vbSpam supplement

CONTENTS

FEATURE

BLOCKING UNWANTED MAIL WITH MAIL AVENGER

David Mazières
New York University, USA

Junk mail filters must evolve constantly to keep pace with increasingly clever spammers and virus writers. Mail Avenger, developed by the New York University Secure Computer Systems group, is an extensible SMTP server designed to facilitate mail filter innovation. It allows users to implement sophisticated filtering policies easily using Unix shell syntax, which is familiar to most administrators and many end users. Mail Avenger runs as a wrapper around existing mail transport agents (MTAs), permitting people to adopt new filters regardless of the underlying mail system in use.

SMTP-TIME FILTERING

The best time to filter mail is as early as possible, during the execution of SMTP, the Internet mail protocol. Refusing mail during an SMTP transaction saves the server from having to spool unwanted messages. Moreover, since legitimate clients notify senders of SMTP failures, inappropriately blocked mail will be brought to the attention of the sender.

A further advantage of filtering during the execution of SMTP (SMTP-time filtering) is that more information is available to filters while the client is still connected to the server. For example, filters can examine the network route to the client or check frequently updated real-time blacklists (RBLs) when deciding whether to accept a message.

Unfortunately, most MTAs make SMTP-time filtering difficult by requiring MTA-specific, trusted plug-in code that can affect all users if it malfunctions. To avoid this hassle, many people run mail filters at delivery time, through .forward, .procmailrc, or .qmail files, which allow one to hook in external filter programs. However, by the time such programs run, the server has already accepted mail from the client, leaving no satisfactory way to reject it.

Filters typically discard bad mail silently or place it in a dedicated junk folder, but either option allows legitimate mail to be overlooked if improperly categorized. Notifying senders of blocked mail by generating bounces is not a good solution, however, because most spam comes from forged sender addresses, meaning that innocent third parties receive unwanted bounces.

SMTP IMPLEMENTATION

Mail Avenger opens up the server-side SMTP implementation, allowing users to control SMTP responses with scripts and external programs. There are three principal commands issued by an SMTP client to send a message to the server. In order, these commands are:

```
MAIL FROM:<sender-address>
RCPT TO:<recipient-address>
DATA
message-body
.
```

The server responds to each command with a three-digit result code, followed by a more detailed explanation (and optionally an extended result code).

To accept mail, the server returns a 200-series result for each command – often just '250 ok'. To reject mail, the server returns a 500-series result for one of the commands (e.g. '550 unknown user'). Alternatively, the server can defer mail by returning a 400-series result (e.g. '451 temporary DNS error'), in which case a legitimate client will keep trying to send the message for a few days.

Mail Avenger performs most filtering in response to RCPT and DATA commands. Generally, it returns success to MAIL commands, unless the sender address is syntactically malformed or some transient error occurs (such as an overload condition or a name server failure).

The result of the RCPT command is determined by running a script depending on the recipient. At small sites, where each recipient corresponds to a Unix user, Mail Avenger runs the script ~/.avenger/rcpt in the user's home directory. For users without ~/.avenger directories, Mail Avenger runs a system-wide fallback script: /etc/avenger/default. Configuration files also allow administrators to map mail aliases and virtual domains to particular users.

~/.avenger/rcpt files are ordinary Unix shell scripts, sourced from a script that pre-defines a number of Mail-Avenger-specific environment variables and shell functions. For example, here are a few of the environment variables Mail Avenger sets:

`CLIENT_NAME, CLIENT_IP`	domain name and IP address of the client
`CLIENT_NETPATH`	the network route to the client
`CLIENT_SYNOS`	a guess of the client's operating system type
`RECIPIENT`	the recipient address of the message
`SENDER`	the sender address of the message
`SENDER_LOCAL, SENDER_HOST`	the user and hostname parts of SENDER

`SENDER_BOUNCERES`	SMTP error if SENDER cannot receive bounces
`SPF`	SPF disposition (whether CLIENT_IP is authorized for SENDER)

Many of these values are also included in a new X-Avenger: header field, which may help certain Bayesian spam filters.

Here are some of the shell functions available:

`accept` *`[MESSAGE]`*	This signifies that the server should accept the RCPT command with response '250 *MESSAGE*'.
`reject` *`[MESSAGE]`*	This signifies that the server should reject the RCPT command with response '550 *MESSAGE*'.
`defer` *`[MESSAGE]`*	This signifies that the server should defer the RCPT command with response '451 *MESSAGE*'.
`redirect` *`user`*	This redirects processing to the rcpt file corresponding to *user.*
`errcheck`	This rejects the mail if some simple default checks fail (for instance, if SPF indicates the mail is a forgery, or SENDER cannot receive bounces).
`greylist`	This defers mail the first time a SENDER uses a particular CLIENT_IP, but accepts if the client tries again at least 30 minutes later from the same CLIENT_IP. This technique has been known to defeat certain automated spambots.
`spf` *`VARIABLE QUERY`* `setvars`	This assigns *VARIABLE* to be the result of a query about CLIENT_IP using the SPF sender-specification language. Note the assignment to *VARIABLE* doesn't happen until the setvars function is called. To reduce latency, one can issue multiple concurrent spf commands (as well as other DNS-related commands that are not mentioned here) and wait for them with a single setvars.
`bodytest` *`COMMAND`*	This makes the RCPT command succeed, but then runs *COMMAND* on the body of the message to determine the result of the DATA command.

UTILITY PROGAMS

In addition to pre-defined shell functions, Mail Avenger comes with a suite of utility programs that help construct concise filtering policies. Some examples follow.

Suppose you have a mailing list that is never used as a sender address, and you wish to refuse bounce messages to the list. Because bounce messages come from an empty SENDER address, you can use the following line in an rcpt file:

```
test -z "$SENDER" && reject "no bounces, please"
```

The following line greylists all mail from *Windows* clients (the most likely to be infected by spam-sending malware), using match, a simple string-matching utility that comes with mail avenger:

```
match -q "*Windows*" "$CLIENT_SYNOS" && greylist
```

To run the spamassassin mail filter on the body of an email message, you can use the following commands:

```
errcheck
bodytest "spamassassin -e 100 > /dev/null"
```

errcheck rejects the mail immediately if it is obviously forged, to avoid wasting time with spamassassin. The bodytest command says to run 'spamassassin -e 100' on the message contents. '-e 100' instructs spamassassin to exit with status 100 if it considers the message to be spam. Exit status 100 tells Mail Avenger to reject the DATA command. (Exit status 0 means accept, while most other values result in deferral.)

A limitation of the previous script is that spamassassin annotates messages to indicate what spam tests were triggered by the message, yet the example discards those annotations. Fortunately, Mail Avenger lets bodytest commands edit messages. Mail Avenger even comes with a utility called edinplace that runs a program, replacing its input file with the program's output. Thus, to preserve spamassassin's annotations, use:

```
errcheck
bodytest edinplace spamassassin -e 100
```

Another powerful feature of Mail Avenger is its support for extension addresses, originally popularized by the qmail MTA.

EXTENSION ADDRESSES

Extension addresses allow users to receive mail at multiple addresses. For example, with a default sendmail installation at site 'example.net', Unix user 'user' receives mail addressed not just to <user@example.net>, but also to <user+ANYTHING@example.net>. Qmail uses the '-' character by default, so that user can receive mail to <user-ANYTHING@example.net>. To determine the result

of RCPT commands for <user+EXTENSION@server.com>, user must create a file ~user/.avenger/rcpt+EXTENSION (where EXTENSION is the actual extension in lower case).

One application of extension addresses is to create restricted addresses under which mailing lists can be subscribed to. Suppose you subscribe to mailing lists under the address <user+lists@example.net>. The lists to which you subscribe are all hosted either at New York University (NYU) or Stanford. You want to ensure that spammers cannot send you mail, even if they get hold of the subscriber list. You can achieve this by specifying a policy in ~/.avenger/rcpt+lists that accepts mail only from clients at NYU or Stanford. For example:

```
spf EDU_OK ptr:nyu.edu \
    ptr:stanford.edu mx:cs.nyu.edu/24
setvars
test "$EDU_OK" = pass && accept
test "$EDU_OK" = error && defer "Temporary DNS error"
reject "Address for NYU/Stanford clients only"
```

The spf command formulates a query about CLIENT_IP. Specifically, 'ptr:nyu.edu' asks whether the client's name ends in 'nyu.edu'. Similarly, 'ptr:stanford.edu' checks whether the client's name ends in 'stanford.edu'. Finally, 'mx:cs.nyu.edu/24' checks whether the first 24 bits of CLIENT_IP are the same as any of the mail exchangers for cs.nyu.edu. If any of the tests are positive, EDU_OK is set to 'pass' and the mail is accepted. If there is a temporary error, EDU_OK is set to 'error' and the mail is deferred. Otherwise, the mail is rejected.

Another feature of extension addresses is the ability to write catch-all rules for all suffixes, as with qmail. For example, the file ~user/.avenger/rcpt+bounce+default in user's home directory will match mail sent to <user+bounce+ANYTHING@example.net>. (As with qmail, the word 'bounce' here is an arbitrary string to embed in email addresses, while 'default' is the literal string 'default'.)

One application is to authenticate bounce messages using temporary codes. Doing so solves the problem of viruses and spammers forging your email address and causing you to receive bounces for mail you have not sent. Mail Avenger comes with a utility called macutil that generates and checks cryptographically-protected expiration dates. By setting the environment variable

```
MACUTIL_SENDER="user+bounce+*@example.net"
```

and then sending mail with the command 'macutil --sendmail' (which takes remaining arguments identical to sendmail), you can send outgoing mail from bounce addresses that resemble the following:

```
<user+bounce+tjmutvvdy6qfws4aztwuhsg6we@example.net>
```

Here, 'tjmutvvdy6qfws4aztwuhsg6we' is an encoded expiration date (cryptographically-protected with a password stored in file ~/.avenger/.macpass). You can then reject any bounces sent to your primary email address, by placing the following in your ~/.avenger/rcpt file:

```
test -z "$SENDER" && reject "no bounces, please"
```

Finally, you can check the validity of codes in the addresses at which you receive bounces. Taking advantage of the SUFFIX environment variable, which is set to the portion of the recipient address matching the trailing 'default' in the rcpt file name, you can place the following in your ~/.avenger/rcpt+bounce+default file:

```
macutil --check "$SUFFIX" \
     || reject "<$RECIPIENT>.. user unknown"
```

Because rcpt files are just shell scripts, it is easy to run external programs as mail filters. Moreover, because these programs run as the users in whose directories the rcpt files reside, a buggy rcpt script affects only recipient addresses that use the script. This makes it easy to develop and test new mail filters on a production mail server by deploying them initially only for certain recipients.

At large sites, system administrators can offer non-technical users a menu of filtering options. Default filtering can be implemented in the system-wide /etc/avenger/default file, while other scripts can be configured in the reserved avenger user's home directory, for example, ~avenger/.avenger/rcpt+strict, ~avenger/.avenger/rcpt+experimental. Users who wish to employ a particular level of filtering can simply place a line like the following in their ~/.avenger/rpct files:

```
redirect avenger+experimental
```

CONCLUDING REMARKS

Mail Avenger is MTA-independent. To spool accepted mail, it runs a configurable program (by default sendmail), and therefore it should be compatible with most existing Unix mail servers.

Mail Avenger has been tested with sendmail, qmail, and postfix on a variety of Unix variants including *Linux*, *OpenBSD*, *FreeBSD*, and MacOS X. Mail Avenger is free software, available from http://www.mailavenger.org/.

RELATED LINKS

[1] Mail Avenger: http://www.mailavenger.org/.

[2] SPF (Sender Policy Framework): http://spf.pobox.com/.

[3] SMTP (Simple Mail Transfer Protocol): http://www.faqs.org/rfcs/rfc2821.html.

[4] Spamassassin: http://spamassassin.apache.org/.