RuleQaApp

The Rule-QA application

This is visible here. It has three display modes:

- the default aggregate overview, where all rules are visible in an overview form for one mass-check
- the rule-detail view, which 'zooms in' to provide lots of detail about a specific rule in a specific mass-check
- the mass-check selector view, where you can pick which mass-check you want to view

Data is loaded from two sources:

- the 'preflight' mass-checks, which run after each checkin: PreflightBuildBot
- nightly mass-checks, which take place in a more leisurely, decentralised manner, but also include the 'core' local and network rules: NightlyMassCheck

Selecting a Mass-Check

Up at the top of the page, there's a table of links under the 'Which Corpus?' heading, allowing you to select a mass-check from the recently-performed set. Each line is a link to display that mass-check. The following data helps you select the mass-check to display:

- Date: the date of the mass-check
- MC-Rev: the revision that was current in *the SpamAssassin part of the repository* when this mass-check started
- Commit and Rev: details about the commit that created that revision, and the time that commit took place
- Author: the person who issued the commit

- [net]: a flag indicating whether the mass-check included network rules

(Note, 'Prior Commit and Rev' is more accurate than 'MC-Rev', since we share a repository with other Apache projects, which results in the 'MC-Rev' figure incrementing without any checkins taking place on *our* part of the repository. I think I've now fixed this bug, mind you.)

In addition, the line below this details the commit message for that revision, and the usernames that submitted logs.

The Mass-Check Selector View

If you click the **(List All)** link just below the mass-check listing table, you'll be brought to the mass-check selector view, which lists a lot more of the mass-checks.

It also divides them into three sets:

- **Network Mass-Checks**: these are the weekly mass-checks that use `--net`, and therefore will include data on network rule effectiveness. They are only run once a week, on Saturday night, as they take a long time to complete.
- **Nightly Mass-Checks**: these are the nightly mass-checks, described at NightlyMassCheck, which include the full core ruleset as well as whatever sandbox rules were checked in at the time.
- **Preflight Mass-Checks**: the preflight mass-checks described at PreflightBuildBot.

The Aggregate Overview

The aggregate overview displays all the rules in a form based on that of HitFrequencies. There's a few minor differences, however; most notably, there are links from each rule name to the rule-detail view.

Rules that meet the rule promotion criteria (RulesProjPromotion) are displayed in all their glory; rules that do not, are greyed out.

If multiple people performed mass-checks on that revision, all their data is aggregated and averaged, as if it was one gigantic mass-check.

Note that you can select a selection of rules using the 'Which Rules?' textbox.

The Rule-Detail View

The rule-detail view displays the following sections:

- **set 0, in aggregate**: same as the aggregate overview
- **set 0, broken down by message age in weeks**: a way to quickly see if the hit-rate is trending up or down for the rule
- **set 0, broken down by contributor**: see how the hit-rate changes from person to person
- **set 0, score-map**: hit-rates broken down by the score of the messages it hits, letting you see if a rule fires mostly on already-high-scoring spam
- **set 0, overlaps between rules**: display overlap with other rules
- **Graph, hit-rate over time**: see below

Note the (more info) links at the top-right of every freqs graph; this allows you to see the header lines from that mass-check, if you so desire. This is useful if you want to find out whose corpus was used, how many mails were used, etc.

Finally, at the bottom, there's a link to go 'back' to the aggregate view.

The 'hit-rate over time' graph

This graph displays how the rule's hit rate has changed over time, breaking it down by time and by submitter. For example, here's a demonstration using the RCVD_HELO_IP_MISMATCH rule.

There are two graphs, one for spam, and one for ham messages. On the left of each graph is the percentage of the messages, of that type, in that time period, that were hit by the rule. On the bottom is the date the messages in question were received, going from the past (on the left) to the current date (on the right).

Each submitter has their own colour, which is used to highlight a scatter-plot of points indicating the hit-rates on their corpus; in addition, they have a Discontiguous Exponential Average line, which attempts to give a reasonable average of these points.