SiteWideBayesSetup

Setting up Site-Wide Bayesian Filtering

1) Using DB_File / BerkeleyDB database (slow performance)

Slow but easy to use if you have very little traffic.

In local.cf, tell SpamAssassin where to find the Bayesian database files:

```
bayes_path /var/spamassassin/bayes_db/bayes
bayes_file_mode 0775
```

Note that the argument to bayes_path is a combination of a directory (/var/spamassassin/bayes_db/) and a filename prefix (bayes).

This tells the system that the Bayesian filter database files will be /var/spamassassin/bayes_db/bayes_msgcount, *seen and _toks. Feel free to move the database wherever you want. Please note this directory needs to be RWX for all users that SpamAssassin will be executed as, or R-X if autolearning and automatic expiry are disabled; many use world RWX to simplify this, but this is insecure and not recommended. The directory also shouldn't contain any files other than your bayes database. If it contains any other files that start with "bayes*" (or whatever other filename prefix you specified) it can break the database locking mechanisms SpamAssassin uses.

2) Using SQL database (good performance)

Decend performance, but requires database server. See sql/README.bayes in the release.

http://svn.apache.org/repos/asf/spamassassin/trunk/sql/README.bayes

3) Using new Redis database (extremely fast performance)

You should try to use the new Redis backend always. It is by far the fastest backend and easiest to maintain, very little moving parts and it will handle autoexpiring by itself.

https://spamassassin.apache.org/full/3.4.x/doc/Mail_SpamAssassin_BayesStore_Redis.html

http://svn.apache.org/repos/asf/spamassassin/trunk/contrib/HOWTO.Bayes-Redis/

Now start feeding the Bayesian filter spam and ham messages.

```
sa-learn --spam --showdots --dir /path/to/directory/full/of/spam/msgs
sa-learn --ham --showdots --dir /path/to/directory/full/of/ham/msgs
```

Do not simply use your inbox to train Bayes! The mailboxes of ham and spam messages used for training should be hand-verified, and should be kept after the initial training in case retraining is ever needed to correct problems with Bayes. It is safe to run sa-learn against the same mailbox multiple times, as a given message will only be learned once (unless its classification as ham or spam has changed).

See SiteWideBayesFeedback for more tips on getting an entire site to feed back spam and ham messages into the Bayesian filter.

Also restart spamd if you're running it so that it will re-read local.cf and enable the Bayes filter:

```
/etc/init.d/spamassassin restart
-or-
service spamassassin restart
```

Your method of restarting spamd may differ, but the above is typical. If you're using any MTA integrations that invoke SpamAssassin as a perl API (i.e. Amavis, MailScanner or mimedefang), that process will need to be restarted or told to reload its configuration as it is effectively it's own spamd.

Restarting spamd/Amavis/MailScanner/mimedefang is not needed after maintenance training or a background expiry, just when you enable or disable bayes.

You may experience difficulties with file permissions. Make sure you chmod any existing bayes files to readable/writable by your user groups (or world if you're doing so).

If you are going to use group rights instead of a world RWX, there are some additional issues you will need consider. If you use spamd and mail gets scanned on behalf of "root" spamd will use "nobody" as its effective user for bayes database access. You should consider this user when planning your group memberships. Also, be aware that the files are deleted and recreated by whatever user happens to be running spamassassin when an expiration is due. If you are not using world RWX this means you need to be aware the files will lose any group ownership you may have set unless you make the directory setgid.

See Mail::SpamAssassin::Conf(3) for details.