modoboa-amavis Documentation

Release 1.1.3

Antoine Nguyen

Jul 20, 2017

Contents

This plugin provides a simple management frontend for amavisd-new. The supported features are:

- SQL quarantine management: available to administrators or users, possibility to delete or release messages
- Per domain customization (using policies): specify how amavisd-new will handle traffic
- Manual training of SpamAssassin using quarantine's content

Note: The per-domain policies feature only works for new installations. Currently, you can't use modoboa with an existing database (ie. with data in `users` and `policies` tables).

Note: This plugin requires amavisd-new version **2.7.0** or higher. If you're planning to use the *Self-service mode*, you'll need version **2.8.0**.

Contents:

CHAPTER 1

Install

Install this extension system-wide or inside a virtual environment by running the following command:

```
$ pip install modoboa-amavis
```

Edit the settings.py file of your modoboa instance and add `modoboa_amavis` inside the `MODOBOA_APPS` variable like this:

```
MODOBOA_APPS = (
    'modoboa',
    'modoboa.core',
    'modoboa.lib',
    'modoboa.admin',
    'modoboa.relaydomains',
    'modoboa.limits',
    'modoboa.parameters',
    # Extensions here
    # ...
    'modoboa_amavis',
)
```

Then, add the following at the end of the file:

```
from modoboa_amavis.settings import *
```

Run the following commands to setup the database tables:

```
$ cd <modoboa_instance_dir>
$ python manage.py migrate
$ python manage.py collectstatic
$ python manage.py load_initial_data
```

Finally, restart the python process running modoboa (uwsgi, gunicorn, apache, whatever).

CHAPTER 2

Setup

Quick Amavis setup

By default, amavis doesn't use a database. To configure this behaviour, you first need to create a dedicated database. This step is a bit manual since no *ready-to-use* SQL schema is provided by amavis. The information is located inside README files, one for MySQL and one for PostgreSQL.

Then, you must tell amavis to use this database for lookups and quarantined messages storing. Here is a working configuration sample:

```
@lookup_sql_dsn =
  (['DBI:<driver>:database=<database>;host=<dbhost>;port=<dbport>', '<dbuser>', '
↪<password>]']);

@storage_sql_dsn =
  (['DBI:<driver>:database=<database>;host=<dbhost>;port=<dbport>', '<dbuser>', '
↪<password>]']);

# PostgreSQL users NEED this parameter!
# MySQL users only need this parameter is email addresses are stored
# using the VARBINARY type.
$sql_allow_8bit_address = 1;

$virus_quarantine_method = 'sql:';
$spam_quarantine_method = 'sql:';
$banned_files_quarantine_method = 'sql:';
$bad_header_quarantine_method = 'sql:';

$virus_quarantine_to = 'virus-quarantine';
$banned_quarantine_to = 'banned-quarantine';
$bad_header_quarantine_to = 'bad-header-quarantine';
$spam_quarantine_to = 'spam-quarantine';
```

Replace values between <> by yours. To know how to configure amavis to allow quarantined messages release, read this *section*.

Note: Amavis configuration allows for separate lookup and storage databases but Modoboa doesn't support it yet.

Connect Modoboa and Amavis

You must tell to Modoboa where it can find the amavis database. Inside `settings.py`, add a new connection to the `DATABASES` variable like this:

```
DATABASES = {
  # Stuff before
  #
  "amavis": {
    "ENGINE" : "<your value>",
    "HOST" : "<your value>",
    "NAME" : "<your value>",
    "USER" : "<your value>",
    "PASSWORD" : "<your value>"
  }
}
```

Replace values between <> with yours.

Cleanup

Storing quarantined messages to a database can quickly become a perfomance killer. Modoboa provides a simple script to periodically purge the quarantine database. To use it, add the following line inside root's crontab:

```
0 0 * * * <modoboa_site>/manage.py qcleanup
#
# Or like this if you use a virtual environment:
# 0 0 * * * <virtualenv path/bin/python> <modoboa_site>/manage.py qcleanup
```

Replace `modoboa_site` with the path of your Modoboa instance.

By default, messages older than 14 days are automatically purged. You can modify this value by changing the `MAX_MESSAGES_AGE` parameter in the online panel.

Release messages

To release messages, first take a look at this page. It explains how to configure amavisd-new to listen somewhere for the AM.PDP protocol. This protocol is used to send requests.

Below is an example of a working configuration:

```
$interface_policy{'SOCK'} = 'AM.PDP-SOCK';
$interface_policy{'9998'} = 'AM.PDP-INET';

$policy_bank{'AM.PDP-SOCK'} = {
  protocol => 'AM.PDP',
  auth_required_release => 0,
};
$policy_bank{'AM.PDP-INET'} = {
```

```
  protocol => 'AM.PDP',
  inet_acl => [qw( 127.0.0.1 [::1] )],
};
```

Don't forget to update the `inet_acl` list if you plan to release from the network.

Once amavisd-new is configured, just tell Modoboa where it can find the *release server* by modifying the following parameters in the online panel:

Name	Description	Default value
Amavis connection mode	Mode used to access the PDP server	unix
PDP server address	PDP server address (if inet mode)	localhost
PDP server port	PDP server port (if inet mode) 9998	
PDP server socket	Path to the PDP server socket (if unix mode)	/var/amavis/amavisd.sock

Deferred release

By default, simple users are not allowed to release messages themselves. They are only allowed to send release requests to administrators.

As administrators are not always available or logged into Modoboa, a notification tool is available. It sends reminder e-mails to every administrators or domain administrators. To use it, add the following example line to root's crontab:

```
0 12 * * * <modoboa_site>/manage.py amnotify --baseurl='<modoboa_url>'
#
# Or like this if you use a virtual environment:
# 0 12 * * * <virtualenv path/bin/python> <modoboa_site>/manage.py amnotify --baseurl=
↪'<modoboa_url>'
```

You are free to change the frequency.

Note: If you want to let users release their messages alone (not recommended), go to the admin panel.

The following parameters are available to let you customize this feature:

Name	Description	Default value
Check requests interval	Interval between two release requests checks	30
Allow direct release	Allow users to directly release their messages	no
Notifications sender	The e-mail address used to send notitications	notification@modoboa.org

Self-service mode

The *self-service* mode lets users act on quarantined messages without beeing authenticated. They can:

- View messages
- Remove messages
- Release messages (or send release requests)

To access a specific message, they only need the following information:

- Message's unique identifier
- Message's secret identifier

This information is controlled by *amavis*, which is in charge of notifying users when new messages are put into quarantine. Each notification (one per message) must embark a direct link containing the required identifiers.

To activate this feature, go the administration panel and set the **Enable self-service mode** parameter to yes.

The last step is to customize the notification messages amavis sends. The most important is to embark a direct link. Take a look at the README.customize file to learn what you're allowed to do.

Here is a link example:

```
http://<modoboa_url>/quarantine/%i/?rcpt=%R&secret_id=[:secret_id]
```

Manual SpamAssassin learning

It is possible to manually train SpamAssassin using the quarantine's content. By train, we mean:

- Mark message(s) as spam (false negative(s))
- Mark message(s) as non-spam (false positive(s))

This feature is available to all users (from super administrators to simple users) but not enabled by default.

SpamAssassin configuration

For better performance and to enable the per-user level, SpamAssassin must store bayes information into a SQL database.

Create a new database and a new user/password (using your favorite database server) and edit the default configuration file (`/etc/spamassassin/local.cf`) to add the following lines inside:

```
bayes_store_module    Mail::SpamAssassin::BayesStore::<Driver>
bayes_sql_dsn         <DSN>
bayes_sql_username    <db username>
bayes_sql_password    <db password>
```

Replace values between <> by yours. Possible values for `Driver` are `PgSQL` or `MySQL` (non exhaustive list). The syntax for `DSN` depends on the driver you choose. Please consult the official documentation.

Enable the feature through Modoboa

Manual learning is disabled by default. You can activate it through the administration panel (*Modoboa* > *Paremeters* > *Amavis*). There two learning levels:

1. Global: available to administrators only. A single (global) bayes database is shared between everyone.
2. Per domain: available to administrators and domain administrators. Each domain can have a dedicated database.
3. Per user: each user can create its own database to customize the way SpamAssassin will detect spam.

The domain and user levels are not activated by default, dedicated parameters are available through the panel.

Note: Domain and user databases are only created the first time someone calls the learning feature through the quarantine.

Warning: A bayes database needs to reach pre-defined thresholds before it can be used by SpamAssassin. The default values are **200** spams and **200** hams.

You will find other paramaters related to this feature. You won't need to change them most of the time, unless SpamAssassin is hosted on a different machine than Modoboa. (in this case, `spamc` will be used instead of `sa-learn`).