BayesInSpamAssassin

Bayes Introduction

The Bayesian classifier in Spamassassin tries to identify spam by looking at what are called *tokens*; words or short character sequences that are commonly found in spam or ham. If I've handed 100 messages to sa-learn that have the phrase *penis enlargement* and told it that those are all spam, when the 101st message comes in with the words *penis* and *enlargment*, the Bayesian classifier will be pretty sure that the new message is spam and will increase the spam score of that message.

If you're having trouble with Bayes, see BayesFaq for help.

Things to remember

- **Do not** train Bayes on different mail streams or public spam corpora. These methods will mislead Bayes into believing certain tokens are spammy or hammy when they are not.
- To *train* Spamassassin, you get a mailbox full of messages that you know are spam and use the sa-learn program to pull out the tokens and remember them for later:
  sa-learn --showdots --mbox --spam *spam-file*
  Then you get a mailbox full of messages you're sure are ham and teach Bayes about those:
  sa-learn --showdots --mbox --ham *ham-file*
  It is important to do both.
- The bayesian classifier can only score new messages if it already has 200 known spams and 200 known hams.
- If Spamassassin fails to identify a spam, teach it so it can do better next time. Run it through the sa-learn program and it will be more likely to correctly identify it as spam next time. Likewise, if SA puts a ham in your spam folder, run that message through **sa-learn --ham** *ham-folder*.
- It's OK to feed emails with Spamassassin markup into the sa-learn command – sa-learn will ignore any standard Spamassassin headers, and if the original email has been encapsulated into an attachment it will decapsulate the email. In other words sa-learn will undo any changes which Spamassassin has done before learning the spam/ham character of the email.
- If you or any upstream service has added any additional headers to the emails which may mislead Bayes, those should probably be removed before feeding the email to sa-learn. Alternatively, use the bayes_ignore_header setting in your local.cf (as detailed in the man page for Mail::SpamAssassin::Conf).
- An example of a *ham-file* could be ~/mail/saved-messages, or wherever your email client saves messages. Make sure all spam is deleted before using sa-learn on a ham-file.
  Similar to the training example above, for a maildir format mailbox, the commands should be altered as shown below.

For a mailbox you're sure contains only spam messages,

sa-learn --showdots --spam *spam-files* or *spam-directory/** for a whole folder of spam

Then you get a mailbox full of messages you're sure are ham and teach Bayes about those:

sa-learn --showdots --ham *ham-files* or *ham-directory/** for a whole folder of ham

Again, it's important to do both.

How to train Bayes without logging on

If you don't read your mail on the account where SpamAssassin is running, it can be challenging to do mistake-based training, where you learn false negatives (i.e., spam that was not caught) as spam. One approach is redirect your false negatives and use procmail to train on them, as described in ProcmailToForwardMail. Another is to use IMAP folders as described in RemoteImapFolder.

(DanKohn)

Here's an alternative... assuming you have an actual account on the server (with ssh access) here's a very simple script which will keep your local ham and spam mbox files in sync with what's on the server, and run sa-learn remotely. In my case I have folders called AA-HAM and AA-SPAM in Evolution. A good habit is, instead of deleting read mail, moving it to the ham folder. Anyways, the script should make things obvious, its a very basic script but it works:

```
#!/bin/sh
# trainspam v0.1

MAILDIR=~/.evolution/mail/local/Inbox.sbd
HAMBOX=AA-HAM
SPAMBOX=AA-SPAM

SERVER=dragon@smithers
TMPDIR=~/tmp
VERBOSE=1

# --- DONT EDIT BEYOND THIS POINT

echo Synchronizing $HAMBOX and $SPAMBOX to $SERVER:$TMPDIR
rsync --partial --progress -z -e ssh $MAILDIR/{$HAMBOX,$SPAMBOX} $SERVER:$TMPDIR

ssh $SERVER "
        echo ; echo 'Learning ham...' ; echo ;
        sa-learn --ham --showdots --mbox $TMPDIR/$HAMBOX ;
        echo 'Learning spam...' ; echo ;
        sa-learn --spam --showdots --mbox $TMPDIR/$SPAMBOX"
```

Some comments.

1. OBVIOUSLY – change the options at the very beginning.

2. The -z option to rsync automatically uses gzip compression; no need to do this first. Also it will only sync newly added parts of the file, it doesn't re upload the file everytime!

3. I didn't get round to the verbose stuff yet, but basically to make stuff cleaner take out all the echo lines, the --partial option to rsync, and the --showdots option to sa-learn.

4. If your key isn't authorized on the server, you'll need to enter in your password twice. Some info on how to do this here: http://www.unixpeople.com/HOWTO/configuring.ssh.html

(GadiCohen)

Training plus reporting

If you only train your own bayes database using *sa-learn*, you will not be reporting the spam message you received to spam checksum services such as dcc, pyzor, or razor. To report the spam to the checksum services, you will need to use *spamassassin -r < the_spam_message_file*. You may also need to register as a spam reporter for services such as razor. If you are not sure your reports are being accepted, run *spamassassin -rD < the_spam_message_file* and look for any debugging output telling you that you need to register.

You can only invoke spamassassin using *spamassassin -r* on single files. This is fine for "mbox" spam mailboxes which are all contained in one file. However, for "maildir" directories, you will need to run *spamassassin -r* on each message individually. If you are not sure which format you have, look at your mail directory. If you see one or more files and each file contains one or more messages, you have "mbox" format. If you see directories containing files, each file name is a long string of numbers, letters, and punctuation, and each file contains one email message, you have "maildir" format.

If you have "maildir" mailboxes, running *spamassassin -r* multiple times can be tedious for large numbers of spam. So you can use this report_spam.pl script to run it for you. The script is written in perl. You can save the script to your spamassassin computer and then run it using *report_spam.pl your_spam_directory*. Each message in your_spam_directory will then be learned in bayes **and** reported to the checksum services.

(KurtYoder)