# Malware Feeds

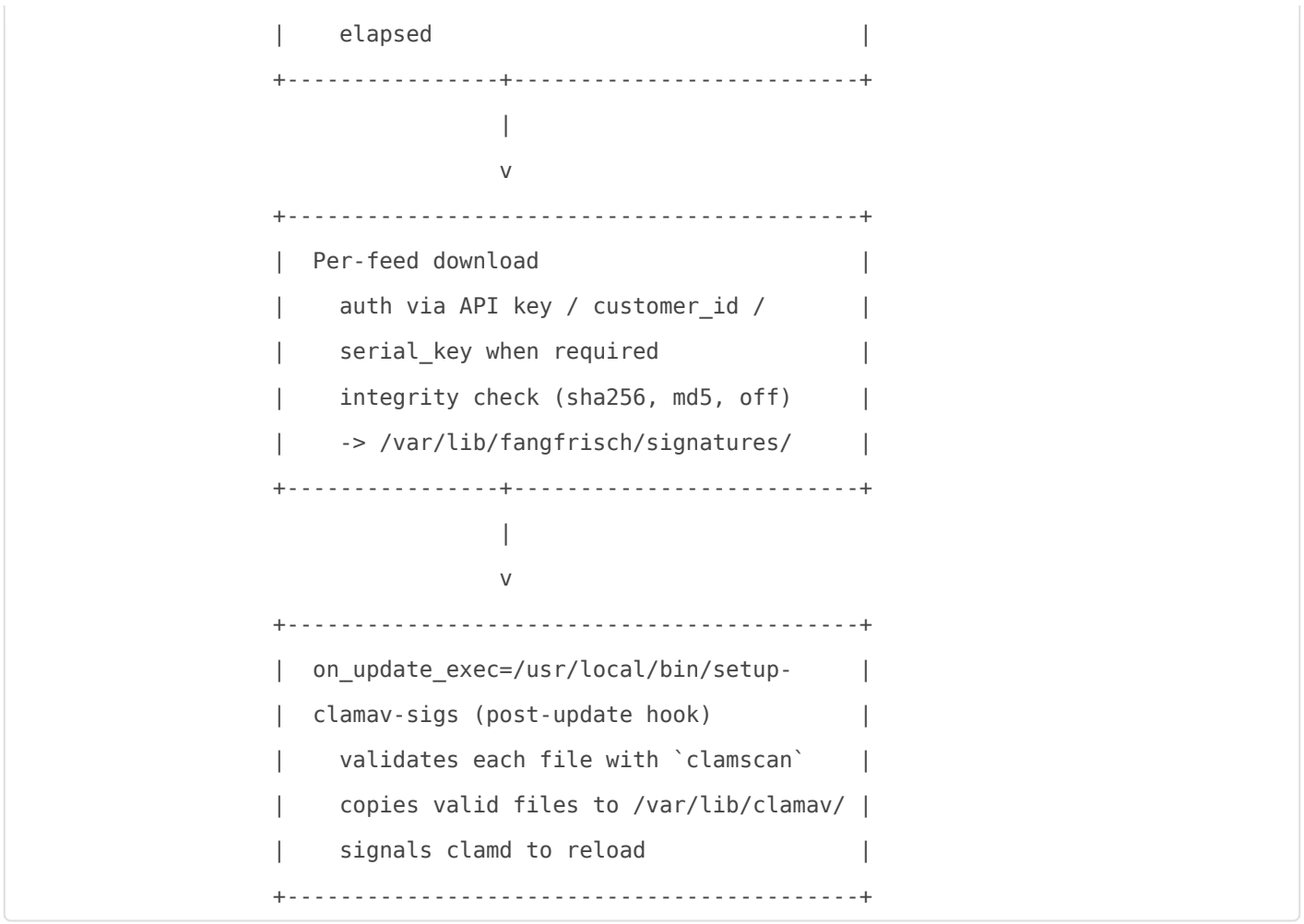
# Malware Feeds

Admin path: **Content Checks > Malware Feeds** (`view_malware_feeds.cfm`, `inc/get_malware_feeds_settings.cfm`, `inc/malware_feeds_save_global.cfm`, `inc/malware_feeds_add_feed.cfm`, `inc/malware_feeds_edit_feed.cfm`, `inc/malware_feeds_delete_feed.cfm`, `inc/malware_feeds_toggle_feed.cfm`, `inc/malware_feeds_save_urls.cfm`, `inc/generate_malware_feeds_configuration.cfm`).

This page manages the third-party ClamAV signature feeds that supplement the stock `freshclam` definitions on Antivirus Settings. The feed manager is Fangfrisch, a small Python tool that handles per-feed authentication, cadence control, integrity verification, and post-download deployment. Hermes ships ten built-in feed definitions (free and commercial), exposes a custom-feed form for additional sources, and a per-feed URL editor for signature file selection. Refresh runs as an Ofelia job inside `hermes_mail_filter`.

This page replaced an earlier `view_antivirus_signature_feeds.cfm` page (orphan cleanup tracked as issue #257); any sidebar bookmark or external link pointing at the old page should be updated.

# How feeds reach ClamAV




The page emits `/etc/fangfrisch/fangfrisch.conf` (an INI file) on every save. Fangfrisch itself is invoked on a fixed Ofelia schedule; the schedule is regenerated from `ofelia_jobs.schedule` and reflects the Global Settings > Refresh Interval picker.

# Container and tool placement

| Component | Detail |
|---|---|
| Container | `hermes_mail_filter` (IPv4 `.105`, same container as ClamAV, Amavis, SpamAssassin) |
| Feed manager | `fangfrisch` (Python, third-party ClamAV signature aggregator) |
| INI config | `/etc/fangfrisch/fangfrisch.conf` (bind-mounted, owned `root:clamav`, mode `640`) |
| State DB | `sqlite:////var/lib/fangfrisch/db.sqlite` (per-feed last-refresh, integrity hashes) |
| Download dir | `/var/lib/fangfrisch/signatures/` (raw downloaded files) |
| Deploy dir | `/var/lib/clamav/` (validated files, ClamAV signature store) |

| Component | Detail |
|---|---|
| Post-update hook | `/usr/local/bin/setup-clamav-sigs` (validates with `clamscan`, copies to deploy dir, signals reload) |
| Scheduler | Ofelia job `hermes-fangfrisch-refresh` row in `ofelia_jobs` |
| Default cadence | `@every 10m` (Fangfrisch then honors per-feed `interval =` to decide what to actually re-fetch) |

# Global Settings card

Four controls write to `parameters2 WHERE module = 'malware_feeds'`. The first three substitute into the `[DEFAULT]` section of `fangfrisch.conf` on every save; the fourth updates the Ofelia row that schedules the refresh job.

| Field | Storage | INI / scheduler effect | Notes |
|---|---|---|---|
| Log Level | `parameters2.value2` (`log_level`) | `[DEFAULT] log_level = ...` | `debug,info,warning,error,fatal`; logs go to `docker logs hermes_mail_filter` |
| Default Max Size | `parameters2.value2` (`max_size`) | `[DEFAULT] max_size = ...` | Per-file cap. Regex anchors a number followed by `KB`, `MB`, `M`, or `B` (e.g. `5MB`, `10M`, `250KB`). Inherited by feeds that don't set their own |
| Update Timeout (sec) | `parameters2.value2` (`on_update_timeout`) | `[DEFAULT] on_update_timeout = ...` | Bounded 1-300. Caps how long `setup-clamav-sigs` is allowed to run |
| Refresh Interval | `parameters2.value2` (`refresh_interval`) AND `ofelia_jobs.schedule` | Ofelia `@every <interval>` | Allowed values: `5m,10m,15m,30m,1h,2h,4h`. Fangfrisch's own per-feed `interval =` still gates whether each feed actually re-downloads on a given run |

The post-update hook path is hard-coded to `/usr/local/bin/setup-clamav-sigs` and shown read-only beneath the form as `[DEFAULT] on_update_exec`. The hook lives inside the `hermes_mail_filter` image and validates each downloaded file with `clamscan` before copying it to `/var/lib/clamav/`; a file that fails validation is left in the Fangfrisch download dir and not deployed.

# Malware Feeds card

Rows from `malware_feeds_config` populate a DataTable; per-row form posts toggle, edit, manage URLs, and (custom feeds only) delete. The schema:

| Column | Role |
|---|---|
| `id` | Surrogate key |
| `section_name` | INI section header, `[<section_name>]`. Lowercase alphanumeric + underscore (`^[a-z0-9_]+$`). Cannot change after creation. Unique. |
| `display_name` | Card label, free text |
| `enabled` | `tinyint(3)`, 0/1. Sliders here flip this. `enabled = yes/no` line in INI |
| `is_builtin` | `tinyint(3)`, 0/1. Built-in rows cannot be deleted (the Delete action button is suppressed in the UI and the delete handler refuses) |
| `prefix` | `${prefix}` interpolation source for URL entries. Optional |
| `interval_value` | Per-feed cadence (e.g. `1h`, `4h`, `1d`); blank = inherit `@every <refresh_interval>` |
| `max_size` | Per-feed cap; blank = inherit Global Default Max Size |
| `integrity_check` | `sha256`, `md5`, `disabled`, or NULL (default `sha256`) |
| `api_key_1_name` / `api_key_1_value` | Optional auth key (e.g. `customer_id`, `receipt`). Value stored AES-encrypted with key `/opt/hermes/keys/hermes.key` |
| `api_key_2_name` / `api_key_2_value` | Second auth key (e.g. MalwarePatrol's `product`). Same encryption |
| `description` | Free text |
| `sort_order` | Display order; custom-add inserts at 100 |

# Built-in feed catalog (factory rows)

| Feed | Type | Default state | Auth | Notes |
|---|---|---|---|---|
| SaneSecurity | Free | Enabled | None | Broad zero-day coverage; mirror `https://ftp.swin.edu.au/sanesecurity/` |
| URLhaus | Free | Enabled | None | Malicious URL signatures from abuse.ch |
| MalwarePatrol | Commercial | Enabled | `receipt`, `product` IDs | Configure both keys via Edit; subscription IDs are documented in the in-card help |

| Feed | Type | Default state | Auth | Notes |
|---|---|---|---|---|
| MalwareExpert | Commercial | Enabled | `serial_key` | URL template embeds the serial in the path |
| SecuriteInfo | Commercial | Enabled | `customer_id` | Free tier available; paid tier unlocks extra URLs |
| TwinWave | Free | Enabled | None | Public GitHub-hosted signatures |
| ClamPunch | Free | Enabled | None | Heuristic family signatures |
| RFXN | Free | Enabled | None | R-fx Networks Linux Malware Detect signatures |
| InterServer | Free | Enabled | None | Hash + URL signatures |
| Ditekshen | Free | Enabled | None | YARA/ClamAV detection rules |

A commercial feed is "enabled" only in the sense that its row is marked `enabled = 1`; without API keys the feed is configured but will not actually fetch (the in-card help describes the per-vendor key requirements and the table icon shows a yellow warning triangle on commercial rows missing keys).

# Add Custom Feed modal

Free-form add for any feed source not in the built-in catalog. Validation:

| Field | Rule |
|---|---|
| Section Name | `^[a-z0-9_]+$`, required, must not already exist |
| Display Name | Required |
| URL Prefix | Optional, becomes the `prefix =` line and the substitution source for `${prefix}` in URL entries |
| Update Interval | Optional, number followed by `m` (minutes), `h` (hours), or `d` (days). Examples: `10m`, `1h`, `1d` |
| Max Size | Optional, number followed by `KB`, `MB`, `M`, or `B`. Examples: `5MB`, `250KB` |
| Integrity Check | Dropdown: default (sha256), sha256, md5, disabled |
| Description | Optional free text |

A new custom feed is inserted with `enabled = 0` and `is_builtin = 0`; the admin then opens the URL manager to register at least one URL before turning the row on.

# Manage URLs modal (per-feed)

Rows from `malware_feed_urls` keyed by `feed_id`. Each URL becomes a line in the corresponding `[<section_name>]` block of `fangfrisch.conf`:

```
url_<url_key> = <url_value>
filename_<url_key> = <filename_override>   ## only when filename_override set
```

When a URL is toggled off, the `url_` prefix is replaced with `!url_` to inactivate the line without losing the configuration. `${prefix}` in the URL value is expanded against the feed's `prefix =` at fetch time.

| Field | Rule |
|---|---|
| Name (`url_key`) | `^[a-z0-9_.]+$`, must be unique within the feed (`UNIQUE KEY uq_feed_url(feed_id, url_key)`) |
| Download URL (`url_value`) | Full URL, or `${prefix}<path>` shorthand when the feed has a prefix |
| Save As (`filename_override`) | Optional. Renames the downloaded file locally; useful when the source filename is too generic |
| Toggle | Per-URL on/off. Disabled URLs are skipped without being deleted |

Built-in feeds may have URLs that Fangfrisch maintains internally — the in-modal note explains that an empty URL table for a built-in feed means it is using its packaged defaults, not that it is broken.

# Save flow

```
1. View page submits action= save_global | add_feed | edit_feed
                          | delete_feed | toggle_feed | url_action
2. malware_feeds_*.cfm validates and UPDATEs/INSERTs/DELETEs the row(s)
3. generate_malware_feeds_configuration.cfm runs on EVERY action:
     a. SELECT module='malware_feeds' rows from parameters2 -> globalSettings
     b. SELECT malware_feeds_config -> all feed rows
     c. SELECT malware_feed_urls -> all URLs grouped by feed_id
```

```
      d. Build [DEFAULT] section + one [<section_name>] block per feed
      e. Decrypt api_key_*_value with AES + /opt/hermes/keys/hermes.key
         (key emitted as `<api_key_*_name> = <plain>`)
      f. Write temp file -> /opt/hermes/tmp/<trans>_fangfrisch.conf
      g. dos2unix (tolerated if missing)
      h. cffile write -> /etc/fangfrisch/fangfrisch.conf
      i. docker exec hermes_mail_filter chown root:clamav + chmod 640
         (tolerated if container is down)
      j. cfinclude ofelia_generate_config.cfm
         (rewrites /etc/ofelia/config.ini if any schedule changed)
  4. cflocation back to view_malware_feeds.cfm
  5. session.m + session.alerttype + session.alertmsg drives the alert banner
```

Every UI action -- including a single-row enable/disable toggle -- runs the full INI regen, ownership fix, and Ofelia config regen. There is no incremental write path; the INI is always rendered from the current database state. This means manual edits to `/etc/fangfrisch/fangfrisch.conf` are lost on the next save -- store all configuration in the database.

# API key encryption

The `api_key_1_value` and `api_key_2_value` columns store AES-Base64 ciphertext using the key in `/opt/hermes/keys/hermes.key`. The edit modal shows a masked preview (20 asterisks + last 4 chars of the plaintext) for visual confirmation without exposing the full key. Decryption happens only in `generate_malware_feeds_configuration.cfm` at the moment the INI is rendered; a decryption failure replaces the key line with a commented `## <name> = [decryption error]` marker rather than aborting the save.

The encryption key file is mounted into `hermes_commandbox` only; neither `hermes_mail_filter` nor any other service reads it. This keeps the plaintext key out of the running config on disk for as short a window as possible (write -> chmod 640 root:clamav -> next Fangfrisch run reads -> file remains until next save replaces it).

# Manual refresh

The Ofelia job runs on schedule, but the same command can be invoked manually from a host shell:

```
docker exec hermes_mail_filter fangfrisch --conf /etc/fangfrisch/fangfrisch.conf refresh
```

Fangfrisch is conservative — it will still skip feeds whose own per-feed `interval =` window has not elapsed. To force a re-download of a single feed regardless of cadence, the Fangfrisch state DB can be cleared for that feed:

```
docker exec hermes_mail_filter sqlite3 /var/lib/fangfrisch/db.sqlite \
  "DELETE FROM refreshlog WHERE source = '<section_name>';"
```

Then re-run the refresh. The post-update hook re-validates with `clamscan` and deploys to `/var/lib/clamav/`. To inspect downloaded files:

```
docker exec hermes_mail_filter ls -la /var/lib/fangfrisch/signatures/
```

# Failure semantics

| Failure | Behavior |
|---|---|
| Global save with non-allowlisted log_level / max_size / timeout / interval | session.m=`malware_feeds_error`, alerttype=danger, alertmsg explains; no DB write |
| Add Custom Feed with duplicate section name | session.m=error, alertmsg names the conflict; INSERT not attempted |
| Toggle/edit on non-existent `feed_id` | session.m=error "Feed not found"; no UPDATE |
| Delete attempted on a built-in feed | UI suppresses the button; handler refuses the row |
| API key decryption error at INI regen | INI line replaced with `## <name> = [decryption error]`; save still completes; Fangfrisch will treat the auth as missing on the next run |
| Container down during `chown`/`chmod` | `cftry` swallows the exec failure; INI is still written to the bind mount and the chown is applied next save when the container is back up |
| dos2unix binary missing | Tolerated via `cftry`; INI is written without the line-ending normalization step |

# Files and containers touched

| Path | Owner | Role |
|---|---|---|
| `config/hermes/var/www/html/admin/2/view_malware_feeds.cfm` | `hermes_commandbox` | The page |
| `config/hermes/var/www/html/admin/2/inc/malware_feeds_*.cfm` | `hermes_commandbox` | Validate / save / regen per action |

| Path | Owner | Role |
|---|---|---|
| `config/hermes/var/www/html/admin/2/inc/generate_malware_feeds_configuration.cfm` | `hermes_commandbox` | Renders the INI from the DB; runs on every action |
| `/etc/fangfrisch/fangfrisch.conf` | `hermes_mail_filter` (bind-mounted, root:clamav, 640) | Live Fangfrisch config |
| `/var/lib/fangfrisch/db.sqlite` | `hermes_mail_filter` | Per-feed last-refresh state |
| `/var/lib/fangfrisch/signatures/` | `hermes_mail_filter` | Raw downloads (pre-validation) |
| `/var/lib/clamav/` | `hermes_mail_filter` (Docker named volume `mail_filter_data_clamav`) | Validated signature store; ClamAV reads from here |
| `/usr/local/bin/setup-clamav-sigs` | `hermes_mail_filter` (image-baked) | Post-update validation + deploy hook |
| `/opt/hermes/keys/hermes.key` | `hermes_commandbox` only | AES key for `api_key_*_value` columns |
| `malware_feeds_config` table | `hermes_db_server` (`hermes` DB) | Per-feed row state |
| `malware_feed_urls` table | `hermes_db_server` | Per-feed URL list (FK cascade delete on feed delete) |
| `parameters2` rows `module='malware_feeds'` | `hermes_db_server` | Global Settings card |
| `ofelia_jobs` row `hermes-fangfrisch-refresh` | `hermes_db_server` | Schedule (auto-updated when Refresh Interval changes) |

# Related

- Antivirus Settings -- the ClamAV engine that consumes the signatures Fangfrisch downloads; engine toggles (ScanMail, ScanArchive, etc.) and the per-engine signature whitelist live on that page
- Scheduled Tasks -- the Ofelia admin page; the `hermes-fangfrisch-refresh` job row is editable there (manual Run Now, enable/disable)
- Score Overrides -- per-rule SpamAssassin weight changes; not related to ClamAV but the closest neighbor for the "tune a built-in rule" pattern
- Antispam Settings -- SpamAssassin runs in the same Amavis pass; a ClamAV virus verdict from a Fangfrisch-supplied signature always pre-empts the spam score
- DNS Resolver -- every Fangfrisch HTTP download resolves through `hermes_unbound`; outbound HTTPS to the feed providers must be reachable
- Email flow -- full pipeline diagram showing where ClamAV (and therefore feed-derived signatures) fits

---

Revision #8
Created 2026-05-31 12:52:27 UTC by Dino Edwards
Updated 2026-05-31 14:01:21 UTC by Dino Edwards