Controlling Spam with SpamAssassin

Colin McGregor

Abstract

How to set up SpamAssassin and teach it to recognize spam.

The people who produce unsolicited commercial e-mail (UCE), or spam, are the big thieves of the Information Age, spewing out messages for pharmaceuticals, time pieces, fast money and fast women. Large chunks of bandwidth that we have to pay for is eaten up by these spammers. After getting these messages, we have to waste time going through our inboxes and deleting the garbage. Further, unlike magazines, newspapers, commercial radio and television, where the advertisements reduce the cost or make the content free, spam gives nothing back to us as readers or viewers.

Although we cannot stop spam, some tools exist to make spam easier to deal with. One such tool is SpamAssassin, which looks at each incoming e-mail message and rates the probability that the e-mail is spam. Messages that are given a high probability of being spam get flagged as such, and other programs, such as Evolution, KMail or Procmail, can deal painlessly with the flagged e-mail.

SpamAssassin works by going through e-mail messages and looking for things that are associated with spam or non-spam e-mail, which add or subtract points from an e-mail's score. So, for example, the word Viagra, and close misspellings of Viagra (as they are used in many pharmaceutical spam messages), adds to the total score. On the other hand, a valid Sender Policy Framework (SPF) record in the e-mail, which shows that the sender location was not forged, subtracts from the score. By default, any message that gets a total score of five or more is assumed to be spam.

One problem with the above calculations is that it is a fair bit of work for your computer, so if your machine is currently straining under the workload it has, or if you deal with a lot of e-mail, you may want to look at a hardware upgrade (faster CPU chip and/or more memory) before starting up SpamAssassin.

A number of Linux distributions include SpamAssassin by default. If yours isn't one of them, it should be very simple to add. If you have a Debian-based distribution, it should be as simple as starting up a terminal window and typing:

```
sudo apt-get install spamassassin
```

Once installed, you can start tweaking SpamAssassin's settings. SpamAssassin's configuration file can be found at `~/spamassassin/user_prefs`. The first setting is `required_score`:

```
required_score  5
```

SpamAssassin is not perfect, no matter how you set things. There will be some spam e-mail allowed through, and some valid e-mail will be classed as spam. The goal with the configuration process is to make sure this happens as seldom as possible. The score of five is an excellent compromise for most people. But, if you find yourself getting a lot of spam coming through as non-spam, even after taking the configuration steps noted below, you may want to lower the required_score.

If, on the other hand, you find after configuration you have a lot of real e-mail identified as spam, you might want to raise the required_score.

There are some people that you always want to hear from, or at least, always want their e-mail to come through, such as coworkers and family members. There also are people that you never want to hear from again, such as annoying exes. SpamAssassin deals with these situations by having a whitelist and blacklist. An e-mail from someone on the whitelist gets 100 subtracted from the score; anyone on the blacklist gets 100 added to the score. To add someone to your white/blacklist, you need to add something like the following to `user_prefs`:

```
whitelist_from       nicepersons@somedomain.somewhere
blacklist_from       nastyperson@somedomain.somewhere
```

Some people have specific reasons why they would want particular spam tests changed. For example, people working at a jewelry store, or watch collectors, might want to allow messages where the word Rolex has been emphasized, accepting that doing so also will increase the amount of replica-watch-related spam they will see. There is a list of SpamAssassin tests at [http://spamassassin.apache.org/tests.html](http://spamassassin.apache.org/tests.html). For example, to change the score that an e-mail message gets when the word Rolex has been emphasized, reducing the chances that such a message would be tagged as spam, put the following line in `user_prefs`:

```
score EM_ROLEX 0
```

If too many legitimate Rolex-brand watch-related e-mail messages are still being tagged as spam, the above could be changed to a negative number.

By default, SpamAssassin assumes e-mail in a number of Asian languages, most notably, but not exclusively Chinese, Japanese and Korean, are probably spam. This is a problem if you use one of those languages. To allow Asian languages, you need to uncomment some lines by removing the `#` character at the start of the last four lines of `user_prefs`.

```
whitelist_from       nicepersons@somedomain.somewhere
blacklist_from       nastyperson@somedomain.somewhere
```

Now, let's further refine SpamAssassin's taste. My first run-through with SpamAssassin was a disappointment. Out of some 2,200 spam messages, only about 10% were correctly identified as spam. Fortunately, with SpamAssassin there is a utility program called sa-learn that will “teach” SpamAssassin what you consider to be spam and ham (non-spam). This process greatly improves SpamAssassin's ability to identify spam messages correctly. The trick here is to create folders, one filled with spam and another filled with the sort of material you want to keep, and then feed each folder into sa-learn. Using the Evolution e-mail program, I created a folder called BULK, and then I manually placed all the spam messages into that folder. Next, I ran the sa-learn program with the following command:

```
sa-learn --mbox --spam ~/.evolution/mail/local/BULK
```

Evolution stores all its e-mail in the mbox mail format, thus the `--mbox` option in the command above. The command for the non-spam messages, which I keep in the Inbox folder, is:

```
sa-learn --mbox --ham ~/.evolution/mail/local/Inbox
```

The learning system SpamAssassin uses starts to become good at around 1,000 spam and 1,000 ham messages. With a semi-exception, the system doesn't improve noticeably until after seeing more than 5,000 e-mail messages. The semi-exception relates to the fact that spam is a moving target. Some spammers are always looking for better ways to get around filter programs, changing their spam as they go. What this means is that you need to re-train SpamAssassin periodically with new spam and new ham. How often depends on your situation, but basically you need to re-train whenever you see a noticeable increase in the amount of spam getting past SpamAssassin. Still, with training, it is very possible to reach spam-detection accuracy rates of more than 99%.

Remember that SpamAssassin remembers what e-mail it has seen before, so although some people may be tempted to run the same 1,000 e-mail messages through sa-learn five times, all this will do is waste time.
Let's see how SpamAssassin, actually rates a sample e-mail. For a test, I created a simple text file, testmail.txt with the following:

From: MyUserID@SomeDomain.Somewhere  
To: aliceithink@somedomain.somewhere  
Date: Sat, 2 Dec 2006 13:34:50 -0400 (EDT)  
Subject: Back from vacation

Alice, I am back from vacation, anything important happen when I was away?

Colin McGregor

Then, I ran SpamAssassin as a test with the following command:

spamassassin -t testmail.txt

I received an output like the following:

From: MyUserID@SomeDomain.Somewhere  
To: aliceithink@somedomain.somewhere  
Date: Sat, 2 Dec 2006 13:34:50 -0400 (EDT)  
Subject: Back from vacation

X-Spam-Checker-Version: SpamAssassin 3.0.3  
(2005-04-27) on diamond
X-Spam-Level: 
X-Spam-Status: No, score=-5.9 required=5.0  
tests=ALL_TRUSTED,BAYES_00, 
NO_REAL_NAME autolearn=ham version=3.0.3

Alice, I am back from vacation, anything important happen when I was away?

Colin McGregor

Spam detection software, running on the system "diamond", has identified this incoming email as possible spam. The original message has been attached to this so you can view it (if it isn't spam) or label similar future email. If you have any questions, see the administrator of that system for details.

Content preview: Alice, I am back from vacation, anything important happen when I was away? Colin McGregor ...

Content analysis details:   (-5.9 points, 5.0 required)

<table>
<thead>
<tr>
<th>pts</th>
<th>rule name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0</td>
<td>NO_REAL_NAME</td>
<td>From: does not include a real name</td>
</tr>
<tr>
<td>-3.3</td>
<td>ALL_TRUSTED</td>
<td>Did not pass through any untrusted hosts</td>
</tr>
<tr>
<td>-2.6</td>
<td>BAYES_00</td>
<td>BODY: Bayesian spam probability is 0 to 1%</td>
</tr>
</tbody>
</table>

With a score of -5.9, SpamAssassin would not consider the above to be actual spam. By editing testmail.txt and repeating the above, you can see how SpamAssassin reacts to various sorts of keywords -- in particular, terms commonly found in spam such as luxury brand-name watches, pharmaceutical products, financial service terms and/or various pornographic terms.

It isn't clear yet what the magic bullet will be to stop spam and regain the bandwidth spam steals from all of us -- better technology, new laws or better enforcement of laws currently in place. Likely an end to spam will require a mixture of actions. In the meantime, SpamAssassin does make dealing with spam a less painful, but not pain-free experience.

**Evolution and SpamAssassin**

The Evolution e-mail display program has a good filtering system for sorting out incoming e-mail, but it is a bit weak when it comes to identifying spam. Fortunately, Evolution allows us to use external programs to help with sorting. From the main screen click on Tools→Filters. Then, click on Add to create a new rule. You need a name for this rule and spam should be just fine. Next, we want to send a copy of each e-mail to SpamAssassin and find out if SpamAssassin views the e-mail as spam; we do not care about the score SpamAssassin gives the e-mail, just a “yes” or “no”. So, we Pipe to Program and then throw everything except the result code away. We do this with the instruction:

```bash
/usr/bin/spamassassin -e > /dev/null
```

If the above command returns a value of 0, it isn't spam. Anything more than 0 means we very likely have a spam and want it dropped into a separate folder. In the example shown in Figure 1, I am sending the e-mail into a folder labeled BULK. After doing the above steps, we want the filter program to stop and wait for the next incoming e-mail.
As noted previously running sa-learn over the same e-mail twice is a waste of time. This raises another point when using Evolution and SpamAssassin, when you delete an e-mail message under Evolution, the program does not delete the e-mail from the ~/.evolution/mail/<file name> e-mail file, it just flags it for future removal. This way, if you make an error deleting an e-mail, you can get it back. To get rid of deleted e-mails completely under Evolution, you must click on Actions→Expunge. During your first days with SpamAssassin, when you might be running sa-learn several times over your BULK folder and your Incoming folder, you may not only want to delete e-mail previously seen by sa-learn, but also to Expunge it.