Email Relay

- Domains
- Relay Host
- Relay Networks
- Relay Recipients
- Virtual Recipients

Domains

Domains

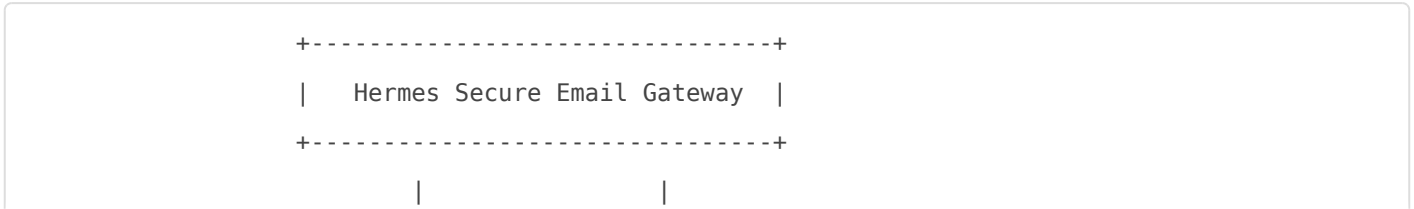
Admin path: **Email Relay > Domains** (`view_domains.cfm`, `inc/domain_add_action.cfm`, `inc/domain_edit_action.cfm`, `inc/domain_delete_action.cfm`, `inc/deletedomain.cfm`, `inc/get_domain_json.cfm`, `inc/generate_transports.cfm`, `inc/generate_relay_domains.cfm`, `inc/generate_sasl_password_transport.cfm`, `inc/generate_postfix_configuration.cfm`, `inc/add_domain_djigzo.cfm`, `inc/delete_domain_djigzo.cfm`).

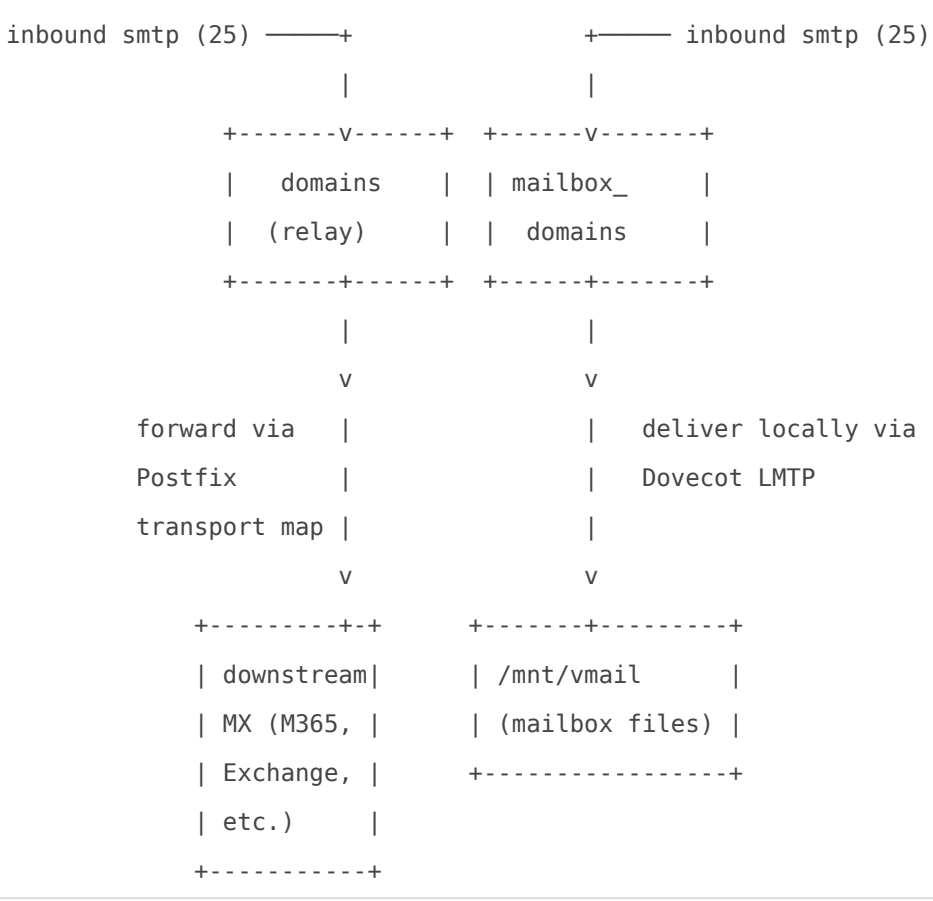
This page manages the list of **inbound relay domains** — the SMTP domains for which Hermes accepts mail and forwards it to a downstream mail server (Microsoft 365, Exchange, Google Workspace, on-prem Postfix/Dovecot, an internal hub MTA, etc.). Each row in the `domains` table is paired with a `transport` row that tells Postfix where to forward, a `senders` row that flags the domain as a recognized sender, and a `recipients` row that gates whether the domain accepts mail for any address or only addresses on the Relay Recipients allowlist.

This is the inbound counterpart to Relay Host. The two pages together define the **relay topology** half of Hermes: inbound domains here, outbound smarthost there.

> **Not to be confused with Email Server > Domains.** That page is for the **mail-server topology** — domains where Hermes IS the destination MTA and delivers locally to Dovecot mailboxes. It writes to the `mailbox_domains` table, not the `domains` table. The two tables and the two admin pages are separate by design because Hermes supports three topologies (see Hermes topology overview below) and a single deployment can run any combination.

Hermes topology overview





Topology	`domains` rows	`mailbox_domains` rows	This page edits
Relay-only	one or more	none	Yes
Mail-server-only	none	one or more	No — use Email Server > Domains
Hybrid	one or more (forwarded)	one or more (delivered locally)	Yes, for the relay subset

`view_domains.cfm` filters its main query with `WHERE (d.type IS NULL OR d.type = '' OR d.type = 'relay')` so it only shows relay-mode rows. Add Domain writes `type='relay'` explicitly so the row is unambiguously routed to this page.

How a relay domain becomes Postfix config

A single Add Domain submission writes four database rows and regenerates four Postfix maps:

```
form submit  ──► domain_add_action.cfm
                        |
```

```
                |  INSERT transport (domain, transport, dest, port, mx, auth, ...)
                |  INSERT senders   (sender = domain, action = OK)
                |  INSERT recipients(recipient = @domain, status = OK|"")
                |  INSERT domains   (domain, transport_id, senders_id,
                |                    recipients_id, type='relay')
                |
                |  --- regenerate ---
                v
        generate_transports.cfm        -> /etc/postfix/transport
                                          + postmap (docker exec)
        generate_relay_domains.cfm     -> /etc/postfix/relay_domains
        sync_sasl_parameters.cfm
        generate_sasl_password_transport.cfm
                                       -> /etc/postfix/sasl_passwd
                                          + postmap (docker exec)
        generate_tls_policy.cfm        -> /etc/postfix/tls_policy
                                          + postmap (docker exec)
        generate_postfix_configuration.cfm
                                       -> /etc/postfix/main.cf
                                          + postfix reload (docker exec)
        add_domain_djigzo.cfm          -> registers domain in Ciphermail
                                          (encryption gateway)
```

The same pipeline runs on edit and delete (with the appropriate deletes substituted for inserts). The page deliberately does not expose a "dry-run" — every change to a domain is a config-changing save, and the cascade always runs to completion.

Configuration storage

Table	Role	Notes
`domains`	One row per relay domain	`type` column gates which admin page edits the row (`relay`, NULL/empty = relay; anything else = managed elsewhere). `id`, `transport_id`, `senders_id`, `recipients_id` are the join keys.

Table	Role	Notes
`transport`	One row per domain delivery target	`transport` column holds the Postfix-formatted string (`smtp:[host]:port` or `smtp:host:port` for MX-lookup mode, or `discard:Discard Email Silently`). `authentication = YES` toggles per-domain SASL. `authentication_username` / `authentication_password` are AES/Base64 encrypted with `/opt/hermes/keys/hermes.key`.
`senders`	One row per domain (sender = domain, action = `OK`)	Used by Postfix `smtpd_sender_restrictions` to recognise the domain as a known sender.
`recipients`	One row per domain (recipient = `@domain`, `domain='1'`)	`status = OK` = accept mail for any address (recipient_delivery = ANY). `status = ''` = require an entry in Relay Recipients (recipient_delivery = SPECIFIED). The default `spam_policies` policy is attached so Amavis applies SVF filtering.
`tls_policies`	Optional, one row per domain	Auto-managed: created with `method=encrypt` when **Enforce TLS** is on and Auth is YES; removed when either is turned off. Manually-added policies (different `description`) are untouched.
`dkim_sign`	Optional, one or more rows per domain	DKIM keys live separately; managed under the per-row **DKIM Keys** button (`edit_domain_dkim.cfm`). DKIM badge in the table reports `Active` / `Disabled` / `None` based on `enabled = '1'` counts.

Fields on the page

Add Domain card

Field	Default	Notes
Domain Name	(empty)	Trimmed, lower-cased, validated by the email-trick. Uniqueness checked against `domains.domain` — duplicates rejected with error 12. Stored as-is on the row.

Field	Default	Notes
Delivery Method	`SMTP (Recommended)`	`smtp` forwards via the destination address; `discard` writes `discard:Discard Email Silently` into the transport row and accepts mail only to drop it. Useful for honeypot or sunset domains.
Recipient Delivery	`ANY`	`OK` = accept any recipient at the domain. `""` = SPECIFIED — only addresses listed under Relay Recipients are accepted; everything else is rejected at SMTP time with `relay_recipient_maps`.
Destination Address	`smtp.<domain>` (placeholder)	FQDN or IP of the downstream MX/smarthost. Lower-cased. Required when method = `smtp`.
Port	`25`	Free-text but validated as integer. No range cap on this page (vs. Relay Host's explicit 1–65535) but Postfix will reject out-of-range.
MX Lookup	`NO`	`NO` writes a bracketed transport `smtp:[host]:port` (Postfix skips MX, connects directly). `YES` writes unbracketed `smtp:host:port` (Postfix resolves MX records). MX mode is **automatically forced off** when Auth = YES, because authenticated submission with MX rotation rarely makes sense.
Auth	`NO`	When `YES`, the username/password and Enforce TLS fields reveal.
Destination Username / Password	(empty)	Required when Auth = YES. Encrypted with `/opt/hermes/keys/hermes.key` before write. On Edit, blank password keeps the existing ciphertext.
Enforce TLS	checked	When Auth = YES, auto-inserts a `tls_policies` row with `method=encrypt` and `description='Auto-added: domain requires authentication'`. Manages itself on subsequent edits — turning either off deletes the auto-added row but leaves manually-added TLS policies alone.

Domains table

Sortable, searchable, exportable (copy/CSV/Excel/PDF/print via the DataTables Buttons extension; `stateSave: true` so column ordering and page-size choices persist across reloads). Columns:

Column	Source	Badge logic
Domain	`domains.domain`	Plain text
Delivery	`transport.method`	`Discard` (warning) or `SMTP` (success)
Destination	`transport.destination`	Dash for discard rows
Port	`transport.port`	Dash for discard
MX	`transport.mx`	Dash for discard
Recipients	`recipients.status`	`Any` (info) when `OK`, `Specified` (secondary) otherwise
Auth	`transport.authentication`	`YES` (warning) or `NO` (secondary)
DKIM	aggregated from `dkim_sign`	`Active` when any enabled key, `Disabled` when keys exist but all disabled, `None` when no keys
TLS	derived from `tls_policies.domain` join	`YES` (success) when a policy exists for the domain, `NO` (secondary) otherwise
Actions	—	Edit (opens modal), DKIM Keys (→ `edit_domain_dkim.cfm`), Delete (opens confirm modal)

Edit Domain modal

Opens via `openEditModal(id)` which fetches `./inc/get_domain_json.cfm` over AJAX, hydrates the form fields, then reveals the modal body. **Domain Name is read-only on edit** — changing a domain name across `domains` / `transport` / `senders` / `recipients` / `dkim_sign` / `tls_policies` is risky enough that the page enforces add-and-delete instead. Every other field is editable.

Blank password keeps the existing ciphertext (the masked hint beneath the input shows `Current: abcd*****` when a stored value exists).

Delete Domain modal

Confirms the destructive action. The handler (`deletedomain.cfm`) runs four dependency checks before allowing the delete:

Check	If it returns rows →
Relay Recipients still pointing at the domain (`recipients.recipient LIKE '%domain%' AND domain IS NULL`)	Error 1, abort

Check	If it returns rows →
Virtual Recipients referencing the domain (`virtual_recipients.virtual_address LIKE '%domain%'`)	Error 2, abort
Postmaster address using the domain (`system_settings.postmaster LIKE '%domain%'`)	Error 3, abort
DKIM keys for the domain (`dkim_sign.domain LIKE '%domain%'`)	Error 4, abort

If all four pass, the handler deletes from `domains`, `transport`, `senders`, and `recipients` (the four rows linked at creation), clears the `tls_policies` row for the domain, removes the Ciphermail registration, and regenerates all Postfix maps.

> **Operational consequence.** The dependency checks force a bottom-up cleanup. To remove a domain you must first delete its recipients, its DKIM keys, and reassign the system postmaster. This is intentional — Hermes will not silently strand referencing rows, and the order also prevents you from losing in-flight mail for active recipients.

Per-domain auth vs. relay host auth

Per-domain authentication on this page is **separate from and additive to** the global Relay Host SASL on the Relay Host page. Both pages write into the same `/etc/postfix/sasl_passwd` file via the shared `generate_sasl_password_transport.cfm` generator:

```
# /etc/postfix/sasl_passwd  (regenerated on every save on either page)
[smtp.upstream-isp.com]:587  globaluser:globalpass    <-- Relay Host page
[mx.partner-a.com]:25        partner_a_user:secret1   <-- Domains page (per-domain)
[mx.partner-b.com]:25        partner_b_user:secret2   <-- Domains page (per-domain)
```

A domain with per-domain auth will use **its own** credentials when Postfix forwards to its destination. The global relay host credentials are used only when a message has no matching per-domain transport (typical for outbound mail to arbitrary recipients).

> **By design.** The error code 15 (`Cannot enable Destination Authentication when Relay Host is enabled`) is reserved in the page's alert table but not currently raised by the action handlers — historically the two auth modes were considered mutually exclusive, but the consolidated SASL generator handles both cleanly, so the constraint was relaxed. The alert is kept in case a future tightening reintroduces the rule.

Discard delivery

Setting Delivery Method to `discard` writes `discard:Discard Email Silently` into the transport. Postfix accepts mail for the domain (passing SMTP-time checks and the content filter), then drops it on the floor — no NDR, no bounce, no forwarding attempt. Useful for:

- Sunset domains that should not generate backscatter
- Honeypot domains for spam-trap analysis
- Catching mail to a domain you control while migration is in progress and you don't want it bouncing

The destination/port/MX/auth/TLS fields are hidden in the UI when discard is selected because none of them apply.

Failure semantics

What breaks	What happens
Domain name empty	`session.m = 10`, redirect, no DB write
Domain name fails email-trick validation	`session.m = 11`, redirect, no DB write
Domain name already exists in `domains`	`session.m = 12`, redirect, no DB write
Delivery method not in `smtp,discard`	`session.m = 20`, redirect, no DB write
Destination address blank when method = smtp	`session.m = 13`, redirect, no DB write
Port not an integer	`session.m = 14`, redirect, no DB write
Auth = YES but username blank	`session.m = 16`, redirect, no DB write
Auth = YES but password blank AND no cached cipher	`session.m = 17`, redirect, no DB write
Delete blocked by dependency check	One of `session.m = 1..4` per the table above, redirect, no DB write

What breaks	What happens
`postmap` of `transport` / `sasl_passwd` / `tls_policy` fails	New map file is on disk but `.db` lags; next mail flow uses stale data until next successful postmap
`postfix reload` fails	Live config keeps the previous values; reload error is in container logs
`add_domain_djigzo.cfm` errors during Ciphermail registration	Domain row is already in the DB; encryption gateway will not know about the domain until the next manual sync. Re-saving the domain triggers a fresh registration attempt.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_domains.cfm`	`hermes_commandbox`	Page + Add/Edit/Delete modals
`config/hermes/var/www/html/admin/2/inc/domain_add_action.cfm`	`hermes_commandbox`	Add handler
`config/hermes/var/www/html/admin/2/inc/domain_edit_action.cfm`	`hermes_commandbox`	Edit handler
`config/hermes/var/www/html/admin/2/inc/domain_delete_action.cfm`	`hermes_commandbox`	Delete dispatch (thin wrapper)
`config/hermes/var/www/html/admin/2/inc/deletedomain.cfm`	`hermes_commandbox`	Delete handler with dependency checks
`config/hermes/var/www/html/admin/2/inc/get_domain_json.cfm`	`hermes_commandbox`	AJAX hydrator for the Edit modal
`config/hermes/var/www/html/admin/2/inc/generate_transports.cfm`	`hermes_commandbox`	Rewrites `/etc/postfix/transport` + postmap
`config/hermes/var/www/html/admin/2/inc/generate_relay_domains.cfm`	`hermes_commandbox`	Rewrites `/etc/postfix/relay_domains`
`config/hermes/var/www/html/admin/2/inc/generate_sasl_password_transport.cfm`	`hermes_commandbox`	Shared `sasl_passwd` generator (also used by Relay Host)
`config/hermes/var/www/html/admin/2/inc/generate_tls_policy.cfm`	`hermes_commandbox`	Rewrites `/etc/postfix/tls_policy` + postmap
`config/hermes/var/www/html/admin/2/inc/generate_postfix_configuration.cfm`	`hermes_commandbox`	Template-to-`main.cf` renderer + `postfix reload`
`config/hermes/var/www/html/admin/2/inc/add_domain_djigzo.cfm` / `delete_domain_djigzo.cfm`	`hermes_commandbox`	Ciphermail (djigzo) domain registration
`/etc/postfix/transport` + `.db`	`hermes_postfix_dkim`	Per-domain transport map (regen target)
`/etc/postfix/relay_domains`	`hermes_postfix_dkim`	List of domains Postfix accepts mail for (regen target)

Path	Owner	Role
`/etc/postfix/sasl_passwd` + `.db`	`hermes_postfix_dkim`	Consolidated SASL credentials (regen target)
`/etc/postfix/tls_policy` + `.db`	`hermes_postfix_dkim`	Per-destination TLS policy (regen target)
`/etc/postfix/main.cf`	`hermes_postfix_dkim`	Live Postfix config (re-rendered on every save)
`/opt/hermes/keys/hermes.key`	`hermes_commandbox`	Symmetric key for AES/Base64 cred encryption
`domains`, `transport`, `senders`, `recipients`, `tls_policies`, `dkim_sign`	`hermes_db_server`	The relay-domain row group

Every shell-out uses `docker exec hermes_postfix_dkim ...` per the standard Hermes pattern.

Related

- Relay Host — outbound smarthost; the page's twin. Shares the `sasl_passwd` generator and is part of the same relay topology.
- Relay Recipients — recipient allowlist used when a domain's Recipient Delivery is set to `SPECIFIED`. Required reading if you tighten recipient validation for a domain.
- Virtual Recipients — alias and catch-all mappings (`alias@dom → real@dom`). Independent of this page but domain deletes block when virtual rows reference the domain.
- Relay Networks — `mynetworks` (which clients may relay outbound without authentication). The networks that hold the per-domain submission clients live here.
- SMTP TLS Settings — manages per-destination TLS policies (the Enforce TLS checkbox on this page is a shortcut into the same table).
- Email Server > Domains — the separate page for mail-server-topology domains, backed by `mailbox_domains`. **Do not confuse with this page.**

Relay Host

Relay Host

Admin path: **Email Relay > Relay Host** (`view_relay_host.cfm`, `inc/get_relay_host_settings.cfm`, `inc/edit_relay_host_settings.cfm`, `inc/generate_sasl_password_transport.cfm`, `inc/generate_postfix_configuration.cfm`).

This page configures the **single global outbound relay host** that Postfix uses to deliver mail to the Internet — the smarthost an ISP, M365, SendGrid, or another upstream MTA supplies when direct delivery is blocked or undesirable. It controls the host/port pair, the optional SASL credentials, and the outbound TLS security level. Saving rewrites the relevant rows in the `parameters` table, regenerates `/etc/postfix/sasl_passwd`, and re-renders `/etc/postfix/main.cf` from the template so the new values take effect on the next message.

Pairs with Domains for the inbound half of the relay topology — Relay Host defines where outbound mail goes; Domains defines which inbound domains Hermes accepts and where each one is forwarded.

When you need a relay host

By default, Hermes attempts direct MX delivery for outbound mail. A relay host is required in any of these scenarios:

Scenario	Why direct delivery fails
Hermes is behind a firewall that blocks outbound TCP/25	Port 25 to the open Internet is filtered
ISP forbids outbound SMTP for residential/business links	Outbound TCP/25 is dropped at the ISP edge
Outbound IP has no PTR record or is on a blocklist	Recipients reject; deliverability tanks
Compliance requires all outbound mail to traverse a known SMTP gateway (M365 connector, SendGrid, on-prem hub)	Centralized policy/journaling/encryption point
Hermes sits on a non-routable internal network	No path to the Internet without a smarthost

If none of those apply and Hermes has a clean public IP with a PTR record, leave **Enable Relay Host** off and let Postfix do direct delivery.

How the relay host fits in the outbound path

```
local pickup / amavis re-inject (10025)
        |
        v
hermes_postfix_dkim (smtp client)
        |
        |  relayhost          = [smtp.example.com]:587   (from parameters)
        |  smtp_sasl_*        = enable + sasl_passwd map (from parameters + sasl_passwd)
        |  smtp_tls_security  = may | encrypt            (from parameters)
        |
        v
upstream smarthost  ──►  recipient MX
```

Only the upstream-bound TCP connection is affected. Inbound SMTP on port 25, the content-filter loop (Amavis on 10024/10026), and Dovecot LMTP delivery are untouched.

Configuration storage

Relay Host settings are spread across two tables. The host/port and SASL toggles live in the **`parameters`** table using the dual-row pattern (`child=2` parent name row, `child=1` value row). The SASL credentials themselves are encrypted at rest in **`system_settings`** to keep cleartext out of the directive table.

Setting	Storage	Notes
Enable Relay Host	`parameters.enabled` on `parameter='relayhost' AND child=2`	Master switch; disabling clears the child value and pushes `relayhost =` (empty) into `main.cf`
Relay Host Address	`parameters.name` on the `relayhost` child row	Bare FQDN/IP for display
Relay Host Port	Parsed from `parameters.parameter` (`[host]:port`)	Stored as the Postfix-formatted bracketed `[host]:port` literal
Outbound TLS Mode	`parameters.parameter` on `smtp_tls_security_level` child row (`""`, `may`, `encrypt`)	Empty value disables both parent and child; `may` = opportunistic STARTTLS; `encrypt` = mandatory TLS

Setting	Storage	Notes
Authentication required	`parameters.enabled` on `smtp_sasl_auth_enable` parent + `parameters.parameter` value `yes` / `no`	Flips the `smtp_sasl_password_maps` parent in lockstep
Relay Host Username	`system_settings.value` row `relay_host_username`	AES/Base64 encrypted with `/opt/hermes/keys/hermes.key`
Relay Host Password	`system_settings.value` row `relay_host_password`	AES/Base64 encrypted with the same key

> **By design.** The legacy schema kept the SASL username/password in plaintext on the `smtp_sasl_password_maps` child row's `name` column. The current code path encrypts both into `system_settings` and clears the legacy column on every save. The first read against a legacy install runs a one-shot migration in `get_relay_host_settings.cfm`: if `system_settings` is empty but the old `parameters.name` colon-delimited string is present, the values are encrypted forward and the plaintext column is cleared. No admin action is required.

Fields on the page

Enable Relay Host

Master switch. When off, all the other fields are hidden, the `relayhost` parent is set `enabled=0`, the child value is wiped, and the SASL parent/child rows + `system_settings` credentials are cleared in the same save. Postfix is then re-rendered with `relayhost =` empty so the next outbound message attempts direct delivery again.

Relay Host Address

Accepts:

- **IPv4** — validated against a dotted-quad regex with 0–255 octet bounds
- **IPv6** — validated against a simplified colon/hex check
- **FQDN** — validated by the email-trick (`IsValid("email", "bob@<host>")`)

Trimmed before storage. The address is stored on its own (in `parameters.name`) and also formatted into the Postfix-required bracketed literal `[host]:port` (in `parameters.parameter`) so that Postfix skips MX lookups and connects directly. Brackets are always emitted for the relay host — round-robin via MX is not part of this page's model; if you need MX-driven relay distribution, configure

DNS upstream of the brackets.

Relay Host Port

1–65535. Default `25`. The page's helper text surfaces the three common values:

Port	Typical use
`25`	Inbound MX / unauthenticated relay
`587`	Submission with STARTTLS + SASL (most modern smarthosts)
`465`	Submission over implicit TLS (SMTPS) — Postfix needs `wrappermode` adjustments not exposed on this page; prefer `587` when the smarthost supports it

Outbound TLS Mode

Maps directly to Postfix's `smtp_tls_security_level` for client connections (not to be confused with the `smtpd_tls_*` server-side settings configured under SMTP TLS Settings).

UI value	`main.cf` value	Behavior
Disabled - No TLS	parent `enabled=0` (no directive emitted)	Plaintext only; STARTTLS not attempted
Opportunistic TLS (Recommended)	`smtp_tls_security_level = may`	STARTTLS used if offered; falls back to plaintext otherwise
Mandatory TLS	`smtp_tls_security_level = encrypt`	STARTTLS required; delivery fails if the upstream does not offer it. No certificate verification — use a TLS policy for that.

Pick **may** for port 587 with STARTTLS, **encrypt** if your smarthost contract requires confirmed encryption. For verified-peer TLS to a specific smarthost, layer on a TLS policy via SMTP TLS Settings.

Authentication

When toggled on, **Username** and **Password** become required. The password input is masked-and-replaceable: it is rendered blank with the first 4 characters of the stored value shown beneath as a hint (`abcd*****`), and a blank submit keeps the existing encrypted value. Set a new value to rotate.

The handler reads `/opt/hermes/keys/hermes.key`, encrypts both fields (`AES` / `Base64`), and writes the ciphertext into `system_settings`. The decryption path is symmetric — `generate_sasl_password_transport.cfm` reads, decrypts, and writes the `[host]:port  user:pass` line to `/etc/postfix/sasl_passwd` before postmapping it.

Save flow — the cascade

Clicking **Save Settings** posts `action=save`. The handler runs a strict sequence:

```
1. Validate Enable + (if enabled) host + port + (if auth) user/pass
2. edit_relay_host_settings.cfm
   - update parameters rows (relayhost, smtp_sasl_auth_enable,
     smtp_sasl_password_maps, smtp_tls_security_level)
   - if auth: encrypt creds, write to system_settings,
     clear legacy plaintext on parameters.name
   - if not auth or disabled: clear system_settings credentials,
     disable all SASL parameter rows
   - call generate_sasl_password_transport.cfm
     -> rewrites /etc/postfix/sasl_passwd
     -> docker exec hermes_postfix_dkim postmap /etc/postfix/sasl_passwd
3. generate_postfix_configuration.cfm
   - copies /etc/postfix/main.cf to main.cf.HERMES (write-time backup)
   - copies /opt/hermes/conf_files/main.cf.HERMES template -> main.cf
   - chown root:root via docker exec hermes_postfix_dkim
   - iterates enabled parameters rows, substitutes the directive name
     and value into main.cf
   - docker exec hermes_postfix_dkim postfix reload
4. cflocation back with session.m = 10 (success banner)
```

Validation failures short-circuit with `session.m` set to the matching error code (1–6) and a redirect — no partial DB writes land.

`sasl_passwd` generation — consolidated, not per-page

`generate_sasl_password_transport.cfm` is a **shared** generator called by both this page and the Domains Add/Edit/Delete handlers. It is the single source of truth for `/etc/postfix/sasl_passwd` and rebuilds the file from scratch each invocation:

```
# /etc/postfix/sasl_passwd  (regenerated on every save)
[smtp.example.com]:587    relayuser:relaypassword       <-- this page (relay host)
[mx1.partner.com]:25      partneruser:partnerpassword   <-- Domains page (per-domain auth)
[mx2.partner.com]:25      otheruser:otherpassword       <-- Domains page (per-domain auth)
```

The relay host entry is added if **all** of:

- `smtp_sasl_auth_enable` parent is enabled
- Decrypted username AND password from `system_settings` are non-empty
- `relayhost` child value is non-empty

Per-domain entries are added from `transport` rows where `authentication = 'YES'`. Postfix uses the bracketed `[host]:port` key on the relay host line to match its own bracketed `relayhost` directive — that exact-key match is why the brackets matter.

> **Operational consequence.** Disabling the relay host on this page wipes the relay-host row from `sasl_passwd` but does **not** touch per-domain entries from the Domains page. Conversely, deleting a domain with `authentication = YES` removes only that domain's entry. The two pages compose cleanly via the shared generator.

Credential rotation

To rotate the relay host password without changing anything else:

1. Open **Email Relay > Relay Host**.
2. Type the new password into the **Password** field.
3. Click **Save Settings**.

The handler encrypts the new value into `system_settings`, `generate_sasl_password_transport.cfm` rewrites `sasl_passwd` with the decrypted new value, `postmap` rebuilds the `.db`, and Postfix picks up the change on the next outbound connection (no daemon restart needed — Postfix re-reads hash maps lazily).

Rotating the encryption key itself (`/opt/hermes/keys/hermes.key`) is handled by `rotate_db_credentials.sh` — see that script for the full re-encryption sweep across `system_settings`

and the `transport` table.

Failure semantics

What breaks	What happens
Host fails IPv4/IPv6/FQDN validation	`session.m = 2`, redirect, no DB write
Port empty or non-integer or out of range	`session.m = 3` or `4`, redirect, no DB write
Auth enabled, username blank	`session.m = 5`, redirect, no DB write
Auth enabled, password blank AND `system_settings.value` empty	`session.m = 6`, redirect, no DB write
Auth enabled, password blank but cached cipher present	Cached value is decrypted and reused; no error
Postfix template substitution fails (`generate_postfix_configuration.cfm`)	The error include surfaces the message; the **previous** `main.cf` has already been overwritten with the template copy at that point — recovery is to restore from `main.cf.HERMES` (the write-time backup the same script creates) and re-save
`docker exec hermes_postfix_dkim postfix reload` fails	The next inbound delivery attempt re-reads `main.cf`; no immediate user-facing symptom unless directives changed
`docker exec hermes_postfix_dkim postmap` fails	The new `sasl_passwd` is on disk but the `.db` lags; outbound auth uses the stale `.db` until the next successful postmap

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_relay_host.cfm`	`hermes_commandbox`	Page
`config/hermes/var/www/html/admin/2/inc/get_relay_host_settings.cfm`	`hermes_commandbox`	Load handler + legacy-cred migration
`config/hermes/var/www/html/admin/2/inc/edit_relay_host_settings.cfm`	`hermes_commandbox`	Save handler
`config/hermes/var/www/html/admin/2/inc/generate_sasl_password_transport.cfm`	`hermes_commandbox`	Consolidated `sasl_passwd` generator (shared with Domains page)
`config/hermes/var/www/html/admin/2/inc/generate_postfix_configuration.cfm`	`hermes_commandbox`	Template-to-`main.cf` renderer + `postfix reload`
`/opt/hermes/conf_files/main.cf.HERMES`	`hermes_commandbox`	Postfix template Hermes renders from
`/etc/postfix/main.cf`	`hermes_postfix_dkim` (volume-mounted)	Live Postfix config (regen target)

Path	Owner	Role
`/etc/postfix/main.cf.HERMES`	`hermes_postfix_dkim` (volume-mounted)	Write-time backup created on every regen
`/etc/postfix/sasl_passwd`	`hermes_postfix_dkim` (volume-mounted)	Plain-text credentials file (regen target)
`/etc/postfix/sasl_passwd.db`	`hermes_postfix_dkim`	postmap-built hash database
`/opt/hermes/keys/hermes.key`	`hermes_commandbox`	Symmetric key for AES/Base64 cred encryption
`system_settings` rows `relay_host_username`, `relay_host_password`	`hermes_db_server`	Encrypted credential storage
`parameters` rows: `relayhost`, `smtp_sasl_auth_enable`, `smtp_sasl_password_maps`, `smtp_tls_security_level` (each as `child=2` parent + `child=1` value)	`hermes_db_server`	Postfix directive driver rows

Every shell-out uses `docker exec hermes_postfix_dkim ...` per the standard Hermes pattern; nothing on this page touches the host's own Postfix (there is none).

Related

- Domains — companion page for inbound relay-mode domains. The two pages share `generate_sasl_password_transport.cfm` and together define the entire relay topology.
- Relay Networks — `mynetworks` (which clients are allowed to relay outbound without authentication). Independent of this page but part of the same outbound story.
- Relay Recipients — recipient validation for inbound relay-mode domains; complements Domains.
- SMTP TLS Settings — outbound TLS policy per destination (peer verification, cipher pinning). The TLS Mode dropdown on this page sets the *default* level; per-destination policies override.
- Server Setup — Postfix `myorigin` / `myhostname` and host IP. Defines the identity the relay host sees in EHLO/MAIL FROM.

Relay Networks

Relay Networks

Admin path: **Email Relay > Relay Networks** (`view_relay_networks.cfm`, `inc/get_relay_networks.cfm`, `inc/generate_postfix_configuration.cfm`).

This page manages the **operator-additive list of trusted IPs and CIDR networks** that are allowed to relay mail through the gateway without SMTP authentication. The list is composed into Postfix's `mynetworks` directive alongside two hardcoded baseline entries (`127.0.0.1` and the Docker subnet) and propagated to Amavis's `@inet_acl` so the content filter trusts the same source IPs. Every directive listed in `mynetworks` matches the `permit_mynetworks` clause at the head of `smtpd_recipient_restrictions` and bypasses RBL, sender, and recipient checks — misconfiguring it turns the gateway into an open relay.

This is the **trusted-sender** half of the inbound-control story. Pairs with Relay Recipients (the trusted-target list) and Relay Host / Domains (the outbound/forwarding configuration).

When you add entries to this page

Scenario	What to add
On-prem mail server submits outbound via Hermes	The mail server's LAN IP or `/32` CIDR
Multifunction printer with scan-to-email	The printer's IP
Backup MTA / monitoring system that sends alerts	The host's IP
Branch-office router doing NAT for relay clients	The router's public `/32`
Microsoft 365 sending via inbound connector to Hermes	M365 outbound SMTP source ranges (large, vendor-published)
Application server with a built-in mailer	The app server's IP

If the source authenticates via SMTP AUTH (a Relay Recipient with a password), it does **not** need to be listed here — `permit_sasl_authenticated` covers it via the credential path.

What `mynetworks` controls — the open-relay risk

```
inbound SMTP (25/587)
        |
        v
hermes_postfix_dkim  (smtpd_recipient_restrictions)
        |
        |  permit_mynetworks                    <-- bypasses all checks below
        |  permit_sasl_authenticated            <-- bypasses checks for authenticated senders
        |  reject_unauth_destination            <-- rejects everything else
        |  reject_unauth_pipelining
        |  check_sender_access mysql:...
        |  reject_*_hostname / reject_*_sender   <-- RBL + hygiene checks
        |  check_policy_service unix:.../policy-spf
        |
        v
accept -> amavis content filter (10024)
```

Any IP listed in `mynetworks` clears `permit_mynetworks` and skips **every other restriction** — RBL lookups, sender domain checks, SPF, recipient domain checks. The same IP also clears Amavis's `@inet_acl` because the file `/etc/amavis/mynetworks` is regenerated from the identical list on every Apply.

> **By design.** Listing an IP here gives the host **unrestricted relay** through the gateway. Add only IPs you control or fully trust. A broad CIDR (anything wider than `/24`) is a red flag. A wildcard entry like `0.0.0.0/0` makes Hermes an open relay reachable from the public Internet — the page does not block such entries but the operational consequence is immediate inclusion on blocklists. Audit periodically.

Hardcoded baseline — what's already trusted

Two entries are seeded into the `parameters` table at install time and are intentionally hidden from this page's table (excluded by `AND parameter <> '127.0.0.1' AND parameter <> '172.16.32.0/24'` in `get_relay_networks.cfm`):

Entry	Source	Purpose
`127.0.0.1`	`hermes_install.sql` seed (`parameters.id=357`)	Localhost — Hermes's own internal Postfix submission, Amavis re-injection on `10025`, scheduler cron jobs, etc.
`172.16.32.0/24`	`hermes_install.sql` seed (`parameters.id=434`)	Default Docker subnet — covers every other Hermes container (CommandBox, OpenLDAP, Authelia, body milter, etc.) talking to Postfix

These are mandatory for normal operation and the page deliberately hides them so they cannot be deleted from the UI. Removing either breaks intra-container submission immediately.

> **Operational consequence.** The Docker subnet is hardcoded to `172.16.32.0/24` in the seed row above and in the `IPV4SUBNET=172.16.32` entry in `.env`. Changing the subnet requires editing both the seed row and `.env` plus a sweep of other config files that reference the same literal (Postfix, Amavis, Dovecot, Ciphermail, OpenDKIM/OpenDMARC, CFML queries). A future change will template this — for now, leave the subnet at the default unless you have a specific routing reason to change it.

Configuration storage — the dual-row pattern

Relay networks live in the `parameters` table using the standard parent-child layout shared by every Postfix directive Hermes manages:

Row	`parameter` column	`child`	`parent_name`	Purpose
Parent (one per directive)	`mynetworks`	`2`	NULL	The directive itself; carries `enabled` and the original description
Child (one per IP/network)	the actual IP or CIDR (e.g. `192.168.50.0/24`)	`1`	`mynetworks`	The value Postfix sees in the comma-separated list

The page reads the parent ID from the parent row (`get_mynetworks_parent`) and uses it as the `parent` foreign key on every child row. `generate_postfix_configuration.cfm` walks all enabled children of the parent in `order1` order and emits them comma-separated into `/etc/postfix/main.cf`.

Extra columns on the child row drive the page's UX:

Column	Values	Used for
`network_entry`	`0` / `1`	`1` when the entry has a `/` (CIDR); `0` for single IPs. Drives the **Network** / **IP** badge in the table.
`note`	free text	Optional admin label (e.g. "Office Printer", "Branch Office VPN"). Plain-text, HTML-encoded on render.
`enabled`	`0` / `1`	Always `1` in normal use; rows are deleted rather than disabled.
`applied`	`1` / `2`	`1` = currently live in `main.cf`; `2` = staged change, not yet applied.
`action`	`NONE` / `insert` / `delete` / `APPLY`	What the next Apply Settings cycle will do with this row.
`order1`	integer	Sort order. New rows append at `MAX(order1) + 1` so existing ordering is preserved.

Staged-edit model — pending changes don't take effect immediately

Unlike most pages in the admin console (which save directly), Relay Networks uses a **two-step commit**: edits are staged in the DB with `applied=2`, then a single **Apply Settings** click flushes everything to Postfix in one cascade.

```
add / edit / delete  ──► row marked applied=2 + action={insert|delete|APPLY}
                                    │
                                    v
                          Pending Changes banner appears
                                    │
                                    v
```

```
                        Apply Settings (action=apply)
                                  |
                                  ├─ DELETE rows with action='delete'
                                  ├─ UPDATE applied=1, action='NONE' for inserts
                                  ├─ UPDATE applied=1, action='NONE' for edits
                                  |
                                  v
                       generate_postfix_configuration.cfm
                                  |
                                  ├─ rewrite /etc/postfix/main.cf from template
                                  ├─ rewrite /etc/amavis/mynetworks
                                  ├─ docker exec hermes_postfix_dkim postfix reload
                                  └─ docker exec hermes_mail_filter /etc/init.d/amavis
force-reload
```

This is intentional. A relay-networks change is a security-sensitive event — staging lets you queue several edits, eyeball the **Pending Additions** / **Pending Deletions** / **Pending Edits** cards (each shown only when its respective query returns rows), then commit in a single reload. **Cancel All Additions** and **Cancel All Deletions** buttons let you back out a pending change before applying.

Bulk-add textarea — format and validation

The Add IP/Network card takes a multi-line textarea. Each non-blank line is parsed independently and either accepted or appended to a `skipped` summary that surfaces in the success/error alert.

Format per line:

```
<IP or CIDR> [optional note]
```

Example input line	Result
`192.168.1.100 Office Printer`	IP `192.168.1.100`, note `Office Printer`
`192.168.1.101`	IP `192.168.1.101`, note `192.168.1.101` (defaults to the address)
`10.0.0.0/24 Server Network`	CIDR `10.0.0.0/24`, note `Server Network`
`192.168.1.300`	Skipped — fails IPv4 octet range check
`10.0.0.0/45`	Skipped — CIDR out of 1–32 range

Validation rules in `view_relay_networks.cfm`:

Check	Pattern	Failure
IPv4 octets	`^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.{3}…`	`Invalid IP address` / `Invalid network address`
CIDR mask	Integer 1–32	`Invalid CIDR mask`
Octet normalization	`Int(octet)` on each	`192.168.001.005` becomes `192.168.1.5` so duplicates can't sneak in via leading zeros
Duplicate check	`SELECT … WHERE parameter = ? AND parent = mynetworks_parent_id AND child = '1'`	`Already exists` (skipped silently in bulk)

IPv6 is **not** supported by this page — the validator pattern only accepts dotted-quad IPv4. If you need IPv6 relay sources, add them directly to `parameters` with the same column layout and run a manual Apply through the UI.

Single-row Edit modal

The Edit pencil opens a Bootstrap modal pre-filled with the row's current IP/Network and note. Two edit modes:

Change	Behavior
Note only changed	Updates the `note` column immediately (no config change) — success banner only, no Apply required
IP/Network changed	Sets `applied=2, action='APPLY'`; Apply Settings is required to push to Postfix

The IP duplicate check (`AND id <> form.edit_id`) lets you edit a row to itself (no-op) but blocks renaming to another row's value.

Bulk delete

The DataTables checkbox column lets you select multiple rows and stage them all for deletion in one shot. Submission goes through the same `bulk_delete` action — each selected row is marked `applied=2, action='delete'`, the **Pending Deletions** card appears, and Apply Settings purges them.

A confirm dialog (`Are you sure you want to delete N selected entries?`) fires before the form submits.

How a saved network reaches Postfix and Amavis

`generate_postfix_configuration.cfm` is the same template-render + postfix-reload helper shared by Relay Host, Domains, and other Postfix-directive pages. For `mynetworks` specifically:

```
1. Substitute every enabled parameters child into the main.cf template
   (mynetworks line becomes "mynetworks = 127.0.0.1, 172.16.32.0/24,
   <every IP/CIDR you added>")
2. cffile write /etc/amavis/mynetworks  -- one entry per line
3. docker exec hermes_postfix_dkim postfix reload
4. docker exec hermes_mail_filter /etc/init.d/amavis force-reload
```

Both Postfix and Amavis trust the same list, so a relay source bypassing SMTP-time checks also bypasses content-filter network checks.

Failure semantics

What breaks	What happens
Textarea empty	`session.m = 30`, redirect, no DB write
All entries fail validation	`session.m = 32`, redirect, summary of skipped entries shown
Mixed: some valid, some invalid	`session.m = 31`, success count + skipped count + collapsible error list
Edit IP changed but duplicate of another row	`session.m = 23`, redirect with the conflicting value surfaced
Bulk delete with no rows checked	`session.m = 16`, redirect
Apply Settings runs but `postfix reload` fails	`session.m = 20` still fires (the page treats reload as best-effort); inspect `docker logs hermes_postfix_dkim` for the error. Previous `main.cf` is preserved in `main.cf.HERMES.BACKUP`.
Apply Settings runs but `amavis force-reload` fails	`generate_postfix_configuration.cfm` aborts with the error surfaced via `error.cfm`; Postfix has already been reloaded, so SMTP-time trust is updated but Amavis is still on the previous list. Re-run Apply to recover.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_relay_networks.cfm`	`hermes_commandbox`	Page + bulk-add / edit / delete handlers
`config/hermes/var/www/html/admin/2/inc/get_relay_networks.cfm`	`hermes_commandbox`	Load queries (active + pending splits)
`config/hermes/var/www/html/admin/2/inc/generate_postfix_configuration.cfm`	`hermes_commandbox`	Template-to-`main.cf` renderer + amavis `mynetworks` writer + reload calls
`/etc/postfix/main.cf`	`hermes_postfix_dkim` (volume-mounted)	Live Postfix config; the `mynetworks = …` line is rewritten on every Apply
`/etc/postfix/main.cf.HERMES.BACKUP`	`hermes_postfix_dkim`	Pre-regen backup
`/etc/amavis/mynetworks`	`hermes_mail_filter` (volume-mounted)	One entry per line; `@inet_acl` source
`parameters` row `mynetworks` (child=2, id=3) + N children (child=1, parent=3)	`hermes_db_server`	Directive parent + per-entry children

Every shell-out uses `docker exec hermes_postfix_dkim …` / `docker exec hermes_mail_filter …` per the standard Hermes pattern.

Related

- Relay Recipients — the recipient-validation list. Together they answer "which sources are trusted to relay (this page) and which destinations does Hermes accept inbound mail for (Relay Recipients)?"
- Relay Host — outbound smarthost. A client trusted by this page that sends outbound mail still flows through the relay host (if configured) on the way out.
- Domains — inbound relay-domain definitions. Domain recipient-validation mode (`OK` / `SPECIFIED`) interacts with Relay Recipients but is independent of this page.
- LDAP RemoteAuth — alternative trust path. A RemoteAuth-mode Relay Recipient authenticates against an upstream AD/LDAP and is admitted via `permit_sasl_authenticated`, not `permit_mynetworks` — adding their source IP here is unnecessary (and weakens the audit trail).
- Authentication Settings — broader picture of how SMTP AUTH, mynetworks, and the `smtpd_recipient_restrictions` chain interact.

Relay Recipients

Relay Recipients

Admin path: **Email Relay > Relay Recipients** (`view_internal_recipients.cfm`, `add_internal_recipients.cfm`, `edit_internal_recipient_backend.cfm`, `inc/delete_internal_recipients.cfm`, `inc/edit_internal_recipients.cfm`, `inc/edit_internal_recipients_djigzo.cfm`, `inc/get_int_recipient_json.cfm`, `inc/send_recipient_welcome_email.cfm`, `inc/send_recipient_welcome_email_remoteauth.cfm`).

> **The page filename is `view_internal_recipients.cfm`, not `view_relay_recipients.cfm`.** The original concept was "internal" recipients (mail accepted into the gateway and forwarded to an internal backend); the UI label was renamed to **Relay Recipients** in commit `c547fdd9` but the filename, table column `recipients.recipient_type='relay'`, and several handler names still carry the legacy `internal_recipients` naming. Treat the two terms as synonymous.

This page manages the **per-address recipient roster** for relay-mode domains — the list of mailboxes Hermes accepts inbound mail for and forwards downstream, and the list of authenticated senders that can relay outbound mail through the gateway. Each row in the `recipients` table is one email address with a stack of per-recipient settings: SVF policy, quarantine notifications, encryption flags (PDF/S/MIME/PGP), S/MIME certificate + PGP keyring slots, backend override, auth mode (local vs RemoteAuth), and 2FA enforcement.

This is the **recipient-validation** half of the relay topology. Pairs with Domains (the domains those recipients live under), Relay Networks (the trusted source IPs), and Virtual Recipients (alias-only addresses that forward without a real account).

Relay Recipient vs Virtual Recipient vs Mailbox

Three different recipient concepts share the email-address namespace in Hermes — keep them straight:

Concept	Stored in	Has a local account?	Delivered to
Relay Recipient (this page)	`recipients` where `recipient_type='relay'`, `domain IS NULL`	Yes — LDAP entry + optional app passwords	Downstream MX (per `domains` row's `transport`)
Virtual Recipient	`virtual_recipients`	No — alias only	Rewrites to another address, which then needs a Relay Recipient or external destination
Mailbox	`mailboxes` (separate `mailbox_domains` topology)	Yes — Dovecot mailbox	Local Dovecot LMTP at `/mnt/vmail`

A Relay Recipient is the only one of the three that authenticates for outbound submission (SMTP AUTH on port 587) and for web/portal login (via Authelia). Virtual Recipients are pure forwarding rules; Mailboxes are the mail-server-topology equivalent. See Email Server > Mailboxes for the Mailbox flow.

What a Relay Recipient row carries

```
recipients table  (one row per email address)
├── recipient                jsmith@company.com
├── recipient_type           'relay'
├── domain                   NULL   (domain rows use domain='1')
├── auth_type                'local' | 'remote'
├── remoteauth_domain        NULL if local; mapping key if remote
├── enforce_mfa              0 | 1   (admin policy — see #225 Phase 2)
├── policy_id  ──────────►   spam_policies.policy_id (SVF policy)
├── pdf_enabled / smime_enabled / pgp_enabled / digital_sign
├── backend_server / backend_port / backend_tls   (per-recipient override)
└── (cert+keyring slots populated lazily by the queue)
```

Side tables linked at create/edit time:

Table	What it stores
`user_settings`	Per-user portal toggles (`report_enabled`, `train_bayes`, `download_msg`), `ldap_username`, mailbox flags
`recipient_certificates`	S/MIME certs issued for the recipient (lazy — populated by `cert_generation_queue`)
`recipient_keystores`	PGP keyrings (lazy — same queue)

Table	What it stores
`app_passwords`	Per-application passwords (Argon2-hashed) for IMAP/SMTP/CalDAV/CardDAV/Nextcloud — see Credential Model
`wblist`	Whitelist/blacklist entries owned by the recipient
`cert_generation_queue`	Pending S/MIME and PGP generation jobs

Add Recipient(s) — `add_internal_recipients.cfm`

The Add Recipient(s) button navigates to a multi-line input form that creates many recipients in one submission. Three add modes:

Local-auth bulk add — one email per line

When **Auth Type** is `Local` (the default), the textarea takes one email per line. The page generates a random password for each new recipient, sends a welcome email via `send_recipient_welcome_email.cfm` that includes a **first-login password-reset link**, and stores the LDAP entry with a placeholder `userPassword` that will be overwritten when the user follows the link.

```
jsmith@company.com
jdoe@company.com
bob.smith@company.com
```

RemoteAuth bulk add — same line format

When **Auth Type** is `Remote` and the selected mapping's DN pattern only uses `{username}` and/or `{email}`, the textarea is still one email per line. No password is generated — the recipient authenticates against the upstream LDAP/AD via the `remoteauth` overlay (see LDAP RemoteAuth). The welcome email goes through `send_recipient_welcome_email_remoteauth.cfm` and tells the user to sign in with their **organization password**, not a Hermes-issued one.

RemoteAuth CSV add — `First,Last,Email` per line

When the RemoteAuth mapping's DN pattern uses `{firstname}` or `{lastname}` (typical for AD `cn=` patterns), the textarea **switches to CSV mode** because email-only input doesn't carry enough data to expand the pattern. Header rows (`"GivenName","Surname","Mail"`) are auto-detected and skipped, and unknown columns are ignored.

Source	Command / file shape
PowerShell	`Get-ADUser -Filter * -Properties GivenName,Surname,Mail | Select GivenName,Surname,Mail | Export-Csv users.csv -NoTypeInformation`
CSVDE (Windows Server built-in)	`csvde -f users.csv -l "givenName,sn,mail"`
Excel / manual	Three columns saved as CSV

See LDAP RemoteAuth § Adding RemoteAuth users in bulk for the full CSV format reference.

The Add form also accepts the same per-recipient stack of options as the Edit Options modal (SVF policy, quarantine notifications, etc.) — those defaults are written to every new row in one shot.

The Recipients table

Sortable, searchable, exportable (copy/CSV/Excel/PDF/print via DataTables Buttons; `stateSave: true`). Columns:

Column	Source	Notes
Checkbox	—	Multi-select for the action buttons above the table
S/MIME	link to `view_recipient_certificates.cfm?type=1&id=…`	Per-recipient cert manager
PGP	link to `view_recipient_keyrings.cfm?type=1&id=…`	Per-recipient keyring manager
Recipient	`recipients.recipient`	Email address
Auth	`recipients.auth_type` + `remoteauth_domain`	`LOCAL` badge (secondary) or `REMOTE` badge (primary, tooltip shows mapping key)
Backend	`recipients.backend_server[:port]`	Per-recipient override or `(domain default)` placeholder
2FA	LDAP `cn=two_factor` + `enforce_mfa`	**Two independent pills** — see Two-pill 2FA column below
Policy	`policy.policy_name` via join	Assigned SVF policy
Quarantine Notifications	`user_settings.report_enabled`	`YES` / `NO` badge

Column	Source	Notes
Train Bayes	`user_settings.train_bayes`	`YES` / `NO`
Download Msgs	`user_settings.download_msg`	`YES` / `NO`
PDF / S/MIME / PGP Encrypt	per-row encryption flags	`YES` / `NO` badges
Sign All	`recipients.digital_sign`	`YES` / `NO`
S/MIME Cert	join against `recipient_certificates`	`YES` (green badge) if a cert exists
PGP Keyring	join against `recipient_keystores`	`YES` (green badge) if a keyring exists

The query filters `WHERE recipients.domain IS NULL AND (recipient_type = 'relay' OR recipient_type IS NULL)` so only relay-mode rows appear — mailbox-topology rows (with `recipient_type='mailbox'`) are managed under Email Server > Mailboxes.

Two-pill 2FA column

The 2FA column shows **two orthogonal states** as independent pills, because admin enforcement and user enrollment are decoupled (#225 Phase 1.5 + Phase 2):

Pill	Source	Means
Enrolled (success badge)	LDAP `cn=two_factor` group membership	The user has registered a 2FA device (TOTP, security key, or Duo Push) and Authelia challenges them at sign-in
Required (warning badge)	`recipients.enforce_mfa = 1`	Admin policy demands 2FA. The recipient sees an urgent banner in the user portal directing them to Account Settings until they enroll

Enrolled	Required	What it looks like	Means
no	no	em-dash	Default state. No 2FA.
yes	no	Enrolled only	Voluntary enrollment. User opted in; admin doesn't enforce.
no	yes	Required only	Admin set the policy; user hasn't yet registered a device.
yes	yes	Both pills	Required and complied with.

The single LDAP `ldapsearch` query against `cn=two_factor,ou=groups,dc=hermes,dc=local` runs once per page render, then each row checks for its DN substring in the result — avoids N+1 LDAP

roundtrips.

Bulk action buttons

Button	Action	Selection requirement
Create Recipient(s)	Navigates to `add_internal_recipients.cfm`	—
Edit Options	Opens the Edit Options modal	At least one row
Edit Encryption	Opens the Edit Encryption modal	At least one row
Edit Backend	Navigates to `edit_internal_recipient_backend.cfm?ids=…`	At least one row
Reset 2FA Devices	Opens the Reset 2FA Devices modal	At least one row
Delete	Opens the delete-confirm modal	At least one row

Selecting zero rows and clicking any of the edit/delete buttons surfaces an alert (`Please select at least one recipient`) instead of opening the modal.

Edit Options modal — AJAX pre-fill vs bulk-edit warning

The Edit Options modal handles SVF policy, quarantine notifications, Train Bayes, Download Messages, and 2FA enforcement (`enforce_mfa`). It has **two modes**, selected by the JS based on how many rows are checked:

Single-select: AJAX pre-fill

When exactly one row is checked, the JS calls `./inc/get_int_recipient_json.cfm?id=<rid>` over POST and hydrates every form field with that recipient's current values before opening the modal. The admin sees the recipient's actual policy, current notification mode, current `enforce_mfa` state, etc. — submit edits only what changed.

Multi-select: bulk-edit warning

When 2+ rows are checked, the modal shows a prominent red **Bulk edit — N recipients selected** alert at the top:

> The fields below are **not pre-filled from each recipient's current settings** — they show the form's default values. Submitting will **OVERWRITE every field on every selected recipient** with whatever you see now.

The 2FA-specific footnote then warns that leaving the Two-Factor Authentication dropdown at `Disable` will reset every selected recipient's `enforce_mfa` to `0` — but **the user is not removed from `cn=two_factor` automatically** (the LDAP cascade only fires on 0→1 transitions). To strip an existing enrollment, the admin must use the Reset 2FA Devices modal with the nuclear-option checkbox.

This is intentional — the bulk-edit form has been a foot-gun in the past (admins thinking "Disable" only changed the one row), so the warning is unmissable. The recommended pattern: **edit a single recipient with their current values pre-filled, select only one row.**

Edit Encryption modal

Handles `pdf_enabled`, `smime_enabled`, `digital_sign`, `pgp_enabled`, and the cert/keyring generation parameters (CA, validity, key size, algorithm, PGP key length). Submit triggers `edit_internal_recipients_djigzo.cfm` which updates the row and **queues async S/MIME cert + PGP keyring generation** into `cert_generation_queue` if the flags flip on and no existing cert/keyring is present.

The page renders a **Background Generation in Progress** info banner while `cert_generation_queue` has any `pending` or `processing` rows, and a **Generation Failures** warning with a **Retry Failed Jobs** button if any rows are in `failed` state. The Retry button updates matching rows to `status='pending', error_message=NULL, started_at=NULL` so the next scheduler tick re-attempts them.

Edit Backend page

Per-recipient override of the downstream backend server / port / TLS mode. The default is `NULL` on all three columns, which falls back to the parent domain's `transport` row (set on the Domains page). Useful for routing specific recipients to a different MX — e.g., a single user whose mailbox is on a different server than the rest of the domain.

The Backend column on the main table shows the override host (and port via tooltip) or `(domain default)` for the fallback case.

Reset 2FA Devices modal

Replaces the older "Recipient Access Control" modal as of #225 Phase 2. The one_factor/two_factor radio is gone — the canonical admin policy is the **Two-Factor Authentication** select on Edit Options. This modal is now single-purpose: clear Authelia TOTP/WebAuthn devices for the selected recipients via `docker exec hermes_authelia authelia storage user totp/webauthn delete`.

Two modes:

Mode	What it does
Default	Deletes TOTP + WebAuthn device registrations in Authelia. User stays under 2FA enforcement and re-registers on next sign-in. "User lost their phone" recovery.
Nuclear (checkbox)	Also moves the user from `cn=two_factor` back to `cn=one_factor`. Admin override of voluntary enrollment, or full account reset.

> **Does not affect Duo Push.** Duo enrollments live on Duo's cloud servers, not in Authelia's database. Use the Duo Admin Console for Duo device management.

> **Cascade interaction.** If the per-recipient `enforce_mfa` policy in Edit Options is still `Enable`, the nuclear option's removal from `cn=two_factor` will be **reversed** on the next save of the Edit Options modal (the 0→1 LDAP cascade fires again). To truly de-enforce, set `enforce_mfa = Disable` first.

Delete

The Delete modal confirms the irreversible action. The `delete_internal_recipients.cfm` handler then runs an unusually-long cleanup sequence per recipient — the kind of cascade that makes orphan rows the rule when CFML deletes are skimped:

```
For each selected recipient ID:
1. Look up ldap_username via user_settings join
```

```
2. docker exec hermes_authelia authelia storage user totp delete <user>
3. docker exec hermes_authelia authelia storage user webauthn delete <user> --all
4. ldap_delete_user_relay.cfm — remove LDAP stub entry + group memberships
5. Cancel any pending password_reset_requests rows for this email
6. DELETE FROM recipients WHERE id = <rid>
7. DELETE FROM recipients_temp WHERE recipient = <email>
8. DELETE FROM wblist WHERE rid = <rid>
9. DELETE FROM user_settings WHERE email = <email>
10. DELETE FROM mailaddr (and wblist by sid) for the address
11. Delete recipient_certificates + cm_keystore from djigzo
12. (caller continues with the next ID)
```

Steps 2–3 prevent a re-created recipient at the same email from silently inheriting the prior owner's TOTP/WebAuthn enrollments. Failures inside `cftry` blocks are non-fatal — the desired end-state ("no devices") is achieved whether or not the user had anything enrolled in the first place.

> **Known gap (#102).** When a Relay Recipient with `auth_type='remote'` is deleted, the deletion of the LDAP stub entry happens, but the RemoteAuth domain-mapping deletion validation in `view_remoteauth.cfm` / `edit_remoteauth_mapping.cfm` does **not** check the `mailboxes` table yet (it only checks `system_users` and `recipients`). When RemoteAuth is wired to mailboxes, that validation must add a third query. Not a bug today — relay recipients are correctly covered — but a forward-looking integration point. See LDAP RemoteAuth § Deletion validation.

Local-auth vs RemoteAuth — the credential split

Aspect	`auth_type = 'local'`	`auth_type = 'remote'`
Web portal sign-in	Hermes LDAP `userPassword` (user sets via reset link)	Upstream AD/LDAP via overlay; Hermes never sees the password
IMAP / SMTP / CalDAV / CardDAV / NC	`app_passwords` row (Argon2-hashed in Hermes DB)	Same — `app_passwords` row in Hermes DB
Password rotation on the upstream	N/A	Web sign-in immediately picks up the new password; existing app passwords keep working until explicitly revoked

Aspect	`auth_type = 'local'`	`auth_type = 'remote'`
Welcome email	"Click here to set your password"	"Sign in with your organization (AD/LDAP) password"

App passwords are **always Hermes-issued**, regardless of `auth_type`. The upstream directory password is exposed only to the web gate via the LDAP overlay's pass-through bind — never to Dovecot or Nextcloud. See Authentication Settings for the full four-credential architecture and LDAP RemoteAuth for the upstream binding details.

Recipient validation in Postfix

The `recipients` table is queried by Postfix at SMTP time via `mysql:/etc/postfix/mysql-recipients.cf` (mapped to `relay_recipient_maps` in `main.cf`). When a Domain has Recipient Delivery set to `SPECIFIED`, mail arriving for an address **not** in this table is rejected with a `550 User unknown` reply. When Recipient Delivery is `ANY`, the lookup is bypassed for that domain and any recipient is accepted (catch-all).

This is the operational reason to add Relay Recipients **before** flipping a domain to SPECIFIED — flipping first will start rejecting live mail.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_internal_recipients.cfm`	`hermes_commandbox`	Main page + Edit Options / Edit Encryption / Reset 2FA / Delete modals
`config/hermes/var/www/html/admin/2/add_internal_recipients.cfm`	`hermes_commandbox`	Bulk-add page (local + RemoteAuth + CSV modes)
`config/hermes/var/www/html/admin/2/edit_internal_recipient_backend.cfm`	`hermes_commandbox`	Per-recipient backend override page
`config/hermes/var/www/html/admin/2/inc/get_int_recipient_json.cfm`	`hermes_commandbox`	AJAX hydrator for single-select Edit Options pre-fill
`config/hermes/var/www/html/admin/2/inc/edit_internal_recipients.cfm`	`hermes_commandbox`	Edit Options handler (+ LDAP cascade on `enforce_mfa` 0→1)
`config/hermes/var/www/html/admin/2/inc/edit_internal_recipients_djigzo.cfm`	`hermes_commandbox`	Edit Encryption handler + cert/keyring queue insertion
`config/hermes/var/www/html/admin/2/inc/delete_internal_recipients.cfm`	`hermes_commandbox`	Per-recipient delete cascade

Path	Owner	Role
`config/hermes/var/www/html/admin/2/inc/send_recipient_welcome_email.cfm`	`hermes_commandbox`	Local-auth welcome email (password-reset link)
`config/hermes/var/www/html/admin/2/inc/send_recipient_welcome_email_remoteauth.cfm`	`hermes_commandbox`	RemoteAuth welcome email (org-password sign-in)
`config/hermes/var/www/html/admin/2/inc/ldap_add_user_relay.cfm` / `ldap_add_user_relay_remoteauth.cfm`	`hermes_commandbox`	LDAP stub creation for local / remote auth
`config/hermes/var/www/html/admin/2/inc/ldap_delete_user_relay.cfm`	`hermes_commandbox`	LDAP stub removal on delete
`config/hermes/var/www/html/admin/2/inc/ldap_change_user_access_control.cfm`	`hermes_commandbox`	Group membership swap (one_factor ⇄ two_factor)
`recipients`, `user_settings`, `app_passwords`, `recipient_certificates`, `recipient_keystores`, `cert_generation_queue`, `wblist`, `mailaddr`, `password_reset_requests`, `recipients_temp`	`hermes_db_server`	The recipient-row group + lazy-generation queue
`cn=<user>,ou=users,dc=hermes,dc=local`	`hermes_ldap`	Per-recipient LDAP entry
`cn=relays,ou=groups,dc=hermes,dc=local`	`hermes_ldap`	Relay-recipient group membership
Authelia `totp_configurations` + `webauthn_devices`	`hermes_authelia` storage backend	Cleaned on delete + Reset 2FA Devices
`/etc/postfix/mysql-recipients.cf`	`hermes_postfix_dkim`	Postfix lookup against `recipients` for `relay_recipient_maps`

Every shell-out uses `docker exec …` per the standard Hermes pattern.

Related

- Domains — relay-domain definitions. Required parent context: a recipient is meaningless without a domain that accepts mail for it. Domain Recipient Delivery `SPECIFIED` is what makes this page's roster authoritative for inbound acceptance.
- Relay Networks — trusted source IPs. The alternative trust path: a source IP listed there can submit outbound without authenticating as a recipient on this page.
- Virtual Recipients — alias-only addresses that forward to a Relay Recipient or external destination. A Virtual Recipient pointing at a deleted Relay Recipient becomes a forwarding hole.
- Relay Host — outbound smarthost. A Relay Recipient that SMTP-AUTHs to send outbound mail still flows through the relay host (if configured) on the way to the Internet.

- LDAP RemoteAuth — required prerequisite for `auth_type='remote'` recipients. Defines the upstream LDAP/AD mappings this page references via `remoteauth_domain`.
- Authentication Settings — full four-credential architecture (web vs IMAP/SMTP vs DAV vs Nextcloud) that recipient app passwords slot into.
- Email Server > Mailboxes — the mail-server-topology equivalent. Don't confuse Relay Recipients (forwarded downstream) with Mailboxes (delivered locally to Dovecot).

Virtual Recipients

Virtual Recipients

Admin path: **Email Relay > Virtual Recipients** (`view_virtual_recipients.cfm`, `inc/addvirtualrecipients.cfm`, `inc/editvirtualrecipient.cfm`, `inc/delete_virtual_recipients.cfm`).

This page manages **forward-only address aliases** on the relay-topology domains configured under Domains. Each row in the `virtual_recipients` table maps one inbound address (or a domain-wide catch-all) to exactly one delivery address. The delivery target can be internal to Hermes, on another relay domain, on a mailbox domain, or anywhere on the public Internet — the row is consumed by Postfix's `virtual_alias_maps` and rewritten at SMTP time, so the forward is transparent to the original sender.

Virtual recipients have **no SMTP authentication, no IMAP/POP3 access, and no password**. They are not user accounts. They are rewrite rules.

Not the same as Mailbox Aliases

The Email Server topology has its own alias page — Email Server > Aliases, backed by the `mailbox_aliases` table — and it serves a different need. The add handler enforces the separation explicitly: trying to add a virtual recipient for a domain flagged as `mailbox` is rejected with the "use Email Server > Aliases" hint.

	Virtual Recipients	Mailbox Aliases
Table	`virtual_recipients`	`mailbox_aliases`
Domain type	Relay domains (`domains.type = 'relay'` or NULL)	Mailbox domains (`mailbox_domains.*`)
Delivery target	Anywhere — internal or external	A local Dovecot mailbox
Resolved by	Postfix `virtual_alias_maps` (MySQL lookup)	Postfix `virtual_alias_maps` (same query, different table)
Auth, IMAP, password	No	No (the resolved mailbox owns those)
Typical use	`info@company.com → admin@company.com, info@externalpartner.example`	`support@company.com → user1@company.com` (where `user1@` is a local mailbox)

The shared `mysql-virtual.cf` lookup is a `UNION` across both tables:

```
SELECT maps          FROM virtual_recipients WHERE virtual_address = '%s'
UNION
SELECT delivers_to   FROM mailbox_aliases    WHERE alias_address   = '%s'
```

Postfix doesn't care which table the answer comes from — but the admin UI separates them so the rule for each topology stays focused.

Storage and lookup path

```
inbound SMTP (port 25) ──► hermes_postfix_dkim
                                  │
                                  │  smtpd checks: helo, sender, recipient
                                  │  relay_recipient_maps / recipient_canonical_maps
                                  │  virtual_alias_maps  ◄── mysql:/etc/postfix/mysql-
virtual.cf
                                  │                          │
                                  │                          ▼
                                  │      ┌──────────────────────────────────────┐
                                  │      │ hermes_db_server                     │
                                  │      │  SELECT maps FROM virtual_recipients │
                                  │      │   UNION                              │
                                  │      │  SELECT delivers_to FROM             │
                                  │      │   mailbox_aliases                    │
                                  │      └──────────────────────────────────────┘
                                  │
                                  v
                           rewritten recipient(s)
                                  │
                                  ▼
                         content filter (amavis on 10024)
                                  │
                                  ▼
                         outbound or local delivery
```

No file regeneration is required when virtual recipients change. The MySQL lookup is live — adding a row in the admin UI takes effect on the next inbound message, with zero Postfix restart or postmap step. This is the operational reason virtual aliases are stored in MySQL rather than a hash

file.

The `virtual_recipients` table

Column	Type	Role
`id`	INT PK	Surrogate key for the row
`virtual_address`	VARCHAR(255)	The address being rewritten. Full email (`info@example.com`) **or** a catch-all token (`@example.com`).
`maps`	VARCHAR(255)	Destination address. Single recipient per row in the current schema.
`alias_type`	VARCHAR(20)	Defaults to `forward`. Reserved for future per-alias behavior flags; not surfaced in the UI today.
`send_as`	TINYINT(3)	Reserved for outbound "send-as" support (allow the destination to send mail as the virtual address). Not wired through Postfix yet.
`policy_id`	INT	Reserved for per-alias Amavis policy attachment. Not surfaced today.
`system`	INT	Provenance marker — `1` = seeded by the install/system-addresses flow (postmaster/abuse/root), `2` = admin-created via this page. The system rows are managed by `update_system_email_addresses.cfm` and recreated when the admin email or postmaster changes.

There is no UNIQUE constraint on `virtual_address` because a single inbound address can fan out to multiple destinations — each destination gets its own row. The add handler dedupes on the `(virtual_address, maps)` pair so the same forward isn't inserted twice.

Two address shapes — specific and catch-all

Specific aliases

A regular forward of one address to one destination:

```
info@company.com       →   owner@company.com
sales@company.com      →   sales-team@externalcrm.example
legal@company.com      →   external-counsel@lawfirm.example
```

The local-part is rewritten by Postfix before content filtering. The recipient never sees the original `info@` / `sales@` / `legal@` address unless the destination mail system surfaces the original envelope.

Catch-alls

A single row starting with `@` matches every local-part on the domain that is **not** already a more specific virtual recipient or a mailbox:

```
@company.com           →   admin@company.com
```

With the catch-all row above, mail to `jdoe@company.com`, `random-string@company.com`, and `does-not-exist@company.com` all forward to `admin@company.com`. Specific aliases on the same domain (`info@company.com → owner@company.com`) win over the catch-all because they match the more specific lookup key first.

Catch-alls are useful for sunset domains, migration phases, or small domains where one mailbox owner is willing to receive everything. They are not appropriate for high-volume domains: every spam attempt against a random local-part lands in the catch-all destination.

Catch-all visibility in the user portal

A user whose mailbox is the **destination** of a catch-all (e.g., `admin@company.com` above) has a special branch in the user portal's Quarantined Messages, Total Messages, and Message History queries. `config/hermes/var/www/html/users/2/index.cfm`, `view_message.cfm`, and `view_message_history.cfm` all consult `virtual_recipients` for catch-all entries that explicitly map TO the logged-in user, then widen the query with a `LIKE '%@domain.tld'` clause so the user sees the messages that were swept up by the catch-all. Specific aliases do **not** get this treatment yet — a known parity gap for the rare case where one user owns many specific aliases and wants the same widened visibility.

Fields on the page

Add Virtual Recipients card

Field	Notes
Virtual Address(es)	Newline-delimited textarea. Each line is one full email address or a `@domain.com` catch-all. Lowercased, trimmed, deduped against `virtual_recipients` AND `mailbox_aliases` before insert.
Delivers To	Single destination address for the whole batch. Validated as an email. Autocomplete sourced from `inc/getintrecipients.cfm` (existing relay recipients and mailbox addresses) so you can typeahead-pick a known recipient.

The handler iterates the textarea line-by-line and accumulates per-line results. The success banner reports the count and addresses that landed, and separate error banners surface invalid-format lines, lines whose domain isn't configured as a relay domain, lines whose domain is a mailbox domain (with the "use Email Server > Aliases" pointer), and duplicate lines. **No transaction wraps the batch** — partial success is the expected behavior.

Virtual Recipients table

Standard DataTables surface — searchable, sortable, exportable (copy / CSV / Excel / PDF / print), `stateSave: true` so column order and page size persist across reloads. Columns:

Column	Source
Checkbox	Bulk-select for delete
Recipient	`virtual_recipients.virtual_address`
Delivers To	`virtual_recipients.maps`
Actions	Edit (opens modal)

Edit modal

Inline edit of `virtual_address` and `maps`. Re-runs the same domain validation, catch-all detection, and dedupe check as Add — including the rejection of mailbox-domain rows.

Delete

Checkbox-driven bulk delete from the table card. The handler (`delete_virtual_recipients.cfm`) just runs `DELETE FROM virtual_recipients WHERE id = ?` per selected row — there is no dependency check, because nothing else in the schema points back at a virtual recipient row.

Content filter bypass — by design, loud

The yellow callout on the page exists for a reason. Postfix rewrites the recipient **before** the message reaches Amavis content filtering, but Amavis policy lookups key on the **post-rewrite** recipient. If the destination address is an external Internet address (Gmail, Outlook.com, a personal mailbox, etc.), Amavis applies the default outbound policy to it — which typically means lighter spam/banned-files enforcement than a domain-scoped inbound policy would.

The net effect: mail aliased through a virtual recipient to an external address is generally **less aggressively filtered** than the same mail delivered to a local mailbox or relayed to a known partner domain. This is fine for legitimate forwards, but admins who use virtual recipients to bridge a sunset domain to a personal Gmail should expect Amavis to be permissive about it. Tighten the policy by editing the destination recipient's `recipients` row directly under Relay Recipients if the destination is itself a known Hermes recipient.

Domain-delete dependency

Deleting a relay domain via Domains is blocked when virtual recipients reference it. `deletedomain.cfm` runs:

```
SELECT * FROM virtual_recipients WHERE virtual_address LIKE '%<domain>%'
```

Any match aborts the domain delete with error code 2 and the admin must clear the matching rows from this page before the domain can be removed. The same back-pressure protects against silently stranding a forward when its destination domain disappears.

System-managed rows

A few rows in `virtual_recipients` are created and managed by the **System > Server Setup** flow, not by this page directly:

Pattern	Created by
`postmaster@<every-domain>` → admin email	`inc/update_system_email_addresses.cfm` on every Server Setup save
`root@<every-domain>` → admin email	Same

Pattern	Created by
`abuse@<every-domain>` → admin email	Same

These rows are marked `system = '1'` (the install/system flow) versus admin-created rows which are marked `system = '2'`. Editing or deleting a system-managed row from this page works mechanically, but the row will be recreated on the next Server Setup save. Edit the admin email there if you want a different destination for these reserved local-parts; do not maintain them by hand here.

Failure semantics

What breaks	What happens
Virtual address blank in Add	error 1 banner, no DB write
Delivers To blank or invalid email in Add	error 2/3 banner, no DB write
Edit virtual address fails email or catch-all format	`session.m = 10`, redirect, no DB write
Edit Delivers To blank or invalid	`session.m = 11` / `12`, redirect, no DB write
Domain not in `domains` table	`session.m = 13` on edit; per-line invalid-domain banner on add — line skipped, others continue
Domain is a mailbox domain	Per-line invalid-domain banner with the "use Email Server > Aliases" hint; line skipped
Duplicate `(virtual_address, maps)` pair in `virtual_recipients` or `mailbox_aliases`	Per-line duplicate banner on add; `session.m = 14` on edit
Delete with no rows selected	`session.m = 1` banner, no DB write
MySQL `hermes_db_server` down	Postfix `virtual_alias_maps` lookups fail. By default Postfix defers mail to the affected recipients with a temporary error and retries on the next queue run; legitimate mail is held, not bounced.

Bulk import

The current page supports newline-delimited paste into the Add textarea, which is the practical bulk path: paste hundreds of `alias@domain.com` lines (all forwarding to one destination) at once, click Add, get a per-line outcome report. A separate CSV import is not provided because the table is intentionally one-destination-per-row — fan-out is expressed by adding the same `virtual_address` multiple times with different `maps`, which is easier to do in the textarea than in a CSV.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_virtual_recipients.cfm`	`hermes_commandbox`	Page + Add card + table + modals
`config/hermes/var/www/html/admin/2/inc/addvirtualrecipients.cfm`	`hermes_commandbox`	Add handler with per-line validation
`config/hermes/var/www/html/admin/2/inc/editvirtualrecipient.cfm`	`hermes_commandbox`	Edit handler
`config/hermes/var/www/html/admin/2/inc/delete_virtual_recipients.cfm`	`hermes_commandbox`	Delete handler (per selected id)
`config/hermes/var/www/html/admin/2/inc/getintrecipients.cfm`	`hermes_commandbox`	Autocomplete source for the Delivers To field
`config/hermes/var/www/html/admin/2/inc/update_system_email_addresses.cfm`	`hermes_commandbox`	Manages the `system = '1'` rows (postmaster/root/abuse)
`/etc/postfix/mysql-virtual.cf`	`hermes_postfix_dkim` (volume-mounted)	Postfix MySQL lookup definition for `virtual_alias_maps`
`virtual_recipients`, `mailbox_aliases`, `domains`	`hermes_db_server`	The lookup tables and the domain-type gate

Nothing on this page shells out to Postfix — there is no postmap, no `postfix reload`, no template regeneration. The MySQL lookup is the only integration surface.

Related

- Domains — the relay-topology domain list these aliases attach to. Domain deletes are blocked when virtual recipients still reference the domain.
- Relay Recipients — recipient validation for domains with Recipient Delivery = SPECIFIED. A specific relay recipient and a virtual recipient can coexist for the same address; the relay recipient wins for recipient-list validation, the virtual recipient still rewrites at delivery.
- Email Server > Aliases — the mailbox- topology equivalent. Aliases for domains where Hermes is the destination MTA live there.
- Email Server > Shared Mailboxes — when several users need to read the same incoming mail (not just one user receiving forwards), use a shared mailbox instead of a fan-out virtual recipient.
- Server Setup — manages the `system = '1'` postmaster/root/abuse forwards. Change the admin email there to retarget those reserved local-parts.