Intelligent filtering with Qmail, SpamAssassin, and Maildrop

SORTING JUNK

We'll show you a custom solution for moving spam to a separate folder and adding new spam signatures to SpamAssassin.

BY MATTHIAS JANSEN

Many email readers offer built-in spam filtering. These mail clients also provide a means for teaching the spam filter new spam signatures. To teach a signature, customers tag an unsolicited message as spam, or move the spam message into a separate folder. Some internally managed mail servers would benefit from the advantages of a user-teachable spam filtering system. This article describes a custom user-teachable filtering system using Perl, the Maildrop delivery agent, and SpamAssassin.

This example is based on the open source Qmail mail transfer agent [1]. Qmail has a simple interface for handling messages before assigning them to a mailbox. To allow this to happen, Qmail can launch any program, such as a customized Perl script. (Other email servers offer similar functionality. If you are using a different mail server, consult the vendor or project documentation.)

In the scenario in this article, an incoming mail message is filtered through SpamAssassin. The Maildrop delivery agent, which is included on most Linux distributions, then drops the message into an appropriate directory depending on the spam status as recorded in the header.

For the teaching feature, we will create a special folder for messages that the user marks as spam, and we'll also show you a script that will check the contents of this special folder and feed new spam

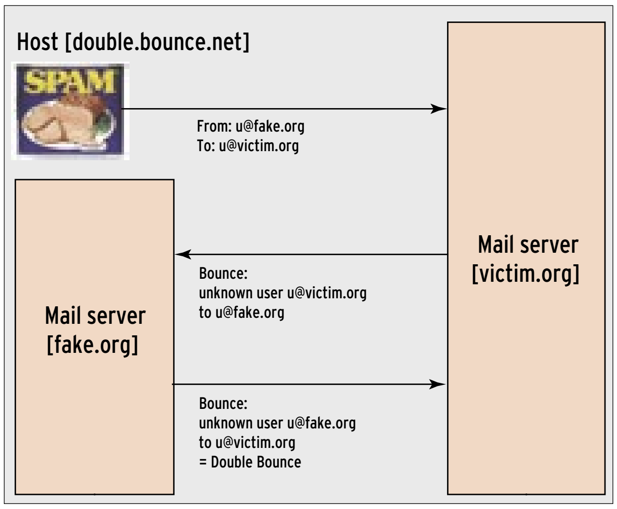

Figure 1: Maildrop performs a test to discover double bounce email – duplicate error messages caused by spoofed senders.

information to SpamAssassin's Bayesian database.

Sorting with Maildrop

The Maildrop deliver agent was originally designed for the Courier mail server. The purpose of Maildrop is to implement filtering rules and delivery instruction for mail messages. Your Linux distro probably has a version of Maildrop, but if not, you can obtain Maildrop from the Courier website [4].

Maildrop supports central configuration via the */etc/mail-drop.conf* file. (A command line parameter lets you specify a different file if needed.) To tell Qmail to call Maildrop for every email, you need to add a | *maildrop* entry to the *.qmail-default* file or to the Maildrop configuration file: | *maildrop /path parameter config file*.

The *.qmail-default* file for the domain in our environment would be */var/vpopmail/domains/jansen-systems.de/.qmail-default*.

If you opt for the user-specific variant, you need to create a */var/vpopmail/domains/yourdomain.com/user/.mail-default* file. Maildrop expects mail input on stdin. Note that it does not support very long messages in memory but, instead, swaps out to disk to save memory.

Mail Doctor

Maildrop has various options for scrutinizing incoming mail. It understands regular expressions, which facilitates filtering on complex constructions. The example in Listing 1 exploits just a fraction of the tool's potential.

The first twelve lines initialize variables and load external Qmail variables (see Table 1). The backticks (`) fulfill the same role as in Bash; that is, they assign the output of the quoted command to the variable in question. This then gives users the ability to call external commands, *vuserinfo* in our case, to discover the user's mail directory.

In lines 18 through 23, Maildrop performs a test that detects double bounce mails (see Figure 1), as it can then simply delete this kind of message. The next test checks whether Qmail (or Vpopmail [2], which is working behind the scenes) has a valid email address for the incoming message.

If not, Listing 1 stipulates that mails should not simply be discarded, as Qmail does not resolve aliases prior to delivery, so that messages to aliases do not have an entry for the *VHOME* variable. This is the reason why the script

Listing 1: /etc/maildroprc

```
01 # Global maildrop filter file
02 SHELL="/bin/bash"
03
04 import EXT
05 import HOST
06
07 logfile "/var/log/maildrop.
   log"
08
09 VPOP="| /var/vpopmail/bin/
   vdelivermail ''
   bounce-no-mailbox"
10 VHOME=`/var/vpopmail/bin/
   vuserinfo -d $EXT@$HOST`
11 SPAMFOLDER="Spamverdacht"
12 LEARNFOLDER="spamteach"
13
14 log "EXT is $EXT"
15 log "HOST is $HOST"
16 log "VHOME is $VHOME"
17
18 `test "$EXT" ==
   "doublebounce"`
19 if ( $RETURNCODE == 0 )
20 {
21   log "Delete doublebounce"
22   exit
23 }
24
25 `test "$VHOME" == ""`
26 if ( $RETURNCODE == 0 )
27 {
28   log "No such user,
   delivering to normal qmail"
29   to "$VPOP"
30 }
31
32 # Create the spamfolder if it
   does not exists
33 `test -d "$VHOME/.maildir/
   .$SPAMFOLDER"`
34 if( $RETURNCODE == 0 )
35 {
36 }
37 else
38 {
39   `/usr/bin/maildirmake -f
   "$SPAMFOLDER" "$VHOME/.
   maildir"`
40   `chown vpopmail:vpopmail -R
   "$VHOME/.maildir/
   .$SPAMFOLDER"`
41
42   `/usr/bin/maildirmake -f
   "$LEARNFOLDER" "$VHOME/.
   maildir"`
43   `chown vpopmail:vpopmail -R
   "$VHOME/.maildir/
   .$LEARNFOLDER"`
44
45   `echo "INBOX.Spamverdacht"
   >> $VHOME/.maildir/
   courierimapsubscribed`
46   `echo "INBOX.spamteach" >>
   $VHOME/.maildir/
   courierimapsubscribed`
47 }
48
49 if (/^X-Spam-Status: *YES/)
50 {
51   log "Spam!!!"
52   # then try delivering it to a
   Spam folder
53   exception {
54     to "$VHOME/.maildir/
   .$SPAMFOLDER"
55   }
56
57   # just for the case something
   went wrong with the spamfolder
58   exception {
59     to "$VPOP"
60   }
61 }
62 else
63 {
64   exception {
65     to "$VPOP"
66   }
67 }
```

passes the mail on to the normal delivery agent in lines 25 through 30; the agent then resolves the alias and passes the message back to Maildrop for further investigation.

Maildrop now checks if the target has a mail folder for spam, and if not, the script simply creates the folder – everybody is bound to receive spam sooner or later. As some mail clients do not possess the ability to automatically create IMAP folders, Squirrelmail [5] for example, lines 33 through 47 add the new folders to the subscription list.

Spam Status and Storage Location

The next step is to check whether SpamAssassin has identified the message as spam. Maildrop could run a regular expression against the raw content of the message, that is, both against the header and against the body. This would allow you to add your own filter rules, to discover the boss's name or a friend's name, for example.

The setup we have looked at thus far will be fine with a simple check to see if SpamAssassin has set the *spam status* in the mail header to a positive value (*Yes*). If this tag exists, the *to* command pushes the message to the designated folder (Figure 2).

Listing 2: spamreport.pl

```
01 #!/usr/bin/perl -w
02 #
03 # Script for collect all spam and generate a spamreport
04 #
05 # Author: Matthias Jansen
06 # Version: 0.2, 2007/01/05
07
08 use strict;
09 use Mail::Sendmail;
10 use HTML::Entities;
11 use Fcntl ':flock'; # import LOCK_* constants
12
13 # get all possible spam for the last 24 hours
14 my @files = `/usr/bin/find /var/vpopmail/domains/ -mtime -1 -a -type f -a \\( -path '*.maildir/.Spamverdacht/cur/*' -o -path '*.maildir/.Spamverdacht/new/*' \\) -a -not -name '*T' -print`;
15 my $lastuser = '';
16 my $count = 0;
17
18 # collect all the data
19 for (my $i=0;$i<@files;$i++) {
20
21   my($domain, $user) = $files[$i] =~ /\/var\/vpopmail\/domains\/([^\/]+)\/([^\/]+)\//;
22
23   my $currentuser = $user."@".$domain;
24
25   if ($lastuser ne $currentuser) {
26     my @temp = ();
27     $spams{$currentuser} = \@temp;
28     $count = 0;
29   }
30
31   $lastuser = $currentuser;
32   if (open(SP,"<".$files[$i])) {
33     # lock the mail before reading
34     flock(SP,LOCK_EX);
35     my $subject = '';
36     my $from = '';
37
38     while(defined(my $line = <SP>) && (length($subject) == 0 || ($from eq ''))) {
39       if ($line =~ /^Subject:\s*(.*)$/i) { $subject = $1; }
40       elsif ($line =~ /^From:\s*(.*)$/i) { $from = $1; }
41     }
42     # encode HTML codes
43     encode_entities($subject);
44     encode_entities($from);
45
46     my %spam = ('subject' => $subject, 'from' => $from);
47     $spams{$currentuser}[$count] = \%spam;
48     $count++;
49
50     flock(SP,LOCK_UN);
51     close(SP);
52   }
53 }
54
55 # generate the spamreport now
56
57 open(TPL,"</root/spamreport.tpl");
58 my @a_tpl = <TPL>;
59 close(TPL);
60 my $tpl = join("",@a_tpl);
61
62 while (my ($user,$data) = each(%spams)) {
63   (my $text = $tpl) =~ s/###EMAIL###/$user/;
64   my $spam_text = '';
65   for (my $i=0;$i<scalar @$data;$i++) {
66     $spam_text.='<tr><td>'.@$data[$i]->{'from'}.'</td><td>'.@$data[$i]->{'subject'}.'</td></tr>';
67   }
68   $text =~ s/###SPAMTEXT###/$spam_text/;
69
70   my %mail = ( To     => $user,
71     From    => 'Spamreporter <spamreport@jansen-systems.de>',
72     Message => $text,
73     Subject => 'Spamreport',
74     'Content-Type' => 'text/html; charset="utf8"'
75   );
76   sendmail(%mail);
77 }
```

To avoid errors, Maildrop adds an exception block to protect commands. The block catches errors and allows the script to continue running, just like exceptions in Java or C++. In case of error, Maildrop simply delivers the message normally.

A different kind of error can knock Maildrop out cold. Common web interfaces, such as Qmailadmin, store the catch-all settings in the *.qmail-default* file. If the web interface discovers a different catch-all function entry, Qmailadmin simply overwrites, thus ousting Maildrop. To prevent this from happening, make sure the change is implemented in the *maildroprc* file's *VPOP* variable only.

Listing 3: spamreport.tpl

```
<html>
<head>
</head>
<body>
<b>Spam Report by YOURNAME for Email Address ###EMAIL###</b>
<br><br>
New mails in spam suspect folder in the last 24 hours<br><br>
<table border=1 cellspacing=0 cellpadding=5>
<tr><th>Sender</th><th>Subject</th></tr>
###SPAMTEXT###
</table>
</body>
</html>
```

Daily Reporting

No matter how good your suspect spam folder is, it will be no use to users with incoming POP3 mail, as POP only delivers to the main inbox.

To allow users to investigate their spam candidates, we'll generate a list of the suspected spam. The Perl script shown in Listing 2 creates a list by checking for files in the spam suspect folder. Line 12 generates the list of candidates, discarding deleted emails (with a *T* at the end of their names).

Lines 23 through 25 check the age of undeleted spam to avoid re-listing spam that has already been listed.

To deliver the report, line 27 extracts the username from the filename; lines 43 through 46 provide the subject and sender of the suspicious message. The email is exclusively locked up front to prevent competitive access.

After completing the list for the report, lines 63 through 83 merge the list with a

advertisement

template (Listing 3) and delivers the mail. In our scenario, Cron launches the Perl script at 7AM every morning.

Back to School

To keep feeding unsolicited mail to the spam suspect folder, the scanner has to adapt to spammers's increasingly sophisticated tricks.

SpamAssassin has an *sa-learn* command that gives users the option of feeding tokens from existing spam and ham (good) mails to the software to improve its built-in heuristics. However, SpamAssassin lacks an internal good/bad detection mechanism. To teach the program this difference, we will add an IMAP folder, called *Spamteach*.

Cron helps the Bash script, *collectspam.sh* in Listing 4, regularly check the contents of this folder and feed it to SpamAssassin's Bayesian database, that is, to *sa-learn*, pointing out that the input is spam.

After the spam messages are integrated into the database, *collectspam.sh* simply deletes the messages. You can run *sa-learn --dump magic* to make sure that everything is working; Listing 5 gives an example.

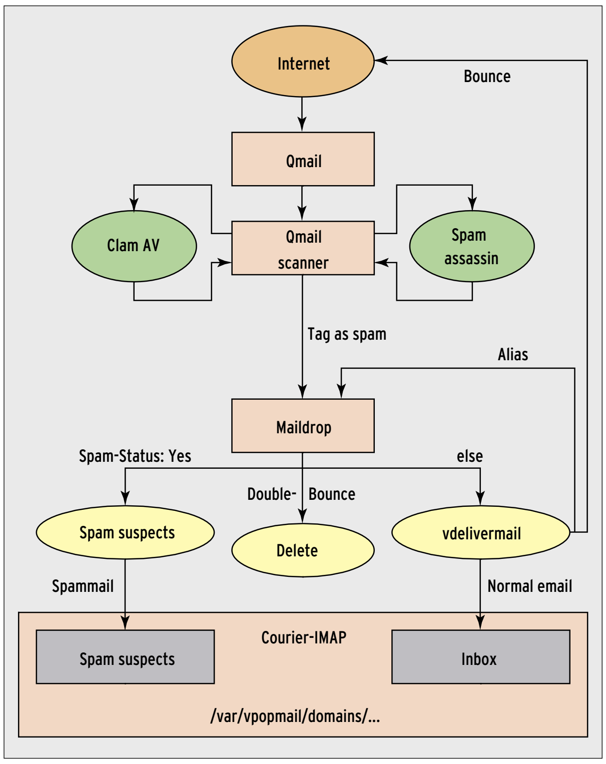

Figure 2: Cooperation between Qmail, SpamAssassin, and Maildrop supports accurate assignment of email to inboxes.

It is hardly worth checking for errors at this point; in the worst case scenario, a teach suggestion might fail. However, if you insist on being notified in case of an error, just remove the redirector to */dev/null*, and Cron will pass output on to you.

Scalability

The configuration that we investigated will let system administrators add effective antispam functionality to their Qmail systems, but without a great deal of extra effort. The MTA has some easy-to-use interfaces to help you, and it also offers support for multiple domain and user environments.

If you use a different MTA, some creative adaptation of the scripts should provide some comparable results. You could easily add a "ham teach" folder to the setup we looked at here to help SpamAssassin more easily detect good mail. ■

Listing 4: collectspam.sh

```
01 #!/bin/sh
02
03 if /usr/bin/ls /var/vpopmail/domains/*/*/.maildir/.spamteach/
   cur/* >/dev/null 2>/dev/null; then
04     /usr/bin/sa-learn --spam /var/vpopmail/domains/
*/*/.maildir/.spamteach/cur/* >/dev/null 2>/dev/null
05     /usr/bin/rm /var/vpopmail/domains/*/*/.maildir/
.spamteach/cur/* >/dev/null 2>/dev/null
06 fi
```

Listing 5: sa-learn --dump magic

```
# sa-learn --dump magic
0.000   0          3  0  non-token data: bayes db version
0.000   0       2829  0  non-token data: nspam
0.000   0       1753  0  non-token data: nham
0.000   0     132715  0  non-token data: ntokens
0.000   0 1139451052  0  non-token data: oldest atime
0.000   0 1163003305  0  non-token data: newest atime
0.000   0 1162978202  0  non-token data: last journal sync atime
0.000   0 1161566741  0  non-token data: last expiry atime
0.000   0   22118400  0  non-token data: last expire atime delta
0.000   0      30090  0  non-token data: last expire reduction count
```

Table 1: Variables

Variable name	Function
EXT	User part of the email address
HOST	Host part of the email address
VPOP	Specifies the default command for mail delivery, including parameters
VHOME	Mail directory for the current user
SPAMFOLDER	Name of the IMAP folder for spam mail.
LEARNFOLDER	Name of the IMAP folder with emails to feed to SpamAssassin.

INFO

[1]	Qmail: *http://www.qmail.org/top.html*
[2]	Vpopmail: *http://www.inter7.com/index.php?page=vpopmail*
[3]	Qmail Rocks On Gentoo: *http://gentoo-wiki.com/QmailRocksOnGentoo*
[4]	Maildrop: *http://www.courier-mta.org/maildrop/*
[5]	Squirrelmail: *http://www.squirrelmail.org*