PreciseMail Anti-Spam Gateway Management Guide, UNIX Edition

September 2019

This manual describes the configuration for the PreciseMail Anti-Spam Gateway product.

Revision/Update Information:	This is a revised manual.
Operating System and Version:	RedHat Linux 7.2 or later
PMDF Version:	PMDF V6.8 or later
Software Version:	PreciseMail Anti-Spam Gateway V3.3

Process Software

13 September 2019

Process Software, LLC
959 Concord Street
Framingham, MA 01701-4682 USA
Voice: +1 508 879 6994; FAX: +1 508 879 0042
info@process.com

Process Software, LLC (‘‘Process’’) makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Process Software reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Process Software to notify any person of such revision or changes.

Contents

EXAMPLES

TABLES

Preface

This guide describes how to manage PreciseMail Anti-Spam Gateway.

Intended Audience

This manual is intended for use by the system manager or any individual responsible for configuring and maintaining the PreciseMail Anti-Spam Gateway anti-spam product.

Document Structure

This guide consists of five chapters and one appendix.

Chapter 1 Contains an overview of how PreciseMail Anti-Spam Gateway works.

Chapter 2 Describes the PreciseMail Anti-Spam Gateway configuration file.

Chapter 3 Describes the PreciseMail Anti-Spam Gateway rules and writing new rules.

Chapter 4 Describes the PreciseMail Anti-Spam Gateway batch jobs.

Chapter 5 Describes additional programs used by the PreciseMail Anti-Spam Gateway.

Chapter 6 Describes the PreciseMail Anti-Spam Gateway user interface.

Chapter 7 Describes the PreciseMail Anti-Spam Gateway Data Synchronization Clusters.

Chapter 8 Describes the PreciseMail Anti-Spam Gateway anti-virus scanning.

Chapter 9 Describes how to debug PreciseMail Anti-Spam Gateway.

Appendix A Contains a list of the files created by an installation.

Related Documents

You can find additional information in the following documents:

- *PreciseMail Anti-Spam Gateway Installation Guide* describes the PreciseMail Anti-Spam Gateway installation procedure.
- *PreciseMail Anti-Spam Gateway User's Guide* describes the PreciseMail Anti-Spam Gateway user interface.
- *PreciseMail Anti-Spam Gateway Release Notes* contain information and updates not included in this manual. The release notes are part of the software distribution kit.

Open Source Software used by PreciseMail Anti-Spam Gateway

PreciseMail Anti-Spam Gateway includes parts of or makes use of the following open source software packages:

- Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. The software can be found on this FTP and web site:

  ```
  ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/
  http://www.pcre.org/
  ```

- Some of the original regular expression rules used to detect spam were borrowed from the SpamAssassin package.
- PreciseMail Anti-Spam Gateway links against OpenSSL as a shared library under the terms of the OpenSSL license.
- PreciseMail Anti-Spam Gateway links against OpenLDAP as a shared library under the terms of the OpenLDAP license.
- ISAM file support is provided on UNIX by the OpenSource PBL library.
- The Info-ZIP UNZIP utility is used in the automatic rule update process. The Info-ZIP license can be found using this URL:

  ```
  http://www.info-zip.org/license.html
  ```

- The PMAS GUI uses the Thickbox and JQuery packages for handling popup windows. Thickbox was written by Cody Lindley, and JQuery was written by John Resig.

  ```
  http://jquery.com/
  http://jquery.com/demo/thickbox/
  ```

The distributions of the source and associated licenses for some of these packages is available at ftp.pmas.process.com. The license texts can also by found in the directory /pmas/doc/3rdparty.

Notwithstanding the foregoing, PreciseMail Anti-Spam Gateway is not open source software. You must purchase a license for each copy of PreciseMail Anti-Spam Gateway that you install.

1 Overview of PreciseMail Anti-Spam Gateway Operation

PreciseMail Anti-Spam Gateway is a high-performance anti-spam solution based on proven heuristic and artificial intelligence technologies. It eliminates spam at the Internet gateway without filtering critical legitimate email messages, producing a large potential cost savings to organizations of all sizes. PreciseMail Anti-Spam Gateway has the unique ability to allow many levels of tuning and customization to meet an individual site's email spam filtering requirements.

Every site's definition of spam is unique, so PreciseMail Anti-Spam Gateway provides the highest level of flexibility as compared to other products. Systems administrators can adjust the weight of each Process Software-supplied spam definition, as well as add new spam definitions or modify existing ones. A combination of rules can trigger a positive score, which indicates an email message is spam. For example, the total score of the phrases “low interest rates” and “click below” would indicate that a message is spam. Systems administrators can set the threshold beyond which a message is considered spam.

Process Software provides customers with regular spam definition updates that can be downloaded and installed at the customer's discretion. There is no need for a site to disclose any information about their mail system or open their firewall to install updates.

In parallel to the spam definition rules, PreciseMail Anti-Spam Gateway uses an artificially intelligent Bayesian engine to identify spam. The Bayesian engine is automatically “taught” how to separate spam messages from your site's legitimate email. By combining heuristic and artificial intelligence technologies, PreciseMail Anti-Spam Gateway is able to accurately identify spam while making it extremely difficult for spammers to circumvent the filters.

When a message is identified as spam, it is quarantined until further review by the recipient. At a set interval (usually twice daily), users are automatically notified via email with a summary of their quarantined messages. At this point, recipients can choose to release quarantined messages that they wish to receive. Email recipients can also use this automated process to set up individual allow and block lists. Since users control their own quarantined messages, there is no need for system administrators to spend time reviewing thousands of quarantined emails to identify potentially legitimate mail.

PreciseMail Anti-Spam Gateway requires minimal configuration and deployment time, making it an ideal “drop-in” solution for spam filtering. The product integrates seamlessly with PMDF and Sendmail. With the addition of the Pass-Through SMTP proxy server, PreciseMail Anti-Spam Gateway can be used to protect any existing mail system.

Key Features:

- Extensive heuristic rule set used to perform message header and content analysis
- Artificially intelligent Bayesian engine can be used in parallel with the heuristic analysis for increased accuracy and reduced false positive rate
- Individual user-controlled allow and block lists
- Administrator-controlled allow and block lists
- User review and retrieval of quarantined mail
- Customizable filters
- Detailed reports on filtered mail
- Optimized to run with PMDF and SendMail
- Can be used as a front-end to any existing SMTP server

1.1 PreciseMail Anti-Spam Gateway Directory Overview

The PreciseMail Anti-Spam Gateway product files reside in a special directory tree, /pmas on UNIX. The following directories exist in the tree and contain the described files:

- bin/ - The PreciseMail Anti-Spam Gateway executable images
- com/ - Script files
- data/ - The configuration, rule, and score files
- discard/ - Copies of messages that have been discarded
- doc/ - Documentation files
- help/ - Help files
- html/ - HTML template files for the user interface
- log/ - Log files produced by PreciseMail Anti-Spam Gateway
- quarantine/ - Copies of messages that have been quarantined
- user_rules/ - User allow list and block list files
- www/ - Static HTML files and CGI scripts for the user interface

On UNIX, the executable images reside in the /pmas/bin directory.

1.2 PMDF and PreciseMail Anti-Spam Gateway

PreciseMail Anti-Spam Gateway is integrated with Process Software’s PMDF product, allowing PreciseMail Anti-Spam Gateway to be easily added to existing PMDF installations. PreciseMail Anti-Spam Gateway includes a new PMDF channel called “pmas”.

Messages are routed from PMDF to PreciseMail Anti-Spam Gateway by means of a CONVERSIONS or SCRIPT entry in the mappings file. (The mappings file is named /pmdf/table/mappings in PMDF.) (The SCRIPT entry is only available in versions of PMDF after V6.2 or in PMDF V6.2 via a downloadable ECO.) The special CHANNEL= keyword is used to specify the "pmas" channel. A typical entry will look like this:

```
CONVERSIONS

   IN-CHAN=tcp_*;OUT-CHAN=l;CONVERT  CHANNEL=pmas,Yes
   IN-CHAN=*;OUT-CHAN=*;CONVERT   No
```

In this example, all messages coming in from SMTP channels ("tcp_*") and destined for local UNIX users (delivered via the "l" channel) will first be routed to the pmas channel.

If you already have a CONVERSIONS entry, you must decide if you want messages routed to the pmas channel before or after they're processed by the conversion channel. This is accomplished by specifying the proper values for IN-CHAN and OUT-CHAN. To have messages routed to the conversion channel first, then to the pmas channel, lines like the following would be used:

```
  IN-CHAN=tcp_*;OUT-CHAN=l;CONVERT          Yes
  IN-CHAN=conversion;OUT-CHAN=l;CONVERT     CHANNEL=pmas,Yes
```

or to have PreciseMail Anti-Spam Gateway run first, use lines similar to these:

```
  IN-CHAN=tcp_*;OUT-CHAN=l;CONVERT          CHANNEL=pmas,Yes
  IN-CHAN=pmas;OUT-CHAN=l;CONVERT           Yes
```

If you're using the SCRIPT channel entries, you have the option of restricting which messages are processed by PreciseMail Anti-Spam Gateway based on the message size. The *MAXBLOCKS* and *MAXLINES* keywords are used to restrict the sizes of messages that will be processed by PreciseMail Anti-Spam Gateway. Since spam messages are typically small messages, these keywords can be used to prevent needless processing of large messages that aren't likely to be spam.

```
SCRIPT

  IN-CHAN=tcp_*;OUT-CHAN=l;SCRIPT   CHANNEL=pmas,maxblocks=200,maxlines=2000,yes
  IN-CHAN=*;OUT-CHAN=*;SCRIPT       No
```

Note: PMDF's Sieve filters (both user filters and the system filter) are applied to messages before they're handed off to the pmas channel for processing by PreciseMail Anti-Spam Gateway. If the Sieve filters discard a message, it will not be processed by PreciseMail Anti-Spam Gateway. The Sieve filters are also applied again after PreciseMail Anti-Spam Gateway has processed the message, allowing users to do such things as file messages in "spam suspect" folders based on the presence of X-PMAS headers or a modified subject line.

1.3 SendMail and PreciseMail Anti-Spam Gateway

PreciseMail Anti-Spam Gateway is integrated with the open source SendMail MTA through the use of a milter named “pmas_milter”.

You must be running SendMail 8.12.x or greater, with milter functionality enabled. You can determine if your installation of SendMail supports milters by running the command:

```
  /usr/lib/sendmail -bt -d0.4 < /dev/null
```

If milters are supported, the token “MILTER” will appear in the output. If not, you need to recompile SendMail with milter functionality enabled. Consult your SendMail documentation for more information.

Note: **If you are running Solaris 8 or higher with a recent patchset, the Sun-supplied version of SendMail supports milters.**

If you are using the custom version of SendMail that Sun supplies with the Solaris operating system, edit the /etc/mail/sendmail.cf file itself. Near the top of the file, add an InputMailFilters option for pmas_milter:

```
O InputMailFilters=/pmas/bin/pmas_milter
```

Just above the MAILER DEFINITIONS block of the sendmail.cf file, add the following line to specify the milter options:

```
X/pmas/bin/pmas_milter, S=local:/pmas/tmp/pmas.sock, F=T,T=C:90s;S:90s;R:90s;E:90s
```

If you are using a standard SendMail binary, edit the sendmail.mc file located in the sendmail-8.12.x/cf/cf directory of your source distribution. Add the following INPUT_MAIL_FILTER macro to the bottom of the file:

```
INPUT_MAIL_FILTER(‘/pmas/bin/pmas_milter’, ‘S=local:/pmas/tmp/pmas.sock, F=T, T=C:90s;S:90s;R:90s;E:90s’)
```

Save the sendmail.mc file, then run the following command to generate a new sendmail.cf file and install it in /etc/mail:

```
# make install-cf
```

For more information about the SendMail configuration directives used by PreciseMail Anti-Spam Gateway, see the PreciseMail Anti-Spam Gateway Installation Guide for your platform.

1.4 The PMAS Pass-Through SMTP Server

To run the PMAS Pass-Through SMTP Server (PTSMTP), you must define the appropriate PMAS configuration variables as described in Section 2.2.8, Pass-Through SMTP Server keywords. There are three additional steps in configuring PMAS PTSMTP.

1.4.1 MX records

If you choose to run PMAS PTSMTP on a separate system from the one on which your primary SMTP server is running, you will need to modify the DNS MX records for your domain name to point to the system that's running the PMAS PTSMTP server. Details on doing this are beyond the scope of this guide; please consult the documentation for your TCP/IP software for details on changing the MX record.

1.4.2 Identifying internal IP addresses

PreciseMail Anti-Spam Gateway adds a "Received:" header to every message that passes through the PTSMTP server. This header will resemble the following example:

```
Received: from pc.example.com❶ ([123.45.67.89]❷ EXTERNAL❸)
 (EHLO pc.example.com)❹ by alpha.example.com❺ ([10.10.10.10])❻
 (PreciseMail V2.1);❼ Tue, 24 Aug 2004 09:30:06 -0500❽
```

The following pieces of information are available in the "Received:" header that PMAS adds:

❶ The host name of the sending system (obtained by doing a reverse DNS lookup using the IP address of the system establishing the SMTP connection)

❷ The actual IP address of the sending system

❸ Whether the system is "EXTERNAL" or "INTERNAL"

❹ The "HELO" or "EHLO" command with which the sending system initiated the connection

❺ The host name of the receiving system

❻ The IP address of the receiving system

❼ The version of PreciseMail that added the header

❽ The date the header was added

The choice of "INTERNAL" or "EXTERNAL" for the IP address is determined based on the contents of the file /pmas/data/internal_ip.txt. This file should list all of the IP addresses and/or subnets that should be considered and marked as "INTERNAL". Each line of the file should specify either an IP address or a CIDR-format subnet IP address, which is a four-octet address followed by a slash and the number of significant bits:

```
#  Each line should specify either an IP address or a CIDR-format IP
#  address, a slash, and the number of significant bits:
#
110.10.10.10
#
#  The high 29 bits are significant (so IP addresses .40 through .47
#  are considered internal).
#
110.10.10.40/29
```

Once internal systems are identified, an Allow_Regex rule can be added to your system-wide allow and block list (described in Section 1.5.2) to automatically allow messages originating internally to bypass being scanned by PMAS:

```
Allow_Regex  Received: from \S+ \(\[(?:\d+\.){3}\d+\] INTERNAL\).*
```

In addition to the "Received:" header, PMAS PTSMTP now adds a separate header that indicates whether or not the source was internal or external. The header will be named either "X-PMAS-Internal:" or "X-PMAS-External:", depending on whether or not the sending IP address is listed in INTERNAL_IP.TXT. (Note that the "PMAS" part of the header name is actually the value of the configuration variable HEADER_PREFIX.) The format of the line is:

```
X-PMAS-Internal: system-name [IP-address] (HELO/EHLO helo-text)
```

The "system-name" is the name returned by DNS for the connecting IP address. The "[IP-address]" is the dotted-decimal, numeric IP address of the sending system. The "(HELO helo-text)" reflects the actual HELO or EHLO line used during the SMTP dialogue. As defined by RFC 2821, the HELO/EHLO parameter is supposed to be the name of the sending system, but many spammers will incorrectly use your host's system name or IP address, providing another opportunity to block spam or score it appropriately.

An alternative to the Allow_Regex rule offered above is:

```
rule allow header:X-PMAS-External noexists
```

That will allow any message that does not have an X-PMAS-External: header, which would be any message from an internal system. This test is safer than checking for the existence of X-PMAS-Internal: as a spammer could provide a fake X-PMAS-Internal: header.

1.4.3 Enabling PMDF XREM Support

One of the problems PMDF sites will have if they deploy the PMAS PTSMTP proxy server is that doing so will invalidate PORT_ACCESS and other rules in PMDF's MAPPINGS file. That's because all connections coming from the PMAS PTSMTP server will appear to PMDF to be from the local system (or whatever system is running PMAS), so tests to, for example, reject messages based on incoming tcp channel are ineffective.

To address this problem, PMAS and PMDF support a Process Software-created SMTP extension called XREM. When XREM is enabled in both PMAS and PMDF, the PMAS PTSMTP proxy server accepts the incoming connection and, once it has connected to the PMDF SMTP server on the backend, tells PMDF what the original remote IP address and port are via the new XREM command. PMDF then treats the connection as if it originated from that remote IP address and port instead of from PMAS. That means that mail that comes in on tcp_local will continue to come from tcp_local, as far as PMDF is concerned. PORT_ACCESS rules and other rules can be used as they've always been used to restrict access, etc.

PMDF sites that use CONVERSIONS or SCRIPT rules to determine which messages get scanned by PMAS may not be able to take advantage of this feature, as by definition, PMAS is run on all non-opted-out messages before the message is ever handed to PMDF.

To enable PMDF XREM support in PMAS, simply set the configuration variable PTSMTP_ENABLE_PMDF_XREM to “yes”.

To enable PMDF XREM support in PMDF V6.5, you’ll need PMDF V6.6 or later, or PMDF V6.5 with the new images (found in the [.PMDF065-XREM] subdirectory of the PMAS V3.2 download directory) installed (PMDF should be restarted afterward).

There is a new option to put in the /pmdf/table/tcp_local_option file, ENABLE_XREM, which will list the ip addresses that are allowed to use the XREM command, separated by commas, for example:

```
ENABLE_XREM=127.0.0.1,10.5.64.1,10.24.25.2,10.4.1.3
```

A detailed discussion of how this affects PMDF log files can be found below.

Assume a connection comes in to PMDF from PMAS, which is running on node 10.5.64.119:

```
08-Jul-2010 15:48:42.20 573e.3.0 tcp_local    +            O TCP|10.5.64.17|25|10.5.64.119|1404 SMTP
```

PMAS passes the XREM command to the PMDF SMTP server with updated info (say, from remote ip 10.24.25.1 and port 999), and a new entry (using "P" for PMAS) is written to the connection log:

```
08-Jul-2010 15:48:42:30 573e.3.1 tcp_local    +            P TCP|10.5.64.17|25|10.24.25.1|999 SMTP
```

and then when the connection is closed, use the new info:

```
08-Jul-2010 15:48:42.57 573e.3.1 tcp_local    +            C TCP|10.5.64.17|25|10.24.25.1|999 SMTP
```

Customers searching the log file for connections can use the “P” and “C” entries to see what remote systems connected and when.

When the PMDF XREM command is received by the PMDF SMTP server, it will check PORT_ACCESS, and if the new remote ip address and port should be rejected, it will respond to XREM with a 500-level error response and then kill the connection.

For example:

```
PORT_ACCESS

  TCP|*|25|10.5.64.1|*          $N$<$TPORT_ACCESS:$ reject$ after$ XREM|\
PORT_ACCESS:$ Connect$ Not$ Allowed
```

With that mapping entry, PMDF would respond to the XREM command with his reply:

```
501 5.5.7 Connection rejected by PORT_ACCESS: PORT_ACCESS: Connect Not Allowed
```

The “$T” and “$<” flags will also be executed (“$T” causes a ’T’ entry in connection.log_current, and “$<” causes a syslog/opcom message).

There will also be an ’X’ entry in the connection.log_current:

```
12-Jul-2010 15:28:35.96 6dae.3.0 tcp_local    +           O TCP|198.115.140.17|25|192.42.95.56|32921 SMTP
12-Jul-2010 15:28:41.46 6dae.3.1 tcp_local    +           P TCP|198.115.140.17|25|10.5.64.1|999 SMTP
12-Jul-2010 15:28:41.46 6dae.3.2 **           +           T TCP|198.115.140.17|25|10.5.64.1|999 PORT_ACCES
12-Jul-2010 15:28:41.47 6dae.3.3 tcp_local    +           X TCP|198.115.140.17|25|10.5.64.1|999 SMTP
```

1.4.4 Enabling SMTP-Over-TLS Support

If you want the PTSMTP Server to support SMTP-over-TLS, you will need to create TLS certificates. If you do not already have a set of TLS certificates for the system that the PTSMTP Server is running on, you can use the /pmas/bin/tls_certreq utility to create them.

The PTSMTP Server expects the public certificate to be named /pmas/data/server-pub.pem and the private key to be named /pmas/data/server-priv.pem . These are the default locations of the certificate files generated by the tls_certreq utility.

The filenames can be overridden by defining the configuration variables PTSMTP_TLS_PUBLIC_CERT and PTSMTP_TLS_PRIVATE_CERT.

Once the TLS certificates are in place, enable the PTSMTP Server's TLS support by specifying values for the ptsmtp_listen_port_tls, ptsmtp_mailserver_host_tls, and ptsmtp_mailserver_port_tls configuration variables.

TLS support over the normal SMTP port (25) is provided using the STARTTLS command in the SMTP dialogue. To enable STARTTLS support in the PTSMTP server, define the configuration variable *ptsmtp_enable_starttls* as "yes".

1.4.5 DNSBL "DNS Blackhole List" Support

DNSBL (DNS-based Blackhole List) servers maintain lists of IP addresses of known spam systems and open relays. The PMAS PTSMTP proxy server includes support for accessing DNSBL systems to identify incoming email that is being sent by one of these known spam systems. If the sending system is listed in the DNSBL, the message can be rejected during the SMTP session, or a special header can be added to the message's RFC822 headers.

DNSBL lookups work by taking the IP address of the sending system, reversing the order of the address and appending it to the name of the DNSBL system, and performing a DNS (Domain Name System) query. For example, to query the SpamHaus DNSBL for IP address 168.10.20.30, a DNS query is conducted using the system name "30.20.10.168.sbl.spamhaus.org". The response that comes back either indicates that the system is not listed, or a special value in the loopback address range is returned (for example, 127.0.0.2). Different sites return different addresses, which sometimes indicate why a particular site is listed in the DNSBL.

Note:

For information on how DNSBL lookups are implemented and the history of DNSBLs, please see the Wikipedia description:

```
http://en.wikipedia.org/wiki/DNSBL
```

A myriad of DNSBL sites exist on the Internet. Some of the sites offer free access to their lists, while others are commercial enterprises. Please check each service's web site for the usage requirements. Process Software does not endorse any particular DNSBL site, but you can find a large list of DNSBL servers at this Open Directory URL:

```
http://dmoz.org/Computers/Internet/Abuse/Spam/Blacklists/
```

PMAS can be configured to consult multiple DNSBL servers for each message.

1.4.5.1 Configuring PMAS DNSBL support

To activate the PMAS DNSBL support, simply create the configuration file /pmas/data/pmas_dnsbl.conf and restart PMAS. For your convenience, a sample DNSBL configuration template is provided in /pmas/data/pmas_dnsbl.template. You can copy the .TEMPLATE file to .CONF and then edit it as needed.

The configuration file is a simple text file consisting of lines containing keywords and their parameters. The keywords are described in the following section. To consult multiple DNSBL servers, simply add a line for each desired server.

1.4.5.2 DNSBL commands

The format for each line in the DNSBL configuration file is as follows:

```
dnsbl_reject site[=value,...] "Message"
dnsbl_warn site[=value,...] "Header to add to message"
dnsbl_accept site
dnsbl_allow_host IP-address
dnsbl_disallow_host IP-address "Message"
dnsbl_allow_email email-address
dnsbl_block_email email-address "Message"
```

Table Table 1–1 describes each of the DNSBL keywords and their meanings.

Table 1–1 DNSBL keywords

Keyword	Description
DNSBL_REJECT	Consults the named DNSBL site and, if the sender's IP address is listed, a ‘‘550’’ rejection message (the text of which is taken from the message parameter) is issued for each received RCPT TO: command. The optional comma-separated list of values can be used to limit matches to specific return addresses from the DNSBL (i.e, 127.1.0.1).

Table 1–1 (Cont.) DNSBL keywords

Keyword	Description
DNSBL_WARN	Consults the named DNSBL site and, if the sender's IP address is listed, the specified header text is added to the message's headers. The specified text must be an RFC822-compliant string similar to this example: `X-PMAS-DNSBL: Sender %%IPADDR%% listed in xxx.xxx`
DNSBL_ACCEPT	Turns the specified DNSBL site from a list of addresses to block to a list of addresses to accept.
DNSBL_ALLOW_HOST	Causes the specified IP address to be accepted regardless of its possible listing in a DNSBL. May be needed to allow a host that has been incorrectly listed with the DNSBL site.
DNSBL_DISALLOW_HOST	Works like dnsbl_reject, but no DNSBL site is consulted. All mail from the specified IP address is rejected with ''550'' SMTP replies containing the specified message text.
DNSBL_ALLOW_EMAIL	Causes email with a MAIL FROM: address that matches the email-address parameter to be accepted regardless of the possible listing of the IP address in a DNSBL. May be used to allow specific email through from sites that are often incorrectly listed by DNSBL sites. The email-address parameter may contain wildcards.
DNSBL_BLOCK_EMAIL	Works like dnsbl_reject, but no DNSBL site is consulted. All mail from the specified email address is rejected with ''550'' SMTP replies containing the specified message text.

In the "Message" and "Header" strings, two variables can be specified. If present, these variables will be replaced by the resulting values:

Variable	Description
%%IPADDR%%	The sending IP address being checked
%%RESULT%%	The resulting IP address returned by the DNSBL

For DNSBL_WARN, local PMAS rules can be created to assign a score based on the presence of the specified header in the message. For example, assume you specify a header that looks like this:

```
X-PMAS-DNSBL-XYZ: Sending site listed in XYZ DNSBL
```

The following sample rule shows how a score can be added to a message if that header is present.

```
header XYZ_DNSBL        exists:X-PMAS-DNSBL-XYZ
describe XYZ_DNSBL      Sending system listed in XYZ DNSBL
score XYZ_DNSBL         50.000
```

1.4.6 RHSBL "Right-Hand-Side Blackhole List" Support

RHSBL (Right-Hand-Side Blackhole List) servers maintain lists of domain names that do not conform to all of the Internet RFCs. The PMAS PTSMTP proxy server includes support for accessing RHSBL systems to identify incoming email that is being sent by one of these known non-compliant systems. If the mail is from a domain listed in the RHSBL, the message can be rejected during the SMTP session, or a special header can be added to the message's RFC822 headers.

RHSBL lookups work by taking the domain name (the right-hand side) of the MAIL FROM: address, appending the name of the RHSBL server, and performing a DNS (Domain Name System) query. For example, to query the abuse.rfc-ignorant.org RHSBL for domain name example.com, a DNS query is conducted using the system name "example.com.abuse.rfc-ignorant.org". The response that comes back either indicates that the system is not listed, or a special value in the loopback address range is returned (for example, 127.0.0.2). Different sites return different addresses, which sometimes indicate why a particular site is listed in the RHSBL.

Note:

For information on how RHSBL lookups are implemented and the history of RHSBLs, please see the DNSBL Wikipedia description:

`http://en.wikipedia.org/wiki/DNSBL`

Several RHSBL sites exist on the Internet. Some of the sites offer free access to their lists, while others are commercial enterprises. Please check each service's web site for the usage requirements. Process Software does not endorse any particular RHSBL site, but you can find several RHSBL servers listed at this Open Directory URL:

`http://dmoz.org/Computers/Internet/Abuse/Spam/Blacklists/`

PMAS can be configured to consult multiple RHSBL servers for each message.

1.4.6.1 Configuring PMAS RHSBL support

RHSBL support uses the same configuration file as the DNSBL support (/pmas/data/pmas_dnsbl.conf). For your convenience, a sample DNSBL configuration template is provided in /pmas/data/pmas_dnsbl.template. You can copy the .TEMPLATE file to .CONF and then edit it as needed.

The configuration file is a simple text file consisting of lines containing keywords and their parameters. The keywords are described in the following section. To consult multiple RHSBL servers, simply add a line for each desired server.

1.4.6.2 RHSBL commands

The format for each line in the DNSBL configuration file is as follows:

```
rhsbl_reject site[=value,...] "Message"
rhsbl_warn site[=value,...] "Header to add to message"
rhsbl_accept site
rhsbl_allow_host IP-address
rhsbl_disallow_host IP-address "Message"
```

Table 1–2 describes each of the RHSBL keywords and their meanings.

Table 1–2 RHSBL keywords

Keyword	Description
RHSBL_REJECT	Consults the named RHSBL site and, if the sender's IP address is listed, a ''550'' rejection message (whose text is taken from the message parameter) is issued for each received RCPT TO: command. The optional comma-separated list of values can be used to limit matches to specific return addresses from the RHSBL (i.e, 127.1.0.1).
RHSBL_WARN	Consults the named RHSBL site and, if the sender's IP address is listed, the specified header text is added to the message's headers. The specified text must be an RFC822-compliant string similar to this example: `X-PMAS-RHSBL: Sender %%IPADDR%% listed in xxx.xxx`
RHSBL_ACCEPT	Turns the specified RHSBL site from a list of addresses to block to a list of addresses to accept.
RHSBL_ALLOW_HOST	Causes the specified IP address to be accepted regardless of its possible listing in a RHSBL. May be needed to allow a host that has been incorrectly listed with the RHSBL site.
RHSBL_DISALLOW_HOST	Works like rhsbl_reject, but no RHSBL site is consulted. All mail from the specified IP address is rejected with ''550'' SMTP replies containing the specified message text.

In the "Message" and "Header" strings, three variables can be specified. If present, these variables will be replaced by the resulting values:

Variable	Description
%%DOMAIN%%	The sending domain name being checked
%%RESULT%%	The resulting IP address returned by the RHSBL
%%SENDER%%	The full email address of the sender

For RHSBL_WARN, local PMAS rules can be created to assign a score based on the presence of the specified header in the message. For example, assume you specify a header that looks like this:

```
X-PMAS-RHSBL-XYZ: Sending site listed in XYZ RHSBL
```

The following sample rule shows how a score can be added to a message if that header is present.

```
header XYZ_RHSBL        exists:X-PMAS-RHSBL-XYZ
describe XYZ_RHSBL      Sending system listed in XYZ RHSBL
score XYZ_RHSBL         50.000
```

1.4.7 Anti-Relay Support

The PMAS PTSMTP server includes anti-relay support to prevent unwanted relaying of mail messages through your system. An SMTP relay is a system that accepts and forwards mail that is neither to nor from a local user. Spammers often make use of open relays, so it is important to prevent your system from being used in that manner.

When the PMAS PTSMTP proxy server is used, anti-relay support in the backend server cannot be used, as all incoming mail appears to be coming from a single host (the system running the PTSMTP server, which may be the same system running the backend server). By configuring the PMAS PTSMTP anti-relay support, you can safely make use of the PMAS PTSMTP configuration without opening your system up as an open relay.

1.4.7.1 How the PMAS PTSMTP Anti-Relay works

To configure the anti-relay support, you will need to create a file, /pmas/data/local_domains.txt, listing all of the local domains for which mail should be accepted. Optionally, you can also create a file, /pmas/data/local_addresses.txt, that contains all of the valid email addresses for specific local domains. If the file doesn't exist, or if a domain is not represented in that file, all addresses for the local domains are accepted.

When a new connection is accepted, the IP address of the sending system is checked to see if it's an internal IP address. If it is, it's allowed to relay and send messages. The list of internal IP addresses is stored in /pmas/data/internal_ip.txt.

If it's an external IP address, the MAIL FROM: received from the sending client is checked to see if the sending address is in one of the local domains (as defined by LOCAL_DOMAINS.TXT). If the address does match a local domain, it is then compared to the local addresses for that domain, if they are defined in LOCAL_ADDRESSES.TXT. If the address is part of the local domain, but is not defined in LOCAL_ADDRESSES.TXT, the MAIL FROM: command is rejected (and never sent to the backend SMTP server).

If it's an external IP address, each of the RCPT TO: commands received from the sending client is checked to see if the recipient address is in one of the local domains (as defined by LOCAL_DOMAINS.TXT). If the RCPT TO: address is not in one of the local domains, the RCPT TO: command is rejected (and never sent to the backend SMTP server).

If the recipient is in the local domain and the LOCAL_ADDRESSES.TXT files exists, each address is also verified against the list of valid addresses for that domain, if any are defined. The usefulness of LOCAL_ADDRESSES.TXT really only comes into play when you have an SMTP server that relays for a domain but knows nothing about the valid addresses for that domain, so it accepts any and all addresses to forward on. You can use LOCAL_ADDRESSES to restrict the addresses that are accepted.

1.4.7.2 Site-specific address verification

Instead of using LOCAL_ADDRESSES.TXT to list local addresses, a site-supplied email address verification routine can be used to perform whatever verification may be necessary. If a site-defined routine is provided, each MAIL FROM: and RCPT TO: command that specifies an address in a local domain (again, as defined by LOCAL_DOMAINS.TXT) is passed to the site-supplied routine for verification. This site-supplied routine can do whatever you program it to do: check an indexed file, consult SYSUAF, perform an LDAP lookup, etc.

A (simple) sample site address verification program can be found in /pmas/api/site_verify_address.

The shareable object containing the site-supplied verification routine must be named libsite_verify_address.so and must be located in the /pmas/bin/ directory.

The function called by PreciseMail to verify the address must be named VERIFY_ADDRESS, must accept a single pointer to a character string containing the email address to be verified, and must return 0 (zero) if the address is invalid or 1 (one) if the address is valid.

If the address is invalid, the command is rejected (and never sent to the backend server).

1.4.7.3 Configuring the Anti-Relay support

The following steps must be taken to enable the PMAS PTSMTP Anti-Relay support:

1 Create and populate the file /pmas/data/local_domains.txt.

2 If it doesn't already exist, create and populate the file /pmas/data/internal_ip.txt.

3 If desired, create and populate the file /pmas/data/local_addresses.txt.

The LOCAL_DOMAINS file lists the local domains for which mail should be accepted. Each line in the file should list a domain name, with optional wildcards. Lines beginning with "#" are ignored as comments. A sample LOCAL_DOMAINS.TXT file would look like this:

```
# Our domains
#
example.com
*.example.com
#
# Another domain for which we accept mail:
#
someotherdomain.com
```

For information on the format of the INTERNAL_IP.TXT file, please see Section 1.4.2, Identifying internal IP addresses.

The optional LOCAL_ADDRESSES file lists the valid addresses for one or more local domains. Each line in the file specifies an email address. Wildcards can be used in the addresses, and lines beginning with "#" are ignored as comments. For example:

```
# Accept mail for postmaster
postmaster@example.com
```

To list multiple addresses for a given domain, you can create a line using the “domain” keyword, followed by lines listing the valid users for that domain (and omitting the domain name). For example:

```
#
#  Valid example.com addresses
#
domain example.com
postmaster
abuse
system
joe.user
```

All the addresses listed above will have an implicit “@example.com” appended when the address checks are performed. To list multiple domains in the LOCAL_ADDRESSES file, simply repeat the sequence above as needed, specifying the various domains and addresses.

There are also two configuration variables that let you control the reject messages sent to the remote client:

```
ptsmtp_norelay_reply    550 5.7.1 Relaying not allowed: %s
ptsmtp_nouser_reply     550 5.1.1 Illegal or unknown user: %s
```

The “%s” in the variable values is replaced with the address being rejected.

1.4.8 Sender Policy Framework (SPF) Support

The PMAS PTSMTP server includes support for performing Sender Policy Framework (SPF) DNS queries for incoming messages. SPF can be used to determine if the system sending a message is authorized to send that message for the domain specified in the MAIL FROM: command (the return-path or sender address). It can be used to help detect forged email addresses using sender information published by domain owners. SPF is defined in RFC 4408.

A full description of how SPF works is beyond the scope of this document. For full details about SPF, including a tutorial on how to set up SPF records for a domain, please see the following websites:

```
http://www.openspf.org/
http://en.wikipedia.org/wiki/Sender_Policy_Framework
```

When SPF checks are enabled in PMAS, DNS queries are made to retrieve the SPF TXT records for the domain specified in the MAIL FROM: address. The IP address of the system sending the message is compared to the SPF policy definition. If the target domain has a published SPF policy, a “Received-SPF:” header is added to the message indicating whether or not the system is authorized to send mail for that domain:

```
Received-SPF: pass (example.com: domain of charter.net designates
 209.225.20.181 as permitted sender) client-ip=209.225.20.181;
 envelope-from=joeuser@charter.net;
```

The first word in the Received-SPF: header indicates the SPF lookup status for the client IP address. Valid values are:

- pass - The sending client is authorized to send mail for the specified domain according to that domain’s publish SPF policy
- fail - The sending client is not authorized to send mail for the specified domain
- softfail - The sending client is neither authorized nor unauthorized to send mail for the specified domain (a debugging aid between NEUTRAL and FAIL)
- neutral - An SPF record exists for the domain, but no SPF policy is defined
- permerror - A permanent error occurred looking up the SPF record (usually a syntax error in the SPF record)
- temperror - A temporary error occurred looking up the SPF record (usually a server timeout preventing retrieval of the SPF record)

If no SPF record for the specified domain is found, PMAS adds an “X-PMAS-SPF:” header is added to the message to indicate that:

```
X-PMAS-SPF: (recv=node.example.com, send-ip=74.120.12.114) Could not find a valid S
```

Note: One of the biggest downsides to using SPF is that messages routed through a mail forwarder will not pass SPF checks for the specified domain. SPF lookups are performed on the sending client’s IP address, and if the sending client is a mail forwarder for your domain, it won’t be listed as being a valid SPF for the sending domain. If your PMAS PTSMTP system receives all of its mail from a mail forwarder, you should not enable SPF (or should customize to skip the checks for the forwarder; see Section 1.4.8.3 for more information).

1.4.8.1 Using SPF results to detect spam

For spam detection, the only really useful SPF status values are “fail” and “softfail”. Spammers were among the first to deploy SPF records, which makes a “pass” result useless by itself in determining whether or not a message is spam.

According to the RFC, a message coming from a client that fails the SPF check should be rejected. A “softfail” result may be quarantined, if desired.

1.4.8.2 Enabling SPF and writing SPF rules

To enable SPF lookups in the PMAS PTSMTP proxy server, the configuration variable *ptsmtp_enable_spf* must be defined as “yes” (the default setting). With SPF enabled, “Received-SPF:” and “X-PMAS-SPF:” headers will be added to messages as appropriate. No other action is performed unless site-specific rules are defined.

Rules can be written and added to the system allow/block list (see Section 3.9) to reject, discard, or quarantine messages based on the results of the SPF check. For example, the following rule would reject any message that fails the SPF check:

```
rule reject header:Received-SPF starts "fail" "Sender not authorized according to S
```

A similar rule could be used to discard or quarantine messages that result in a “softfail” or “temperror”:

```
rule discard header:Received-SPF matches_regexp "(?:softfail|temperror)"
```

While it is not recommended that you allow all messages that pass the SPF check, it is safe to allow messages that pass for certain, known domains. For example, Amazon.com publishes an SPF record, so you can safely allow messages from Amazon that pass the SPF check. A rule can be constructed using the information found in the “Received-SPF:” header.

```
rule allow header:Received-SPF matches_regexp \
   "^pass \(example\.com: domain of bounces\.amazon\.com"
```

Because all message headers can be forged, including Received-SPF: headers, an allow rule based on SPF results should include your local system name (as it appears in the header) in the test to ensure that the rule is the one added by PMAS on your system and not a forged header or a header added by another system.

An “X-PMAS-SPF:” header is added for temperror and permerror results from SPF lookups, too.

1.4.8.3 Configuring SPF

The SPF configuration file, /pmas/data/ptsmtp_spf.conf, can be used to customize the SPF component and to list IP addresses for which SPF checks should not be performed. A configuration template file, /pmas/data/ptsmtp_spf.template, that can be copied and customized is provided.

Table 1–3 lists the keywords that can be used in the SPF configuration file.

Table 1–3 SPF Configuration File Keywords

Keyword	Description
SPF_DOMAIN	Specifies the local system name that is added to Received-SPF: headers created by PMAS. By default, the value of the PMAS configuration variable LOCAL_SYSTEM_NAME is used.
SPF_ALWAYS_ADD_HEADER	On a successful SPF lookup, a Received-SPF: header is added. If SPF is not configured for the domain, or if the SPF check is skipped for other reasons, an X-PMAS-SPF: header is added instead. Defining this variable as ‘‘no’’ will prevent the addition of the X-PMAS-SPF: header.
SPF_LOGGING	If defined as ‘‘yes’’, all SPF check results are logged to /pmas/log/ptsmtp_spf.log. The default value is ‘‘no’’.
SPF_TRUSTED_HOST	Specifies the IP address of a ‘‘trusted host’’, a system for which SPF checks should not be made. Any mail forwarders from which your system receives email should be listed as a trusted host. Multiple SPF_TRUSTED_HOST lines can be specified in the configuration file.

Messages received from a trusted host will have a header like the following example added to the message:

```
X-PMAS-SPF: SPF check skipped for trusted IP (recv=node.example.com,
 send-ip=10.11.12.13)
```

1.4.9 Tarpitting Support

A *tarpit* (also known as *Teergrube*, the German word for tarpit) is a method of delaying incoming SMTP connections for as long as possible. The PMAS PTSMTP proxy server supports simple tarpitting.

When tarpitting is enabled, responses to RCPT TO: commands are intentionally delayed, increasing the amount of time it takes for the sending client to send the message. Some people view this as a way to "punish" spammers by increasing their message transmission times, thus decreasing the number of messages they can send. Two new configuration variables control tarpitting: PTSMTP_TARPIT_COUNT specifies the number of RCPT TO: commands per session that are allowed before tarpitting is activated, and PTSMTP_TARPIT_DELAY species the number of seconds each RCPT TO: response should be delayed.

1.4.10 Filtering Support

One of the ways hackers and bots try to gain access to a system is by doing dictionary attacks to try to determine passwords for accounts. One method for doing so is by connecting to an SMTP server and issuing AUTH commands to try locate a valid username and password. Clients that do this will often connect and issue dozens of AUTH commands in a single session. This needlessly ties up resources, in addition to providing potential access to a system if the attack is successful.

Working on the assumption that virtually all legitimate authenticated SMTP sessions will succeed—meaning that the users' authentication information is stored in their email clients and will never result in a bad password—the PTSMTP filter plugin lets you block incoming IP addresses that fail SMTP AUTH attempts. Instead of connecting to the proxy server and issuing dozens of attempts, the system can be filtered out completely at the network stack level, preventing further attempts.

The PTSMTP filter plugin works on Linux using the built-in IPTABLES firewall, though any other software can also be used. See below for details.

Note: The filter plugin triggers on a single SMTP AUTH failure. No attempt is made to track IP addresses and attempts. It should only be enabled in situations where users will not be inadvertently entering bad passwords.

1.4.10.1 Enabling the PTSMTP Filter plugin

The filter plugin is activated when the file exists. A ptmstp_filter.template version of the file is provided. To enable filtering, simply rename or copy the .template file to .conf.

The ptsmtp_filter.conf file is used to specify the command that should be used to add the client IP address to the firewall to block connections. A typical configuration file might look like this (which is also provided as an example in the file), where COMMAND is a keyword:

```
command iptables -A INPUT -s %s -j DROP
```

The "%s" is replaced by the IP address of the offending client. This command will add an entry to iptables to drop any packets from that IP address.

A script could be run instead to provide for more sophisticated processing.

1.5 Message Scoring for Spamicity

PreciseMail Anti-Spam Gateway opens each email message and runs a variety of tests against the message headers and the text portions of the message body. Each of these tests has a score associated with it that represents the likelihood that that test indicates a spam message. Scores can be positive (spam-like) or negative (non-spam-like); the sum of scores for all the matching rules represents the message's final score, which is then used to determine the message's fate (see Section 1.6, Email Message Disposition).

1.5.1 Matching rules

Most of the rules consist of regular expressions that are matched against the message's headers and body parts. These rules are defined in files in the /pmas/data/ directory. It is possible to change the scores for any of the rules and add new, site-specific rules. See Chapter 3, The PreciseMail Anti-Spam Gateway Rules, for full details on modifying these rules.

1.5.2 Allow and Block Lists

It is not desirable for all email to pass through PreciseMail Anti-Spam Gateway. Messages from local users, from known users or domains, and from public mailing lists are examples of messages that most sites will not want processed for spamicity, either to avoid the overhead of checking messages that won't be spam or to avoid the risk of a spam-like message from a trusted source being treated as spam.

PreciseMail Anti-Spam Gateway allows sites to specify rules for messages that should always be accepted (allow rules) and rules for messages that should always be rejected (block rules). Both system-wide and user-specific rules are supported. The system-wide rules are defined in the file 00_allowblocklists.cf in /pmas/data/. This text file can be edited to specify addresses and header-matching regular expressions to determine which messages should always be kept or rejected.

Note: The various block rules cause messages to be rejected during the SMTP session or silently deleted. Blocked messages can not be viewed or retrieved as they are not stored anywhere.

User-defined allow and block rules are normally created by users through the PreciseMail Anti-Spam Gateway Processor, an email-based command processor. They can also be edited with a text editor by system administrators. The user-specific files are stored in /pmas/user_rules/ and are named like USER.SYSTEM_DOMAIN.

More information on the allow and block rules can be found in Section 3.9, Adding Allow, Block, Quarantine, Discard, Score, and Reject Rules.

1.6 Email Message Disposition

PreciseMail Anti-Spam Gateway can be directed to do several things with a message it determines to be spam: the message can be discarded; the message can be quarantined; or the message can have X-PMAS headers added to it or have its subject line modified and be passed on for delivery.

1.6.1 Quarantining Messages

By default, PreciseMail Anti-Spam Gateway will “quarantine” messages that have a score above a certain threshold (5.0, by default). When a message is quarantined, it is written to the PreciseMail Anti-Spam Gateway quarantine directory (/pmas/quarantine/) and an entry for the message is logged to a quarantine index file. PreciseMail Anti-Spam Gateway then directs the PMDF pmas channel to discard the message instead of forwarding it on for delivery.

At site-defined times of the day, a cron job is executed to process the messages in the quarantine directory. This cron job processes the quarantine index file and sends a mail message to each user for whom messages have been quarantined. The mailed message shows the user the Date:, From:, and Subject: of each message that has been quarantined on his or her behalf and includes instructions that allow the user to retrieve a quarantined message.

Quarantined messages are kept for a certain site-defined number of days (14 is the default), after which time they are automatically deleted from the system.

1.6.2 Discarding Messages

PreciseMail Anti-Spam Gateway can be configured to discard messages that have a score above a certain discard threshold value. If discarding is enabled, discarded messages are not really deleted by default. Instead, they are written to the PreciseMail Anti-Spam Gateway discard directory (pmas/discard/), where they are kept for a site-defined number of days (14 is the default). The difference between the discard and quarantine directories is that users are not notified of messages that are discarded.

The discard directory is maintained for emergency retrieval of messages that scored above the discard threshold. Because spam detection is not an exact science, and because legitimate message contents vary from site to site, it is possible that a message might trigger enough rules to be discarded, but still be considered legitimate by a site. Instead of such messages being thrown away, they are kept in the discard directory to allow system administrators to retrieve them, if necessary.

A discard index file is maintained (as is done for the quarantined messages) and PMAS system administrators can view the discarded messages via the PMAS admin GUI web interface. Administrators can also choose to allow users to view their own discarded messages, but that will probably confuse most novice users.

Note: **If a message was mistakenly discarded, it can be released via the PMAS admin GUI web interface or by sending a message to the PreciseMail Processor to release the message.**

1.6.3 Domain-specific quarantine and discard index files

PMAS supports domain-based quarantine and discard index files. Normally, all messages quarantined and discarded are recorded in single index files, one for each type. For sites that host multiple domains, quarantine and discard lookups for a user resulted in searches through the records for users in all of the domains. It is possible now to have separate quarantine and discard index files for each domain, speeding up quarantine and discard lookups, as well as providing the ability to send quarantine notifications at different times for different domains.

To enable the domain-specific index files, simply define the desired configuration variables DISCARD_INDEX_FILE and QUARANTINE_INDEX_FILE as the string “$DOMAIN$”:

```
quarantine_index_file   $DOMAIN$
```

When domain-specific index files are enabled, the index file will be named using the domain; for example, PMAS_ROOT:[QUARANTINE]EXAMPLE_COM.DAT. One file will be created for each domain for which messages are quarantined and/or discarded.

The NOTIFY_QUARANTINED program was modified to accept a domain name as a command line parameter. If specified, only the quarantine index for that domain is processed. For example: domain is processed. For example:

```
$ notify :== $pmas_exe:notify_quarantined.exe
$ notify example.com
```

Command procedures can be written to deliver domain-specific quarantine notices at various times.

1.6.4 Modifying and Forwarding Messages

Messages that are not quarantined or discarded are forwarded on for delivery to their target recipients. By default, PreciseMail Anti-Spam Gateway will add X-PMAS headers to the RFC822 headers of messages that have a non-zero score to indicate which rules were triggered by a particular message. Typical headers that are added will look like this:

```
X-PMAS-BDY-MAILTO_LINK: Includes a URL link to send an email (0.100)
X-PMAS-META-MANY_EXCLAMATIONS: Subject has many exclamations (1.097)
X-PMAS-Final-Score: 1.197
```

The first two lines indicate two of the rules the message triggered, and the last line shows its final score.

PreciseMail Anti-Spam Gateway can also be configured to modify the subject lines of messages that have scores above a certain threshold. If enabled, "[SPAM]", or any site-defined string, will be prepended to subject lines of messages.

Depending on the mail clients in use, users can configure their mail clients or PMDF Sieve files to automatically route messages with certain X-PMAS headers or a modified subject line to "spam suspect" folders.

1.6.5 Rejecting Messages

PreciseMail Anti-Spam Gateway can be configured to reject messages that have a score above a certain reject threshold value if the PMAS Pass-Through SMTP Server (PTSMTP) is used. By rejecting messages, you eliminate the overhead of having those messages processed by your primary mail software, and you might encourage spammers to drop that particular address (though that's probably not very likely).

Messages that are rejected are never stored on your system, so it is not possible to recover rejected messages. Care should be taken to ensure that the rejection threshold is set to an appropriate level.

Note: By default, enabling the rejection of messages causes PMAS to reject blocklisted messages instead of silently deleting them. You can adjust the BLOCK_SCORE **keyword to prevent this from happening, if you don't want blocklisted messages rejected.**

1.7 Logging

PreciseMail Anti-Spam Gateway maintains a log file showing the disposition of each message processed. The log file is named pmas.log and can be found in /pmas/log/. Each entry in the log file is made up of nine different fields, all separated by a vertical bar " | ":

1 The system date and time the message was processed

2 The disposition code (see the table below)

3 The final score for the message

4 The envelope From address (the sender)

5 The envelope To address (the recipient)

6 The contents of the Subject: header from the message

7 The contents of the Message-ID: header from the message

8 A comma-separated list of rule names that were matched

9 The output file name if the message was quarantined or discarded

The disposition codes that may be found in the file are shown in the table below.

Code	Meaning	Description
N	Nada	Nothing was done to the message, which was forwarded to the user; only found if there was a severe error during processing
O	Opt-out	User opted-out of PMAS scanning
F	Forward	Message was forwarded to the user
AS	Allowed	Message matched a system-wide allow rule and was forwarded
AU	Allowed	Message matched a user-defined allow rule and was forwarded
BS	Blocked	Message matched a system-wide block rule and was deleted or rejected
BU	Blocked	Message matched a user-defined block rule and was deleted or rejected
QS	Quarantineed	Message was quarantined
QU	Quarantineed	Message matched a user-defined quarantine rule and was quarantined
DS	Discarded	Message was discarded
DU	Discarded	Message matched a user-defined discard rule and was discarded
TS	Tag	Message was tagged by a system rule or setting
TU	Tag	Message was tagged by a user-defined rule
RS	Rejected	Message was rejected during SMTP session by a system-wide rule
RU	Rejected	Message was rejected during SMTP session by a user-defined rule
M	Megarejected	Message was megarejected during SMTP session by a system-defined rule
L	Listed	Message was rejected via a DNSBL or RHSBL lookup

A nightly cron job creates a new pmas.log file each day. The previous day's log file is renamed to .log-yyyy-mm-dd.

1.8 Statistics

Statistics about the messages processed by PreciseMail Anti-Spam Gateway can be generated using the pmas_stats program in /pmas/bin/. It takes two parameters: the name of the log file and an optional output file name. If an output file name is not specified, the statistics information is written to your terminal.

```
# /pmas/bin/pmas_stats /pmas/log/pmas.log-2003-06-26 report.txt
```

1.9 The Alias File

When processing messages, PreciseMail Anti-Spam Gateway will look for an alias file (aliases.txt in /pmas/data/, by default). This file specifies alternate addresses that are to be used when quarantining messages for certain recipients. For example, the quarantine notification messages for a mailing list should typically be directed to an alternate user and not to the mailing list itself.

Each entry in the file consists of two to four items separated by whitespace:

- The incoming envelope address
- The desired alias address
- An optional address used for user interface authentication
- An optional authentication method

An entry in the alias file to route quarantine notices to another address might look like:

```
info-apes@example.com          owner-info-apes@example.com
```

Quarantine notification messages for mail destined for “info-apes” will be sent to “owner-info-apes” instead.

Alias lookups are case-insensitive. Wildcards are supported for the incoming address (the left-side of an entry); valid wildcard characters are “*” to match any number of characters and “?” to match any single character.

Other sample aliases might look like these lines:

```
#
#  Various aliases for Galen
#
galen@nodea.example.com   galen@example.com
sales@example.com   galen@example.com
#
info-*@example.com   galen@example.com
```

The alias file is also consulted by PreciseMail Anti-Spam Gateway when it looks for user-defined allow/block files. Using the example above, a message coming in for “sales” will be compared against the allow/block rules for “galen@example.com”.

Finally, the alias file is also consulted by the PreciseMail Anti-Spam Gateway User Interface when a user logs in. By performing alias lookups, users with multiple email addresses that all point to the same account can log in under any of those addresses by specifying the proper password for the resulting alias. Using the example above, a user can log in to “sales@example.com” by specifying the password for “galen@example.com”.

1.9.1 Regular expressions

For more sophisticated wildcard matching and substitution, regular expressions are also supported. To specify that an address (the value on the left) is a regular expression, it should begin and end with a vertical bar " | ". The regular expression matching is caseless (lowercase and uppercase are treated the same). You can use parentheses within the expression to assign matching substrings to variables named $1, $2, through $9 for substitution in the resulting alias. For example, the following entry will convert all email addresses of the form "First.Last@example" to "last@example":

```
|.+\.(.+)@example\.com|      $1@example.com
```

The parentheses above capture the substring between the "." and the "@", assigning it to variable $1. For the email address "Joe.User@example.com", the resulting alias would be "User@example.com".

1.9.2 Special Mailing List Aliases

There is one special case for the alias address (the right side). When a user releases a message listed in a quarantine notice, the message is released to the original envelope address; in the case of a mailing list with no alias entry, the message would be released to the mailing list address. It may be desirable to allow quarantine notices to be sent to a mailing list (and thus to all the members of that list) and allow any one of those recipients to release the quarantined message without having the message released to all recipients. You can designate such a list using the special "*" alias address:

```
info-apes@example.com            *
```

With that entry, mail sent to "info-apes" that is to be quarantined will result in the quarantine notice being sent to "info-apes@example.com". Any recipient of that notice can release the message, but the released message will be sent only to the requesting user and not to the list.

Note: As a security measure, the PreciseMail Anti-Spam Gateway Processor will only release messages to the original envelope To: address, regardless of the return address for the release request. A malicious user could request the release of a message (if he or she knows the filename for a message), but the message will only be mailed to the original recipient and not to the malicious user. The special "*" alias above circumvents this process, allowing anyone to release and receive one of these special messages if he or she knew the filename to release. This is a very unlikely scenario, but you should be aware of it when setting up such an alias.

1.9.3 Authentication Aliases

The optional third and fourth values on a line in the alias file are used for authenticating access via the web-based PreciseMail Anti-Spam Gateway User Interface. They can be specified to allow access to quarantine areas for addresses that do not actually have a direct user correlation. For example, you might want to redirect mailing list quarantine notices to a special address like this:

```
info-*@example.com        owner-info-blah@example.com
```

For all the "info-*" lists, quarantine notices and allow/block lookups will use the address "owner-info-blah". While you could specify a particular user's email address, that would result in a quarantine notice possibly mixing that user's personal mail with the mailing list mail. The alias above will generate a separate quarantine notice, but there also would not be a way to use the web-based user interface if "owner-info-blah" isn't a real account. By using an authentication alias, you can specify the user account against which the user interface should authenticate. For example:

```
info-*@example.com  owner-info-blah@example.com   joe@example.com
```

User "joe" can log into the user interface using the email address "owner-info-blah@example.com" and supplying the password for account "joe@example.com". As long as the password matches, the user will be logged in as "owner-info-blah@example.com".

The various authentication methods are defined using the PMAS configuration variable AUTH_METHODS (Section 2.2.11.1). Those methods can be overridden for a particular alias by specifying a particular method as the optional fourth parameter. For example, if user "joe" should be authenticated via IMAP4 from a particular system, the following entry can be used:

```
joe@example.com  joe@example.com  joe@imap4.somewhere.com imap4
```

When this user logs in, PMAS will use IMAP4 authentication using the IMAP4 server "imap4.somewhere.com" and the username "joe". IMAP4 and POP3 servers usually authenticate using only the username portion of an address. However, some servers support virtual domains and require the full email address for authentication. To effect this requirement, the qualifier "/virtual" can be appended to "imap4" or "pop3":

```
joe@example.com  joe@example.com  joe@imap4.somewhere.com imap4/virtual
```

In the event that the password sent needs to be quoted, you can use the qualifier "/quote":

```
joe@example.com  joe@example.com  joe@imap4.somewhere.com imap4/virtual/quote
```

It's possible that the full email address used should contain a different domain name. This can be done by specifying the desired full address inside parentheses, as in this example:

```
(joe@elsewhere.com)@imap4.somewhere.com
```

In that example, the host imap4.somewhere.com would be contacted, but the email address used in the authentication would be “joe@elsewhere.com”.

Finally, authentication using SSL, which is required with Gmail, is also supported using the “/ssl” qualifier. A special port can be specified after the system name:

```
joe@example.com  joe@example.com  joe@pop.gmail.com:995/ssl pop3
```

1.10 PreciseMail Anti-Spam Gateway Processor

The PreciseMail Anti-Spam Gateway Processor is an email-based user interface to PreciseMail Anti-Spam Gateway. Users can send mail to a special email address that’s routed to the processor via the PIPE channel or a command alias (please see the *PreciseMail Anti-Spam Gateway Installation Guide* for details). Processor commands are included in the body of the message. The PreciseMail Anti-Spam Gateway Processor can be used to release quarantined messages and to create and manage user-defined allow/block files.

2 Configuring PreciseMail Anti-Spam Gateway

This chapter describes the PreciseMail Anti-Spam Gateway configuration file and the configuration keywords.

All of the configuration files are stored in the directory /pmas/data/.

2.1 Configuration file

The primary PreciseMail Anti-Spam Gateway configuration file is named pmas_config.dat. It is a text file that consists of keywords and their associated settings. To change a setting, simply add or modify a line to specify a new value for a given keyword.

When PreciseMail Anti-Spam Gateway is installed, a pmas_config.template file is provided that lists and documents all of the available configuration settings. If you wish to customize the PreciseMail Anti-Spam Gateway configuration, simply copy this .template file to pmas_config.dat and edit it as desired.

Comments can be included in the file by starting the line with "#" or "!". A partial sample file looks like this:

```
#
#  PreciseMail Anti-Spam Gateway Config File
#  Last modified: 26-JUN-2003 00:55 by GOATHUNTER
#
add_headers yes
quarantine_messages yes
discard_messages yes
discard_threshold 20.0
```

2.2 The Configuration Keywords

Table 2–1 and Table 2–2 list the configuration keywords, while the subsequent subsections describe the keywords in some detail.

Table 2–1 PreciseMail Anti-Spam Gateway Configuration Keywords Listed Alphabetically

Keyword	Section	Description
ADD_HEADERS	Section 2.2.2.1	Controls if X-PMAS headers are added
ADD_SPAM_YES_HEADER	Section 2.2.2.2	Controls if "X-PMAS-Spam: Yes" header is added
ADD_SPAM_YES_THRESHOLD	Section 2.2.2.3	Threshold for adding "X-PMAS-Spam: Yes" header
ADMIN_EMAIL_ADDRESS	Section 2.2.10.17	The system administrator email address
ALIAS_FILE	Section 2.2.14.1	Specifies the address alias filename
ALLOW_USER_DISCARD	Section 2.2.10.20	Allow users to discard messages

Table 2–1 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Listed Alphabetically

Keyword	Section	Description
ALLOW_USER_OPTIN	Section 2.2.10.19	Allow users to opt in or out
ALLOW_USER_QUARANTINE	Section 2.2.10.21	Allow users to quarantine messages
ALLOW_USER_TAGGING	Section 2.2.10.22	Allow users to tag messages
ANTIVIRUS_ENABLED	Section 2.2.9.1	Enables anti-virus scanning in PTSMTP proxy server
ANTIVIRUS_DIR	Section 2.2.9.3	Specifies the location of the anti-virus data files
ANTIVIRUS_PACKAGE	Section 2.2.9.2	Specifies the name of the anti-virus package
AUTH_IMAP4_HOSTS	Section 2.2.11.4	List of IMAP4 authentication hosts
AUTH_METHODS	Section 2.2.11.1	Lists the available authorization methods
AUTH_POP3_HOSTS	Section 2.2.11.2	List of POP3 authentication hosts
AUTOBAYESIAN_HAM_THRESHOLD	Section 2.2.12.3	Autotraining for ham threshold
AUTOBAYESIAN_SPAM_THRESHOLD	Section 2.2.12.4	Autotraining for spam threshold
AUTOTRAIN_BAYESIAN	Section 2.2.12.2	Controls the autotraining of the Bayesian engine
AUTOUPDATE_ADDRESS	Section 2.2.17.4	Specifies email address used for email-based updates
AUTOUPDATE_RULES	Section 2.2.17.1	Controls whether or not automatic ruleset updates are performed
AUTOUPDATE_SOPHOS	Section 2.2.17.2	Controls whether or not Sophos AV database updates are performed
AUTOUPDATE_STATS	Section 2.2.17.3	Controls the uploading of spam-related statistical data to Process Software
BAYESIAN_MULTIPLIER	Section 2.2.12.5	Multiplier for the Bayesian spamicity rating
BLOCK_DISPOSITION	Section 2.2.5.3	Determines the actual disposition of messages that are blocked.
BLOCK_SCORE	Section 2.2.5.4	Specifies the score assigned to messages that are blocked. May be adjusted to reject blocked messages instead of discarding them.
CLU_CLIENTS	Section 2.2.18.1	Specifies secondary nodes of a Data Synchronization Cluster.
CLU_SERVER	Section 2.2.18.2	Specifies the primary node of a Data Synchronization Cluster.
CLU_SERVER_CLIENT	Section 2.2.18.3	Enables Data Synchronization Cluster functionality.
DEBUG_LEVEL	Section 2.2.13.1	The debug message output level
DEBUG_LOGFILE	Section 2.2.13.2	Specifies the name of the debug log file
DISCARD_MESSAGES	Section 2.2.4.1	Controls the discarding of messages
DISCARD_MSG_LIFETIME	Section 2.2.4.3	The number of days discarded messages are kept
DISCARD_THRESHOLD	Section 2.2.4.2	The score threshold for discarding messages
GUI_ALLOW_DISCARD_VIEW	Section 2.2.10.13	Allow users to view their discarded messages
GUI_ALLOW_QUARALL_DEFAULT	Section 2.2.10.8	Allow users to set their default quarantined message display to ‘‘all’’
GUI_ALLOW_QUARANTINE_ALL	Section 2.2.10.7	Allow users to display all their quarantined messages
GUI_COOKIE_LIFETIME	Section 2.2.10.1	Specifies the lifetime in hours of the PMAS login cookie
GUI_DEFAULT_QUARANTINE_ALL	Section 2.2.10.9	Change the default display for quarantined messages to ALL
GUI_QUARANTINE_DISPLAY_DAILY	Section 2.2.10.10	Change the default display for quarantined messages to daily

Table 2–1 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Listed Alphabetically

Keyword	Section	Description
GUI_DELETE_UPON_RELEASE	Section 2.2.10.11	Causes released messages to be deleted from the quarantine
GUI_FORCE_JAVA_SORT_OPERA	Section 2.2.10.15	Forces usage of the Javascript quarantine sort with the Opera browser
GUI_JAVA_SORT_MAXMSG	Section 2.2.10.14	Specifies the maximum number of messages for a quarantine view in which the Javascript sort is used
GUI_RENAME_UPON_DELETE	Section 2.2.10.12	Causes deleted messages to actually be renamed to _DELETED
GUI_URI_HOST	Section 2.2.10.2	Specifies the host name used in PMAS URLs
GUI_URI_USE_HTTP_HOST	Section 2.2.10.3	Specifies which host name is used in PMAS URLs
GUI_URI_PATH	Section 2.2.10.5	Specifies the path name for PMAS URLs
GUI_URI_PROTOCOL	Section 2.2.10.6	Specifies the protocol for the served pages
GUI_URI_SCRIPT_PATH	Section 2.2.10.4	Specifies the script path name for PMAS URLs
HEADER_PREFIX	Section 2.2.2.4	The prefix appended to message headers added by PMAS
IMAP4_CONNECT_TIMEOUT	Section 2.2.11.5	Specifies a connect timeout for IMAP4 authorization requests
LDAP_AUTH_FILTER	Section 2.2.11.8	The LDAP search filter
LDAP_AUTH_SERVER	Section 2.2.11.6	LDAP server name
LDAP_BASE_DN	Section 2.2.11.7	The LDAP base distinguished name
LDAP_GROUP_FILTER	Section 2.2.11.9	The LDAP search filter for user groups
LDAP_GROUPMEMBER_ATTR	Section 2.2.11.10	The name of the attribute that contains a group member’s DN
LDAP_SEARCHACCT_DN	Section 2.2.11.11	The Distinguished Name of the LDAP search user
LDAP_SEARCHACCT_PASSWORD	Section 2.2.11.12	The password for the LDAP search user
LDAP_USE_TLS	Section 2.2.11.13	Enables use of LDAPS for user authentication
LOCAL_DOMAIN_NAME	Section 2.2.1.2	Specifies the default local domain name
MASTER_LOGFILE	Section 2.2.14.3	Specifies the master log file name
MAXIMUM_USER_BLOCK_RULES	Section 2.2.5.5	Specifies the maximum number of user block rules that will be applied to messages
MIME_HELP_MESSAGE	Section 2.2.16.5	Specifies that the HELP reply is a MIME message
MIME_QUARANTINE_MESSAGE	Section 2.2.15.2	Controls the type of quarantine notification messages
MODIFY_SUBJECT	Section 2.2.2.5	Controls if Subject: lines are modified
MODIFY_SUBJECT_APPEND	Section 2.2.2.6	Appends the subject_tag to Subject: lines
MODIFY_SUBJECT_THRESHOLD	Section 2.2.2.7	The score threshold for modifying Subject: lines
NOTIFY_DEBUG_LEVEL	Section 2.2.15.4	The debug message level for the notify job
NOTIFY_MSG_SUBJECT	Section 2.2.15.3	Specifies the Subject: line for quarantine notification messages
OPT_IN_BY_DEFAULT	Section 2.2.10.18	Specifies the ‘‘opt-in’’ default for users
POP3_CONNECT_TIMEOUT	Section 2.2.11.3	Specifies a connect timeout for POP3 authorization requests
PMAS_SYSTEM_NAME	Section 2.2.2.8	The system name added to the X-PMAS-Software: header
PROCESSOR_USER_ADDRESS	Section 2.2.16.2	The full email address for the PreciseMail Anti-Spam Gateway Processor

Table 2–1 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Listed Alphabetically

Keyword	Section	Description
PROCESSOR_USER_NAME	Section 2.2.16.1	The local username for the PreciseMail Anti-Spam Gateway Processor
PTSMTP_ENABLE_STARTTLS	Section 2.2.8.9	Enables STARTTLS support in the PMAS PTSMTP server.
PTSMTP_IDLE_TIME	Section 2.2.8.16	Specifies the number of seconds a temporary PMAS PTSMTP process is idle before it exits.
PTSMTP_LISTEN_HOST	Section 2.2.8.1	Specifies the interface IP addresses on which the PMAS PTSMTP Server listens for incoming SMTP connections.
PTSMTP_LISTEN_PORT	Section 2.2.8.2	Specifies the port on which the PMAS PTSMTP Server listens for incoming SMTP connections.
PTSMTP_LISTEN_PORT_TLS	Section 2.2.8.5	Specifies the port on which the PMAS PTSMTP Server listens for incoming SMTP-over-TLS connections.
PTSMTP_MAILSERVER_HOST	Section 2.2.8.3	Specifies the hostname for the system on which the primary SMTP server is running.
PTSMTP_MAILSERVER_HOST_TLS	Section 2.2.8.6	Specifies the hostname for the primary SMTP server that supports TLS.
PTSMTP_MAILSERVER_PORT	Section 2.2.8.4	Specifies the port number for the primary SMTP server.
PTSMTP_MAILSERVER_PORT_TLS	Section 2.2.8.7	Specifies the port number that the primary SMTP server listens to for incoming TLS connections.
PTSMTP_QUARANTINE_REPLY	Section 2.2.8.17	Specifies the SMTP-dialogue message that's sent to the SMTP client when a message is quarantined.
PTSMTP_NORELAY_REPLY	Section 2.2.8.21	Specifies the SMTP-dialogue message that's sent to the SMTP client when a message is rejected due to anti-relay.
PTSMTP_NOUSER_REPLY	Section 2.2.8.22	Specifies the SMTP-dialogue message that's sent to the SMTP client when a message is rejected because the specified address is not a valid local address.
PTSMTP_REJECT_REPLY	Section 2.2.8.18	Specifies the SMTP-dialogue message that's sent to the SMTP client when a message is rejected.
PTSMTP_TLS_PROTOCOLS	Section 2.2.8.10	Specifies the supported TLS protocols.
PTSMTP_TLS_CIPHERS	Section 2.2.8.11	Specifies the supported TLS ciphers.
PTSMTP_TLS_PRIVATE_CERT	Section 2.2.8.12	Specifies the filename of the TLS private certificate.
PTSMTP_TLS_PUBLIC_CERT	Section 2.2.8.13	Specifies the filename of the TLS public certificate.
PTSMTP_WORKER_MIN	Section 2.2.8.14	Specifies the number of permanent PTSMTP worker processes that run to handle incoming SMTP connections.
PTSMTP_WORKER_MAX	Section 2.2.8.15	Specifies the maximum number of PTSMTP worker processes that can run at any one time to handle incoming SMTP connections.
QUARANTINE_INDEX_FILE	Section 2.2.3.4	Specifies the name of the quarantine index file
QUARANTINE_MESSAGES	Section 2.2.3.1	Controls the quarantining of messages
QUARANTINE_MSG_LIFETIME	Section 2.2.3.3	The number of days quarantined messages are kept
QUARANTINE_RFC822_FROM	Section 2.2.3.5	Controls which From: address is used for quarantine notification

Table 2–1 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Listed Alphabetically

Keyword	Section	Description
QUARANTINE_RFC822_TO	Section 2.2.3.6	Controls which To: address is displayed in quarantine notifications
QUARANTINE_THRESHOLD	Section 2.2.3.2	The score threshold for quarantining messages
PMAS_PROCESS_DEBUG_LEVEL	Section 2.2.16.3	The debug message output level for the PreciseMail Anti-Spam Gateway Processor
PMAS_PROCESS_DEBUG_LOGFILE	Section 2.2.16.4	The name of the Processor debug log file
RCVD_DNSBL_ENABLED	Table 3–6	Enables Received: DNSBL lookups
RCVD_DNSBL_SCORE	Table 3–6	Specifies the score applied to positive Received: DNSBL lookups
RCVD_DNSBL_LOG_ENABLED	Table 3–6	Specifies if positive Received: DNSBL lookups are logged
RDNS_URI_ENABLED	Table 3–8	Enables rDNS lookups for URIs
RDNS_URI_SCORE	Table 3–8	Specifies the score applied to URI rDNS failures
RDNS_LOG_ENABLED	Table 3–8	Specifies if URI rDNS failures are logged
REJECT_MESSAGES	Section 2.2.5.1	Enables the rejection of messages by the PMAS PTSMTP Server.
REJECT_THRESHOLD	Section 2.2.5.2	Specifies the spam score threshold at which incoming SMTP messages are rejected.
REPUTATION_URI_ENABLED	Section 2.2.7.1	Enables reputation URI filtering
REP_URI_MULTIPLIER	Section 2.2.7.2	Specifies a multiplier that is applied to the reputation URI results
REP_URI_NONSPAM_EFFECTS	Section 2.2.7.3	Causes non-spam reputation URI results to affect the score
REP_URI_PHISH_SCORE	Section 2.2.7.4	Additional score for a phishing site
REP_URI_ADULT_SCORE	Section 2.2.7.5	Additional score for an adult site
SEND_QUARNOTICES_DEFAULT	Section 2.2.15.1	Controls whether or not quarantine notices are sent to users by default.
SMTP_MAILSERVER_HOST	Section 2.2.8.19	Specifies the SMTP server hostname for messages generated by PMAS.
SMTP_MAILSERVER_PORT	Section 2.2.8.20	Specifies the listen port for the SMTP Server that receives messages generated by PMAS.
SPAM_LEVEL_CHAR	Section 2.2.2.11	Specifies the character used for X-PMAS-Spam-Level header
SPAM_LEVEL_STARS	Section 2.2.2.10	Controls if X-PMAS-Spam-Level header is added
STATS_USE_THRESHOLDS	Section 2.2.4.4	Enables pmas_stats program to use configuration file threshold values when not quarantining or discarding
SUBJECT_TAG	Section 2.2.2.9	Specifies the text prepended to modified Subject: lines
UPDATES_CONNECT_TIMEOUT	Section 2.2.17.5	Specifies a connect timeout for rule update checks
UDNS_NAMESERVER	Section 2.2.6.11	Specifies the IP address of the nameserver to use for DNS lookups
UDNS_DEBUG	Section 2.2.6.11	Turns on debugging output for the uDNS routines
URI_DNSBL_ENABLED	Table 3–4	Enables URI DNSBL lookups
URI_DNSBL_SCORE	Table 3–4	Specifies the score applied to positive URI DNSBL lookups
URI_DNSBL_LOG_ENABLED	Table 3–4	Specifies if positive URI DNSBL lookups are logged

Table 2–1 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Listed Alphabetically

Keyword	Section	Description
USE_BAYESIAN	Section 2.2.12.1	Enables or disables use of the Bayesian engine
USE_CURRENT_ENVELOPE_TO	Section 2.2.1.1	Selects which envelope to address is used
USER_DATABASE	Section 2.2.10.16	The name of the PMAS user preferences database
USERLIST_SUBDIR_LEVEL	Section 2.2.14.4	Specifies the number of subdirectories used for storing user allow and block list files
VERIFY_MAIL_FROM_ADDRESSES	Section 2.2.6.1	Enables verification of envelope MAIL FROM: addresses
VERIFY_MAIL_FROM_TIMEOUT	Section 2.2.6.2	Specifies the timeout in seconds for MAIL FROM: verification connections
VIRUS_DISPOSITION	Section 2.2.9.4	Specifies the action to take for virus-laden messages
VMF_CONNECT_TIMEOUT	Section 2.2.6.10	Specifies a connect timeout for VMF checks
VMF_ERROR_SCORE	Section 2.2.6.3	Score assigned on a MAIL FROM: verification error
VMF_NO_DNS_SCORE	Section 2.2.6.4	Score assigned for no DNS entry during MAIL FROM: verification
VMF_NO_ADDRESS_SCORE	Section 2.2.6.5	Score assigned for rejection during MAIL FROM: verification
VMF_NO_MAILFROM_SCORE	Section 2.2.6.6	Score assigned for invalid address during MAIL FROM: verification
VMF_NO_MX_SCORE	Section 2.2.6.7	Score assigned for no MX record during MAIL FROM: verification
VMF_NO_SMTP_SCORE	Section 2.2.6.8	Score assigned for no SMTP server during MAIL FROM: verification
VMF_OK_SCORE	Section 2.2.6.9	Score assigned for accepted address during MAIL FROM: verification

Table 2–2 PreciseMail Anti-Spam Gateway Configuration Keywords Grouped Functionally

Keyword	Section	Description
	Modifying Messages	
ADD_HEADERS	Section 2.2.2.1	Controls if X-PMAS headers are added
ADD_SPAM_YES_HEADER	Section 2.2.2.2	Controls if ‘‘X-PMAS-Spam: Yes’’ header is added
ADD_SPAM_YES_THRESHOLD	Section 2.2.2.3	Threshold for adding ‘‘X-PMAS-Spam: Yes’’ header
HEADER_PREFIX	Section 2.2.2.4	The prefix appended to message headers added by PMAS
MODIFY_SUBJECT	Section 2.2.2.5	Controls if Subject: lines are modified
MODIFY_SUBJECT_APPEND	Section 2.2.2.6	Appends the subject_tag to Subject: lines
MODIFY_SUBJECT_THRESHOLD	Section 2.2.2.7	The score threshold for modifying Subject: lines
PMAS_SYSTEM_NAME	Section 2.2.2.8	The system name added to the X-PMAS-Software: header
SUBJECT_TAG	Section 2.2.2.9	Specifies the text prepended to modified Subject: lines
SPAM_LEVEL_STARS	Section 2.2.2.10	Controls if X-PMAS-Spam-Level header is added
SPAM_LEVEL_CHAR	Section 2.2.2.11	Specifies the character used for X-PMAS-Spam-Level header

Table 2–2 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Grouped Functionally

Keyword	Section	Description
Quarantining Messages		
QUARANTINE_INDEX_FILE	Section 2.2.3.4	Specifies the name of the quarantine index file
QUARANTINE_MESSAGES	Section 2.2.3.1	Controls the quarantining of messages
QUARANTINE_MSG_LIFETIME	Section 2.2.3.3	The number of days quarantined messages are kept
QUARANTINE_RFC822_FROM	Section 2.2.3.5	Controls which From: address is used for quarantine notification
QUARANTINE_RFC822_TO	Section 2.2.3.6	Controls which To: address is displayed in quarantine notifications
QUARANTINE_THRESHOLD	Section 2.2.3.2	The score threshold for quarantining messages
Discarding Messages		
DISCARD_MESSAGES	Section 2.2.4.1	Controls the discarding of messages
DISCARD_THRESHOLD	Section 2.2.4.2	The score threshold for discarding messages
DISCARD_MSG_LIFETIME	Section 2.2.4.3	The number of days discarded messages are kept
Rejecting Messages (PMAS PTSMTP only)		
REJECT_MESSAGES	Section 2.2.5.1	Enables the rejection of messages by the PMAS PTSMTP Server.
REJECT_THRESHOLD	Section 2.2.5.2	Specifies the spam score threshold at which incoming SMTP messages are rejected.
BLOCK_DISPOSITION	Section 2.2.5.3	Determines the actual disposition of messages that are blocked.
BLOCK_SCORE	Section 2.2.5.4	Specifies the score assigned to messages that are blocked. May be adjusted to reject blocked messages instead of discarding them.
MAXIMUM_USER_BLOCK_RULES	Section 2.2.5.5	Specifies the maximum number of user block rules that will be applied to messages

Table 2–2 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Grouped Functionally

Keyword	Section	Description
	User Interface Keywords	
GUI_COOKIE_LIFETIME	Section 2.2.10.1	Specifies the lifetime in hours of the PMAS login cookie
GUI_URI_HOST	Section 2.2.10.2	Specifies the host name used in PMAS URLs
GUI_URI_USE_HTTP_HOST	Section 2.2.10.3	Specifies which host name is used in PMAS URLs
GUI_URI_PATH	Section 2.2.10.5	Specifies the path name for PMAS URLs
GUI_URI_PROTOCOL	Section 2.2.10.6	Specifies the protocol for the served pages
GUI_URI_SCRIPT_PATH	Section 2.2.10.4	Specifies the script path name for PMAS URLs
GUI_ALLOW_QUARALL_DEFAULT	Section 2.2.10.8	Allow users to set their default quarantined message display to ‘‘all’’
GUI_ALLOW_QUARANTINE_ALL	Section 2.2.10.7	Allow users to display all their quarantined messages
GUI_DEFAULT_QUARANTINE_ALL	Section 2.2.10.9	Change the default display for quarantined messages to ALL
GUI_QUARANTINE_DISPLAY_DAILY	Section 2.2.10.10	Change the default display for quarantined messages to daily
GUI_DELETE_UPON_RELEASE	Section 2.2.10.11	Causes released messages to be deleted from the quarantine
GUI_FORCE_JAVA_SORT_OPERA	Section 2.2.10.15	Forces usage of the Javascript quarantine sort with the Opera browser
GUI_JAVA_SORT_MAXMSG	Section 2.2.10.14	Specifies the maximum number of messages for a quarantine view in which the Javascript sort is used
GUI_RENAME_UPON_DELETE	Section 2.2.10.12	Causes deleted messages to actually be renamed to _DELETED
GUI_ALLOW_DISCARD_VIEW	Section 2.2.10.13	Allow users to view their discarded messages
USER_DATABASE	Section 2.2.10.16	The name of the PMAS user preferences database
ADMIN_EMAIL_ADDRESS	Section 2.2.10.17	The system administrator email address
OPT_IN_BY_DEFAULT	Section 2.2.10.18	Specifies the ‘‘opt-in’’ default for users
ALLOW_USER_OPTIN	Section 2.2.10.19	Allow users to opt in or out
ALLOW_USER_DISCARD	Section 2.2.10.20	Allow users to discard messages
ALLOW_USER_QUARANTINE	Section 2.2.10.21	Allow users to quarantine messages
ALLOW_USER_TAGGING	Section 2.2.10.22	Allow users to tag messages

Table 2–2 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Grouped Functionally

Keyword	Section	Description
	User Authentication Keywords	
AUTH_METHODS	Section 2.2.11.1	Lists the available authorization methods
AUTH_POP3_HOSTS	Section 2.2.11.2	List of POP3 authentication hosts
POP3_CONNECT_TIMEOUT	Section 2.2.11.3	Specifies a connect timeout for POP3 authorization requests
AUTH_IMAP4_HOSTS	Section 2.2.11.4	List of IMAP4 authentication hosts
IMAP4_CONNECT_TIMEOUT	Section 2.2.11.5	Specifies a connect timeout for IMAP4 authorization requests
LDAP_AUTH_FILTER	Section 2.2.11.8	The LDAP search filter
LDAP_AUTH_SERVER	Section 2.2.11.6	LDAP server name
LDAP_BASE_DN	Section 2.2.11.7	The LDAP base distinguished name
LDAP_GROUP_FILTER	Section 2.2.11.9	The LDAP search filter for user groups
LDAP_GROUPMEMBER_ATTR	Section 2.2.11.10	The name of the attribute that contains a group member’s DN
LDAP_SEARCHACCT_DN	Section 2.2.11.11	The Distinguished Name of the LDAP search user
LDAP_SEARCHACCT_PASSWORD	Section 2.2.11.12	The password for the LDAP search user
LDAP_USE_TLS	Section 2.2.11.13	Enables use of LDAPS for user authentication
	Bayesian Engine keywords	
USE_BAYESIAN	Section 2.2.12.1	Enables or disables use of the Bayesian engine
AUTOTRAIN_BAYESIAN	Section 2.2.12.2	Controls the autotraining of the Bayesian engine
AUTOBAYESIAN_HAM_THRESHOLD	Section 2.2.12.3	Autotraining for ham threshold
AUTOBAYESIAN_SPAM_THRESHOLD	Section 2.2.12.4	Autotraining for spam threshold
BAYESIAN_MULTIPLIER	Section 2.2.12.5	Multiplier for the Bayesian spamicity rating

Table 2–2 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Grouped Functionally

Keyword	Section	Description
	MAIL FROM: verification keywords	
VERIFY_MAIL_FROM_ADDRESSES	Section 2.2.6.1	Enables verification of envelope MAIL FROM: addresses
VERIFY_MAIL_FROM_TIMEOUT	Section 2.2.6.2	Specifies the timeout in seconds for MAIL FROM: verification connections
VMF_CONNECT_TIMEOUT	Section 2.2.6.10	Specifies a connect timeout for VMF checks
VMF_ERROR_SCORE	Section 2.2.6.3	Score assigned on a MAIL FROM: verification error
VMF_NO_DNS_SCORE	Section 2.2.6.4	Score assigned for no DNS entry during MAIL FROM: verification
VMF_NO_ADDRESS_SCORE	Section 2.2.6.5	Score assigned for rejection during MAIL FROM: verification
VMF_NO_MAILFROM_SCORE	Section 2.2.6.6	Score assigned for invalid address during MAIL FROM: verification
VMF_NO_MX_SCORE	Section 2.2.6.7	Score assigned for no MX record during MAIL FROM: verification
VMF_NO_SMTP_SCORE	Section 2.2.6.8	Score assigned for no SMTP server during MAIL FROM: verification
VMF_OK_SCORE	Section 2.2.6.9	Score assigned for accepted address during MAIL FROM: verification
UDNS_NAMESERVER	Section 2.2.6.11	Specifies the IP address of the nameserver to use for VMF DNS lookups
UDNS_DEBUG	Section 2.2.6.11	Turns on debugging output for the uDNS routines
	Reputation URI filtering keywords	
REPUTATION_URI_ENABLED	Section 2.2.7.1	Enables reputation URI filtering
REP_URI_MULTIPLIER	Section 2.2.7.2	Specifies a multiplier that is applied to the reputation URI results
REP_URI_NONSPAM_EFFECTS	Section 2.2.7.3	Causes non-spam reputation URI results to affect the score
REP_URI_PHISH_SCORE	Section 2.2.7.4	Specific score for a phishing site
REP_URI_ADULT_SCORE	Section 2.2.7.5	Specific score for an adult site
	URI DNSBL lookup keywords	
URI_DNSBL_ENABLED	Table 3–4	Enables URI DNSBL lookups
URI_DNSBL_SCORE	Table 3–4	Specifies the score applied to positive URI DNSBL lookups
URI_DNSBL_LOG_ENABLED	Table 3–4	Specifies if positive URI DNSBL lookups are logged
UDNS_NAMESERVER	Section 2.2.6.11	Specifies the IP address of the nameserver to use for URI DNSBL DNS lookups

Table 2–2 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Grouped Functionally

Keyword	Section	Description
	Received: DNSBL lookup keywords	
RCVD_DNSBL_ENABLED	Table 3–6	Enables Received: DNSBL lookups
RCVD_DNSBL_SCORE	Table 3–6	Specifies the score applied to positive Received: DNSBL lookups
RCVD_DNSBL_LOG_ENABLED	Table 3–6	Specifies if positive Received: DNSBL lookups are logged
	rDNS URI lookup keywords	
RDNS_URI_ENABLED	Table 3–8	Enables rDNS lookups for URIs
RDNS_URI_SCORE	Table 3–8	Specifies the score applied to URI rDNS failures
RDNS_LOG_ENABLED	Table 3–8	Specifies if URI rDNS failures are logged
UDNS_NAMESERVER	Section 2.2.6.11	Specifies the IP address of the nameserver to use for rDNS URI DNS lookups

Table 2–2 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Grouped Functionally

Keyword	Section	Description
	Pass-Through SMTP Server (PTSMTP)	
PTSMTP_ENABLE_PMDF_XREM	Section 2.2.8.8	Enables support for the PMDF XREM extension.
PTSMTP_ENABLE_STARTTLS	Section 2.2.8.9	Enables STARTTLS support in the PMAS PTSMTP server.
PTSMTP_IDLE_TIME	Section 2.2.8.16	Specifies the number of seconds a temporary PMAS PTSMTP process is idle before it exits.
PTSMTP_LISTEN_HOST	Section 2.2.8.1	Specifies the interface IP addresses on which the PMAS PTSMTP Server listens for incoming SMTP connections.
PTSMTP_LISTEN_PORT	Section 2.2.8.2	Specifies the port on which the PMAS PTSMTP Server listens for incoming SMTP connections.
PTSMTP_MAILSERVER_HOST	Section 2.2.8.3	Specifies the hostname for the system on which the primary SMTP server is running.
PTSMTP_MAILSERVER_PORT	Section 2.2.8.4	Specifies the port number for the primary SMTP server.
PTSMTP_LISTEN_PORT_TLS	Section 2.2.8.5	Specifies the port on which the PMAS PTSMTP Server listens for incoming SMTP-over-TLS connections.
PTSMTP_MAILSERVER_HOST_TLS	Section 2.2.8.6	Specifies the hostname for the primary SMTP server that supports TLS.
PTSMTP_MAILSERVER_PORT_TLS	Section 2.2.8.7	Specifies the port number that the primary SMTP server listens to for incoming TLS connections.
PTSMTP_NORELAY_REPLY	Section 2.2.8.21	Specifies the SMTP-dialogue message that's sent to the SMTP client when a message is rejected due to anti-relay.
PTSMTP_NOUSER_REPLY	Section 2.2.8.22	Specifies the SMTP-dialogue message that's sent to the SMTP client when a message is rejected because the specified address is not a valid local address.
PTSMTP_QUARANTINE_REPLY	Section 2.2.8.17	Specifies the SMTP-dialogue message that's sent to the SMTP client when a message is quarantined.
PTSMTP_REJECT_REPLY	Section 2.2.8.18	Specifies the SMTP-dialogue message that's sent to the SMTP client when a message is rejected.
PTSMTP_TLS_PRIVATE_CERT	Section 2.2.8.12	Specifies the filename of the TLS private certificate.
PTSMTP_TLS_PUBLIC_CERT	Section 2.2.8.13	Specifies the filename of the TLS public certificate.
PTSMTP_WORKER_MIN	Section 2.2.8.14	Specifies the number of permanent PTSMTP worker processes that run to handle incoming SMTP connections.
PTSMTP_WORKER_MAX	Section 2.2.8.15	Specifies the maximum number of PTSMTP worker processes that can run at any one time to handle incoming SMTP connections.
SMTP_MAILSERVER_HOST	Section 2.2.8.19	Specifies the SMTP server hostname for messages generated by PMAS.
SMTP_MAILSERVER_PORT	Section 2.2.8.20	Specifies the listen port for the SMTP Server that receives messages generated by PMAS.

Table 2–2 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Grouped Functionally

Keyword	Section	Description
Anti-Virus (via the PTSMTP Proxy Server)		
ANTIVIRUS_ENABLED	Section 2.2.9.1	Enables anti-virus scanning in PTSMTP proxy server
ANTIVIRUS_DIR	Section 2.2.9.3	Specifies the location of the anti-virus data files
ANTIVIRUS_PACKAGE	Section 2.2.9.2	Specifies the name of the anti-virus package
VIRUS_DISPOSITION	Section 2.2.9.4	Specifies the action to take for virus-laden messages
Automatic Ruleset and AV Database Updates		
AUTOUPDATE_ADDRESS	Section 2.2.17.4	Specifies email address used for email-based updates
AUTOUPDATE_RULES	Section 2.2.17.1	Controls whether or not automatic ruleset updates are performed
AUTOUPDATE_SOPHOS	Section 2.2.17.2	Controls whether or not Sophos AV database updates are performed
AUTOUPDATE_STATS	Section 2.2.17.3	Controls the uploading of spam-related statistical data to Process Software
Data Synchronization Cluster		
CLU_CLIENTS	Section 2.2.18.1	Specifies secondary nodes of a Data Synchronization Cluster.
CLU_SERVER	Section 2.2.18.2	Specifies the primary node of a Data Synchronization Cluster.
CLU_SERVER_CLIENT	Section 2.2.18.3	Enables Data Synchronization Cluster functionality.
Debugging		
DEBUG_LEVEL	Section 2.2.13.1	The debug message output level
DEBUG_LOGFILE	Section 2.2.13.2	Specifies the name of the debug log file
Miscellaneous		
ALIAS_FILE	Section 2.2.14.1	Specifies the address alias filename
LOCAL_DOMAIN_NAME	Section 2.2.1.2	Specifies the default local domain name
MASTER_LOGFILE	Section 2.2.14.3	Specifies the master log file name
STATS_USE_THRESHOLDS	Section 2.2.4.4	Enables pmas_stats program to use configuration file threshold values when not quarantining or discarding
USE_CURRENT_ENVELOPE_TO	Section 2.2.1.1	Selects which envelope to address is used
USERLIST_SUBDIR_LEVEL	Section 2.2.14.4	Specifies the number of subdirectories used for storing user allow and block list files

Table 2–2 (Cont.) PreciseMail Anti-Spam Gateway Configuration Keywords Grouped Functionally

Keyword	Section	Description
	Quarantine Notification Batch Job	
MIME_QUARANTINE_MESSAGE	Section 2.2.15.2	Controls the type of quarantine notification messages
NOTIFY_DEBUG_LEVEL	Section 2.2.15.4	The debug message level for the notify job
NOTIFY_MSG_SUBJECT	Section 2.2.15.3	Specifies the Subject: line for quarantine notification messages
SEND_QUARNOTICES_DEFAULT	Section 2.2.15.1	Controls whether or not quarantine notices are sent to users by default.
	PreciseMail Anti-Spam Gateway Processor	
PROCESSOR_USER_ADDRESS	Section 2.2.16.2	The full email address for the PreciseMail Anti-Spam Gateway Processor
PROCESSOR_USER_NAME	Section 2.2.16.1	The local username for the PreciseMail Anti-Spam Gateway Processor
PMAS_PROCESS_DEBUG_LEVEL	Section 2.2.16.3	The debug message output level for the PreciseMail Anti-Spam Gateway Processor
PMAS_PROCESS_DEBUG_LOGFILE	Section 2.2.16.4	The name of the Processor debug log file
MIME_HELP_MESSAGE	Section 2.2.16.5	Specifies that the HELP reply is a MIME message

2.2.1 Choosing Envelope Addresses

By default, PreciseMail Anti-Spam Gateway uses the *original* envelope “To:” address when looking up aliases and user allow/block rules. This address is the original address specified for the message; in PMDF mail logs, this is the address shown prefixed by “rfc822;”. Using the keywords described below, PMAS can be instructed to use the *current* envelope “To:” address, which is the envelope address after it has possibly been modified by the mailing software (in PMDF logs, this is the address shown after the original envelope “To:”; for a delivery to an “l”, or local, channel, this address will consist of only a username.

2.2.1.1 USE_CURRENT_ENVELOPE_TO keyword

Specifies whether the original current envelope “To:” addresses are used by PreciseMail Anti-Spam Gateway. The default value is *NO*, which means that the original envelope “To:” addresses will be used for alias and user allow/block lookups. Setting this keyword’s value to *YES* will cause the current envelope “To:” address to be used for alias and user allow/block lookups.

2.2.1.2 LOCAL_DOMAIN_NAME keyword

Specifies the local domain name to be supplied for current envelope "To:" addresses that consist only of usernames (i.e, local deliveries). If USE_CURRENT_ENVELOPE_TO is set to *YES*, the address for local deliveries may not include a domain name (e.g, just "POSTMASTER"). The user allow/block file lookup in that case would be /pmas/user_rules/postmaster. You can define LOCAL_DOMAIN_NAME to specify a domain name that should be appended to all current envelope "To:" addresses that do not have a domain name specified.

Note: If the *local_domain_name* keyword is defined, its value will also be used as the default domain name for users logging in to the PreciseMail Anti-Spam Gateway User Interface. Full email addresses should be supplied; if a user supplies only a username and this keyword is defined, its value will be combined with the username to form a complete email address.

2.2.2 Modifying Messages

The following keywords control how messages may be modified by PreciseMail Anti-Spam Gateway.

2.2.2.1 ADD_HEADERS keyword

Defined as *yes* (the default) or *no*. If enabled, X-PMAS-* headers are added to each message that has a non-zero score, whether it is ham (non-spam) or spam. If disabled, no headers are added to messages. Headers are added to messages that are both forwarded and quarantined.

Note: Even with ADD_HEADERS set to "no", X-PMAS-Software: and X-PMAS-Internal: or X-PMAS-External: headers will still be added to messages. The first indicates that the message was actually processed by PMAS. The -Internal and -External headers are added and used by PMAS PTSMTP to indicate whether or not the sending system was internal or external to your network.

2.2.2.2 ADD_SPAM_YES_HEADER keyword

Some sites prefer to add a special header if a message scores above a certain spam threshold. By setting this variable to "yes", PMAS will add a header like the following to all messages with a score higher than the value of ADD_SPAM_YES_THRESHOLD. The default value is "no".

2.2.2.3 ADD_SPAM_YES_THRESHOLD keyword

If ADD_SPAM_YES_HEADER is defined as "yes", this variable specifies the score threshold for determining whether or not an "X-PMAS-Spam: Yes" header is added to messages.

If "X-PMAS-Spam: yes" is added, PMAS also adds an additional header, "X-Auto-Response-Suppress:". This header causes Microsoft Exchange to suppress "Out of Office" messages when the message is delivered.

2.2.2.4 HEADER_PREFIX keyword

Specifies the prefix added to message headers added to messages. All headers begin, by default, with "X-PMAS-". You can change the string "PMAS" to some other string using this variable. The string is case-sensitive, so "Spam" and "SPAM" will generate "X-Spam" and "X-SPAM", respectively.

2.2.2.5 MODIFY_SUBJECT keyword

Defined as *yes* or *no* (the default). If yes, the subject line of all non-quarantined messages that have a score of at least that specified by MODIFY_SUBJECT_THRESHOLD will be modified to indicate that the message is spam.

2.2.2.6 MODIFY_SUBJECT_APPEND keyword

Defined as *yes* or *no* (the default). When MODIFY_SUBJECT is defined as *yes*, the text defined by SUBJECT_TAG is prepended to the subject line. If this keyword is defined as *yes*, the text will be appended to the Subject: line instead.

2.2.2.7 MODIFY_SUBJECT_THRESHOLD keyword

Specifies the score threshold at which messages Subject: lines are modified. The default value is 3.0.

2.2.2.8 PMAS_SYSTEM_NAME keyword

Specifies the name of the system running PMAS. If defined, the value of PMAS_SYSTEM_NAME is shown in the X-PMAS-Software: header that's added to email processed by PMAS. This allows you to determine which system processed the message.

2.2.2.9 SUBJECT_TAG keyword

Specifies the text that is prepended to modified Subject: lines. The default tag is "[SPAM]".

There are two special sequences supported for the Subject: tag.

- The final score for the message can be included in the tag using the text "%SCORE%". If included in the subject tag, that text is replaced with the final score:

  ```
  subject_tag    [SPAM: %SCORE%]
  ```

  produces a header like the following:

  ```
  Subject: [SPAM: 4.555] Your spam message
  ```

  The variable "%SCOREINT%" can be used to show only the integer portion of the score.

- The "spam level" can be included in the tag using the text sequence "%LEVEL%". If this sequence appears in the tag, it is replaced by a sequence of characters (by default "*"; the character can be changed with the SPAM_LEVEL_CHAR keyword (see Section 2.2.2.11). The number of SPAM_LEVEL_CHAR characters inserted matches the score of the message. For example, a message with a score of 3.3 and the following subject_tag definition:

```
subject_tag   [SPAM %LEVEL%]
```

would result in a subject prefix or suffix that looked like:

```
[SPAM ***]
```

Both “%SCORE%” and “%LEVEL%” can be specified in the same Subject: tag.

2.2.2.10 SPAM_LEVEL_STARS keyword

Specifies whether or not an X-PMAS-SPAM-LEVEL header is added to messages. The header consists of a string of repeated characters (“*” by default), with one character per whole unit score. For example, a message with a final score of 4.3 would receive an X-PMAS-SPAM-LEVEL header that looks like this:

```
X-PMAS-Spam-Level: ****
```

The inclusion of this header makes it easier to configure mail clients to filter messages based on the score. For example, messages with a score of 3 or higher can be filtered by checking for an X-PMAS-SPAM-LEVEL header that contains at least 3 stars. This feature is turned on by default.

2.2.2.11 SPAM_LEVEL_CHAR keyword

Specifies the character that is to be repeated for the X-PMAS-SPAM-LEVEL header. The default character is “*”.

2.2.3 Quarantining Messages

The following keywords control the quarantining of messages.

2.2.3.1 QUARANTINE_MESSAGES keyword

Defined as *yes* (the default) or *no*. If enabled, messages with a final score equal to or higher than the score defined by QUARANTINE_THRESHOLD are automatically quarantined on behalf of the user.

2.2.3.2 QUARANTINE_THRESHOLD keyword

Specifies the numeric score for determining whether or not a message should be quarantined. The default value is 5.0. Messages with a final score of 5.0 or higher will be quarantined.

2.2.3.3 QUARANTINE_MSG_LIFETIME keyword

Specifies the number of days that quarantined messages are kept on the system. The default value is 14 days. Quarantined messages are automatically deleted from the system after the specified number of days has elapsed. (This deletion is performed by the PreciseMail Anti-Spam Gateway Notification batch job.)

2.2.3.4 QUARANTINE_INDEX_FILE keyword

Specifies the name of the quarantine index file. The default filename is /pmas/quarantine/pmas_quarantine.dat. This index file contains a line for each quarantined message. It is used by the quarantine notification batch job to create the notification messages that are mailed to users.

2.2.3.5 QUARANTINE_RFC822_FROM keyword

Defined as *yes* (the default) or *no*. Indicates whether or not the RFC822 From: address is stored in the quarantine index file (and is thus used in quarantine notification messages). If defined as yes, the entire RFC822 From: line is used. If defined as no, the envelope FROM address is saved (which may be a more accurate From: address, but does not include personal names).

2.2.3.6 QUARANTINE_RFC822_TO keyword

Defined as *yes* (the default) or *no*. Indicates whether or not the RFC822 To: address is stored in the quarantine index file (and is thus displayed in quarantine notification messages). If defined as yes, the entire RFC822 To: line is used. If defined as no, the original envelope TO address is displayed (which may be a more accurate To: address).

2.2.4 Discarding Messages

The following keywords control the discarding of messages.

2.2.4.1 DISCARD_MESSAGES keyword

Defined as *yes* or *no* (the default). If enabled, messages with a final score equal to or higher than the score defined by DISCARD_THRESHOLD are automatically discarded on behalf of the user.

Note: As described in Section 1.6.2, messages are not actually discarded by default, but are instead copied to the discard directory.

2.2.4.2 DISCARD_THRESHOLD keyword

Specifies the numeric score for determining whether or not a message should be discarded. The default value is 50.0. Messages with a final score of 50.0 or higher will be discarded if DISCARD_MESSAGES is defined as *yes*.

2.2.4.3 DISCARD_MSG_LIFETIME keyword

Specifies the number of days that discarded messages are kept on the system. The default value is 14 days. Discarded messages are automatically deleted from the system after the specified number of days has elapsed. (This deletion is performed by the PreciseMail Anti-Spam Gateway Notification batch job.)

2.2.4.4 STATS_USE_THRESHOLDS keyword

Enables the pmas_stats program to use configuration file threshold values when not quarantining or discarding. Defined as yes (the default) or no. By default, the pmas_stats program will use the quarantine and discard threshold values when reporting how many messages were quarantined and discarded. If you are quarantining but not discarding and would like the statistics to accurately reflect that, set this variable to no.

2.2.5 Rejecting Messages

The following keywords control the rejecting of messages when the PMAS Pass-Through SMTP Server is used.

2.2.5.1 REJECT_MESSAGES keyword

Defined as *yes* or *no* (the default). If enabled, messages with a final score equal to or higher than the score defined by REJECT_THRESHOLD are automatically rejected during the SMTP session.

Note: Messages are only rejected if there is only one recipient or if all recipients reject the message.

2.2.5.2 REJECT_THRESHOLD keyword

Specifies the numeric score for determining whether or not a message should be rejected. The default value is 200.0. Messages with a final score of 200.0 or higher will be rejected if REJECT_MESSAGES is defined as *yes*.

2.2.5.3 BLOCK_DISPOSITION keyword

Specifies the disposition of messages blocked by a blocklist entry. By default, the variable has a value of zero (0) and blocked messages are simply deleted—the user never receives any indication that a message was received, and the sender is not notified that the message was deleted. For PMAS PTSMTP sites, when the block_score is greater than or equal to the reject_threshold, blocked messages are rejected during the SMTP session. If block_disposition is set to a value of 1 and discard_messages is enabled, blocked messages will be discarded instead of just silently deleted. A value of 2 indicates that blocked message should be rejected during the SMTP session. Finally, a value of 3 indicates that blocked messages should be quarantined.

2.2.5.4 BLOCK_SCORE keyword

Specifies the numeric score assigned to messages that are blocked via a system- or user-blocklist. The default value is 200.0. You can adjust this score to determine whether or not blocklisted messages are rejected or just silently thrown away. By default, blocklisted will receive a score of 200.0 and be rejected if REJECT_MESSAGES is defined as *yes*.

2.2.5.5 MAXIMUM_USER_BLOCK_RULES keyword

Specifies the maximum number of user block rules that will be read and applied to messages. The default value is 1000. This keyword is designed to counteract the impact on processing time caused by users who systematically block email addresses on all spam they receive, ending up with thousands of block rules in their user rules files.

2.2.6 Envelope MAIL FROM: Verification keywords

The PreciseMail Anti-Spam engine can attempt to check the validity of envelope MAIL FROM: addresses and assign scores to messages based on the results of those checks. The envelope MAIL FROM: address, also known as the sender or return address, is often forged for spam messages and frequently specifies an email address that is not a valid address (i.e, you can't actually send mail to the address). Enabling MAIL FROM: verification checks causes PMAS to attempt to see if the address is valid or not.

During the verification process, the PMAS engine attempts to verify the address by establishing an SMTP connection to the mail server for the address and issuing the commands to send mail to the address. Once the status of the RCPT TO: command for the address is received, the SMTP connection is dropped, and the resulting status determines what score is assigned to the message.

Note: The MAIL FROM: address used when PMAS establishes the SMTP connection to the remote server is "smtp-check@node", where "node" is defined by the configuration variable LOCAL_DOMAIN_NAME**.**

To avoid false positives from addresses that are legitimate but intentionally don't exist (for example, a return address that begins with "donotreply@"), a VMF exceptions file can be used. The file /pmas/data/vmf_exceptions.txt contains wildcard or regular expressions strings that specify addresses for which VMF should not be used. A template file is provided.

The following list specifies some of the issues that must be considered before enabling MAIL FROM: address verification.

- The PMAS engine contacts the primary MX server for each MAIL FROM: address that is seen (and that isn't allowed or blocked by other rules) to verify the address. The amount of time this connection takes depends on the speed of your system, the speed of your Internet connection, the speed of the Internet systems between your system and the target MX server, and the speed of the target MX server.

  On average, the entire process takes from 1 to 15 seconds of elapsed time, depending on whether or not the address can be checked and how long it takes to establish the SMTP connection. The configuration variable VMF_CONNECT_TIMEOUT controls how long an attempt will be made to connect to the target MX server. The default value is 30 seconds.

  Note: The delays incurred while doing this lookup will happen for every message that's scanned by the PMAS engine. If your site has a very high volume of email traffic, you may not be able to afford the delays incurred by these tests.

- The MAIL FROM: verification check is very reliable for determining that an address is invalid, but does not guarantee that an address is valid. The "invalid address" status is returned when the remote SMTP server returns an error status in response to the RCPT TO: command. Many systems are configured to accept all addresses, valid or not, and will generate bounce messages after the SMTP session if the address is not valid. Therefore, a rejection of the address on the RCPT TO: command conclusively states that the address is invalid, but an acceptance does not necessarily mean that the address is real. This means that the MAIL FROM: verification process can be used to punish a message that is from an invalid address, but cannot be used to reward (via a negative score) a message that is from an address that is accepted by the remote SMTP server.

Note: If you are not running a nameserver on the local system (127.0.0.1), you must define the PMAS configuration variable UDNS_NAMESERVER **to specify the IP address of the nameserver to use:**

```
udns_nameserver 10.1.1.12
```

The default nameserver used is 127.0.0.1.

The following table details the status conditions that can be returned by the Process Software verification server. For each match for a message, the rule names shown below are assigned to the message, allowing them to be used in meta rules, if so desired.

Rule name	Description
VMF_NO_ADDRESS	The target system rejected the email address as being invalid.
VMF_NO_MX	The target system does not have an MX (mail exchange) record defined. Strictly speaking, this is a violation of the Internet RFCs.
VMF_NO_DNS	The target system does not have a DNS entry.
VMF_NO_SMTP	The target system is not running an SMTP server.
VMF_NO_MAILFROM	The target system rejected the SMTP connection from the Process Software verification system.
VMF_ERROR	Some error occurred internal to the Process Software verification server.

The MAIL FROM: verification checks are controlled by the following configuration variables.

2.2.6.1 VERIFY_MAIL_FROM_ADDRESSES keyword

Defined as *yes* or *no* (the default). Enables verification of envelope MAIL FROM: addresses.

2.2.6.2 VERIFY_MAIL_FROM_TIMEOUT keyword

Specifies the number of seconds to wait before giving up on the MAIL FROM: verification. The default value of 0 causes the Process Software server timeout to be used (30 seconds). Because some systems are slow to respond to new SMTP sessions, a timeout is built into the process to avoid unnecessarily lengthy delays.

2.2.6.3 VMF_ERROR_SCORE keyword

The score assigned to a message when there's an error returned by the VMF check. The error can mean that the Process Software VMF system wasn't reachable, or that there was some error contacting the remote mailer. By default, no penalty is assigned in such cases (the default score is 0.0).

2.2.6.4 VMF_NO_DNS_SCORE keyword

The score assigned to a message when the VMF server does not find a DNS entry for the target domain's mail system. This is a common condition of bogus domain names in addresses. The default score is 5.0.

2.2.6.5 VMF_NO_ADDRESS_SCORE keyword

The score assigned to a message when the remote system rejects the MAIL FROM: address (i.e, the MAIL FROM: address is not a valid address, according to the mail system for that domain). The default score is 8.0.

2.2.6.6 VMF_NO_MAILFROM_SCORE keyword

The score assigned to a message when the target system rejects the SMTP connection from the Process Software VMF server. The default score is 2.5.

2.2.6.7 VMF_NO_MX_SCORE keyword

The score assigned to a message when the VMF finds no MX record for the target domain. All domains should have MX records defined, but not all do. Spammers often don't bother to define MX records because their domains are so transient. The default score is 1.0.

2.2.6.8 VMF_NO_SMTP_SCORE keyword

The score assigned to a message when the VMF server finds that the target system is not running an SMTP server (and thus cannot receive email—i.e, the MAIL FROM: address refers to a system that does not accept incoming email). The default score is 4.0.

2.2.6.9 VMF_OK_SCORE keyword

The score assigned to a message when the address checks out as being OK. By default, no score is assigned in such a case. Negative values are not recommended because the acceptance of the MAIL FROM: address by the remote system is not a guarantee that the address is legitimate; it only means that the remote system will accept the email. The default score is 0.0.

2.2.6.10 VMF_CONNECT_TIMEOUT keyword

Specifies the number of seconds that PMAS will attempt to contact the Process Software server before giving up. By default, VMF connections will wait for 30 seconds. If no connection is made in that time, the connect attempt is aborted.

2.2.6.11 UDNS_NAMESERVER keyword

Specifies the IP address of the nameserver to use for DNS lookups. The default is 127.0.0.1. If you are not running a DNS server on the local system, you must define this variable.

2.2.7 URI reputation filtering keywords

Process Software maintains an active database of several million web sites. Each site and its content is checked for over 20 indicators that it's used by spammers. PreciseMail Anti-Spam Gateway can be configured to check URIs in incoming messages against this database, and use the results to augment a message's score. This system is particularly effective against spam messages that contain a few words of unrelated text and a URL.

During the filtering process, the PMAS engine will query the Process Software data center with every URI in the message. The reputation system returns a spamicity value to the PMAS engine for every URI in the message, as well as information about whether the URI is known to host phishing or adult content. Those values determine which score is assigned to the message.

The URI reputation filtering module is controlled by the following configuration variables.

2.2.7.1 REPUTATION_URI_ENABLED keyword

Defined as *yes* or *no* (the default). Setting this variable to "yes" enables the URI reputation filtering module.

2.2.7.2 REP_URI_MULTIPLIER keyword

The reputation value of a URI is a value between -1 and 1. You can specify a higher multiplier value to make the reputation score have more weight in the final determination if a message is spam or not.

2.2.7.3 REP_URI_NONSPAM_EFFECTS keyword

Defined as *yes* or *no* (the default). By default, the URI reputation filter is only used to increase a message's spamicity. Setting this variable to "yes" will allow a "good" reputation to decrease a message's spamicity. Note that this may cause a significant increase in false negatives for some sites.

2.2.7.4 REP_URI_PHISH_SCORE keyword

If a message contains a known phishing URI, the value of this variable (5.0, by default) will be added to the total message score.

2.2.7.5 REP_URI_ADULT_SCORE keyword

If a message contains a URI for adult content, the value of this variable (1.0, by default) will be added to the total message score.

2.2.7.6 REP_URI_CONNECT_TIMEOUT keyword

Specifies the number of seconds that PMAS will attempt to contact the Process Software data center before giving up. By default, URI reputation connections will wait for 30 seconds. If no connection is made in that time, the connect attempt is aborted.

2.2.8 Pass-Through SMTP Server keywords

The PreciseMail Pass-Through SMTP (PTSMTP) Server acts as a proxy server for all incoming mail. The PTSMTP server does not replace your existing SMTP server, but instead works with your existing SMTP server, passing incoming messages directly to your existing server for delivery. Messages are scanned by the PreciseMail engine as they pass through, and quarantined or discarded messages are never actually sent to your primary SMTP server.

To properly set up the PTSMTP server, you must configure it to run on the well-known SMTP port (port 25) and reconfigure your actual SMTP server to run on an alternate port. SMTP clients will open a connection to the PTSMTP server on port 25, which will in turn open a pass-through connection to your actual SMTP server on its alternate port. Messages will be scanned and diverted or passed through as appropriate according to your PMAS configuration settings. These settings are described in the following sections.

2.2.8.1 PTSMTP_LISTEN_HOST keyword

Specifies the IP address(es) of the interfaces on which the PMAS PTSMTP Server listens for incoming SMTP connections. The default value is “*”, which will cause PMAS PTSMTP to listen on all interfaces.

2.2.8.2 PTSMTP_LISTEN_PORT keyword

Specifies the port on which the PMAS PTSMTP Server listens for incoming SMTP connections. This port has no default value; if you do not specify a port number, the PTSMTP server is not started by PMAS. In a normal configuration, this value will be set to 25, which is the well-known SMTP port as defined by RFC 1700.

2.2.8.3 PTSMTP_MAILSERVER_HOST keyword

Specifies the hostname for the system on which the primary SMTP server is running. When the PTSMTP server accepts an incoming connection, it opens a connection to this host to talk to your primary SMTP server. The default value for this keyword is "127.0.0.1", which is the loopback IP address for the system on which PTSMTP is running. This default value is appropriate for configurations in which both the PTSMTP server and the primary SMTP server are running on the same machine.

This value is used with the value of PTSMTP_MAILSERVER_PORT to establish the pass-through connection between the PTSMTP server and your primary SMTP server.

2.2.8.4 PTSMTP_MAILSERVER_PORT keyword

Specifies the port number for the primary SMTP server. This is the alternate port number on which your primary SMTP server has been configured to listen, as described in Section 2.2.8. This value is used with the value of PTSMTP_MAILSERVER_HOST to establish the pass-through connection between the PTSMTP server and your primary SMTP server.

2.2.8.5 PTSMTP_LISTEN_PORT_TLS keyword

Specifies the TCP port number that the PTSMTP server will listen to for incoming SMTP-over-TLS connections. TLS provides end-to-end encryption of message traffic for security reasons.

2.2.8.6 PTSMTP_MAILSERVER_HOST_TLS keyword

Specifies the hostname for the system on which an SMTP server that supports SMTP-over-TLS is running. Usually, this will be the same hostname that was specified in ptsmtp_mailserver_host.

2.2.8.7 PTSMTP_MAILSERVER_PORT_TLS keyword

Specifies the port number that the SMTP server specified in ptsmtp_mailserver_host_tls listens on for incoming TLS connections. Usually this is TCP port 465, unless you have configured your real SMTP server to listen for TLS connections on a non-standard port.

If this value is undefined or set to 0, the PTSMTP Server's TLS support will be deactivated.

2.2.8.8 PTSMTP_ENABLE_PMDF_XREM keyword

Enables support for the PMDF SMTP extension XREM. This Process Software SMTP extension allows the PMAS PTSMTP server to tell PMDF the IP address of the connecting client system. When PMDF is the backend server, it sees all incoming PTSMTP connections as coming from the IP address of the system running PMAS PTSMTP (usually, localhost or 127.0.0.1). When XREM is enabled in both PMAS and PMDF, the PMAS PTSMTP server sends a special XREM command to the PMDF SMTP server to tell it the actual IP address of the connecting client. From that point forward, as far as PMDF is concerned, it thinks it's dealing with a connection from the client's IP address, allowing all mapping rules, etc, to work as they normally would if the proxy server were not in use. This allows PMDF sites to deploy the PMAS PTSMTP proxy server without changing their PMDF configurations in any way, aside from enabling XREM.

2.2.8.9 PTSMTP_ENABLE_STARTTLS keyword

Enables or disable STARTTLS support in the PTSMTP proxy server. SMTP connections using the default SMTP port (25) that want to use TLS do so by issuing a STARTTLS command during the SMTP dialogue. By default, the PTSMTP proxy server does not support the STARTTLS command. To enable STARTTLS, define this variable as "yes".

2.2.8.10 PTSMTP_TLS_PROTOCOLS keyword

Specifies the TLS protocols that are to be allowed when negotiation TLS connections. The value is a comma-separated list containing one or more of these strings: TLSV1, TLSV1.1, TLSV1.2, TLSV1.3 (Linux-only).

If only TLSV1.2 is specified, clients attempting to connect with TLSv1.1 or lower will not be allowed to connect. As of PMAS V3.3, SSLv2 and SSLv3 are no longer supported.

2.2.8.11 PTSMTP_TLS_CIPHERS keyword

Specifies the TLS ciphers that are to be used when negotiation TLS connections.protocols that are to be allowed. The value is a standard cipher list supported by OpenSSL. For example, the string "ECDH+AESGCM:HIGH" ensures that only more secure ciphers are allowed.

2.2.8.12 PTSMTP_TLS_PRIVATE_CERT keyword

Specifies the filename of the TLS private certificate. The default filename is /pmas/data/server-priv.pem.

For more information about this file, see Section 1.4.4.

2.2.8.13 PTSMTP_TLS_PUBLIC_CERT keyword

Specifies the filename of the TLS public certificate. The default filename is /pmas/data/server-pub.pem.

For more information about this file, see Section 1.4.4.

2.2.8.14 PTSMTP_WORKER_MIN keyword

Specifies the number of permanent PTSMTP worker processes that run to handle incoming SMTP connections. Each incoming SMTP connection is handled by a separate instance of the PTSMTP server. When the PTSMTP server is started, a number of permanent worker processes are created that will be used to service incoming requests. Additional worker processes may be created (up to PTSMTP_WORKER_MAX processes) to handle additional simultaneous requests.

The default value for PTSMTP_WORKER_MIN is 2. For systems handling large email traffic loads, the recommended value for this keyword is a number between 8 and 16, though larger values may be required for extremely large email loads.

2.2.8.15 PTSMTP_WORKER_MAX keyword

Specifies the maximum number of PTSMTP worker processes that can run at any one time to handle incoming SMTP connections. If more simultaneous incoming SMTP connections are received than there are worker processes to handle them, additional worker processes, up to a maximum number specified by this keyword, will be created. These temporary processes will automatically shutdown after they've been idle for the number of seconds defined by the PTSMTP_IDLE_TIME keyword.

2.2.8.16 PTSMTP_IDLE_TIME keyword

Specifies the number of seconds a temporary PMAS PTSMTP worker process is idle before it automatically shuts itself down. See Section 2.2.8.15 for more details.

2.2.8.17 PTSMTP_QUARANTINE_REPLY keyword

Specifies the SMTP-dialogue message that's sent to the SMTP client when a message is quarantined. The default value for this keyword is:

```
ptsmtp_quarantine_reply    "250 2.0.0 Message queued for delivery"
```

When a message is quarantined or discarded, the sending SMTP client is notified that the message was successfully delivered via the "250 2.0.0" message above. If an error was returned, the client would simply retry delivery at a later time, so a success is returned to the client, even though the message has been sidelined by PMAS.

This keyword can be used to change the text of the message that's sent back to the client, but any reply should begin with "250 2.0.0":

```
ptsmtp_quarantine_reply    "250 2.0.0 Message accepted (but quarantined)"
```

Note: This reply is part of the SMTP dialogue between the SMTP client and the PMAS PTSMTP server. Success messages are never returned to the user, so end-users will not see this message.

2.2.8.18 PTSMTP_REJECT_REPLY keyword

Specifies the SMTP-dialogue message that's sent to the SMTP client when a message is rejected (see Section 2.2.5 for details on rejecting messages). The default value for this keyword is:

```
ptsmtp_reject_reply "550 5.7.1 Requested mail action not taken: rejected for policy reasons"
```

When a message is rejected, the sending SMTP client is notified that the message was rejected the "550 5.7.1" message above. This is a permanent failure message, which should cause the SMTP client to return the message to the sender as permanently undeliverable.

This keyword can be used to change the text of the message that's sent back to the client, but any reply should begin with "550 5.7.1":

```
ptsmtp_reject_reply  "550 5.7.1 Rejected!  This looks like spam."
```

Note: This reply is part of the SMTP dialogue between the SMTP client and the PMAS PTSMTP server. The failure message may be returned to the sender in the form of a bounce message, so end-users may see the text you supply in this reply.

2.2.8.19 SMTP_MAILSERVER_HOST keyword

Specifies the SMTP server hostname for messages generated by PMAS. The PreciseMail Processor, the quarantine notification job, and web-based GUI CGI program that releases messages need to know the system name and port number for an SMTP server that can deliver the messages they generate. If the messages generated by those programs were sent through the normal SMTP port (25), they would be susceptible to scanning by PMAS just as any other incoming SMTP message. By defining this keyword, you can have the messages delivered directly to your primary SMTP server, avoiding interaction with PMAS altogether.

The default value for this keyword is the value of the PTSMTP_MAILSERVER_HOST keyword. This keyword will not be needed for most configurations, but it is available in case you have a need to send messages to a different SMTP server.

2.2.8.20 SMTP_MAILSERVER_PORT keyword

Specifies the listen port for the SMTP Server that receives messages generated by PMAS. See Section 2.2.8.19 for details. The default value for this keyword is the value of the PTSMTP_MAILSERVER_PORT keyword.

2.2.8.21 PTSMTP_NORELAY_REPLY keyword

Specifies the SMTP reply to be sent to a sending client when an address is rejected due to a relay attempt (see Section 1.4.7, Anti-Relay Support, for details. The default value for this keyword is:

```
550 5.7.1 Relaying not allowed: %s
```

The text “%s” is replaced with the rejected address.

2.2.8.22 PTSMTP_NOUSER_REPLY keyword

Specifies the SMTP reply to be sent to a sending client when an address is rejected as not being a local address (see Section 1.4.7, Anti-Relay Support, for details. The default value for this keyword is:

```
550 5.1.1 Illegal or unknown user: %s
```

The text “%s” is replaced with the rejected address.

2.2.9 Anti-Virus Package keywords

The PreciseMail Pass-Through SMTP (PTSMTP) Server includes support for calling out to an anti-virus package to scan incoming messages for viruses before they are scanned by the PMAS spam engine. For details on anti-virus scanning, see Chapter 8, Anti-Virus Scanning.

The PMAS anti-virus support is controlled by the following configuration variables.

2.2.9.1 ANTIVIRUS_ENABLED keyword

Defined as *yes* or *no* (the default). Determines whether or not anti-virus support is enabled. When set to *yes*, a special anti-virus plugin is loaded by the PMAS PTSMTP Proxy Server processes.

2.2.9.2 ANTIVIRUS_PACKAGE keyword

Specifies the name of the anti-virus package to be used. Currently, the only supported anti-virus engine is the “Sophos” engine.

2.2.9.3 ANTIVIRUS_DIR keyword

Specifies the device and directory in which the anti-virus data files reside. The default value is /usr/local/sav.

Note: You must set this parameter to point to the correct device and directory in order for the anti-virus data files to be loaded.

2.2.9.4 VIRUS_DISPOSITION keyword

Determines the disposition of messages that bear viruses. PMAS will automatically replace the infected bodypart with a text message indicating that a virus was detected. What happens to the message after that is determined by this variable. The following options are available:

- 0 - Reject the message during the SMTP session
- 1 - Block the message (the PMAS score is set to BLOCK_SCORE)
- 2 - Discard the message (the PMAS score is set to DISCARD_THRESHOLD)
- 3 - Quarantine the message (the PMAS score is set to QUARANTINE_THRESHOLD)
- 4 - Forward the message (the message is subjected to the anti-spam rules)

If you choose to reject messages containing viruses, that rejection occurs during the SMTP session, and the message never actually enters your system (i.e, it is never written to a disk file).

If you choose to block, discard, or quarantine the message, it will be assigned a PMAS score according to the settings of the configuration variables described above. Note that any user settings for discard and quarantine thresholds may honored for the message's final disposition. If the user's quarantine threshold is higher than the system setting, the message may still be delivered to the user (but with the actual virus replaced by a harmless text message). Discarded or quarantined messages will include a header indicating that a virus was found, as described in Chapter 8.

If you choose to forward the messages, they are still subjected to the normal PMAS anti-spam rules and may be tagged, quarantined, or discarded as appropriate. Again, any message actually forwarded to the user will not contain the virus-infested bodypart.

2.2.10 User Interface keywords

The following keywords control and define the web-based GUI (Graphical User Interface) for PreciseMail Anti-Spam Gateway.

2.2.10.1 GUI_COOKIE_LIFETIME keyword

Specifies the lifetime of the PMAS login cookie in minutes. When a user logs in to the PreciseMail Anti-Spam Gateway UI, a browser cookie is generated to store login information. By default, the cookie that is created is a “session cookie”, which means it is maintained by the browser until the browser is closed, at which time the cookie is deleted. If you'd like for the login cookies to expire after a specific number of minutes, this keyword can be used to specify the desired number of minutes, where a value of “0” means “session cookie”. To set cookies that effectively never expire, simply set this value to a large number of minutes.

2.2.10.2 GUI_URI_HOST keyword

Specifies the hostname to be used in the creation of the PMAS GUI URLs (Uniform Resource Locators). The value of this keyword is combined with the values of GUI_URI_SCRIPT_PATH and GUI_URI_PATH to create the URLs for the various CGI and HTML pages that are to be served by the web server. For example, the URL for the quarantine page might look like this:

```
http://caesar.example.com:8080/scripts/pmas/quarantine.exe
```

To create that URL, the GUI_URI_HOST keyword would be defined as “caesar.example.com:8080”.

Note: The default value of GUI_URI_HOST is “-”, which tells the non-GUI parts of PMAS that the GUI is not enabled. You *must* define this variable in order for the GUI to work properly.

2.2.10.3 GUI_URI_USE_HTTP_HOST keyword

Controls whether or not the HTTP_HOST value passed to the CGI script is used as the value of GUI_URI_HOST. Setting this variable to “yes” allows for a configuration file to be shared for different domains and still have that particular domain show up in all generated URLs.

2.2.10.4 GUI_URI_SCRIPT_PATH keyword

Specifies the URL path for the CGI scripts that drive the PreciseMail Anti-Spam Gateway user interface. The path should start with a slash “/”, but not have a trailing slash. Using the example above, the value of this keyword would be */scripts/pmas*. The value you supply will depend upon how you chose to set up your web server to serve the CGI scripts.

The default value is */scripts/pmas*.

2.2.10.5 GUI_URI_PATH keyword

Specifies the URL path for the HTML pages and images used for the PreciseMail Anti-Spam Gateway user interface. The path should start with a slash “/”, but not have a trailing slash. The default value is */pmas*, which, when combined with the value of GUI_HOST_NAME, would create a URL like the following:

```
http://caesar.example.com:8080/pmas/index.html
```

2.2.10.6 GUI_URI_PROTOCOL keyword

Specifies the protocol used for serving the HTML pages. The allowed values are “http:” and “https:” (secure-mode).

Note: Whether or not “https:” is valid for your site depends on whether or not your web server has secure-mode support built into it.

2.2.10.7 GUI_ALLOW_QUARANTINE_ALL keyword

Specifies whether or not users are allowed to display all of their quarantined messages via the GUI. On a busy system with a lot of quarantined messages, allowing users to display all of their messages could impact system performance as there will be lots of disk I/O done to accommodate the requests. The default value is “yes”. If your system is negatively impacted by the “ALL” displays, you can change this variable to “no” to remove “ALL” as an option.

2.2.10.8 GUI_ALLOW_QUARALL_DEFAULT keyword

Specifies whether or not users are allowed to set their default quarantined message display to “ALL”. This keyword differs from GUI_ALLOW_QUARANTINE_ALL in that the latter doesn’t affect their ability to change the default view (i.e, they can view all messages, but they can’t set that as their default view unless this variable is set to “yes” (or unless the variable GUI_DEFAULT_QUARANTINE_ALL is set to “yes”).

2.2.10.9 GUI_DEFAULT_QUARANTINE_ALL keyword

Specifies whether or not all of a user’s quarantined messages are displayed when the user chooses to view his or her quarantine via the GUI. By default, only the messages quarantined for the current day are displayed. By setting this variable to “yes”, the default display will change to show all quarantined messages for each user.

2.2.10.10 GUI_QUARANTINE_DISPLAY_DAILY keyword

Changes the GUI quarantine display so that only messages quarantined on that particular day are displayed. Normally, the quarantine display shows all messages quarantined since the last notifications were mailed out (typically, the previous afternoon). By defining *gui_quarantine_display_daily* as “yes”, only messages quarantined since midnight of the current day are shown in the default display.

2.2.10.11 GUI_DELETE_UPON_RELEASE keyword

Specifies whether or not messages are deleted from the quarantine directory after they’re released by the user. By default, messages are left in the quarantine area (and in the quarantine display) in case there is a problem releasing the messages (while PMAS knows if the release to the MTA was successful or not, final delivery of the messages could fail). By setting this variable to “yes”, messages will be deleted from the quarantine area and display.

2.2.10.12 GUI_RENAME_UPON_DELETE keyword

Specifies whether or not messages deleted from the quarantine and discard directories are actually deleted. By default, messages deleted from the quarantine and discard views are physically deleted from the system, making it impossible to retrieve them later. By setting this variable to “yes”, messages will be deleted from the quarantine and discard views, but will actually still exist on disk, renamed with a “_DELETED” extension. The pmas_admin user can display deleted messages in the quarantine and discard views.

2.2.10.13 GUI_ALLOW_DISCARD_VIEW keyword

Specifies whether or not users are allowed to view their discarded messages via the GUI (assuming discarding of messages is enabled). The discard view works exactly as the quarantine view does, but only messages scored high enough to be discarded are displayed.

Note: Allowing users to view their discarded messages allows them to retrieve accidentally-discarded messages without admin intervention. However, enabling the discard view may confuse novice users who don’t understand the distinction between quarantined and discarded messages.

2.2.10.14 GUI_JAVA_SORT_MAXMSGS keyword

Specifies the maximum number of messages in the quarantine and discard views for which the Javascript sorting is used. Beginning with PMAS V2.4, the quarantine and discard views use Javascript routines to handle the sorting of messages in the display. The use of the Javascript routines eliminates the need to reprocess the quarantine index files when a user wants to re-sort the display. Unfortunately, large numbers of messages can overwhelm the Javascript/browser memory limitations, causing browser lock-ups or other errors. To avoid these, the previous, non-Javascript sort is used if the view contains more than the number of messages specified by this variable. The default value is 200 messages.

2.2.10.15 GUI_FORCE_JAVA_SORT_OPERA keyword

Specifies whether or not the Javascript sort for the quarantine and discard views is used with the Opera browser. Opera V9.01 does not support the mechanism used by the Javascript sort routines, so users using Opera will see the previous, non-Javascript sort. If future versions of Opera include this support, setting this variable to “yes” will disable the browser-check that’s currently performed.

2.2.10.16 USER_DATABASE keyword

Specifies the name of the database used to store the PreciseMail Anti-Spam Gateway user preferences. This file is an indexed ISAM file that is automatically created by PMAS. The default value is */pmas/data/pmas_user_db.dat*.

2.2.10.17 ADMIN_EMAIL_ADDRESS keyword

Specifies the email address of the PreciseMail Anti-Spam Gateway system administrator. When users view their quarantined messages via the PMAS GUI, one of the options available to them is to send messages to the system administrator to report false positives (messages that were quarantined but should not have been). Those messages are mailed to the email address specified by this keyword. There is no default value for this keyword, which means that the “Send to Administrator” buttons do not function if an address is not supplied.

2.2.10.18 OPT_IN_BY_DEFAULT keyword

Specifies the default “opt in” value for users. The default value is *YES*, which means that incoming mail for all users is scanned by PreciseMail Anti-Spam Gateway (i.e, all users are considered to have “opted in” for scanning of their mail). If defined as *NO*, PreciseMail Anti-Spam Gateway will only scan the incoming mail of those users who explicitly chose to “opt in” via their PreciseMail Anti-Spam Gateway preferences settings.

If you plan to deploy PreciseMail Anti-Spam Gateway for a small subset of your user base, you will probably want to set OPT_IN_BY_DEFAULT to *NO* and have the users in the subset elect to opt in for PMAS scanning.

2.2.10.19 ALLOW_USER_OPTIN keyword

Specifies whether or not users are allowed to change their PMAS “opt in” settings. The default value is *YES*. If defined as *NO*, users will not be given the choice of opting in or out of PreciseMail Anti-Spam Gateway scanning on their PMAS Preferences screen.

2.2.10.20 ALLOW_USER_DISCARD keyword

Specifies whether or not users are allowed to choose to discard messages and set their own discard threshold values. The default value is *NO*, which means users will not be given the choice of discarding messages on their PMAS Preferences screen unless this value is changed to *YES*. The default of *NO* ensures that the system administrator can determine whether or not discarding is allowed, as doing so affects systems resources.

2.2.10.21 ALLOW_USER_QUARANTINE keyword

Specifies whether or not users are allowed to choose to quarantine messages and set their own quarantine threshold values. The default value is *NO*, which means users will not be given the choice of quarantining messages on their PMAS Preferences screen unless this value is changed to *YES*. The default of *NO* ensures that the system administrator can determine whether or not quarantining is allowed, as doing so affects system resources.

2.2.10.22 ALLOW_USER_TAGGING keyword

Specifies whether or not users are allowed to choose to tag their suspected spam by modifying the Subject: headers of messages. The default value is *YES*. If defined as *NO*, users will not be given the choice of tagging messages on their PMAS Preferences screen.

If you do not wish to enable message tagging system-wide and at the same time wish to prevent your users from choosing to tag their messages, define this keyword as *NO*.

2.2.11 User Authentication keywords

The following keywords control the authentication methods used to control access to user quarantines and PMAS preferences.

2.2.11.1 AUTH_METHODS keyword

Specifies a comma-separated string containing a list of authentication methods and the order in which they should be tried. Valid authentication methods are:

- *PMAS* (passwords stored in the PMAS user database)
- *SYSTEM* (SYSUAF on VMS, /etc/passwd file on UNIX)
- *POP3* (connect to POP3 server and attempt to authenticate user)
- *IMAP4* (connect to IMAP4 server and attempt to authenticate user)
- *LDAP*

The default value for this keyword is:

```
auth_methods    PMAS,SYSTEM
```

With this setting, users trying to log in will first be authenticated against the PMAS user database, and then via the system password database.

The /pmas/bin/pmasadmin utility can be used to add passwords for arbitrary email addresses to the PMAS user database. This capability allows you to create PMAS authentication passwords for email addresses that do not have a corresponding SYSTEM, POP3, IMAP4, or LDAP account.

The /pmas/bin/pmasadmin command takes several arguments. Running the program with no arguments displays the valid arguments that may be specified. To add a password for an email address, simply use the “user create” command and specify the email address and password:

```
# /pmas/bin/pmasadmin user create cornelius@example.com forbiddenzone
```

For more information about the pmasadmin utility, see Section 5.1.

2.2.11.2 AUTH_POP3_HOSTS keyword

Specifies a comma-separated string containing a list of POP3 servers to try authenticating against and the order in which they should be tried. If *POP3* isn’t specified in the list defined by AUTH_METHODS, this keyword is ignored. If *POP3* is specified and this keyword is left blank, *localhost* is assumed.

If the POP3 server is running on a port other than the standard POP3 port (110), you can specify the alternate port number by following the host name with a colon “:”, followed by the port number. For example:

```
auth_pop3_hosts    nodea.example.com:2110,nodeb.example.com
```

2.2.11.3 POP3_CONNECT_TIMEOUT keyword

Specifies the number of seconds that PMAS will attempt to contact an POP3 server before giving up. By default, POP3 authentication attempts will wait for 30 seconds to connect to the POP3 server. If no connection is made in that time, the connect attempt is aborted.

2.2.11.4 AUTH_IMAP4_HOSTS keyword

Specifies a comma-separated string containing a list of IMAP4 servers to try authenticating against and the order in which they should be tried. If *IMAP4* isn’t specified in the list defined by AUTH_METHODS, this keyword is ignored. If *IMAP4* is specified and this keyword is left blank, *localhost* is assumed.

If the IMAP4 server is running on a port other than the standard IMAP4 port (143), you can specify the alternate port number by following the host name with a colon “:”, followed by the port number. For example:

```
auth_imap4_hosts    nodea.example.com:2112,nodeb.example.com
```

2.2.11.5 IMAP4_CONNECT_TIMEOUT keyword

Specifies the number of seconds that PMAS will attempt to contact an IMAP4 server before giving up. By default, IMAP4 authentication attempts will wait for 30 seconds to connect to the IMAP4 server. If no connection is made in that time, the connect attempt is aborted.

2.2.11.6 LDAP_AUTH_SERVER keyword

Specifies the name of the LDAP host to search for authentication information. There is no default value. Multiple LDAP servers can be specified in a comma-separated list. Each will be tried until a successful connection is made.

The LDAP support in PMASLOGIN and AUTHDEUG allows for a cacert file to be specified for LDAP over TLS. If the file /pmas/data/ldap_cacert.txt exists, it is loaded as the TLS_CACERTFILE for OpenLDAP.

2.2.11.7 LDAP_BASE_DN keyword

Specifies the entry in the LDAP directory under which searches occur (sometimes also known as the search base). Consult your LDAP server's documentation set for more information specific to your implementation.

Both LDAP_BASE_DN and LDAP_AUTH_FILTER allow the following expansion tags to be used in their values:

Tag	Description
%e	The user's email address
%d	The domain hosting the user's email account
%u	The user's login name
%c	The domain hosting the user's email account, split into its components

For example, a site might set the values of LDAP_BASE_DN and LDAP_AUTH_FILTER as :

```
ldap_base_dn            o=%d
ldap_auth_filter        (&(objectclass=person)(uid=%u))
```

and:

```
ldap_base_dn            DC=%c,DC=%c
ldap_auth_filter        (&(objectclass=person)(uid=%u))
```

If a user logged in as jdoe@example.com, the values of these configuration variables would be expanded to:

```
ldap_base_dn:     o=example.com
ldap_auth_filter: (&(objectclass=person)(uid=jdoe))
```

and:

```
ldap_base_dn:     DC=example,DC=com
ldap_auth_filter: (&(objectclass=person)(uid=jdoe))
```

2.2.11.8 LDAP_AUTH_FILTER keyword

Specifies the LDAP search filter used to find the directory entry for a user who is authenticating to the web user interface.

LDAP_AUTH_FILTER supports the same tag expansions as LDAP_BASE_DN.

2.2.11.9 LDAP_GROUP_FILTER keyword

Specifies the LDAP search filter used to list user groups in the LDAP directory.

2.2.11.10 LDAP_GROUPMEMBER_ATTR keyword

Specifies the name of the LDAP attribute that contains the identity of an LDAP group member. On most LDAP servers, this will be "uniqueMember". On most ActiveDirectory servers, the value will be "member".

2.2.11.11 LDAP_SEARCHACCT_DN keyword

PreciseMail Anti-Spam Gateway must query the LDAP server to find the Distinguished Name of the user attempting to log in before the user can be authenticated. By default, this initial query will be done anonymously. Some directory servers (notably Microsoft's Active Directory) do not allow anonymous queries.

LDAP_SEARCHACCT_DN specifies the Distinguished Name of a user with search privileges on the directory server that PMAS will connect as. By default, the value is NULL which indicates an anonymous login.

2.2.11.12 LDAP_SEARCHACCT_PASSWORD keyword

Specifies the password for the search user whose Distinguished Name is specified in LDAP_SEARCHACCT_DN. By default, the value is NULL which indicates an anonymous login.

2.2.11.13 LDAP_USE_TLS keyword

If your LDAP server supports LDAPS (LDAP-over-TLS), setting the value of this variable to 1 will instruct the PMAS web interface to attempt to use LDAPS for user authentication. If an LDAPS connection cannot be established, a standard LDAP connection will be used to authenticate the user.

Setting the value of this variable to 2 will force an LDAPS connection. If an LDAPS connection cannot be established, the user will receive an error and will not be able to log in.

2.2.12 Bayesian Engine keywords

PreciseMail Anti-Spam Gateway includes an artificial-intelligence engine called a Bayesian engine that can be taught what spam messages look like. It uses word-occurrence frequencies for both ham and spam messages to determine the likely spamicity for a given message.

The validity of the results from the Bayesian engine depend greatly on how well the engine is trained and by how many messages, both spam and ham, are used to train it. The Bayesian engine adds additional system overhead for each message processed, so care should be taken when deciding whether or not to enable the engine. By default, the engine is disabled. It is recommend that sites “autotrain” the engine (see Section 2.2.12.2) for a period of several days before turning the engine on via the USE_BAYESIAN keyword.

2.2.12.1 USE_BAYESIAN keyword

Defined as *yes* or *no* (the default). Determines whether or not the message is passed through the Bayesian engine for spamicity scoring. If enabled, the resulting Bayesian spamicity score is added to the heuristic score to determine the final score for the message.

Scores returned by the Bayesian engine range from -1 to 1. The BAYESIAN_MULTIPLIER keyword (see Section 2.2.12.5) can be used to give the Bayesian result more weight when calculating the final score.

2.2.12.2 AUTOTRAIN_BAYESIAN keyword

Defined as *yes* or *no* (the default). If enabled, messages are automatically passed to the Bayesian engine for training purposes. The message score and the threshold keywords described below determine whether or not messages are classified as ham or spam by the engine as it trains.

2.2.12.3 AUTOBAYESIAN_HAM_THRESHOLD keyword

Specifies the numeric score for determining whether or not a message should be treated as ham for autotraining the Bayesian engine. The default value is 0.0. Messages with a final score of 0.0 or lower will be used to teach the Bayesian engine about ham messages if AUTOTRAIN_BAYESIAN is defined as *yes*.

2.2.12.4 AUTOBAYESIAN_SPAM_THRESHOLD keyword

Specifies the numeric score for determining whether or not a message should be treated as spam for autotraining the Bayesian engine. The default value is 10.0. Messages with a final score of 10.0 or higher will be used to teach the Bayesian engine about spam messages if AUTOTRAIN_BAYESIAN is defined as *yes*.

2.2.12.5 BAYESIAN_MULTIPLIER keyword

Specifies the weight of the Bayesian engine spamicity rating for a message. The engine returns a score of -1 to 1. That score is multiplied by the BAYESIAN_MULTIPLIER value and then added to the message’s heuristic score. The default value is 1. When larger values are specified, the Bayesian results will be given more weight when the final score is determined. For example, a multiplier value of 3 turns the Bayesian result into a score in the range -3 to 3.

2.2.13 Debugging

The following keywords control the debugging output from the PreciseMail Anti-Spam Gateway main program as it processes messages.

2.2.13.1 DEBUG_LEVEL keyword

Specifies a bitmask value between 0 and 15 that determines the amount of debugging information generated. The default value is 0, which means no log file is created. See Example 9–1, Example Interactive Run of PMAS, for a full annotated example.

The bits have the following meanings:

Bit	Value	Meaning
0	1	Basic debugging
1	2	Rule matching
2	4	Rule debugging
3	8	Bayesian debugging
4	16	SMTP debugging for messages sent by PMAS via SMTP
5	32	Full SMTP debugging for messages sent by PMAS via SMTP
6	64	PMAS GUI CGI debugging
7	128	PMAS cluster debugging

2.2.13.2 DEBUG_LOGFILE keyword

Specifies the name of the debug log file for the PreciseMail Anti-Spam Gateway main image. The default is /pmas/log/pmas_debug.log.

2.2.14 Miscellaneous

The following keywords apply to other areas of PreciseMail Anti-Spam Gateway.

2.2.14.1 ALIAS_FILE keyword

Specifies the name of the file containing local site aliases for addresses. The alias file can be used to redirect mail for specific addresses. The alias file is also referenced when determining if a user has an allow/block file. The default file name is /pmas/data/aliases.txt.

2.2.14.2 GATHER_STATS keyword

Specifies whether or not the PMAS Stats batch jobs runs every hour to gather statistics. The default value is “yes”. To disable the statistics gathering, set this variable to “no”. For more information, see Section 4.3.

2.2.14.3 MASTER_LOGFILE keyword

PreciseMail Anti-Spam Gateway maintains a log file in which it records the disposition of each processed message. The default file name is /pmas/log/pmas.log. See Section 1.7 for more information about this file.

2.2.14.4 USERLIST_SUBDIR_LEVEL keyword

Specifies the number of subdirectory levels to be used for storing user allow/block list files. For directory efficiency, you can specify up to 8 subdirectory levels for user allow/block files. The number specifies how many characters from the email address are used for subdirectory levels. For example, for address "JOEUSER", a value of 3 would result in a directory name of ../j/o/e/ With the default value of 0, the files are simply stored in the top-level directory specified by user_rules_directory.

If you change the USERLIST_SUBDIR_LEVEL setting, you are responsible for moving the user files to the appropriate directories. PreciseMail Anti-Spam Gateway does not move the files automatically.

2.2.15 Quarantine Notification Batch Job

The following keywords are used to control the quarantine notification batch job that runs periodically to notify users of messages held on their behalf.

2.2.15.1 SEND_QUARNOTICES_DEFAULT keyword

Specifies whether or not quarantine notices are mailed to users by default. This value is propagated to new user records in the user database when added by PMASADMIN or via the Preferences GUI. The default value is “yes”.

2.2.15.2 MIME_QUARANTINE_MESSAGE keyword

Defined as *yes* (the default) or *no*. By default, the quarantine notices mailed to users are sent as multipart plain and HTML messages. If you prefer, you can have just text messages sent; the HTML version shows the Subject: lines in boldface, making it easier for users who use HTML mail readers to see the quarantined message subjects.

2.2.15.3 NOTIFY_MSG_SUBJECT keyword

Specifies the text for the Subject: header for quarantine notice messages. The default text is “PreciseMail Quarantined Messages (expire DD-MMM-YYYY)”. The variable “%EXPDATE%” can be specified in the subject text. If found, it’s replaced with the expiration date of the messages listed in the message.

2.2.15.4 NOTIFY_DEBUG_LEVEL keyword

Specifies a bitmask value between 0 and 3 that determines the amount of debugging information generated by the quarantine notification job. The default value is 0. The log file name for the batch job is /pmas/log/pmas_notify.log.

The bits have the following meanings:

Bit	Value	Meaning
0	1	Basic debugging
1	2	Verbose matching

2.2.16 PreciseMail Anti-Spam Gateway Processor

The following keywords control the PreciseMail Anti-Spam Gateway Processor, the email-based user interface for releasing quarantined messages and defining user allow/block files. (See Section 1.10 for more information.)

2.2.16.1 PROCESSOR_USER_NAME keyword

Specifies the username portion of the address for the PreciseMail Anti-Spam Gateway Processor email-based user interface. This value should match the alias defined to forward mail to the processor via the PIPE channel or a command alias. The default value is “PreciseMail”.

2.2.16.2 PROCESSOR_USER_ADDRESS keyword

Specifies the full email address for the PreciseMail Anti-Spam Gateway Processor email-based user interface. This value is used as sending address for all messages mailed by the processor. The default value is created from the values of PROCESSOR_USER_NAME and PMAS_SYSTEM_NAME.

2.2.16.3 PMAS_PROCESS_DEBUG_LEVEL keyword

Specifies a bitmask value between 0 and 3 that determines the amount of debugging information generated by the quarantine notification job. The default value is 0.

The bits have the following meanings:

Bit	Value	Meaning
0	1	Basic debugging
1	2	Verbose matching

2.2.16.4 PMAS_PROCESS_DEBUG_LOGFILE keyword

Specifies the name of the PreciseMail Anti-Spam Gateway Processor debug log files. The default log file name is /pmas/log/pmas_process.log.

2.2.16.5 MIME_HELP_MESSAGE keyword

Specifies whether or not the message returned by the PreciseMail Anti-Spam Gateway Processor in response to the “HELP” command is a MIME message. The default setting of *NO* causes a plain text reply (found in /pmas/help/pmas_process_help.txt) When MIME_HELP_MESSAGE is set to *YES*, the PreciseMail Processor will assume that the file also includes all the appropriate headers, including a “Subject:” header, necessary to deliver a reply as a plain text, HTML, or multipart MIME message.

A sample template file named pmas_process_help.template can be found in /pmas/help. Simply edit the file as desired and rename it to pmas_process_help.txt in the /pmas/help directory.

2.2.17 Automatic updates keywords

The filtering rules used by PreciseMail Anti-Spam Gateway and the optional Sophos anti-virus engine can be automatically updated whenever an update is released. These keywords control the automatic update system.

2.2.17.1 AUTOUPDATE_RULES keyword

If set to *2* (the default), PreciseMail Anti-Spam Gateway rule updates will be automatically downloaded and installed when new updates are released. If set to *0*, automatic updating will be turned off. The system administrator will need to manually download and install rule updates with this setting.

A setting of *1* enables a legacy automatic update system that has been deprecated by PMAS V2.2. Sites currently using the legacy system are highly encouraged to switch to the new update system by changing the value of this configuration variable to *2*.

2.2.17.2 AUTOUPDATE_SOPHOS keyword

Setting this variable to *yes* (the default) will enable automatic updating of the Sophos virus definitions (ides.zip) whenever Sophos releases an update.

Note: You must have a valid Sophos license from Process Software for this feature to be enabled. Contact your Process Software sales representative for more information.

2.2.17.3 AUTOUPDATE_STATS keyword

When this variable is set to *yes* (the default), spam scanning statistics from the system will be uploaded to Process Software at regular intervals. These statistics are kept private, and are only used in aggregate to improve PreciseMail Anti-Spam Gateway. No email addresses, message content, or other site-private information is included in the statistics.

2.2.17.4 AUTOUPDATE_ADDRESS keyword

Specifies the fully-qualified email address that the PreciseMail Anti-Spam Gateway email-based user interface listens to on this system.

Note: This configuration variable is used by the deprecated SMTP-based auto update system. Sites using the deprecated system are strongly advised to switch to the new update system.

2.2.17.5 UPDATES_CONNECT_TIMEOUT keyword

Specifies the number of seconds that PMAS will attempt to contact the Process Software data center for updates before giving up. By default, the update procedure will wait to connect to the data center for 30 seconds. If no connection is made in that time, the connection attempt is aborted.

2.2.18 Data Synchronization Cluster keywords

PreciseMail Anti-Spam Gateway provides the ability to centralize configuration, management, and data for domains with multiple MTAs. These keywords enable and control the cluster software. For more information, see the chapter on Data Synchronization Clusters in this Guide.

2.2.18.1 CLU_CLIENTS keyword

If the system is the primary node of a Data Synchronization Cluster, this variable specifies all of the secondary nodes in the cluster in a comma-separated list. If the current system is not the primary node of a Data Synchronization Cluster, this variable is ignored.

2.2.18.2 CLU_SERVER keyword

If the current system is a secondary node in a Data Synchronization Cluster, this variable specifies the hostname of the cluster's primary node. If the current system is not a secondary node in a data synchronization cluster, this variable is ignored.

2.2.18.3 CLU_SERVER_CLIENT keyword

Specifies if the current system is a standalone, primary, or secondary node. Each Data Synchronization Cluster must have exactly one primary node and one or more secondary nodes. A value of *0* (the default) means that the current system is not part of a Data Synchronization Cluster. A value of *1* means that this is the primary node of a cluster, and a value of *2* means that the current system is a secondary node in a cluster.

2.2.19 Customizing the Quarantine Notices

When messages are quarantined by PreciseMail Anti-Spam Gateway, a batch job periodically delivers quarantine notices to users (as discussed in Section 2.2.3, Quarantining Messages. The text for the quarantine notices can be customized by creating the following files:

- /pmas/data/quarantine_message.txt
- /pmas/data/quarantine_message_plain_row.txt

- /pmas/data/quarantine_message_html_row.txt

There is a quarantine_message.template file supplied as part of the installation; it can be renamed to .txt and customized as desired). If this file does not exist, the default message text and format is used.

The quarantine notice messages may be delivered as either plain text messages or multipart MIME messages, with both a plain and an HTML part. The type of message generated is defined by the "mime_quarantine_message" keyword (see Section 2.2.15.2). The supplied template message is a multipart MIME message. If you're sending plain text, the MIME headers and the HTML portion of the file can be omitted.

There are six variables that can be specified in the quarantine message template. The variable names are uppercase and are delimited by percent signs ("%"). The following variables are supported:

%MESSAGES%	Required	Specifies the point in the plain bodypart where the quarantined message information is inserted.
%HTMLMESSAGES%	Required	Specifies the point in the HTML bodypart where the quarantined message information is inserted. (This variable is optional if MIME messages are not used.)
%USERNAME%	Optional	Specifies the username portion of the PreciseMail Anti-Spam Gateway Processor address (see Section 2.2.16.1).
%NUMDAYS%	Optional	Specifies the number of days quarantine messages will be held (see Section 2.2.3.3).
%LASTDATE%	Optional	Specifies the date quarantined messages will be automatically deleted; calculated from the current date and the configuration variable ''quarantine_msg_lifetime''.
%GUI_URL%	Optional	Specifies the URL for the quarantine CGI script in the PMAS web-based user interface. The configuration variable GUI_URI_HOST must be defined for this keyword to be replaced.

To specify the layout of the information regarding each quarantined message, the files quarantine_message_plain_row.txt and quarantine_message_html_row.txt can be created. The following variables can be used in these files. Each occurrence will be replaced by the following information:

%ADDRESS%	The email address for which the message was quarantined
%DATE%	The date the message was quarantined
%FROM%	The full From: address
%FROMADDR%	The email address from the From: header
%FROMNAME%	The personal name from the From: header
%FROMTRUNC%	The full From: address truncated to 40 characters
%GUI_URI_PROTOCOL%	The value of GUI_URI_PROTOCOL
%GUI_URI_HOST%	The value of GUI_URI_HOST
%GUI_URI_PATH%	The value of GUI_URI_PATH
%GUI_URI_SCRIPT_PATH%	The value of GUI_URI_SCRIPT_PATH
%MSGFILENAME%	The file specification for the quarantined message
%PREVIEW_URL%	The URL to effect a preview of the message via the PMAS GUI

%RELEASE_URL%	The URL to effect a release of the message via the PMAS GUI
%RELEASE_MAILTO%	The address to which the message will be released
%SCORE%	The spam score for the message
%SUBJECT%	The contents of the Subject: header
%SUBJECTTRUNC%	The contents of the Subject: header truncated to 40 characters
%TIME%	The time the message was received
%TO%	The contents of the To: header from the message
%TOTRUNC%	The contents of the To: header truncated to 40 characters

A sample, simple plain row layout might look something like this:

```
%FROMADDR% - (%SCORE%) Message: %MSGFILENAME%
%SUBJECT%
```

The HTML row layout must include whatever HTML tags are required for proper HTML formatting.

3 The PreciseMail Anti-Spam Gateway Rules

This chapter describes the PreciseMail Anti-Spam Gateway rule files, the types of rules that exist, and how new rules can be added and existing scores modified for a site.

The PreciseMail Anti-Spam Gateway rule files are stored in the directory /pmas/data/ and are named with a .CF extension. Table 3–1 shows the files that comprise the rule set.

Table 3–1 PreciseMail Anti-Spam Gateway Rule Files

File name	Description
00_LOCAL_TESTS.CF	Local rules and scores
00_ALLOWBLOCKLISTS.CF	Local allow and block rules
20_ANTI_RATWARE.CF	Rules to try to identify ‘‘legitimate’’ mail clients.
20_BODY_TESTS.CF	Rules applied to message bodies.
20_COMPENSATE.CF	Rules to compensate for some of the aggressive rules.
20_HEAD_TESTS.CF	Rules applied to message headers.
20_HTML_TESTS.CF	Rules applied to HTML messages.
20_META_TESTS.CF	Meta rules made up of header and body meta tests.
20_PHRASES.CF	Rules for identifying popular spam phrases.
20_PORN.CF	Rules for identifying words associated with porn messages.
20_RATWARE.CF	Rules for identifying messages sent by popular spam software.
20_URI_TESTS.CF	Rules applied to URIs in the message body.
50_SCORES.CF	Scores for the rules in the 20_* files.
99_LOCAL_SCORES.CF	Local scores to override the scores in 50_SCORES.CF.

Note: Only the 00_*.CF and 99_*.CF files should be modified by a site. The other files should be considered read-only and will be automatically replaced by future PreciseMail Anti-Spam Gateway updates.

PMAS supports the inclusion of files via the “@” symbol in rule files. To include the contents of a file into another file, simply specify “@”, followed by the filename:

It is recommended that any such lines be added to 00_LOCAL_TESTS.CF, 00_ALLOWBLOCKLISTS.CF, and/or 99_LOCAL_SCORES.CF, as these files are not updated by new rule sets. Only one level of file-inclusion is supported (i.e, an inclusion line in an included file will be ignored). Similarly, only like files should be included by each file (any file included by 00_ALLOWBLOCKLISTS.CF must contain only allow, block, and “rule” rules, for example).

3.1 Rule file format

Most of the rules executed by PreciseMail Anti-Spam Gateway are regular expressions that are applied to the various message parts. Each rule is named and has three separate parts: the rule itself, a description, and a score. A sample rule looks like this:

```
header DEAR_SUBJECT     Subject =~ /^[Dd]ear .*,.*/
describe DEAR_SUBJECT   Subject starts with "Dear somebody,"
score DEAR_SUBJECT      10.00
```

The first line specifies the kind of test being defined (the "header" keyword), the name of the test, and the test itself (in this case, a regular expression that's applied to the RFC822 Subject: header for the message). Test names must consist only of letters, numbers, and the underscore character ("_").

The second line provides a description for the rule. This description will be included in the X-PMAS-* headers, if those are added to processed messages.

The third line specifies the score for the DEAR_SUBJECT rule. In this case, a rather high score of 10.0 is defined, which means any message that matches that rule will be considered spam.

PMAS rule file lines can be continued by ending a line with the backslash character "\". Leading whitespace on the continuation line is *not* ignored, so caution should be used when continuing a line in the middle of a regular expression.

```
header TEST_SUBJECT Subject =~ /^This is\
 a test/i
```

Table 3–2 describes the keywords that can be specified in the PreciseMail Anti-Spam Gateway rule files.

Table 3–2 Rule File Keywords

Keyword	Description
describe	Provides a description for the named rule
score	Specifies the score for the named rule
header	Rules that are applied to particular headers
body	Rules that are applied to the message body
rawbody	Rules that are applied to the raw message body
full	Rules that are applied to the full message body
uri	Rules applied to URIs found in the body of the message
meta	Boolean rules that test the results of other rules

3.2 Regular Expressions

Regular expressions are the key to the PreciseMail Anti-Spam Gateway spam-detecting rules. A regular expression ("regex") is a set of symbols and text used to match patterns of text. Regular expressions provide extremely sophisticated pattern-matching capabilities, allowing one rule to match a variety of slightly-different text patterns.

3.2.1 Regular Expression syntax

A full description of regular expressions is beyond the scope of this manual. There are a number of online resources on the World-Wide Web that provide the details needed. Two such URLs are:

```
http://www.regular-expressions.info/
http://etext.lib.virginia.edu/services/helpsheets/unix/regex.html
```

Some of the more common regular expression metacharacters and constructs are shown in Table 3–3.

Table 3–3 Simple Regular Expression Characters

Expression	Description
c	a single (non-meta) character matches itself
.	matches any single character except newline
?	postfix operator; preceding item is optional
*	postfix operator; preceding item 0 or more times
+	postfix operator; preceding item 1 or more times
|	infix operator; matches either argument
^	matches the empty string at the beginning of a line
$	matches the empty string at the end of a line
[chars]	match any character in the given class; if the first character after [is ^, match any character not in the given class; a range of characters may be specified by ‘‘first-last’’; for example, [^A-Za-z0-9] matches any character that’s not alphanumeric

The PreciseMail Anti-Spam Gateway regular expression support is provided by the PCRE (Perl Compatible Regular Expressions) library written by Philip Hazel. Full details on the types of regular expressions that can be written for PreciseMail Anti-Spam Gateway can be found on the PCRE and Perl documentation web sites:

```
http://www.pcre.org/
http://www.perl.com/doc/manual/html/pod/perlre.html
```

3.2.2 Using Regular Expressions in PreciseMail Anti-Spam Gateway Rules

Regular expressions used in PreciseMail Anti-Spam Gateway rules must be delimited (typically with a pair of slashes (“/”), but any delimiting character can be used). For example:

```
/[Rr][Ee]: .*FREE.*/
```

As documented on the PCRE site, you can also specify qualifiers after the closing delimiter. The one that will most likely be of most interest is “i”, which makes the regular expression case-insensitive:

```
/Re: .*FREE.*/i
```

That example would match “RE: FREE”, “Re: Free”, and other similar variations.

3.2.3 Testing Regular Expressions

You can test regular expressions for various matches using the PCRETEST program supplied in /pmas/bin/. The program will prompt for a regular expression and then will repeatedly prompt for text strings to which the expression should be applied. The program will report whether or not there was a match.

```
$ /pmas/bin/pcretest
PCRE version 4.5 01-Dec-2003

  re> /re: .* free .*/i
data> Re: This is a free offer!
 0: Re: This is a free offer!
data> This is a free offer!
No match
data> Re: Let FREEDOM ring!
No match
data> /RE: Get FREE Stuff!/
 0: RE: Get FREE Stuff!/
data>
```

Writing regular expressions can be tricky; it’s a good idea to always thoroughly test your expressions before adding them as rules for PreciseMail Anti-Spam Gateway.

3.3 Writing Header Rules

Header rules are applied to the RFC822 headers of messages being processed by PreciseMail Anti-Spam Gateway. The format of a typical header rule is:

```
header  rule-name  Header-name =~ /regular expression/
```

The “=~” is required syntax to separate the header name from the regular expression. For example:

```
header DEAR_SUBJECT    Subject =~ /^[Dd]ear .*,.*/
describe DEAR_SUBJECT  Subject starts with ‘‘Dear somebody,’’

header FROM_ENDS_IN_NUMS        From =~ /\d\d\@/
describe FROM_ENDS_IN_NUMS      From: ends in numbers
```

For efficiency reasons when executing header lookups, the header name must be one of more than 500 common header names known by PMAS. To test for a header unknown to PMAS, the special keyword “ALL” can be used. For the “ALL” header test, all of a message’s headers are appended together and separated by newlines (linefeeds). You can test for unknown headers using a rule similar to the following example:

```
header __LOCAL_HEADER   ALL =~ /\nX-Local-Header: text/
```

Other special header keywords include the following:

- FROM - Contains the contents of the decoded “From:” header
- FROM_ORIG - Contains the contents of the “From:” header before any decoding happens
- SUBJECT - Contains the contents of the decoded “Subject:” header
- SUBJECT_ORIG - The original “Subject:” header before any quoted-printable or base64 decoding is done
- SUBJECT_NOPUNC - The “Subject:” header with all punctuation (non-printable) characters removed
- ENVELOPE_FROM - Contains the envelope return address (the MAIL FROM: address in an SMTP transaction)
- ENVELOPE_TO - Contains all the recipient addresses, separated by newlines (the RCPT TO: addresses in an SMTP transaction)
- ENVELOPE_TO_ALIAS - Contains all the recipient addresses after PMAS alias processing has occurred, separated by newlines
- RECEIVED - Contains the values of all the “Received:” headers, separated by newlines
- MESSAGE_ID - Contains the values of the “Message-id:”, “Resent-Message-id:”, and “X-Message-id:” headers, separated by newlines
- ALL - Contains all of the message headers as one long newline-separated string

There are two special forms of the header rule; one is used to call internal PreciseMail Anti-Spam Gateway routines (“eval” tests), and the other is used to test for the existence of a header (“exists” tests). The following shows an example of each of these tests:

```
header CTYPE_JUST_HTML     eval:check_for_content_type_just_html()
describe CTYPE_JUST_HTML   HTML-only mail, with no text version

header __EBAY_MAILTRACKER  exists:X-eBay-MailTracker
```

There are rules already present for all of the supplied “eval” routines, so they will not be listed here.

3.4 Writing Body Rules

Body rules are applied against the text portions of a message. If a text message or a text part of a multipart message is encoded in base64 or quoted-printable formats, PreciseMail Anti-Spam Gateway will automatically decode those parts back to plain text before the rules are applied. This gets around a common spammer trick, which is to base64-encode the spam message to try to hide the text from spam checkers.

There are three different types of body rules, and each is applied to the message body in different ways:

- **body**—Applies to the textual parts of the message body. Any non-text MIME parts are stripped, and the text will be decoded from base64 or quoted-printable encoding. All HTML tags and line breaks will be removed before matching.

- **rawbody—**Applies to the textual parts of the message body. The text will be decoded from base64 or quoted-printable encoding, but HTML tags and line breaks will still be present.
- **full—**Applies to the "full body" of the message, which is its undecoded text, including all parts (text, images, etc.).

Most body rules are either "body" or "rawbody" rules. The format of a body test is:

```
keyword  TEST_NAME   /regular-expression/qualifiers
```

The following examples demonstrates the body and rawbody tests:

```
rawbody RANDOM_WORD_HTML    /<!--RANDOM_WORD-->/
describe RANDOM_WORD_HTML   Spammer forgot to fill in RANDOM_WORDs in HTML

body GUARANTEED_ATS         /\bgu@r@nteed\b/i
describe GUARANTEED_ATS     Body includes ‘‘gu@r@nteed’’ to try to hide
```

3.5 Writing URI Rules

URI rules are applied to all of the URIs (Uniform Resource Indicators) found in the body of a message. They can be very useful in identifying links to known spam web sites embedded in HTML or plain text messages.

The format of the URI rule is

```
uri  TEST_NAME  /regular-expression/qualifiers
```

The following examples demonstrate a couple of simple URI rules:

```
uri OFF_LIST_URI                /off_list\.pl/
describe OFF_LIST_URI           URI specifies Perl list remover

uri VIAGRA_URI                  /viagra/
describe VIAGRA_URI             URI has viagra in it
```

Those rules would match any of these URIs in a message body:

```
http://www.example.com/cgi?off_list.pl
http://www.getviagranow.com/index.html
http://www.example.com/cgi?&Referer=Blah&code=viagra&q=2
```

3.6 Writing Meta Rules

A "meta" rule is a boolean test of the results of other header, body, or URI rules. Meta rules can be used to produce tests that will only match messages that meet several criteria. It is perhaps best explained by an example:

```
header __EBAY_MAILTRACKER  exists:X-eBay-MailTracker
header __EBAY_MSGID        Message-ID =~ /.*@.*\.ebay.com>/
header __EBAY_FROM         From =~ /\w@ebay\.com/
header __EBAY_RCVD         Received =~ /^from .*\.ebay\.com .* by .*\.ebay\.com.*/m
meta EBAY_MAIL             (__EBAY_MAILTRACKER && __EBAY_MSGID && __EBAY_FROM && __EBAY_RCVD)
describe EBAY_MAIL         Mail is legitimate mail from eBay
```

The rules above are used to try to verify that a message that claims to be from eBay.com really is. Some spammers will use “ebay.com” in the From: addresses, so just testing the From: header is not sufficient. The meta rule “EBAY_MAIL” above only matches if the following are true for a message:

- The message has a “X-eBay-MailTracker” header
- The Message-id: header is an eBay-style message ID
- The From: address claims to be from ebay.com
- There is a Received: header that shows the message was received from an ebay.com system by an ebay.com system

While all of those conditions could be spoofed, spammers do not generally go to such extremes (at least not yet).

Rules referenced in a meta rule test can be logically tested using parentheses for grouping and the boolean operators “&&” (logical AND), “ | | ” (logical OR), and “!” (logical NOT). A more complicated example follows (for readability, the rule is broken into multiple lines, but must be all on one line in the rule file):

```
meta MISSING_OUTLOOK_NAME   ((__HAS_MIMEOLE || __HAS_MSMAIL_PRI)
                             && __HAS_X_MAILER && !__HAS_OUTLOOK_IN_MAILER)
```

Note: Rules whose symbolic names begin with two underscores (such as __EBAY_RCVD above) are assigned a score of 1.0, but do not count toward a message’s final score unless they are used in a matching meta rule, and then only the meta rule’s score is used.

3.7 Rule Scoring

The “score” keyword is used to specify the score for a given rule. There should be a score given for every defined rule; if a rule is not given a score, it automatically is assigned a score of 1.0.

The scores for the pre-supplied rules are stored in the file 50_scores.cf in /pmas/data/. If you want to change the score for an existing rule, you should not edit that file, but should instead add a new score line to the file 99_local_scores.cf

```
score EBAY_MAIL  -10.0
```

If you are defining local rules in 00_local_tests.cf or in 99_local_scores.cf , it’s probably best for tracking purposes to supply the score in the file following the new rule:

```
header DEAR_SUBJECT3         Subject =~ /^\w+@\w+\.\w+, .*/
describe DEAR_SUBJECT3       Subject starts with "e-mail@address, "
score DEAR_SUBJECT3          10.00
```

3.8 Changing Descriptions and Disabling Rules

If you want to change the description for one of the pre-defined rules, or if you'd like to completely disable one of the pre-defined rules, you can specify "disable" and "describe" entries in /pmas/data/ 99_local_scores.cf.

```
disable OBFUSCATED_WORDS
describe PORN_WORDS   Message has what may be naughty words
```

The example above will disable the OBFUSCATED_WORDS test so it won't be applied to messages, and the description of the PORN_WORDS test will be changed to the specified text.

3.9 Adding Allow, Block, Quarantine, Discard, Score, and Reject Rules

In addition to the PreciseMail Anti-Spam Gateway rules described in the preceding sections, PreciseMail Anti-Spam Gateway also supports system-wide and user-defined "allow lists" and "block lists". An allow list is a set of rules that causes matching messages to be automatically forwarded on to the target recipient, while the block list is a set of rules that causes matching messages to be automatically deleted. There are also new rules that allow you to allow, block, quarantine, score, discard, or reject messages.

The system allow/block rules are defined in the file 00_allowblocklists.cf in /pmas/data/. The allow/block rules are applied to messages before any other rules; if a message matches an allow or block rule, it is immediately forwarded or deleted without having the other header and body rules applied to it. This mechanism can be used to let local mail and mail from trusted sources bypass the normal PreciseMail Anti-Spam Gateway checks.

There are six types of rules that can be defined in this file, three allow rules and three block rules:

- **rule** - Rules to allow, block, quarantine, and reject messages
- **Allow_EnvFrom** - Matches envelope FROM addresses
- **Allow_From** - Matches other From headers
- **Allow_Regex** - Regular expression tests for message headers
- **Block_EnvFrom** - Matches envelope FROM addresses
- **Block_From** - Matches other From headers
- **Block_Regex** - Regular expression tests for message headers

Note: The "rule" rules described in Section 3.9.4 can also appear in the system allow/block file.

Each line in the 00_allowblocklists.cf file should contain one of the keywords above and the pattern it matches. Each keyword is described in the sections below. Comment lines in the file are designated by lines that begin with "#" or "!".

```
#
#  Sample allow/block file
#
Allow_EnvFrom *@process.com
Block_EnvFrom *offers*@*
```

Note: Allow rules are applied to the message before block rules, which means that the allow rule has precedence over the block rule. If the same pattern is specified in both an allow list and a block list rule, the allow rule will be used.

User-defined allow/block files are normally created by users through the web-based PMAS GUI or the email-based PreciseMail Anti-Spam Gateway Processor. User files can include any of the six rules described here, though not all of them are documented in the help file for the Processor (intentionally, as it is expected that the ability to create regular expression rules is beyond the skills of most users).

Note: Most users new to PMAS will try to block spam by blocking the email addresses of quarantined spam. Of course, most of those email addresses are not legitimate and will never be used again, so this is generally an ineffective method of stopping spam. However, users may continue to block these addresses, and that can result in user rule files containing tens of thousands of block rules that will never trigger. It takes time to run so many tests, so a configuration variable, MAXIMUM_USER_BLOCK_RULES**, specifies the maximum number of user block rules that will be read and applied. The default value is 1000.**

3.9.1 Allow_EnvFrom and Block_EnvFrom rules

The Allow_EnvFrom and Block_EnvFrom keywords specify patterns that are matched against the envelope FROM address for a message. This address is the address given in the MAIL FROM: command in the original SMTP exchange; it can be found in the Return-Path: header in messages delivered by PMDF.

The pattern for the envelope FROM tests is *not* a regular expression, but is instead a simple wildcard pattern. The characters “?” and “*” are used to match any one character and any number of characters, respectively. If the envelope FROM address matches one of these rules, the message is immediately forwarded or discarded, depending on the type of rule.

```
#
#  Accept all mail from process.com addresses
#
Allow_EnvFrom *@process.com
Allow_EnvFrom *@*.process.com
#
#  Discard all mail from example.com
#
Block_EnvFrom *@example.com
Block_EnvFrom *@*.example.com
```

3.9.2 Allow_From and Block_From rules

The Allow_From and Block_From keywords specify patterns that are matched against several From-related RFC822 headers. While the envelope FROM address is typically a “safer” address (i.e, more likely to be the actual return address), it is often very different from the address from which messages appear to come. For example, a lot of companies outsource their email mailing lists to third-party companies, so a message that is from “xyz.com” may have an envelope FROM address that in no way indicates “xyz.com”. Because this discrepancy tends to confuse users, the Allow_From and Block_From rules can be used to match the addresses that users do normally see, addresses in the RFC822 headers for messages.

The pattern for the FROM tests is *not* a regular expression, but is instead a simple wildcard pattern. The characters “?” and “*” are used to match any one character and any number of characters, respectively. The pattern is compared against the addresses in the following headers, if they are present. The pattern is applied to each of these headers in the order shown until a match is made or the list of headers is exhausted:

- Envelope FROM
- Sender
- Resent-sender
- Resent-from
- X-Envelope-From
- From
- Reply-to

If the given pattern matches the address in any one of these headers, the message is immediately forwarded or discarded, depending on the type of rule.

```
#
#  Accept all mail from these addresses
#
Allow_From goathunter@*
Allow_From *@process.com
#
#  Discard all mail from example.com
#
Block_From *@example.com
Block_From *@*.example.com
```

3.9.3 Allow_Regex and Block_Regex rules

The Allow_Regex and Block_Regex rules specify regular expression patterns that are compared with the RFC822 headers for a message. If a match is made, the message is immediately forwarded or discarded, depending on the type of test.

The *_Regex rules are provided to let you define more sophisticated allow/block tests than just checking From: addresses, although they can also be used to check any of the From-related headers too. Everything after the whitespace after the keyword is considered part of the regular expression.

Note: Because From addresses are easily and often faked in spam messages, it is recommended that you define allow rules for your site that do not just rely on the from address. The example below shows a couple of tests for the Received: headers that are more reliable indicators of local mail than just using the From address.

The following example shows some Allow_Regex rules you might want to implement:

```
#
#  Allow messages with Received: headers that indicate that the
#  message was received by the local system from the local system.
#
Allow_Regex  Received: from PROCESS\.COM by PROCESS\.COM.*
#
#  Also match Received: headers showing the local system received the
#  message from another system in our domain (and with one of our
#  IP addresses).
#
Allow_Regex  Received: from (?:\w*\.)*example\.com \(\[(232.45|44.95.42).*\]*\) by EXAMPLE\.COM.*
#
#  Allow messages that have already been processed by PreciseMail Anti-Spam Gateway
#
Allow_Regex  X-PMAS-Final-Score: .*
#
#  Allow messages that have an X-Listname: header that indicates
#  it’s a mailing list message from a process.com mailing list.
#
Allow_Regex  X-Listname: .*@process\.com>
```

And the following example shows some typical Block_Regex rules you might want to implement:

```
#
#  Block the common Viagra spam
#
Block_Regex  Subject: .*[Vv][iI1][Aa][Gg][Rr][Aa].*
#
#  Block messages that are nice enough to tell us they’re ads
#
Block_Regex  Subject: ADV:.*
#
#  Block some common virus subjects
#
Block_Regex  Subject: .*W32\.Klez\.E removal tools.*
Block_Regex  Subject: .*W32\.Elkern  removal tools.*
```

3.9.4 Rule rules

The “rule” rules grant the administrator and users more control over message disposition. They include the functionality of the “allow_” and “block_” rules, as well as several other options. These rules can appear in either the system-wide allow/block file or in each user’s allow/block file. These rules are normally defined using the PMAS GUI.

The format of the “rule” rules is:

```
rule action object conditional-phrases "text"
```

The first word, “rule”, is a keyword identifying the line as a “rule” rule.

The “action” specifies the action to be taken for messages triggering the rules. The following table lists the valid actions.

allow	Allow the message through without processing the message
block	Block the message (automatically delete it)
score	Apply the specified score to the message’s overall score
quarantine	Quarantine the message
discard	Discard the message
reject	Reject the message during the SMTP dialogue (PTSMTP-only)
megareject	Reject the message during the SMTP dialogue, regardless of any allow rules that might match (PTSMTP-only)

The “object” specifies the item to which the test is to be applied. The following table lists the valid objects:

subject	Tests the RFC2822 Subject: header
to	Tests the RFC2822 To: header
from	Tests the From:, Reply-to: and Sender: RFC822 headers
cc	Tests the RFC2822 Cc: header
received	Tests the RFC2822 Received: headers
header	Tests an arbitrary RFC2822 header; the header to test is specified after the keyword, separated by a colon: *header:X-PMAS-Internal*
body	Tests the message body after all decoding has been done, and after all binary parts, HTML tags, and line breaks have been removed
rawbody	Tests the ‘‘raw’’ message body, after all decoding has been done, but with all HTML tags and line breaks present
full	Tests the ‘‘full’’ message body, before any kind of processing is done
envelope_from	Tests the RFC2821 envelope MAIL FROM: address
envelope_to	Tests the RFC2821 envelope RCPT TO: addresses
envelope_to_alias	Tests the envelope RCPT TO: addresses after PMAS alias processing has occurred

The “conditional-phrase” specifies the type of test that is to be performed on the specified object. The following table lists the valid conditional-phrases.

exists	Tests that the specified object exists; no text is specified
noexists	Tests that the specified object does not exist; no text is specified
contains	Tests that the object contains the specified text; a match will occur if the text exists anywhere in the object

nocontains
Tests that the object does not contain the specified text; a match will occur if the text does not exist anywhere in the object

contains_words
Tests that the object contains the specified word or phrase; a match will occur only if the text matches full words

nocontains_words
Tests that the object does not contain the specified word or phrase; a match will occur only if the text does not occur

starts
Tests that the object starts with the specified text

ends
Tests that the object ends with the specified text

is
Tests that the object exactly matches the specified text

matches_wild
Tests that the object matches a wildcarded string; wildcards “*” (match 0 or more of any character) and “?” (match any single character) can be used

matches_regexp
Tests that the object matches the specified regular expression

Finally, the “text” is a quoted string that specifies whatever text the conditional-phrases require. Double-quotes around the text are required.

Note: All of the “rule” rules are case-insensitive, though the “matches_regexp” rule can be written so that it’s case-sensitive.

Some example rules follow:

```
rule block subject contains "virus detected"
rule allow subject is "PreciseMail User Spam Report"
rule allow header:X-PMAS-External noexists
rule quarantine subject contains_words "sex"
rule allow subject starts "[McCammon-News]"
rule block subject matches_regexp ".*(?-i:WARNING).*"
rule reject header:X-PMAS-External matches_regexp "\((?:HELO|EHLO)\s+my.domain\)"
rule score=-10.00 body contains_words "PreciseMail"
```

The “reject” action, which is only applicable to the PMAS PTSMTP proxy server, allows for an optional sixth parameter that specifies the rejection text to be issued during the SMTP dialogue. The default text is:

```
550 5.7.1 Requested mail action not taken: rejected for policy reasons
```

If a “reject” rule is encountered by the PMAS PMDF channel, it is treated like a “block” command.

The “megareject” action, which is only applicable to the PMAS PTSMTP proxy server, is applied before and takes precedence over user and system user allow rules; it also applies to opted-out users. It is designed to be used to stop messages with particular “Subject:” headers, body contents, etc, that should never be allowed into the system.

Only system administrators can define “megareject” rules. They are ignored in user rule files.

The “score” action takes a numeric value that is applied to the message’s overall spam score. The value is specified after the “score” verb, separated from it by an equal sign “=”:

```
rule score=-10.00 body contains_words "precisemail"
rule score=100.00 subject contains_words "virus"
```

Negative scores can be applied to compensate for other positive-score rules that may apply to the message.

Note that if any but one of the the rule rules are triggered for a message, processing of the message ends at that point and no further rules are applied. The exception to this is the “score” rule. Multiple “score” rules may be applied to a message, and all normal PMAS message scanning is performed.

3.10 Performing DNSBL lookups on URIs and Received: IP addresses

When processing messages, PMAS can perform DNSBL (DNS-based Blackhole List; see Section 1.4.5 for more information about DNSBL) lookups on the URIs in a message body, as well as on the IP addresses specified in the “Received:” headers for a message. Some DNSBLs maintain separate lists for URIs; please consult the documentation for the DNSBL you plan to use as to which list is the best to use for URI lookups.

3.10.1 URI DNSBL lookups

If URI DNSBL lookups are enabled, a DNSBL lookup is performed for each URI in the message body. In the interest of performance, lookups end if a match is found.

Note: If you are not running a nameserver on the local system (127.0.0.1), you must define the PMAS configuration variable UDNS_NAMESERVER **to specify the IP address of the nameserver to use:**

```
udns_nameserver 10.1.1.12
```

The default nameserver used is 127.0.0.1.

URI DNSBL lookups are configured by PMAS configuration variables and via keywords in the configuration file /pmas/data/pmas_dnsbl.conf (which can be created from the supplied pmas_dnsbl.template file). The valid URI DNSBL configuration variables and keywords are defined in Table 3–4.

Table 3–4 URI DNSBL configuration variables

Keyword	Description
URI_DNSBL_ENABLED	Enables URI DNSBL lookups. Must be defined as “yes” or “no” (the default).
URI_DNSBL_SCORE	Specifies the score that should be applied to the messsage’s overall score if the URI domain is listed in a specified DNSBL. The default value is 10.0.
URI_DNSBL_LOG_ENABLED	Specifies whether or not successful URI DNSBL lookups are logged to A new log file is automatically created each day.

Table 3–5 describes the Received: DNSBL keywords.

Table 3–5 URI DNSBL keywords

Keyword	Description
URIDNSBL	Consults the named DNSBL site for each URI found in a message body. The PMAS configuration variable URI_DNSBL_ENABLED must be defined as "yes". If a URI's host is found in the specified DNSBL, a header indicating this is added to the message's headers, and the score amount defined by the configuration variable URI_DNSBL_SCORE is added to the message's overall score. An optional comma-separated list of values can be used to limit matches to specific return addresses from the DNSBL (e.g, 127.1.0.1).
URIDNSBL_SKIP_DOMAIN	Specifies domain names for which uridnsbl lookups should not be performed. Sites that are legitimate but falsely appear in DNSBLs can be specified. Multiple keywords can be specified to skip multiple domains.
UDNS_NAMESERVER	Specifies the IP address of the nameserver to use for DNS lookups
UDNS_DEBUG	Turns on debugging output for the uDNS routines

This sample configuration would result in DNSBL lookups using the Spamhaus SBL-XBL list for all URIs in a message body. URIs for the domain "process.com" would not be looked up.

```
uridnsbl sbl-xbl.spamhaus.org
uridnsbl_skip_domain process.com
```

3.10.2 Received: IP address DNSBL lookups

If Received: DNSBL lookups are enabled, a DNSBL lookup is performed for each IP address found in the "Received:" headers for each message. Internal IP addresses (see Section 1.4.2) are ignored. The lookups can also be configured to ignore specific addresses.

Received: DNSBL lookups are configured by PMAS configuration variables and via keywords in the configuration file /pmas/data/pmas_dnsbl.conf (which can be created from the supplied pmas_dnsbl.template file). The valid Received: DNSBL configuration variables and keywords are defined in Table 3–6

Table 3–6 Received: DNSBL configuration variables

Keyword	Description
RCVD_DNSBL_ENABLED	Enables Received: DNSBL lookups. Must be defined as "yes" or "no" (the default).
RCVD_DNSBL_SCORE	Specifies the score that should be applied to the messsage's overall score if the URI domain is listed in a specified DNSBL. The default value is 10.0.
RCVD_DNSBL_LOG_ENABLED	Specifies whether or not successful Received: DNSBL lookups are logged to A new log file is automatically created each day.

Table 3–6 (Cont.) Received: DNSBL configuration variables

Keyword	Description
RCVD_SKIP_1ST_RCVD	Specifies whether or not the first Received: header is processed for DNSBL lookups. When the PMAS PTSMTP proxy server is used, the first Received: header is the header added by PMAS. If DNSBL lookups are performed when the connection is first made, processing the address in the first Received: header would be redundant. By setting this variable to "yes" (which is the default), the IP addresses in that Received: header will be examined.

Table 3–7 describes the Received: DNSBL keywords.

Table 3–7 Received: DNSBL keywords

Keyword	Description
RCVDDNSBL	Consults the named DNSBL site for each IP address found in a message's Received: headers. The PMAS configuration variable RCVD_DNSBL_ENABLED must be defined as "yes". If a Received: IP address is found in the specified DNSBL, a header indicating this is added to the message's headers, and the score amount defined by the configuration variable RCVD_DNSBL_SCORE is added to the message's overall score. An optional comma-separated list of values can be used to limit matches to specific return addresses from the DNSBL (e.g, 127.1.0.1).
RCVDDNSBL_SKIP_ADDRESS	Specifies IP addresses for which uridnsbl lookups should not be performed. Sites that falsely appear in DNSBLs can be specified. Multiple keywords can be specified to skip multiple addresses.

This sample configuration would result in DNSBL lookups using the Spamhaus SBL-XBL list for all IP address found in the Received: headers for a message. The IP address "10.0.0.1" would not be looked up.

```
rcvddnsbl sbl-xbl.spamhaus.org
rcvddnsbl_skip_address 10.0.0.1
```

3.11 Reverse DNS lookups for URIs

PMAS can perform reverse DNS lookups on domain names specified in URIs in a message body. Reverse DNS lookup (rDNS) is a process to determine the hostname or host associated with a given IP address or host address. Most legitimate systems will have rDNS entries defined, but a lot of spam domains will not. By performing an rDNS lookup on each of the URIs, potential spam sites can be identified.

Note: It is entirely possible that legitimate systems will not pass rDNS lookups. Some site DNS administrators do not define PTR records for all of their IP addresses. This can include some well-known sites, like MySpace; "www.myspace.com" does not have an rDNS PTR record. Caution should be used when assigning scores based on rDNS lookups, and results should be logged and the log file examined periodically for any false positives.

Note: If you are not running a nameserver on the local system (127.0.0.1), you must define the PMAS configuration variable UDNS_NAMESERVER **to specify the IP address of the nameserver to use:**

```
udns_nameserver 10.1.1.12
```

The default nameserver used is 127.0.0.1.

rDNS lookups on URIs is controlled by the PMAS configuration variables shown in Table 3–8.

Table 3–8 URI rDNS lookup configuration variables

Keyword	Description
RDNS_URI_ENABLED	Enables URI rDNS lookups. Must be defined as “yes” or “no” (the default).
RDNS_URI_SCORE	Specifies the score that should be applied to the messsage’s overall score for each URI domain that does not have an rDNS entry. The default value is 5.0.
RDNS_LOG_ENABLED	Specifies whether or not rDNS lookup failures are logged to A new log file is automatically created each day.
UDNS_NAMESERVER	Specifies the IP address of the nameserver to use for DNS lookups
UDNS_DEBUG	Turns on debugging output for the uDNS routines

Known hostnames or addresses that do not have rDNS entries defined can be listed in the configuration file /pmas/data/rdns_exceptions.txt. Each line should specify a hostname that should be excluded from rDNS URI checks. Wildcards ’*’ and ’?’ are supported.

```
# Don’t check MySpace
*.myspace.com
```

A template file with some known non-rDNS domains is provided as /pmas/data/rdns_exceptions.template.

4 PreciseMail Anti-Spam Gateway cron Jobs

This chapter describes the PreciseMail Anti-Spam Gateway Notify and PreciseMail Anti-Spam Gateway Nightly cron jobs.

4.1 PreciseMail Anti-Spam Gateway Notify

If PreciseMail Anti-Spam Gateway is configured to quarantine or discard messages, you should run the PreciseMail Anti-Spam Gateway Notify cron job. This job is responsible for performing the following tasks:

- Delivering quarantine notification messages to users for whom messages have been quarantined
- Deleting quarantined messages that are older than the configured message lifetime (14 days by default)
- Deleting discarded messages that are older than the configured message lifetime (14 days by default)

The program that is executed as the PreciseMail Anti-Spam Gateway Notify cron job is /pmas/bin/notify_quarantined. It should be added to the cron job table of the user as which the mail server runs as described in the *PreciseMail Anti-Spam Gateway Installation Guide*.

The configuration variables for the PreciseMail Anti-Spam Gateway Notify cron job are described in Section 2.2.15, Quarantine Notification Batch Job.

4.2 PreciseMail Anti-Spam Gateway Nightly

The PreciseMail Anti-Spam Gateway Nightly cron job runs at 12:05 AM each night. Its job is to manage the files in the PreciseMail Anti-Spam Gateway log directory, removing old debug files and renaming the master log file from .log to .log-yyyy-mm-dd so that a new .log file is created each day.

The script that is executed as the PreciseMail Anti-Spam Gateway Nightly cron job is /pmas/com/run_nightly.sh. It should be added to the cron job table of the user as which the mail server runs as described in the *PreciseMail Anti-Spam Gateway Installation Guide*.

4.3 PreciseMail Anti-Spam Gateway Statistics Maintenance

The statistics maintenance cron job runs every hour on the hour. Its job is to update the internal files used by the web statistics engine.

The program that is executed as the statistics maintenance cron job is /pmas/bin/bastats. It should be added to the cron job table of the user as which the mail server runs as described in the *PreciseMail Anti-Spam Gateway Installation Guide*.

5 Additional PreciseMail Anti-Spam Gateway Programs

This chapter describes additional utility programs provided with PreciseMail Anti-Spam Gateway.

5.1 The pmasadmin Program

The pmasadmin program allows system administrators to perform some PMAS user and group management from the command line. pmasadmin takes several arguments, depending on the operation being performed. Valid commands are:

-
  ```
  pmasadmin user create EMAIL PASSWORD
  ```

  Create a new user with the specified email address and password in the user database. Usually, users are automatically added to the user database by PreciseMail Anti-Spam Gateway. You should only need to run this command if you are adding a special user that does not exist on your mail server.

-
  ```
  pmasadmin user set_password EMAIL PASSWORD
  ```

  Change the password for a user in the user database. If you have specified "PMAS" as an available authentication method, the specified user can login to the web interface with the specified password.

-
  ```
  pmasadmin user delete EMAIL
  ```

  Remove the specified user from the user database. You may specify "NOCONFIRM" after the user's email address to avoid being prompted for confirmation of deletion.

-
  ```
  pmasadmin user optin EMAIL
  ```

  Scan all incoming mail for the specified user.

-
  ```
  pmasadmin user optout EMAIL
  ```

  Do not scan any of the specified user's mail.

-
  ```
  pmasadmin user details EMAIL
  ```

  Display the filter settings for the specified user.

-
  ```
  pmasadmin user rename OLD-EMAIL NEW-EMAIL
  ```

  Renames the record for the specified user.

-
  ```
  pmasadmin user dump [filename]
  ```

  Dumps an ASCII representation of the contents of the user database to the screen or the named file. Used to transfer the database contents between different types of systems.

-
  ```
  pmasadmin user load filename
  ```

  Loads the user database with the contents of the file created with the “user dump” command.

-
  ```
  pmasadmin group create GROUP
  ```

  Create a group of users with the specified group name.

-
  ```
  pmasadmin group delete GROUP
  ```

  Delete the specified group.

-
  ```
  pmasadmin group list
  ```

  Display a list of all groups.

-
  ```
  pmasadmin group details GROUP
  ```

  Display the filter settings that apply to members of the specified group.

-
  ```
  pmasadmin group add_member GROUP EMAIL
  ```

  Add the specified user to the specified group.

-
  ```
  pmasadmin group delete_member GROUP EMAIL
  ```

  Delete the specified member from the specified group.

-
  ```
  pmasadmin group list_members GROUP
  ```

  Display a list of the members of the specified group.

-
  ```
  pmasadmin group get_membership EMAIL
  ```

Determine and display the effective group membership for the specified user.

5.2 The BAYES Program

If you elect to use the Bayesian engine (Section 2.2.12), you can use the bayes program (found in /pmas/bin/) to train the Bayesian databases interactively. The program can be used to train for both “spam” and “ham” (legitimate email). When training the databases, files containing the entire email message (including all headers) should be specified as input to the BAYES program. The BAYES program accepts one parameter, either “-s” (the input file is spam) or “-h” (the input file is ham):

```
$ bayes -s < spam.txt
$ bayes -h < ham.txt
```

6 PreciseMail Anti-Spam Gateway User Interface

This chapter describes the PreciseMail Anti-Spam Gateway web-based user interface from a system administrator view.

The PreciseMail Anti-Spam Gateway User Interface is implemented as a series of scripts and pages served by a web server. Users must “log in” to the PMAS UI by specifying an email address and a password. The authentication methods supported by PreciseMail Anti-Spam Gateway are outlined in Section 2.2.11.

Once you have configured the web server to serve the PreciseMail Anti-Spam Gateway files, as outlined in the PreciseMail Anti-Spam Gateway Installation Guide, your users will be able to use the interface using a URL similar to the following:

```
http://yournode.example.com/pmas/index.html
```

By default, the URL for the quarantine area is also included in the quarantine notices that are mailed to users.

The user interface is driven from a set of HTML template files stored in the directory /pmas/html . There are two sets of files located there: .html files and .template files. The .template versions are the original versions created when PreciseMail Anti-Spam Gateway was installed. They may change with each release of PMAS. The files can be customized on a per-site basis by copying the .template files to .html and editing the new file.

Note: **The CGI programs will look for each .html file and, if there is no .html file, the .template files will be used. It is not necessary to copy the .template files to .html unless you want to customize the files.**

Note: **The .template files will be replaced with each installation of PreciseMail Anti-Spam Gateway. If you customize the files, be sure you name your customized versions with a .html extension. You will also be required to merge any local customizations with future .template updates.**

6.1 Customizing the HTML template files

Process Software does not recommend that sites customize the HTML template files that make up the various screens for the user interface. However, we also recognize that many sites will want to disable certain features, add site-specific logos, or change the base language for the pages. All of these actions can be performed by simply copying the .template file to .html and editing the .html file with any text editor.

Extreme care should be taken when modifying the HTML source files. They contain various page-specific tags that are filled in dynamically as the pages are served by the CGI script programs. These tags take the form *%name%*. Some examples include *%MSG_DATE%*, *%ADDRESS%*, and *%EMAIL_ADDRESS%*. Some of the tags are used as placeholders for information. In general, if a tag is located on a line by itself, it must remain in the file on a line by itself in order for the pages to be properly filled in by the CGI script.

A working knowledge of HTML is essential for anyone attempting to modify the HTML template files. The .template versions should never be modified directly.

6.2 Controlling User Preferences

There are a number of configuration keywords that can be used to control the user preferences that are available via the PreciseMail Anti-Spam Gateway user interface. For full descriptions of these keywords, please see Section 2.2.10, User Interface keywords.

6.3 The PMAS Administrator Interface

The web-based GUI user interface can also be used by the PMAS system administrator to configure PMAS or change the PMAS configuration. To log in to the administrator page, a special login name, *PMAS_ADMIN*, is used. When PMAS is initially installed, a default password for the PMAS_ADMIN account is written to the PMAS user database file.

Note: The PASSWD utility, supplied in /pmas/bin, can be used to change the PMAS_ADMIN password from its default value, as well as to define passwords for other email addresses. See the PreciseMail Anti-Spam Gateway Installation Guide for more information on changing the administrator password.

Once you are logged in as the PMAS_ADMIN, you'll be able to configure PMAS or view reports that detail the message processing performed by PreciseMail Anti-Spam Gateway.

7 PreciseMail Clustering

PreciseMail Anti-Spam Gateway currently provides two methods to centralize management and consolidate data for email domains that have multiple MTA systems. Data Synchronization Cluster (DSC) technology allows you to completely consolidate up to three MTA systems. The new Advanced Infrastructure system is designed to handle up to 15 high-volume MTA systems.

Note: **Advanced Infrastructure is being phased into PreciseMail Anti-Spam Gateway over the course of several releases. The Advanced Infrastructure module included in PreciseMail V3.0 allows configuration and statistics to be centralized for up to 15 high-volume MTA systems. When completely released, Advanced Infrastructure will deprecate DSC.**

Note: **Advanced Infrastructure and DSC are not compatible systems - you may not run both of them on the same system.**

7.1 Advanced Infrastructure

Advanced Infrastructure (AI) is the next-generation replacement technology for PreciseMail clustering. Currently, AI allows sites to consolidate configuration and filtering statistics for up to 15 systems running PreciseMail.

7.1.1 Setting Up Advanced Infrastructure

AI runs in one of two modes: simple or advanced. Simple mode is a basic client/server system with one master server and one or more clients who depend on it. Advanced mode allows cluster tasks to be distributed across multiple systems, making it suitable for complex high availability and large volume sites. Regardless of whether you choose to run AI in simple or advanced mode, the basic setup tasks are the same for every system in the cluster:

1 Enable AI by setting the value of the configuration variable AI_ENABLED to "yes".

2 Make sure that all of the systems in the AI cluster are able to communicate with each other on TCP port 17867. (If a firewall or router blocks incoming or outgoing connections to this port, AI cannot function properly.) You can select the TCP port used by AI by changing the value of the AI_TCP_PORT configuration variable. **If you change the value of this variable, you MUST change it to the same value on every system in the cluster.**

3 Change the value of AI_AUTH_TOKEN to a site-specific password. This password will be used by the members of the AI cluster to verify that they are communicating with other valid cluster members.

Once you have performed these basic setup tasks on each system in the cluster, decide whether you simple mode or advanced mode is more suited to your site and follow the appropriate instructions.

7.1.2 Simple Mode

In "simple" mode, an AI installation has one server node and one or more client nodes. The server node performs all of the cluster server roles, and each client node performs all of the client roles. (For a detailed list and description of cluster roles, see Section 7.1.4 in this document.)

To enable simple mode, perform the following tasks:

1 On every system in the cluster except for the server node, set the value of the AI_PRIMARY_NODE configuration variable to the host name of the server node.

2 On the server system, set the value of the AI_SECONDARY_NODES configuration variable to the host names of each of the client systems, separated by a comma.

3 Restart PreciseMail Anti-Spam Gateway.

7.1.3 Advanced Mode

In "advanced" mode, cluster tasks can be distributed across multiple systems. Before configuring an AI cluster in advanced mode, you should have a good understanding of clustering tasks as described in Section 7.1.4

To enable advanced mode, perform the following tasks:

1 On every system in the cluster, set the value of the AI_ENABLE_ADVANCED configuration variable to "yes".

2 On the system you wish to be the configuration server, set AI_CONFIG_CLIENTS to a comma-separated list of the host names of every configuration client in the cluster.

3 On every cluster system except for the configuration server, set the value of AI_CONFIG_SERVER to the host name of the configuration server.

4 On the system you wish to be the statistics server, set AI_STATS_CLIENTS to a comma-separated list of the host names of every statistics client in the cluster.

5 On every cluster system except for the statistics server, set the value of AI_STATS_SERVER to the host name of the statistics server.

6 On the system you wish to be the user rules server, set AI_USERRULES_CLIENTS to a comma-separated list of the host names of every user rules client in the cluster.

7 On every cluster system except for the configuration server, set the value of AI_USERRULES_SERVER to the host name of the user rules server.

8 On the system you wish to be the groups files server, set AI_GROUPS_CLIENTS to a comma-separated list of the host names of every groups files client in the cluster.

9 On every cluster system except for the configuration server, set the value of AI_GROUPS_SERVER to the host name of the groups files server.

10 Restart PreciseMail Anti-Spam Gateway.

7.1.4 Advanced Infrastructure Roles

Under Advanced Infrastructure, cluster tasks are split into distinct roles. Each role has both a server and a client component. Cluster systems may perform any combination of roles, as long as they aren't both the server and client components of the same task (i.e., a system may be both a configuration server and a statistics server, but a system may not be both a configuration server and a configuration client). Below is a list of roles currently supported by AI with a brief description of each:

7.1.4.1 Configuration Server

The system performing the configuration server role is responsible for making sure every client system in the cluster has an identical basic configuration. The configuration files kept in sync are:

- 00_allowblocklists.cf (site-wide allow/block list)
- 00_local_tests.cf (site-specific rules)
- 99_local_scores.cf (site-specific rule scores)
- aliases.txt (user aliases)
- internal_ip.txt (list of internal IP addresses)
- local_domains.txt (list of site'slocal domains)
- pmas_config.dat (master PreciseMail configuration file)

When PreciseMail Anti-Spam Gateway is restarted on the server or a change is made to a configuration file in the administrator's web interface, any changes made to the configuration files are mirrored on the clients. Every 5 minutes the AI housekeeper process checks the configuration files to see if they have changed, and mirrors the changes on the client systems if they have.

7.1.4.2 Configuration Client

A configuration client listens for and implements configuration changes from the configuration server. On startup and again every 4 hours, the AI housekeeper process queries the configuration server to verify that the client has the most current set of configuration files.

7.1.4.3 Statistics Server

A statistics server listens for incoming filtering statistics information from statistics clients. The incoming data is integrated with the SQL statistics databases on the server, where it's available through the administrator's web interface.

7.1.4.4 Statistics Client

Rather than update local statistics databases, the hourly bastats process on a statistics client generates an SQL datastream which is sent to the statistics server.

7.1.4.5 User Rules Server

A user rules server stores the files containing all of the user-defined rules. Any changes made to the user rules files are mirrored on the clients. Every 5 minutes the AI housekeeper process checks the user rule files to see if they have changed, and mirrors the changes on the client systems if they have.

7.1.4.6 User Rules Client

A user rules client listens for and implements user rules changes from the user rules server. On startup and again every 4 hours, the AI housekeeper process queries the user rules server to verify that the client has the most current set of user rules.

7.1.4.7 Groups Server

A groups server stores the files containing all of the group definitions and settings. Any changes made to the groups files are mirrored on the clients. Every 5 minutes the AI housekeeper process checks the groups files to see if they have changed, and mirrors the changes on the client systems if they have.

7.1.4.8 Groups Client

A groups client listens for and implements group changes from the groups server. On startup and again every 4 hours, the AI housekeeper process queries the groups server to verify that the client has the most current set of groups files.

7.1.5 Advanced Infrastructure Processes

Every node in an Advanced Infrastructure cluster runs two daemon processes:

- AI cluster daemon (pmas_ai) - responsible for handling incoming cluster requests from other cluster nodes.
- AI Housekeeper (pmas_ai_housekeeper) - performs scheduled tasks to ensure that the cluster is operating properly and to resolve any problems that may have occured.

7.2 Data Synchronization Clusters

PreciseMail Anti-Spam Gateway's Data Synchronization Cluster (DSC) technology allows sites to consolidate configuration, management, and data for email domains that have 2 or 3 MTA systems.

7.2.1 Overview

Many email architectures make use of one or more gateway systems that scan incoming mail for spam and viruses before passing it on to a backend mailstore. If there is more than one such gateway system, configuring the systems as a DSC allows them to share data to simplify both management and end-user access.

The DSC functionality defines one node as the primary node, and one or more nodes as secondaries. All administration and user operations are performed once on the primary node, and the changes are automatically propagated out to the secondary nodes. In addition, all messages that are quarantined or discarded on secondary nodes are coalesced on the primary node.

Note: The DSC technology itself is not resource intensive, but running it on heavily loaded systems will lead to a degraded user experience. Please make sure that the member systems of a DSC are not overloaded and have a high-speed, low-latency network link.

Note: While it is theoretically possible to have an unlimited number of secondary nodes in a DSC, it is recommended that there be no more than two secondary nodes with the current DSC implementation.

7.2.2 Configuring a Data Synchronization Cluster

Perform the following steps to set up a DSC:

1 Choose one node to be the primary node of the cluster. Remember that this node needs to have enough disk space to store quarantined and discarded messages for every node in the cluster. Install and configure PMAS on this system as you normally would. Make sure you add the new clu_batch cron job, as described in the Post-Installation section of the Installation Guide for your platform.

2 Choose one or more nodes to be secondary nodes in the cluster. Install PMAS normally on each of them. Using either the web-based administration interface or editing pmas_config.dat, set the DSC configuration variables to specify the current node as a secondary node and the hostname of the primary cluster node.

Note: All PMAS configuration settings, user settings, aliases, and local rules on secondary nodes will be removed and replaced with settings from the primary node. Existing quarantined and discarded files will not be touched.

3 On each secondary node, run the pmas start script (located in /etc/init.d/). Use the ps command to verify that the pmas_cluster process is running.

4 On the primary node, set the DSC configuration variables to specify the current node as a primary node and the names of the secondary nodes. If you do this with the web-based administration interface, the commands to initially synchronize the new secondary nodes are automatically generated for you. If you edit pmas_config.dat by hand, you must create the file /pmas/data/clu_queue.dat and add one line for each secondary node that looks like:

```
host.example.com SYNC_SLAVE
```

Make sure the file is owned by the user that the MTA runs as.

5 Either manually run the /pmas/bin/clu_batch utility on the primary node or wait for cron to run it. When clu_batch runs, it will perform an initial sync on each of the secondary cluster nodes. If you have a large number of users with a lot of personalized data, this process may require several minutes.

6 Make sure that the web-based and email-based user interfaces on the secondary nodes are disabled or hidden. Changes made on secondary nodes will not be retained.

7.2.3 High Availability

Strictly speaking, DSC is not a high availability solution. A failure in one node of the cluster may make some user-level services inaccessible, but it will not stop all incoming mail.

If a secondary cluster node becomes inaccessible, all synchronization commands from the primary node will be queued up. When the secondary node becomes available again, the queued up actions will be performed.

If a primary cluster node becomes inaccessible, users and administrators will not be able to access quarantined messages or change settings and preferences. However, all secondary nodes will continue to filter messages so the domain will continue to receive mail that has been filtered for spam. Shortly after the primary node is returned to operation, all messages that the secondary nodes quarantined or discarded will be available to end users and administrators.

8 Anti-Virus Scanning

The PreciseMail Pass-Through SMTP (PTSMTP) Server includes support for calling out to an anti-virus package to scan incoming messages for viruses before they are scanned by the PMAS spam engine. Currently, ClamAV and the Sophos Anti-Virus package are supported.

Section 2.2.9, Anti-Virus Package keywords, describes the configuration variables that control the anti-virus scanning.

All message parts are scanned by the anti-virus engine (text, HTML, and binary attachments). When a virus is detected in an incoming message, the infected body part (which is most likely, but doesn't have to be, the attachment) is automatically replaced by a text message indicating that a virus was found and removed from the message. The body part that replaces the infected attachment can be found in the file /pmas/data/virus_replacement.template. You can customize the text for your site by copying that file to /pmas/data/virus_replacement.txt and editing it as desired (you can also change it from a plain-text part to HTML or any other valid MIME bodypart).

When a message is scanned by the PMAS anti-virus plugin, it receives RFC822 headers that indicate the results of the scans. All messages will receive a header similar to this one, indicating the version of the anti-virus engine and the system name:

```
X-PMAS-Sophos: PMAS/Sophos [2.28.4, 3.91, 7/3/2005] (ursus.example.com)
```

If no virus is found when the message is scanned, a header like the following is added to indicate that the message was scanned and no virus was identified:

```
X-PMAS-NoVirus: No virus found
```

If one or more viruses are found in the message, PMAS will add a header indicating the number of viruses found and removed, as well as a header identifying each virus that was removed:

```
X-PMAS-Virus-Found: 1 virus detected and removed!
X-PMAS-Virus-01: W32/Netsky-P
```

Note: The string "PMAS" in the examples above can be customized using the HEADER_PREFIX **configuration variable.**

Messages containing viruses can be rejected, blocked, discarded, quarantined, or forwarded to the user. The action taken is controlled by the configuration variable VIRUS_DISPOSITION. For details of the options available, see Section 2.2.9.4.

Messages containing viruses are logged in the normal PMAS log file (see Section 1.7). In place of matched rule names, the PMAS log entry for a message containing a virus will show the virus name or names that were detected. Each virus name will be prefaced with the string "VIRUS_" and will look similar to this:

```
VIRUS_W32_NETSKY_P
```

You can easily locate virus entries by searching the log file for the string " | VIRUS_":

```
> egrep "\|virus_" /pmas/log/pmas.log
```

8.1 System Considerations for Anti-Virus Scanning

Before you enable anti-virus scanning via the PTSMTP Proxy Server, you should be aware of the implications of doing so regarding your system performance.

8.1.1 System Memory Demands

When anti-virus scanning is enabled, each of the PMAS PTSMTP Proxy Server worker processes will load the anti-virus rules into memory. The Sophos databases alone currently consume approximately 8 megabytes, which means that a lot of system memory/pagefile space can be taken up the PTSMTP worker processes. In addition, the proxy server is not a store-and-forward system, which means that each incoming mail message is held in RAM memory, and the process's memory quotas will determine how large a message can be scanned.

Note: PMAS V2.2 does not yet include a check for the maximum message size to scan, but this functionality will be added in a future release.

Loading the anti-virus databases is a disk-I/O-intensive operation. It can take anywhere from a few seconds to several minutes for all of the worker processes to load the anti-virus databases, depending on system type, system load, and disk speed. It is highly recommended that you do *not* specify a range of worker processes (see Section 2.2.8.14 and Section 2.2.8.15, but instead determine a static number of processes to run. If you allow worker processes to be created as-needed, be aware that your system performance may suffer during the creation of those temporary processes as they load the anti-virus databases.

Once the databases are loaded, the actual scanning of message bodies is very fast, and the databases aren't reloaded until they've been updated. For PMDF sites, this will result in tremendous performance savings over the traditional method of running Sophos VSWEEP for each body part via the conversion channel, but it comes at the cost of more system memory being consumed by the worker processes.

Note: Automatic database updates will be added in a future PreciseMail Anti-Spam Gateway release. For now, when you update the anti-virus databases, you should restart the PMAS PTSMTP processes by re-running the PMAS startup procedure.

8.1.2 Configuring the Sophos Anti-Virus Engine

The Sophos Anti-Virus engine has a default configuration that determines which types of files are to be scanned for viruses and other aspects of the scanning process. Please see the Sophos documentation for details of the default values for the various options.

PMAS allows you to customize the Sophos AV options using the file /pmas/data/pmas_sophos_config.dat. A default template file, /pmas/data/pmas_sophos_config.template, is provided when PMAS is installed. To customize the options, simply copy the .TEMPLATE file to the .DAT file and then edit the .DAT file as desired. Changes to the file will be automatically detected by the various worker processes.

8.1.3 Configuring the ClamAV Anti-Virus Engine

PMAS PTSMTP supports calling out to a system running ClamAV to scan messages for viruses. Three configuration variables control this:

Table 8–1 ClamAV configuration variables

Keyword	Description
CLAMD_HOST	Specifies the hostname or IP address of the system running clamd
CLAMD_PORT	Specifies the port on which clamd is listening
NO_VIRUS_SCAN_SUBJECT	Specifies text that is prepended to Subject: headers for messages that aren't scanned by clamd (for example, if the clamd system is unreachable)

9 Debugging PreciseMail Anti-Spam Gateway

This chapter describes techniques for debugging the PreciseMail Anti-Spam Gateway programs.

As described in Chapter 2, Configuring PreciseMail Anti-Spam Gateway, configuration variables are available to determine the amount of debugging output displayed by PreciseMail Anti-Spam Gateway programs and where that information is written (the names of log files).

For serious PreciseMail Anti-Spam Gateway problems, alarms will be generated via the UNIX syslog facility. The alarms will be sent to syslog with a priority of "error" under the "mail" facility.

In addition to turning on debugging and looking at the generated log files, it's possible to run the PreciseMail Anti-Spam Gateway main image interactively to see how it processes a particular message. On UNIX, the main PreciseMail Anti-Spam Gateway image is located at /pmas/bin/precisemail. It takes up to two parameters: the name of a file containing a mail message to be scanned (the entire message, include the RFC822 headers) and, optionally, the name of an output file if the message is modified (headers added, etc.). (Environment variables for the envelope FROM and TO addresses must be defined.) The example below shows a sample run.

Example 9–1 Example Interactive Run of PMAS

```
# cat X.X
Received: from example.com by sample.com (PMDF V6.2-plus #36614)
 id <01KXBK4CLEXC8WW21B@example.com>
 (original mail from GOATHUNTER@goatley.com) for goathunter+spam@goatley.com
 (ORCPT bobuser@sample.com); Fri, 20 Jun 2003 14:27:07 -0500 (CDT)
Date: Fri, 20 Jun 2003 14:27:04 -0500 (CDT)
From: Joe User <joeuser@example.com>
Subject: Test
To: Bob User <bobuser@sample.com>
Message-id: <01KXBK48XVCU8WW4D4example.com>
MIME-version: 1.0
Content-type: TEXT/PLAIN; CHARSET=us-ascii

Test message that will match several bogus rules created specifically
for this test.   http://www.getviagranow.com/
```

Example 9–1 Cont'd on next page

Example 9–1 (Cont.) Example Interactive Run of PMAS

```
# export ENVELOPE_FROM="joeuser@example.com"
# export ENVELOPE_TO="bobuser@sample.com"
#
# precisemail X.X Y.Y
26-JUN-2003 23:54:03.40: Reading in alias file /pmas/data/aliases.txt,  if present....
26-JUN-2003 23:54:03.42: Looking up alias for address <bobuser@sample.com>
26-JUN-2003 23:54:03.42: No alias found
26-JUN-2003 23:54:03.43: Envelope From: joeuser@example.com
26-JUN-2003 23:54:03.43: Envelope To:   bobuser@sample.com (bobuser@sample.com)
26-JUN-2003 23:54:03.45: Reading message file X.X
26-JUN-2003 23:54:03.52:   From:    Joe User <joeuser@example.com>
26-JUN-2003 23:54:03.53:   To:      Bob User <bobuser@sample.com>
26-JUN-2003 23:54:03.53:   Cc:      (no Cc)
26-JUN-2003 23:54:03.55:   Subject: Test
26-JUN-2003 23:54:03.57: No allow file: /pmas/user_rules/BOBUSER.SAMPLE_COM
26-JUN-2003 23:54:03.57: Running Allow checks....
26-JUN-2003 23:54:03.58: Running Block checks....
26-JUN-2003 23:54:03.63: Reading user rule file /pmas/data/00_allowblocklists.cf
26-JUN-2003 23:54:03.64: Running Allow checks....
26-JUN-2003 23:54:03.65: Running Block checks....
26-JUN-2003 23:54:03.65: Reading in rule and score files....
26-JUN-2003 23:54:03.67: Reading rule file /pmas/data/00_local_tests.cf
26-JUN-2003 23:54:03.74: Reading rule file /pmas/data/20_head_tests.cf
26-JUN-2003 23:54:03.80: Reading rule file /pmas/data/20_ratware.cf
26-JUN-2003 23:54:03.83: Reading rule file /pmas/data/20_anti_ratware.cf
26-JUN-2003 23:54:03.86: Reading rule file /pmas/data/20_body_tests.cf
26-JUN-2003 23:54:03.89: Reading rule file /pmas/data/20_compensate.cf
26-JUN-2003 23:54:03.90: Reading rule file /pmas/data/20_html_tests.cf
26-JUN-2003 23:54:03.92: Reading rule file /pmas/data/20_meta_tests.cf
26-JUN-2003 23:54:03.94: Reading rule file /pmas/data/20_uri_tests.cf
26-JUN-2003 23:54:03.95: Reading rule file /pmas/data/20_phrases.cf
26-JUN-2003 23:54:04.00: Reading rule file /pmas/data/20_porn.cf
26-JUN-2003 23:54:04.03: Reading rule file /pmas/data/50_scores.cf
26-JUN-2003 23:54:04.13: Reading rule file /pmas/data/99_local_scores.cf
26-JUN-2003 23:54:04.13: Running the header tests....
26-JUN-2003 23:54:04.14: Header TEST_SUBJECT: Subject: consists only of "Test" (1.000)
26-JUN-2003 23:54:04.16: Header MSGID_HAS_NO_AT: Message-Id has no @ sign (0.100)
26-JUN-2003 23:54:04.18: Header __CT: (0.000)
26-JUN-2003 23:54:04.20: Header __MIME_VERSION: (0.000)
26-JUN-2003 23:54:04.22: Header __CT_TEXT_PLAIN: (0.000)
26-JUN-2003 23:54:04.25: Header __HAS_MSGID: (0.000)
26-JUN-2003 23:54:04.26: Running the URI tests....
26-JUN-2003 23:54:04.27: URI VIAGRA_URI: URI has viagra in it (10.000)
26-JUN-2003 23:54:04.28: Running the body tests....
26-JUN-2003 23:54:04.29: Body BOGUS_RULES: Body has "bogus rules" (2.000)
26-JUN-2003 23:54:04.34: Running the meta tests....
26-JUN-2003 23:54:04.36: Meta INVALID_MSGID: Message-Id is not valid, according to RFC 2822 (0.675)

26-JUN-2003 23:54:04.81: Final score: 13.775

26-JUN-2003 23:54:05.09: Exiting with status 5
$
```

For the PreciseMail Anti-Spam Gateway Pass-Through SMTP Server (PTSMTP), log files that show PTSMTP activity for the worker processes are created in /pmas/log/.

A Files Created During Installation

The files in Table A–1 are created during the installation of the PreciseMail Anti-Spam Gateway software.

Table A–1 PreciseMail Anti-Spam Gateway files created during installation

File name	Description
	Files in /pmas/bin/
authdebug	Executable image for testing GUI authentication.
bastats	Executable image that gathers PMAS statistics.
bayes	Stand-alone Bayesian executable image.
clu_batch	Data Synch Cluster batch job.
dnsblplug.so	DNSBL plugin shareable used by the PTSMTP proxy server.
import_config	Executable image that merges updated pmas_config.dat template information.
libcrypto.*	TLS support shareable.
liblber.*	LDAP support shareable.
libldap.*	LDAP support shareable.
libssl.*	TLS support shareable.
notify_quarantined	Executable image for the quarantine notification job.
pcretest	Executable image for testing regular expressions.
pmas_cluster	Data Synch Cluster daemon.
pmas_master	Executable image for the PreciseMail Anti-Spam Gateway channel.
pmas_milter	Executable image for the PreciseMail Anti-Spam Gatewaymilter.
pmas_process	Executable image for the PreciseMail Processor user interface.
pmas_stats	Executable image to parse the contents of pmas.log and generate a report on PreciseMail activities.
pmas_version	Displays summary information for installed PMAS version.
pmasadmin.so	PMAS administrator command-line utility.
pmasplug.so	PMAS shareable used by the PTSMTP server.
precisemail	Image that can be run from the command line to filter messages.
ptsmtp	Pass-through SMTP proxy server image.
sophplug.so	Sophos shareable used by the PTSMTP server.
tls_certreq	Executable image to generate TLS certificates.

Table A–1 (Cont.) PreciseMail Anti-Spam Gateway files created during installation

File name	Description
	Files in /pmas/com/
cronjobs	Example cron table entries for PreciseMail.
run_nightly.sh	Script responsible for maintaining the log directory.
update.sh	Script run by the autoupdate procedure to unpack and install new rules.
	Files in /pmas/doc/
release_notes.txt	Release notes for PreciseMail Anti-Spam Gateway
pmas_install_guide_*platform*.pdf	PreciseMail Anti-Spam Gateway Installation Guide (Adobe PDF)
pmas_install_guide_*platform*.ps	PreciseMail Anti-Spam Gateway Installation Guide (PostScript)
pmas_install_guide_*platform*.txt	PreciseMail Anti-Spam Gateway Installation Guide (ASCII)
pmas_mgmt_guide_*platform*.pdf	PreciseMail Anti-Spam Gateway Management Guide (Adobe PDF)
pmas_mgmt_guide_*platform*.ps	PreciseMail Anti-Spam Gateway Management Guide (PostScript)
pmas_mgmt_guide_*platform*.txt	PreciseMail Anti-Spam Gateway Management Guide (ASCII)
pmas_users_guide.pdf	PreciseMail Anti-Spam Gateway User's Guide (Adobe PDF)
pmas_users_guide.ps	PreciseMail Anti-Spam Gateway User's Guide (PostScript)
pmas_users_guide.txt	PreciseMail Anti-Spam Gateway User's Guide (ASCII)
	Files in /pmas/html
Various	HTML template files for the PMAS GUI.
	Files in /pmas/www/cgi-bin
adminconfig	Executable image for the Administrative Configuration module.
admingroups	Executable image for the Administrative Groups Configuration module.
adminlicense	Executable image for the Administrative License module.
adminreports	Executable image for the Administrative Reports module.
allowlist	Executable image for the Allow List page.
blocklist	Executable image for the Block List page.
pmaslogin	Executable image for the PMAS Login page.
pmaslogout	Executable image for the PMAS Logout.
pmasprefs	Executable image for the PMAS Preferences page.
pmasstart	Executable image for the PMAS Start page.
quarantine	Executable image for the PMAS Quarantine page.
quarcgi	Executable image for the PMAS Quarantine options.
rulelist	Executable image for the PMAS Rules List page.

Table A–1 (Cont.) PreciseMail Anti-Spam Gateway files created during installation

File name	Description
Files in /pmas/www/htdocs/	
Various	HTML, JavaScript and CSS files for the PMAS GUI.
Files in /pmas/help/	
pmas_process_help.txt	Help file for the user interface.
pmas_process_help.template	HTML template for the help file

Table A–1 (Cont.) PreciseMail Anti-Spam Gateway files created during installation

File name	Description
	Files in /pmas/data/
00_allowblocklists.cf	Local allow and block rules
00_local_tests.cf	Local rules and scores
20_anti_ratware.cf	Rules to try to identify ‘‘legitimate’’ mail clients.
20_body_tests.cf	Rules applied to message bodies.
20_compensate.cf	Rules to compensate for some of the aggressive rules.
20_head_tests.cf	Rules applied to message headers.
20_html_tests.cf	Rules applied to HTML messages.
20_meta_tests.cf	Meta rules made up of header and body meta tests.
20_phrases.cf	Rules for identifying popular spam phrases.
20_porn.cf	Rules for identifying words associated with porn messages.
20_ratware.cf	Rules for identifying messages sent by popular spam software.
20_uri_tests.cf	Rules applied to URIs in the message body.
50_scores.cf	Scores for the rules in the 20_* files.
50_version.cf	Ruleset version.
99_local_scores.cf	Local scores to override the scores in 50_scores.cf.
aliases.txt	Sample aliases file.
internal_ip.txt	List of IP addresses and CIDR blocks for the pass-through proxy to treat as internal systems.
optional_rules.cf	Optional rules that may not be appropriate for all sites.
quarantine_message.template	Template used for user quarantine notification messages.
pmas_config.template	Sample configuration file for PreciseMail.
pmas_confirm_msg.template	Sample template for the confirmation message generated by the PMAS Processor.
pmas_dnsbl.template	Template configuration file for the DNSBL features
pmas_process_reply.template	Sample template for the replies sent by the PMAS processor.
pmas_sophos_config.template	Sample template for Sophos AV engine configuration variables.
ptsmtp.conf	Read-only configuration file for the pass-through proxy server.
ptsmtp_plugins.conf	Secondary configuration file for the PTSMTP controller; generated from PMAS config variables.
ptsmtp_pmas.conf	Secondary configuration files for the pass-through proxy server.
ptsmtp_spf.template	Sample template for PTSMTP SPF configuration.
ptsmtp_sophos.conf	Configuration file that sets debug level for Sophos plugin to Pass-through proxy server.
quarantine_message.template	Template used for user quarantine notification messages.
rdns_exceptions.template	Sample template for rDNS exceptions.
virus_replacement.template	Template for the text that replaces a virus attachment.
vmf_exceptions.template	Sample template for VMF (Verify MAIL FROM) exceptions.

Table A–1 (Cont.) PreciseMail Anti-Spam Gateway files created during installation

File name	Description
Files in /pmas/api/userdb	
example1.c	User database API example program.
example2.c	User database API example program.
example3.c	User database API example program.
example4.c	User database API example program.
example5.c	User database API example program.
example6.c	User database API example program.
makefile	Make file to build the UserDB API example programs.
userdb_api.h	User database API include file.

Glossary

Alias file: Specifies alternate addresses that are to be used when quarantining messages for certain recipients.

Bayesian (learning agent): Uses artificial intelligence to filter spam. Spam and legitimate email is submitted to this agent and it learns the characteristic of both spam message and legitimate email. Because Bayesian filters can be trained, their effectiveness improves over time.

Allow List: A list of addresses from whom all messages should be accepted, regardless of their spam score.

Block List: A list of known spam offenders from whom all incoming email messages will be deleted. For example, if a user constantly receives spam messages from naughty_spammer@example.com, they might wish to place that address on their block list.

Body rule: Applies to the textual parts of the message body. Any non-text MIME parts are stripped, and the text will be decoded from base64 or quoted-printable encoding, but HTML tags and line breaks will be removed before matching.

Corpus: A large collection of mail messages used to test an anti-spam filter's accuracy.

Discarded messages: PreciseMail Anti-Spam Gateway can be configured to discard messages that have a score above a certain discard threshold level. Discarded messages are automatically deleted from the system after the specified number of days has elapsed. The default value is 14 days.

Disposition Code: Determines whether a message will be forwarded, discarded, quarantined, blocked, or allowed.

False negative: A spam message that is incorrectly identified as non-spam by an anti-spam filter.

False positive: A non-spam message that is incorrectly identified as spam by an anti-spam filter.

Full body rule: Applies to the full body of the email message, including text and images.

Ham: Any non-spam email message (an email message that a recipient wishes to receive).

Heuristic (rules) filtering: Tests message header and body against criteria specified by a spam filter.

Meta rule: A boolean test of the results of the other header, body, or Uniform Resource Indicators (URI) rules. Meta rules can be used to produce tests that will only match messages that meet specific criteria.

Quarantined messages: Messages that are identified as spam are quarantined for a defined number of days (14 is the default) until further review by the recipient. Users are automatically notified by email, and can either delete or retrieve quarantined messages.

Raw body rule: Applies to the textual parts of the message body. The text will be decoded from base64 or quoted-printable encoding, but HTML tags and line breaks will still be present.

Regular expression: “regex” A set of symbols and text used to match patterns of text.

Rule scoring: A “score” keyword is used to specify the score for a given rule. There should be a score given for every defined rule. If a rule is not given a score, it is automatically assigned a score of 1.0.

Spam: An unsolicited commercial email sent to multiple email recipients against their wishes.

Spamicity: Based on the sum of scores for all the matching rules, a message’s final score determines its likelihood of being spam, known as its “spamicity”.

Uniform Resource Indicators (URI) Rules: Applied to all of the URIs found in the body of a message. They can be useful in identifying links known to spam web sites embedded in HTML or plain text messages.

Index