TxRep

The TxRep - Reputation Plugin

Introduction

TxRep was designed as an enhanced replacement of the AutoWhitelist plugin

TxRep, just like AWL, tracks scores of messages previously received, and adjusts the current message score, either by boosting messages from senders who send ham or penalizing senders who have sent spam previously. This not only treats some senders as if they were whitelisted but also treats spammers as if they were blacklisted. Each message from a particular sender adjusts the historical total score which can change them from a spammer if they send non-spam messages. Senders who are considered non-spammers can become treated as spammers if they send messages which appear to be spam. Simpler told TxRep is a score averaging system. It keeps track of the historical average of a sender, and pushes any subsequent mail towards that average.

The most important difference of TxRep in comparison to AWL is its ability to learn. It can be trained by sa-learn, it has also the auto-learn ability, and old messages can be re-learned anytime to adjust historical records after a revision of rules. There are more differences, though. Below, there is a brief list of features introduced by TxRep to work around some shortcomings of AWL:

1. Improved scoring algorithm 2. Learning 3. Auto-Learning 4. Re-learning 5. Record Aging 6. Blacklisting and Whitelisting 7. Complex Sender Identification 8. Message Tracking 9. Dual User and Global Storages 10. Outbound Whitelisting

More details are available on the TxRep POD page.

How Does It Work?

The algorithm works using a local database of entries. Each entry has a key formed by the identificator, and optionally the IP address it originated at, and the DKIM signature. It contains a TOTAL score of messages and a COUNT of messages. The MEAN score is TOTAL/COUNT. Each sender is identified by several IDs: the From email address in combination with the originating IP block (or DKIM signature, or SPF pass, if available), the standalone From email address (without any IP), the domain name of the From address, the full IP address, and the HELO name of the originating client. Each of these ID types has a configurable weight factor when calculating the overall sender's reputation. The overall reputation score is calculated using the formula shown below:

```
sender_reputation = txrep_weight_email_ip * email_ip_reputation +
                    txrep_weight_email    * email_reputation    +
                    txrep_weight_domain   * domain_reputation   +
                    txrep_weight_ip       * ip_reputation       +
                    txrep_weight_helo     * helo_reputation
```

The default values of the weight factors:

- {{txrep_weight_email_ip = 10 }}(of total 19.5, hence 51%)
- {{txrep_weight_email = 3 }}(of total 19.5, hence 15%)
- {{txrep_weight_domain = 2 }}(of total 19.5, hence 10%)
- {{txrep_weight_ip = 4 }}(of total 19.5, hence 21%)
- {{txrep_weight_helo = 0.5 }}(of total 19.5, hence 3%)

Depending on configuration, TxRep uses either a global storage to keep the reputation record (same for all users), or a User storage (a separate storage for each user ID that can run SpamAssassin). Alternatively, when the `txrep_user2global_ratio` is enabled, both storages are used concurrently. When both storages are used, each of the two reputations are calculated in the same way as shown above, using sender values from the respective storages (when available), and ten the overall reputation is calculated with the following formula:

```
total reputation = ( txrep_user2global_ratio * user + global ) / ( txrep_user2global_ratio + 1 )
```

The default value of `txrep_user2global_ratio` is 0 (dual storage disabled). The setting takes values between 0 and 10. The value around 2 may be a good starting point when enabling the feature (user storage reputation has twice the weight of the global reputation). Before enabling the dual storage, make sure your system is configured to call SpamAssassin under the respective user id. In many installations, SA is always called with the same user id. In such cases, activating the dual storage would be useless.

The overall txrep_factor can be adjusted in the configuration to adjust the impact of the reputation, which may be useful when starting off. The value of the corrective TXREP tag is calculated in the following way:

```
corrected score = current score + txrep_factor * (reputation + current score)/(count+1)
TXREP tag value = corrected score - current score
```

The default value of the `txrep_factor` is 0.5, but unlike at AWL, the final result is also depending on the count of recorded messages of given sender. In the result, the factor of 0.5 is equivalent to the AWL factor of 0.25 at senders with one record, and its influence rises close to the projected value of 0.5 logarithmically with the number of sender messages recorded.

Additionally to the algorithms shown above, the reputation is also influenced by the `txrep_dilution_factor`. This factor was introduced to help wearing out the influence of old records. When the factor is used, the new score will always have a slightly higher weight than the stored values. It means that the influence of old records progressively drops with each new message from the sender. The formula below is used:

```
newtotal = (oldcount + 1) * (newscore + txrep_dilution_factor  * oldtotal) / (txrep_dilution_factor * oldcount
+ 1)
```

The default value of the `txrep_dilution_factor` is 0.98, and it takes values between 0.7 (fast dilution / expiry), and 1.0 (no dilution at all).

The schema txrep-diagram.gif demonstrates the calculation of the TXREP tag value.

How do I train spam/ham?

In exactly the same way (and in the same time) as you train spam and ham to the Bayesian SA system:

- `sa-learn --spam file`
- `sa-learn --ham file`

It means that if your server is set up to use IMAP folders, webmail, or other tools for training SpamAssassin, the TxRep reputation will be adjusted at senders of all the trained messages anytime you use it. More details about the algorithm are available in TxRep POD.

Additionally, TxRep, in similar way to the Bayes plugin, can boost the automated whitelisting/blacklisting at the scan time, when the score of the message triggers the auto-learn process. TxRep will add or subtract additional points to the stored reputation in such case. To activate this feature, you need to enable the option txrep_autolearn. Do not activate the auto-learn option before SpamAssassin is well tuned, and before it sorts spam and ham correctly. With poorly trained SpamAssassin, the auto-learn function of TxRep would boost also all false results. Add the `autolearn` value to the email headers (i.e. `"add_header all Status ... autolearn=`*AUTOLEARN*`"` in `local.cf`), and activate the `txrep_autolearn` option only after you verified that SA triggers the autolearn process only in the cases when you clearly want to boost the sender's reputation (in one or the other way).

How do I whitelist/blacklist someone?

See ManualWhitelist for different options available in SpamAssassin. With TxRep, the blacklisting/whitelisting can be done manually with the help of the following command line options of SpamAssassin:

- `spamassassin --add-addr-to-blacklist=foe@spam.biz`
- `spamassassin --add-addr-to-whitelist=friend@ham.org`

It is necessary to understand that whitelisting/blacklisting through TxRep is not the same as whitelisting/blacklisting in a cf file, using the `whitelist_from` or `blacklist_from`
directives. TxRep whitelisting/blacklisting adjusts the reputation of the plain email address by a high score (details can be found in TxRep POD). This blacklisted or whitelisted reputation score can wear out over time, as scores of new messages from the sender are added to the total reputation score.

Besides whitelisting/blacklisting of email addresses, in the same way also domain names, IP addresses, and dot-less HELO names can be whitelisted or blacklisted. For example:

- `spamassassin --add-addr-to-blacklist=spamming.biz`
- `spamassassin --add-addr-to-whitelist=12.123.12.234`
- `spamassassin --add-addr-to-whitelist=1234:abcd:1111:12::3`
- `spamassassin --add-addr-to-blacklist=foe-pc`

Please note that when blacklisting/whitelisting an email address or domain, all records of the address or domain bound to certain IP address, DKIM signature, or an SPF pass, will be removed from the database, and only the plain record (not bound to any specific IP address) is kept. When whitelisting /blacklisting an email address or domain name, you can bind them to a specified DKIM signature or an SPF record by appending the DKIM signing domain or the tag 'spf' after the ID in the following way:

- `spamassassin --add-addr-to-blacklist=spamming.biz,spf`
- `spamassassin --add-addr-to-whitelist=friend@good.org,good.org`

When a message contains both a DKIM signature and an SPF pass, the DKIM signature takes the priority, so the reputation record bound to the 'spf' tag won't be checked. You can whitelist both variants - the email address with DKIM signer, and with the 'spf' tag. When white/blacklisting also the plain email address (or domain name), do it always before adding the variants with a signature and/or with SPF, because when adding a plain record, all other records associated with that address are removed from the reputation storage.

Only email addresses and domain names can be bound to DKIM or SPF. Records of IP addresses and HELO names are always stored without DKIM/SPF.

TxRep can also automatically whitelist all recipients of outgoing email. To enable this feature, set the option txrep_whitelist_out. Please note that this feature can only work when SpamAssassin processes outbound email too, and, of course, it does not work for email sent through 3rd party SMTP servers.

Database Storages & Utilities

TxRep uses the same storage handlers as its predecessor AWL, therefor TxRep Berkeley DB format backend files can be examined, pruned, and manipulated with the same tools as at AWL. See also the AWL page for some more details

- 'sa-awl' - simple tool available in the distribution package
- 'sa-heatu' v1 - external enhanced tool
- 'sa-heatu' v2 - newer, enhanced version

When using the SQL storage type, multitude of SQL tools can be used for the same purpose - for example PhpMyAdmin for MySQL, PhpPgAdmin for Postgresql, etc.

Although requested, there is currently no Redis storage handler available for AWL or TxRep, but MySQL storage tuned with the MEMORY engine, or InnoDB engine with a sufficiently big innodb_buffer_pool parameter, or together with the MySQL memcache plugin, would offer similar performance as Redis, while allowing much better vertical and horizontal scalability (it would work better for both bigger tables and multiple concurrent accesses as well).

Contributors

- Ivo Truxa