Antispam Settings

Antispam Settings

Admin path: **Content Checks > Antispam Settings** (`view_antispam_maintenance.cfm`, `inc/get_spam_settings.cfm`, `inc/spam_settings_save.cfm`, `inc/update_amavis_config_files.cfm`, `inc/update_spamassassin_config_files.cfm`, `inc/restart_amavis.cfm`, `inc/restart_spamassassin.cfm`, `inc/antispam_init_pyzor.cfm`, `inc/antispam_init_razor.cfm`, `inc/antispam_clear_bayes.cfm`).

This page configures the SpamAssassin engine that Amavis calls inside `hermes_mail_filter` for every message that clears the SMTP-time perimeter, plus the Amavis-level handling policies that decide what happens to a message once it has been scored or otherwise classified. Per-rule weight adjustments live on Score Overrides; this page is engine settings and quarantine destiny only.

Where SpamAssassin sits in the flow

```
|       DCC / Razor / Pyzor net DBs |
|       Bayes statistical engine    |
|       custom rules + scores       |
|   - banned-file checks            |
|   - final_*_destiny -> quarantine/DSN/discard
+---------------+-------------------+
                |
                v
+----------------------------------+
|  Re-inject -> hermes_postfix_dkim:10026
+----------------------------------+
```

A virus verdict from ClamAV always pre-empts the spam score; the `final_virus_destiny` setting on this page decides what Amavis does with that already-classified virus. The `final_spam_destiny`, `final_banned_destiny`, and `final_bad_header_destiny` settings work the same way for the other three Amavis verdict categories.

Container and tool placement

Component	Detail
Container	`hermes_mail_filter` (IPv4 `.105`)
Engine	SpamAssassin (`spamd` / `Mail::SpamAssassin` Perl modules called from Amavis)
Amavis config	`/etc/amavis/conf.d/50-user` (rendered from `/opt/hermes/conf_files/50-user.HERMES` on every save)
SpamAssassin config	`/etc/spamassassin/local.cf` (rendered from `/opt/hermes/conf_files/local.cf.HERMES` on every save)
Bayes DB	Lives in the SpamAssassin user dir inside `hermes_mail_filter` (`sa-learn --dump magic` reports the actual path)
Network plugin state	`/etc/razor/identity` (Razor), Pyzor's per-user config dir, DCC's local socket — all inside `hermes_mail_filter`
Reload mechanism	`spamassassin --lint` + `docker container restart hermes_mail_filter` on every save

The container exposes **no host ports** — Amavis is reached only by Postfix internally at `hermes_mail_filter:10024` and re-injects to `hermes_postfix_dkim:10026`.

Spam Detection Plugins card

Three boolean toggles enable third-party network-aware spam DBs. Storage: `spam_settings.value` for parameters `use_dcc`, `use_razor2`, `use_pyzor` (each row keyed by `parameter`, value `0` or `1`).

Plugin	What it does	Maintenance action
DCC (Distributed Checksum Clearinghouse)	Fuzzy-checksum bulk-mail detection; matches a message against a network of receivers' checksum counters	None — `cdcc` runs as part of the SpamAssassin call chain
Razor2 (Vipul's Razor v2)	Collaborative spam catalog; checksum + signature lookup against the Razor network	**Initialize Razor** (see Maintenance) before first use
Pyzor	Collaborative digest-based spam detection	**Initialize Pyzor** before first use

Each toggle substitutes into `local.cf` via the placeholders `USE-DCC`, `USE-PYZOR`, `USE-RAZOR2` -> `use_dcc 0|1`, `use_pyzor 0|1`, `use_razor2 0|1`.

> **Operational consequence — network DB connectivity.** All three plugins make outbound queries (DCC over UDP, Razor and Pyzor over TCP) at scan time. If outbound to the public Internet is blocked from `hermes_mail_filter`, the plugins quietly time out per message and add measurable per-scan latency. Disable plugins the gateway cannot actually reach.

Subject Tagging card

Single field, `sa_spam_subject_tag` in `spam_settings`. Substitutes into `50-user` via the `sa-spam-subject-tag` placeholder, which sets Amavis's `$sa_spam_subject_tag`. Default `[SUSPECTED SPAM]`. Required (empty value rejected with error 2). Only applied when `sa_spam_modifies_subj = 1` (a fixed value in `spam_settings`, not exposed in the UI).

Message Handling Policies card

Four radio pairs, one per Amavis verdict category. Each row stores `D_DISCARD` or `D_BOUNCE` in `spam_settings.value` and substitutes into `50-user` via `final-<category>-destiny`. Amavis acts on the value as follows:

Setting	DB row	"Quarantine Only" (`D_DISCARD`)	"Quarantine & Send DSN" (`D_BOUNCE`)
Virus Messages	`final_virus_destiny`	Message goes to quarantine; no DSN	Message goes to quarantine; DSN sent to envelope sender
Banned File Messages	`final_banned_destiny`	Same as above for banned-file matches	DSN sent
Spam Messages	`final_spam_destiny`	Quarantined silently	DSN sent
Bad-Header Messages	`final_bad_header_destiny`	Quarantined silently	DSN sent

The labels are deliberately conservative — `D_DISCARD` does **not** delete the message, it routes it to Amavis's quarantine where Message History can review and release it. Defaults: virus + banned send DSN; spam + bad-header quarantine silently.

> **Operational consequence — Send DSN on spam.** Setting `final_spam_destiny = D_BOUNCE` means Hermes will deliver a non-delivery report to the envelope sender of every quarantined spam. Because the envelope sender is almost always forged on spam, the DSN will either bounce, contribute to backscatter against innocent third parties, or land in a victim's spam folder. The safe default for spam is `D_DISCARD`; reserve DSN for virus and banned-file (where the sender is more likely to be legitimate).

Bayes Database card

SpamAssassin's per-installation statistical learning engine. Three controls, stored in `spam_settings`:

Field	DB row	Substitution placeholder	Effect
Enable Bayes Database	`use_bayes`	`USE-BAYES` -> `use_bayes` followed by `0` or `1`	Master switch; when off, Bayes rules contribute no score
Enable Auto-Learning	`bayes_auto_learn`	`BAYES-AUTO-LEARN` -> `bayes_auto_learn` followed by `0` or `1`	When on, SpamAssassin trains the Bayes DB automatically based on the message's final score relative to the thresholds below
Spam Threshold	`bayes_auto_learn_threshold_spam`	`BAYESAUTOLEARN-SPAM` -> `bayes_auto_learn_threshold_spam <value>`	Final score above which auto-learn treats the message as spam. Must be numeric and in the range `0.01 .. 999`

Field	DB row	Substitution placeholder	Effect
Non-Spam Threshold	`bayes_auto_learn_threshold_nonspam`	`BAYESAUTOLEARN-HAM` -> `bayes_auto_learn_threshold_nonspam <value>`	Final score below which auto-learn treats the message as ham. Must be numeric and in the range `-999 .. -0.01`

The thresholds are SpamAssassin's `bayes_auto_learn_threshold_spam` and `bayes_auto_learn_threshold_nonspam` directives. JavaScript on the page collapses the thresholds when Bayes or auto-learning is disabled.

> **Operational consequence — Bayes poisoning.** Auto-learning trusts the final score (which already includes Bayes's own contribution) to decide whether to train. A bad spam wave that sneaks past the score threshold can train Bayes to think more spam is ham, which lowers detection on the next batch. If detection quality regresses noticeably after enabling auto-learning, use the Clear Bayes Database action and re-train manually or via a known-good corpus before re-enabling.

Save flow

```
1. View page submits action="save_settings" (all four cards in one POST)
2. spam_settings_save.cfm validates:
     - sa_spam_subject_tag non-empty (error 2)
     - if bayes_auto_learn=1:
         spam threshold numeric (error 5), > 0 and <= 999 (error 4),
                 non-empty (error 3)
         non-spam threshold numeric (error 10), < 0 and >= -999 (error 8),
                 non-empty (error 7)
3. On valid input, UPDATEs 13 rows in spam_settings (sa_spam_subject_tag,
   four final_*_destiny, use_bayes, bayes_auto_learn, both thresholds,
   use_dcc, use_razor2, use_pyzor)
4. cfinclude update_amavis_config_files.cfm:
     - Reads /opt/hermes/conf_files/50-user.HERMES
     - Substitutes SERVER-NAME, SERVER-DOMAIN, sa-spam-subject-tag,
       final-{virus,banned,spam,bad-header}-destiny,
       enable-dkim-{verification,signing},
       HERMES-USERNAME, HERMES-PASSWORD,
       FILE-RULES-GO-HERE (from file_rule_components table),
```

```
        DKIM-KEYS-GO-HERE (from dkim_sign table)
      - Backs up /etc/amavis/conf.d/50-user -> 50-user.HERMES.BACKUP
      - Moves rendered file into place
  5. cfinclude update_spamassassin_config_files.cfm:
      - Reads /opt/hermes/conf_files/local.cf.HERMES
      - Substitutes USE-DCC, USE-PYZOR, USE-RAZOR2, USE-BAYES,
        BAYES-AUTO-LEARN, BAYESAUTOLEARN-SPAM, BAYESAUTOLEARN-HAM
      - Appends per-rule score lines (from spam_settings where spamfilter=1)
      - Appends custom message rules (from message_rules table)
      - Backs up /etc/spamassassin/local.cf -> local.cf.HERMES.BACKUP
      - Moves rendered file into place
  6. cfinclude restart_amavis.cfm -> restart_mail_filter.cfm:
      - docker container restart hermes_mail_filter
  7. cfinclude restart_spamassassin.cfm:
      - docker exec hermes_mail_filter /usr/bin/spamassassin --lint
      - docker container restart hermes_mail_filter
  8. session.m = 1 -> green "Anti-spam settings have been saved and applied" alert
  9. cflocation back to view_antispam_maintenance.cfm
```

The same container is restarted twice (once for Amavis, once for SpamAssassin) because the restart includes are intentionally independent helpers used elsewhere; both calls resolve to the same `docker container restart hermes_mail_filter`. Outbound mail queues briefly during the restart cycle (typically a few seconds); Postfix will retry.

Maintenance card group

Three buttons, each running a single `docker exec` against `hermes_mail_filter` and surfacing stdout/stderr to the operator.

Initialize Pyzor

Action handler: `antispam_init_pyzor.cfm`

```
docker exec hermes_mail_filter /usr/bin/pyzor ping
```

Pings the Pyzor servers; success is detected by the literal string `200` in the output. The command both verifies connectivity and writes the per-user Pyzor config the first time it runs. Required before `use_pyzor = 1` returns meaningful results.

Initialize Razor

Action handler: `antispam_init_razor.cfm`

```
docker exec hermes_mail_filter /bin/bash -c \
  'rm -f /etc/razor/identity && razor-admin -create && razor-admin -register'
```

Deletes the existing Razor identity, creates a fresh config, and registers the gateway with the Razor network. Success is detected by `Register successful` or `created` in the output. Re-run if Razor queries start failing (typically after the identity is rotated or the network rejects the existing identity).

Clear Bayes Database

Action handler: `antispam_clear_bayes.cfm`

```
docker exec hermes_mail_filter /usr/bin/sa-learn --clear
```

Wipes the learned spam/ham corpus. SpamAssassin will need to re-learn from scratch before Bayes rules contribute meaningful scores again. Use only when the database is known-poisoned or when migrating between servers without preserving training. The button is gated behind a JavaScript `confirm()` and renders inside a yellow warning card.

Failure semantics

Failure	Behavior
Empty `sa_spam_subject_tag`	session.m=2, red alert, no save
Bayes spam threshold empty	session.m=3
Bayes spam threshold not numeric	session.m=5
Bayes spam threshold <= 0 or > 999	session.m=4
Bayes non-spam threshold empty	session.m=7
Bayes non-spam threshold not numeric	session.m=10
Bayes non-spam threshold >= 0 or < -999	session.m=8
Any cfcatch during the save -> apply chain	session.m=9, red alert with `session.saveError` showing `cfcatch.message`

Failure	Behavior
`spamassassin --lint` failure during restart	`error.cfm` cfabort with the lint failure message; the rendered `local.cf` is already in place but Amavis is not restarted further
Pyzor ping output without `200`	session.m=12, red alert; full output shown in a `<pre>` for diagnosis
Razor init output without `Register successful` or `created`	session.m=14, similar surfacing
Bayes clear `cfcatch`	session.m=16 with the catch message

`spamassassin --lint` is the canonical pre-restart sanity check — when a custom rule (added via Score Overrides or message rules) has invalid syntax, the lint catches it before the container restart finishes and prevents Amavis from starting against a broken config.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_antispam_maintenance.cfm`	`hermes_commandbox`	The page
`config/hermes/var/www/html/admin/2/inc/spam_settings_save.cfm`	`hermes_commandbox`	Validation + UPDATE + apply chain
`config/hermes/var/www/html/admin/2/inc/get_spam_settings.cfm`	`hermes_commandbox`	Loads current `spam_settings` rows
`config/hermes/var/www/html/admin/2/inc/update_amavis_config_files.cfm`	`hermes_commandbox`	Renders `50-user` from template + DB
`config/hermes/var/www/html/admin/2/inc/update_spamassassin_config_files.cfm`	`hermes_commandbox`	Renders `local.cf` from template + DB
`config/hermes/var/www/html/admin/2/inc/restart_amavis.cfm` / `restart_spamassassin.cfm` / `restart_mail_filter.cfm`	`hermes_commandbox`	`docker container restart hermes_mail_filter`
`config/hermes/var/www/html/admin/2/inc/antispam_init_pyzor.cfm` / `antispam_init_razor.cfm` / `antispam_clear_bayes.cfm`	`hermes_commandbox`	Maintenance docker-exec helpers
`config/hermes/opt/hermes/conf_files/50-user.HERMES`	template (read) -> `hermes_mail_filter` (live `/etc/amavis/conf.d/50-user`)	Amavis directives template
`config/hermes/opt/hermes/conf_files/local.cf.HERMES`	template (read) -> `hermes_mail_filter` (live `/etc/spamassassin/local.cf`)	SpamAssassin directives template
`/etc/amavis/conf.d/50-user.HERMES.BACKUP`	`hermes_mail_filter`	Pre-write backup, refreshed each save

Path	Owner	Role
`/etc/spamassassin/local.cf.HERMES.BACKUP`	`hermes_mail_filter`	Pre-write backup, refreshed each save
`spam_settings` table	`hermes_db_server` (`hermes` DB)	Source of truth for every UI value on this page; also holds per-rule scores (`spamfilter=1` rows) for Score Overrides
`message_rules` table	`hermes_db_server`	Custom header/body/full message rules; rendered into `local.cf`
`file_rule_components` / `files` tables	`hermes_db_server`	Banned-file rules; rendered into `50-user`
`dkim_sign` table	`hermes_db_server`	Per-domain DKIM keys; rendered into `50-user` for outbound signing

Related

- Antivirus Settings -- the ClamAV engine that runs in the same Amavis pass and whose virus verdict always pre-empts the spam score
- Malware Feeds -- third-party ClamAV signature feeds; orthogonal to spam scoring but consumed in the same scan
- Score Overrides -- per-rule SpamAssassin weight adjustments (the `spamfilter=1` rows in `spam_settings`); this page sets the engine knobs, that page sets the rule weights
- Message Rules -- custom header / body / full message regex rules that ride into `local.cf` on every save here
- SVF Policies -- per-sender and per-recipient spam-handling overrides that apply before the engine-wide `final_*_destiny` settings on this page
- Perimeter Checks -- the SMTP-time gate; every check on this page runs only after a connection clears the perimeter
- ARC Settings -- seals over the body Amavis passed, so a high spam score (and any quarantine action) naturally pre-empts the seal
- DMARC Settings -- a DMARC-fail verdict can promote a message to a higher spam score via SpamAssassin's DMARC rule weights (tunable on Score Overrides)
- Scheduled Tasks -- `sa-update` for the SpamAssassin rule set runs on its own Ofelia schedule; the Bayes DB is per-installation and not updated by `sa-update`
- Email flow -- full pipeline diagram

Revision #14
Created 2026-05-31 12:52:19 UTC by Dino Edwards
Updated 2026-06-13 12:30:16 UTC by Dino Edwards