

Mail Policy Overview

Christian Külker

2023-01-15

Contents

1	Open Source Policy Daemons	2
2	Greylistd	2
2.1	Links	2
3	MT Policy Daemon	4
3.1	Installation	4
3.2	Links	4
3.3	Plugins	4
4	policyd-rate-limit	6
4.1	Links	6
5	Policy Weight Daemon	6
5.1	Function	6
5.2	Features	7
5.3	Discussion	7
5.4	Further Read	7
5.5	Links	7
6	postfix-policyd-spf-perl	8
6.1	Links	8
7	postfix-policyd-spf-python	8
7.1	Packages	8
7.2	Links	8
8	postgrey	8
8.1	Discussion	9

8.2 Links	9
9 History	9
10 Disclaimer of Warranty	9
11 Limitation of Liability	10

This overview is not complete. It started on 2023-01-13 as a summary document for Postfix policy daemons to understand the current state of policy daemons, as some of the well-known policy daemons mentioned in the literature are no longer maintained.

The Postfix mail server can internally block or accept mail in a number of ways. In cases where this is not sufficient, a Postfix `smtpd` policy (see [SMTPD_POLICY_README](#)) can be used to add functionality to Postfix. An example of this is Postfix [greylisting](#).

1 Open Source Policy Daemons

Name	Server	License	Deb	Lang	Updated	Note
greylistd	Exim	GPL2	yes	Python	2020	Greylist
Mail::MtPolicyd	Postfix	GPL2	yes	Perl	2020	
policyd-rate-limit	Postfix	GPL3	yes	Python	2022	Rate
Policy Weight	Postfix	GPL2	yes	Perl	2009	MX, HALO
postfix-policyd-spf-perl	Postfix	GPL2	yes	Perl	2018	SPF
postfix-policyd-spf-python	Postfix	GPL2, Apache 2.0	yes	Python	2022	SPF
postgrey	Postfix	GPL2	yes	Perl	2022	Greylist

2 Greylistd

The `postgreyd` daemon provides simple `greylisting` services for `Exim` version 4.

2.1 Links

- [greylistd source code](#)
- [greylistd package](#)
- [greylistd developer information](#)
- [greylistd publicity](#)

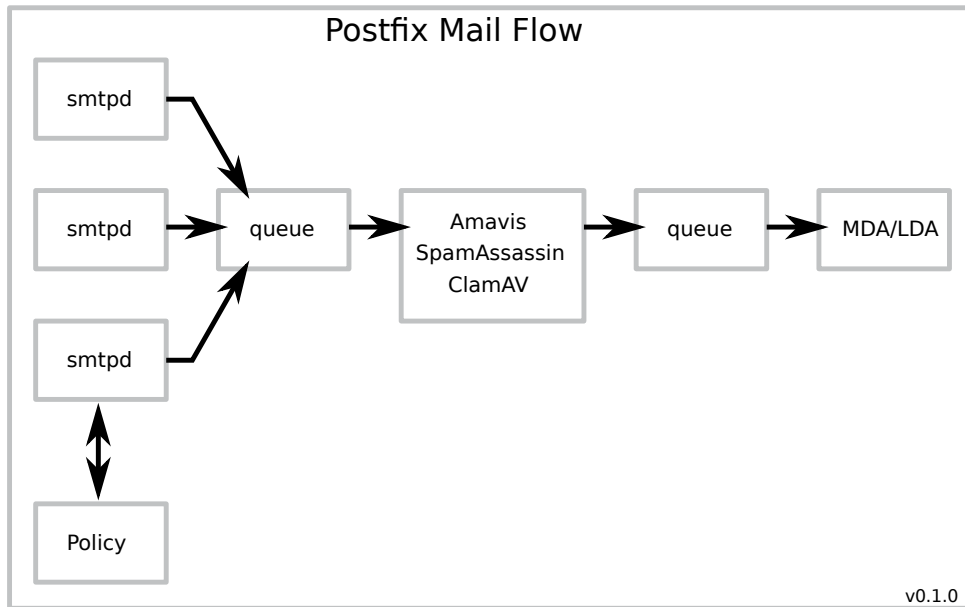


Figure 1: Postfix Mail Flow

3 MT Policy Daemon

A modular policy daemon for postfix.

3.1 Installation

Mail::MtPolicyd supports 3 connection:

1. SQL
2. Memcached
3. LDAP

```
aptitude install memcached mtpolicyd
```

3.2 Links

- [Mail::MtPolicyd](#)
- [Mail::MtPolicyd Cookbook](#)
- [Mail::MtPolicyd Debian package](#)
- [Mail::MtPolicyd source code](#)
- [Mail::MtPolicyd Debian popularity](#)

3.3 Plugins

Plugin	Note
Accounting	accounting in sql tables
Action	returns an action
AddScoreHeader	adding the score as header to the mail
ClearFields	unset session variables
Condition	conditions based on session values
CtlpRep	the Commtouch IP reputation service (ctipd)
DBL	checking helo, sender domain, rdns against an DBL
Eval	capture the output of plugins
Fail2Ban	block an address with fail2ban
GeoIPAction	checking geo information of an ip
GeoIPLookup	checking geo information of a client_address
Greylist	greylisting mechanism with an auto whitelist

Plugin	Note
Greylist::AWL::Base	base class for grelisting AWL storage backends
Greylist::AWL::Redis	backend for redis greylisting awl storage
Greylist::AWL::Sql	backend for SQL greylisting awl storage
Greylist::Ticket::Base	base class for greylisting ticket storage backends
Greylist::Ticket::Memcached	greylisting ticket storage backend for memcached
Greylist::Ticket::Redis	greylisting ticket storage backend for redis
Honeypot	creating an honeypot
LdapUserConfig	retrieving per user configuration from LDAP
PostfixMap	accessing a postfix access map
Proxy	forward request to another policy daemon
Quota	accounting in sql tables
RBL	checking the client-address against an RBL
RBLAction	checking the client-address against an RBL
RegexList	regex matching
Result	result returned by a plugin
Role::ConfigurableFields	role for plugins using configurable fields
Role::PluginChain	role for plugins to support a nested plugin chain
Role::Scoring	role for plugins using scoring
Role::SqlUtils	role with support function for plugins using sql
Role::UserConfig	role for plugins using per user/request configuration
SMTPVerify	remote SMTP address checks
SPF	apply SPF checks
SaAwlAction	checking spamassassin AWL reputation
SaAwlLookup	querying a spamassassin AWL database for reputation
ScoreAction	running an action based on the score
SetField	sets and key=value in the session
SqlList	accessing a SQL white/black/access list
SqlUserConfig	retrieving the user config of a user
Stress	postfix stress mode

4 policyd-rate-limit

Postfix `policyd` server, which allows you to limit the number of mails accepted by Postfix over multiple time periods, by `SASL` usernames and/or `IP` addresses, to limit the number of mails a user can send.

4.1 Links

- [policyd-rate-limit](#)
- [policyd-rate-limit Debian package](#)
- [policyd-rate-limit source code](#)
- [policyd-rate-limit Debian popularity](#)

5 Policy Weight Daemon

The Policy Weight Daemon is a well-known policy daemon for Postfix. It is also mentioned in the German “Das Postfix Buch” and the [Policy Weight Debian popularity](#) was high between 2010 and 2016. From the [Policy Weight unofficial source code](#) repository it can be understood that development was probably stopped around 2013 (2018-5) or earlier and the last stable version and to-do list is from 2009. The location of the [Policy Weight source code](#) repository is unknown as of 2023-01-13. The [Policy Weight home](#) declares itself as “Project discontinued”.

As of the [Policy Weight Debian package](#) page, there are some known Debian package management tasks (bugs), but no major software bugs as of 2023-01-13. However, without an upstream bug tracker, this is difficult to know. From the [Policy Weight to do list todo.txt](#) of the [Policy Weight unofficial source code](#) it is clear that some important `IPv6` features are missing.

The [Policy Weight old mailing list archive](#) is available until 2010. The [Policy Weight new mailing list](#) location contains the [Policy Weight new mailing list archive](#) up to 2016.

5.1 Function

The policy weight daemon is used by Postfix `smtpd` before accepting the mail body `DATA`. Several checks are performed by the daemon, using caching to minimize queries. Individual checks are weighted (hence the name) with a score (which can be changed by the administrator), which is aggregated. If the sum is above a definable threshold, the mail is rejected before `DATA` is fetched, saving bandwidth.

5.2 Features

- DNSBL
- RHSBL
- MX (checks of DNS MX records)
- A DNS (checks of DNS A records)
- HELO smtp

5.3 Discussion

The main advantages of policy daemons are: rejecting mail before delivery and saving bandwidth. This is in contrast to SpamAssassin, Amavis and others that operate after the mail DATA has been fetched. Policy Weight does not add delays like greylisting would. It performs rejection of fake HELO or sender information with better accuracy and fewer false positives than Sender Policy Framework (SPF).

Some of the functionality may be covered by newer versions of Postfix, version 2.8 and above. For example, the [POSTSCREEN_README](#) gives information about testing MX (MX policy test) and other DNS information. While policy weight works **after** smtpd, [postscreen](#) works **before** smtpd, which can save even more bandwidth.

5.4 Further Read

- <https://doku.fietz.net/index.php/Policyd-weight>
- https://www.heinlein-support.de/upload/mk4/2-08_policyd-weight.pdf
- https://dokuwiki.nausch.org/doku.php/centos:mail_c6:spam_2

5.5 Links

- [Policy Weight home](#)
- [Policy Weight Debian package](#)
- [Policy Weight source code](#) is unknown
- [\[Policy Weight unofficially source code\]](#)
- [Policy Weight Debian popularity](#)
- [Policy Weight to do list](#)
- [Policy Weight new mailing list](#)
- [Policy Weight new mailing list archive](#)
- [Policy Weight old mailing list archive](#)
- [Policy Weight Wikipedia](#)

6 postfix-policyd-spf-perl

Simple Postfix policy server for `RFC 4408/7208` `SPF` checking.

6.1 Links

- [postfix-policyd-spf-perl](#)
- [postfix-policyd-spf-perl popularity](#)
- [postfix-policyd-spf-perl Debian developer information](#)
- [postfix-policyd-spf-perl Debian package](#)
- [postfix-policyd-spf-perl source code](#)

7 postfix-policyd-spf-python

Postfix policy server and `sendmail` milter called `spf-engine` for `SPF` checking.

7.1 Packages

- `postfix-policyd-spf-python`
- `pyspf-milter`
- `python3-spf-engine`

7.2 Links

- [postfix-policyd-spf-python](#)
- [postfix-policyd-spf-python popularity](#)
- [postfix-policyd-spf-python Debian developer information](#)
- [postfix-policyd-spf-python Debian package](#)
- [postfix-policyd-spf-python source code](#)

8 postgrey

[Postgrey](#) is a Postfix policy server that provides greylisting. When a delivery request of an email is received by Postfix via `SMTP` (`smtpd`), three data points, the client IP, the sender, and the recipient, are collected and stored if this is the first time these data points have been seen. An email is rejected if it is seen for the first time according to these three data points, or if the time elapsed since the first encounter is less than 5 minutes (or another configurable value). The rejection is reported to the sender in the form of a temporary error. Most valid senders will resend later, as required by `RFC`. Many `SPAM` hosts

will not do this, as tracking the send status of mails is usually not worth the resources, as these mailers send millions of mails.

8.1 Discussion

Greylisting is an effective way to reduce SPAM. Around 2010 to 2017, most non-SPAM hosts that encountered a greylisting host behaved according to the [RFC](#) and very rarely email was not delivered. However, since 2017 there is a trend where more and more companies are moving dedicated mail servers to centralized cloud services that do not handle greylisting well. This can be seen when a new account is set up and the email address needs to be verified. These cloud-based verification emails are often not resent, resulting in an inability to log in unless the cloud provider's DNS is accepted. If the mail server has only one user, the user can temporarily disable greylisting when signing up for a new service. If the mail server has many users, it is not possible to accept-list or disable greylisting. Once an email address validation email is sent, it is too late to turn off greylisting because that attempt will not be successful.

8.2 Links

- [postgrey](#)
- [postgrey mailing list](#)
- [postgrey mailing list archive](#)
- [postgrey source code](#)
- [postgrey package](#)
- [postgrey developer information](#)
- [postgrey publicity](#)

9 History

Version	Date	Notes
0.1.2	2023-01-15	Improve grammar, postgrey greylstd
0.1.1	2023-01-14	mtpolicy policyd-rate-limit postfix-policyd-spf-perl
0.1.0	2023-01-13	Initial release Policy Weight

10 Disclaimer of Warranty

THERE IS NO WARRANTY FOR THIS INFORMATION, DOCUMENTS AND PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE INFORMATION, DOC-

UMENT OR THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE INFORMATION, DOCUMENTS AND PROGRAMS IS WITH YOU. SHOULD THE INFORMATION, DOCUMENTS OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

11 Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.