MPPv4 Installation and Upgrade Guide

August 2008

ABSTRACT

This document provides guidelines for installing and upgrading MPP and its components.

Message Partners, Inc.
271 North Avenue, Suite 1210
New Rochelle, NY 10801
USA
(877) 302-2027
+1 (914) 712-9050
+1 (914) 206-9609 - Fax
info@messagepartners.com

Table of Contents

1 Introduction

1.1 Overview

MPP is an email security and compliance application that provides a complete solution for email filtering and compliance for service providers, small and medium sized businesses and providers that serve these markets. This guide provides an overview of installing MPP and its system components as well as installing the MPP Virtual Appliance and the MPP MacOS X bundle installation. The MPP configuration guide provides an overview of configuration options and design tips and the MPP Manager guide provides an overview of MPP GUI controls.

1.2 Deployment Scenarios

MPP can be installed on SMTP gateways or directly on your email server. MPP is compatible with Sendmail, Qmail, Postfix, CommuniGate Pro, Exim, Sun Java System Enterprise Server or Surgemail MTA's. MPP is available as a VMWare virtual appliance that can be installed as a scanning gateway. MPP is also compatible with collaboration platforms such as Zimbra, Zarafa and others.

2 Architecture Overview

MPP has three primary components:

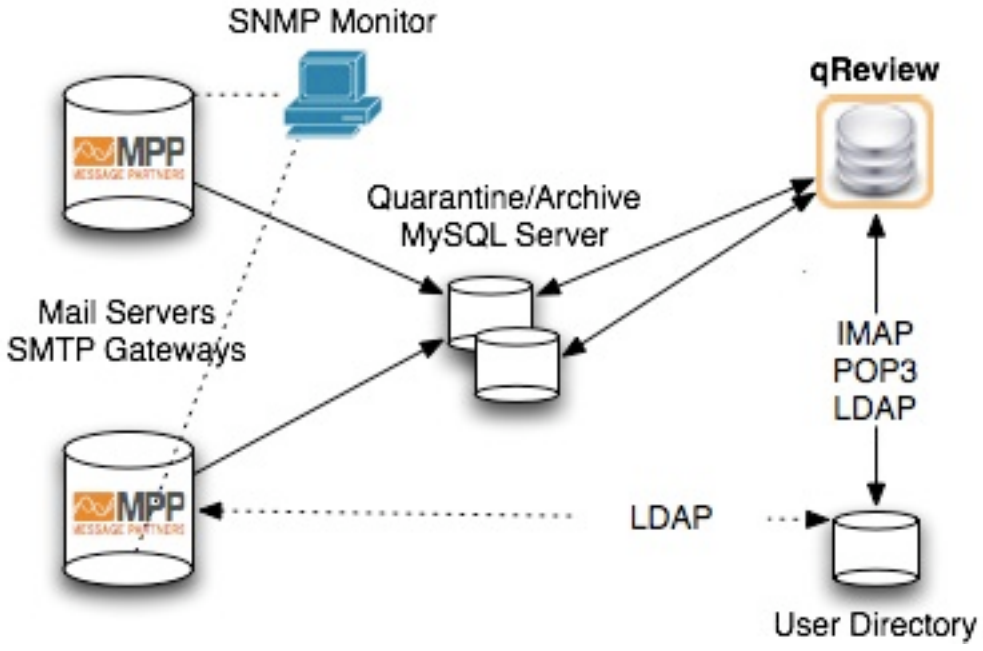
- MPP Core
- Plug-ins
- MPP Manager

2.1 MPP System Architecture

The basic MPP system is made up of a Mail Transfer Agent (Postfix, Sendmail, Qmail, Exim, CGPro, SurgeMail, Zimbra, Sun Java Systems Messaging Server), Webmin and MPP. These are the only required components for a basic implementation of MPP.

When MPP is deployed in conjunction with MySQL and/or LDAP end-users or domain admins can configure features such as per-user spam white/black lists, address validation, quarantine and policy-assignment dynamically with qReview, a component of MPP Manager. QReview allows end-users or domain admins can access their spam quarantines and archives and to set basic settings.

Below is a logical diagram of the MPP system architecture. They are separated for clarity, however, all components may be installed on one or multiple servers depending on the scale of your installation.



3 MPP System Installation

3.1 Linux, Solaris, FreeBSD

This section describes the steps for installing MPP on Linux, FreeBSD or Solaris servers. It is NOT applicable to MacOS X or the MPP Virtual Appliance. If you are upgrading from MPPv3 please skip to the upgrade setion.

3.2 Installing MPP

First download the correct MPP package for your operating system from http://messageapartners.com.

> **1.** This section does not apply to installing the MPP Virtual Appliance or MacOS X bundle installer. To install these products please see subsequent sections

1. Run MPP Installer Package

Linux: rpm –Uvh <package name> or install.sh for non-RedHat platform

Solaris: pkgadd -d <package name>

FreeBSD: pkg_add <package name>

2. Run MPP Configure Script

This script will configure most MTA's (Postfix, QMAIL, Sendmail, CGP) to use MPP, configure MPP to use your selected scanners and run MPP plug-in update scripts.

```
/usr/local/MPP/scripts/configure.pl
```

Manual configuration of MTA's is not generally required, however, subsequent sections in this guide provide detail about modifications required for MTA's to use MPP.

3. Install MPP Manager

Install MPP Manager from /usr/local/MPP/www/mppserver.tar.gz

```
tar –zxvf mppserver.tar.gz
cd mppserver*
./setup.pl
```

MPP Manager requires the XML tool, expat, which is a part of most modern operating systems. In the rare case that you need to install it EXPAT can be installed from sources or using your OS installation tools such as yum or Yast. If this fails for some reason expat can be installed from sources using these steps.

Download from ftp://ftp.messagepartners.com/pub/expat/expat-2.0.1.tar.gz

```
tar –xvf expat-2.0.1.tar.gz
cd expat-2.0.1
./configure
make
make install
```

Most Perl modules required by MPP Manger are included along with MPP Manager and require no other installation. However, in the case that you don't have the required modules Perl modules should be installed from your OS install tools such as yum or as a last resort, via CPAN. – DBD, XML::Parser, HTML::Entities, GD.

4. Confirm the Installation

Logon on to http://yourhost:20001 using the username and password you specified during the installation. If MPP is not running check the specific reason by viewing the logs in status->monitor->MPP Logs

Send a message to mpp@messagepartners.com or support@messagepartners.com if you have issues or need a full trial key that does not mark messages during scanning. Message Partners offers installation services.

3.3 Trial Keys

The MPP trial kit comes with a trial key that is intended to provide you with a temporary method of operating MPP. Since these keys are manually updated with our package it is possible that they can be expired when you begin your trial of MPP.

The trial key will mark headers, messages and warn recipients for viruses. The trial keys are not intended for production use. For production trials the trial key should be replaced with an evaluation key by sending an email to info@messagepartners.com with your contact information, number of users and the scanner plug-ins that you wish to evaluate

3.4 Optional – Configuring MySQL

MPP functionality is enhanced when used in conjunction with MySQL for things like quarantine and archive storage, WBL lists, Access Control Lists and more. More detail is provided in the MPP Configuration Guide about this topic.

Here are basic instructions to configure MPP to use MySQL. All that is required to use MPP with MySQL is to create an empty MySQL database and a user that can access it. The MySQL database can be local or remote.

```
mysql -p -u root
Create database mpp;
Grant all on mpp.* to mpp@localhost identified by 'mpp1233';
```

A single database is fine for evaluation purposes but a production system should have separate databases for spam quarantine, archives and message tracking.

Database setup is detailed in the MPP Configuration guide but once you have created the empty database you can create the MPP tables via the MPP GUI under System->Database->Setup.

MPP SQL tables can be created with MPP GUI, however, command line instructions are provided here for convenience.

```
cat /usr/local/MPP/sql/mpp-mysql.sql | mysql -p -u root -t
databasename
```

*Change database name to the name of your table.

3.4.1 Typical MySQL Errors

If MPP cannot connect to the database after all tables are created and permisions check check /etc/my.cnf file to ensure that the client section has the same socket path as other sections.

```
[mysql.server]
user=mysql
basedir=/var/lib
socket=/var/lib/mysql/mysql.sock
```

```
[client]
socket=/var/lib/mysql/mysql.sock

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

4 Installing MPP on Mac OS X

MPP has a comprehensive installer for Leopard i386 and Tiger PPC that installs all components of MPP and MPP Manager in one step. Simply download the package, run the DMG installer and point your browser to http://localhost:20001 to review settings. All default settings are made for optimal spam filtering, your email server is automatically detected and modified, MPP Manager is installed and all libraries are updated.

NOTE: The default username and password for login is admin/raempp.

The MPP bundle installer is compatible with Postfix, the default email server on Mac OS X, or CommuniGate Pro. If you are using CGP, add mppcgpproxy as a helper after the bundle installation completes.

The MPP bundle installer does not require MySQL, however, if MySQL functionality is required it can easily be configured after the installation is complete.

5 Installing the MPP Virtual Appliance

5.1 Overview

The MPP Virtual Appliance is a pre-configured SMTP proxy that will filter email for virus and spam and archive email. All presets have been made and after a configuration script is run you will be ready for production. Since it is assumed that the appliance will be an SMTP relay be prepared with the name of the domain that it should filter and the IP address of the real email server that it should relay to.

5.2 System Requirements

The MPP Virtual Appliance works with all VMWare virtualization including VMware Player for Windows, VMware Fusion for MacOS X, VMware Server for Windows/Linux or any VMWare enterprise product.

CPU: minimum Intel Pentium 4 2Ghz or equivalen, RAM: minimum minimum 1GB / recommended 2Gb, HDD free space: minimum 30Gb, Networking: Ethernet card (for VMware bridged networking).

If you plan on converting your trial into a production system make sure to allocate enough hard drive space to accommodate your spam quarantine or archive. Message Partners will work with you to establish these guidelines.

5.3 Installing

1. Download the MPP Virtual appliance and add it as a virtual machine.

   For VMware Player open MPP_appliance/vmimage_files/MPPDemo.vmx to start appliance.

   For VMware Server register MPPDemo.vmx as new virtual machine and start VM

2. Run configure script

   ```
   /usr/local/MPP/scripts/mppappconf
   ```

   The configure script will set the root password, timezone config, setup relay domains, setup MPP MySQL quarantine and/or MySQL archive along with new root password. If you make a mistake, don't worry, you can always start over and fix the parts that need correction.

3. To adjust settings go to http://host:20001 to configure

   NOTE: The default username and password for login is admin/raempp.

4. To access spam quarantine and archives go to http://host:20000
5. To place in production point your mx record for your domain to the IP of the virtual appliance.

5.4 Use Appliance Without Relay

It is possible to have the virtual appliance archive email as a standalone product and NOT relay email or filter for spam. Please contact Message Partners support for help with this application.

6 MPP Plug-in Modules

MPP supports plug-in modules to increase functionality.

6.1 Sophos

Sophos is our highest performing scanner due to its capability to scan files in memory and fast updates of signatures. MPP only uses the SAVI library files; we do not require the Sophos Sweeper command line scanner.

Installation of Sophos is automated as part of the MPP installation.

Sophos updates can be configured from MPP Manager, on the front page in the plug-in updates section. Be sure to setup both Sophos monthly and daily updates.

6.2 Kaspersky

Message Partners provides a custom kit for KAV, though we work with standard components for KAV for FileServers or KAV for MailServervers

6.3 Nod32

You must have a valid NOD32 for Linux Mail Server (version 2.50 or higher) license. After installation, simply add '`nod32`' as a `scan_engine` in mppd.conf.xml or via the MPP GUI file and restart MPP. Also note that Nod32 will only work with MPP 2.4 and greater. NOD32 must be obtained from an authorized NOD32 reseller or distributor.

6.4 F-Prot

Install the standard F-Prot for Mail servers. MPP uses the F-Prot Daemon Scanner, f-protd.

6.5 SpamAssassin

MPP interfaces with the spamd component of SpamAssassin.

6.5.1 ClamD

MPP interoperates with libclamav or clamd. Libclamav is included for testing purposes and can be utilized by using the 'clamav' content scanner. Clamd is the preferred method of using MPP with ClamAV.

6.6 Cloudmark

Installation of Cloudmark is automatic, as Cloudmark is statically linked with MPP. MPP uses the Cloudmark Authority SDK, CMAE. Initial configuration of MPP with Cloudmark is automated by the configure.pl script. The key.txt file enables usage of this engine.

Cloudmark uses an internal scheduler for updates. The Cloudmark configuration file placed in `/usr/local/MPP/cloudmark/conf`, may be modified to change the frequency. We also have a tab in our MPP GUI for adjusting the various parameters.

We have a cartridge update script, `/usr/local/MPP/scripts/cloudmarkupdate.sh` that updates the cartridge.so file, located in `/usr/local/MPP/cloudmark/lib`. The script is updated periodically and we announce its availability on the MPP mailing list. You can add the script to run in cron monthly. It will also update the CMAE SDK/API, `/usr/lib/libcmae.so`, when new ones become available.

6.7 Mailshell

Installation of Mailshell is automatic, as Mailshell is statically linked with MPP. Initial configuration of MPP with Mailshell is automated by the `configure.pl` script. The key.txt file enables usage of this engine. The SDK/API resides in `/usr/lib` and is named `libspamcatcher.so.0.0.0`, and needs to be updated periodically. Subscribe to the MPP mailing list to be notified.

Add the `/usr/local/MPP/mailshellupdate.sh` binary to cron to update Mailshell's rules. The configure.pl script should do this for you.

You can edit some options in the `/usr/local/MPP/mailshell/conf/spamcatcher.conf` file. Please read the warnings (in the file) carefully as some options can limit performance. For example, running live RBL queries (option: `rbl_list`) can result in more delay. Also, `enable_spamcompiler_cache=no` could result in using large amounts of memory (>100MB) since the SDK would not be allowed to store compiled rules in an on-disk cache. You are welcome to experiment with options like blocked_charset_list, and note that he SDK extracts the "charset" attribute from the "Content-Type:" header of a message and your mileage may vary based on mail clients.

6.8 Commtouch

The Commtouch engine components are not included in MPP and must be obtained from Message Partners by sending an email to info@messagepartners.com or contacting your representative directly. The Commtouch installation package comes with detailed instructions. Commtouch is available for Linux, Solaris, MacOS X (intel) and FreeBSD.

6.9 McAffee VirusScan (UVScan)

MPP interfaces with McAffee command line scanner using an innovative daemon like interface. This yields extreme performance gains over other implementations of this command line scanner.

6.10 Updating Content Engines

MPP has controls to update Clam, Sophos and Mailshell in the MPP GUI. F-PROT, KAV and Nod32 must be configured for updates via their update scripts. Cloudmark is self-updating for micro-updates; however, it is necessary to run cloudmarkupdate.sh in order to keep the core cartridge up to date. CM will update itself every few minutes, they release cartridge updates about once a quarter and we will announce on the MPP list when these occur.

Make sure to add cron entries for daily and monthly scripts. Cloudmarkupdate.sh and Sophosmonthly.pl should be run monthly or when we announce new updates on our mailing list. Clamavupdate.sh and sophosdaily.sh should be run multiple times a day depending on your requirements. If you run hourly please schedule on random minutes in the hour rather than at 0.

Engine Name	Core Engine Files	Periodic Definition Updates
Cloudmark	Update manually when announced mpp@messagepartners.com via cloudamarkupdate.sh	Automatically, requires no scheduling. Every 5 minutes by default using ftp or http, configure in MPP/cloudmark/etc
Sophos	Update manually every month when announced on mpp@messagepartners.com vi	Schedule MPP/scripts/sophosdaily.sh in crontab to run every 2 hours, on

	sophosmomthly.sh	minutes other than 00
Mailshell	System library file, updated with MPP binaries.	Schedule mailshellupdate to run in cron
F-PROT	Manual update required, generally quarterly. Check the F-PROT site for updates to f-protd	Schedule check-updates.pl in cron
Clamav	Embedded in MPP binaries	Schedule MPP/scripts/clamavupdate.sh in cron
Clamd	Manual updates, or use freshclam	Schedule freshclam in cron
Nod32	Manual update, requires NOD32 for Linux Mailservers	Use standard Nod32 update tools
Commtouch	MP packaged installation kit, ask info@messagepartners.com for kit.	Self-automated, requires no interaction.
McAffee	MPP works with UVscan	

7 MPP Upgrade Instructions

When there is an update to MPP we release 2 versions – a binary only package and a complete install package. If there are no major changes in the configuration files then it is easiest to use the binary packages to update MPP. Simply stop mppd, download the binary package from our ftp server, ftp://ftp.raeinternet.com/pub/mpp2/ , bunzip2 and untar in the MPP directory and restart MPP. If you want to take advantage of new commands it is best to configure them via the MPP GUI and let it create the associated XML commands.

From time-to-time we may release an update that will require extensive changes in the configuration file. In these cases it is best to backup your MPP dir and install the new package..

7.1 Updating from MPPv3 to MPPv4

Upgrading from MPP 3.6 to MPP 4.3.0 can be done using binary only archives for your OS of choice:

OS X 10.5+ / i386:

ftp://ftp.messagepartners.com/pub/mpp4/osx/i386/mppd-4.3.0-rc7-Darwin.gcc40.i386.tar.bz2

OS X 10.4+/ PPC:

ftp://ftp.messagepartners.com/pub/mpp4/osx/ppc/mppd-4.3.0-rc7-Darwin.gcc40.powerpc.tar.bz2

Linux GCC32 / i386:

ftp://ftp.messagepartners.com/pub/mpp4/linux/i386/mppd-4.3.0-rc7-Linux.glibc22.gcc32.i686.tar.bz2

Linux GCC34 / i386:

ftp://ftp.messagepartners.com/pub/mpp4/linux/i386/mppd-4.3.0-rc7-Linux.glibc23.gcc34.i686.tar.bz2

Download any of previous archives for your OS, then follow these steps:

1. **stop MTA and MPP**
2. **unpack archive in /usr/local/MPP**
3. **cd /usr/local/MPP, move binaries archive here**
4. **tar xjvf mppd-4.3.0-1.OS.ARCH.tar**
5. **remove policy_timeout_* options from /usr/local/MPP/mppd.conf.xml**
6. **if MySQL archive /quarantine is in use, please apply the following SQL script to your current MPP DB(s) in use:**

   ftp://ftp.messagepartners.com/pub/mpp4/sql/migrate_4_1_0.sql

```
mysql -hMPP_HOST -uMPP_USER -pMPP_PASS MPP_DB < migrate_4_1_0.sql
```

Mailshell users should upgrade Spamcatcher library and spamcatcher.conf

The file spamcatcher.conf is available for download here:

```
ftp://ftp.messagepartners.com/pub/mailshell/spamcatcher-5.1.0/spamcatcher.conf
```

Library for your OS of choice is available here:

OS X 10.5+ / i386: ftp://ftp.messagepartners.com/pub/mailshell/spamcatcher-5.1.0/osx/i386/libspamcatcher.0.0.0.dylib

OS X 10.4+ / PPC: ftp://ftp.messagepartners.com/pub/mailshell/spamcatcher-5.1.0/osx/ppc/libspamcatcher.0.0.0.dylib

Linux / i386: ftp://ftp.messagepartners.com/pub/mailshell/spamcatcher-5.1.0/linux/i386/libspamcatcher.so.0.0.0

Copy libspamcatcher library in /usr/lib for OS X / Linux and spamcatcher.conf file in /usr/local/MPP/mailshell/conf

7. **Restart mppd and MTA**

7.2 MPP Removal Instructions

Make a backup of the configuration.

Remove all configurations for the mppd from your mail server configuration files and restart your mail server to confirm that the MPP is no longer scanning.

Uninstall the package using the examples below:

For Redhat, rpm -e mpprpmxxx

For Solaris, pkgrm mpprpmxxx

For OS X download and run this command:

ftp://ftp.messagepartners.com/pub/mpp4/osx/Uninstall.command

There is also a remove script that you can use: /usr/local/MPP/uninstall.pl

Remove the log files in /var/log/MPP

Remove the start up scripts, e.g., for Linux: /etc/init.d/mppd, for OS X: /Library/StartupItems/MPP

8 MTA Integration

8.1 Introduction

In most all cases it is not required to manually configure your MTA to use MPP as the configure script will take care of this. However, there are cases such as for Exim or CommuniGate Pro, where manual steps are required. The following section describes the manual configuration process of configuring MPP and MTA's.

MPP supports MTA;s using their native filter interfaces i.e. milter, content filter, etc.. It's usually a simple matter to change some configuration files. First, the email server software has to be reconfigured to use external filters. Second, MPP needs to identify which mail server is being used.

For Postfix, Sendmail, Qmail and CGP configure.pl will automatically configure all mail server settings and it is not necessary to perform these steps manually. This script will automatically configure the correct engine and email_server setting in mppd.conf.xml for all email servers.

You must manually configure MPP as a helper in CommuniGate Pro and configure MTA configuration files with Exim and Sun Java System Messaging Server to work with MPP.

8.2 CommuniGate Pro

MPP requires that CommuniGate Pro to define an "external helper" that points to mppcgpproxy. This communication is via a Unix socket. This configuration change must be made manually using the following steps.

1. Ensure that there is a symbolic link in /var/CommuniGate to /usr/local/MPP/mppcgpproxy. If it is not there, manually create it.

```
# ln -s /usr/local/MPP/mppcgpproxy /var/CommuniGate/mppcgpproxy
```

2. Create a server wide rule:

4. For CGP 5.x: Queue -> Rules -> Create New

Click **SETTINGS** > **RULES**

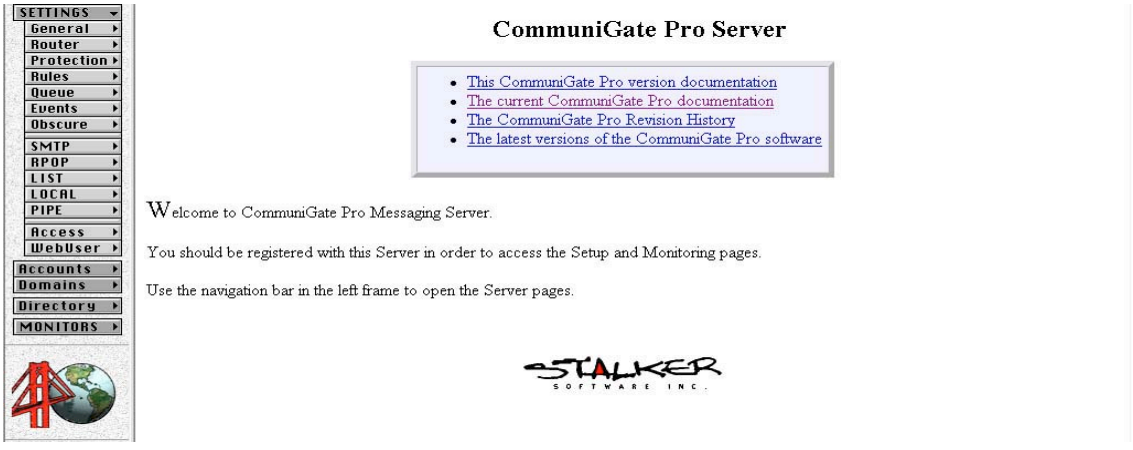


Figure 1 – CommuniGate Splash Screen

1) Click **CREATE NEW**. If there is already an existing rule, you may choose to create a new rule or edit the existing one. Make sure that MPP has the highest priority.

Figure 2 – CGP New Rule Creation

2) Under **DATA**, select **MESSAGE SIZE**

Under Operation, select **GREATER THAN**.

And then put "5" (i.e. 5kb) in the **PARAMETER** box next to **OPERATION**.

Under **ACTION**, select **EXTERNAL FILTER**.

Type "mppcgpproxy" in the **PARAMETERS** input box next to **ACTION**, and then click **UPDATE**.

As of CGP 5.1 there is a new option for asynchronous operation, this should be checked.

Figure 3 – CGP Message Size Parameter

3) Create an External Helper:

Click **SETTINGS** > **GENERAL** > **HELPERS**

4) Check **USE FILTER** by clicking on the select box on the left side.

Type "mppcgpproxy" in the input box next to **USE FILTER**.

Type "mppcgpproxy" in the input box next to **PROGRAM PATH**.

Under **TIME-OUT**, select **DISABLED**

Under **AUTO-RESTART**, select 1 minute and then click **UPDATE**.

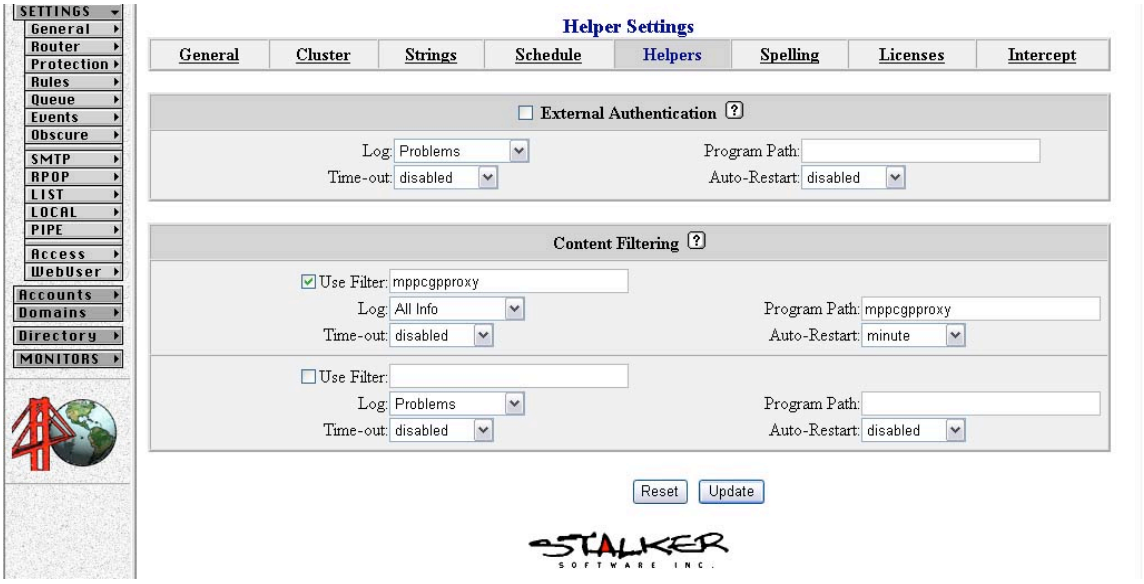


Figure 4 – CGP Content Filtering Configuration

5) If the helper closes output, confirm the following: Make sure mppd is running and configured for email server set to CommuniGate Pro. You can either use Webmin or use command line to do that.

6) If the helper closes output, confirm the following: Make sure mppd is running and configured for email server set to CommuniGate Pro. You can either use Webmin or use command line to do that.

1. Make sure that the configuration file /usr/local/MPP/mppd.conf.xml contains the line, <email_server>cgp</email_server>

2. Or use Webmin's Configure Screen

7) Restart mppd. If it is running use /usr/local/MPP/mppd -r or /usr/local/MPP/mppd -f /usr/local/MPP/mppd.conf.xml to start mppd.

8.3 Sendmail

MPP requires that sendmail defines an input milter that points to the MPP daemon, mppd. This communication is via a Unix socket. Sendmail 8.12 with libmilter is required. This configuration change is made by the configure script for Linux only. Other OS's must follow the procedure below.

8) Add the following to sendmail.mc

```
 INPUT_MAIL_FILTER(`mppd', `S=/var/run/mppd.sock,
F=T,T=S:4m;R:4m;E:5m')
```

Create a new sendmail.cf file using m4

```
# m4 sendmail.mc > /etc/mail/sendmail.cf
```

9) Confirm that the mppd milter definitions are in sendmail.cf:

```
# Input mail filters
O InputMailFilters=mppd
# Milter options
O Milter.macros.connect=j, _, {daemon_name}, {if_name}, {if_addr}
O Milter.macros.helo={tls_version}, {cipher}, {cipher_bits},
{cert_subject},
  {cert_issuer}
O Milter.macros.envfrom=i, {auth_type}, {auth_authen}, {auth_ssf},
{auth_author},
  {mail_mailer}, {mail_host}, {mail_addr}
O Milter.macros.envrcpt={rcpt_mailer}, {rcpt_host}, {rcpt_addr}
```

and

```
Xmppd, S=/var/run/mppd.sock, F=T, T=E:5m
```

10) Confirm that the mppd is running and configured for sendmail: You can either use Webmin or use the command line to do that.

3. Make sure that the configuration file /usr/local/MPP/mppd.conf.xml contains the line, <email_server>sendmail</email_server>

4. Or use Webmin's Configure Screen

11) Restart sendmail

8.3.1 Postfix

MPP works with both the Content Filter interface of Postfix and we also have an integrated Postfix policy server for pre-queue processing of white and black list, rbl's and client-host black lists, spam traps and more . This gives MPP a unique capability to process messages pre and post-queue and to reject messages before the SMTP data transaction where appropriate.

It is HIGHLY recommended to use the MPP Postfix Policy server configuration, however, if you are using Postfix < 2.2 then you must use this configuration, otherwise use the configuration outlined in the next section.

The configure script for MPP or MPP GUI will make all of the configuration settings automatically in MPP and in the Postfix configuration files but explanations of XML commands are provided below.

Configuring MPP for Postfix

The MPP configure script will configure MPP for Postfix automatically. The following commands will be added:

```
<email_server>postfix</email_server>
<email_server_in_protocol>lmtp</email_server_in_protocol>
```

To activate the MPP policy server when the MPP Postfix content filter is also configured add these commands. Of course, the GUI can add these commands as well.

```
<policy_enabled>yes</policy_enabled
<policy_filter_string>mppscan:[127.0.0.1]:10025</policy_filter_str
ing>
```

To activate the MPP Postfix Policy Server in stand alone mode, without the post-queue content filter add this command only:

```
<policy_enabled>yes</policy_enabled
```

There are many commands to customize the policy server that are described below and in the GUI interface.

1) In main.cf add the following:

```
content_filter = mppscan:[localhost]:10025

and add check_policy_service inet:127.0.0.1:9998 to
smtpd_recipient_restrictions  and smtpd_data_restrictions
```

In master.cf add the following

```
 smtp      inet  n       -       n       -       -       smtpd
          -o content_filter=

  # -- Added for MPP --
  localhost:10026 inet    n       -       n       -       10
smtpd
          -o content_filter=
          -o local_recipient_maps=
          -o relay_recipient_maps=
          -o myhostname=localhost.domain.tld
          -o smtpd_helo_restrictions=
          -o smtpd_client_restrictions=
          -o smtpd_sender_restrictions=
          -o smtpd_recipient_restrictions=permit_mynetworks,reject
          -o smtpd_data_restrictions=
          -o smtpd_end_of_data_restrictions=
          -o mynetworks=127.0.0.0/8
          -o smtpd_authorized_xforward_hosts=127.0.0.0/8
  mppscan    unix    -       -       n       -       10      lmtp
          -o lmtp_send_xforward_command=yes
          -o lmtp_cache_connection=no
  # -- end --
```

Requires that postfix defines a “content filter” that points to the MPP daemon, mppd. This communication is via a TCP socket using SMTP or LMTP, LMTP is the preferred transport. Postfix 1.11 or higher is required. This configuration change is made by the configure script.

In main.cf add the following:

```
content_filter = mppscan:[localhost]:10025
```

12) In master.cf add the following:

```
localhost:10026 inet n - n - 10 smtpd
    -o content_filter=
    -o local_recipient_maps=
    -o relay_recipient_maps=
    -o myhostname=localhost.domain.tld
    -o smtpd_helo_restrictions=
    -o smtpd_client_restrictions=
    -o smtpd_sender_restrictions=
    -o smtpd_recipient_restrictions=permit_mynetworks,reject
    -o mynetworks=127.0.0.0/8
   -o smtpd_authorized_xforward_hosts=127.0.0.0/8
scan    unix    -       -       n       -       10      lmtp
     -o lmtp_send_xforward_command=yes
     -o lmtp_cache_connection=no
```

The MPP Postfix Policy Server can run in standalone mode, that is there is no Postfix content filter configured at all. Certain features like per-group RBL's, spamtraps with auto-blacklist capabilities and rate limiting controls make MPP a flexible and powerful policy server. Group logic is still applied to messages, however, in this mode MPP can not take multiple actions for multi-policy email.

Configuration in policy server only mode is the same as with a content filter, however, there is not content filter definition in Postfix configuration files

1) In main.cf add the following:

```
check_policy_service inet:127.0.0.1:9998 to
smtpd_recipient_restrictions  and smtpd_data_restrictions
```

8.4 Qmail

The MPP configuration script will automatically configure qmail for you, however, the steps are outlined below if you prefer to do this manually. MPP will rename your original qmail-queue to qmail-queue.mpp while qmail-queue becomes a symbolic link to /usr/local/MPP/mppqmailproxy

1. Stop Qmail. This can usually be done in the following commands, which are part of the qmail daemon tools:

```
# cd /service

# svc –d qmail-smtpd

# svc –d qmail-send
```

2. Rename the default qmail message queuing program

```
# cd /var/qmail/bin

# mv qmail-queue qmail-queue.mpp
```

3. Check permission of the file by this command:

```
# ls –la /var/qmail/bin/qmail-queue.mpp

The output should be
             -rws—x--x 1 qmailq qmail 20692 Feb 7 15:48 /var/qmail/bin/qmail-
             queue.mpp
```

If it does not have the right ownership and permissions, change them by using:

```
# chmod 4711 /var/qmail/bin/qmail-queue.mpp
# chown qmailq:qmail /var/qmail/bin/qmail-queue.mpp
```

Replace the qmail-queue with the MPP proxy.

```
# ln -s /usr/local/MPP/mppqmailproxy qmail-queue
```

5. If you are upgrading the libraries please make sure that /usr/local/MPP/mppqmailproxy has the following permissions by using:

```
# ls –la /usr/local/MPP/mppgmailproxy

Output should be:
              -rws—x--x 1 qmailq qmail 645528 Feb 3 18:13
              /usr/local/MPP/mppqmailproxy
```

If it does not have the permissions, change the permissions by using the following commands:

```
# chmod 4711 /usr/local/MPP/mppqmailproxy
# chown qmailq:qmail /usr/local/MPP/mppqmailproxy
```

Set softlimit of Qmail in qmail-smtpd startup script to at least 6MB's.The placement of the qmail-smtpd startup script varies with different installations: e.g. for netqmail, the qmail-smtpd is by default in, /var/qmail/supervise/qmail-smtpd/run. Change it to

```
exec /usr/local/bin/softlimit -m 6000000
```

Start Qmail.

8.5 Surgemail

MPP requires that Surgemail defines an external virus filter that points to mppsurgemailproxy. This communication is via a Unix socket. This configuration change is made by the configure script.

13) Insure that there is a symbolic link in /usr/local/surgemail/ to /usr/local/MPP/mppsurgemailproxy. If it is not there, manually create it:

```
# ln -s /usr/local/MPP/mppsurgemailproxy \ /usr/local/surgemail/mppsurgemailproxy
```

14) Set the user and group for mppsurgemailproxy.

```
# chown mail:mail /usr/local/MPP/mppsurgemailproxy
```

15) Make sure that the configuration changes have been made.

5. Make sure that the configuration file /usr/local/MPP/mppd.conf.xml contains the line, <email_server>surgemail</email_server>

6. Or use Webmin's Configure Screen

16) Start mppd.

17) Configure surgemail to use an external filter by making modifying `surgemail.ini`.

The file is located in /etc/ by default. Look for the following line. Change it to say:

```
g_virus_filtercmd="/usr/local/surgemail/mppsurgemailproxy"
type=""
```

18) Restart SurgeMail server

```
# cd /usr/local/surgemail/
```

```
# ./surgemail_stop.sh

# ./surgemail_start.sh
```

8.6 Exim4

Exim 4 support is added in v2.5 and higher.

1) Setup MPP with the configure.pl script, by choosing Exim as the MTA .

2) Setup Exim4. There will be 2 Exim instances using different configuration files:

i.e. /etc/exim/exim.conf and /etc/exim/exim.conf.outgoing

A) To create listenting Exim, edit /etc/exim/exim.conf

```
Listens on the 25/tcp (standard smtp)
daemon_smtp_ports= 25
begin routers
#This has to be the 1st router!!
mpp_router:
driver = manualroute
transport = local_smtp_10025
route_data = 127.0.0.1
self = send

begin transports
local_smtp_10025:
driver = lmtp
gethostbyname
allow_localhost = true
port = 10025
```

B) delivering Exim. Edit /etc/exim/exim.conf.outgoing (created from original exim.conf) is listening on 10026/tcp (because MPP passes the scanned mail to TCP port 10026).

```
daemon_smtp_ports = 10026
local_interfaces = 127.0.0.1
```

Starting Exim:

==========

A) Listening Exim can be started using:

```
/etc/init.d/exim start
or
exim -bd -q1h
```

B) Delivering Exim

```
exim -bd -C /etc/exim/exim.conf.outgoing
```

4) MPP Configuration:

```
<email_server>exim</email_server>
<email_server_in_protocol>lmtp</email_server_in_protocol>
```

8.7 Sun Java System Messaging Server

MPP integrates with the JSMS by the following scenario. When Server receives mail it is submitted for scanning to MPP via SMTP protocol. This is archived by configuring *Submission Channel* and *Submission Rewrite Rules.* MPP scans mail and injects scanned messages back to Server via SMTP protocol. This is archived by configuring *Injection Channel.* Then Server delivers mail to its final destination. MPP also injects forwarded mail and alerts to Server via *Sendmail Command*.

8.7.1 Submission Channel

Submission Channel is a transport channel through which Server submits mail to MPP. It consists from Server's submitting channel (refer to [2] and [3] for detailed description of Servers' channels concept) and MPP's receiving.

To configure *Submission Channel*:

Add to *imta.cnf* configuration file for MTA the following channel definition where $PORT and $HOST must be substituted by location to which submit messages for MPP to accept them. This should be correlated with <email_server_in_socket> XML option in mppd.conf.xml (see further). Note that $HOST can be specified as host name (your.host.com) or as address literal ([127.0.0.1]). By default MPP listens at port 10025 on loopback interface 127.0.0.1.

```
(mandatory blank line)
!
! submits mail to MPP for scanning
tcp_mpp_submit lmtp master port $PORT daemon $HOST
tcp_mpp_submit-daemon
```

In mppd.conf.xml configuration file for MPP specify “sunmail” for <email_server> option. Optionally specify <email_server_in_socket>, <email_server_in_timeout_read>, <email_server_in_timeout_write> and <email_server_in_timeout_connect> (refer [4] for details about this options). The value of <email_server_in_socket> must correlate with $PORT and $HOST values used in previous step. By default MPP listens at port 10025 on loopback interface 127.0.0.1.

8.7.2 Submission Rewrite Rules

Submission Rewrite Rules is Server's rewrite rules (refer to [2] and [3] for details about Server's rewrite rules concept) that route mail to *Submission Channel*. Important requirement for a particular rule is to prevent looping. That is why rules must be source-channel-specific.

For example the following rules can be specified at the top of rewrite rules in default *imta.cnf* configuration file for MTA:

```
! Rules for submitting mail to mppd filter
$* $U%$H$Mtcp_submit@tcp_mpp_submit-daemon
$* $U%$H$Mtcp_local@tcp_mpp_submit-daemon
```

First rule routes mail from standard mail submission channel (SMTP on port 587) to *Submission Channel*. *$Mtcp_submit* parameter ensures that mail only from corresponding channel will be submitted to MPP thus preventing looping and giving the possibility to perform filtering on per-channel basis.

Second rule routes mail from standard SMTP channel (SMTP on port 25) to *Submission Channel*. The role of *$Mtcp_local* parameter is the same as in previous rule

8.7.3 Injection Channel

Injection Channel is transport channel through which MPP injects scanned mail back to Server. It consists of MPP's submitting channel (refer [4] for details about MPP's channel concept) and Server's receiving channel (refer [2] and [3] for details about Server's channel concept).

Configuring Injection Channel

```
Add to imta.cnf configuration file for MTA the following channel
definition

(mandatory blank line)
!
! receives scanned mail injected by MPP
tcp_mpp_inject smtp slave
tcp_mpp_inject-daemon
```

Add to *dispatcher.cnf* configuration file for Dispatcher the following service definition

```
[SERVICE=MPP_INJECT]
PORT=$PORT
IMAGE=IMTA_BIN:tcp_smtp_server
LOGFILE=IMTA_LOG:tcp_smtp_server.log
STACKSIZE=2048000
INTERFACE_ADDRESS=$HOST
PARAMETER=CHANNEL=tcp_mpp_inject
```

where $PORT and $HOST must be substituted by values to which MPP will inject scanned mail. This must correlate with <email_server_out_socket> option in *mppd.conf.xml* configuration file for MPP. By default MPP injects scanned mail to port 10026 on loopback interface 127.0.0.1.

In mppd.conf.xml configuration file for MPP specify “sunmail” for <email_server> option. Optionally specify <email_server_out_socket>, <email_server_out_timeout_read>, <email_server_out_timeout_write> and <email_server_out_timeout_connect> (refer [4] for details about this options). The value of <email_server_out_socket> must correlate with $PORT and $HOST values used in previous step. By default MPP injects scanned mail to port 10026 on loopback interface 127.0.0.1.

8.7.4 Sendmail Command

Sendmail command is a standard UNIX sendmail command line tool to send mail from local system. To use forwarding and alerts functionality of MPP this command should be present and properly configured.

To configure *Sendmail Command*:

Follow steps described in “Handling sendmail clients” described in Messaging Server manual to configure sendmail interface.

Make sure that *sendmail* application is in $PATH location when mppd starts.

8.7.5 MPPD Configuration

Set sunmail as the email server, Set the Email Server Input protocol to LMTP, Restart mppd

9 Installing Sphinx for Full Text Archive Searches

This tuturioal will help you indexing MySQL data created by MPP, by extracting text parts from messages and index them with Sphinx.

We will use the "main+delta" concept to incrementally index the extracted data: once all data from content_index, then only new data every hour.

1. Download content_index.sql and fetchdata.pl

wget -c ftp://ftp.raeinternet.com/pub/mpp3/beta/scripts/content_index.sql

wget -c ftp://ftp.raeinternet.com/pub/mpp3/beta/scripts/fetchdata.pl (change EDITME to real MySQL password)

2. Download sphinx: wget -c http://www.sphinxsearch.com/downloads/sphinx-0.9.7.tar.gz

3. tar xzvf sphinx-0.9.7.tar.gz

4. cd sphinx-0.9.7

5. ./configure --prefix=/usr/local/sphinx ; make ; make install (as root/admin)

6. create /usr/local/sphinx/etc/sphinx.conf with the following content

wget -c ftp://ftp.raeinternet.com/pub/mpp3/beta/scripts/sphinx.conf (change EDITME to real MySQL pass)

7. fetchdata.pl can be configured to parse X entries (my $totalMessages = 10000000;)

8. copy fetchdata.pl in /usr/local/MPP/scripts/ and run it

/usr/local/MPP/scripts/fetchdata.pl

9 Then add it as cronjob:

crontab -e

5 * * * * /usr/local/MPP/scripts/fetchdata.pl >/dev/null 2>&1 </dev/null

10. Create both Spinx indexes: mppindex and mppdeltaindex

/usr/local/sphinx/bin/indexer --config /usr/local/sphinx/etc/sphinx.conf --all

11. Add a cronjob to update mppdeltaindex every hour

45 * * * * /usr/local/sphinx/bin/indexer --config /usr/local/sphinx/etc/sphinx.conf mppdeltaindex --rotate >/dev/null 2>&1 </dev/null

12. To use Perl/PHP API, you would need searchd dameon running

/usr/local/sphinx/bin/searchd --config /usr/local/sphinx/etc/sphinx.conf

Now, we are able to perform full text searches for command line (it will extract messages were there matches too for verification)

/usr/local/sphinx/bin/search -c /usr/local/sphinx/etc/sphinx.conf ovidiu More info can be found here: http://www.sphinxsearch.com/doc.html

NOTE: Once per month disable temporary indexer cronjob and rebuild main index:

/usr/local/sphinx/bin/indexer --config /usr/local/sphinx/etc/sphinx.conf --all --rotate