WeLoveVolunteers

We're looking for people to volunteer and make code contributions. Patches, code, perl, regression tests, rules, you get the picture. You'll have to send in a Contributor License Agreement before it can be accepted, but that's easy.

You'll need to download the latest version of SpamAssassin from SVN: DownloadFromSvn

So, what are we looking for right now? (Note: This has a tendency to rapidly get out of date. Subscribe to the dev MailingLists so you'll stay up to date, and let us know if you're planning on doing something big so we don't duplicate effort or end up wasting anyone's time.)

Documentation

- We need people to help improve the documentation, to improve its accuracy, completeness, and clarity. BetterDocumentation
- If you also know perl, you can also read our code to ensure the documentation agrees with what the code does, or to write man pages for perl modules that don't currently have one (see man perlpod).
- This wiki is not nearly as complete or as useful as we'd like it to be! Please, help us by editing it! Send a brief request to the dev@ list to be added to the Contributors Group including your wiki username. No mailing list subscription required.
- There's a lot of stuff in the masses/ directory of the tree that has essentially no documentation.

New Rules

- We are looking for people to help us write new rules, and help us add rules written by others (i.e. rules from CustomRulesets, with permission of course) into our code base. A good knowledge of SpamAssassin and experience writing rules would be useful. Let us know if you're interested by sending mail to the dev mailing list (MailingLists). Generally the developers are focused more on code issues rather than rules, so the more help we get in the rules department means we have more time to devote to writing code. (We want to make this easier and better – RulesProjectPlan)
- If you simply want to suggest a number of rules, please open a ticket in Bugzilla.

Rule Quality

- We rely on individuals donating CPU time to run a NightlyMassCheck to help with our rule development. All this entails is keeping an accurate corpus of ham and spam and using some CPU time every night! We are especially in need of people receiving mail in languages other than English.

Code

Bug Fixes

- Log into Bugzilla and look for bugs that you can fix. Fix it, and attach a patch (from latest SVN) to the bug report. See also UsingBugzilla.

Bug Tidying

- There are many bugs that just need either a couple votes and a commit, or to be closed. Find a bug, read through all the comments, and provide a concise summary of what needs to be done with it. If a patch is attached, make sure the summary starts with "[review]". For commits to stable (3.3.x as of May 2011), three "+1" votes are required from committers. Anything that was committed to trunk before the last time it was branched (3.3.x was branched in January 21 2010) should probably be closed. With 514 open bugs over 2 years old as of May 2011, we can use all the help cleaning up we can get.

Reduce Resources

- Submit code to speed something up without breaking anything. Minimum is probably about a 1% speed-up in overall check speed.
- AutoWhitelist needs AutoWhitelistExpiry bug 3082 and bayes_seen databases need to have automatic expiry.

Bayes

- Code and corpus tests that for ramping up the probability for previously unseen tokens. This could be done either heuristically or by keeping real counts of unseen tokens in the Bayes token database. The idea is that words that have never been learned before get high probabilities.
- Custom database file and code for faster performance and space savings (probably to be compared against qdbm and tdb since they look most promising right now as non-custom databases).
- Bi-grams: that is, multi-word windowing as used in CRM-114, using two-word tokens (or possibly even higher). Not sure this will provide much higher accuracy now that spammers are using whole-phrase bayes poisoning, though. (JustinMason)
- Implementing Dobly noise-reduction - bug 3078.
- Dynamically determining the autolearning thresholds based on incoming email rather than using hard-coded numbers. See bug 1829 for more.

Other ideas

- Translation : translation of rule descriptions, the manual, the website in other languages

- Feedback button : client side button to enable a one touch feedback for users to recategorized a message (false positive or negative to correct state). Relevant page from the Anti-Spam Research Group (part of a sister organization to the one that creates RFCs).