Spam Control

Protection and Control

Are you noticing that your email is not just from friends? Do you find yourself having to spend time wading through hundreds of messages that you do not need? It is time to call Spam Control.

BY RICHARD IBBOTSON

Peter Doeberl, visipix.com

UCE, or spam as most of us call it, has increased so much in the past year that some Senators in the U.S have been having formal meetings about it. Microsoft and AOL are making public claims that they have taken steps to reduce spam. If you are a system administrator, it's highly likely that you will have noticed an increase in spam in spite of claims from many multinational companies that they are working on reducing spam. Someone may not be telling the truth.

Much of this strange phenomenon doesn't originate from the ancestors of the Vikings, but it does come from a small group of people in the States. Hence the interest from the US upper house. In England the law has recently changed with fines of up to £5000 in a magistrates' court, or an unlimited penalty from a jury. Similar laws are now appearing throughout Europe with Italy threatening up to three years imprisonment. You can only send unsolicited e-mail if the recipient has indicated a willingness to receive it. The regulations do not cover sending spam to business e-mail addresses. If you are in charge of running a mail server for a small company, you may be placed under a legal obligation to sort out your spam problems.

First of all a word about spammers. Your average spammer isn't – as you might think – someone who hasn't got anything better to do. There are some spammers who are built that way. In the present day, the spammer has a hidden agenda. Can they use spamming to break into your machine? Possibly. Can they socially re-engineer you, or your family, to get at you or your data in that way? Highly likely. Spam can contain viruses or other things that help the spammer to obtain information about your computer.

Whatever you might think about spam, the fact is that, in the hands of script kiddies or black hats, it can be used cracking tool – assuming the right techniques. So, it's not just an annoyance, but potentially dangerous when ignored, and without appropriate action to remove it from your mailbox or network. You should put some effort into spam detection and removal.

Will it cost me or my company a great deal of money to remove spam? The good news is that, as long as you work within the world of the GNU General Public License, you can do most things to help yourself or your organization without it costing any more than your own time. You can of course get hold of similar tools for MS Windows based platforms, but you will most likely need to spend a lot of money on software to do that. The free alternative is also somewhat superior to the proprietary software as it is open code and – having being around for a while – it has evolved based on experience.

What software can I use to reduce spam? There's quite a lot of spam control software out there. For the purposes of keeping things simple, I shall briefly discuss Procmail, Mail::Audit, and Spam Assassin. There are many others, but these are the most widely used and understood. The general assumption is that if you are using GNU/Linux or BSD, then you will be using something like Fetchmail, Procmail, and possibly Postfix or Exim, to download and send e-mail.

Procmail

Procmail [1] has been around for a long time. You can use it to process incoming mail in many ways before it is delivered to your mailbox. Procmail can be tortuously difficult to configure and use. Fortunately there is a Procmail list out there on the net. You can ask questions and get some good answers.

In recent years, Procmail has been superseded by other software, but there's nothing wrong with using it on a day to

THE AUTHOR

Richard Ibbotson is the organizer for Sheffield Linux User's Group. Photography is one of his major interests along with fishing, golf, wine and of course the Campaign for Real Ale. You can view the Sheflug web site at http://www.sheflug.org.

day basis if that's what you want. I thought I would mention it here just to make sure that it wasn't forgotten. All of the documentation is on the Procmail site. Make sure you have a look at it before you do anything with Procmail. You can also type *man procmail* and *man procmailex* at the command line to read a summary of what to do with Procmail.

Mail::Audit

Mail::Audit [2] is a library for creating easy mail filters produced by Simon Cousins because he wanted something other than Procmail. To quote Simon, "Procmail is nasty. It has a tortuous and complicated recipe format, and I don't like it. I wanted something flexible whereby I could filter my mail using Perl tests." It's basically yet another way of producing mail filters, but not just your usual filters. It can be extremely sophisticated and is definitely much better than Procmail on its own.

Before proceeding with installation, you should first of all check to see that your Perl RPMs or Debian packages are installed. The best way of installing Mail::Audit is from the nearest CPAN ftp site. To install, open up a root window and type:

```
perl -MCPAN -e shell
o conf prerequisites_policy ask
install Mail::Audit
```

At this point you may be prompted to install a new version of the CPAN modules. If this happens, update your CPAN modules first, and then continue with the Mail::Audit installation. You might also find that, if this is the first time that you have run CPAN on your machine, you will be asked for first time configuration options. All you have to do is follow the easy to understand instructions on the screen in front of you.

Once Mail::Audit is installed you will then have to create a *.forward* file in your home directory and something which we will call a *.spam* file. Both of these will be hidden files when they are created. You can find hidden files by running the *ls -a* command on a directory. The contents of the *.forward* file will look something like this:

```
"| IFS=' '  && exec /home/bob/
.spam -f- || exit 75 #bob"
```

You need to include the quotation marks in the file. The | means pipe, and */home/bob* is your home directory. So, now you point all of your mail at the *.spam* configuration file so that all mail is then scanned for the nuisance spam. See the example *.spam* file box.

Note that in the example */home/bob/Mail* is where all the mail folders are. Also note that if spam is found, it is placed in */home/bob/Mail/Spam* rather than being completely discarded. This means that, in the unlikely event of a false positive, the user can have a look in the spam folder to try to find lost mail. You will need to keep an eye on the disk usage as the spam levels can quickly fill up your space. If you want to, you can replace

```
$mail->accept( "/home/bob/
Mail/Spam" );
```

with

```
$mail->accept( "/dev/null" );
```

This will delete any mail that looks like spam. After installation you should run a few tests to make sure that everything is working. You can do this by opening up a root window and watching the logs whilst downloading mail. Type *less +F /var/log/mail.info,* or tail whichever log file is appropriate. To run a test, enter *fetchmail -d0* to begin a mail run, and watch the log file being created in front of you.

You might need to chmod the *.forward* or *.spam* file with the correct permissions in order to get it to work for you. This will become obvious when you see your log files. If all is well, you should now have a working version of

Example .spam file

```
01 #!/usr/bin/perl
02
03 use strict;
04 use warnings;
05
06 use Mail::Audit qw/KillDups/;
07 use Mail::Audit;
08 use Mail::SpamAssassin;
09
10 my $mailbox =
   "/home/bob/Mail/inbox";
11
12 my $mail     = Mail::Audit-
   >new( nomime => 1, );
13 my $spamtest =
   Mail::SpamAssassin->new();
14 my $status   = $spamtest-
   >check( $mail );
15
16 if ( $status->is_spam() ) {
17     $status->rewrite_mail();
18     $mail->accept(
   "/home/bob/Mail/Spam" );
19 } else {
20     $mail->accept(
   "home/bob/Mail/inbox" );
21 }
```

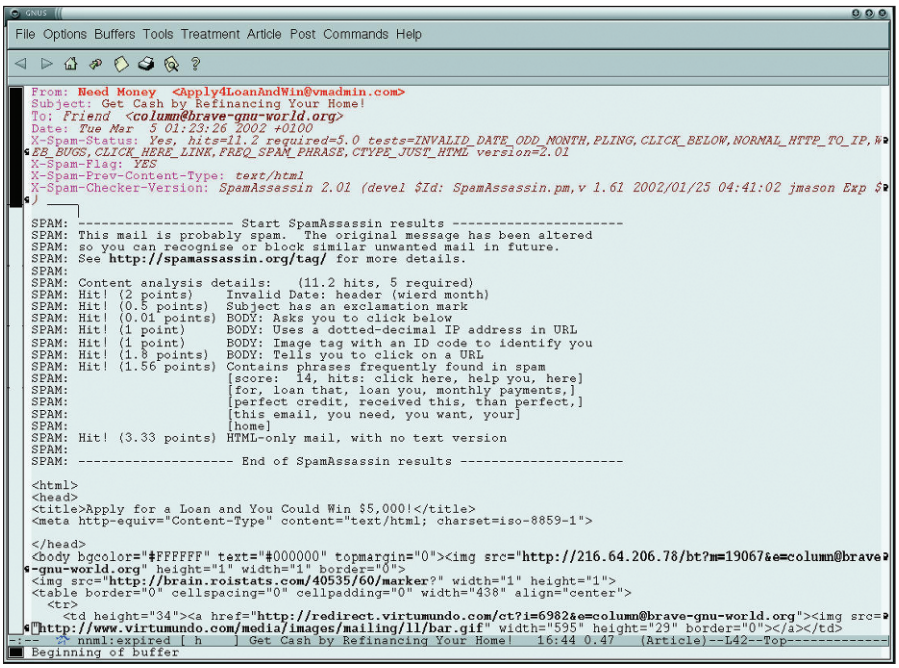

Figure 1: SpamAssassin detecting and scoring a spam mail.

Mail::Audit and you can sit back and look forward to a reasonably spam free existence. Not all spam will be blocked, but thankfully most of it will be.

SpamAssassin

SpamAssassin [3] is a mail filter which labels e-mail as spam when it matches certain criteria. It is probably the best thing since sliced bread. It is rules-based in that every email is given a cumulative score depending on whether it matches certain criteria. You can add appropriate lines to the *~/.spamassassin/user_prefs* file to change the scoring for these rules. A full list of the rules can be found at [4]. The spam identification used within SpamAssassin includes:

- header analysis: spammers use a number of tricks to mask their identities, fooling you into thinking they have sent a valid message, or fool you into thinking you must have subscribed at some stage.
- text analysis: again, spam mails often have a characteristic style (to put it politely), and some characteristic disclaimers.
- blacklists: SpamAssassin supports many useful existing blacklists, such as mail-abuse.org [5] or ordb.org [6].
- Razor: Vipul's Razor [7] is a collaborative spam-tracking database, which works by taking a signature of spam messages. Since spam typically operates by sending an identical message to hundreds of people, Razor short-circuits this by allowing the first person to receive a spam to add it to the database – at which point everyone else will automatically block it.

At the time of writing, the 2.61 version has just come into use and most people trust it to tag and deal with their spam. Once e-mail is marked and tagged, it can then be used as a future reference for tracking spam or for spam reporting. If you combine SpamAssassin with Mail::Audit, you can get some really good results. On a typical day, this author receives 300 to 500 e-mails and only about one message per week might be a false positive.

Installing SpamAssassin

There are many packages out there for Debian, Slackware and the other GNU/Linux distributions. It can be a lot easier to install the CPAN modules using the following:

```
perl -MCPAN -e shell
o conf prerequisites_policy ask
install Pod::Usage
install ExtUtils::MakeMaker
install HTML::Parser
install Net::DNS
install Mail::SpamAssassin
```

You might need to install some other modules such as Pod::Usage. You should find that CPAN will prompt you if other modules are required. Have a look at the documentation on the SpamAssassin site

SpamAssassin user preferences

```
01 require_version 2.60
02
03 # How many hits before a mail
   is considered spam.
04 required_hits          5
05
06 # Whitelist and blacklist
   addresses are now file-glob-
   style patterns, so
07 # "friend@somewhere.com",
   "*@isp.com", or "*.domain.net"
   will all work.
08 # whitelist_from
   someone@somewhere.com
09
10 # score SYMBOLIC_TEST_NAME
   n.nn
11
12 report_safe              1
13 ok_languages             en
14 ok_locales               en
15 use_dcc          1
16 use_pyzor               1
17 trusted_networks
   10.0.0/16
18 use_razor2              1
19 razor_timeout
   10
20 use_bayes               1
21 rbl_timeout
   15
22 check_mx_attempts       2
23 dns_available           yes
24 bayes_auto_learn              1
```

As we can see in this example, 5 hits are required before SpamAssassin will tag any e-mail. This is a very low score and may or may not be desirable. You may wish to set it to around 8 or 12. Some experimentation is probably required for first time use so that you can make decisions at a later date, based on experience. Also note that whitelists and blacklists are mentioned, but not used in this example. Looking further down the configuration file, English is defined as the default language. DCC checks – as well as the Pyzor and Razor – are enabled, as is the DNS check which makes sure that the whole thing will work.

before you get started for those other modules.

After installing you will have to configure the rules and the *user_prefs* file which will allow you to start and use SpamAssassin. All of the rules, such as *25_body_tests_pl.cf* and the others, will probably be in the wrong place. The SpamAssassin documentation explains that you need to move the *.cf rules to */etc/mail/spamassassin,* or similar, to get SpamAssassin to work. This is probably the most difficult part of configuration. The rules can be in different places when using Debian, Slackware or Suse distributions. You'll have to use your own ideas to get it right.

Because of this, it might be good to run *spamassassin -lint* before configuring the user_prefs file. If you do that, you'll find that a large pile of error messages falls down the screen if the *.cf rules are in the wrong place. If all that happens is that the command line returns to the # bash prompt, then all may be well. After configuring this with the help of *perldoc Mail::SpamAssassin::Conf* or *man Mail::SpamAssassin::Conf,* you can then run *spamassassin -lint* once again to check for error messages. If you get any errors at this point, you just need to comment out any rules in user_prefs with a # symbol.

After running *ls -a* in your home or root directory, you should now find that you have a directory called *.spamassassin*. This is where your *user_prefs* file lives, along with *bayes_seen* and *bayes_toks*.

For more advanced usage, such as spam tracking and reporting, you might wish to read about and install DCC (Distributed Checksum Clearinghouse) [8], Razor [7], or the open Python version Pyzor [9]. You can have a look at the web sites that are related to this software for further info. After reading the documentation, you should be aware that whitelists and blacklists are also possible. These were useful before sophisticated spam checking software such as Mail::Audit and SpamAssassin came along. You can still use this feature of you like, but SpamAssassin with its auto-learning feature, or even running in simple mode, doesn't seem to need whitelists and blacklists.

Hopefully these few pages will help the home user or an administrator in a small office environment to understand some useful facts of life about spam and how to deal with it. ■

INFO

[1] Procmail: *http://www.procmail.org/*

[2] Mail::Audit: *http://search.cpan.org/~simon/Mail-Audit-2.1/Audit.pm*

[3] SpamAssassin: *http://www.spamassassin.org*

[4] SpamAssassin rules: *http://eu.spamassassin.org/tests.html*

[5] Mail Abuse Prevention System: *http://www.mail-abuse.org*

[6] Open Relay DataBase: *http://www.ordb.org*

[7] Razor: *http://razor.sourceforge.net/*

[8] DCC: *http://www.rhyolite.com/anti-spam/dcc/*

[9] Pysor: *http://pyzor.sourceforge.net/*