magic-smtpd User’s Manual

Josh Wilsdon <josh@wizard.ca>

2nd January 2006

Contents

Chapter 1

Introduction

The MagicMail Server is an Internet email transport software package developed by LinuxMagic Inc. With all of the optional features disabled, magic-smtpd acts as a drop in replacement for the qmail-smtpd daemon that is included with Qmail. This means that any system currently running Qmail should be able to install magic-smtpd in place of qmail-smtpd without changing any configuration options. This document describes the installation, configuration and use of the 0.8.4 version of the magic-smtpd program and the supporting tools included with this package.

1.1 Features

The main advantages of using magic-smtpd as a replacement for the SMTP daemon that comes with qmail (qmail-smtpd), are that magic-smtpd has:

1. The ability to check the validity of users before accepting mail destined to them, in order to reduce the number of bounced messages. [Chapter 3]
2. The ability to do SMTP authentication to allow legitimate users with dynamic addresses the ability to relay through your server. [Chapter 4]
3. The ability to limit the rate/amount of email coming in on a given SMTP session in order to discourage spamming and reduce server resource usage. [Section 8.4]
4. The ability to do basic spam checking at the SMTP level. [Chapter 5]
5. Support for encrypted sessions using TLS extensions. [Chapter 6]

The commercial version of magic-smtpd also integrates with magic-local, and other proprietary tools to provide additional features including more advanced spam checking, and virus checking at the delivery level.

Chapter 2

Installation

2.1 Introduction and Prerequisites

Currently, the only supported method for installation is via source code. In order to do a source install you will need the following:

- Linux (see NOTES section later in this Chapter)
- the GNU C compiler + standard C development libraries and headers
- Berkeley/Sleepycat DB development files version 2 or greater (see NOTES section below)
- OpenSSL (if you want to compile in TLS support)
- the magicmail-X.X.X.tar.gz tarball (see NOTES section later in this Chapter, X.X.X will be the version number)
- Qmail, already installed and running with qmail-smtpd running under tcpserver

If you have compiled other software on your machine in the past, and you currently run Qmail, you most likely have the minimum requirements to build this package.

If you are upgrading your magic-smtpd installation from a previous version, see Section 2.4 for information specific to upgrading.

2.2 Build Process

Building the magic-smtpd binary should be mostly straight forward if you have built other software packages in the past. Please read this entire section before beginning. The process should progress as follows (each step is annotated below):

1. tar -zxvf magicmail-<version>.tar.gz
2. cd magicmail-<version>

3. make
4. mkdir -p /etc/magic-mail/control
5. ./magic-smtpd/magic-smtpd -s

Step 1 uses the "tar" command to both uncompress and un-archive the tarball. The "<version>" string should be replaced with the actual version (so that it matches the file you actually have). This will create a directory "magicmail-<version>" in the current directory. Steps 2 and 3 change the working directory to the magic-smtpd source directory and begin the build process respectively. If you receive errors when compiling, you will need to resolve these before continuing. If you need help, feel free to take advantage of the mailing lists (see Section 12.1). Step 4 will create an empty control directory, and step 5 should produce output consisting of the current magic-smtpd configuration if the magic-smtpd binary has been properly built.

2.3 Installation

This section will guide you through the installation of the new binary you have built in Section 2.2. Please read this entire section before beginning. The process should progress as follows (each step is annotated below):

1. cp magic-smtpd/magic-smtpd /var/qmail/bin/magic-smtpd
2. chown root:qmail /var/qmail/bin/magic-smtpd
3. mv -i /var/qmail/bin/qmail-smtpd /var/qmail/bin/qmail-smtpd.old
4. ln -s /var/qmail/bin/magic-smtpd /var/qmail/bin/qmail-smtpd
5. test that your email is working correctly

Steps 1 and 2 place the magic-smtpd binary in the proper location with the proper permissions. If your qmail installation is in a directory other than /var/qmail, you will need to alter the commands accordingly. Steps 3 and 4 are very critical. If they are not performed properly, the SMTP service could be left in a non-functional state. During the time between execution of step 3 and execution of step 4, your SMTP server will be rejecting new connections. Once you have created the link for the magic-smtpd daemon (step 4) you should begin testing your new configuration (this is step 5). To do this, you can use any mail client which connects to your SMTP server, and attempt to send an email. If the mail is successfully sent, you have successfully installed the magic-smtpd daemon. Once you have verified that your configuration is functioning properly, you can begin to configure the additional features provided by magic-smtpd which are described in the rest of this document.

2.4 Upgrading from previous versions

If you have a previous installation of magic-smtpd, you will need to modify your configuration slightly in order to have things work properly with this version. Especially noteworthy with the upgrade to version 0.8.3-rc1 and later from versions older than this. In this version the way the spam rules are loaded has changed. Please review Chapter 5 and make sure your configuration is in a compatible state before you replace your magic-smtpd binary.

2.5 Compiling in additional features

In order to use features which are not compiled in by default (such as TLS and DBFile support), you will need to modify the Makefile.inc file before compilation. There is a configuration section in this file in which you can uncomment (remove the '#' character at the beginning of the line) additional features that you would like to compile in. The most common features you might want to compile in are TLS and DBFile support which can be done by uncommenting the "USE_TLS" and "USE_LM_DBFILE" options respectively. After changing this file you will need to run "make clean" followed by a "make" in order for these changes to take effect. See Chapter 9 for details on determining which support is compiled into your magic-smtpd binary.

2.6 NOTES

- This software may work on other UNIX-like operating systems, but this has not been tested, nor can it be supported at this time. If you need help with this, please refer to the mailing lists.
- You only need the Berkeley DB development files if you are going to be compiling in DBFile support. As this is not currently recommended it is not enabled by default.
- You only need the OpenSSL development files if you are going to be compiling in TLS support. This is not enabled by default.
- The latest tarball can be found on the LinuxMagic website at the URL: http://www.linuxmagic.com/opensource/mag
  smtpd/
- If you are not running qmail-smtpd under tcpserver, you will need to adjust the instructions in this section accordingly.

Chapter 3

Valid-User Checking

3.1 Description

Valid-User Checking is used to reject mail destined to email addresses which do not exist on the server, at the SMTP level. Using this feature decreases the number of messages your server needs to queue for bounces. In many cases this offers a significant savings in terms of server resources. To begin using magic-smtpd for Valid-User Checking, you should look over the next 3 sections and configure one of the 3 methods. Once you have configured the system for your chosen method, refer to Section 3.5 for details on activating your configuration. After activating your configuration you can refer to Section 3.6 for instructions on testing your configuration.

3.2 External Program

This is the currently recommended method for the OpenSource version. To use this method, you will need an executable program or script with the following characteristics:

- takes an email address in user@domain format as its only command line argument
- exits with value 0 (user exists) or 1 (user does not exist)

An example of such a program which works with the vpopmail set of tools, is included in the scripts/ directory of the OpenSource magicmail package.

3.3 Berkeley DB file

When a server is configured to use this method, all spam rules and user information are stored in Berkeley DB database files. This allows very fast access with a very small I/O load. The format of the DB file is documented in the file doc/builddb-KEY_FORMAT.

The builddb tool (see Chapter 10) can be used to build this file if an appropriate module is available. To use this method, support must be compiled into your binary, and you must have the “user_info_dbfile” and “spam_rule_dbfile” control files set with the location of your DB files. You must also set the control file “use_dbfile” to a boolean true value.

NOTE: As of this writing there are no builddb modules available in the OpenSource package. This means that unless you either build the dbfile manually or write your own module, you will be unable to use the dbfile method at this time.

3.4 Magicmail Database

This method is only available in the commercial version of MagicMail. With this method, the magic-smtpd daemon gets all spam and user information from the MagicMail database.

3.5 Enabling Valid-User Checking

To enable valid user checking you must first have configured one of the mechanisms detailed in the previous 3 sections. After configuring the appropriate method, you must activate this feature by specifying a boolean true value in the control file "check_valid_users" (see Section 8.1 for details on using control files).

3.6 Testing Valid-User Checking

In order to verify that Valid-User checking is working, you will need to have 2 email addresses handy: one which you know is an existing user and one that you know is not. In the following examples the addresses “validuser@wizard.ca” and “invaliduser@wizard.ca” will be used for the valid and invalid addresses respectively. Once you have these email addresses you can test as follows (the *italicized* portions indicate your input):

```
./magic-smtpd
220 wizard.ca ESMTP
MAIL FROM: josh@wizard.ca
250 ok
RCPT TO: invaliduser@wizard.ca
550 User does not exist
QUIT
221 wizard.ca
```

this was a test with a user known to not be valid. If you receive a “250 ok” message rather than the “550 User does not exist” message in response to your “RCPT TO:” command, the system thinks that the user does exist. If this works properly, you should also test with a known-valid user:

```
./magic-smtpd
220 wizard.ca ESMTP
MAIL FROM: josh@wizard.ca
250 ok
RCPT TO: validuser@wizard.ca
250 ok
QUIT
221 wizard.ca
```

If you receive a “550 User does not exist” rather than the “250 ok” in response to your “RCPT TO:” command, this means that your configuration is incorrect. If it does not work for you, please recheck your configuration and try again. If it works properly and you get the proper response to both of these test cases: congratulations, you have successfully configured Valid-User checking.

Chapter 4

SMTP Authentication

4.1 Introduction

The magic-smtpd daemon supports SMTP Authentication (As described in RFC 2554). After authenticating themselves, users are allowed to relay mail to remote servers. This allows clients that have dynamic addresses to use the server running magic-smtpd as their outgoing mail-server without much work on the part of the administrator. The authentication can be done either against an external program, against a Berkeley DB file or (in the commercial version only) against the MagicMail database. Sections 4.3, 4.4 and 4.5 give more information on configuring each of these authentication types.

4.2 Support AUTH types

Currently magic-smtpd only supports the “LOGIN” AUTH type. This is the type that is used by Microsoft’s Outlook and Outlook Express and the most commonly used by mail clients. In the future we plan to add support for at least the “PLAIN” and “CRAM-MD5” types. We may add support for more types if there is sufficient demand and time permits.

4.3 External Program

This is the currently recommended method for the OpenSource version. In order to use this method for SMTP authentication you must specify the external program to use for authentication (including the absolute path) in the control file “ext_check_passwd_prog”. This program must be compatible with Dan Bernstein’s checkpassword program interface. More information on this interface is available at the following URL:

http://cr.yp.to/checkpwd/interface.html

Being that the program only needs to be compatible with the checkpassword program, almost any program which is being used with an existing Qmail installation for POP3 authentication (assuming you are using the included qmail-popup and qmail-pop3d programs) can be used directly for SMTP authentication, including the vchkpw program included in the vpopmail package.

4.4 Berkeley DB file

With this method, SMTP authentication will be done against a Berkeley DB file. To use this method, you must set the boolean control option "use_dbfile" to a boolean "true" value. With this method usernames and encrypted passwords are stored in a Berkeley DB file. When a user attempts authentication, magic-smtpd will access the file and compare the password the user has sent with their authentication request with the password in the file. The file used for this authentication can be specified using the "user_info_dbfile" control file. The format of the records read from this file must match that generated by the builddb program which is included in the source distribution.

NOTES:

- Berkeley DB versions 2, 3 and 4 are supported.
- If an external program is also specified for authentication, the external program will be run first and the Berkeley DB file will only be used if the specified program has an error.
- In the commercial version, if the use_database option is selected, the database will also take precedence over the Berkeley DB file and the external program. Processing will only continue to the next method if there is an error reading from the MagicMail database.
- As of this writing there are no builddb modules available in the OpenSource package. This means that unless you either build the dbfile manually or write your own module, you will be unable to use the dbfile method at this time.

4.5 MagicMail Database

This method is only available in the commercial version of MagicMail. With this method, the magic-smtpd daemon does user authentication against the MagicMail database.

4.6 Enabling SMTP Authentication

Once you have selected one of the authentication backend types from the preceding 3 sections, you will still need to activate SMTP authentication before it can be used on your server. In order to do this, you must set a boolean true value in the control file "auth_enable". After doing this, you can test that SMTP authentication is working by following the instructions in the next section.

4.7 Testing SMTP Authentication

In order to test SMTP Authentication, you will need to know the base64 encoded value of a valid username and password for your mailserver. To find these, you can run the following commands on the command-line (after replacing the strings *username* and *password* with the actual username and password):

```
perl -e "use MIME::Base64; print encode_base64(’username’)"
perl -e "use MIME::Base64; print encode_base64(’password’)"
```

The resulting strings were: *dXNlcm5hbWU=* and *cGFzc3dvcmQ=* in this example. After you have the base64 encoded strings, you can make a telnet connection to the localhost (these commands should be run on the machine running magic-smtpd) and do a test of the SMTP Authentication feature (the italicized portions *indicate* your input):

```
telnet localhost 25
Trying 127.0.0.1....
Connected to localhost.
Escape character is ’^]’.
220 wizard.ca ESMTP
EHLO wizard.ca
250-wizard.ca
250-AUTH LOGIN
250-AUTH=LOGIN
250-PIPELINING
250 8BITMIME
AUTH LOGIN dXNlcm5hbWU=
334 UGFzc3dvcmQ6
cGFzc3dvcmQ=
235 ok, go ahead (#2.0.0)
```

If you do not see the “AUTH LOGIN” lines as part of the response to your “EHLO” command, this means that the SMTP Authentication has not been properly enabled. Please double check your settings and try again. The “AUTH LOGIN” command’s argument is the base64 encoded username. The subsequent “334 UGFzc3dvcmQ6” prompt actually says “334 Password:” as this string is base64 encoded, so your response should be the base64 encoded version of your password. If something is misconfigured, or your password or username are incorrect you will instead see either “454 problems performing authorization (#4.3.0)” or “535 authorization failed (#5.7.0)”. If you receive either of these messages, you will need to double check your settings and the username and password you are using. If you receive the “235 ok, go ahead (#2.0.0)” message as above: congratulations, you now have your server configured for SMTP Authentication and should be able to configure your mail client to use this.

Chapter 5

Spam Checking

5.1 How spam rules are loaded

5.1.1 External Program

This is the currently recommended method for the OpenSource version. With version 0.8.3-rc1 the mechanism for loading spam rules using an external program has changed. With this version, in order for spam rules to be loaded properly, the control file “ext_spam_rule_prog” must exist and contain the full path to a valid program. This program must take an email address as it’s sole command line argument, and output to stdout that user’s spam rules. The output format should be:

```
rule\n
value\n
```

with the \n indicating a single newline character. Rules that have multiple values (ie lists) should be output in the following manner:

```
rule\n
list item 1\n
rule\n
list item 2\n
```

such that each value (whether part of a list or single-value rule) results in exactly 2 lines being output to stdout. A sample program is included with this distribution which should suffice as-is or with small modifications for many configurations. More information about this program can be found in Section 11. For help with debugging problems related to spam rules, see Section 5.1.

A list of spam rules that can be configured is included in Appendix A, and a detailed description of each rule is included in Section 5.4.

5.1.2 Berkeley DB File

With this method, spam rules will be loaded from a Berkeley DB file. To use this method, you must set the boolean control option “use_dbfile” to a boolean “true” value. When email comes in the user’s email_id will be looked up from the user_info_dbfile and the rules from that id will be looked up in the dbfile containing the spam rules. The file which contains these spam rules can be specified using the “spam_rule_dbfile” control file. The format of the records read from this file must match that generated by the builddb program which is included in the source distribution.

NOTES:

- Berkeley DB versions 2, 3 and 4 are supported.
- If an external program is also specified for authentication, the external program will be run first and the Berkeley DB file will only be used if the specified program has an error.
- In the commercial version, if the use_database option is selected, the database will also take precedence over the Berkeley DB file and the external program. Processing will only continue to the next method if there is an error reading from the MagicMail database.
- As of this writing there are no builddb modules available in the OpenSource package. This means that unless you either build the dbfile manually or write your own module, you will be unable to use the dbfile method at this time.

5.1.3 PostgreSQL database

This method is only available in the commercial version of MagicMail. With this method, the magic-smtpd daemon loads users’ spam rules from the MagicMail database.

5.2 Global Rules

5.2.1 With DBFile and Database

With either the database or dbfile methods for loading spam rules, there is a special email_id (id 0) which will be used (if they exist) as the “global rules”. These rules can be set by the administrator in the commercial version using the MagicMail web interface. Section 5.4.22 explains some more about including these global rules conditionally for users.

5.2.2 With External Program

When using the external program to load spam rules, the email address “0” is treated specially. The rules attached to this email address are considered the “Global Rules”. This email address can be used to populate global lists which can then be included by another email address (see Section 5.4.22 for details on including global rules for an email address). This is normally managed by a system administrator as a means of setting certain entries on whitelists or blacklists that the administrator either wants everyone to receive (such as mailings from the billing department), or known spam signatures that should be caught by everyone as spam unless they specifically whitelist them.

5.3 Spam Rule Templates

If you use the either the DBFile or Database method for loading spam rules, you can also use spam templates. In order to use these templates, you will first need to have the templates created in the DBFile. The format is detailed in the doc/builddb-KEY_FORMAT file. Basically these templates are a list of rules which a user can include in order to save themselves from maintaining the rules themselves. In order to select a template a user must have the “spam_check” rule set to “true” and the “spam_check_level” rule set to the id of the spam template they wish to use. When a user has a template selected, they can (and should) still manage their own lists (from_whitelist, from_blacklist, etc), and templates should always have the lists enabled.

5.4 SMTP Spam Checking

5.4.1 NOTES

- See Section5.1 for details on enabling the spam rules listed here with your chosen method.
- With the OpenSource version, you will always need to enable to the two rules “spam_check”, “smtp_check” in order to be able to utilize any of the other spam checking options. The “smtp_blocking” rule will always be enabled when using spam checking in this version.
- With the OpenSource version, you must always use the values “true” or “false” for boolean spam controls. Do not use “1” or “0” or any other values.
- Whitelists always take precedence over blacklists and other rules. This means that if a message is on a whitelist, it will not be marked as spam no matter which blacklists or other rules determine it to be spam.

5.4.2 spam_check

This option must be set to “true” in order for any spam checking to be done. If this option is not set, or is set to “false”, none of the other checks will be done.

5.4.3 smtp_check

This enables checks in the magic-smtpd daemon. In the OpenSource version this should always be set to "true" as without this none of the other checks will be done.

5.4.4 smtp_blocking

With the commercial version of MagicMail, when spam is detected at the SMTP level there are two things which can be done:

1. the message can be refused
2. the message can be marked to go into the quarantine

This option is used to select between them. If this is set to "true" the message will be refused if it is detected to be spam. If this option is set to "false" (the default) the message will be marked so that when magic-local receives the message, it will be placed in the quarantine folder.

NOTE: With the OpenSource version this rule will always be set "true" when loading a user's spam rules. When a message matches a spam rule and is not also whitelisted the "RCPT" command will be rejected with a message "550 User does not exist to you".

5.4.5 block_all_mail

As the name suggests, this rule will mark all mail as spam. This is more useful than it appears at first glance. When combined with whitelists this allows a user to block mail from everyone that does not match a whitelist entry (since whitelists take precedence over this and other rules).

5.4.6 valid_from_domain

If this rule is set, the domain of the email address set as the SMTP "MAIL FROM" address will be checked. Thus if the remote uses the command "MAIL FROM: josh@wizard.ca", the domain that will be checked will be wizard.ca. If the domain has neither an A record nor an MX record available via DNS, this rule will mark the message as spam.

5.4.7 check_dynamic_reverse_dns

If this rule is set and tcpserver has set the TCPREMOTEHOST variable, the TCPREMOTEHOST is checked to see if it is a host in the form X-X-X-X.domain.com (where the X's are numbers which could be IP address octets). If the hostname is found to match this pattern, this rule will mark the message as spam.

NOTE: in version 0.8.4 support has been added for using a list of regular expression patterns to detect more complex dynamic reverse DNS entries. A sample file is included in the doc/ directory named "dynamic_dns_regexes". You can copy this file to /etc/magic-mail/control in order to begin using these regular expressions to detect dynamic reverse DNS entries for users who have this rule enabled.

5.4.8 require_full_addr

This rule will mark the message as spam if the "RCPT TO:" recipient address does not include an @ character and at least one character before and after it.

5.4.9 block_mail_from_self

This rule will mark the message as spam if the "RCPT TO:" recipient address is exactly the same as the "MAIL FROM:" sender address.

5.4.10 block_ip_in_addr

This rule will mark the message as spam if the email address in either the "MAIL FROM:" or "RCPT TO:" is in the form user@X.X.X.X where X.X.X.X is an IP Address.

5.4.11 valid_bounce

This rule will mark the message as spam if the magic-smtpd daemon cannot connect to tcp port 25 on a mail exchange (as specified in DNS by an MX record) for the domain of the email address included with the "MAIL FROM:" command.

5.4.12 require_helo

This rule will mark the message as spam if the remote SMTP client does not send either a HELO or an EHLO command at the beginning of their SMTP transaction.

5.4.13 ip_helo_domain

This rule will mark the message as spam if the argument to a HELO command is an IP address in the form "HELO [X.X.X.X]", which is valid according the the RFC standards.

5.4.14 resolve_helo_domain

This rule will attempt to resolve the domain sent as an argument to the HELO command. If it is not able to find an A or MX record for this domain, it will mark the message as spam.

5.4.15 valid_helo_domain

This rule will mark a message as spam if the HELO argument:

- doesn't contain at least one dot
- starts or ends with a dot
- contains two consecutive dots

- contains invalid characters (not alphanumeric, dash, dot, [or])
- has a bare IP address (ie “HELO X.X.X.X”)

5.4.16 mail_from_strict_addr_parse

This rule will mark the message as spam if the address given in the “MAIL FROM:” command is not in the format:

> <user@domain>

where “user” and “domain” are at least 1 character and the <,@ and > characters are all present.

5.4.17 check_ip_reverse_dns

This rule will mark the message as spam if the IP address of the SMTP client that is connecting to the magic-smtpd server does not have a reverse DNS record (a PTR record).

5.4.18 from_blacklist, from_whitelist

The from_whitelist and from_blacklist rules are lists of regular expressions that will be run on the email address passed as an argument to “MAIL FROM:” by the connecting SMTP client. If the address is matched by an entry in the from_whitelist the message will not be marked as spam regardless of whether other rules have marked it as spam or not. If the message is matched by an entry in the from_blacklist it will be marked as spam. As always, the whitelist takes precedence in the case where an entry is both whitelisted and blacklisted. Entries on both lists should be valid extended POSIX regular expressions. An example would be:

> ^j.*@wizard.ca$

which would match “josh@wizard.ca”, “joe@wizard.ca” and any other email address with a name that starts with j and whose domain is wizard.ca. If you use a simple substring like:

> wizard

this would match “josh@wizard.ca” or “wizard@domain.com” or any other address which contains the string “wizard”.

NOTE: These regular expressions can be fairly complicated. Please make sure you know what you are doing if you use more complicated forms.

5.4.19 helo_blacklist, helo_whitelist

The helo_whitelist and helo_blacklist rules are lists of regular expressions that will be run on the hostname passed as an argument to the “HELO” or “EHLO” by the connecting SMTP client. If the hostname is matched by an entry in the helo_whitelist the message will not be marked as spam regardless of whether other rules have marked it as spam or not. If the message is matched by the helo_blacklist it will be marked as spam. As always, the whitelist takes precedence in the case where an entry is both whitelisted and blacklisted. Entries on both lists should be valid extended POSIX regular expressions. An example would be:

```
^m.*wizard.ca$
```

which would match “mail.wizard.ca”, “mx.wizard.ca” and any other address with a name that starts with m and whose domain is wizard.ca. If you use a simple substring like:

```
wizard
```

this would match “mail.wizard.ca” or “wizard.domain.com” or any other hostname which contains the string “wizard”.

NOTE: These regular expressions can be fairly complicated. Please make sure you know what you are doing if you use more complicated forms.

5.4.20 country_blacklist

The country_blacklist define lists of country codes. These country codes must be the 2 letter country codes as defined by IANA. A list of these country codes can be found at:

http://www.iana.org/cctld/cctld-whois.htm

Each entry in this list should consist only of the two letters of the country code. No other characters should be present. The connecting IP address will be checked against a database in order to determine which country the client is connected from. If the country is on the country_blacklist, the message will be marked as spam unless another whitelist rule has already marked it as whitelisted.

NOTE: please see Section5.6 for some special information pertaining to country code based rules.

5.4.21 ip_blacklist, ip_whitelist

The ip_whitelist and ip_blacklist define lists IP addresses which should never or always be treated as sending spam respectively. If a message comes from an IP address on the ip_whitelist, it will never be marked as spam regardless of which other rules determine it to be spam. If the IP address of the connecting client is on the ip_blacklist the message will be marked as spam. The whitelists always take precedence over blacklists, so if a message is both whitelisted and blacklisted it will not be marked as spam.

5.4.22 use_global_*

The rules which begin with a use_global prefix simply state for each global list, whether or not that global list should be used for the user for whom they are set.

5.4.23 spam_check_level

When using either the DBFile or the Database backend, this option can be set to the id of the template to be used for a given email address. The template with that id will then be loaded by the magic-smtpd daemon before doing spam checks for that email address.

5.5 Delivery Level Spam Checking

Spam checks which are done at the delivery level are done by the magic-local program. This program is available only with the commercial version of MagicMail Server.

5.6 Special Considerations for Country Rules

In order to do lookups to match IP addresses to country codes, magic-smtpd utilizes the database that comes with the perl IP::Country module. There are two files included with this module named “ip.gif” and “cc.gif”. The directory in which both of these files exist should be specified in the “ip2country_datadir” MagicMail control file.

Chapter 6

TLS Support

6.1 Introduction

TLS (Transport Layer Security) support can be optionally built into the magic-smtpd binary (see Section 2.2) . With TLS support compiled in, clients which support the STARTTLS extension will be able to establish encrypted SMTP connections to the magic-smtpd daemon. After the initial TLS negotiation, all further data including SMTP Authentication, SMTP conversation and any messages transmitted will be sent via this encrypted channel. The remaining sections in this Chapter describe how to setup and enable TLS support in the magic-smtpd program.

6.2 Creating a certificate

In order to use the TLS extensions with magic-smtpd, you will need an SSL certificate for magic-smtpd to use to identify itself. If you already have a certificate that is being used for something like IMAP-SSL or POP3-SSL, you may be able to use that same certificate with magic-smtpd. If you do not already have a certificate, you will need to generate a new one. You can use the following commands to generate a new (self-signed) certificate for use with magic-smtpd:

```
openssl genrsa 1024 > cert.key
openssl req -new -x509 -nodes -sha1 -days 365 -key cert.key > cert.crt
openssl gendh 1024 > cert.dh
cat cert.crt cert.key cert.dh > /etc/magic-mail/control/cert.pem
```

filling in the appropriate data in the second step (Country, State, Organization, etc), and making sure that you set the “Common Name” field to be the hostname of the server as the clients will see it. If you are only using this certificate for magic-smtpd, you can delete the cert.key, cert.crt and cert.dh files once you have generated the cert.pem file. You should also change the owner of the file to match the user who is running the magic-smtpd daemon, and do a “chmod 400 /etc/magic-mail/control/cert.pem” in order to prevent other users from reading the private key from this file.

If you would like to create a certificate which is signed by a certificate authority such as Verisign or Thawte, you will need to follow their instructions for creating your certificate. Once you have the signed certificate, you should be able to generate a .pem file in much the same fashion as is done above. Due to the wide variety of configurations available, doing this is outside the scope of this document.

6.3 Configuring TLS options

The functioning of the TLS support in the magic-smtpd binary is controlled by the control files which begin with a "tls_" prefix. The default options should work assuming that you have created a self-signed certificate as detailed in the above section. If you are doing something differently from these instructions, refer to Section 8.5 for details on the configuration options available and how they are used.

6.4 Enabling TLS

Once you have generated a certificate and done any other necessary configuration, you are ready to enable TLS support on your server. To enable the TLS extension, you will need to run the following command:

```
echo “1” > /etc/magic-mail/control/tls_enable
```

and verify the configuration using the "magic-smtpd -s" command. Once you have verified that the configuration is complete, you should test to ensure the TLS support is working properly. The next section will give you some pointers with regard to testing this feature. If for some reason this testing fails, you can disable TLS support until you have fixed the configuration by running the command:

```
echo “0” > /etc/magic-mail/control/tls_enable
```

6.5 Testing TLS support

If you have OpenSSL version 0.9.7 or higher you can use the following command to test TLS support:

```
openssl s_client -starttls smtp -ign_eof -crlf -connect 127.0.0.1:25
```

the output should print many lines of text including a certificate and some other debug information before finally giving you a prompt that looks like:

```
220 wizard.ca ESMTP
```

which indicates that it connected successfully. If you receive an error, TLS is not working properly.

If you do not have OpenSSL 0.9.7 or higher, you will need to attempt to connect using a mail client which has been configured for TLS support. Unfortunately configuring mail clients for use as test clients for TLS is beyond the scope of this document.

Upon successfully connecting using TLS you should also see a message in your mail logs saying “CONNECTED using SSL”. If you are unable to get this working, feel free to ask for help on the mailing list (See Section 12.1).

Chapter 7

FAQ - Frequently Asked Questions

7.1 Does magic-smtpd support TLS?

Yes. As of version 0.8.3-rc1 magic-smtpd supports TLS. Please refer to chapter 6 for details on configuring this.

7.2 Does magic-smtpd work on FreeBSD/OpenBSD/Solaris?

Previous versions have been reported to work on FreeBSD with some small patches to the build process. We have developed MagicMail on Linux running on the x86 architecture and do not currently have the resources to test on other operating systems or architectures. In theory it should work fine on any Unix-like system. Hopefully at some point in the future we will have the resources available to test each release on several combinations of operating system and hardware.

7.3 Does magic-smtpd work on 64 bit machines?

Starting with version 0.8.4 we have tested that magic-smtpd compiles and runs on AMD64 machines running Debian Linux (with 64 bit kernel and userland). It may also work on other 64 bit systems, but has not been tested. If you have either positive or negative experiences running it on 64 bit machines, please let us know.

7.4 Can magic-smtpd do virus checking?

No. Virus scanning is supported in the commercial version of MagicMail, but that checking is done at the magic-local (delivery) level, not at the SMTP level.

7.5 Can magic-smtpd block attachments?

No. As magic-smtpd does not scan the body or headers of the message, it has no way to determine the type of attachments, or even whether there are attachments at all.

Chapter 8

Configuration

8.1 Using Control Files

Configuration of the magic-smtpd daemon is done through Qmail style control files. These files exist in the directory specified at compile time which by default is "/etc/magic-mail/control". Each file is named after the option for which it holds values. The content of each file will be treated by magic-smtpd as the value(s) of that option. Each control file is expected to have a specific type of values. These types are described in detail in Section 8.2. You can refer to [Appendix B] for a list of available control files and the type of data they expect. If any of these files do not exist, the default value for that option will be used.

8.2 Control File Types

When magic-smtpd reads its control files, it has a specific type of value that it expects in each control file. You can find out which type is expected in a given control file by looking at the output of the "magic-smtpd -s" command, or by looking at the table in Appendix B. The next sections describe each of these types in detail.

8.2.1 Integer

Control options which expect integer values are expected to have the following properties:

- less than 2147483647
- greater than -2147483647
- contains only the digits 0-9 and optionally prefixed with - or + sign

8.2.2 Program

Control options which are listed as type “program” are expected to have a single executable binary as their value. This program should be specified with an absolute path such as:

```
/var/qmail/bin/qmail-queue
```

8.2.3 Boolean

Control options which are listed as type “boolean” are expected to have either a “true” or “false” value. Several different forms are accepted (case insensitive):

true	false
true	false
on	off
yes	no
1	0

8.2.4 String

Control files listed as having the “string” type are expected to have a single line string value. This can be an arbitrary length string but must consist of only one line. It is expected that this line be composed of ASCII characters only. Foreign character sets and Unicode are not supported at this time.

8.2.5 Directory

Control options which are listed as type "directory" are expected to have a single directory (readable to the user magic-smtpd runs as) as their value. This directory must exist, and should be specified with an absolute path such as:

```
/var/qmail/queue/
```

NOTE: currently it is required to place a trailing “/” character on the directory specified by this option.

8.2.6 Filename

Control options which are listed as type "filename" are expected to have a single file (readable to the user magic-smtpd runs as) as their value. This file must exist, and should be specified with an absolute path such as:

```
/var/qmail/control/defaultdomain
```

8.3 Logging

Magic-smtpd uses syslog to log information about its execution. The logs will go to the syslog “mail” facility with appropriate priorities. For logs to be written out to disk, a syslog daemon must be running. On most machines this daemon is called syslogd and the configuration file is in /etc/syslogd.conf. The configuration of this daemon is outside the scope of this document, but it must be configured properly before log messages will be received from magic-smtpd. You can adjust what logging information goes to what file by modifying your syslog configuration.

NOTE: In version 0.8.4, we have removed the log_level control file as it was causing confusion. All configuration of logging should now be handled in syslog.

8.4 Limits

This section describes those control files which can be used to put limits on resource usage.

8.4.1 max_hops

This option specifies the maximum number of “Received:” and “Delivered-To:” headers allowed in a message before flagging a message as being in a mail delivery loop. (default is 100)

8.4.2 max_invalid_rcpt

This option specifies the maximum number of “RCPT” commands which contain nonexistent users that will be accepted per connection before printing a 550 message and exiting. If this value is set to “0” there will be no limit. (this the default)

8.4.3 max_line_length

This option specifies the maximum length of a command string or message line that will be read before returning an error message to the sending client. (the default is 1024)

8.4.4 max_rcpt

This option specifies the maximum number of “RCPT” commands allowed per client connection. If this value is set to “0” there will be no limit. (this the default)

8.4.5 max_smtp_cmds

This is the maximum number of SMTP commands that will be processed per individual connection. Exceeding this amount will return a 552 error code to the client and disconnect. If this value is set to “0” there will be no limit. (this the default)

8.4.6 rcpt_delay_at

The number of "RCPT" commands to allow before imposing the RCPT delay penalty for a client. If this value is set to "0" there will be no limit. (this the default)

8.4.7 rcpt_delay_inc

The number of seconds to increment the value of RCPT delay for each "RCPT" command after the rcpt_delay_at threshold is exceeded. If this value is set to "0" there will be no limit. (this the default)

8.4.8 rcpt_delay_max

The maximum number of seconds to raise RCPT delay to. If this value is set to "0" there will be no limit. (this the default)

8.5 Other Controls

This section describes those control files which are not related to Logging (Section 8.3) or Limits (Section 8.4).

8.5.1 auth_enable

This option can be used to enable or disable SMTP Authentication. If this option is set to a boolean true value, SMTP Authentication will be available. See Chapter 4 for more information about SMTP Authentication.

8.5.2 block_list_dir

This option specifies the directory which contains the BMS block lists. These lists can be used with the block_lists spam rule.

NOTE: This is not available in the open source version.

8.5.3 check_valid_users

This option can be used to enable or disable Valid User Checking. If this option is set to a boolean true value, The recipients specified in by the RCPT command will be checked before being accepted. If the user does not exist using your configured Valid User Checking mechanism, the remote server will receive a "550 User does not exist" message. See Chapter 3 for more information on Valid User Checking.

8.5.4 check_valid_from

This option works in the same manner as the check_valid_users option, but checks the MAIL FROM address. If this option is enabled and the domain on the MAIL FROM address is local, the address will be checked against the same valid user functions as the RCPT would be normally. This can be used to prevent people from forging the MAIL FROM address using a non-existent local address.

8.5.5 dbname, dbhost, dbport, dbuser, dbpwd, fallback_db, spam_log_db, spam_table

These options are only used by the commercial version of MagicMail and do not exist in the open source version.

8.5.6 drac_host

This option is only useful in the commercial version of MagicMail. It is used by the pop/imap checkpassword program to determine whether or not to use DRAC for POP-before-SMTP functionality, and if so what DRAC host should be used.

8.5.7 dump_core

If this option is enabled, the server will write a core dump file to /var/cores/magic-smtpd/PID/core if a segmentation violation signal is received (segfault). If you are having problems with magic-smtpd segfaulting, turning this option on is a good idea as these core dumps will be helpful to developers trying to track down your problem.

8.5.8 dynamic_dns_regex_filename

This option specifies a filename that will be checked when the check_dynamic_reverse_dns spam rule is enabled. If this file exists and contains valid regular expressions, the expression will be run on the reverse DNS address of the connecting mail server. If there is a match, the rule will mark the message as spam. If the file doesn't exist the default pattern will be used as discussed in 5.4.7.

8.5.9 ext_check_passwd_prog

When using SMTP Authentication with the External Program method, the program specified by this control file will be used for the authentication. The program must operate with the interface of Dan Bernstein's checkpassword program, and must be executable by the user who is running magic-smtpd. See Section 4.3 for more details on SMTP Authentication with this method.

8.5.10 ext_check_user_prog

When using Valid User Checking with the External Program method, the program specified by this control file will be run for each RCPT recipient in order to determine whether that recipient is valid or not. See Section 3.2 for more details.

8.5.11 ext_spam_rule_prog

When using Spam Checking with the External Program method, the program specified by this control file will be run for each RCPT recipient and the output of this program will be treated as that recipient's spam rules. An example program which provides this functionality named "spamdir" is included with the OpenSource release of magic-smtpd.

8.5.12 ip2country_datadir

This option specifies the directory containing the ip and country code databases from the perl IP::Country module. See Section 5.6 for more details.

8.5.13 qmail_local

This option is used by magic-local to determine the path of qmail's version of qmail-local. It is only used by the commercial version of MagicMail and should be ignored in the OpenSource version.

8.5.14 qmail_queue

This option can be used to specify an alternate qmail-queue binary. This can be especially useful for running programs like qmail-scanner to do virus and spam checking. This option works in quite the same manner as the QMAILQUEUE environment variable does with the QMAILQUEUE patch applied to qmail. magic-smtpd also supports the QMAILQUEUE environment variable for compatibility.

8.5.15 rfc_addr_only

If this option is enabled (it is disabled by default), the magic-smtpd daemon will reject and RCPT commands which do not include the "<" and ">" brackets. This can prevent some spam from even needing to be scanned. It can also prevent legitimate mail from misconfigured or poorly designed software, so one must take care when enabling this option.

8.5.16 stray_newline_detection

By default qmail is very strict about stray newlines in messages, as these are forbidden by RFC 2821. Unfortunately several mailservers and many client applications are not well behaved. This option can be set to "false" in order to allow the magic-smtpd daemon to be less strict about checking for stray newlines. The default is to do the stray newline detection in the same manner as qmail-smtpd.

8.5.17 spam_check_enable

This option turns on or off spam checking. If this option is set to a boolean true value (by default this is disabled), spam rules will be checked on a per user basis in a manner determined by the method that is being used for loading users' rules. See Chapter 5 for more information on spam checking.

8.5.18 spam_log_file

This option is new in version 0.8.4. If specified, and the user running magic-smtpd has access to write to the file, this will be used to record spam hits. The entries will be logged in CSV format. The fields logged are:

- unix timestamp
- hostname
- user
- domain
- sender
- Subject: (always NULL for magic-smtpd)
- spam rule
- additional information about spam hit
- internal score of message
- IP address of remote SMTP
- flag whether global (Y or N)

8.5.19 tls_cadir directory

This option specifies the directory to use for loading CA (Certificate Authority) certificates. This directory should contain the certificates of the CA's you want your machine to trust.

8.5.20 tls_cafile

This option specifies a file which contains one or more CA certificates. The certificates in this file (if it exists) override the certificates in the directory specified by the tls_cadir directory.

8.5.21 tls_certificate

This option specifies the filename of the certificate to present to clients who connect using TLS.

8.5.22 tls_dhparams

This option specifies the filename of a file which contains a set of Diffie-Hellman parameters to use in the SSL session negotiation. If this file does not exist magic-smtpd will generate a new set of parameters each time which can be extremely slow.

8.5.23 tls_enable

This option specifies whether or not to enable the TLS extension. If this is set to a boolean false value the STARTTLS command will not be available. If this is set to a boolean true value (and support is compiled in), the STARTTLS will be available to be used by clients.

8.5.24 tls_keyfile

This option specifies the filename of the private key file to use with the certificate specified by the tls_certificate option.

8.5.25 tls_password

If your private key is encrypted with a password, and you do not want to type the password each time an SMTP connection is received, you will want to populate this file with the unencrypted password for your private key.

8.5.26 use_database

This option is used to enable support for the MagicMail database as a backend for SMTP Authentication, Valid User Checking and loading spam rules. It is only available in the commercial version of MagicMail and should be ignored in the OpenSource version.

8.5.27 use_dbfile

This option is used to enable support for using Berkeley DB files for SMTP Authentication, Valid User Checking and loading of spam rules. If support for Berkeley DB has been compiled in to magic-smtpd, you can set this option to a boolean true value in order to enable this method.

8.5.28 user_info_dbfile, spam_rule_dbfile

When Berkeley DB files are being used for SMTP Authentication, Valid User Checking and/or loading of spam rules, these options can be used to specify the location of the DB files to use. The user_info_dbfile should specify the DB file which contains user information such as directories, passwords and quotas. The spam_rule_dbfile should contain the spam rules. See Chapter 10 for more information on these DB files, and the program which can build them for you.

8.5.29 welcome_message

This option specifies the location of the file which will be linked to all new users directories. It should always be read only and must be on the same partition as the mail files. This option is not used by magic-smtpd and is only used by the commercial version of MagicMail. It should be ignored in the OpenSource version.

8.6 Troubleshooting

8.6.1 Listing Configuration

One of the first things you should do when troubleshooting problems with the magic-smtpd daemon is to run the daemon with the “-s” command line option. This will print out a list of the settings that the daemon is configured to use. If any settings in this list seem incorrect, you should ensure that you have set values for all control files appropriate for the feature which is not functioning correctly.

8.6.2 generating a backtrace

When magic-smtpd is segfaulting, it will be helpful for developers for you to send a backtrace with your bug report. To do this you will need to enable the “dump_core” option and create the /var/cores/magic-smtpd directory. You can do this with the following commands:

```
echo "true" > /etc/magic-mail/control/dump_core
mkdir -p /var/cores/magic-smtpd
chmod 1777 /var/cores/magic-smtpd
```

Then when you get a segfault there should be a directory with a core file in this /var/cores/magic-smtpd directory. If you run:

```
gdb /PATH/TO/magic-smtpd /var/cores/magic-smtpd/DIR/core
```

where you replace /PATH/TO with the actual absolute path of magic-smtpd, and replace DIR with the directory that was created in /var/cores/magic-smtpd, you should get to a gdb prompt. From this prompt if you run the command:

```
backtrace
```

you should get a bunch of output. You can then quit (type the command "quit"). If you send the output of the gdb backtrace, it would be very helpful in debugging your problem.

8.6.3 When all else fails

If you come across an issue that you are unable to resolve your next resort should be to ask for help on the mailing list (Section 12.1). It would be most helpful if you include the output of “magic-smtpd -s” and “magic-smtpd -v” with your problem report, along with a backtrace if magic-smtpd is segfaulting.

Chapter 9

Versions

9.1 How to determine which version you have

In order to determine which version of magic-smtpd you have you can simply run the command “magic-smtpd -v”. This will generate a string such as the following:

```
OpenSource 0.8.3-rc1 (compiled: Jan 11 2005) +EXTPROG +DBFILE +TLS
```

Which will tell you which version you are running (in this case “OpenSource 0.8.3-rc1”), the date it was compiled, and which extensions are enabled (in this case EXTPROG and DBFILE and TLS).

9.2 Differences between Commercial and OpenSource versions

The main differences between the commercial and the OpenSource versions of MagicMail are: the MagicMail database (which isn’t available in the Opensource version), the administration interface, and the available support. For more information about the commercial version please visit:

http://magicmail.linuxmagic.com/

Chapter 10

builddb program

The builddb program can be used to build a database suitable for use with the magic-smtpd daemon. Currently however we do not have an OpenSource module for this program and thus it is not able to build a file except for the commercial version.

Chapter 11

spamdir program

11.1 Introduction

The 0.8.3-rc1 and later versions of magic-smtpd come with a program called "spamdir" which can be used as an "ext_spam_rule_prog" program to load spam rules with in the magic-smtpd daemon. This chapter provides more information about this program.

11.2 How it works

The spamdir program expects all users who have spam rules to have a directory in /etc/magic-mail/spam_rules matching their email address. This directory should contain the spam rules in a similar format to the old control_file format. Global rules are stored in the /etc/magic-mail/spam_rules/0 directory. Each file in the directory for an email address or in the global rules directory is treated as the name of a spam rule. Each line (terminated by a single newline character) is treated as a value for this rule. These directories will not be created automatically.

11.3 Examples

If an email comes in to "josh@wizard.ca" the spamdir program will look in the directory:

> /etc/magic-mail/spam_rules/josh@wizard.ca/

and read all files in that directory as spam rules. If the rules for this email address also specify that the global rules should be loaded (eg. has use_global_from_whitelist set to true), the rules from:

> /etc/magic-mail/spam_rules/0/

will also be loaded if they exist, and any entries in the file:

```
/etc/magic-mail/spam_rules/0/from_whitelist
```

would be used by magic-smtpd to compare with the message to josh@wizard.ca.

In order to set some rules for josh@wizard.ca you could do something like the following:

```
mkdir -p /etc/magic-mail/spam_rules/josh@wizard.ca/
echo “true” > /etc/magic-mail/spam_rules/josh@wizard.ca/spam_check
echo “true” > /etc/magic-mail/spam_rules/josh@wizard.ca/smtp_check
echo “true” > /etc/magic-mail/spam_rules/josh@wizard.ca/smtp_blocking
echo “true” > /etc/magic-mail/spam_rules/josh@wizard.ca/require_helo
echo “^josh@wizard.ca$” > /etc/magic-mail/spam_rules/josh@wizard.ca/from_whitelist
echo “^josh@linuxmagic.com$” >> /etc/magic-mail/spam_rules/josh@wizard.ca/from_whitelist
```

which would enable the spam_check, smtp_check, smtp_blocking and require_helo rules and add two entries to the from_whitelist.

11.4 Testing

Once you have enabled the spamdir program by placing the full path to the binary in the “ext_spam_rules_prog” control file, and you have created spam rules for a user, you should be able to use the spamrules command (this binary is also included with the magic-smtpd package) to list the rules.

```
spamrules -e josh@wizard.ca
```

should list the rules and lists for the josh@wizard.ca user, and the command:

```
spamrules -e 0
```

should list the global rules and lists. This spamrules binary uses the same code to load the spam rules as the magic-smtpd daemon, so if the rules are displayed by the spamrules program, they should be working in the SMTP daemon as well.

Chapter 12

Where to get help

12.1 Mailing Lists

The “magicmail-users” mailing list has been created for discussion of issues related to the Magicmail system. You can subscribe to this list by sending an empty email message to “magicmail-users-subscribe@linuxmagic.com”.

12.2 Commercial Support

If you are unable to resolve your problems on your own, Wizard IT Services can provide support. Support is done on an hourly rate with 15 minute increments. For more details on this support service please contact Wizard IT by telephone at (604) 589-0037 or by email at <sales@wizard.ca>. You can also check out their website at http://www.wizard.ca/.

Appendix A

Spam Rules

Rule Name	Value Type	SMTP
spam_check	boolean	Yes
smtp_blocking	boolean	Yes
valid_from_domain	boolean	Yes
block_non_printable	boolean	No
required_header_list	string	No
subject_whitelist	string	No
subject_blacklist	string	No
header_from_whitelist	string	No
header_from_blacklist	string	No
smtp_check	boolean	Yes
delivery_check	boolean	No
check_dynamic_reverse_dns	boolean	Yes
require_full_addr	boolean	Yes
block_all_mail	boolean	Yes
spam_check_level	string	Yes
block_lists	list	Yes
use_global_block_lists	boolean	Yes
block_mail_from_self	boolean	Yes
block_ip_in_addr	boolean	Yes
require_me_in_dest	boolean	No
valid_bounce	boolean	Yes
require_helo	boolean	Yes
valid_helo_domain	boolean	Yes
ip_helo_domain	boolean	Yes
resolve_helo_domain	boolean	Yes
mail_from_strict_addr_parse	boolean	Yes
check_ip_reverse_dns	boolean	Yes
use_global_from_blacklist	boolean	Yes
use_global_from_whitelist	boolean	Yes
from_blacklist	list	Yes
from_whitelist	list	Yes
helo_blacklist	list	Yes
helo_whitelist	list	Yes
use_global_helo_blacklist	boolean	Yes
use_global_helo_whitelist	boolean	Yes
use_global_country_blacklist	boolean	Yes
country_blacklist	list	Yes
use_global_ip_blacklist	boolean	Yes
use_global_ip_whitelist	boolean	Yes

Appendix B

Configuration Options

Filename	Value Type	Default Value	Relevance
auth_enable	boolean	0	Both
block_list_dir	directory	/var/cache/bms	Commercial
bms_honor_quarantine	boolean	0	Commercial
check_valid_from	boolean	0	Both
check_valid_users	boolean	0	Both
dbname	string		Commercial
dbhost	string		Commercial
dbport	string		Commercial
dbuser	string		Commercial
dbpwd	string		Commercial
defaultdomain	string		Both
drac_host	string		Commercial
dump_core	boolean	0	Both
dynamic_dns_regex_filename	filename	/etc/magic-mail/control/dynamic_dns_regexes	Both
ext_check_passwd_prog	program		OpenSource
ext_check_user_prog	program		OpenSource
ext_spam_rule_prog	program		OpenSource
fallback_db	boolean	1	Commercial
ip2country_datadir	directory	/usr/local/share/perl/5.6.1/IP/Country/Fast	Both
max_hops	integer	100	Both
max_invalid_rcpt	integer	0	Both
max_line_length	integer	1024	Both
max_rcpt	integer	0	Both
max_smtp_cmds	integer	0	Both
qmail_local	program	/var/qmail/bin/qmail-local-real	Commercial
qmail_queue	program	/var/qmail/bin/qmail-queue	Both
rcpt_delay_at	integer	0	Both
rcpt_delay_inc	integer	0	Both
rcpt_delay_max	integer	0	Both
rfc_addr_only	boolean	0	Both
stray_newline_detection	boolean	1	Both
spam_check_enable	boolean	0	Both
spam_log_db	boolean	0	Commercial
spam_log_file	filename		Both
spam_rule_dbfile	filename	/etc/magic-mail/dbfiles/spam.db	Both
spam_table	string		Commercial
tls_cadir	directory	/etc/ssl/certs	Both
tls_cafile	filename		Both