UsingNetworkTests

Using network tests to increase accuracy

SpamAssassin supports several optional components to increase accuracy. Along with Bayes training, its set of 'network tests' is key if you want to improve your hit-rate. In testing, it typically **halves the false negative rate** – in other words, it means that users will see half as many missed spams.

By default, most installations of SpamAssassin don't turn on the network tests, since they impose a small delay on each message as it passes through the filter. Since this increases the **system memory load** (although it will not increase the **system CPU load**), it's something that could cause trouble on high-volume sites. As such, it requires a little thought by the admin before it's turned on, so it's not simply turned on 'out of the box' as a result.

Network tests and load

Here's the possible danger.

Normally, in the non-network-test case, a typical average scan time per message is under 0.5 seconds.

In the network-test case, when a message is scanned by SpamAssassin, network test queries are sent to various servers on the internet; the SpamAssassin engine will then **wait for replies** to those queries, and this can take up to 15 seconds (with a typical average of about 2 seconds per message).

At first glance, it appears that this will greatly slow down scanning. However, that's not the case; this does not happen in serial (one message after another). Instead, multiple messages can be queried in parallel (several messages scanned at the same time, and waiting for responses to their network queries).

Now, compared to the non-network-test case, this increases memory load on the scanning system, since there is now a higher number of scanner processes running. However, it will not increase the system CPU load, since those processes are idle.

In most situations, this will work fine. Since the system CPU load is no higher, your machine is unlikely to get bogged down – assuming you have enough memory installed to cope with the additional processes active. If you're not running the `spamd` server, you should be just fine anyway. (This is only likely to be a problem on high-load sites, and if you're seeing high-load, you should be running spamd anyway!)

If you are running spamd – after turning on network tests, you should monitor how many `spamd` processes are running. Compare against the setting for the spamd `-m` switch (which controls the maximum number of spamd children allowed). If you see it hitting the limit under normal mail load, you need to increase the `-m` setting – and possibly add more RAM or swap space to cope with the higher memory load.

How to turn on network tests

Edit your spamd start-up script, or start-up options file (depending on which OS you're running, these may be different). There should be a `-L` or `--local` switch in that file. Remove it to enable network tests.

Tuning network tests for large sites

For high-volume sites, you may wish to turn on a subset of the network tests. Some of those rules are faster than others!

DNS Blocklists (DnsBlocklists)

The RCVD_IN_* rules are fastest of all the network tests. These also include a very reliable timeout system, so are safe under almost any load.

For DnsBlocklists and URIDNSBL rules (below), you should run a local CachingNameserver on the same machine (or at the least, the same LAN) as the scanner, to increase speed.

URIDNSBL Rules (including SURBL lookups et al)

These perform a high number of DNS lookups, so make sure you have a local CachingNameserver, and be prepared to see scan times increase by a second or two on average.

Collaborative spam identification databases (HashSharingSystem)

SpamAssassin includes support for three different Hash-based Network Tests, which compare fingerprints for received messages against shared lists of previously-seen spam messages.

- The Razor database http://razor.sourceforge.net by Vipul Ved Prakash and Jordan Ritter allow Unix clients to work out of the same database used by the commercial customers of the Cloudmark system. See UsingRazor for help on using Razor.
- Pyzor, a completely free database and software system, written by Frank Tobin. See UsingPyzor for help or visit http://pyzor.sourceforge.net for more information.
- DCC, the Distributed Checksum Clearinghouse http://www.rhyolite.com/anti-spam/dcc/ . See UsingDcc for help with DCC issues.

Razor and Pyzor, in the past, have intermittently had outages that can result in timeouts and higher system load. Set the `razor_timeout` and `pyzor_timeout` values to a low value, if you're worried about this.

If you're a very large site, processing upwards of tens of thousands of messages a day, the DCC maintainers have requested that you consider setting up your own DCC server as described in dccd(8), and arrange to peer with the rest of the public servers, to reduce their load.