



SPAMASSASSIN

By Vicki Brown

Regaining Control of your Inbox

Ready, aim, fire!

About the author...

Vicki Brown has been using Unix systems since 1983 and Mac OS since 1986. She is delighted that Mac OS X gives her the opportunity to use both at the same time.

SEPARATING THE WHEAT FROM THE CHAFF

The popularity of the Internet, plus increases in connectivity (and email access), has given rise to a corresponding avalanche of spam. We all know what spam is – unsolicited junk email offering goods or services that we don't want, don't need, and often didn't want to know existed! We all agree that we would like less of it. Failing that, how can we wade through it, manage it, and delete it without losing our desirable mail (or our minds) in the process?

Many “solutions” have been proposed to deal with the spam problem. These range from anti-spam filters integrated inside email applications (e.g., the junk mail filters in Apple's Mail) to standalone products of various levels of complexity, ease of use, effectiveness, and cost.

My current solution of choice is SpamAssassin (<http://www.spamassassin.org>), in combination with procmail (<http://www.procmail.org>). I postprocess the results through my desktop email client, (Eudora), for a very powerful and accurate spam-killing solution.

SpamAssassin provides many features, including:

- user-configurable spam score threshold
- ability to re-write Subject lines
- user-configurable filters, using Perl regular expressions
- modification of existing Spam scores
- use of statistical, “Bayesian” analysis
- “auto-learning”
- optional use of DNS blacklists (e.g., Real-time Blackhole List)
- optional use of Network Checksum Tests (services that compare message checksums to known spam)
- “whitelists” (From addresses that are considered OK)
- Accepted message languages

Note that SpamAssassin and procmail both run on the server side (where mail is originally delivered), not on the client side (i.e., usually not on the desktop) and neither has a GUI front end or a particularly “user-friendly” configuration mode. If you, or your users, are not technically inclined, I recommend that you choose a different path. However, if you enjoy

tinkering, have some understanding of regular expressions (e.g., you've used Perl) and like to work with text-based configuration files, you will find the SpamAssassin/procmail combination to be powerful, flexible and, more important, accurate.

CONFIGURATION

SpamAssassin comes pre-configured with a large set of tests that it will perform on all incoming mail. In addition, you can add new tests, skip tests, raise or lower the "score" assigned by a given test or, using procmail, cause some mail to skip SpamAssassin altogether. See <http://www.spamassassin.org/doc.html> for more documentation than this review has room to provide. You'll even find a pointer, at that URL, to a SpamAssassin configuration generator tool, designed to make it easier to customize an installation of SpamAssassin with some common options.

Much of the power of SpamAssassin comes from its configurability, its use of Perl regular expression pattern matching, and its interaction with procmail.

For example, many spammers have started adding ever-changing sets of numbers to the ends of subject lines, to fool the simpler junk mail filters. One GUI-based anti-spam application I tried had a set of filters designed to try to catch mail of this form. The filters looked like this:

```
if Subject ends in 0
or if Subject ends in 1
or if Subject ends in 2
...
or if Subject ends in 9
```

This is rather cumbersome; worse, it doesn't handle the spammers who throw in a space at the very end. SpamAssassin's approach is both shorter and more flexible. This regular expression matches any digit, followed by 0 or more whitespace characters, at the end of the Subject line.

```
Subject =~ /\d\s*$/
```

SpamAssassin's interaction with procmail also allows me to specify whether mail is even sent through the SpamAssassin filters. For example, I can specify that mailing list messages should be delivered directly. This procmail recipe checks the message headers for one that matches the given Reply-to string, sending mail from the SpamAssassin-Talk list to my pre-defined default mailbox, without further processing.

```
:0 H
* ^Reply-to:. *spamassassin-talk
$DEFAULT
```

RESULTS

SpamAssassin doesn't actually delete any spam. Instead, it tags each piece of mail it processes with a set of headers, e.g.,

```
X-Spam-Status: Yes, hits=6.2 required=2.5
tests=AWL,CLICK_BELOW,FREE_TRIAL,HTTP_WITH_EMAIL_IN_URL,
spam_PHRASE_05_08,VLB_spam_OFFER_4,VLB_TO_NOT_NAME,WEB_B
UGS
version=2.43
X-Spam-Flag: YES
```

```
X-Spam-Level: *****
X-Spam-Checker-Version: SpamAssassin 2.43 (1.115.2.20-
2002-10-15-exp)
```

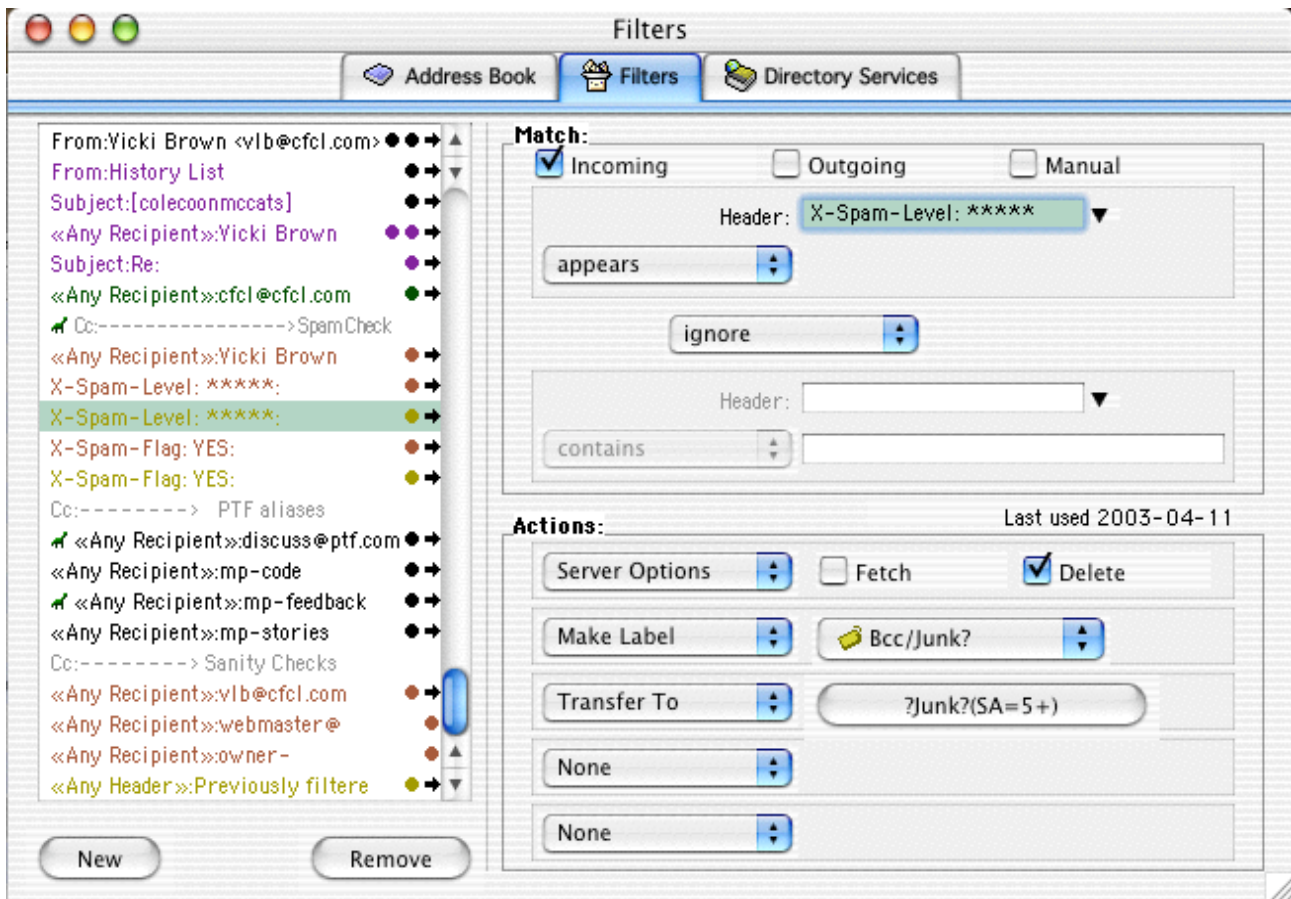
X-Spam-Flag is a simple Boolean; if a message is considered potential spam, the flag is set (and the value is YES). X-Spam-Level provides more information; one "*" is given for each integer value of the spam score. X-Spam-Status provides full scoring information as well as which tests passed. Note that some tests raise the spam score while others are designed to lower it. After all tests have been run, the resulting score is compared to a (user configurable) threshold. If the score exceeds the threshold, the message is tagged as spam.

The user gets to decide how to handle the mail after it's been processed and tagged. One possibility is to continue processing with procmail. For example, the SpamAssassin docs assert that mail with a score of 15 or higher is almost certainly spam (with 0.05% false positives, according to SpamAssassin's rules/STATISTICS.txt file). After some use, I decided to lower this cutoff even more. I have configured procmail to delete any mail tagged with a Spam score of 10 or higher, using the following recipe:

```
:0:
* ^X-Spam-Level: \*\*\*\*\*\*\*\*\*\*\*
/dev/null
```

Other than that, all remaining mail is delivered. I have occasionally found false positives in mail with scores of less than 10; I want to check that mail by eye.

Here's where my third tool comes in. I use Eudora as my mail application and make extensive use of Eudora's filter mechanism to post-process my potential spam. I separate potential spam by mailboxes and labeling, based on certain criteria such as whether the message contains my full name (or only my email address), how high the spam score is, and any other interesting criteria. I scan the "Junk" mailboxes a few times a day, pull out false positives (setting new filters to catch those the next time!), and trash the real junk.



INSTALLATION

This is only a review, not a tutorial, so the installation section will hand wave a lot. You'll need a server machine running some variant of *nix (e.g., FreeBSD, Linux, or Mac OS X). If your server runs Mac OS X, you'll need to make sure it has been configured to deliver mail locally (e.g., using sendmail or qmail). How to set up a server is beyond the scope of this review.

You should also note that, although SpamAssassin can be run without procmail, its flexibility (and power) increases when it is used in conjunction with procmail. SpamAssassin does not include code to handle local mail delivery; it relies on procmail (or something else) for delivery. Unless you know that a reliable "something else" is available, use procmail.

Procmail may already be on your server; it is pre-installed on most *nix systems these days (use `whereis procmail` or `locate procmail` to check, then read the documentation to determine how to configure procmail for your server situation). If it's not pre-installed, download and build procmail from the procmail.org web site.

SpamAssassin is probably not pre-installed on your system; however, installation is simple. In fact, you'll probably be able to install it without first downloading the archive. SpamAssassin is written as a Perl module; the easiest way to install it is by using Perl's CPAN shell (from the command line, as root):

```
perl -MCPAN -e shell
```

```
o conf prerequisites_policy ask
install Mail::SpamAssassin
quit
```

Alternatively, you can always download the latest archive from spamassassin.org, then build and install SpamAssassin according to the documentation. If you have any problems, be sure to read the Installation notes on the SpamAssassin site. Check the documentation for more information on configuration, as well as useful options.

SUMMARY

If you're looking for a powerful, flexible, and accurate solution to the spam problem, I recommend that you consider the team of SpamAssassin and procmail. Both programs are free and come with example recipes and suggestions for how to use them to your best advantage. Each has a support community that you can draw upon for assistance and ideas, as well as mailing lists to keep you informed. If you have a server on which to install it (note that SpamAssassin can be installed server-wide or on a per-user basis) and know (or are willing to learn about), regular expressions, SpamAssassin may be the spam-killing tool you've been looking for. Give it a try.

REFERENCES

SpamAssassin home page: <http://www.spamassassin.org>

SpamAssassin discussion lists: <http://www.spamassassin.org/lists.html>

Procmail home page: <http://www.procmail.org>

Procmail discussion lists: <http://www.procmail.org/era/llists.html>