RescoreMassCheck320

Rescore Mass-Check

(see RescoreMassCheck310 for the 3.1.x historical page, or RescoreMassCheck for the instructions for future builds)

This is the procedure we use to generate new scores. It takes quite a while and is labour-intensive, so we do it infrequently.

We generate new scores by analyzing a massive collection of mail (a "corpus"), and running software to create a score-set that gets the best possible set of scores, so that the maximum possible number of mails in that corpus are correctly classified (ie. so that SA thinks the ham messages are nearly all ham, and the spam messages are nearly all spam).

Summary

The corpus consists of many (approximately 1 million) pieces of real-world, hand sorted mail.

A smallish number of people (about 7), including some of the developers themselves, work as volunteer "corpus submitters". They hand-classify their mail and then run mass-check over it. They submit the output logs mass-check generates. Occasionally people review the submitted logs for obvious mistakes, but it is largely a trust system.

If you want to see the statistics from the last corpus run, check the STATISTICS.txt files that come in the SA tarball. It will tell you how many emails were used, and what the hit rates of all the rules were.

Procedure

Here's the process for generating the scores as of SpamAssassin 3.2.0:

1. heads-up

Inform everyone in advance on the users and dev lists that we will be starting mass-checks shortly, and they should get their corpora nice and clean (see CorpusCleaning) and sign up for RsyncAccounts.

Enable all rules using the helper script to do this:

```
  masses/enable-all-evolved-rules < rules/50_scores.cf  \
                          > rules/51_newscores.cf
  mv rules/51_newscores.cf rules/50_scores.cf
  svn diff     [and ensure it looks sane]
  svn commit   [create a new bug attachment for review if in R-T-C mode]
```

Copy the nightly-log-submission rsync accounts to the rescore-log-submission accounts (see RsyncConfig) (not clear why we don't just use one set of accounts here, but hey):

```
ssh spamassassin.zones.apache.org
sudo cp /home/corpus-rsync/secrets /home/corpus-rsync/secrets-submit
```

Move the old rescore logs from the previous release (if they're still around) to the archives:

```
ssh spamassassin.zones.apache.org
cd /home/corpus-rsync
OLDVERSION="3.1"
sudo mv corpus/submit scoregen-$OLDVERSION
sudo mkdir corpus/submit
sudo chown rsync corpus/submit
sudo gtar cvfz ARCHIVE/scoregen-$OLDVERSION.tgz scoregen-$OLDVERSION
```

1.5. build a mass-checker tarball

Build a mass-check snapshot tarball, as follows:

```
svn export http://svn.apache.org/repos/asf/spamassassin/trunk mcsnapshot
tar cvfz mcsnapshot.tgz mcsnapshot

svn cp \
        https://svn.apache.org/repos/asf/spamassassin/trunk \
        https://svn.apache.org/repos/asf/spamassassin/tags/3_2_0_mcsnapshot_1
```

(we can't use the standard build process here anymore since the dist tarball no longer includes "masses". Use a descriptive, unique tag name.)

2. announce mass-check

RescoreDetails is the full announcement text (and instructions) for this phase. It's sufficient just to send out a mail something like the one we used in 3.1.0:

```
To: users
Cc: dev
Subject: NOTICE: 3.2.0 rescoring mass-checks

OK, if you're planning to send us mass-check logs for the
3.2.0 rescoring, now's the time!

http://wiki.apache.org/spamassassin/RescoreDetails has all
the details.

cheers!

--j.
```

3. allow several days to complete (it takes a really long time!)

Provide enough time, at least a week including a weekend if possible, giving people enough time to get around to running it given that they may be busy with day-job stuff. 😉

4. close up the rsync site

Before the deadline, send out reminder mails to users and dev – if anyone's forgotten all about it, they have a final chance to pipe up at that point. finally, clean up the backup files and make the submission area read-only (in other words, closed for new data):

```
ssh spamassassin.zones.apache.org
cd /home/corpus-rsync/corpus/submit
sudo rm -f *~           [clean up backups]
sudo chmod a-w *.log .  [make the dir and files read-only]
```

That ensures that the data isn't going to change under your feet.

We then take the log files rsync'd up to the server, and use those logs for all 4 score sets. The initial logs are for score set 3 (the fourth), sets 0, 1, and 2 can be generated from set 4 by stripping out the network tests and/or the Bayes tests.

4.1. filter out too-old logs

```
ssh spamassassin.zones.apache.org
cd /home/jm/ftp/spamassassin/masses    [or wherever]

./log-grep-recent -m 38 /home/corpus-rsync/corpus/submit/ham-*.log > ham-full.log

./log-grep-recent -m 6 /home/corpus-rsync/corpus/submit/spam-*.log > spam-full.log
```

We may have to tweak the number of months specified for each type, if there's too much or too little mail resulting from the grep. but 38 months / 6 months worked well for 3.2.0.

4.2 tweak rules for evolver

Go through the rulesrc dir, comment out all "score" lines except for rules that you think the scores are accurate like carefully-vetted net rules, or 0.001 informational rules.

4.3 resync to mcsnapshot rules list

Resync the active rules list to the "active" set as it was in the mass-check snapshot, required since rules/active.list is regenerated every night! Note: if you've made changes to the ruleset that mean you can't use the same set of active rules now, you have a big problem...

We don't just use the entire "rules" dir as it was back then, since every time we run the evolver we first have to fix a few minor bugs in the "rules" files – e.g. scores marked as unmutable in 50_scores.cf incorrectly etc.

```
cd /path/to/checkout/of/trunk
svn co \
  https://svn.apache.org/repos/asf/spamassassin/tags/3_2_0_mcsnapshot_1/rules \
  rules-mcsnapshot
cp rules-mcsnapshot/active.list rules/active.list
make
```

Remove the sandbox ruleset so the evolver doesn't trust them:

```
mv rules/70_sandbox.cf 70_sandbox_off.cf
```

5. generate scores for score sets

See RunningGa. (in the past we used RunningPerceptron, but it acted up during 3.2.0 generation, so we used the GA again.)

Once this is complete, rules/50_scores.cf will have the generated scores, created by runGA. (TODO: I think.)

Set aside the randomized logs set created by runGA, for use in later statistics-generation steps, because they are effectively the "source code" for the rescore run:

```
  cd masses
  tar cvfz rescore-logs.tgz gen-set{0,1,2,3}-*
```

(use "gtar" on the solaris zone.)

These can be pretty big (although nowadays the scripts using hard links for the duplicate logfiles, which saves a lot of space).

Also, check in the "config" files you used for each scoreset:

```
  svn commit -m "runGA config files used" masses/config.set*
```

6. upload the test logs to zone

Since stuff like the STATISTICS cannot ever be regenerated without the (randomised) test logs, these need to be saved, too. Currently, I think the best bet is to upload the `rescore-logs.tgz` file somewhere on spamassassin.zones.apache.org; it doesn't have to be in a public place, ASF-committer-account-required is fine. Just mention that path in the rescoring bug's comments. last time, I did this:

```
sudo mkdir /home/corpus-rsync/ARCHIVE/3.2.0
sudo mv rescore-logs.tgz /home/corpus-rsync/ARCHIVE/3.2.0/rescore-logs-bug5270.tgz
```

6.5. mark evolved-score rules as 'always published'

Normally, rules in the sandbox are promoted to the "active" 72_active.cf ruleset, or demoted to the "test" 70_sandbox.cf ruleset, based on their accuracy in the nightly mass-checks. However, now that the evolver has assigned scores for them, they need to be always published regardless of how they might do in the previous night's checks. Run:

```
  cd masses
  ./force-publish-active-rules ../rules/active.list ../rulesrc/10_force_active.cf
  svn commit -m "force publish of rescored rules" ../rulesrc/10_force_active.cf
```

6.6. fix test failures

Run `prove -v t/basic_lint.t` and `prove -v t/meta.t`. Manually edit the rules files to fix any test failures caused by the new scores. For example, some meta rules may now depend on rules that have been assigned a score of 0; either make those rules into __SUBRULES, or give them a score of 0.001.

7. upload proposed new scores

Attach the new proposed 50_scores.cf as a patch to the rescoring bug on the bugzilla, for voting and comments. There will always be comments 😉

```
  cd ..
  svn diff rules/50_scores.cf > ~/newscores.diff
  [upload ~/newscores.diff in your web browser]
```

Then wait for votes...

8. Make the stats files

Once the scores are voted on and tweaked to everyone's satisfaction, you'll need to rebuild STATISTICS files with the new scores. First, (just to make sure you're in sync!) repatch your scores file to match what's been voted on:

```
  svn revert rules/50_scores.cf
  wget -o newscores.diff http://bugzilla.spamassassin.org/....attachment?id=....
  patch -p0 < newscores.diff
```

Run these commands:

```
  cd masses
  cp config.set0 config ; bash ./runGA stats
  cp config.set1 config ; bash ./runGA stats
  cp config.set2 config ; bash ./runGA stats
  cp config.set3 config ; bash ./runGA stats
```

There'll be a lot of output along these lines:

```
ignoring 'TO_ADDRESS_EQ_REAL': immutable and score == 0
```

But that can be ignored.

8. upload new stats files

Attach the new proposed STATISTICS*.txt as a patch to the rescoring bug on the bugzilla:

```
  cd ..
  svn diff rules/STAT* > ~/newstats.diff
  [upload ~/newstats.diff in your web browser]
```

And let all and sundry vote on that, too (or just check it in depending on whether you're in R-T-C or not). Once the new scores and STATS files are approved and into SVN, the log data is in a safe archival spot on the zone, the bugzilla bug notes that location, and the "config" files are checked in, you're done.