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.

Revision #8
Created 2026-05-31 12:52:10 UTC by Dino Edwards
Updated 2026-05-31 14:01:12 UTC by Dino Edwards