SaUpdateBackend

Rule Promotion and the sa-update Backend

'sa-update' updates are created on the zone, nightly, by a cron job. Here are some details as to how this works.

At 0830 UTC, the **build/mkupdates/run_nightly** script is run from svn trunk. This runs the script **build/mkupdates/listpromotable**, which queries the rule-QA web app for the previous day's rule accuracy figures, and determines which rules from the rules source dirs (rulesrc/core and rulesrc/sandbox/*) are to be *promoted* to the active ruleset, as described in RuleLifeCycle.

Promotion Criteria

You can see the previous day's accuracy figures at this URL. Promotable rules show up in dark text on the RuleQaApp, non-promoted-by-default rules show up in light grey text.

The promotion criteria are detailed on some other wiki page, RulesProjPromotion perhaps? (TODO).

Anyway, using those figures, and some other criteria:

- "tflags nopublish" rules are never published
- rules with a `T_` prefix to their names are never published - if you're getting these in a normal installation, you should probably report a bug.
- rules do not currently need an explicit "tflags publish" line to be published but the existence of either "tflags publish" or "tflags nopublish" is recommended to make it clear that the rule is intended to be published if it meets the promotion criteria.
- "tflags net", "userconf" or "learn" rules are always published
- rules that fail lint – if that can be attributed to the rule! – are not published
- rules that require a plugin that **build/mkupdates/listpromotable** thinks is not part of the default plugin set, are not published

A proposed new list of 'active' rules is created in the file 'rules/active.list'.

The Active List

The active list, in the form of the file **rules/active.list** is checked in to SVN trunk, and is used by **build/mkrules** to determine which rules are active.

- Rules marked active, and their dependencies (if they're meta rules), are written to **rules/72_active.cf**;
- Rules which are not in the active list, but which were loaded from a sandbox directory in *rulesrc/sandbox/**, are written to **rules/70_sandbox.cf**, and renamed to always include a `T_` prefix;
- Rules which are not in the active list, but which were loaded from a core-rules dir in *rulesrc/core*, are written to **rules/70_inactive.cf**.

More info on these rule states can be found at RuleLifeCycle.

build/mkrules attempts to keep the contents of those files consistent. In other words, meta subrules for a rule in the active list will always likewise be copied to the active file – even if they are sandbox `T_` rules. It will also ignore rules that are inside an `ifplugin` block for a plugin that is not available.

rules/70_sandbox.cf and **rules/70_inactive.cf** are used for development, and for mass-checks, but will not be installed during `make install` and are not packaged in sa-update tarballs. Rules in **rules/70_sandbox.cf** are the only ones used in the "bbmass" preflight mass-checks.

The sa-update Tarball

build/mkrules is run, using the active list to create **rules/72_active.cf**, **rules/70_sandbox.cf** and **rules/70_inactive.cf**. The latter two files are then discarded, and `spamassassin --lint` run to ensure the ruleset is still in a consistent state. If lint fails, the update process is abandoned.

A `make install` is run, to a temporary directory, and the resulting ruleset is packaged using tar, linted again from that dir, and then signed and checksummed.

The DNS zone is updated using the SVN revision number, and the tarball copied into place.

Administrivia

The cron jobs run as the user *updatesd*. The crontab looks like:

```
30 8 * * * bash /home/updatesd/svn/spamassassin/build/mkupdates/run_nightly > /var/www/buildbot.spamassassin.org
/updatesd/mkupdates.txt 2>&1
50 8 * * * bash /home/updatesd/svn/spamassassin/build/mkupdates/run_part2 > /var/www/buildbot.spamassassin.org
/updatesd/mkupdatespt2.txt 2>&1
```

SA-Update Mirrors

sa-update uses the MIRRORED.BY file to provide distributed update capabilities. The setup for a mirror is described in SaUpdateMirrorSetup.

The Other End

See RuleUpdates for details on running "sa-update".