SaUpdatePlan

Rules Project: sa-update and rules release cycle

(part of RulesProjectPlan)

Problem description: '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.'

Theo has written sa-update – a new script that will be included with SpamAssassin 3.1.0. In theory, we will be able to distribute rules more frequently, and rules releases won't be tied to code releases.

See SaUpdateBackend for details of how this has been implemented.

Scoring

The primary reason for the long release time for new rules is the need to score new rules (and rescore old rules) to generate optimal scores, to flag as much spam as reasonably, while keeping false positives to a conservative minimum.

BobMenschel: The ideal would be to find some way to incorporate new rules into a GA/Perceptron-line mechanism, perhaps a Perceptron run which a) assumes whatever hit frequency applied to the last full scoring run, b) freezes all scores in all score sets according to the most recent distribution, and then c) incorporates an sa-update scoring run and calculates appropriate scores for the new rules.

If that's not practical, then perhaps we can use some standardized algorithms to determine provisional scores. The algorithms we use for general purpose rules within SARE seem to work very well, adding significantly to spam scores without causing any significant number of FPs.

DanielQuinlan favored the first options, saying, "That would not be too hard and would be more accurate than any estimation technique. There is definitely a correlation between hit rates, S/O ratio, RANK, etc. to the ultimate perceptron-generated score, but the correlations are not all that high, unfortunately."

JustinMason: yes, agreed; perceptron just does a better job, every time. Having said that, we don't need to institute a policy requiring regular perceptron runs; we can actually measure false positive rates across an entire corpus, using the 'fp-fn-statistics' masses tool, and get an idea of whether the current scoreset is FP-prone or FN-prone as a whole (indicating that the perceptron needs to be run soon), or not.

in other words, let's defer on making this a task right now 😉