BrainStorming

The Brain{Storm,Dump}

_ This is a page for SpamAssassin developers and other interested parties to collaboratively jot down some ideas. Some of these might evolve into real code one day others might stay here forever. But as long as they are here, they are generally up for adaption and proof-of-concept implementations are welcome. Actual feature-requests should still go to Bugzilla. If you write down an idea, try to think of all the pros and cons which come to your mind. And don't forget to mark your submissions with your name_

Multi-level rules

MalteStretz: The idea:

- Each rule is assigned an integer level or reliability. There are only a few (three to five) of those classes/blocks.
- First rules with the highest reliability are run.
- If their conclusion already marks the message as spam or ham with a very high propability, all lower levels are skipped.
- This is similar to the now gone feature of shortcutting other tests when a certain score was reached but has the advantage, that it doesn't has the overhead to check the limit after each rule but after each block/level (of which there are only a few).
- (Hopefully) tests are run in a pre-defined as-optimal-as-possible order which could be determined by some algorithm (GA, perceptron).

Predicted Rule Reliability

BobMenschel: Close to these ideas, but different, is the specification (possibly via tflags indicating that a rule is expected to be "highly reliable" or "less reliable" than most.

For instance, a rule which tests for obfuscation in a long string or one that's not subject to false hits would be "highly reliable" (obfu on "viagra" comes to mind), while a single-word uri test would be "less reliable" than most.

During the perceptron scoring mechanism, we would adjust highly reliable rule scores higher, and less reliable rule scores lower than otherwise, to give a boost to the former, and to avoid false positives from the latter.

Bayes-driven rules

MalteStretz: The idea (see also "Multi-level rules"):

- The engine could learn the "reliability" of rules.
- A rule which didn't work good before might be skipped in subsequent runs.
- Possible con: There are rules which don't apply often but if they do they are very reliable. Those {might,will} be misclassified.

MalteStretz: I just noticed that BartSchaefer submitted bug 3785 "Suggestion: Use Bayes classifier to select among score sets" just a few hours ago. Very similar to this idea.

SVM and RF

DanielQuinlan: trusted networks heuristic:

- doing reverse look-up of HELO, if it matches the user's domain, and forward DNS matches of that matches the IP address, then trust it - does that help? (may be helpful for extending the boundary)
- find an IP that matches exteral MX, know it's external then (probably not as helpful for finding a boundary?)

JustinMason: tried both of these before, expensive in terms of time, requires network lookups to compute trusted network boundary, also I got false negatives! (spams that were "close enough" to the MX record). Also this algo is very similar ot spamcop's.

DanielQuinlan: limit number of messages per sender (from morning talk)

- for Bayes learning to avoid over-training for that user (expiry, rolling over - how?)
- for rescoring process (like this idea)

MalteStretz: What does "SVM" and "RF" mean? (Too lazy too Google)

Trustwebs for Whitelisting

MalteStretz: See [TrustNetNotes].