys that are less than seven days old.

Accepting Legitimate Untagged Bounced Messages

AsyncOS also includes a HAT setting related to Bounce Verification for considering whether untagged bounces are valid. The default setting is “No,” which means that untagged bounces are considered invalid and the email gateway 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 email gateway 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).

Procedure

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.

Preventing a Bounced Message Storm Using Bounce Verification

Procedure

Step 1 Enter a tagging key. For more information, see Configuring Bounce Verification Address Tagging Keys, on page 49.

Step 2 Edit the bounce verification settings. For more information, see Configuring Bounce Verification Settings, on page 50.

Step 3 Enable bounce verification via Destination Controls. For more information, see Working with Destination Controls, on page 40.

What to do next

Related Topics

- Configuring Bounce Verification Address Tagging Keys, on page 49
- Configuring Bounce Verification Settings, on page 50
- Configuring Bounce Verification Using the CLI, on page 50
- Bounce Verification and Cluster Configuration, on page 50

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:

Procedure

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.

What to do next

Related Topics

- Purging Keys, on page 50

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 Bounce Verification Settings

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

Procedure

Step 1 Choose **Mail Policies > Bounce Verification**.

Step 2 Click **Edit Settings**.

Step 3 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 4 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 5 Submit and commit your changes.

Configuring Bounce Verification Using the CLI

You can use the bvconfig and destconfig commands in the CLI to configure bounce verification. These commands are discussed in the *CLI Reference Guide for AsyncOS for Cisco Secure Email Gateway* .

Bounce Verification and Cluster Configuration

Bounce verification works in a cluster configuration as long as both email gateways 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 email gateway.

Set Email Delivery Parameters

The deliveryconfig command sets parameters to be used when delivering email from the email gateway.

The email gateway 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 the "Assigning Network and IP Addresses" appendix for more information.

Related Topics

- Default Delivery IP Interface, on page 51
- Possible Delivery Feature, on page 51
- Default Maximum Concurrency, on page 51
- deliveryconfig Example, on page 52

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

Caution If you enable this feature, message delivery will not be reliable and may lead to loss of messages. Also, your email gateway will not be RFC 5321-compliant. For more information, see http://tools.ietf.org/html/rfc5321#section-6.1..

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 acknowledgment of receipt. AsyncOS logs this recipient as a possible delivery in the mail logs and counts the message as completed.

Default Maximum Concurrency

You also specify the default maximum number of concurrent connections the email gateway 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 gateway from dominating weaker networks. For example, certain

firewalls do not support large numbers of connections, and this could induce Denial of Service (DoS) warnings in these environments.

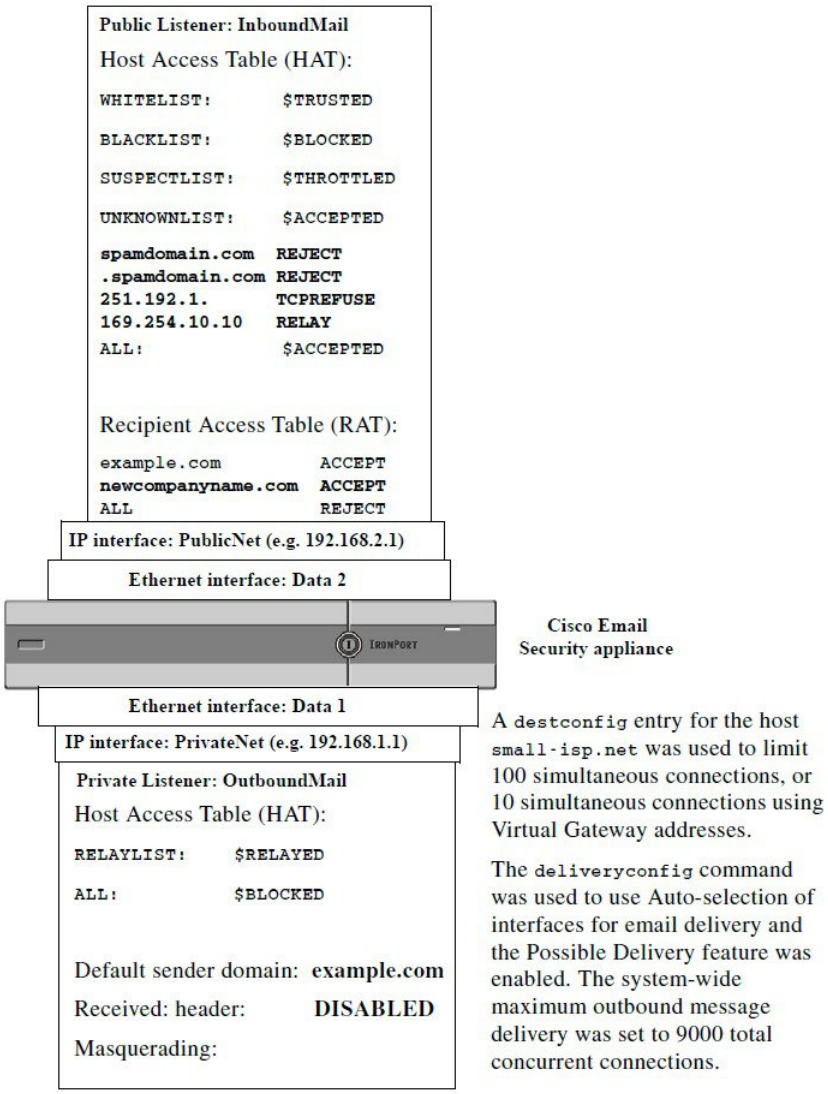
deliveryconfig Example

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"? [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 6: Setting Destination and Delivery Parameters



Configuring Mail Gateways for all Hosted Domains Using Virtual Gateway™ Technology

This section describes Cisco 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 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 email gateway. The number of Virtual Gateway addresses available on all the email gateway models is 255.

Related Topics

- Overview, on page 54
- Setting Up Virtual Gateway Addresses, on page 54
- Monitoring the Virtual Gateway Addresses, on page 61
- Managing Delivery Connections per Virtual Gateway Address, on page 61

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 email gateway 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 email gateway 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 email gateway 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 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 Virtual Gateway addresses, you must allocate a set of IP addresses that will be used to send email from. (For more information, see the “Assigning Network and IP Addresses” appendix.) 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.

Related Topics

- Creating New IP Interfaces for Use with Virtual Gateways, on page 55
- Mapping Messages to IP Interfaces for Delivery, on page 57
- Importing an altsrchost File, on page 58
- altsrchost Limits, on page 58
- Example Text File with Valid Mappings for the altsrchost Command, on page 58
- Adding an altsrchost Mapping through the CLI, on page 59

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.

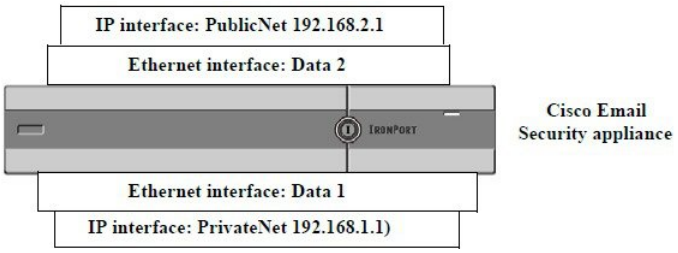
After creating the required IP interfaces, you have two options for setting up the Virtual Gateway addresses and defining which email campaign will be sent from each IP interface or interface group:

- 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.
- 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. (This method is more flexible and powerful than the one above.)

For more information about creating IP interfaces, see the “Accessing the Email Gateway” appendix.

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

Figure 7: 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 8: IP Interface Page

IP Interfaces

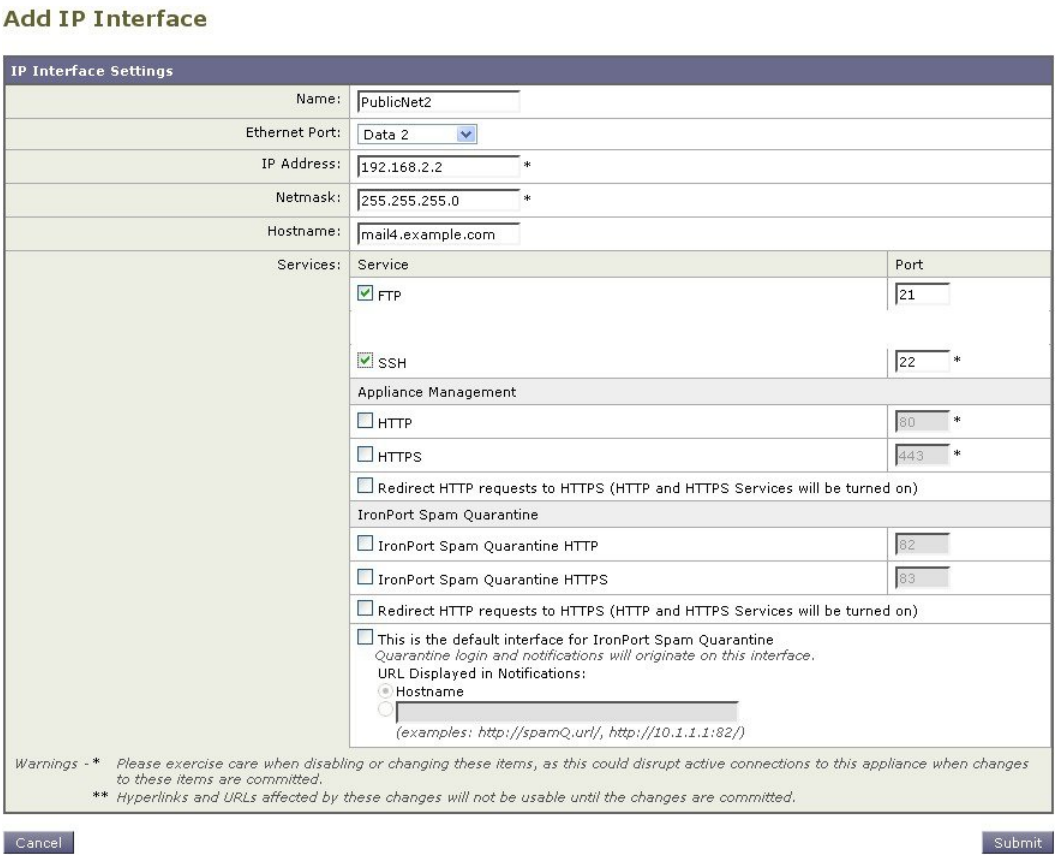
Network Interfaces and IP Addresses

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	

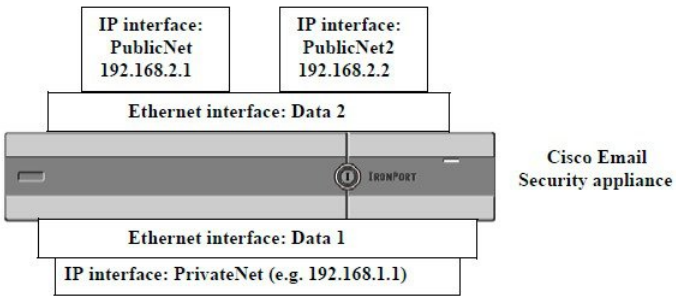
Next, the Add IP Interface page is used to create a new interface named PublicNet2 on the Data2 Ethernet interface. The IP address of 192.168.2.2 is used, and the hostname of mail4.example.com is specified. The services for FTP (port 21) and SSH (port 22) are then enabled.

Figure 9: Add IP Interface Page



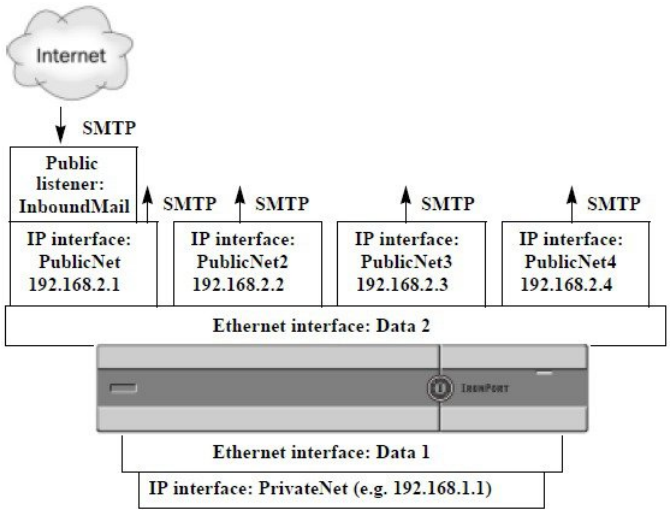
Our Email Gateway configuration now looks like this:

Figure 10: Adding Another Public Interface



Using Virtual Gateway addresses, a configuration like the one shown in the following figure is also possible.

Figure 11: 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 email gateway 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 Using Message Filters to Enforce Email Policiesfor 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).

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.

Procedure

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 FTP, SSH, and SCP Access 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 FTP, SSH, and SCP Accessfor 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.

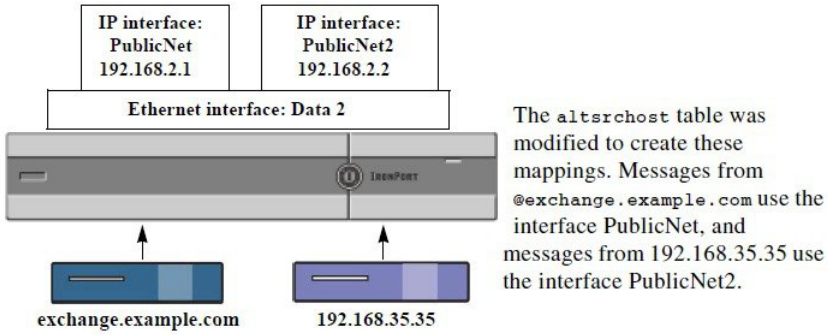
```
mail3.example.com> altsrchost

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

Do you want to save the current configuration for rollback? [Y]> n

Changes committed: Fri May 23 11:42:12 2014 GMT
```

An illustration of the configuration change in this example is shown in the following figure:

Figure 12: Example: Selecting an IP Interface or Interface Group to Use



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 the “Reading the Available Components of Monitoring” section in the “Managing and Monitoring Using the CLI” chapter.

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 Using Destination Controls, on page 38for 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 email gateway, use the 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.

Global Unsubscribe has a maximum limit of 10,000 addresses. Global Unsubscribe addresses can be in one of four forms:

Table 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 (@).

@.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.

Related Topics

- Adding a Global Unsubscribe Address Using The CLI, on page 63
- Exporting and Importing a Global Unsubscribe File, on page 64

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 email gateway 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
Do you want to save the current configuration for rollback? [Y]> n
Changes committed: Fri May 23 11:42:12 2014 GMT
```

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.

Procedure

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 FTP, SSH, and SCP Access 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 FTP, SSH, and SCP Access for more information.)

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

Review: Email Pipeline

The following tables 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 *Table - Email Pipeline for the Cisco Secure Email Gateway: Routing and Delivery Features* 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 Troubleshooting chapter.

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

Table 11: Email Pipeline for the Cisco Secure Email Gateway: 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 12: Email Pipeline for the Email Security 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.
	Advanced Malware Protection		Advanced Malware Protection performs file reputation scanning and file analysis, in order to detect malware in attachments.
	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.
	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.