Filtering spam and viruses at the mail server with Amavisd-new

SCAN MANAGER


Deutsche Post WorldNet

Sometimes the best time to stop bad mail is before it arrives. Amavisd-new is an Open Source interface for integrating spam and virus filtering with your mail server. **BY LARKIN CUNNINGHAM**

The majority of viruses are propagated via email. While it is still possible to receive viruses from floppy disks and CD-ROMs, or from Internet worms and rootkits that directly attack vulnerabilities in your operating system, you are much more likely to receive viruses via email – often by inadvertently opening seemingly innocent attachments.

Of course it is important to have an up-to-date virus checker on your desktop, but defense in depth is the best approach, and this is why you should be protected from spam and viruses at your receiving SMTP server. Virus protection at the mail server prevents undesirable email from reaching your desktop, and it also keeps viruses out of your POP3 or IMAP account. Filtering from the mail server reduces the load on your desktop PC or laptop, reduces the amount of bandwidth you will use, and greatly reduces the risk of spam and viruses reaching your desktop. An additional advantage of providing this extra layer of protection is that, by using a virus checker on your desktop that is different from the content checker on your SMTP server, you reduce the risk of a virus passing undetected.

Amavisd-new [1] is an Open Source tool that serves as an interface from a mail server to virus scanners and other forms of content checkers. Although some virus tools provide their own mechanisms for filtering mail at the server, Amavisd-new offers performance advantages in some environments, and it also provides a single, vendor-neutral configuration point for managing both virus and spam filtering.

High Performance

Amavisd-new is a flexible, high performance Perl-based application that runs as a daemonized set of master and child processes. Amavisd-new acts as an SMTP server, receiving email messages from your SMTP server (e.g., Postfix, exim, or qmail), processing the messages, and sending them or reinjecting them back into your SMTP server.

Amavisd-new supports anti-spam tools such as SpamAssasin [2], as well as a wide range of commercial and open source virus checkers. The popular Clam Antivirus [3] is supported in three ways; the clamd daemon (the best performance), the Mail::ClamAV Perl package (not so good performance) and the clamscan command line option (as a backup when clamd is unavailable, for example). Many other popular virus checkers are also supported, including F-Prot, Sophos, Grisoft's AVG, KasperskyLab AVP, AntiVir, F-Secure, McAfee and Panda. But be aware of the licensing for the product you choose. Obviously ClamAV will not incur any license fee, but some of the other options may have a license fee for an SMTP server that is significantly greater than the desktop license fee.

You can configure Amavisd-new to block attachments with commonly dangerous extensions, such as .exe, .bat, and .vbs (particularly dangerous to Windows clients who may be using the mail server.) You can also specify a wide range of decoders/decompressors to examine archives such as .cpio, .rpm, .deb, . zoo, .tar, .gz, and .bz2.

Amavisd-new can theoretically support any SMTP server, but it works better with the usual suspects, including Sendmail, exim, and qmail. Amavisd-new works best with Postfix [4], which can reinject mail back into itself after content filtering.

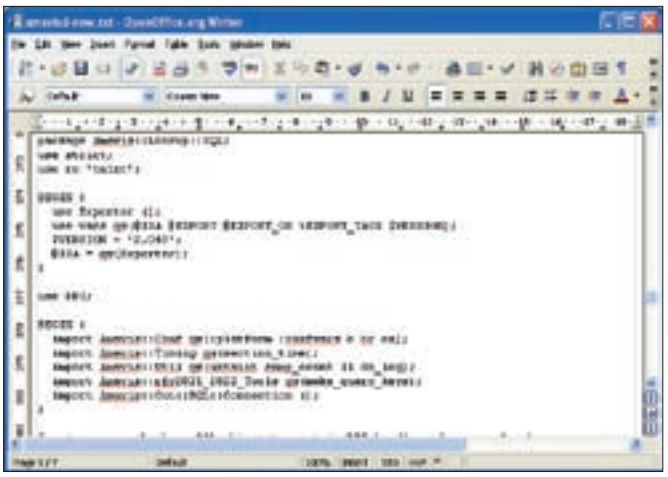

Figure 1: Amavisd-new is just one big Perl script.

Installation Requirements

To install Amavisd-new, you need a working Perl installation. Perl 5.8.2 or greater is recommended. Though earlier versions of Perl will work, later versions work better. You need to install a number of Perl packages (Table 1). Be sure to update any packages you already have to the latest versions.

Obviously, for effective spam filtering, the latest Mail::SpamAssassin package should be installed. Spammers attempt to stay one step ahead of SpamAssassin, so you should always update the SpamAssassin package as soon as a new version becomes available.

You should also install a number of additional programs to allow for the widest range of decoding and decompressing of email attachments. These additional programs include gzip, bzip2, arc, lha, rar, zoo, pax, cpio, freeze, ripole, cabextract, and many others. See the Amavisd-new website [1] for more on installation. After you have set up the background applications, the installation is relatively straightforward. Amavisd-new does not require the compilation of any code, since it uses the Perl interpreter.

Table 1: Required Perl Packages

Archive::Tar
Archive::Zip
Compress::Zlib
Convert::TNEF
Convert::UUlib
MIME::Base64
MIME::Parser
Mail::Internet
Net::Server
Net::SMTP
Digest::MD5
IO::Stringy
Time::HiRes
Unix::Syslog
BerkeleyDB

Configuring Amavisd-new with SQL

There are several ways to configure Amavisd-new. You can define your Amavisd-new configuration using the amavisd.conf file, lists files, hash lookup files, regular expressions, and LDAP or SQL lookups.

The SQL solution offers you the most freedom to build a front-end configuration tool using a scripting language such as PHP. You can use any database that supports standard SQL and is supported by Perl DBD/DBI libraries. SQL DDL (Data Definition Language) is provided in the README.sql file to create tables, indexes, and some sample data.

Notes are included for MySQL, PostgreSQL, and SQLite. Many other databases are also supported, including Oracle and DB2. Be aware that many of the tables have serial / sequence primary keys, so for Oracle, you'll need to add several sequences and triggers in addition to the tables.

You must specify two DSNs (Data Source Names). One DSN is for lookups and the other for storage (logging). You can improve performance by using a database that is very fast for read-only access to do lookups and a database that is fast for writes for storage. Amavisd-new also lets you specify multiple DSNs to allow for failover.

Policies

If you are familiar with firewall configuration, you will be familiar with the term *policy*. A firewall policy is a set of instructions to a firewall specifying what to do with certain types of network traffic. For example, a firewall policy might specify whether to block traffic to a certain network port or whether to redirect traffic from one port to a different port. An Amavisd-new policy is similar. With each policy, you can direct Amavisd-new to bypass spam checking or virus filtering, be tougher or more lenient with spam, tag the spam email's subject line with a marker (like {*Spam?*}), and forward all spam to a designated spam account.

Table 2: Some of the options in the policy table

virus_lover	Do not reject the email even if virus infected
spam_lover	Do not reject the email even if identified as spam
Note: The _lover options do not tell Amavisd-new not to scan, just to ignore the results of the scan.	
bypass_virus_checks	Do not scan the email for viruses
bypass_spam_checks	Do not scan the email for spam and do not add the X-Spam headers
spam_modifies_subj	Add some text (a tag) to the beginning of the subject if it is spam
virus_quarantine_to	An email account to send all virus infected emails to
spam_quarantine_to	An email account to send all spam to
spam_tag_level	The SpamAssassin score above which spam emails get X-Spam headers
spam_tag2_level	The SpamAssassin score above which spam gets the subject tagged
spam_kill_level	The SpamAssassin score above which 'evasive' action is taken, determined by $final_spam_destiny in the configuration file (default is to discard the email, i.e. Send it to a blackhole)
spam_subject_tag	The text to use for the spam tag (see spam_modifies_subj)
spam_subject_tag2	The text to use for the high scoring spam tag

Logging and the Law

A hot topic for ISPs at the moment is a potential requirement to store email data for up to three years. This is in response to recent terrorist atrocities and is designed to give law enforcement agencies information to track the activities of terrorists, in addition to telephone and mobile phone call and SMS logs. While storing email content would inevitably lead to a massive storage burden for ISPs, the archiving of the logging information generated by Amavisd-new could address many of the requirements of any future European or local government legislation.

Each email domain or user can have its own policy. You can specify a policy to cover all email accounts within a domain and also decide to have additional policies for individual email accounts within the same domain. For example, you might have a default policy to cover all users in a domain (like a catch-all policy), but perhaps Tom's email account is being flooded with spam. An individual policy for Tom's account could have a lower tolerance. Table 2 details just some of the policy options.

A user table allows you to assign policies to specific domains and individual email accounts. The user table's email field determines whether multiple email accounts or just a single email account will be affected by the policy. For example, the email @domain.com refers to all email accounts in the domain.com domain. But, tom@domain.com refers to a single email address. If both @domain.com and tom@domain.com exist in the users table, precedence is given to the specific email address.

Black and White Listing

When important emails you want to receive continually get blocked because they are identified as spam, you can take action to ensure that the sender of the emails is exempt from spam checking by adding them to the white list. Conversely, when you continually receive spam from a source you know is a spammer, you can add the email address to the blacklist.

Amavisd-new's black and white lists work by cross referencing the *mailaddr* (email address) table with the user table and the *wblist* (white / black list) table. Black or white listed emails are added to the *mailaddr* table with a unique identifier. The rows in the *wblist* table specify the email address and target user to which the rule applies.

Logging and Archiving

One of the newer features of Amavisd-new is the ability to log all email activity. The logging information is comprehensive and includes information such as senders, recipients, times, dates, subjects, and spam or virus scores. Logging lets you keep accurate statistics on the level of clean versus spam versus virus emails received. The fact that you can record these statistics at the user level gives the system administrator an opportunity to see where content checking resources are most needed. This gives an ISP (Internet Service Provider) the opportunity to bill for bandwidth, processing time, the number of messages processed, or the number of spam or virus emails processed.

Conclusion

For those who are considering building a robust and scalable architecture for battling spam and viruses, I recommend taking a look at the Postfix and Amavisd-new combination. Particularly for ISPs, there is an imperative to providing customers with spam and virus filtering, whether it is free or as an add-on to their service.

Growing levels of spam and viruses and greater numbers of customers are putting a strain on the infrastructure of many ISPs. An open source solution using Postfix and Amavisd-new could be the answer. Both Postfix and Amavisd-new can be configured to use MySQL, PostgreSQL, or Oracle for retrieving data. It is possible to have a central database accessed by multiple Postfix and Amavisd-new servers. And it is possible to employ multiple databases, allowing for greater fault tolerance. By using a number of failover techniques, including configuration of DSN records, it is possible to build a scalable, fully fault tolerant, and load balanced infrastructure. ■

Administration and Statistics Made Easy

Using a database instead of configuration files allows for easy administration. The database option also allows for dynamic updating of configuration data without having to reload the Amavisd-new daemons. The same is true if you opt to configure your Postfix users and domains in a SQL database. By querying the storage database using SQL, it is easy to garner useful information on spam and virus statistics. You could achieve this using phpMyAdmin, for example, if MySQL is the chosen database.

An ISP with a large number of customers could develop an interface that would allow customers to update their own policies to be more or less aggressive against spam or viruses. It would also allow users to view statistics on the levels of clean, spam, and virus infected emails. These types of statistics can only help in the education of email users. Figure 2 shows an example of the type of statistics you can easily present to customers. The example was developed using ColdFusion MX 7, but you could also use PHP with jpGraph.

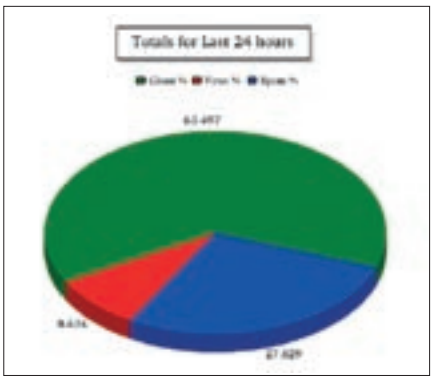

Figure 2: Amavisd-new provides detailed logging information that allows for informative charts.

INFO

[1] Amavisd-new: *http://www.ijs.si/software/amavisd/*

[2] SpamAssassin: *http://spamassassin.apache.org*

[3] Clam Antivirus: *http://www.clamav.net*

[4] Postfix SMTP Server: *http://www.postfix.org*

THE AUTHOR

Larkin Cunningham is an IT Consultant specializing in the areas of Linux Administration, Java, Application Frameworks, Oracle and just about anything in the open source community that makes the lives of IT Managers easier and Finance Directors happier. He can be contacted at *larkin.cunningham@gmail.com*.