

- 
- HOME
  - DOWNLOAD
  - DOCUMENTATION
  - FAQ
  - SUPPORT
  - MAILING LISTS
  - LINKS

## RECENT RELEASES

### 9/5/2008: Version 4.0.4

Bug fixes only for a compiler warning on 64-bit Linux systems and an error in the logging that was sending all messages to both the standard error and the system logs.

[View the changelog](#)

[View the upgrade instructions](#)

[Download spamdyke 4.0.4](#)

### 8/15/2008: Version 4.0.3

Bug fixes only for an integer parsing error on FreeBSD and incorrect handling of invalid nameservers in `/etc/resolv.conf`.

[Download spamdyke 4.0.3](#)

### 8/6/2008: Version 4.0.2

Bug fixes only for a graylist directory error that resulted in very deep directory structures for empty senders.

[Download spamdyke 4.0.2](#)

## QMAIL LINKS

- [qmail main site](#)
- [QmailToaster](#)
- [Life With Qmail](#)
- [Inter7](#)

## SPAMDYKE DOCUMENTATION

This file is updated with each version of spamdyke to reflect the latest features and behavior. If you need documentation for an older version, each version's README file is included in the download package for that version.

### This document applies to spamdyke version 4.0.4.

- [About spamdyke](#)
- [Support](#)
- [How spamdyke works: When a message is not blocked](#)
- [How spamdyke works: When a message is blocked](#)
- [Usage](#)
- [Configuration Files](#)
- [Configuration Directories](#)
- [Configuration Tests](#)
- [Log Messages](#)
- [SMTP Error Codes](#)
- [Logging All Data](#)
- [Permissions](#)
- [DNS Queries](#)
- [Filter Levels](#)
- [TLS](#)
- [SMTP AUTH](#)
- [Relaying](#)
- [Reverse DNS](#)
- [Blacklists](#)
- [DNS RBLs](#)

- DNS RHSBLs
- Whitelists
- Rejecting Senders and Recipients
- DNS Whitelists
- Whitelisting Senders and Recipients
- Graylisting / Greylisting
- Earlytalkers
- Limiting Numbers of Recipients
- Timeouts
- Extra Utilities

## OTHER DOCUMENTATION

The following additional documents are available:

Installation instructions: INSTALL.txt

Upgrading instructions: UPGRADING.txt

Frequently Asked Questions: FAQ.html

Change log: Changelog.txt

To-do list: TODO.txt

## ABOUT SPAMDYKE

```
help
version
```

spamdyke is a filter for monitoring and intercepting incoming SMTP connections to a qmail server. It acts as a transparent middleman, observing the conversation without interference unless it sees something it should block.

Because it can silently monitor, it can also log mail traffic in several different ways.

spamdyke is ©2008 Sam Clippinger, samc (at) silence (dot) org. It is distributed under the GNU General Public License (version 2 only) from <http://www.spamdyke.org/>

The `--help` command line option will give a brief summary of the available command line options. The `--version` command line option will give just the version and copyright statement.

## SUPPORT

spamdyke support is available from the spamdyke-users mailing list: [www.spamdyke.org/mailman/listinfo/spamdyke-users](http://www.spamdyke.org/mailman/listinfo/spamdyke-users).

The mailing list archives are searchable thanks to mail-archive.com: [www.mail-archive.com/spamdyke-users@spamdyke.org](http://www.mail-archive.com/spamdyke-users@spamdyke.org).

All of the spamdyke documentation and downloadable files are available from the spamdyke website: [www.spamdyke.org](http://www.spamdyke.org).

If all else fails, email the author directly at samc (at) silence (dot) org.

## HOW SPAMDYKE WORKS: WHEN A MESSAGE IS NOT BLOCKED

spamdyke works by acting as a middleman between qmail and the network (in Unix terms, it's a pipe). When no spamdyke filters are triggered and a message is delivered normally, spamdyke silently passes data in both directions. As the SMTP conversation takes place, spamdyke collects a few pieces of data (e.g. the sender and recipient addresses) so they can be logged.

spamdyke does modify the incoming message in one way. The SMTP protocol requires the remote sender to end every line with a two character terminator -- a carriage return and a line feed. Unlike most other mail servers, qmail chooses to strictly enforce this requirement. If a remote sender uses only a line feed to end a line (a typical and easy mistake to make), qmail will reject the message:

```
451 See
http://pobox.com/~djb/docs/smtplf.html.
```

Because qmail's strict enforcement of the protocol tends to cause more problems than it solves, spamdyke silently helps mail clients avoid this error by inserting a carriage return before any bare line feed characters it sees. This doesn't affect the messages, it only allows poorly-written mail clients to send email.

## HOW SPAMDYKE WORKS: WHEN A MESSAGE IS BLOCKED

spamdyke's filters are described in detail below. When one of them is triggered, spamdyke moves in to block the incoming message.

First, it considers the enabled filters and waits until there is no way the client can avoid a rejection. For example, if authentication *could* take place but has not done so, spamdyke will wait to see if the remote sender authenticates. Authenticated or whitelisted connections are never filtered.

Next, once spamdyke is certain the message should be filtered, it cuts the connection between the remote sender and qmail. In the background, spamdyke closes the connection to qmail, so qmail will exit normally, believing the remote sender disconnected.

spamdyke continues sending responses to the remote server, just as qmail would have. Once the remote sender has identified the sender and recipient, spamdyke sends an error code and refuses to accept the message. The remote server disconnects, never knowing that spamdyke hijacked the conversation. spamdyke, meanwhile, uses the sender

and recipient information it gathered to construct its log messages.

## USAGE

spamdyke's behavior is controlled through options given on the command line or in configuration files (or both).

On the command line, long options should be prefixed with two hyphens (`--`). Some options have short versions, which should be prefixed with one hyphen (`-`).

In a configuration file, only the long versions are valid and an equals sign must separate the value from the option.

See Configuration Files for details.

For example, consider the `max-recipients` option, which restricts the maximum number of recipients per message. On the command line, limiting the number of recipients to 5 might look like this:

```
spamdyke --max-recipients 5 ...
```

Or, since its short version has the same meaning, the command line could look like this:

```
spamdyke -a 5 ...
```

In a configuration file, only the long version is valid and an equals sign is required, so the entry would look like this:

```
max-recipients=5
```

If the option's value contains spaces, it should be surrounded by quotes on the command line. For example, consider the `rejection-text-ip-blacklist` option, which changes the error message spamdyke sends if the remote server's IP address is blacklisted. On the command line, changing the message might look like this:

```
spamdyke --rejection-text-ip-blacklist "Go away spammer" ...
```

However, in a configuration file quotes are not allowed, so the entry would look like this:

```
rejection-text-ip-blacklist=Go away spammer
```

After all options are given, spamdyke expects the rest of its command line to contain the `qmail` command. For example:

```
spamdyke -a 5 /var/qmail/bin/qmail-smtpd
```

Sometimes, depending on the options in use, spamdyke's command line parser can become confused. If spamdyke believes the `qmail` command is a parameter to one of its options, you may see the following error message:

```
ERROR: Missing qmail-smtpd command
```

To resolve this, place two hyphens (`--`) between the end of spamdyke's options and the `qmail` command. For example:

```
spamdyke -a 5 -- /var/qmail/bin/qmail-smtpd
```

The following options are only valid on the command line:

Long Version	Short Version	Parameter	Description
<code>config-test</code>			Tests the configuration as much as possible and reports any errors that can be discovered without actually accepting an

			<p>incoming message. Use this option with all other options that are given during normal operation. To check file permissions properly, use the <code>run-as-user</code> option.</p> <p>See Configuration Tests for details.</p>
<code>config-test-smtpauth-password</code>		<code>PASSWORD</code>	<p>While testing the configuration with <code>config-test</code>, run the commands given with <code>smtp-auth-command</code> to test authentication processing. Use <code>PASSWORD</code> as the authentication password. This option has no effect unless <code>config-test</code>, <code>config-test-smtpauth-username</code> and <code>smtp-auth-command</code> are given.</p> <p>If <code>config-test-smtpauth-password</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>config-test-smtpauth-password</code> is not given, spamdyke will not test the authentication command(s).</p> <p>See Configuration Tests for details.</p>
<code>config-test-smtpauth-username</code>		<code>USERNAME</code>	<p>While testing the configuration with <code>config-test</code>, run the commands given with <code>smtp-auth-command</code> to test authentication processing. Use <code>USERNAME</code> as the authentication username. This option has no effect unless <code>config-test</code>, <code>config-test-smtpauth-password</code> and <code>smtp-auth-command</code> are given.</p> <p>If <code>config-test-smtpauth-username</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>config-test-smtpauth-username</code> is not given, spamdyke will not test the authentication command(s).</p> <p>See Configuration Tests for details.</p>
<code>help</code>	<code>h</code>		Displays a summary of spamdyke's options, then exits.
<code>version</code>	<code>v</code>		Displays the spamdyke version and copyright statement, then exits.

The following options are valid on the command line and in configuration files. Some options are not valid in files within configuration directories; those options are noted below.

See Configuration Directories for details.

Long Version	Short Version (command line only)	Parameter	Description
<code>access-file</code>		<code>FILE</code>	<p>Use <code>FILE</code> to determine if the remote host is allowed to connect and/or relay. Only needed when using spamdyke to provide SMTP AUTH for an unpatched qmail installation. Most often, <code>FILE</code> is <code>/etc/tcp.smtp</code>.</p> <p>If <code>access-file</code> is given multiple times, each <code>FILE</code> is scanned (in the given order) until a match is found.</p> <p>If <code>access-file</code> is not given, spamdyke will not search any files for relaying permission.</p> <p><code>access-file</code> is not valid within configuration</p>

			<p>directories.</p> <p>See Relaying for details.</p>
<code>config-dir</code>		<code>DIR</code>	<p>Search the directory structure starting at <code>DIR</code> for configuration files that match the remote server's IP address, the remote server's rDNS name, the sender's email address, the recipient's email address or any combination of the four criteria.</p> <p>If <code>config-dir</code> is given multiple times, each <code>DIR</code> is scanned (in the given order) until a match is found.</p> <p>If <code>config-dir</code> is not given, spamdyke will not scan any directories for configuration files.</p> <p><code>config-dir</code> is not valid within configuration directories.</p> <p>See Configuration Directories for details.</p>
<code>config-dir-search</code>		<code>first, all-ip, all-rdns, all-sender or all-recipient</code>	<p>Search the directory structure given by <code>config-dir</code> using the given search rules.</p> <p>If <code>config-dir-search</code> is given multiple times, the given values are used in combination.</p> <p>If <code>config-dir-search</code> is not given, spamdyke will use a value of <code>first</code>.</p> <p><code>config-dir-search</code> is not valid within configuration directories.</p> <p>See Configuration Directories for details.</p>
<code>config-file</code>	<code>f</code>	<code>FILE</code>	<p>Read additional configuration options from <code>FILE</code> as though they were given on the command line.</p> <p>If <code>config-file</code> is given multiple times, each <code>FILE</code> is read in the given order.</p> <p>If <code>config-file</code> is not given, spamdyke will not read a configuration file.</p> <p>See Configuration Files for details.</p>
<code>connection-timeout-secs</code>	<code>t</code>	<code>SECS</code>	<p>Forcibly disconnect after a total of <code>SECS</code> seconds, regardless of activity. A value of <code>0</code> disables this feature.</p> <p>If <code>connection-timeout-secs</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>connection-timeout-secs</code> is not given, spamdyke will not enforce a connection timeout.</p> <p><code>connection-timeout-secs</code> is not valid within configuration directories.</p> <p>See Timeouts for details.</p>

<code>dns-blacklist-entry</code>	<code>x</code>	<code>DNSRBL</code>	<p>Check the remote server's IP address against the realtime blackhole list <code>DNSRBL</code>. If it is found, the connection is rejected. NOTE: Using more than a few DNS blacklists can cause serious performance problems.</p> <p>If <code>dns-blacklist-entry</code> is given multiple times, spamdyke will check each given <code>DNSRBL</code> for the remote server's IP address.</p> <p>If <code>dns-blacklist-entry</code> and <code>dns-blacklist-file</code> are not given, spamdyke will not check any blackhole lists.</p> <p>See DNS RBLs for details.</p>
<code>dns-blacklist-file</code>		<code>FILE</code>	<p>Check the remote server's IP address against each of the realtime blackhole lists listed in <code>FILE</code>. If it is found on any of the lists, the connection is rejected. NOTE: Using more than a few DNS blacklists can cause serious performance problems.</p> <p>If <code>dns-blacklist-file</code> is given multiple times, spamdyke will check each of the blackhole lists listed in each of the files for the remote server's IP address until a match is found.</p> <p>If <code>dns-blacklist-entry</code> and <code>dns-blacklist-file</code> are not given, spamdyke will not search any files for blackhole lists.</p> <p>See DNS RBLs for details.</p>
<code>dns-level</code>		<code>none, normal or aggressive</code>	<p><code>none</code>: Do not perform any DNS queries. All DNS-based filters will behave as though no response was received from any nameserver.</p> <p><code>normal</code>: Send single DNS queries to one nameserver at a time and wait for responses. This mimics the standard system resolver library's behavior.</p> <p><code>aggressive</code>: Send multiple DNS queries to multiple DNS servers simultaneously to find answers as quickly as possible.</p> <p>If <code>dns-level</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>dns-level</code> is not given, spamdyke will use a value of <code>aggressive</code>.</p> <p><code>dns-level</code> is not valid within configuration directories.</p> <p>See DNS Queries for details.</p>
<code>dns-max-retries-primary</code>		<code>NUM</code>	<p>Query the primary nameserver(s) <code>NUM</code> times before also querying the secondary nameserver(s). If <code>NUM</code> is larger than the value of <code>dns-max-retries-total</code>, the value of <code>dns-max-retries-total</code> is used instead.</p>

			<p>If <code>dns-max-retries-primary</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>dns-max-retries-primary</code> is not given, spamdyke will use a value of <code>1</code>.</p> <p><code>dns-max-retries-primary</code> is not valid within configuration directories.</p> <p>See DNS Queries for details.</p>
<code>dns-max-retries-total</code>		<code>NUM</code>	<p>Send a maximum of <code>NUM</code> queries to any nameserver(s), primary or secondary.</p> <p>If <code>dns-max-retries-total</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>dns-max-retries-total</code> is not given, spamdyke will use a value of <code>3</code>.</p> <p><code>dns-max-retries-total</code> is not valid within configuration directories.</p> <p>See DNS Queries for details.</p>
<code>dns-server-ip</code>		<code>IPADDRESS[:PORT]</code>	<p>Use <code>IPADDRESS</code> as a secondary nameserver. If <code>PORT</code> is given, DNS queries will be send to that port number.</p> <p>If <code>dns-server-ip</code> is given multiple times, each of the given nameservers will be queried.</p> <p>If <code>dns-server-ip</code> and <code>dns-server-ip-primary</code> are not given, spamdyke will read the list of nameservers from <code>/etc/resolv.conf</code>.</p> <p><code>dns-server-ip</code> is not valid within configuration directories.</p> <p>See DNS Queries for details.</p>
<code>dns-server-ip-primary</code>		<code>IPADDRESS[:PORT]</code>	<p>Use <code>IPADDRESS</code> as a primary nameserver. If <code>PORT</code> is given, DNS queries will be send to that port number.</p> <p>If <code>dns-server-ip-primary</code> is given multiple times, each of the given nameservers will be queried before any secondary nameservers are queried.</p> <p>If <code>dns-server-ip</code> and <code>dns-server-ip-primary</code> are not given, spamdyke will read the list of nameservers from <code>/etc/resolv.conf</code>.</p> <p><code>dns-server-ip-primary</code> is not valid within configuration directories.</p> <p>See DNS Queries for details.</p>
<code>dns-timeout-secs</code>		<code>SECS</code>	<p>Do not take more than a total of <code>SECS</code> seconds to perform a DNS query, including all of the</p>

			<p>retries.</p> <p>If <code>dns-timeout-secs</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>dns-timeout-secs</code> is not given, spamdyke will use the value <code>30</code>.</p> <p><code>dns-timeout-secs</code> is not valid within configuration directories.</p> <p>See DNS Queries for details.</p>
<code>dns-whitelist-entry</code>		<code>DNSWHITELIST</code>	<p>Check the remote server's IP address against the DNS whitelist <code>DNSWHITELIST</code> (essentially a DNSRBL that contains whitelisted IPs). If it is found, all filters are bypassed. NOTE: Using more than a few DNS whitelists can cause serious performance problems.</p> <p>If <code>dns-whitelist-entry</code> is given multiple times, spamdyke will check each given <code>DNSWHITELIST</code> for the remote server's IP address.</p> <p>If <code>dns-whitelist-entry</code> and <code>dns-whitelist-file</code> are not given, spamdyke will not check any DNS whitelists.</p> <p>See DNS Whitelists for details.</p>
<code>dns-whitelist-file</code>		<code>FILE</code>	<p>Check the remote server's IP address against each of the DNS whitelists (essentially a DNSRBL that contains whitelisted IPs) listed in <code>FILE</code>. If it is found on any of the lists, all filters are bypassed. NOTE: Using more than a few DNS whitelists can cause serious performance problems.</p> <p>If <code>dns-whitelist-file</code> is given multiple times, spamdyke will check each DNS whitelist listed in each given <code>FILE</code> for the remote server's IP address.</p> <p>If <code>dns-whitelist-entry</code> and <code>dns-whitelist-file</code> are not given, spamdyke will not check any DNS whitelists.</p> <p>See DNS Whitelists for details.</p>
<code>filter-level</code>		<code>allow-all, normal, require-auth or reject-all</code>	<p><code>allow-all</code>: Allow all connections to bypass all filters, effectively whitelisting everything.</p> <p><code>normal</code>: Apply enabled filters according to the options on the command line and in the configuration file(s).</p> <p><code>require-auth</code>: Reject all connections that haven't authenticated using SMTP AUTH.</p> <p><code>reject-all</code>: Reject all connections, regardless of authentication or whitelists.</p> <p>If <code>filter-level</code> is given multiple times,</p>

			<p>spamdyke will use the last value it finds.</p> <p>If <code>filter-level</code> is not given, spamdyke will use a value of <code>normal</code>.</p> <p>See Filter Levels for details.</p>
<code>full-log-dir</code>	<code>L</code>	<code>DIR</code>	<p>Log all SMTP data to files in <code>DIR</code>. This is handy for troubleshooting delivery problems but it is not meant to be used long-term. This option imposes a performance penalty!</p> <p>If <code>full-log-dir</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>full-log-dir</code> is not given, spamdyke will not log all SMTP data.</p> <p><code>full-log-dir</code> is not valid within configuration directories.</p> <p>See Logging All Data for details.</p>
<code>graylist-dir</code>	<code>g</code>	<code>DIR</code>	<p>Search for and create graylist files in directory structures within <code>DIR</code>. This option has no effect unless <code>graylist-level</code> is given.</p> <p>If <code>graylist-dir</code> is given multiple times, spamdyke will search each given directory in the given order for the recipient's domain directory and stop when it finds the first one.</p> <p>If <code>graylist-dir</code> is not given, spamdyke will not graylist connections.</p> <p>See Graylisting / Greylisting for details.</p>
<code>graylist-exception-ip-entry</code>		<code>IPADDRESS</code>	<p>Reverse the behavior of the graylist filter for remote servers whose IP addresses match <code>IPADDRESS</code>. This option has no effect unless <code>graylist-level</code> and <code>graylist-dir</code> are given.</p> <p>If <code>graylist-exception-ip-entry</code> is given multiple times, spamdyke will match the remote server's IP address against given <code>IPADDRESS</code>.</p> <p>If <code>graylist-exception-ip-entry</code>, <code>graylist-exception-ip-file</code>, <code>graylist-exception-rdns-dir</code>, <code>graylist-exception-rdns-entry</code> and <code>graylist-exception-rdns-file</code> are not given, spamdyke will graylist each connection according to the value of <code>graylist-level</code>.</p> <p>See Graylisting / Greylisting for details.</p>
<code>graylist-exception-ip-file</code>		<code>FILE</code>	<p>Reverse the behavior of the graylist filter for remote servers whose IP addresses match entries in <code>FILE</code>. This option provides better performance than <code>graylist-exception-ip-entry</code> for more than a few entries. This option has no effect unless <code>graylist-level</code></p>

			<p>and <code>graylist-dir</code> are given.</p> <p>If <code>graylist-exception-ip-file</code> is given multiple times, spamdyke will match the remote server's IP address against each entry in each given <code>FILE</code>.</p> <p>If <code>graylist-exception-ip-entry</code>, <code>graylist-exception-ip-file</code>, <code>graylist-exception-rdns-dir</code>, <code>graylist-exception-rdns-entry</code> and <code>graylist-exception-rdns-file</code> are not given, spamdyke will graylist each connection according to the value of <code>graylist-level</code>.</p> <p>See Graylisting / Greylisting for details.</p>
<code>graylist-exception-rdns-dir</code>		<code>DIR</code>	<p>Reverse the behavior of the graylist filter for remote servers whose rDNS names match files in <code>DIR</code>. This option provides much better performance than <code>graylist-exception-rdns-file</code> for large numbers of entries. This option has no effect unless <code>graylist-level</code> and <code>graylist-dir</code> are given.</p> <p>If <code>graylist-exception-rdns-dir</code> is given multiple times, spamdyke will search each <code>DIR</code> for files that match the remote server's rDNS name.</p> <p>If <code>graylist-exception-ip-entry</code>, <code>graylist-exception-ip-file</code>, <code>graylist-exception-rdns-dir</code>, <code>graylist-exception-rdns-entry</code> and <code>graylist-exception-rdns-file</code> are not given, spamdyke will graylist each connection according to the value of <code>graylist-level</code>.</p> <p>See Graylisting / Greylisting for details.</p>
<code>graylist-exception-rdns-entry</code>		<code>RDNSNAME</code>	<p>Reverse the behavior of the graylist filter for remote servers whose rDNS names match <code>RDNSNAME</code>. This option has no effect unless <code>graylist-level</code> and <code>graylist-dir</code> are given.</p> <p>If <code>graylist-exception-rdns-entry</code> is given multiple times, spamdyke will match the remote server's rDNS name against each given <code>RDNSNAME</code>.</p> <p>If <code>graylist-exception-ip-entry</code>, <code>graylist-exception-ip-file</code>, <code>graylist-exception-rdns-dir</code>, <code>graylist-exception-rdns-entry</code> and <code>graylist-exception-rdns-file</code> are not given, spamdyke will graylist each connection according to the value of <code>graylist-level</code>.</p> <p>See Graylisting / Greylisting for details.</p>
<code>graylist-exception-</code>		<code>FILE</code>	<p>Reverse the behavior of the graylist filter for remote servers whose rDNS names match</p>

<p><code>rdns-file</code></p>			<p>entries in <code>FILE</code>. This option provides better performance than <code>graylist-exception-rdns-entry</code> for more than a few entries. This option has no effect unless <code>graylist-level</code> and <code>graylist-dir</code> are given.</p> <p>If <code>graylist-exception-rdns-file</code> is given multiple times, spamdyke will match the remote server's rDNS name against each entry in each given <code>FILE</code>.</p> <p>If <code>graylist-exception-ip-entry</code>, <code>graylist-exception-ip-file</code>, <code>graylist-exception-rdns-dir</code>, <code>graylist-exception-rdns-entry</code> and <code>graylist-exception-rdns-file</code> are not given, spamdyke will graylist each connection according to the value of <code>graylist-level</code>.</p> <p>See Graylisting / Greylisting for details.</p>
<p><code>graylist-level</code></p>		<p><code>none</code>, <code>always</code>, <code>always-create-dir</code>, <code>only</code> or <code>only-create-dir</code></p>	<p><code>none</code>: Do not graylist any connections.</p> <p><code>always</code>: Graylist all connections that have an existing recipient domain directory, except those that match one of the options <code>graylist-exception-ip-entry</code>, <code>graylist-exception-ip-file</code>, <code>graylist-exception-rdns-dir</code>, <code>graylist-exception-rdns-entry</code> or <code>graylist-exception-rdns-file</code>. If <code>local-domains-entry</code> or <code>local-domains-file</code> is not given, this value has no effect.</p> <p><code>always-create-dir</code>: Graylist all connections except those that match one of the options <code>graylist-exception-ip-entry</code>, <code>graylist-exception-ip-file</code>, <code>graylist-exception-rdns-dir</code>, <code>graylist-exception-rdns-entry</code> or <code>graylist-exception-rdns-file</code>. If the recipient's domain directory does not exist, create it. If <code>local-domains-entry</code> or <code>local-domains-file</code> is not given, this value has no effect.</p> <p><code>only</code>: Do not graylist any connections unless the recipient's domain directory exists and the connection matches one of the options <code>graylist-exception-ip-entry</code>, <code>graylist-exception-ip-file</code>, <code>graylist-exception-rdns-dir</code>, <code>graylist-exception-rdns-entry</code> or <code>graylist-exception-rdns-file</code>. If <code>local-domains-entry</code> or <code>local-domains-file</code> is not given, this value has no effect.</p> <p><code>only-create-dir</code>: Do not graylist any connections unless it matches one of the options <code>graylist-exception-ip-entry</code>,</p>

			<p><code>graylist-exception-ip-file</code>,  <code>graylist-exception-rdns-dir</code>,  <code>graylist-exception-rdns-entry</code> or  <code>graylist-exception-rdns-file</code>. If the recipient's domain directory does not exist, create it. If <code>local-domains-entry</code> or <code>local-domains-file</code> is not given, this value has no effect.</p> <p>If <code>graylist-level</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>graylist-level</code> is not given, spamdyke will use a value of <code>none</code>.</p> <p>See Graylisting / Greylisting for details.</p>
<code>graylist-max-secs</code>	M	SECS	<p>Invalidate graylist entries after they are <code>SECS</code> seconds old. A value of <code>0</code> prevents graylist entries from ever expiring. Requires <code>graylist-dir</code> and <code>graylist-level</code>.</p> <p>If <code>graylist-max-secs</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>graylist-max-secs</code> is not given, spamdyke will use a value of <code>0</code>.</p> <p>See Graylisting / Greylisting for details.</p>
<code>graylist-min-secs</code>	m	SECS	<p>Require a graylist entry to be present for <code>SECS</code> seconds before allowing incoming mail. A value of <code>0</code> will not require any delay; mail will be accepted in any connection immediately after the initial graylisting. Requires <code>graylist-dir</code> and <code>graylist-level</code>.</p> <p>If <code>graylist-min-secs</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>graylist-min-secs</code> is not given, spamdyke will use a value of <code>0</code>.</p> <p>See Graylisting / Greylisting for details.</p>
<code>greeting-delay-secs</code>	e	SECS	<p>Delay sending the SMTP greeting banner <code>SECS</code> seconds to see if the remote server begins sending data early. If it does, the connection is rejected.</p> <p>If <code>greeting-delay-secs</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>greeting-delay-secs</code> is not given, spamdyke will use a value of <code>0</code>.</p> <p>See Earlytalkers for details.</p>
<code>hostname</code>		NAME	<p>Use <code>NAME</code> as the fully qualified domain name of this host. This value is only used to create an encrypted challenge during SMTP AUTH challenge-response protocols.</p> <p>If <code>hostname</code> is given multiple times, spamdyke</p>

			<p>will use the last value it finds.</p> <p>If <code>hostname</code>, <code>hostname-command</code> and <code>hostname-file</code> are not given, spamdyke will search for the host's name in the environment or will use a default name.</p> <p><code>hostname</code> is not valid within configuration directories.</p> <p>See SMTP AUTH for details.</p>
<code>hostname-command</code>		<code>COMMAND</code>	<p>Read the fully qualified domain name of this host from the output of <code>COMMAND</code>. Most often, this value is <code>/bin/hostname -f</code>. This value is only used to create an encrypted challenge during SMTP AUTH challenge-response protocols. This option is ignored if <code>hostname</code> or <code>hostname-file</code> are given.</p> <p>If <code>hostname-command</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>hostname</code>, <code>hostname-command</code> and <code>hostname-file</code> are not given, spamdyke will search for the host's name in the environment or will use a default name.</p> <p><code>hostname-command</code> is not valid within configuration directories.</p> <p>See SMTP AUTH for details.</p>
<code>hostname-file</code>		<code>FILE</code>	<p>Read the fully qualified domain name of this host from the first line of <code>FILE</code>. This value is only used to create an encrypted challenge during SMTP AUTH challenge-response protocols. This option is ignored if <code>hostname</code> is given.</p> <p>If <code>hostname-file</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>hostname</code>, <code>hostname-command</code> and <code>hostname-file</code> are not given, spamdyke will search for the host's name in the environment or will use a default name.</p> <p><code>hostname-file</code> is not valid within configuration directories.</p> <p>See SMTP AUTH for details.</p>
<code>idle-timeout-secs</code>	<code>T</code>	<code>SECS</code>	<p>Forcibly disconnect after <code>SECS</code> seconds of inactivity. A value of <code>0</code> disables this feature.</p> <p>If <code>idle-timeout-secs</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>idle-timeout-secs</code> is not given, spamdyke will use a value of <code>0</code>.</p> <p><code>idle-timeout-secs</code> is not valid within configuration directories.</p>

			See Timeouts for details.
<code>ip-blacklist-entry</code>		<code>IPADDRESS</code>	<p>Reject the connection if the remote server's IP address matches <code>IPADDRESS</code>.</p> <p>If <code>ip-blacklist-entry</code> is given multiple times, spamdyke will check the remote server's IP address against each given <code>IPADDRESS</code>.</p> <p>If <code>ip-blacklist-entry</code> and <code>ip-blacklist-file</code> are not given, spamdyke will not attempt to match the remote server's IP address against blacklist entries.</p> <p>See Blacklists for details.</p>
<code>ip-blacklist-file</code>	<code>B</code>	<code>FILE</code>	<p>Reject the connection if the remote server's IP address matches an entry in <code>FILE</code>. This option provides better performance than <code>ip-blacklist-entry</code> for more than a few entries.</p> <p>If <code>ip-blacklist-file</code> is given multiple times, spamdyke will check the remote server's IP address against each entry in each given <code>FILE</code>.</p> <p>If <code>ip-blacklist-entry</code> and <code>ip-blacklist-file</code> are not given, spamdyke will not attempt to match the remote server's IP address against blacklist entries.</p> <p>See Blacklists for details.</p>
<code>ip-in-rdns-keyword-blacklist-entry</code>		<code>KEYWORD</code>	<p>Search the remote server's rDNS name for its IP address <b>and</b> <code>KEYWORD</code>. If both are found, reject the connection.</p> <p>If <code>ip-in-rdns-keyword-blacklist-entry</code> is given multiple times, spamdyke will search the remote server's rDNS name for its IP address and each given <code>KEYWORD</code>.</p> <p>If <code>ip-in-rdns-keyword-blacklist-entry</code> and <code>ip-in-rdns-keyword-blacklist-file</code> are not given, spamdyke will not reject connections because the remote server's rDNS name contains its IP address.</p> <p>See Reverse DNS for details.</p>
<code>ip-in-rdns-keyword-blacklist-file</code>	<code>k</code>	<code>FILE</code>	<p>Search the remote server's rDNS name for its IP address <b>and</b> a keyword listed in <code>FILE</code>. If both are found, reject the connection. This option provides better performance than <code>ip-in-rdns-keyword-blacklist-entry</code> for more than a few entries.</p> <p>If <code>ip-in-rdns-keyword-blacklist-file</code> is given multiple times, spamdyke will search the remote server's rDNS name for its IP address and each keyword listed in each given <code>FILE</code>.</p>

			<p>If <code>ip-in-rdns-keyword-blacklist-entry</code> and <code>ip-in-rdns-keyword-blacklist-file</code> are not given, spamdyke will not reject connections because the remote server's rDNS name contains its IP address.</p> <p>See Reverse DNS for details.</p>
<code>ip-in-rdns-keyword-whitelist-entry</code>		KEYWORD	<p>Search the remote server's rDNS name for its IP address <b>and</b> KEYWORD. If both are found, bypass all filters.</p> <p>If <code>ip-in-rdns-keyword-whitelist-entry</code> is given multiple times, spamdyke will search the remote server's rDNS name for its IP address and each given KEYWORD.</p> <p>If <code>ip-in-rdns-keyword-whitelist-entry</code> and <code>ip-in-rdns-keyword-whitelist-file</code> are not given, spamdyke will not bypass all filters because the remote server's rDNS name contains its IP address.</p> <p>See Reverse DNS for details.</p>
<code>ip-in-rdns-keyword-whitelist-file</code>		FILE	<p>Search the remote server's rDNS name for its IP address <b>and</b> a keyword listed in FILE. If both are found, bypass all filters. This option provides better performance than <code>ip-in-rdns-keyword-whitelist-entry</code> for more than a few entries.</p> <p>If <code>ip-in-rdns-keyword-whitelist-file</code> is given multiple times, spamdyke will search the remote server's rDNS name for its IP address and each keyword listed in each given FILE.</p> <p>If <code>ip-in-rdns-keyword-whitelist-entry</code> and <code>ip-in-rdns-keyword-whitelist-file</code> are not given, spamdyke will not bypass all filters because the remote server's rDNS name contains its IP address.</p> <p>See Reverse DNS for details.</p>
<code>ip-whitelist-entry</code>		IPADDRESS	<p>If the remote server's IP address matches IPADDRESS, bypass all filters.</p> <p>If <code>ip-whitelist-entry</code> is given multiple times, spamdyke will check the remote server's IP address against each given IPADDRESS.</p> <p>If <code>ip-whitelist-entry</code> and <code>ip-whitelist-file</code> are not given, spamdyke will not attempt to match the remote server's IP address against whitelist entries.</p> <p>See Whitelists for details.</p>
<code>ip-whitelist-file</code>	W	FILE	<p>If the remote server's IP address matches an entry in FILE, bypass all filters. This option provides better performance than <code>ip-</code></p>

			<p><code>whitelist-entry</code> for more than a few entries.</p> <p>If <code>ip-whitelist-file</code> is given multiple times, spamdyke will check the remote server's IP address against each entry in each given <code>FILE</code>.</p> <p>If <code>ip-whitelist-entry</code> and <code>ip-whitelist-file</code> are not given, spamdyke will not attempt to match the remote server's IP address against whitelist entries.</p> <p>See <a href="#">Whitelists</a> for details.</p>
<code>local-domains-entry</code>		<code>DOMAIN</code>	<p>Treat <code>DOMAIN</code> as a locally hosted domain (to determine if an email address is local or remote).</p> <p>If <code>local-domains-entry</code> is given multiple times, spamdyke will consider each given <code>DOMAIN</code> to be local.</p> <p>If <code>local-domains-entry</code> and <code>local-domains-file</code> are not given, spamdyke will disable all filters that depend on distinguishing between local and remote addresses.</p> <p><code>local-domains-entry</code> is not valid within configuration directories.</p> <p>See <a href="#">Rejecting Senders and Recipients</a> for details.</p>
<code>local-domains-file</code>	<code>d</code>	<code>FILE</code>	<p>Search <code>FILE</code> for a list of locally hosted domains (to determine if an email address is local or remote). Most often, <code>FILE</code> is <code>/var/qmail/control/rcpthosts</code>.</p> <p>If <code>local-domains-file</code> is given multiple times, spamdyke will consider each entry in each given <code>FILE</code> to be local.</p> <p>If <code>local-domains-entry</code> and <code>local-domains-file</code> are not given, spamdyke will disable all filters that depend on distinguishing between local and remote addresses.</p> <p><code>local-domains-file</code> is not valid within configuration directories.</p> <p>See <a href="#">Rejecting Senders and Recipients</a> for details.</p>
<code>log-level</code>	<code>l</code> (lowercase ell)	<code>none</code> , <code>error</code> , <code>info</code> , <code>verbose</code> , <code>debug</code> or <code>excessive</code>	<p><code>none</code>: No logging.</p> <p><code>error</code>: Log errors only.</p> <p><code>info</code>: Everything from <code>error</code> plus connection messages.</p> <p><code>verbose</code>: Everything from <code>info</code> plus non-critical errors such as network errors caused by the remote host, protocol errors, config-test</p>

			<p>status messages and child process error messages.</p> <p><code>debug</code>: Everything from <code>verbose</code> plus high-level debugging messages to show the processing path within spamdyke.</p> <p><code>excessive</code>: Everything from <code>debug</code> plus low-level debugging messages to show data values and small status messages within spamdyke.</p> <p>If <code>log-level</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>log-level</code> is not given, spamdyke will use a value of <code>error</code>.</p> <p><code>log-level</code> is not valid within configuration directories.</p> <p>See Log Messages for details.</p>
<code>log-target</code>		<code>stderr</code> or <code>syslog</code>	<p><code>stderr</code>: Send log messages to standard error (stderr).</p> <p><code>syslog</code>: Send log messages to the system log file via syslogd.</p> <p>If <code>log-target</code> is given multiple times, spamdyke will use a combination of the given values.</p> <p>If <code>log-target</code> is not given, spamdyke will use a value of <code>syslog</code>.</p> <p><code>log-target</code> is not valid within configuration directories.</p> <p>See Log Messages for details.</p>
<code>max-recipients</code>	<code>a</code>	<code>NUM</code>	<p>Allow a maximum of <code>NUM</code> recipients per connection.</p> <p>If <code>max-recipients</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>max-recipients</code> is not given, spamdyke will not limit the number of recipients.</p> <p><code>max-recipients</code> is not valid within configuration directories.</p> <p>See Limiting Numbers of Recipients for details.</p>
<code>policy-url</code>	<code>u</code>	<code>URL</code>	<p>Append <code>URL</code> to the rejection message to explain why the rejection occurred. NOTE: most servers hide rejection messages from their users and most users don't read bounce messages. Maximum 100 characters.</p> <p>If <code>policy-url</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>policy-url</code> is not given, spamdyke will not</p>

			<p>append a URL to the rejection message.</p> <p>See SMTP Error Codes for details.</p>
<code>rdns-blacklist-dir</code>	<code>b</code>	<code>DIR</code>	<p>Reject the connection if the remote server's rDNS name matches a file in <code>DIR</code>. This option provides better performance than <code>rdns-blacklist-file</code> for large numbers of entries.</p> <p>If <code>rdns-blacklist-dir</code> is given multiple times, spamdyke will search each <code>DIR</code> for files that match the remote server's rDNS name.</p> <p>If <code>rdns-blacklist-dir</code>, <code>rdns-blacklist-entry</code> and <code>rdns-blacklist-file</code> are not given, spamdyke will not attempt to match the remote server's rDNS name against blacklist entries.</p> <p>See Blacklists for details.</p>
<code>rdns-blacklist-entry</code>		<code>DOMAIN</code>	<p>Reject the connection if the remote server's rDNS name matches <code>DOMAIN</code>.</p> <p>If <code>rdns-blacklist-entry</code> is given multiple times, spamdyke will attempt to match the remote server's rDNS name against each given <code>DOMAIN</code>.</p> <p>If <code>rdns-blacklist-dir</code>, <code>rdns-blacklist-entry</code> and <code>rdns-blacklist-file</code> are not given, spamdyke will not attempt to match the remote server's rDNS name against blacklist entries.</p> <p>See Blacklists for details.</p>
<code>rdns-blacklist-file</code>		<code>FILE</code>	<p>Reject the connection if the remote server's rDNS name matches an entry in <code>FILE</code>. This option provides better performance than <code>rdns-blacklist-entry</code> for more than a few entries.</p> <p>If <code>rdns-blacklist-file</code> is given multiple times, spamdyke will search each given <code>FILE</code> for entries that match the remote server's rDNS name.</p> <p>If <code>rdns-blacklist-dir</code>, <code>rdns-blacklist-entry</code> and <code>rdns-blacklist-file</code> are not given, spamdyke will not attempt to match the remote server's rDNS name against blacklist entries.</p> <p>See Blacklists for details.</p>
<code>rdns-whitelist-dir</code>		<code>DIR</code>	<p>If the remote server's rDNS name matches a file in <code>DIR</code>, bypass all filters. This option provides better performance than <code>rdns-whitelist-file</code> for large numbers of entries.</p> <p>If <code>rdns-whitelist-dir</code> is given multiple times, spamdyke will search each <code>DIR</code> for files</p>

			<p>that match the remote server's rDNS name.</p> <p>If <code>rdns-whitelist-dir</code>, <code>rdns-whitelist-entry</code> and <code>rdns-whitelist-file</code> are not given, spamdyke will not attempt to match the remote server's rDNS name against whitelist entries.</p> <p>See Whitelists for details.</p>
<code>rdns-whitelist-entry</code>		DOMAIN	<p>If the remote server's rDNS name matches DOMAIN, bypass all filters.</p> <p>If <code>rdns-whitelist-entry</code> is given multiple times, spamdyke will attempt to match the remote server's rDNS name against each given DOMAIN.</p> <p>If <code>rdns-whitelist-dir</code>, <code>rdns-whitelist-entry</code> and <code>rdns-whitelist-file</code> are not given, spamdyke will not attempt to match the remote server's rDNS name against whitelist entries.</p> <p>See Whitelists for details.</p>
<code>rdns-whitelist-file</code>	w	FILE	<p>If the remote server's rDNS name matches an entry in FILE, bypass all filters. This option provides better performance than <code>rdns-whitelist-entry</code> for more than a few entries.</p> <p>If <code>rdns-whitelist-file</code> is given multiple times, spamdyke will search each given FILE for entries that match the remote server's rDNS name.</p> <p>If <code>rdns-whitelist-dir</code>, <code>rdns-whitelist-entry</code> and <code>rdns-whitelist-file</code> are not given, spamdyke will not attempt to match the remote server's rDNS name against whitelist entries.</p> <p>See Whitelists for details.</p>
<code>recipient-blacklist-entry</code>		ADDRESS	<p>Reject any recipient addresses that match ADDRESS.</p> <p>If <code>recipient-blacklist-entry</code> is given multiple times, spamdyke will attempt to match each recipient address against each given ADDRESS.</p> <p>If <code>recipient-blacklist-entry</code> and <code>recipient-blacklist-file</code> are not given, spamdyke will not attempt to match recipient addresses against blacklist entries.</p> <p>See Rejecting Senders and Recipients for details.</p>
<code>recipient-blacklist-file</code>	S	FILE	<p>Reject any recipient addresses that match entries in FILE. This option provides better performance than <code>recipient-blacklist-</code></p>

			<p><code>entry</code> for more than a few entries.</p> <p>If <code>recipient-blacklist-file</code> is given multiple times, spamdyke will attempt to match each recipient address against each entry in each given <code>FILE</code>.</p> <p>If <code>recipient-blacklist-entry</code> and <code>recipient-blacklist-file</code> are not given, spamdyke will not attempt to match recipient addresses against blacklist entries.</p> <p>See Rejecting Senders and Recipients for details.</p>
<code>recipient-whitelist-entry</code>		<code>ADDRESS</code>	<p>If the recipient's address matches <code>ADDRESS</code>, bypass all filters.</p> <p>If <code>recipient-whitelist-entry</code> is given multiple times, spamdyke will attempt to match each recipient address against each given <code>ADDRESS</code>.</p> <p>If <code>recipient-whitelist-entry</code> and <code>recipient-whitelist-file</code> are not given, spamdyke will not attempt to match recipient addresses against whitelist entries.</p> <p>See Whitelisting Senders and Recipients for details.</p>
<code>recipient-whitelist-file</code>		<code>FILE</code>	<p>If the recipient's email address matches an entry in <code>FILE</code>, bypass all filters. This option provides better performance than <code>recipient-whitelist-entry</code> for more than a few entries.</p> <p>If <code>recipient-whitelist-file</code> is given multiple times, spamdyke will attempt to match each recipient address against each entry in each given <code>FILE</code>.</p> <p>If <code>recipient-whitelist-entry</code> and <code>recipient-whitelist-file</code> are not given, spamdyke will not attempt to match recipient addresses against whitelist entries.</p> <p>See Whitelisting Senders and Recipients for details.</p>
<code>reject-empty-rdns</code>	<code>r</code>	<p><i>optional:</i>  <code>0, 1, false, true, no</code>  or <code>yes</code></p>	<p>Reject the connection if the remote server has no rDNS name.</p> <p>If <code>reject-empty-rdns</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>reject-empty-rdns</code> is not given, spamdyke will not reject connections with missing rDNS names.</p> <p>If <code>reject-empty-rdns</code> is given without an argument, spamdyke will use a value of <code>yes</code>.</p> <p>See Reverse DNS for details.</p>

<pre>reject-ip-in-cc-rdns</pre>	<pre>c</pre>	<p><i>optional:</i> 0, 1, false, true, no or yes</p>	<p>Search the remote server's rDNS name for its IP address <b>and</b> a two-letter country code. If both are found, reject the connection.</p> <p>If <code>reject-ip-in-cc-rdns</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>reject-ip-in-cc-rdns</code> is not given, spamdyke will not reject connections from servers whose rDNS names contain their IP address and a country code.</p> <p>If <code>reject-ip-in-cc-rdns</code> is given without an argument, spamdyke will use a value of <code>yes</code>.</p> <p>See Reverse DNS for details.</p>
<pre>reject-missing-sender-mx</pre>		<p><i>optional:</i> 0, 1, false, true, no or yes</p>	<p>Check the domain name of the sender's email address for a mail exchanger (an MX or an A record). If neither are found, reject the connection. Requires <code>local-domains-entry</code> or <code>local-domains-file</code>.</p> <p>If <code>reject-missing-sender-mx</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>reject-missing-sender-mx</code> is not given, spamdyke will not reject connections from senders whose email domains do not have an MX record.</p> <p>If <code>reject-missing-sender-mx</code> is given without an argument, spamdyke will use a value of <code>yes</code>.</p> <p>See Rejecting Senders and Recipients for details.</p>
<pre>reject-unresolvable-rdns</pre>	<pre>R</pre>	<p><i>optional:</i> 0, 1, false, true, no or yes</p>	<p>Reject the connection if the remote server's rDNS name does not resolve (search for an A record).</p> <p>If <code>reject-unresolvable-rdns</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>reject-unresolvable-rdns</code> is not given, spamdyke will not reject connections from remote servers whose rDNS names do not resolve.</p> <p>If <code>reject-unresolvable-rdns</code> is given without an argument, spamdyke will use a value of <code>yes</code>.</p> <p>See Reverse DNS for details.</p>
<pre>rejection-text-access-denied</pre>		<pre>TEXT</pre>	<p>Send <code>TEXT</code> to the client as an error message if the remote server is not allowed to send mail due to an entry in the access file.</p> <p>If <code>rejection-text-access-denied</code> is given multiple times, spamdyke will use the last value it finds.</p>

			<p>If <code>rejection-text-access-denied</code> is not given, spamdyke will use the text <code>Refused. Access is denied.</code></p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-auth-failure</code>		<code>TEXT</code>	<p>Send <code>TEXT</code> to the client as an error message if authentication fails for any reason.</p> <p>If <code>rejection-text-auth-failure</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-auth-failure</code> is not given, spamdyke will use the text <code>Refused. Authentication failed.</code></p> <p><code>rejection-text-auth-failure</code> is not valid within configuration directories.</p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-auth-unknown</code>		<code>TEXT</code>	<p>Send <code>TEXT</code> to the client as an error message if the remote server attempts to authenticate using an unsupported authentication method. This should never happen.</p> <p>If <code>rejection-text-auth-unknown</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-auth-unknown</code> is not given, spamdyke will use the text <code>Refused. Unknown authentication method.</code></p> <p><code>rejection-text-auth-unknown</code> is not valid within configuration directories.</p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-dns-blacklist</code>		<code>TEXT</code>	<p>Send <code>TEXT</code> to the client as an error message if the remote server's IP address is found on a DNS blacklist (RBL). The name of the matching RBL will be appended to <code>TEXT</code>. Note: this flag has no effect if the RBL returns a text message; that text will be used instead.</p> <p>If <code>rejection-text-dns-blacklist</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-dns-blacklist</code> is not given, spamdyke will use the text <code>Refused. Your IP address is listed in the DNS RBL at</code></p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-earlytalker</code>		<code>TEXT</code>	<p>Send <code>TEXT</code> to the client as an error message if the remote server sends data before the SMTP greeting banner is displayed.</p> <p>If <code>rejection-text-earlytalker</code> is given multiple times, spamdyke will use the last value</p>

			<p>it finds.</p> <p>If <code>rejection-text-earlytalker</code> is not given, spamdyke will use the text <code>Refused. You are not following the SMTP protocol.</code></p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-empty-rdns</code>		TEXT	<p>Send TEXT to the client as an error message if the remote server has no rDNS name.</p> <p>If <code>rejection-text-empty-rdns</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-empty-rdns</code> is not given, spamdyke will use the text <code>Refused. You have no reverse DNS entry.</code></p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-graylist</code>		TEXT	<p>Send TEXT to the client as an error message if the recipient address has been graylisted.</p> <p>If <code>rejection-text-graylist</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-graylist</code> is not given, spamdyke will use the text <code>Your address has been graylisted. Try again later.</code></p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-ip-blacklist</code>		TEXT	<p>Send TEXT to the client as an error message if the remote server's IP address is found in an IP blacklist file or matches an IP blacklist entry.</p> <p>If <code>rejection-text-ip-blacklist</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-ip-blacklist</code> is not given, spamdyke will use the text <code>Refused. Your IP address is blacklisted.</code></p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-ip-in-cc-rdns</code>		TEXT	<p>Send TEXT to the client as an error message if the remote server's rDNS name contains the remote server's IP address and ends in a two-character country code.</p> <p>If <code>rejection-text-ip-in-cc-rdns</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-ip-in-cc-rdns</code> is not given, spamdyke will use the text <code>Refused. Your reverse DNS entry contains your IP address and a country code.</code></p>

			See SMTP Error Codes for details.
rejection-text-ip-in-rdns-keyword-blacklist		TEXT	<p>Send TEXT to the client as an error message if the remote server's rDNS name contains the remote server's IP address and a banned keyword.</p> <p>If rejection-text-ip-in-rdns-keyword-blacklist is given multiple times, spamdyke will use the last value it finds.</p> <p>If rejection-text-ip-in-rdns-keyword-blacklist is not given, spamdyke will use the text Refused. Your reverse DNS entry contains your IP address and a banned keyword.</p> <p>See SMTP Error Codes for details.</p>
rejection-text-local-recipient		TEXT	<p>Send TEXT to the client as an error message if the specified recipient does not include a domain name.</p> <p>If rejection-text-local-recipient is given multiple times, spamdyke will use the last value it finds.</p> <p>If rejection-text-local-recipient is not given, spamdyke will use the text Improper recipient address. Try supplying a domain name.</p> <p>See SMTP Error Codes for details.</p>
rejection-text-max-recipients		TEXT	<p>Send TEXT to the client as an error message if the remote server gives too many recipient addresses.</p> <p>If rejection-text-max-recipients is given multiple times, spamdyke will use the last value it finds.</p> <p>If rejection-text-max-recipients is not given, spamdyke will use the text Too many recipients. Try the remaining addresses again later.</p> <p>See SMTP Error Codes for details.</p>
rejection-text-missing-sender-mx		TEXT	<p>Send TEXT to the client as an error message if the sender's domain name does not have a DNS entry for a mail exchanger (MX).</p> <p>If rejection-text-missing-sender-mx is given multiple times, spamdyke will use the last value it finds.</p> <p>If rejection-text-missing-sender-mx is not given, spamdyke will use the text Refused. The domain of your sender address has no mail exchanger (MX).</p> <p>See SMTP Error Codes for details.</p>

<pre>rejection- text-rdns- blacklist</pre>		<pre>TEXT</pre>	<p>Send <code>TEXT</code> to the client as an error message if the remote server's rDNS name is found in a blacklist file or matches a blacklist entry.</p> <p>If <code>rejection-text-rdns-blacklist</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-rdns-blacklist</code> is not given, spamdyke will use the text <code>Refused. Your domain name is blacklisted.</code></p> <p>See SMTP Error Codes for details.</p>
<pre>rejection- text- recipient- blacklist</pre>		<pre>TEXT</pre>	<p>Send <code>TEXT</code> to the client as an error message if the recipient address is blacklisted.</p> <p>If <code>rejection-text-recipient-blacklist</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-recipient-blacklist</code> is not given, spamdyke will use the text <code>Refused. Mail is not being accepted at this address.</code></p> <p>See SMTP Error Codes for details.</p>
<pre>rejection- text-reject- all</pre>		<pre>TEXT</pre>	<p>Send <code>TEXT</code> to the client as an error message if all mail is being rejected.</p> <p>If <code>rejection-text-reject-all</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-reject-all</code> is not given, spamdyke will use the text <code>Refused. Mail is not being accepted.</code></p> <p>See SMTP Error Codes for details.</p>
<pre>rejection- text- relaying- denied</pre>		<pre>TEXT</pre>	<p>Send <code>TEXT</code> to the client as an error message if the recipient is not local and the remote server is not allowed relay.</p> <p>If <code>rejection-text-relaying-denied</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-relaying-denied</code> is not given, spamdyke will use the text <code>Refused. Sending to remote addresses (relaying) is not allowed.</code></p> <p>See SMTP Error Codes for details.</p>
<pre>rejection- text-rhs- blacklist</pre>		<pre>TEXT</pre>	<p>Send <code>TEXT</code> to the client as an error message if the remote server's rDNS name or the sender's domain name are found on a right-hand side blacklist (RHSBL). The name of the matching RHSBL will be appended to <code>TEXT</code>. Note: this flag has no effect if the RHSBL returns a text message; that text will be used instead.</p>

			<p>If <code>rejection-text-rhs-blacklist</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-rhs-blacklist</code> is not given, spamdyke will use the text <code>Refused</code>. Your domain name is listed in the RHSBL at</p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-sender-blacklist</code>		TEXT	<p>Send <code>TEXT</code> to the client as an error message if the sender's email address is blacklisted.</p> <p>If <code>rejection-text-sender-blacklist</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-sender-blacklist</code> is not given, spamdyke will use the text <code>Refused</code>. Your sender address has been blacklisted.</p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-smtp-auth-required</code>		TEXT	<p>Send <code>TEXT</code> to the client as an error message if authentication is required to send email and the remote server has not authenticated.</p> <p>If <code>rejection-text-smtp-auth-required</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-smtp-auth-required</code> is not given, spamdyke will use the text <code>Refused</code>. Authentication is required to send mail.</p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-timeout</code>		TEXT	<p>Send <code>TEXT</code> to the client as an error message if the connection times out.</p> <p>If <code>rejection-text-text-timeout</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>rejection-text-text-timeout</code> is not given, spamdyke will use the text <code>Timeout</code>. Talk faster next time.</p> <p><code>rejection-text-timeout</code> is not valid within configuration directories.</p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-tls-failure</code>		TEXT	<p>Send <code>TEXT</code> to the client as an error message if a SSL/TLS session cannot be started with the remote server.</p> <p>If <code>rejection-text-text-tls-failure</code> is given multiple times, spamdyke will use the last value it finds.</p>

			<p>If <code>rejection-text-text-tls-failure</code> is not given, <code>spamdyke</code> will use the text <code>Failed to negotiate TLS connection</code>.</p> <p><code>rejection-text-tls-failure</code> is not valid within configuration directories.</p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-unresolvable-rdns</code>		<code>TEXT</code>	<p>Send <code>TEXT</code> to the client as an error message if the remote server's rDNS name does not resolve.</p> <p>If <code>rejection-text-unresolvable-rdns</code> is given multiple times, <code>spamdyke</code> will use the last value it finds.</p> <p>If <code>rejection-text-unresolvable-rdns</code> is not given, <code>spamdyke</code> will use the text <code>Refused. Your reverse DNS entry does not resolve</code>.</p> <p>See SMTP Error Codes for details.</p>
<code>rejection-text-zero-recipients</code>		<code>TEXT</code>	<p>Send <code>TEXT</code> to the client as an error message if none of the recipients given by the remote server are accepted.</p> <p>If <code>rejection-text-zero-recipients</code> is given multiple times, <code>spamdyke</code> will use the last value it finds.</p> <p>If <code>rejection-text-zero-recipients</code> is not given, <code>spamdyke</code> will use the text <code>Refused. You must specify at least one valid recipient</code>.</p> <p><code>rejection-text-zero-recipients</code> is not valid within configuration directories.</p> <p>See SMTP Error Codes for details.</p>
<code>relay-level</code>		<code>block-all, no-check, normal or allow-all</code>	<p><code>block-all</code>: Block all relaying attempts, even if the sender has authenticated or the access file or an environment variable should allow relaying. Messages to local recipients will still be accepted. Requires <code>local-domains-entry</code> or <code>local-domains-file</code> and <code>access-file</code>.</p> <p><code>no-check</code>: Do not prevent relaying; allow <code>qmail</code> (or another filter) to prevent relaying as appropriate.</p> <p><code>normal</code>: Prevent relaying unless the sender authenticates, the access file allows relaying or an environment variable allows relaying. Requires <code>local-domains-entry</code> or <code>local-domains-file</code> and <code>access-file</code>.</p> <p><code>allow-all</code>: Allow relaying from all senders. Note: This creates an open relay and is not recommended.</p>

			<p>If <code>relay-level</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>relay-level</code> is not given, spamdyke will use a value of <code>normal</code>.</p> <p><code>relay-level</code> is not valid within configuration directories.</p> <p>See Relaying for details.</p>
<code>rhs-blacklist-entry</code>	X	RHSBL	<p>Check the remote server's rDNS name and the sender email address' domain name against the right hand-side blackhole list <code>RHSBL</code>. If it is found, the connection is rejected. NOTE: Using more than a few RHS blacklists can cause serious performance problems.</p> <p>If <code>rhs-blacklist-entry</code> is given multiple times, spamdyke will check each given <code>RHSBL</code> for the remote server's rDNS name and the sender's email address domain name.</p> <p>If <code>rhs-blacklist-entry</code> and <code>rhs-blacklist-file</code> are not given, spamdyke will not check any blackhole lists for the remote server's rDNS name and the sender's email address domain name.</p> <p>See DNS RHSBLs for details.</p>
<code>rhs-blacklist-file</code>		FILE	<p>Check the remote server's domain name and the sender email address' domain name against each of the right hand-side blackhole lists listed in <code>FILE</code>. If it is found on any of the lists, the connection is rejected. NOTE: Using more than a few RHS blacklists can cause serious performance problems.</p> <p>If <code>rhs-blacklist-file</code> is given multiple times, spamdyke will check each blackhole list listed in each given <code>FILE</code> for the remote server's rDNS name and the sender's email address domain name.</p> <p>If <code>rhs-blacklist-file</code> and <code>rhs-blacklist-file</code> are not given, spamdyke will not check any blackhole lists for the remote server's rDNS name and the sender's email address domain name.</p> <p>See DNS RHSBLs for details.</p>
<code>rhs-whitelist-entry</code>		RHSWHITELIST	<p>Check the remote server's domain name and the sender email address' domain name against the right hand-side whitelist <code>RHSWHITELIST</code> (essentially an RHSBL that contains whitelisted domains). If it is found, all filters are bypassed. NOTE: Using more than a few RHS whitelists can cause serious performance problems.</p> <p>If <code>rhs-whitelist-entry</code> is given multiple times, spamdyke will check each given <code>RHSWHITELIST</code> for the remote server's rDNS</p>

			<p>name and the sender's email address domain name.</p> <p>If <code>rhs-whitelist-entry</code> and <code>rhs-whitelist-file</code> are not given, spamdyke will not check any whitelists for the remote server's rDNS name and the sender's email address domain name.</p> <p>See DNS Whitelists for details.</p>
<code>rhs-whitelist-file</code>		<code>FILE</code>	<p>Check the remote server's domain name and the sender email address' domain name against each of the right hand-side whitelists (essentially an RHSBL that contains whitelisted domains) listed in <code>FILE</code>. If it is found on any of the lists, all filters are bypassed. NOTE: Using more than a few RHS whitelists can cause serious performance problems.</p> <p>If <code>rhs-whitelist-file</code> is given multiple times, spamdyke will check each whitelist listed in each given <code>FILE</code> for the remote server's rDNS name and the sender's email address domain name.</p> <p>If <code>rhs-whitelist-file</code> and <code>rhs-whitelist-entry</code> are not given, spamdyke will not check any whitelist for the remote server's rDNS name and the sender's email address domain name.</p> <p>See DNS Whitelists for details.</p>
<code>run-as-user</code>		<code>USER[:GROUP]</code>	<p>As soon as possible, change the running user identity to the user with the username or ID <code>USER</code>. If <code>GROUP</code> is provided, also change the group identity to the system group with the name <code>GROUP</code> or ID <code>GROUP</code>. This feature requires spamdyke to be started as a user with the ability to switch identities (typically the superuser).</p> <p>If <code>run-as-user</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>run-as-user</code> is not given, spamdyke will not attempt to switch user identities.</p> <p><code>run-as-user</code> is not valid within configuration directories.</p> <p>See Permissions for details.</p>
<code>sender-blacklist-entry</code>		<code>ADDRESS</code>	<p>Reject the connection if the sender's email address matches <code>ADDRESS</code>.</p> <p>If <code>sender-blacklist-entry</code> is given multiple times, spamdyke will reject the connection if the sender's email address matches any of the given <code>ADDRESS</code> values.</p> <p>If <code>sender-blacklist-entry</code> and <code>sender-blacklist-file</code> are not given, spamdyke</p>

			<p>will not blacklist any sender email addresses.</p> <p>See Rejecting Senders and Recipients for details.</p>
<code>sender-blacklist-file</code>	<code>s</code>	<code>FILE</code>	<p>Reject the connection if the sender's email address matches an entry in <code>FILE</code>. This option provides better performance than <code>sender-blacklist-entry</code> for more than a few entries.</p> <p>If <code>sender-blacklist-file</code> is given multiple times, spamdyke will reject the connection if the sender's email address matches any of the entries in each given <code>FILE</code>.</p> <p>If <code>sender-blacklist-entry</code> and <code>sender-blacklist-file</code> are not given, spamdyke will not blacklist any sender email addresses.</p> <p>See Rejecting Senders and Recipients for details.</p>
<code>sender-whitelist-entry</code>		<code>ADDRESS</code>	<p>If the sender's email address matches <code>ADDRESS</code>, bypass all filters.</p> <p>If <code>sender-whitelist-entry</code> is given multiple times, spamdyke will compare the sender's email address to each given <code>ADDRESS</code>.</p> <p>If <code>sender-whitelist-entry</code> and <code>sender-whitelist-file</code> are not given, spamdyke will not whitelist any sender email addresses.</p> <p>See Whitelisting Senders and Recipients for details.</p>
<code>sender-whitelist-file</code>		<code>FILE</code>	<p>If the sender's email address matches an entry in <code>FILE</code>, bypass all filters. This option provides better performance than <code>sender-whitelist-entry</code> for more than a few entries.</p> <p>If <code>sender-whitelist-file</code> is given multiple times, spamdyke will compare the sender's email address to each entry in each given <code>FILE</code>.</p> <p>If <code>sender-whitelist-entry</code> and <code>sender-whitelist-file</code> are not given, spamdyke will not whitelist any sender email addresses.</p> <p>See Whitelisting Senders and Recipients for details.</p>
<code>smtp-auth-command</code>		<code>COMMAND</code>	<p>Perform SMTP AUTH verification using <code>COMMAND</code>. If the authentication is valid, all filters will be bypassed. This option may have no effect, depending on the value of <code>smtp-auth-level</code>.</p> <p>If <code>smtp-auth-command</code> is given multiple times, spamdyke will authenticate using each</p>

			<p>given <code>COMMAND</code> until one of them indicates success.</p> <p>If <code>smtp-auth-command</code> is not given, spamdyke will not process authentication. Depending on the value of <code>smtp-auth-level</code>, authentication may still be possible.</p> <p><code>smtp-auth-command</code> is not valid within configuration directories.</p> <p>See SMTP AUTH for details.</p>
<code>smtp-auth-level</code>		<p><code>none</code>, <code>observe</code>, <code>ondemand</code>, <code>ondemand-encrypted</code>, <code>always</code> or <code>always-encrypted</code></p>	<p><code>none</code>: Do not offer or allow authentication, even if qmail has been patched to provide it.</p> <p><code>observe</code>: Observe authentication only (and trust qmail's responses), do not offer it. This value has no effect if qmail has not been patched to offer authentication.</p> <p><code>ondemand</code>: If qmail offers authentication, observe any authentication attempts and trust qmail's responses. If qmail does not offer authentication, spamdyke will offer cleartext authentication, then process it using the value of <code>smtp-auth-command</code>.</p> <p><code>ondemand-encrypted</code>: If qmail offers authentication, observe any authentication attempts and trust qmail's responses. If qmail does not offer authentication, spamdyke will offer cleartext and encrypted authentication, then process it using the value of <code>smtp-auth-command</code>.</p> <p><code>always</code>: Always offer cleartext authentication, then process it using the value of <code>smtp-auth-command</code>. If qmail attempts to offer authentication, spamdyke will hide qmail's offer and prevent the authentication data from reaching qmail.</p> <p><code>always-encrypted</code>: Always offer cleartext and encrypted authentication, then process it using the value of <code>smtp-auth-command</code>. If qmail attempts to offer authentication, spamdyke will hide qmail's offer and prevent the authentication data from reaching qmail.</p> <p>If <code>smtp-auth-level</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>smtp-auth-level</code> is not given, spamdyke will use a value of <code>observe</code>.</p> <p><code>smtp-auth-level</code> is not valid within configuration directories.</p> <p>See SMTP AUTH for details.</p>
<code>tls-certificate-</code>		<code>FILE</code>	<p>Decrypt SSL/TLS traffic using the SSL certificate in <code>FILE</code>. The certificate must be in</p>

file			<p>PEM format. If <code>FILE</code> does not also contain the private key, <code>tls-privatekey-file</code> must be used. This option has no effect unless <code>tls-level</code> is also given.</p> <p>If <code>tls-certificate-file</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>tls-certificate-file</code> is not given, spamdyke will not decrypt SSL/TLS traffic. The encrypted traffic will be passed through to qmail.</p> <p><code>tls-certificate-file</code> is not valid within configuration directories.</p> <p>See TLS for details.</p>
tls-level		none, smtp or smtps	<p><code>none</code>: Do not offer or allow SSL/TLS, even if qmail supports it.</p> <p><code>smtp</code>: If <code>tls-certificate-file</code> is given, offer TLS during the SMTP conversation and decrypt the traffic. If <code>tls-certificate-file</code> is not given, allow qmail to offer TLS (if it has been patched to provide TLS) and pass the encrypted traffic to qmail.</p> <p><code>smtps</code>: Initiate a SSL session at the beginning of the connection, before SMTP begins.</p> <p>If <code>tls-level</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>tls-level</code> is not given, spamdyke will use a value of <code>smtp</code>.</p> <p><code>tls-level</code> is not valid within configuration directories.</p> <p>See TLS for details.</p>
tls-privatekey-file		FILE	<p>Read the private key for the SSL certificate (given with <code>tls-certificate-file</code>) from <code>FILE</code>. <code>FILE</code> must be in PEM format. Requires <code>tls-certificate-file</code>.</p> <p>If <code>tls-privatekey-file</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>tls-privatekey-file</code> is not given, spamdyke will look for the private key in the certificate file.</p> <p><code>tls-privatekey-file</code> is not valid within configuration directories.</p> <p>See TLS for details.</p>
tls-privatekey-password		PASSWORD	<p>Use <code>PASSWORD</code> to decrypt the SSL private key (given with <code>tls-certificate-file</code> or <code>tls-privatekey-file</code>), if necessary. NOTE: this option reveals the password in the process list! Requires <code>tls-certificate-file</code> and/or <code>tls-privatekey-file</code>.</p>

			<p>If <code>tls-privatekey-password</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>tls-privatekey-password</code> and <code>tls-privatekey-password-file</code> are not given, spamdyke will attempt to load the private key without a password.</p> <p><code>tls-privatekey-password</code> is not valid within configuration directories.</p> <p>See TLS for details.</p>
<code>tls-privatekey-password-file</code>		<code>FILE</code>	<p>Read the password to decrypt the private key for the SSL certificate (from <code>tls-certificate-file</code>) from the first line of <code>FILE</code>, if necessary. Requires <code>tls-certificate-file</code> and/or <code>tls-password-file</code>.</p> <p>If <code>tls-privatekey-password-file</code> is given multiple times, spamdyke will use the last value it finds.</p> <p>If <code>tls-privatekey-password</code> and <code>tls-privatekey-password-file</code> are not given, spamdyke will attempt to load the private key without a password.</p> <p><code>tls-privatekey-password-file</code> is not valid within configuration directories.</p> <p>See TLS for details.</p>

## CONFIGURATION FILES

### `config-file`

The configuration file format is very simple. Each line should use the following format:

`OPTION=VALUE`

`OPTION` is the long version of a spamdyke option. See Usage for details.

`VALUE` is the parameter for the option. Note: While multi-word values must be quoted on the command line, quotes are not allowed in configuration files. spamdyke reads the entire `VALUE` after the equals sign, even if it contains spaces, so no quoting is needed.

Blank lines and lines beginning with `#` are ignored.

For example:

```
smtp-auth-
command=/home/vpopmail/bin/vchkpw
/bin/true
rdns-blacklist-
dir=/home/vpopmail/blacklist_rdns.d
graylist-
dir=/home/vpopmail/graylist.d
```

```
check-dnsrbl=dul.dnsbl.sorbs.net
check-dnsrbl=zombie.dnsbl.sorbs.net
max-recipients=5
```

True/false options can be given without a `VALUE` to activate them. `yes`, `true` and `1` are also acceptable. The options can also be explicitly deactivated with `no`, `false` or `0` (or the option can be simply removed). For example, the following lines all have the same effect:

```
reject-empty-rdns
reject-empty-rdns=yes
reject-empty-rdns=true
reject-empty-rdns=1
```

A configuration file is utilized by passing the command line option `config-file` to spamdyke:

```
spamdyke --config-file
/etc/spamdyke.conf ...
```

The `config-file` option can also be used within configuration files to include other configuration files if desired. When configuration files are in use, options may still be provided on the command line as well, in any combination. If contradictory options are found, the option in the configuration file will be used. For example, if the following command line were used:

```
spamdyke --reject-empty-rdns --
config-file /etc/spamdyke.conf ...
```

And `/etc/spamdyke.conf` contained the following line:

```
reject-empty-rdns=false
```

spamdyke would deactivate the `reject-empty-rdns` filter because the configuration file is read after the command line has been scanned. This can be confusing, so the best practice is to avoid specifying the same option in multiple places without good reason.

Some options can be given multiple times and spamdyke will use all of the values it finds. For example, if the following lines are given, spamdyke will search each of the files for a match to the sender's email address:

```
sender-blacklist-
file=/home/vpopmail/sender_blacklis
t.txt
sender-blacklist-
file=/home/vpopmail/more_sender_bla
cklist.txt
sender-blacklist-
file=/home/vpopmail/additional_send
er_blacklist.txt
```

However, in some situations, it may be necessary to remove one or all of the values. Most commonly, this occurs when the values are set in a global configuration file and are unset in a configuration directory. To remove one specific value from a list, the value should be specified with an exclamation point preceding the value:

```
OPTION=!VALUE
```

For example:

```
sender-blacklist-
file=!/home/vpopmail/sender_blackli
st.txt
```

If the given value was not previously set, no action will be taken.

To clear all values from a list, three exclamation points should be given instead of a value:

```
OPTION=!!!
```

For example, if the following line is given, the `sender-blacklist-file` option will be cleared and spamdyke will behave as though `sender-blacklist-file` had never appeared:

```
sender-blacklist-file=!!!
```

Note: spamdyke processes configuration directives in the order they are read. If an option is cleared and later set again, the option will retain the last value. For example, if the following five lines appear in this order:

```
sender-blacklist-  
file=/home/vpopmail/sender_blacklis  
t.txt  
sender-blacklist-  
file=/home/vpopmail/more_sender_bla  
cklist.txt  
sender-blacklist-  
file=/home/vpopmail/additional_send  
er_blacklist.txt  
sender-blacklist-file=!!!  
sender-blacklist-  
file=/home/vpopmail/last_blacklist.  
txt
```

spamdyke will set the first three values, clear them, then set the last value. When the sender blacklist filter runs, it will search the file

```
/home/vpopmail/last_blacklist.txt.
```

**NOTE:** It may seem that scanning a configuration file instead of the command line would impose a performance penalty each time spamdyke is started. However, the reverse seems to be true. Some rudimentary testing has indicated the configuration files are actually faster. This is likely due to inefficiencies in GNU's `getopt_long()` function.

## CONFIGURATION DIRECTORIES

```
config-dir  
config-dir-search
```

Configuration directories allow spamdyke's behavior to be altered based on the remote server's IP address, the remote server's rDNS name, the sender's email address, the recipient's email address or any combination of those things. This can be very useful when, for example, graylisting should be deactivated for a specific sender. IP addresses can be whitelisted for specific recipients without whitelisting them for everyone. The possibilities are nearly endless.

**NOTE:** Configuration directories are confusing and complicated. Unless you really need the advanced configuration scenarios they offer, don't use them.

Configuration directories are given with the `config-dir` option. The option's value should be the path to the directory that contains the subdirectories explained below. If `config-dir` is given multiple times, spamdyke will

search each given directory structure and load all of the matching files before it continues processing the SMTP connection.

In essence, a configuration directory is a special directory structure that contains configuration files. spamdyke determines which files to load based on the names of the directories and the details of the SMTP connection. Not all options are valid within configuration directories, but in all other respects the files follow the same rules as global configuration files. See Usage for details of which options are valid within configuration directories. See Configuration Files for details of the configuration file format.

When spamdyke loads a file from a configuration directory, it will do so because the names of the directories and the name of the file match all or part of the information from the SMTP connection. The last piece of information should always be used as the name of the file, not the name of a directory.

To create a file using the IP address of the remote server, first create a directory structure that begins with `_ip_` and uses the first three octets of the address as directory names. For example, if the IP address is `11.22.33.44`, the directory structure should look like this:

```
.../_ip_/11/22/33
```

Use the fourth octet as the name of the configuration file. For example:

```
.../_ip_/11/22/33/44
```

To create a file using the rDNS name of the remote server, first create a directory structure that begins with `_rdns_` and contains directories named using the rDNS name with its words reversed. For example, if the rDNS name is `mail.internal.headquarters.example.com`, the directory structure should look like this:

```
.../_rdns_/com/example/headquarters  
/internal
```

The last word of the rDNS name is used as the name of the configuration file. For example:

```
.../_rdns_/com/example/headquarters  
/internal/mail
```

To create a file using the sender's email address, first create a directory structure that begins with `_sender_` and contains directories using the domain portion of the sender's email address with its words reversed and ending in `_at_`. For example, if the sender's email address is `mom@home.example.com`, the directory structure should look like this:

```
.../_sender_/com/example/home/_at_
```

The sender's username is used as the name of the configuration file. For example:

```
.../_sender_/com/example/home/_at_  
mom
```

To create a file using the recipient's email address, first create a directory structure that begins with `_recipient_` and contains directories using the domain portion of the recipient's email address with its words reversed and ending in `_at_`. For example, if the recipient's email address is `kid@school.example.com`, the directory structure should look like this:

```
.../_recipient_/com/example/school/
_at_
```

The recipient's username is used as the name of the configuration file. For example:

```
.../_recipient_/com/example/school/
_at_/kid
```

spamdyke will match partial IP addresses, rDNS names, sender email addresses or recipient email addresses left-most portions of the directory structure are omitted. Note: spamdyke will never read a configuration file named `_at_`. If the sender's or recipient's username are omitted, the `_at_` directory must be omitted as well. For example, if the IP address is `11.22.33.44`, the rDNS name is `mail.internal.headquarters.example.com`, the sender's email address is `mom@home.example.com` and the recipient's email address is `kid@school.example.com`, spamdyke will look for configuration files with the following paths:

```
.../_ip_/11/22/33/44
.../_ip_/11/22/33
.../_ip_/11/22
.../_ip_/11
.../_rdns_/com/example/headquarters
/internal/mail
.../_rdns_/com/example/headquarters
/internal
.../_rdns_/com/example/headquarters
.../_rdns_/com/example
.../_rdns_/com
.../_recipient_/com/example/school/
_at_/kid
.../_recipient_/com/example/school
.../_recipient_/com/example
.../_recipient_/com
.../_sender_/com/example/home/_at_/
mom
.../_sender_/com/example/home
.../_sender_/com/example
.../_sender_/com
```

Configuration directories can be nested to create more specific targets. For example, if the IP address of the remote server is `11.22.33.44` and the sender's email address is `mom@home.example.com`, spamdyke will read a configuration file if its path is either of the following:

```
.../_ip_/11/22/33/44/_sender_/com/e
xample/home/_at_/mom
.../_sender_/com/example/home/_at_/
mom/_ip_/11/22/33/44
```

If only portions of the IP address or sender address are used, the possible list of file paths increases dramatically:

```
.../_ip_/11/22/33/44/_sender_/com/e
xample/home
```

```

.../_ip_/11/22/33/44/_sender_/com/e
xample
.../_ip_/11/22/33/44/_sender_/com
.../_ip_/11/22/33/_sender_/com/exam
ple/home/_at_/mom
.../_ip_/11/22/_sender_/com/example
/home/_at_/mom
.../_ip_/11/_sender_/com/example/ho
me/_at_/mom
.../_sender_/com/example/home/_ip_/
11/22/33
.../_sender_/com/example/_ip_/11/22
.../_sender_/com/_ip_/11

```

spamdyke will check for every possible combination of partial paths (not all permutations are listed here). rDNS name directories and recipient directories can also be nested to create configuration files that will only be loaded if all four pieces of information match.

If all of that isn't confusing enough, spamdyke will only read one file from a `_ip_`, `_rdns_`, `_sender_` or `_recipient_` directory, even if more matches are possible. For example, if the remote IP address is `11.22.33.44`, the sender's email address is `mom@home.example.com` and the recipient's email address is `kid@school.example.com` and two files exist with the following paths:

```

.../_ip_/11/22/33/44/_sender_/com/e
xample/home/_at_/mom
.../_ip_/11/22/33/44/_recipient_/co
m/example/school/_at_/kid

```

spamdyke will only read one of the files because the `_ip_` directory cannot be matched more than once. This behavior can be changed with the `config-dir-search` option.

`config-dir-search` can be given multiple times; the values will be added together to create a composite value.

The possible values are:

- `first`: Match each `_ip_`, `_rdns_`, `_sender_` or `_recipient_` directory only once. **NOTE:** The `first` value erases the composite value created from the other possible values, essentially "resetting" the `config-dir-search` option.
- `all-ip`: Match each `_ip_` directory as many times as possible.
- `all-rdns`: Match each `_rdns_` directory as many times as possible.
- `all-sender`: Match each `_sender_` directory as many times as possible.
- `all-recipient`: Match each `_recipient_` directory as many times as possible.

To aid with troubleshooting, spamdyke will log the paths it searches if the `log-level` option is `debug` or higher.

## CONFIGURATION TESTS

```
config-test
config-test-smtpauth-username
config-test-smtpauth-password
```

spamdyke has the ability to scan its configuration and look for common configuration mistakes. It checks file paths, permissions, graylist folders, directory structures, SMTP AUTH commands, TLS certificates and more. This feature was inspired by Apache's ability to check its configuration file for syntax errors.

To use the testing feature:

1. Find and copy the entire spamdyke command line from your "supervise" script or xinetd configuration file, including the qmail command(s).
2. At a command prompt, login as root and paste the spamdyke command without running it.
3. Add the option `--config-test` among the spamdyke options (before the qmail command). If appropriate, add the options `--config-test-smtpauth-username` and `--config-test-smtpauth-password`.
4. Run the command and carefully read the results. More output can be obtained by increasing the logging level (no test output goes to syslog).

If the `run-as-user` option is not given in your spamdyke configuration, it should be used on the command line to give spamdyke the name (or user ID) of the account used to run the mail server. The group name (or ID) can also be given. Before spamdyke runs its configuration tests, it will change process ownership to run as the given user. That way, the filesystem permissions tests will be accurate.

If spamdyke is configured to provide SMTP AUTH (using the `smtp-auth-level` and `smtp-auth-command` options), the `--config-test-smtpauth-username` and `--config-test-smtpauth-password` options should be used to provide a valid username and password for authentication. spamdyke will run the SMTP AUTH command to test its capabilities and make recommendations.

**IMPORTANT! DANGER! WARNING!** DO NOT EVER PUT THE `--config-test` OPTION IN THE SPAMDYKE COMMAND LINE THAT IS RUN FOR INCOMING CONNECTIONS! YOUR MAIL SERVER WILL IMMEDIATELY STOP RECEIVING MAIL AND REMOTE USERS WILL SEE ONLY THE DIAGNOSTIC OUTPUT! If you make this mistake and ask for help, expect to be publicly mocked. You have been warned.

## LOG MESSAGES

```
log-level
log-target
```

The `log-target` option controls where spamdyke logs its messages. By default, `log-target` is set to `syslog`, which sends log messages to the system syslog facility. When `log-target` is set to `stderr`, messages are sent to standard error (stderr) instead. For most qmail installations, this will cause spamdyke's messages to be logged by the "multilog" program, along with qmail-smtpd's output. If `log-target` is given multiple times with different values, spamdyke will send its output to each given target.

When spamdyke logs to syslog, it uses the `LOG_MAIL` facility, which typically puts the messages in `/var/log/maillog`. (Note: Plesk reconfigures syslog to put the messages in

`/usr/local/psa/var/log/maillog`.)

Regardless of how the messages are logged, errors are always preceded by the text `ERROR:` and are fairly self-explanatory. Whenever possible, spamdyke will recover from an error and continue processing mail.

Philosophically, it's better to continue receiving spam than to block all mail.

The `log-level` option controls how much logging takes place. The following values are supported:

- `none`: No logging at all, even if errors occur. This is not recommended.
- `error`: Critical errors only, including authentication failures. This is the default when `log-level` is not given.
- `info`: Everything from `error` plus logging of messages (sender, recipient, IP address, rDNS name and authenticated username). This is the value used when `log-level` is given with no value.
- `verbose`: Everything from `info` plus non-critical errors such as network errors caused by the remote host, protocol errors, `config-test` status messages and child process error messages. At this level, spamdyke will also print messages to show which filter blocked the connection (if applicable) and some details about the filter's settings. These messages will be prefixed with `FILTER:`.
- `debug`: Everything from `verbose` plus high-level debugging messages, intended to show the processing path within spamdyke. This level is handy for troubleshooting but it can be rather noisy. Extra messages generated by this level will be prefixed with `DEBUG( ):` and will show the file and line number within the spamdyke source code where the message was produced. **NOTE:** If the `configure` script is run with the `--without-debug-output` option, spamdyke will accept the `debug` value but it will not print any more output than if `verbose` were used.
- `excessive`: Everything from `debug` plus lots of internal status messages. This value should only be used for development. Extra messages generated by

this level will be prefixed with `EXCESSIVE()` and will show the file and line number within the spamdyke source code where the message was produced. **NOTE:** Unless the `configure` script is run with the `--with-excessive-output` option, spamdyke will not produce any more output for `excessive` than if `debug` were used.

Note that `log-level` must be used with care on the command line. Specifically, when `--log-level` is used, the value must be separated by an equals sign and no spaces. When `-l` is used, the value must not be separated by spaces or anything else. For example, the following two command lines will work:

```
/usr/local/bin/spamdyke --log-level=verbose ...
/usr/local/bin/spamdyke -lverbose
...
```

The `log-level` option may also be given with no value at all, which is the same as specifying `info`. The following two command lines are also valid:

```
/usr/local/bin/spamdyke --log-level
...
/usr/local/bin/spamdyke -l ...
```

Each message log entry (produced when the value of `log-level` is `info` or higher) takes the following form (error messages and debugging statements are text preceeded by `ERROR:`, `FILTER:`, `DEBUG:` or `EXCESSIVE:`):

```
CODE from: SENDER to: RECIPIENT
origin_ip: IPADDRESS origin_rdns:
RDNSNAME auth: USERNAME [ reason:
REALCODE ]
```

This format makes the logs very easy to parse from other scripts for monitoring and graphing.

The possible values of `CODE` are listed below:

CODE	Description	Related Option(s)
ALLOWED	The message passed all filters. qmail may still bounce the message for other reasons, however.	
ALLOWED_AUTHENTICATED	The remote client successfully authenticated using SMTP AUTH with spamdyke. If qmail is patched to provide SMTP AUTH, this code will never be used.	
ALLOWED_TLS	The remote client successfully started a TLS session with spamdyke.	
DENIED_ACCESS_DENIED	The connection was blocked because the remote server's IP address or rDNS name was found in the access file with a <code>deny</code> command.	<code>access-file</code>
DENIED_AUTH_REQUIRED	The message was blocked because the remote server has not authenticated, which is required.	<code>filter-level</code>
DENIED_BLACKLIST_IP	The connection was blocked because the remote server's IP address is blacklisted.	<code>ip-blacklist-entry</code> <code>ip-blacklist-file</code>
DENIED_BLACKLIST_NAME	The connection was blocked because the remote server's rDNS name is blacklisted.	<code>rdns-blacklist-entry</code> <code>rdns-blacklist-</code>

		file rdns-blacklist-dir
DENIED_EARLYTALKER	The connection was blocked because the remote server began sending data before the SMTP greeting was issued.	greeting-delay-secs
DENIED_GRAYLISTED	The recipient was blocked because the sender/recipient combination was graylisted. The SMTP connection continues after this error occurs.	graylist-level
DENIED_IP_IN_CC_RDNS	The connection was blocked because the remote server's IP address was found in the remote server's rDNS name <u>and</u> the remote server's rDNS name ends in a country code.	reject-ip-in-cc-rdns
DENIED_IP_IN_RDNS	The connection was blocked because the remote server's IP address was found in the remote server's rDNS name <u>and</u> a prohibited keyword was found in the remote server's rDNS name.	ip-in-rdns-keyword-blacklist-entry ip-in-rdns-keyword-blacklist-file
DENIED_OTHER	The connection was rejected by qmail (or another downstream filter), not spamdyke.	
DENIED_RBL_MATCH	The connection was blocked because the remote server's IP address was found on a DNS RBL.	dns-blacklist-entry dns-blacklist-file
DENIED_RDNS_MISSING	The connection was blocked because the remote server has no rDNS name at all.	reject-empty-rdns
DENIED_RDNS_RESOLVE	The connection was blocked because the remote server's rDNS name does not resolve.	reject-unresolvable-rdns
DENIED_RHSBL_MATCH	The connection was blocked because the remote server's reverse DNS name was found on a right hand-side DNS blacklist (RHSBL) <b>OR</b> because the sender's domain name was found on a right hand-side DNS blacklist (RHSBL).	rhs-blacklist-entry rhs-blacklist-file
DENIED_RECIPIENT_BLACKLISTED	The recipient was blocked because the recipient email address is blacklisted.	recipient-blacklist-entry recipient-blacklist-file
DENIED_REJECT_ALL	The message was blocked because all mail is being rejected.	filter-level
DENIED_RELAYING	The recipient was blocked because the recipient's domain is not locally hosted and the remote server is not allowed to relay.	relay-level
DENIED_SENDER_BLACKLISTED	The connection was blocked because the sender's email address is blacklisted.	sender-blacklist-entry sender-blacklist-file
DENIED_SENDER_NO_MX	The connection was blocked because the sender's domain has no mail exchanger, making the sender address invalid.	reject-missing-sender-mx
DENIED_TOO_MANY_RECIPIENTS	The recipient was blocked because the limit was reached for this connection. The	max-recipients

	SMTP connection continues after this error occurs.	
<code>DENIED_UNQUALIFIED_RECIPIENT</code>	The recipient was blocked because the address had no domain name. The SMTP connection continues after this error occurs.	
<code>DENIED_ZERO_RECIPIENTS</code>	The message was blocked because no valid recipients have been specified.	
<code>FAILED_AUTH</code>	The remote server attempted to authenticate but the given username and/or password were incorrect.	<code>smtp-auth-level</code>
<code>FAILED_TLS</code>	The remote client attempted to start a TLS session but SSL negotiation failed.	
<code>TIMEOUT</code>	The connection timed out, either in total time or idle time. If the connection was already being blocked for another reason, the code for that error is given as <code>REALCODE</code> .	<code>connection-timeout-secs</code> <code>idle-timeout-secs</code>
<code>TLS_ENCRYPTED</code>	The remote server has started a TLS session with gmail. spamdyke does not have access to the server's certificate file, so it cannot decrypt the traffic to log any information about senders or recipients.	<code>tls-level</code>
<code>UNKNOWN_AUTH</code>	The remote server requested an authentication method spamdyke doesn't support. This shouldn't happen.	

`SENDER` is the sender email address, if known, or `(unknown)` otherwise. NOTE: According to RFC 821, it is legal to deliver messages with no sender address. Most bounce messages are delivered this way.

`RECIPIENT` is the recipient email address, if known, or `(unknown)` otherwise. If `CODE` is `ALLOWED`, the recipient email address will be known.

`IPADDRESS` is the IP address of the remote server. This value is always known.

`RDNSNAME` is the rDNS name of the remote server, if known, or `(unknown)` otherwise.

`USERNAME` is the username given during authentication, if authentication was successful, or `(unknown)` otherwise.

`REALCODE` is only present if `CODE` is `TIMEOUT` and the connection was going to be blocked anyway. For example, if a remote server has no rDNS entry and the connection is going to be blocked but the connection times out instead,

`CODE` will be `TIMEOUT` and `REALCODE` will be

`DENIED_RDNS_MISSING`.

## SMTP ERROR CODES

```

policy-url
rejection-text-access-denied
rejection-text-auth-failure
rejection-text-auth-unknown

```

```

rejection-text-dns-blacklist
rejection-text-earlytalker
rejection-text-empty-rdns
rejection-text-graylist
rejection-text-ip-blacklist
rejection-text-ip-in-cc-rdns
rejection-text-ip-in-rdns-keyword-
blacklist
rejection-text-local-recipient
rejection-text-max-recipients
rejection-text-missing-sender-mx
rejection-text-rdns-blacklist
rejection-text-recipient-blacklist
rejection-text-reject-all
rejection-text-relaying-denied
rejection-text-rhs-blacklist
rejection-text-sender-blacklist
rejection-text-smtp-auth-required
rejection-text-timeout
rejection-text-tls-failure
rejection-text-unresolvable-rdns
rejection-text-zero-recipients

```

When spamdyke blocks a connection and returns an error code to a remote server, the text it sends is different from what appears in the logs (above). It is more user-friendly, just in case a human ever reads it (some, but not all, mail servers display the rejection message in bounce messages).

The messages can be changed using the options that are listed in the third column of the table below.

The messages that correspond to the syslog codes are:

syslog code	SMTP message	Option to change message
DENIED_ACCESS_DENIED	Refused. Access is denied.	rejection-text-access-denied
DENIED_AUTH_REQUIRED	Refused. Authentication is required to send mail.	rejection-text-smtp-auth-required
DENIED_BLACKLIST_IP	Refused. Your IP address is blacklisted.	rejection-text-ip-blacklist
DENIED_BLACKLIST_NAME	Refused. Your domain name is blacklisted.	rejection-text-rdns-blacklist
DENIED_EARLYTALKER	Refused. You are not following the SMTP protocol.	rejection-text-earlytalker
DENIED_GRAYLISTED	Your address has been graylisted. Try again later.	rejection-text-graylist
DENIED_IP_IN_CC_RDNS	Refused. Your reverse DNS entry contains your IP address and a country code.	rejection-text-ip-in-cc-rdns
DENIED_IP_IN_RDNS	Refused. Your reverse DNS entry contains your IP address and a banned keyword.	rejection-text-ip-in-rdns-keyword-blacklist
DENIED_OTHER	The text returned by qmail (or the downstream filter that generated the	

	rejection).	
DENIED_RBL_MATCH	The text returned by the DNS RBL (if any) or Refused. Your IP address is listed in the RBL at <i>name</i> .	rejection-text-dns-blacklist
DENIED_RDNS_MISSING	Refused. You have no reverse DNS entry.	rejection-text-empty-rdns
DENIED_RDNS_RESOLVE	Refused. Your reverse DNS entry does not resolve.	rejection-text-unresolvable-rdns
DENIED_RHSBL_MATCH	The text returned by the RHSBL (if any) or Refused. Your domain name is listed in the RHSBL at <i>name</i> .	rejection-text-rhs-blacklist
DENIED_RECIPIENT_BLACKLISTED	Refused. Mail is not being accepted at this address.	rejection-text-recipient-blacklist
DENIED_REJECT_ALL	Refused. Mail is not being accepted.	rejection-text-reject-all
DENIED_RELAYING	Refused. Sending to remote addresses (relaying) is not allowed.	rejection-text-relaying-denied
DENIED_SENDER_BLACKLISTED	Refused. Your sender address has been blacklisted.	rejection-text-sender-blacklist
DENIED_SENDER_NO_MX	Refused. The domain of your sender address has no mail exchanger (MX).	rejection-text-missing-sender-mx
DENIED_TOO_MANY_RECIPIENTS	Too many recipients. Try the remaining addresses again later.	rejection-text-max-recipients
DENIED_UNQUALIFIED_RECIPIENT	Improper recipient address. Try supplying a domain name.	rejection-text-local-recipient
DENIED_ZERO_RECIPIENTS	Refused. You must specify at least one valid recipient.	rejection-text-zero-recipients
FAILURE_AUTH	Refused. Authentication failed.	rejection-text-auth-failure
FAILURE_TLS	Failed to negotiate TLS connection.	rejection-text-tls-failure
TIMEOUT	Timeout. Talk faster next time.	rejection-text-timeout
UNKNOWN_AUTH	Refused. Unknown authentication method.	rejection-text-auth-unknown

If a policy location URL is given with the `policy-url` option, it will be appended to the end of the message, just in case a human ever reads it. This option should always be used. When a legitimate remote user is incorrectly blocked, the URL should provide your contact information so the error can be corrected.

spamdyke will always append the syslog code to the policy URL so a web browser will jump to an anchor within the

HTML document. Most of the time, the code is prefixed with a # character. For example, if the policy URL is:

```
http://www.example.com/policy.html
```

spamdyke would generate the following URL for a rejection due to a missing reverse DNS entry:

```
http://www.example.com/policy.html#  
DENIED_RDNS_MISSING
```

However, if the policy URL ends in an equals sign (=), spamdyke will assume the URL is for a dynamic page and will not add the # character. For example, if the policy URL is:

```
http://www.example.com/policy?code=
```

spamdyke would generate the following URL for a rejection due to a missing reverse DNS entry:

```
http://www.example.com/policy?code=  
DENIED_RDNS_MISSING
```

## LOGGING ALL DATA

```
full-log-dir
```

spamdyke has the ability to log all SMTP data to files. This is very helpful when debugging but (depending on the mail server traffic levels) it can generate a huge number of files. This feature is activated with the `full-log-dir` option.

Each connection will be logged to a different file in the folder given, with a naming convention that incorporates the current date and time, the process ID of the spamdyke process and a random number:

```
YYYYMMDD_HHMMSS_PID_RANDOMNUM
```

The data from the remote server and qmail are both logged to the file. Each transmission is preceded with a line showing its origin and destination as well as the time and date.

If the remote client establishes a TLS session with qmail and spamdyke passes the encrypted data, the logs will contain the data as hexadecimal bytes.

The logs will contain all debugging output spamdyke can produce, the same output that would be produced if `log-level` were set to its highest level. spamdyke will also print its current configuration into the log file every time it processes a configuration file, to show what options are active and what their values are.

Log files are created with a `0600` mode to protect them from being read by unauthorized users. Please take other precautions to protect them and don't leave them lying around.

**NOTE:** This feature is intended to be used for debugging delivery problems, **not** for monitoring email content. Among other issues, the format of the log files does not make it easy to reconstruct a whole message. If you must monitor your users' email data, please use a packet sniffer on a

separate machine or SMTP proxy software designed for the task.

## PERMISSIONS

```
run-as-user
```

On most Unix-based systems, only the superuser can start a daemon that listens on port 25 (the SMTP port).

However, since running daemons as the superuser should be avoided whenever possible, most qmail installations use utilities to switch the daemon's owner to a normal user before qmail actually runs.

Some older qmail installations lack this ability, notably ones that use the `inetd` "superserver" to start qmail. To help provide security in these situations, spamdyke provides the `run-as-user` option to force spamdyke to switch to a different user account as soon as it starts. All of spamdyke's child processes (including qmail) will run under the new user account.

The value of `run-as-user` takes the following form:

```
USER[:GROUP]
```

`USER` should be the name or user ID of a valid system user. If `GROUP` is not provided, spamdyke will join `USER`'s default group. If `GROUP` is provided, it can be the name or group ID of a valid system group.

On systems where qmail does not run as the superuser, `run-as-user` can be used to force spamdyke to change accounts when the `config-test` option is used, so that its tests of filesystem permissions will be meaningful.

## DNS QUERIES

```
dns-level  
dns-max-retries-primary  
dns-max-retries-total  
dns-server-ip  
dns-server-ip-primary  
dns-timeout-secs
```

Because so many of spamdyke's filters rely so heavily on DNS queries, spamdyke provides a number of options to tune its DNS activities. **Take care when using any of these options, as setting them incorrectly can prevent spamdyke from functioning correctly.** Most of the time, there is no need to use them -- spamdyke will read its DNS information from `/etc/resolv.conf` and the environment, as documented in the system `resolver(5)` manual page.

spamdyke ranks nameservers into two categories: "primary" and "secondary". Primary nameservers are

queried first; secondary nameservers are only queried if no response is received from a primary nameserver.

Normally, spamdyke reads its list of nameservers from `/etc/resolv.conf`, just like any other program. It considers the first nameserver it finds in that file to be a primary nameserver. All others are considered to be secondary nameservers. This list can be overridden, however, using the `dns-server-ip` and `dns-server-ip-primary` options. spamdyke also honors the `port`, `timeout` and `options` directives (if any) in the `/etc/resolv.conf` file. If the environment variable `RES_OPTIONS` is present, spamdyke will parse it for a timeout value.

If either `dns-server-ip` or `dns-server-ip-primary` are used, the file `/etc/resolv.conf` is not loaded. Both options take values in the same format:

`IPADDRESS[:PORT]`

The IP address must be given in dotted-quad format (e.g. `11.22.33.44`). The port number, if it is provided, must be an integer between `1` and `65535`, inclusive. If the port number is not given, the value `53` is used.

`dns-server-ip` and `dns-server-ip-primary` can each be given multiple times to specify multiple servers in each category.

If `dns-server-ip` and `dns-server-ip-primary` are not given and no nameservers can be found by reading `/etc/resolv.conf`, the IP address `127.0.0.1` (the localhost address) is used.

By default, spamdyke will attempt to query its primary nameserver once, then all of its nameservers twice more, taking no more than 30 seconds total. The option `dns-max-retries-total` controls the total number of attempts spamdyke makes to contact its nameservers (the number of query packets actually sent may be much greater than this value; see below). The option `dns-max-retries-primary` controls how long spamdyke sends queries only to its primary nameservers. This value must be less than or equal to `dns-max-retries-total`.

The option `dns-timeout-secs` controls the total time spamdyke will spend waiting for the results of a single query. All of its retries will be carried out within this time limit. **NOTE: The timeout value must be large enough for nameservers to complete their work and respond. If it is too low, spamdyke will never successfully complete a query.**

The `dns-level` option controls how aggressively spamdyke queries its nameservers. It takes the following values:

- `aggressive`: Query all primary nameservers simultaneously until `dns-max-retries-primary` is reached, then query all nameservers simultaneously

until `dns-max-retries-total` is reached. This is the default. For example, if there are 2 primary nameservers, 3 secondary nameservers, `dns-max-retries-primary` is 4 and `dns-max-retries-total` is 6, spamdyke will: send a simultaneous query to each of the 2 primary nameservers 4 times, for a total of 8 queries. It will then send a simultaneous query to all 5 nameservers 2 more times, for a grand total of 18 queries.

- `normal`: Imitate the standard system resolver's behavior by sending queries to one nameserver at a time. spamdyke will send queries to each of the primary nameservers in sequence until `dns-max-retries-primary` is reached, then it will add the secondary nameservers to the sequence and continue sending queries until `dns-max-retries-total` is reached. Depending on the number of nameservers and the values of `dns-max-retries-primary` and `dns-max-retries-total`, some nameservers may never be queried.
- `none`: Do not query any nameservers at all. All DNS-based features will behave as though their queries returned no results.

**NOTE: Depending on the type of query spamdyke is performing, multiple packets are typically sent to each nameserver. For example, when querying a DNS RBL, a "query" consists of 3 packets -- one requesting A records, one requesting TXT records and one requesting CNAME records. In the `aggressive` example above, the 18 queries would result in 52 data packets.**

## FILTER LEVELS

### `filter-level`

spamdyke's overall filter behavior can be controlled with the `filter-level` option. **NOTE: `filter-level` takes precedence over all other filters and features, including authentication and whitelisting.**

The possible values are:

- `allow-all`: Allow all connections, effectively whitelisting everything.
- `normal`: Allow only the connections that pass the enabled filters. This is the default.
- `require-auth`: Allow only authenticated connections, even the ones that match a whitelist entry.
- `reject-all`: Reject all connections, even if they are whitelisted or authenticated.

The values `allow-all` and `reject-all` are most useful within configuration directories, where they can be set e.g. for specific recipients.

## TLS

```
tls-certificate-file
tls-level
tls-privatekey-file
tls-privatekey-password
tls-privatekey-password-file
```

TLS is another name for SSL, the same encryption protocol used by secure websites. TLS can be used during SMTP to provide secure communications between the remote client and the server.

spamdyke supports TLS in several ways. First, with no TLS options given, spamdyke will identify a TLS conversation and simply pass the data back and forth between qmail and the remote client. In this mode, spamdyke cannot read the SMTP data (obviously -- it's encrypted). This prevents some of its filters from functioning, including graylisting, sender and recipient blacklisting, limiting the number of recipients, checking the sender's domain name for an MX record and relaying.

Second, spamdyke can provide TLS itself. To do this, it must be compiled with TLS support, which requires OpenSSL libraries on the server (see the Installation instructions for details). The value of the `tls-level` option controls how TLS is provided:

- `none`: Do not provide or allow TLS, even if qmail supports it. qmail's attempt to advertise its TLS support will be hidden and the remote server's request for TLS will be denied.
- `smtp`: Provide TLS during the SMTP session, so that it can be started if the remote server requests it. spamdyke will decrypt all of the data and pass the plaintext to qmail. qmail will not be aware that TLS is happening. In this mode, qmail does not need to be patched to provide TLS.
- `smtps`: Start an SSL session as soon as the connection is opened. This mode is called "SMTP over SSL" or "SMTPS". The remote client must support this method. Typically, SMTPS is offered on port 465, not port 25.

If `tls-level` is `smtp` or `smtps`, the server certificate must also be provided with the `tls-certificate-file` parameter.

The server certificate file must be in PEM format. If the private key is not in the certificate file, it must be provided in PEM format with the `tls-privatekey-file` parameter.

If the private key is encrypted with a password, the password must be provided with the `tls-privatekey-password` parameter. Because providing the private key password on the command line is very insecure, the password can also be contained in a file and loaded using the `tls-privatekey-password-file` parameter.

Generating self-signed certificates is very easy with OpenSSL. Countless tutorials are available on the web. If there are any problems reading the certificate, the private key or decrypting the private key, spamdyke will log the errors to syslog and fall back to passing the TLS data through to qmail, if qmail has been patched to provide TLS (or spamdyke will send the remote client an error message if qmail doesn't provide TLS). spamdyke will also log the error messages produced by OpenSSL, even though they're rarely helpful.

NOTE: spamdyke does not disable any of its filters simply because a remote client uses TLS or SSL. In SMTP, TLS/SSL is simply a method of securing the communication channel. It is not an authentication method. While it's true spammers aren't using TLS and therefore any client that does use it is unlikely to be a spammer, there's no reason to assume that will be true forever. spamdyke will only disable its filters for clients it finds on its whitelists or ones that use SMTP AUTH.

If in doubt about enabling TLS, do it. Encrypting email data is always a good thing.

## SMTP AUTH

```
hostname
hostname-command
hostname-file
smtp-auth-command
smtp-auth-level
```

SMTP AUTH is a mechanism for remote users to authenticate before sending email (defined in RFC 2554).

This is very handy when users are likely to connect from remote locations (e.g. coffee shops, hotels) and want to bypass relaying restrictions and other filters. spamdyke will disable all of its filters for authenticated connections unless the `filter-level` option overrides this behavior. See Filter Levels for details.

spamdyke supports SMTP AUTH according to the value of the `smtp-auth-level` option:

- **none**: Do not offer or allow SMTP AUTH. If qmail has been patched to provide SMTP AUTH, spamdyke will block its advertisement of SMTP AUTH and will prevent the remote server from authenticating.
- **observe**: If qmail has been patched to provide SMTP AUTH, simply observe the authentication and trust qmail's response. If qmail has not been patched to provide SMTP AUTH, do not provide SMTP AUTH. This is the default.
- **ondemand**: If qmail has been patched to provide SMTP AUTH, simply observe the authentication and trust qmail's response. If qmail has not been patched to

provide SMTP AUTH, offer it and process the authentication without letting qmail know it took place.

- `ondemand-encrypted`: Offer SMTP AUTH as needed, just like `ondemand`, but also offer the encrypted protocol CRAM-MD5. See below for a full description.
- `always`: Always offer and process SMTP AUTH, even if qmail has been patched to provide SMTP AUTH. This value should only be used if qmail's authentication is malfunctioning for some reason.
- `always-encrypted`: Always offer and process SMTP AUTH, just like `always`, but also offer the encrypted protocol CRAM-MD5. See below for a full description.

When the value of `smtp-auth-level` is `ondemand`, `ondemand-encrypted`, `always` or `always-encrypted`, the option `smtp-auth-command` must also be given to provide spamdyke with the command for processing authentication. If multiple commands are available, `smtp-auth-command` may be given multiple times. spamdyke will run each command in order until one of them is successful.

For bare qmail installations where the users are stored in `/etc/passwd` the value of `smtp-auth-command` is usually:

```
/bin/checkpassword /bin/true
```

For vpopmail users, the value of `smtp-auth-command` is usually:

```
/home/vpopmail/bin/vchkpw /bin/true
```

For Plesk users, the values of `smtp-auth-command` are usually:

```
/var/qmail/bin/smtp_auth  
/var/qmail/bin/true  
/var/qmail/bin/cmd5checkpw  
/var/qmail/bin/true
```

Some background on encrypted authentication: SMTP AUTH can be done using one of several protocols, defined by RFC 2554. Two of them are very simple: LOGIN and PLAIN. Although they work slightly differently, LOGIN and PLAIN both send the username and password to the server, which authenticates them. With these protocols, the server does not have to store the user's password in an unencrypted format. If the passwords are encrypted on the server, the authentication process can simply encrypt the password it receives and compare the two encrypted versions. If they match, the password must have been correct.

Unfortunately, this means that the password must be sent to the server in an unencrypted format so the authentication process can encrypt it for comparison. Unless an external encryption protocol is used (e.g. TLS or SMTPS), the passwords are theoretically vulnerable to eavesdropping. For this reason, another protocol was created called CRAM-MD5. CRAM-MD5 is a "challenge/response"

protocol, which allows authentication to take place without sending the password to the server in an unencrypted form. In essence, the protocol starts when the server generates some random text and sends it to the remote client (the "challenge"). The remote client encrypts the text with the user's password and sends it back to the server (the "response"). The server also encrypts the text with the user's password and compares the result to the response from the client. If they match, the remote client must have the correct password.

The challenge text is generated from several sources, including the name of the local server. The `hostname`, `hostname-command` or `hostname-file` option can be used to provide the local server's name for this reason.

When `hostname` is used, the value should be the server's name. When `hostname-command` is used, the value should be a command that will print the server's name as its first line of output. When `hostname-filename` is used, the value should be the path to a file that contains the server's name as its first line. **NOTE: If the local server's name is not provided, CRAM-MD5 will still work and it will still generate random challenge text. Using the local server's name only makes it *slightly* more secure. It's not worth a large effort to provide it.**

Unfortunately, this means that the password must be stored on the server in an unencrypted format so the authentication process can use it to encrypt the challenge text. Theoretically, this leaves the passwords vulnerable to theft by an intruder who breaks into the server.

When `smtp-auth-level` is `ondemand-encryption` or `always-encryption`, spamdyke will offer the CRAM-MD5 protocol to the remote client. However, if the passwords are not stored on the server in an unencrypted format **or** if the authentication command does not understand CRAM-MD5, CRAM-MD5 will always fail.

If you're not sure which value to use, run spamdyke with the `config-test` option. It will test the authentication commands and recommend a value. See Configuration Tests for details.

If in doubt about enabling SMTP AUTH, do it.

Authenticating your users is always a good thing.

## RELAYING

```
access-file
relay-level
```

By default, spamdyke does nothing to prevent relaying (a remote user using the server to send email to another remote user). However, when qmail has not been patched

to provide SMTP AUTH (or when spamdyke's `smtp-auth-level` option is used to prevent qmail from seeing authentication attempts), spamdyke must control relaying so that authenticated users will be allowed to relay.

Whether a recipient is local is determined by searching the local domains list (given by `local-domains-entry` or `local-domains-file`; see Rejecting Senders and Recipients for details).

NOTE: spamdyke does not consider the sender address when deciding to block a recipient for relaying. Sender addresses can be (and usually are) forged by spammers. spamdyke's `relay-level` option controls how spamdyke controls relaying. The available options are:

- `block-all`: Block all attempts at relaying, even if the remote server has authenticated.
- `no-check`: Do not check for or prevent relaying and do not interfere with qmail's relay filter.
- `normal`: Prevent relaying according to the contents of the access file and the list of local domains. Authenticated and whitelisted connections will be allowed to relay. This is the default.
- `allow-all`: Allow all connections to relay messages (create an open relay). This is not recommended.

An access file may be given with the `access-file` option. spamdyke will search the file for the incoming server's IP address and/or rDNS name.

Each line in the access file should use one of the following formats:

```
REMOTE_INFO@REMOTE_IP:ACCESS
REMOTE_INFO@=REMOTE_NAME:ACCESS
REMOTE_IP:ACCESS
REMOTE_NAME:ACCESS
:ACCESS
```

`REMOTE_INFO` is the value returned from an "info" query of the remote server. spamdyke doesn't perform "info" queries but tcpserver does (by default). It sets the `TCPREMOTEINFO` environment variable if it finds anything. spamdyke uses `TCPREMOTEINFO` if it's set. (Does anyone actually use "info" any more?)

`REMOTE_IP` is the IP address of the remote server. It can also be a partial IP address, a dotted quad with ranges, a dotted quad with a number of bits or a dotted quad with a netmask. See IP Address Files for a full explanation of the acceptable formats.

`REMOTE_NAME` is the rDNS name of the remote server. It can also be a partial name. See rDNS Files for a full explanation of the acceptable formats.

`ACCESS` is the permission setting -- either `allow` or `deny`. Connections are allowed by default (if no match is found). If access is denied, no mail is accepted at all, whether relayed or not.

Blank lines and lines starting with # are ignored.

The end of each line may optionally specify a series of environment variables to be set before qmail is started.

They should use the following format and be separated by commas:

```
NAME="VALUE"
```

Confusingly, the double quotes shown above can actually be any character, if the value contains double quotes (escaped values are not supported). For example:

```
NAME=.VALUE.
```

For example, if the remote server's IP address is

```
11.22.33.44
```

 and its rDNS name is

```
mail.example.com
```

, each of the following lines will

match, allow connections and set several environment variables:

```
11.22.33.44:allow,FOOVAR="foovalue",BARVAR=.barvalue.,BAZVAR=-bazvalue-
11.20-
100.33.44:allow,FOOVAR="foovalue",BARVAR=.barvalue.,BAZVAR=-bazvalue-
11.22.:allow,FOOVAR="foovalue",BARVAR=.barvalue.,BAZVAR=-bazvalue-
11.22.33.0/24:allow,FOOVAR="foovalue",BARVAR=.barvalue.,BAZVAR=-bazvalue-
11.22.0.0/255.255.0.0:allow,FOOVAR="foovalue",BARVAR=.barvalue.,BAZVAR=-bazvalue-
=mail.example.com:allow,FOOVAR="foovalue",BARVAR=.barvalue.,BAZVAR=-bazvalue-
=.example.com:allow,FOOVAR="foovalue",BARVAR=.barvalue.,BAZVAR=-bazvalue-
=.com:allow,FOOVAR="foovalue",BARVAR=.barvalue.,BAZVAR=-bazvalue-
:allow,FOOVAR="foovalue",BARVAR=.barvalue.,BAZVAR=-bazvalue-
```

Conveniently, this is exactly the format tcpserver uses for its /etc/tcp.smtp file. NOTE: spamdyke can't read CDB files, only the plain text file, so be careful which file you list on the command line.

Remote servers are allowed to relay if the environment variable RELAYCLIENT is set to any value. Most qmail guides recommend an entry like this one:

```
11.22.33.44:allow,RELAYCLIENT=" "
```

## REVERSE DNS

```
ip-in-rdns-keyword-blacklist-entry
ip-in-rdns-keyword-blacklist-file
ip-in-rdns-keyword-whitelist-entry
ip-in-rdns-keyword-whitelist-file
reject-empty-rdns
```

`reject-ip-in-cc-rdns`  
`reject-unresolvable-rdns`

Reverse DNS is the part of the DNS system that maps IP addresses back to names. **If you don't understand reverse DNS, please please *please* read one of the (thousands of) online tutorials on the subject. Every mail server administrator should know about reverse DNS. It's not complicated. Please take the time to learn it.**

spamdyke does a lot of work with rDNS names. The first and most basic test is to make sure the remote server has an rDNS name, any name. This filter is activated with the `reject-empty-rdns` option. AOL and most other major ISPs use this test. By default, this filter is not run.

The next test is to make sure the remote server's rDNS name resolves. This test only attempts to get at least one IP address from the name. It does not require the rDNS name's IP address to match the remote server's IP address. For example, if the remote server's IP address is `11.22.33.44` and its rDNS name is `mail.example.com`, this test will query `mail.example.com`. Even if it resolves to `66.77.88.99`, the test will pass. Note: The name `localhost` is handled specially. If the rDNS name is `localhost` and the IP address is not `127.0.0.1`, the test fails. This filter is activated with the `reject-unresolvable-rdns` option. By default, this filter is not run.

NOTE: `reject-unresolvable-rdns` does not imply `reject-empty-rdns`. In other words, using just the `reject-unresolvable-rdns` option will block connections from servers with unresolvable rDNS names but it will not block connections from servers with no rDNS names at all. Most users will want to use `reject-empty-rdns` if they use `reject-unresolvable-rdns`.

spamdyke also has several filters that look for an IP address in the rDNS name, since that typically indicates a server sending email that shouldn't be (e.g. a virus-infected Windows machine on a cable modem). If the rDNS name contains the IP address and a keyword given with the `ip-in-rdns-keyword-blacklist-entry` option, spamdyke will block the connection.

Simple keywords can be provided. For example, `dynamic` will match an rDNS name like `11.22.33.44.dynamic.example.com`. NOTE: The keyword will not be matched in the domain name (i.e. the keyword `example` would not be found).

Domain names can be provided if they are preceded by a dot. For example, `.example.com` (or just `.com`) will match `11.22.33.44.dynamic.example.com`.

Complex patterns can also be provided to match multiple keywords in sequence. When the keywords are separated by spaces, spamdyke will only find a match if all of the keywords are found. For example, `cable dynamic`

`.example.com` will match

`11.22.33.44.cable.modem.dynamic.customer.`

`example.com` but it will not match

`11.22.33.44.cable.modem.static.customer.example.com.`

All keyword searches are case-insensitive.

If more than a few keywords are provided, the `ip-in-rdns-keyword-blacklist-file` option is much more efficient. The keywords (or patterns) should be listed one per line. Blank lines and lines beginning with `#` will be ignored.

spamdyke also accepts the `ip-in-keyword-whitelist-entry` and `ip-in-keyword-whitelist-file` options that match keywords exactly the same way as the blacklist options. The whitelist options bypass all filters when a connection matches, however, instead of blocking the connection.

spamdyke offers one other option that searches for an IP address in the rDNS name: `reject-ip-in-cc-rdns`. In essence, this option is the same as using `ip-in-rdns-keyword-blacklist-entry` for every two-letter country code (e.g. `.us`). This option is not very useful for large servers or servers outside the United States. As ICANN begins allowing arbitrary top-level domains, two-letter domains will no longer be a reliable way of detecting non-US servers.

When matching an IP address in an rDNS name, spamdyke looks for the IP address in many forms; for example, if the IP address is `11.22.33.44`, spamdyke will look for the following patterns in the rDNS name (the dots in the examples below can be any single character):

`11.22.33.44`

`011.022.033.044`

`11.022.033.044`

`11.22.033.044`

`11.22.33.044`

`44.33.22.11`

`44.11.22.33`

`33.22.11.44`

`44.33.1122`

`3344.11.22`

`11.22.8492` (last two octets combined and converted to an integer)

`11223344`

`11.22.3344`

`11.223344`

`011022033044`

`11022033044`

`1122033044`

112233044  
44332211  
044033022011  
3930621781 (entire address converted to an integer)  
5080d7e3 (each octet converted to hexadecimal)

## BLACKLISTS

```
ip-blacklist-entry  
ip-blacklist-file  
rdns-blacklist-dir  
rdns-blacklist-entry  
rdns-blacklist-file
```

First, let's all agree that blacklists are evil, arbitrary, unforgiving, unfair, etc, etc, blah blah blah. OK. If you feel that way, don't use one.

**But**, if you decide to use one, it can block over half the spam you would otherwise receive. Blacklists are very effective against professional spammers who buy thousands of domain names and run their own mail servers (constantly moving them around, of course). It's your decision.

NOTE: Constructing and maintaining a blacklist is left as an exercise for the reader. spamdyke will use a blacklist but it won't help you build one.

spamdyke will search for the remote server's rDNS name in a file and block the connection if it is found. This is activated with the `rdns-blacklist-file` option. See rDNS Files for details on the format of this file. If the `rdns-blacklist-file` option is given multiple times, each file will be checked before the connection is allowed.

If only a few rDNS names should be blacklisted, the option `rdns-blacklist-entry` can be used instead. The entries should follow the same format as the lines in an rDNS file.

spamdyke will also search for the remote server's rDNS name in a directory structure and block the connection if it is found. For large lists of rDNS names (more than 100), the directory structure is much faster than a file. This is activated with the `rdns-blacklist-dir` option. See rDNS Directories for details on the structure of this directory. If the `rdns-blacklist-dir` option is given multiple times, each directory will be checked before the connection is allowed.

spamdyke will also search for the remote server's IP address in a file and block the connection if it is found. This is activated with the `ip-blacklist-file` option. See IP Address Files for details on the format of this file. If the `ip-blacklist-file` option is given multiple times, each

blacklist file will be checked before the connection is allowed.

If only a few IP addresses should be blacklisted, the option `ip-blacklist-entry` can be used instead. The entries should follow the same format as the lines in an IP address file.

## DNS RBLs

```
dns-blacklist-entry  
dns-blacklist-file
```

DNS Realtime Blackhole Lists are services maintained by third parties that list IP addresses (presumably the IP addresses of spammers). The criteria for getting listed on a DNS RBL vary by organization and it's often very hard to get delisted. Sometimes, listings are politically motivated; some DNS RBL operators try to force large ISPs to cancel spammers' accounts by listing all of the ISP's IP addresses, innocent or guilty. If you choose to use a DNS RBL, please do some preliminary research to understand its policies and history of complaints.

spamdyke will reject connections from an IP address listed in a given DNS RBL. This feature is activated with the `dns-blacklist-entry` option. By default, spamdyke does not use a DNS RBL. If the `dns-blacklist-entry` option is given multiple times, each DNS RBL will be checked before the connection is allowed.

If more than a few DNS RBLs are needed, the `dns-blacklist-file` option is more efficient than `dns-blacklist-entry`. Each line in the file should contain the name of one DNS RBL.

**NOTE: Checking DNS RBLs can impose a serious performance penalty. Using more than three DNS RBLs is not recommended.**

## DNS RHSBLs

```
rhs-blacklist-entry  
rhs-blacklist-file
```

DNS Righthand-side Blacklists are services maintained by third parties that list domain names (presumably the domain names of spammers). The criteria for getting listed on a DNS RHSBL vary by organization and it's often very hard to get delisted. Sometimes, listing are politically motivated; some DNS RHSBL operators try to force large ISPs to cancel spammers' accounts by listing all of the ISP's domain names, innocent or guilty. If you choose to

use a DNS RHSBL, please do some preliminary research to understand their policies and history of complaints. spamdyke will reject connections from servers whose reverse DNS names are listed in a given DNS RHSBL. spamdyke will also reject connections from senders whose email domain names are listed in a given DNS RHSBL. These features are activated with the `rhs-blacklist-entry` option. By default, spamdyke does not use a DNS RHSBL. If the `rhs-blacklist-entry` option is given multiple times, each DNS RHSBL will be checked before the connection is allowed.

If more than a few DNS RHSBLs are needed, the `rhs-blacklist-file` option is more efficient than `rhs-blacklist-entry`. Each line in the file should contain the name of one DNS RHSBL.

**NOTE: Checking DNS RHSBLs can impose a serious performance penalty. Using more than three DNS RHSBLs is not recommended.**

## WHITELISTS

```
ip-whitelist-entry
ip-whitelist-file
rdns-whitelist-dir
rdns-whitelist-entry
rdns-whitelist-file
```

spamdyke will match the remote server's rDNS name against a list given with the `rdns-whitelist-entry` option and skip all filters if it is found. If more than a few entries are given, the `rdns-whitelist-file` option is much more efficient; it can be used to put the entries in a file, one per line. See rDNS Files for details on the format of this file. If `rdns-whitelist-file` option is given multiple times, each whitelist file will be checked before the connection is blocked.

spamdyke will also search for the remote server's rDNS name in a directory structure and skip all filters if it is found. This is activated with the `rdns-whitelist-dir` option. See rDNS Directories for details on the format of this directory structure. If `rdns-whitelist-dir` option is given multiple times, each directory will be checked before the connection is blocked.

spamdyke will also search for the remote server's IP address in a list of entries skip all filters if it is found. This is activated with the `ip-whitelist-entry` option. If more than a few entries are given, the `ip-whitelist-file` option is more efficient; it can be used to put the entries in a file. See IP Address Files for details on the format of this file. If the `ip-whitelist-file` option is given multiple

times, each whitelist file will be checked before the connection is blocked.

## REJECTING SENDERS AND RECIPIENTS

```
local-domains-entry
local-domains-file
recipient-blacklist-entry
recipient-blacklist-file
reject-missing-sender-mx
sender-blacklist-entry
sender-blacklist-file
```

It's rare, but sometimes spammers will choose to target a specific address and pound it with millions of messages. Most often, this happens when a spammer chooses to use one of your email addresses as the sender address on a spam run. When that happens, you'll receive bounce messages for all of their spams. (This is commonly referred to as a "joe job".)

spamdyke will block all incoming messages to a specific address with the `recipient-blacklist-entry` option. If more than a few addresses are given, the `recipient-blacklist-file` option is much more efficient. The given file must contain one email address per line. Blank lines and lines beginning with `#` are ignored. If the `recipient-blacklist-file` option is given multiple times, each blacklist file will be checked before the connection is blocked.

One form of wildcard address is supported. All usernames within a domain (and its subdomains) may be blocked by a line starting with `@`. For example, if the file contained the following entry:

```
@example.com
```

spamdyke will block mail to `fred@example.com`, `fred@mail.example.com`, `barney@mail.internal.example.com`, etc.

Similarly, spammers will rarely (but occasionally) use the same sender address for a while. spamdyke will block all incoming messages from a specific address with the `sender-blacklist-entry` option. If more than a few addresses are given, the `sender-blacklist-file` option is more efficient. The file format is identical to the recipient blacklist file, described above.

spamdyke will also look at the sender's domain name to check if it has a mail exchanger or an IP address listed in DNS. If a mail exchanger is found, the mail exchanger must have an IP address. Without a valid mail exchanger or an IP address, no mail could possibly go to the sender address. This test tends to block a lot of mail from compromised web servers that aren't supposed to be

sending email. It is activated with the `reject-missing-sender-mx` option. By default, this test is not run.

To use the `reject-missing-sender-mx` option, a list of local domain names must be given. This is necessary because if your server is willing to receive mail for a given domain, it must be a mail exchanger for that domain, no matter what DNS says.

The local domain list should be provided with the `local-domains-file` option. The file should contain one domain name per line. For example, if the file contained `example.com`, a message from `fred@example.com` would not be blocked by `reject-missing-sender-mx` (although the message may still be blocked by other filters). If the `local-domains-file` option is given multiple times, each file will be checked before any action is taken.

The list of local domains for spamdyke should be the same as the one used by qmail. For this reason, the `local-domains-file` option should almost always be used.

However, in some situations, it may be necessary to give additional local domains to spamdyke that aren't listed in a file. In that case, the `local-domains-entry` option can be used.

The local domain list also supports wildcards. If a domain name in the file starts with a dot, any sender address within a subdomain will also pass the `reject-missing-sender-mx` filter. For example, if the provided file contained `.example.com`, then `fred@foo.example.com`, `barney@bar.example.com` and `wilma@example.com` would all pass the `reject-missing-sender-mx` filter, no matter what DNS records exist for their domain.

Conveniently, this wildcard system matches the system used in qmail's `controls/rcpthosts` file.

## DNS WHITELISTS

```
dns-whitelist-entry
dns-whitelist-file
rhs-whitelist-entry
rhs-whitelist-file
```

spamdyke has the ability to consult DNS whitelists and allow connections from hosts or senders who match entries on them. DNS whitelists are essentially DNS RBLs and DNS RHSBLs that list allowed IP addresses and domain names instead of blocked ones. All of the same cautionary statements apply to DNS whitelists as to DNS blacklists. See DNS RBLs and DNS RHSBLs for details.

To use a DNS Realtime Whitelist (the opposite of a DNS RBL), the option `dns-whitelist-entry` should be given. To use a DNS Righthand-side Whitelist, the option `rhs-whitelist-entry` should be given. By default, spamdyke does not use a DNS whitelist. If either option is given multiple times, each list will be consulted before the connection is blocked.

If more than a few lists are given, the `dns-whitelist-file` or `rhs-whitelist-file` options may be used to provide the lists in files.

**NOTE: Checking DNS whitelists can impose a serious performance penalty. Using more than three DNS whitelists is not recommended.**

## WHITELISTING SENDERS AND RECIPIENTS

```
recipient-whitelist-entry
recipient-whitelist-file
sender-whitelist-entry
sender-whitelist-file
```

Sometimes, adding IP addresses and reverse DNS names to whitelist files is not enough to satisfy some users. Either they continue to receive mail from unexpected places or they just think spamdyke is blocking their email. In those cases, a last resort can be to whitelist the sender or recipient address.

**NOTE: Using these features is a bad idea!** Sender addresses are very easy to forge; this is why spam is so hard to block. Recipient addresses are obviously already known to the spammers; this is why spam is delivered. Whitelisting any addresses this way will allow spam to get through.

To whitelist a sender or recipient address, the `sender-whitelist-entry` or `recipient-whitelist-entry` option should be used, respectively. The entries can use the same formats as those for `sender-blacklist-entry` and `recipient-blacklist-entry`; see Rejecting Senders and Recipients for details.

Whitelist entries can be placed in files and referenced with `sender-whitelist-file` or `recipient-whitelist-file`. This is more efficient for more than a few entries.

## GRAYLISTING / GREYLISTING

```
graylist-dir
graylist-exception-ip-entry
graylist-exception-ip-file
graylist-exception-rdns-dir
```

```
graylist-exception-rdns-entry
graylist-exception-rdns-file
graylist-level
graylist-max-secs
graylist-min-secs
```

Graylisting is the technique of denying mail delivery the first time a sender tries to deliver to a recipient. The next time the remote server attempts to deliver the message, it is accepted. All future messages from the sender to the recipient are also allowed.

Graylisting works because spammers don't use real mail servers, so when the initial delivery fails, they don't retry it. Even if they do, they change to a different random sender address, which is also graylisted.

spamdyke's graylisting is activated first with the `graylist-level` option, which will accept the following values:

- `none`: Do not graylist any connections. This is the default.
- `always`: Always graylist connections if a domain folder exists for the recipient and the remote server's IP address or rDNS name are not found on one of the exception lists.
- `always-create-dir`: Always graylist connections unless the remote server's IP address or rDNS name are found on one of the exception lists. If a domain folder does not exist for the recipient, create it.
- `only`: Do not graylist any connections unless a domain folder exists for the recipient and the remote server's IP address or rDNS name are found on one of the exception lists.
- `only-create-dir`: Do not graylist any connections unless the remote server's IP address or rDNS name are found on one of the exception lists. If a domain folder does not exist for the recipient, create it.

The option `graylist-dir` is required if `graylist-level` is not `none`. The value of `graylist-dir` must be the path to a directory where spamdyke has permission to create folders and write files.

A list of local domains must also be provided with `local-domains-entry` or `local-domains-file`. Without a list of local domains, spamdyke cannot know which domains should be graylisted. See Rejecting Senders and Recipients for details.

The "domain directories" required by `graylist-level`'s `always` and `only` values are just folders named using the recipient's domain name. For example, if the value of `graylist-dir` is:

```
/var/tmp/graylist.d
```

and the recipient's domain is `example.com`, the domain directory would be:

```
/var/tmp/graylist.d/example.com
```

Domain directories allow graylisting to be selectively activated for only some domains. If all domains should be

graylisted, use `always-create-dir` or `only-create-dir` instead.

NOTE: Domain directories must be named in lowercase.

Unfortunately, spamdyke cannot perform a case insensitive search for directory paths.

If `graylist-dir` is given multiple times, spamdyke will search each given directory for the recipient's domain folder. If the domain folder is not found and the value of `graylist-level` is `always-create-dir` or `only-create-dir`, spamdyke will create a domain folder in the last directory given.

As spamdyke creates graylist entries, it breaks apart the sender's and recipient's addresses to create folders. For example, a graylist entry for messages from `sender@remote.com` to `recipient@local.com` will be a file with the following path:

```
.../local.com/recipient/remote.com/  
sender@remote.com
```

NOTE: This directory structure is different from the one created by spamdyke prior to version 4.0.0. If spamdyke encounters an old-style directory structure, it will silently convert the entries to the new structure as they are used.

If the options `graylist-exception-ip-entry` or `graylist-exception-rdns-entry` are given, spamdyke will reverse its graylist policy for remote servers whose IP addresses or rDNS names, respectively, match the given entries.

If more than a few IP address or rDNS entries are given, the options `graylist-exception-ip-file` and `graylist-exception-rdns-file` are more efficient. See IP Address Files and rDNS Files for details on the formats of these files.

If more than 100 entries are given in an rDNS file, the option `graylist-exception-rdns-dir` is more efficient than `graylist-exception-rdns-file`. See rDNS Directories for details.

spamdyke accepts two options to control the lifetimes of its graylist entries. The option `graylist-min-secs` controls the minimum amount of time an entry must exist before it is valid. For example, if the value `300` is given, a graylist entry is not valid until it has existed for at least 5 minutes. This prevents a spammer from attempting delivery twice in rapid succession.

The option `graylist-max-secs` controls the maximum amount of time an entry will remain valid after it is created. For example, if the value `604800` is given, a graylist entry will become invalid after 7 days. If the sender attempts to deliver a message after the entry has expired, they will be graylisted again. NOTE: A graylist entry's expiration date is reset each time a message passes the filter. If the maximum age is 2 weeks and the sender sends a message

every day, their entry will never expire because it is continually reset.

As of 2008, not many ISPs are using graylisting yet, so it's still very effective. However, it won't be difficult for spammers to circumvent it (once enough ISPs have implemented it), so its effectiveness won't last forever.

## EARLYTALKERS

### `greeting-delay-secs`

Several years back, someone noticed that some spam software didn't follow the SMTP protocol. Specifically, spammers were opening connections to remote mail servers and pushing out all of the mail commands as quickly as they could, without waiting for (or checking) responses. By delaying the sending of the greeting banner, it's possible to catch these spammers and block them. spamdyke will delay sending the opening greeting in order to wait for the sender to send data. If that happens, spamdyke will block the connection. This is activated with the `greeting-delay-secs` option.

Unfortunately, as of 2007, spammers have mostly updated their software so this trick doesn't work any more. It still catches a few of them occasionally though.

Even if it doesn't catch the spammers any more, this is still a good test to run because it slows down the rate at which spammers can send email. Making it more expensive for them is not a bad thing.

## LIMITING NUMBERS OF RECIPIENTS

### `max-recipients`

RFC 821 says an SMTP server must allow the remote server to specify as many recipients as the SMTP server has memory to hold. This allows spammers to send a single message to (potentially) thousands of users in one connection.

spamdyke offers a way to violate the RFC by limiting the number of recipients per message. Once that number has been reached, the remaining recipients will be rejected. Legitimate mail servers use the multi-recipient feature as well, but they will retry their deliveries. Spammers generally don't.

This feature is activated by the `max-recipients` option. By default, spamdyke allows unlimited recipients per connection.

If you enable this feature, make sure all of your users are using SMTP AUTH or are allowed to relay or are

whitelisted. Otherwise, their MUAs will show them error messages when they try to send to large address book mailing lists and they will complain. Loudly.

## TIMEOUTS

```
connection-timeout-secs
idle-timeout-secs
```

Most spam software is pretty stupid and doesn't handle error codes from SMTP servers. Instead, they send their commands and wait for specific responses. When those responses don't come, the software just sits forever and waits.

spamdyske will time out these connections in one of two ways. First, an absolute time limit can be imposed on a connection. This is activated with the `connection-timeout-secs` option. If this feature is enabled, it should be set to a very high value. Large (legitimate) messages can take a very long time to deliver, especially if the link is slow. A value of `0` disables this feature. By default, this feature is not activated.

Second, an idle time limit can be imposed on a connection. This is activated with the `idle-timeout-secs` option. If no data is received within the given number of seconds, the connection is closed. A value of `0` disables this feature. By default, this feature is not activated.

## EXTRA UTILITIES

There are some additional programs included with spamdyke in the `utils` folder. They are:

`domain2path`

A utility for constructing rDNS folder paths from domain names. Useful in scripts. See rDNS Directories for details on `domain2path`.

`domainsplit`

A utility for finding the base domain name of a fully qualified domain name. For example, given `foo.bar.baz.example.com`, `domainsplit` will return `example.com`. Also useful in scripts.

`dnscat`

An example program that demonstrates finding DNS A records (IP addresses) for fully qualified domain names.

`dnscatany`

An example program that demonstrates finding DNS records of "any" type (which doesn't actually return all types) for fully qualified domain names.

`dnscatany_libc`

An example program that demonstrates finding DNS records of "any" type (which doesn't actually return all types) for fully qualified domain names.

This version uses the system resolver library instead of sending UDP packets itself.

dnscname

An example program that demonstrates finding DNS CNAME records (alias names) for fully qualified domain names.

dnsmx

An example program that demonstrates finding DNS MX records (mail exchangers) for domain names.

dnsns

An example program that demonstrates finding DNS NS records (nameservers) for domain names.

dnsptr

An example program that demonstrates finding DNS PTR records (reverse DNS names) for fully qualified domain names or IP addresses.

dnssoa

An example program that demonstrates finding DNS SOA records (start-of-authority) for domain names.

dnstxt

An example program that demonstrates finding DNS TXT records (text information) for fully qualified domain names.

None of these utilities depend on being installed in any specific folder. None of them depend on the presence of the others or spamdyke. **spamdyke does not require them or use them.**