

Antivirus Settings

Antivirus Settings

Admin path: **Content Checks > Antivirus Settings** (`view_antivirus_settings.cfm`, `inc/get_antivirus_settings.cfm`, `inc/antivirus_set_settings.cfm`, `inc/antivirus_add_whitelists.cfm`, `inc/antivirus_delete_entry.cfm`, `inc/generate_antivirus_configuration.cfm`, `inc/restart_clamav.cfm`).

This page configures the ClamAV antivirus engine that runs inside `hermes_mail_filter` and is called by Amavis on every message that clears the SMTP-time perimeter. Two cards: the main settings card (sixteen toggles that map to `clamd.conf` directives) and a Pro-only AV Signature Whitelist for suppressing known-bad-signature false positives. Refreshing third-party signature *feeds* (Sanesecurity, SecuritInfo, MalwarePatrol, etc.) is configured separately on [Malware Feeds](#); this page configures the engine itself.

Where antivirus sits in the flow

```
+-----+
inbound msg -->| Perimeter Checks pass      |
+-----+-----+
                |
                v
+-----+-----+
| Postfix smtpd_proxy_filter                |
|   -> hermes_mail_filter:10024            |
+-----+-----+
                |
                v
+-----+-----+
| Amavis (hermes_mail_filter)              |
|   - SpamAssassin scoring                 |
|   - ClamAV antivirus <---- this page configures this engine
```


ClamAV Antivirus Settings card

Sixteen toggles, each rendered from the `avSettings` array in `view_antivirus_settings.cfm` with an inline hint and a "Recommended" label on the safer default. Every toggle writes `parameters2.value2 = 'true' | 'false'` for `module = 'clamav'`; on save, `generate_antivirus_configuration.cfm` selects every active row and emits one `<directive> <value>` line per toggle into a temp file, substitutes the temp file into the `HERMES_ANTIVIRUS_SETTINGS_GO_HERE` placeholder of `clamd.conf.HERMES`, backs up the live config to `clamd.conf.HERMES`, and moves the rendered file into place.

UI Toggle	<code>clamd.conf</code> directive	Recommended	Notes
Scan Email Attachments	<code>ScanMail</code>	Enabled	Master switch for inbound attachment scanning
Scan Archives	<code>ScanArchive</code>	Enabled	Recurse into ZIP, RAR, 7z, etc. Without this, only the archive wrapper is scanned
Mark Encrypted Archives as Viruses	<code>ArchiveBlockEncrypted</code>	Disabled	Aggressive; commonly false-positives on legitimate password-protected files
Scan Portable Executables	<code>ScanPE</code>	Enabled	Windows PE format; required for decompression of UPX / FSG / Petite packers
Scan OLE2 Files	<code>ScanOLE2</code>	Enabled	MS Office <code>.doc/.xls/.ppt</code> and <code>.msi</code>
Block OLE2 VBA Macros	<code>OLE2BlockMacros</code>	Disabled	Blocks ALL macro-enabled documents regardless of intent (detected as <code>Heuristics.OLE2.ContainsMacros</code>); useful in strict environments, breaks legitimate macros otherwise
Scan PDF Files	<code>ScanPDF</code>	Enabled	PDF embedded JS, exploit detection
Scan HTML/JavaScript Content	<code>ScanHTML</code>	Enabled	HTML normalization + JavaScript/ScriptEncoder decryption; phishing + script-exploit detection
Algorithmic Detection	<code>AlgorithmicDetection</code>	Enabled	Engine-level heuristics for complex malware and graphic-file exploits
Scan ELF Files	<code>ScanELF</code>	Enabled	Linux/Unix executable format

UI Toggle	clamd.conf directive	Recommended	Notes
Phishing Signature Detection	PhishingSignatures	Enabled	ClamAV's phishing signature DB
Scan Email URLs for Phishing	PhishingScanURLs	Enabled	URL extraction + phishing URL DB lookup
Block SSL Mismatches in URLs	PhishingAlwaysBlockSSLMismatch	Disabled	False-positives on CDN and redirect URLs
Block Cloaked URLs	PhishingAlwaysBlockCloak	Disabled	False-positives on URL shorteners and marketing-tracker links
Detect Potentially Unwanted Applications	DetectPUA	Enabled	Adware, dialers, non-malicious-but-unwanted software
Heuristic Scan Precedence	HeuristicScanPrecedence	Enabled	When on, heuristic hits stop the scan immediately (saves CPU). When off, scanning continues so a signature-based hit can override a heuristic match

“ **Operational consequence — disabling ScanMail**. This effectively turns off antivirus for inbound mail. Amavis will still consult ClamAV for ban-pattern decisions but the engine will skip the attachment scan. Leave on except for very short-term diagnostics.

Operational consequence — 0LE2BlockMacros = true. Every macro-enabled Office document is blocked as Heuristics.0LE2.ContainsMacros, including documents from your own users. Most organizations get better results with macro-blocking enforced at the endpoint (Microsoft 365 Protected View, Group Policy) rather than at the gateway. Turn on only after warning users and ensuring you have a release workflow.

AV Signature Whitelist card (Pro)

When ClamAV produces a false positive on a known-safe file, the admin enters the exact ClamAV signature name (e.g. Heuristics.0LE2.ContainsMacros) and Hermes appends it to /var/lib/clamav/local.ign2. ClamAV reads local.ign2 at engine start and suppresses any detection whose signature name matches a line in the file.

Storage: parameters2 WHERE module = 'clamav-bypass' (one row per signature name, parameter column holds the signature string). On every save and on every delete,

`generate_antivirus_configuration.cfm` rewrites the whole `local.ign2` from the table, runs `dos2unix` to scrub line endings, backs up the current file to `local.ign2.HERMES`, and moves the new file into place. ClamAV is then restarted via `restart_clamav.cfm` to pick up the change.

How to find a signature name

The in-card info box gives admins the lookup steps:

1. From **Message History**, find the blocked message (Type column shows `Virus` or `Banned`)
2. Grep the mail-filter log for the message ID: `docker logs hermes_mail_filter 2>&1 | grep <mail_id>`
3. The log line shows the signature in parentheses, e.g. `Blocked INFECTED (Heuristics.OLE2.ContainsMacros)`
4. Or scan a file directly: `docker exec hermes_mail_filter clamscan /path/to/file`

“ **Operational consequence — whitelisting is by signature name, not by file hash.** If you whitelist `Heuristics.OLE2.ContainsMacros`, you have effectively turned off macro detection globally. Prefer narrow signature names (specific malware family) over heuristic families when possible.

Signature refresh

Two independent refresh loops keep the engine current:

Source	Mechanism	Cadence	Database
Official ClamAV (<code>main.cvd</code> , <code>daily.cvd</code> , <code>bytecode.cvd</code>)	<code>freshclam</code> daemon inside <code>hermes_mail_filter</code>	Default 1h (configurable in <code>/etc/clamav/freshclam.conf</code>)	<code>/var/lib/clamav/</code>
Third-party feeds (Sanesecurity, SecuritInfo, MalwarePatrol, etc.)	<code>fangfrisch refresh</code> via Ofelia job <code>hermes-fangfrisch-refresh</code>	Every 10 minutes (only feeds whose own publish cycle has elapsed actually re-download)	<code>/var/lib/fangfrisch/signatures/</code> then linked into <code>/var/lib/clamav/</code> by <code>setup-clamav-sigs</code>

`fangfrisch` is the small Python tool that handles auth, cadence control, and integrity verification for third-party feeds; the feed list and per-feed enable/disable lives on [Malware Feeds](#). Enabling premium feeds (SecuritInfo paid, MalwarePatrol paid) requires Pro licensing — the feed list itself is gated on the same page.

Resource footprint

Loading the full signature database into RAM costs roughly 1.5–2 GB of memory. If

`hermes_mail_filter` is under-provisioned (e.g. shared host with 4 GB total), `clamd` will fail to start, mail will queue behind Amavis, and the only sign in the UI is a quiet rise in deferred queue depth. Plan for at least 4 GB dedicated to the `hermes_mail_filter` container on systems with all third-party feeds enabled.

The default ClamAV file-size cap is 25 MB (`MaxFileSize 25M` in `clamd.conf`). Messages larger than this are passed without scan and flagged with a `Heuristics.Limits.Exceeded` indicator. Raising the cap requires editing `clamd.conf.HERMES` directly; the UI does not expose it because raising it disproportionately increases RAM and CPU per scan.

Save flow

1. View page submits `action="AV Settings"` (sixteen booleans),
`action="Add AV Whitelist"` (textarea),
`action="Delete Entry"` (id list)
2. `view_antivirus_settings.cfm` validates every `avFields` entry exists and is `true|false` (any failure -> `error.cfm` + `cfabort`)
3. `antivirus_set_settings.cfm` UPDATES `parameters2.value2` for each toggle (16 UPDATES, `module='clamav'`)
4. `generate_antivirus_configuration.cfm`:
 - a. `SELECT active='1'` rows from `parameters2` `module='clamav'` -> temp `avsettings` file
 - b. `dos2unix` the temp file
 - c. Substitute into `clamd.conf.HERMES` placeholder `HERMES_ANTIVIRUS_SETTINGS_GO_HERE`
 - d. Back up `/etc/clamav/clamd.conf` -> `clamd.conf.HERMES`, move new file into place
 - e. Rebuild `/var/lib/clamav/local.ign2` from `parameters2` `module='clamav-bypass'`
 - f. `dos2unix`, back up `local.ign2` -> `local.ign2.HERMES`, move new file into place
 - g. `cfinclude restart_clamav.cfm` (docker container restart `hermes_mail_filter` ClamAV process)
5. `session.m = 9` -> green "Antivirus Settings were saved successfully" alert

`generate_antivirus_configuration.cfm` also runs on whitelist add/delete — every change to either card triggers the same full regen + ClamAV restart cycle. The page does not return until the restart has completed (timeout per `cfexecute`).

Failure semantics

Failure	Behavior
Toggle form missing a required boolean field	<code>m = "Antivirus Settings: form.<f> does not exist", error.cfm, cfabort</code>
Toggle value not in <code>true, false</code>	<code>m = "Antivirus Settings: form.<f> is not true or false", error.cfm, cfabort</code>
Delete clicked with no selection	<code>session.m = 11</code>
Add Whitelist with empty textarea	<code>session.m = 13</code>
<code>dos2unix</code> failure on the temp avsettings or local.ign2 file	<code>error.cfm</code> + cfabort with the failing path in the message
<code>cp /etc/clamav/clamd.conf -> .HERMES</code> failure	<code>error.cfm</code> + cfabort
<code>mv <tmp>_clamd.conf -> /etc/clamav/clamd.conf</code> failure	<code>error.cfm</code> + cfabort
<code>restart_clamav.cfm</code> failure	Surfaces as cfcatch from the docker restart step

The save is **not** transactional across the steps — if the SQL updates succeed but the ClamAV restart fails, the DB state has already advanced. The next save will re-render and re-apply because every save regenerates the entire file from the current row state (no incremental writes).

Files and containers touched

Path	Owner	Role
<code>config/hermes/var/www/html/admin/2/view_antivirus_settings.cfm</code>	<code>hermes_commandbox</code>	The page
<code>config/hermes/var/www/html/admin/2/include/antivirus_*.cfm</code>	<code>hermes_commandbox</code>	Validate / save / regenerate / restart
<code>config/hermes/var/www/html/admin/2/include/get_antivirus_settings.cfm</code>	<code>hermes_commandbox</code>	Loads current <code>parameters2</code> <code>module='clamav'</code> values
<code>config/hermes/opt/hermes/conf_files/clamd.conf.HERMES</code>	<code>hermes_commandbox</code> (read -> <code>hermes_mail_filter</code> (live <code>/etc/clamav/clamd.conf</code>))	Canonical template with <code>HERMES_ANTIVIRUS_SETTINGS_GO_HERE</code> placeholder
<code>config/mail_filter/etc/clamav/clamd.conf</code>	<code>hermes_mail_filter</code> (live config, bind-mounted)	Read by <code>clamd</code> at start
<code>/var/lib/clamav/local.ign2</code>	<code>hermes_mail_filter</code> (Docker named volume <code>mail_filter_data_clamav</code>)	Signature whitelist; rewritten on every save
<code>/var/lib/clamav/*.cvd, *.cld, *.ndb, etc.</code>	<code>hermes_mail_filter</code>	Signature databases (official + third-party)
<code>parameters2</code> table, <code>module='clamav'</code>	<code>hermes_db_server</code> (<code>hermes</code> DB)	Source of truth for the sixteen toggles

Path	Owner	Role
<code>parameters2</code> table, <code>module='clamav-bypass'</code>	<code>hermes_db_server</code> (<code>hermes</code> DB)	Source of truth for the AV Signature Whitelist
<code>malware_databases</code> table	<code>hermes_db_server</code> (<code>hermes</code> DB)	Third-party feed list (configured on Malware Feeds)
<code>ofelia_jobs</code> row <code>hermes-fangfrisch-refresh</code>	<code>hermes_db_server</code>	10-minute feed refresh scheduler
<code>hermes_mail_filter</code> container	—	<code>clamd</code> , <code>freshclam</code> , <code>fangfrisch</code> , Amavis, SpamAssassin

Related

- [Malware Feeds](#) — the third-party signature feed configuration (Sanesecurity, SecuritInfo, MalwarePatrol, etc.) that Fangfrisch refreshes every 10 minutes
- [Perimeter Checks](#) — every check on this page runs only after a connection clears the SMTP-time perimeter
- [Anti-Spam Settings](#) — runs in the same Amavis pass; a virus verdict overrides any spam score
- [Score Overrides](#) — per-rule weight changes for SpamAssassin
- [Email Policies > Disclaimers](#) — body modification that runs after Amavis re-injection; never conflicts with ClamAV because it happens post-scan
- [ARC Settings](#) — seals over the body Amavis passed, so a virus verdict naturally pre-empts everything downstream
- [DNS Resolver](#) — URL phishing lookups (`PhishingScanURLs`) and signature-feed downloads (Fangfrisch) all resolve through `hermes_unbound`
- [Email flow](#) — full pipeline diagram

Revision #8

Created 2026-05-31 12:52:20 UTC by Dino Edwards

Updated 2026-05-31 14:01:17 UTC by Dino Edwards