

# AutoWhitelist

## The Auto-WhiteList

In March 2014 (rev 1579980), [SpamAssassin](#) introduced the [TxRep](#) plugin. The new plugin enhances the functionality of AWL, and works around some of its shortcomings.

### Introduction

Tracks scores from messages previously received and adjusts the 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.

A sender is identified using both their address and the most significant 2 octets of their IP address (ex: mailer-daemon @club-internet.fr 194.158) This is intended to penalize spam claiming to be From you with forged headers.

Sometimes it may appear that the AWL is assigning scores the wrong way; read [AwlWrongWay](#) for details.

A simple system is score averaging. It keeps track of the historical average of a sender, and pushes any subsequent mail towards that average.

### How Does It Work?

The algorithm works using a local database of entries. Each entry has a key formed by the From: address of the mail, and the IP address it originated at. It contains a TOTAL score of messages and a COUNT of messages. The MEAN score is TOTAL/COUNT. The algorithm:

1. Compute the SCORE of the message from RULES
2. Compute DELTA as (MEAN-SCORE)\*auto\_whitelist\_factor
3. Increment historical TOTAL by SCORE
4. Increment historical COUNT by one
5. Set the final score of the message to SCORE+DELTA

auto\_whitelist\_factor can be adjusted in the configuration to vary the impact of the MEAN, and may be useful when starting off.

For example:

The first mail received from a sender scores 20 based on RULES. When they send you a second message that scores 2.0 based on RULES, the MEAN will push the score up to 11 ( 20/1 - 2.0 ) on the second mail. This is auto blacklisting, based on their past history of spam.

If that same person sent an mail that scored 0, and then later sent one that scored 7, the AWL would push the score down to 3.5. This is auto-whitelisting based on past history of nonspam.

### How do I whitelist/blacklist someone?

Not with this! This is a automatic heuristic system. You want to use the manual whitelist/blacklist. See [ManualWhitelist](#).

### Previous version implementation

- In 3.3, the plugin is not loaded by default.
- In 3.1, it is a plugin, loaded by default in the file `v310.pre`.
- In 3.0, it is enabled by default and turned off with the configuration setting `use_auto_whitelist 0`.
- In earlier versions, it is enabled using the `-a` flag to `spamassassin` or `spamd`.

### Utility

The auto-whitelist database (when using the DB Format backend) can be examined and pruned using the program 'sa-awl'. NOTE: This was incorrectly written as sa-awlUtil in previous documentation.

---

An enhanced version of that utility is available at

<http://www.real-world-systems.com/mail/sa-heatu.html> <http://www.Real-World-Systems.com/mail/sa-heatu.tar>

This tar file contains only the perl source and a small readme file with some usage examples.

Spam Assassin - \*H\*uristic \*E\*mail \*A\*ddress \*T\*racker Utility  
Includes the ability to remove a entries for a single email address.  
This is useful when a spammer sends one or more ham messages before sending spam.

Newer version is available at <http://truxoft.com/resources/sa-heatu.v4.02.tar.gz> Besides other enhancements, it offers the possibility to dump, expire, or delete records by multiple criteria, including regexes. There is no online document describing all the new features, but the command line help and the integrated changelog give detailed information about them (sa-heatu -h resp. sa-heatu --changelog).

---

### Discussion

DGerman says: " Notice that I have avoided the use of the string **autowhiteList** where possible. Even if "we were there first", this it is more than whitelisting."

TODO: verify example calculation, verify `auto_whitelist_factor` and show default

Most other products use an **AutoWhitelist** as *automation* of a *manual whitelist*, which is created and/or maintained by monitoring outgoing email to **add entries to a whitelist** without a need for human intervention. In essence, rather than adjusting a *sender's average score*, as in **SpamAssassin**, most other products literally *automate* the addition of names to the manual whitelist. Because **autowhitelist-ing** is an industry standard term for automation of a **whitelist**, this feature in **SpamAssassin** should be re-named to better align with what it actual does, such as maybe **AutoReScore**? In essence, be aware the term *AutoWhitelist* means something else on most (if not all) other anti-spam products. – JDR

**SpamAssassin** has been using the term "autowhitelist" for this feature since about 2001 – I think we were there first 😊 – jmason

From <http://wiki.apache.org/spamassassin/ManualWhitelist> "the name is a misnomer. The **AutoWhitelist** is designed as an automatic score averaging system." Maybe score averaging system? -Roger

It seems to me that this would be the place to document the default value of **auto\_whitelist\_factor**. I gather from other sources that it's 0.5. – [DanielNorton](#)

Yes, the default `auto_whitelist_factor` is indeed 0.5, as described in the documentation at [Mail::SpamAssassin::Plugin::AWL auto\\_whitelist\\_factor](#). – [Adam Katz](#)

As to how it faces users, the description can be overridden in the configuration. My ([Adam Katz](#)) production systems redescribe AWL so as to remove the confusion:

```
describe AWL Adjust score towards average for this sender
```