# RulesProjectPlan

## SpamAssassin Rules Project

**(DRAFT - this part of the wiki is a discussion document, based on emails to dev list. Please feel free to add comments, but be sure to make clear that it's your opinion, by signing your name to them. Your real name is preferred, btw.)**

### The Problem

Here it is, stated by DuncanFindlay: 'SpamAssassin is not as effective as it could be because of the rules that are being used to detect spam. There are two problems here:

1. The "not enough rules" problem: SpamAssassin does not have enough high quality spam-catching rules. Anecdotally, our FN ratio seems to be much higher with 3.1 than with 3.0 (we won't know for sure until the mass-checks are done). There may be a variety of reasons for this:
    - The SpamAssassin committers are not spending much time writing rules. Attempts to recruit people to become committers to write rules have been somewhat unsuccessful. We could always use more committers and contributors; what can we do to encourage more contribution?
    - People that do write rules for their own use are not willing to go through the fairly elaborate process in order to submit them to SpamAssassin (this currently requires rules to go through bugzilla and then through 70_testing.cf and eventually into our distribution). What can we do to make this process easier and more inviting?

2. The "release cycle" problem: Any high quality rules that are incorporated into SpamAssassin are not distributed until the next release. Since rules and code are tied together, the release cycle for rules is too long. Submitted rules are not distributed while they are most effective, and rules lose their effectiveness too quickly.'

3. [added by LorenWilton] The instant an actual rule is posted on the user's list, it will lose about 80% of its effectiveness, usually within about 16 hours. Within a week it will be virtually useless. Sometimes the rule will regain some effectiveness a few months later, and in rare cases posting a rule will not affect the hit rate. But in general, public posting in a readable forum of a rule body will negate the usefulness of the rule almost instantly.'

4. [added by BobMenschel from others' discussion] SA rules development handles rules aimed at spam in English best, since most SA rules developers that feed the distribution system speak and correspond in English, and the great majority of the testing corpora are based in English. We're not as good at developing, validating, testing, or scoring rules in other languages.

### The Solution

Based on the problem areas outlined above, here are the pages for each aspect of the problem, and proposed solutions:

- Encouraging contribution: RulesProjMoreInput
- the sandboxes solution: RulesProjSandboxes
- Streamlining the 'getting rules into SpamAssassin' task: RulesProjStreamlining
- Speeding up the release cycle for rules: SaUpdatePlan
- the secrecy problem: RulesProjSecrecy
- the language problem: RulesNotEnglish
- a continuous mass-checking system: RulesProjBuildBot

### Outstanding Tasks/Votes

Here's a list of the tasks that have fallen out of the above plan so far... we now need to vote to go forward with these, then put them into action.

First step – the sandboxes:

- PMC vote to approve the sandboxes project (RulesProjSandboxes).
    - *VOTE: passed!*
    - *Done*
- reorganise the rules directory into core/ , sandbox/, and extra/; link that rules project SVN repository to 3.2.0's 'rules' dir; use SVN externals to do this.
    - *VOTE: passed*
    - *Done*
- move current ruleset into a new "core" area
    - *Done*
- write scripts to test, filter, and pull rules from sandboxes automatically into core/ production ruleset
    - *Dropped in favour of:*
- write scripts to test, filter, and pull rules from sandboxes and core, as a compilation step, into an output directory (see RulesProjPromotion)
    - *Done*
- start using the above scripts to generate ruleset in svn
    - *Done!*

Phase two – mass-checking systems:

- Weekly mass-check
  - *DONE: all rules, with --net*
- Nightly mass-check: web-based user interface for the following data:
  - *DONE: freqs for all rules*
  - *DONE: freqs collated across all users' corpora, or individually*
  - *DONE: overlaps between rules*
  - *DONE: historical rule hits data?*
  - *DONE: rule-by-rule comparative performance figures?*
  - *DONE: promotion criteria as defined in RulesProjPromotion, so rules that can be promoted can be identified at a glance*

Phase three:

- *DONE: the RulesProjBuildBot system, comprising these tasks:*
  - _DONE: set up new buildbot master - http://buildbot.spamassassin.org:8011/_
  - *DONE: set up user in zone*
  - *DONE: set up new buildbot slave*
  - *DONE: set up chroot jail*
  - *DONE: get mass-check running in chroot*
  - *DONE: copy in corpus*
  - *DONE: set up additional slaves for additional corpora*
  - *DONE: write mass-check wrapper script to:*
    - *DONE: mass-check that corpus, using rules from .../rulesrc/sandbox/ only*
    - *DONE: implement strict ulimits*
  - *DONE: write mass-check-completed script to:*
    - *DONE: output freqs so it's visible through the Buildbot UI*
  - *DONE: write mail-handling script to extract mail-submitted rule attachments (PreflightByMail)*

Loose-ends-tying-up:

- TODO: still need to nail down promotion criteria, in particular performance figures
- TODO: bug fixing as they crop up