


```

| Anti-Spam Settings thresholds |
| sa_tag_level |
| sa_tag2_level <-- cutoff points |
| sa_kill_level set there |
+-----+

```

Score Overrides tunes the **contributions**; Anti-Spam Settings tunes the **cutoffs**. A message reaches `quarantine` because the sum of contributions crosses the cutoff — moving either side of that equation changes behavior, and they are independent knobs.

What an override actually changes

Override value	Effect on the rule	Use it when
Positive (e.g. <code>3.5</code>)	Adds more to the spam score on match	A rule catches a genuine pattern your senders see often but the default score is too low to flag
<code>0</code>	Rule still runs but contributes nothing	A rule produces too many false positives in your mail mix and you want to neuter it without ripping it out of the database
Negative (e.g. <code>-2.0</code>)	Subtracts from the spam score on match	The rule indicates legitimacy in your environment (e.g. a trusted-relay heuristic) and you want it to act as a bonus

Setting a score to `0` is the safe equivalent of "disable this rule" — SpamAssassin still evaluates it (so the test name still appears in `X-Spam-Status` and you can confirm it fired), but the message total is unaffected. Removing the override does not delete the underlying SpamAssassin rule; it only stops Hermes's `local.cf` from overriding the shipped default.

The page

A collapsible scoring helper (the same text the operator gets in the in-page guide), a hard-locked "DKIM and SPF rules are not evaluated" warning, an Add Override modal, a DataTable of current overrides, and an Edit / Delete modal pair.

Add Override modal

Field	Stored as	Notes
-------	-----------	-------

Test Name	<code>spam_settings.parameter</code>	The SpamAssassin rule name, uppercase with underscores (e.g. <code>BAYES_99</code> , <code>HTML_MESSAGE</code> , <code>FREEMAIL_FROM</code>)
Score	<code>spam_settings.value</code>	Numeric, validated <code>-999 <= value <= 999</code> . Set to <code>0</code> to neuter the rule
Description	<code>spam_settings.description</code>	Free-text label that surfaces in the DataTable; optional

Add validates: Test Name non-blank, Score numeric and in range, the `(parameter)` natural key not already present, and the rule name not in the SPF / DKIM / ADSP plugin family (see warning below). On success: INSERT row with `spamfilter='1'`, `active='1'`, `applied='1'`; then immediately regenerate `local.cf` and reload the engine — same chain Save uses.

Score Overrides DataTable

Column	Source
(checkbox)	Selection for bulk Delete Selected
Test Name	<code>spam_settings.parameter</code>
Score	<code>spam_settings.value</code>
Description	<code>spam_settings.description</code>
Edit	Per-row pencil button -> Edit modal

System-managed rows (`system_managed = 1`) get a lock icon instead of a checkbox, a "System-managed" badge next to the test name, and a disabled Edit button. They are filtered out of any DELETE generated by the page even if a forged POST targets them (`AND system_managed = 0` is part of the delete query). The lock exists for rules that encode a Hermes architectural decision — for example, the per-rule scores Hermes maintains for the trusted-relay Return Path lookups.

Edit Modal

Test Name is read-only — changing it is semantically a different rule and would orphan the override. Only Score and Description are editable. Save runs the same regen + reload chain as Add.

DKIM / SPF / ADSP overrides are silently meaningless

The page mounts a warning callout flagging that **any override targeting a DKIM, SPF, or ADSP rule has no effect** in Hermes, and the Add handler rejects them with alert `m = 13`. The rule families covered:

- `DKIM_*` (e.g. `DKIM_INVALID`, `DKIM_VALID`, `DKIM_ADSP_ALL`)
- `SPF_*` (e.g. `SPF_PASS`, `SPF_FAIL`, `SPF_HELO_SOFTFAIL`)
- Any rule whose name contains `ADSP`

The SpamAssassin DKIM and SPF plugins are intentionally not loaded in Hermes's `init.pre` — the authoritative DKIM verdict is the `Authentication-Results:` header that OpenDKIM writes at `:25`, and the authoritative SPF verdict is the `Received-SPF:` header that `postfix-policyd-spf-python` writes at envelope time. SpamAssassin's in-content re-check would otherwise produce false-positive failures against Hermes-modified bodies (External Sender Banner, disclaimer, signature insertion) and could pick up the wrong upstream IP from the Received chain in multi-hop scenarios (federal mail, M365 GOV cloud, etc.). Letting an operator write an override for a rule that literally cannot fire would silently mislead them, so the guard runs at the Add handler.

The block is case-insensitive (`UCase` + `Left` / `FindNoCase`) so mixed-case rule names cannot sidestep it.

Save and apply flow

1. View page submits `action="add" | "edit" | "delete"`
2. `view_score_overrides.cfm` validates the row (per-action rules above)
3. `INSERT / UPDATE / DELETE` on `spam_settings` (`spamfilter='1'`), guarded by `system_managed=0` on `UPDATE` and `DELETE`
4. `update_spamassassin_config_files.cfm`:
 - a. Read `/opt/hermes/conf_files/local.cf.HERMES` (template)
 - b. Substitute `USE-BAYES`, `USE-DCC`, `USE-PYZOR`, `USE-RAZOR2`, and `bayes_auto_learn` placeholders from their own `spam_settings` rows
 - c. `SELECT` every `spamfilter='1' active='1'` row -> `tmp/_sa_tests` file:

```
score <parameter> <value>
(one line per row)
```
 - d. Substitute the `#CUSTOM-TESTS` placeholder in `local.cf` with the rendered score list
 - e. Render Message Rules into the `#CUSTOM-MESSAGE-RULES` placeholder
 - f. Back up `/etc/spamassassin/local.cf` -> `local.cf.HERMES.BACKUP`, move the rendered file into place
 - g. `UPDATE spam_settings SET applied='1' WHERE applied='2'`
5. `update_amavis_config_files.cfm`:
 - Regenerate Amavis 50-user from template (subject tags, destinies,

DKIM-verification toggle, file rules) so a SA setting change that also affects Amavis takes effect in the same write

6. restart_spamassassin.cfm:

- docker exec hermes_mail_filter /usr/bin/spamassassin --lint (validation; abort on failure)
- Then docker container restart hermes_mail_filter

7. restart_amavis.cfm: same docker container restart hermes_mail_filter (idempotent; the engine is back from step 6)

8. session.m = 1 / 7 / 8 -> success alert with "regenerated" wording

The restart in step 6 is a full container restart — `hermes_mail_filter` runs SpamAssassin, ClamAV, Amavis, and Fangfrisch, all of which re-initialize together. Inbound mail held in Postfix's queue during the restart is retried on the next queue run; no message is lost.

Failure semantics

Alert	Trigger
<code>m = 1</code>	Add succeeded and SpamAssassin reloaded
<code>m = 2</code>	Test Name blank
<code>m = 3</code>	Test Name already exists
<code>m = 4</code>	Score out of <code>-999..999</code> range
<code>m = 5</code>	Score blank
<code>m = 6</code>	Score not numeric
<code>m = 7</code>	Edit succeeded and SpamAssassin reloaded
<code>m = 8</code>	Delete succeeded and SpamAssassin reloaded
<code>m = 10</code>	Delete clicked with no rows selected
<code>m = 11</code>	The Apply chain (regen + restart) threw — DB write may already have happened
<code>m = 12</code>	Attempt to edit or delete a <code>system_managed = 1</code> row (forged POST defense; the UI hides the action)
<code>m = 13</code>	Add of a DKIM / SPF / ADSP family rule — rejected because the underlying plugin is disabled

`m = 11` is the partial-failure case: the DB row has already been inserted / updated / deleted but `local.cf` regen or the lint / restart step failed. The page does not roll back the DB write — the next successful save will re-render `local.cf` from the current table state, so the system is self-healing on the next click.

Finding rule names

The page guide gives the lookup steps that work for any received message:

1. From [Message History](#), open any message and view headers; the `X-Spam-Status:` header lists every rule that fired and its score
2. SpamAssassin rule names are uppercase with underscores (e.g. `BAYES_99`, `HTML_MESSAGE`, `FREEMAIL_FROM`, `RDNS_NONE`, `URIBL_BLOCKED`)
3. To see the default score and description for a rule: `docker exec hermes_mail_filter spamassassin --debug rules 2>&1 | grep -i <RULE_NAME>`

Files and containers touched

Path	Owner	Role
<code>config/hermes/var/www/html/admin/2/view_score_overrides.cfm</code>	<code>hermes_commandbox</code>	The page (validation + alerts + DataTable)
<code>config/hermes/var/www/html/admin/2/inc/update_spamassassin_config_files.cfm</code>	<code>hermes_commandbox</code>	Renders <code>local.cf</code> from template + score rows + message rules
<code>config/hermes/var/www/html/admin/2/inc/update_amavis_config_files.cfm</code>	<code>hermes_commandbox</code>	Re-renders Amavis <code>50-user</code> (called in the same chain to keep SA-related Amavis flags in sync)
<code>config/hermes/var/www/html/admin/2/inc/restart_spamassassin.cfm</code>	<code>hermes_commandbox</code>	Lints the new <code>local.cf</code> then restarts <code>hermes_mail_filter</code>
<code>config/hermes/var/www/html/admin/2/inc/restart_amavis.cfm</code>	<code>hermes_commandbox</code>	Calls <code>restart_mail_filter.cfm</code>
<code>config/hermes/opt/hermes/conf_files/local.cf.HERMES</code>	<code>hermes_commandbox</code> (read) -> <code>hermes_mail_filter</code> (live <code>/etc/spamassassin/local.cf</code>)	Canonical template with <code>##CUSTOM-TESTS</code> and <code>##CUSTOM-MESSAGE-RULES</code> placeholders
<code>/etc/spamassassin/local.cf</code>	<code>hermes_mail_filter</code>	Live file SpamAssassin reads at engine start
<code>/etc/spamassassin/local.cf.HERMES.BACKUP</code>	<code>hermes_mail_filter</code>	Pre-write backup taken every save
<code>spam_settings</code> table, <code>spamfilter = '1'</code>	<code>hermes_db_server</code> (<code>hermes</code> DB)	Source of truth for every override (and for the Bayes / DCC / Razor / Pyzor / threshold values used by Anti-Spam Settings)
<code>hermes_mail_filter</code> container	—	Hosts SpamAssassin, ClamAV, Amavis, Fangfrisch — restarted as a unit on every save

Related

- [Anti-Spam Settings](#) — sets the GLOBAL spam thresholds (`sa_tag_level`, `sa_tag2_level`, `sa_kill_level`); the cutoffs that the per-rule contributions tuned here are summed against
- [Message Rules](#) — custom SpamAssassin rules (header / body / regex) written into the same `local.cf` via the `##CUSTOM-MESSAGE-RULES` placeholder during the same regen cycle
- [Antivirus Settings](#) — ClamAV runs in the same Amavis pass; a virus verdict pre-empts any spam-score result
- [Perimeter Checks](#) — SMTP-time rejects that fire before SpamAssassin ever sees the message
- [File Extensions](#) / [File Expressions](#) / [File Rules](#) — Amavis attachment filtering that runs alongside SpamAssassin scoring in the same pass
- [DMARC Settings](#) / [ARC Settings](#) — every rule in the DKIM / SPF family is the authoritative verifier whose verdict the warning callout refers back to
- [Scheduled Tasks](#) — Bayes auto-learn and signature refresh cadence are scheduled here, not on the Score Overrides page
- [System Logs](#) — every rule fire and its score appears in `mail.log` under the `amavis[...]:` lines, prefixed `tests=...`

Revision #48

Created 2026-05-31 12:52:31 UTC by Dino Edwards

Updated 2026-06-20 13:33:18 UTC by Dino Edwards