Network Block/Allow

Network Block/Allow

Admin path: **Content Checks > Network Block/Allow** (`view_network_block_allow.cfm`, `inc/get_network_block_allow.cfm`, `inc/network_add_entries.cfm`, `inc/network_edit_entry.cfm`, `inc/network_delete_entry.cfm`, `inc/generate_postscreen_access.cfm`).

This page manages the **operator-curated CIDR list** that Postfix's `postscreen` daemon consults at TCP-accept time, before any DNSBL scoring or SMTP handshake. Each entry pairs a single IP or CIDR with an action — `permit` (allow / RBL bypass) or `reject` (block) — and the list is written verbatim to `/etc/postfix/postscreen_access.cidr` on every save. The directive that wires it in lives in `main.cf`:

```
postscreen_access_list = permit_mynetworks, cidr:/etc/postfix/postscreen_access.cidr
```

This is the **third-party-list override** for the perimeter — the place an admin overrides a misfiring RBL hit without disabling the RBL itself, and the place a known-bad source is dropped before it can even attempt SMTP.

Where this list sits in the flow

```
+-------------------------+
|  Inbound TCP connect    |
+-----------+-------------+
            |
            v
+-----------------------------------------------+
|  postscreen :25 (hermes_postfix_dkim)         |
|                                               |
|  1. postscreen_access_list                    |
|     permit_mynetworks                         |
|     cidr:/etc/postfix/postscreen_access.cidr  |
|     -> permit  -> hand off to smtpd, skip all |
```

```
|                    scoring (RBL, greet, etc.)   |
|     -> reject  -> 550, connection closed        |
|     -> no hit  -> fall through                  |
|                                                 |
|  2. postscreen_dnsbl_sites (RBL scoring)        |
|     -> threshold met -> 550                     |
|                                                 |
|  3. pipelining / non-SMTP / bare-newline        |
|     (if enabled on Perimeter Checks)            |
|                                                 |
+-----------+-------------------------------------+
            | passes -> hand to smtpd
            v
+-----------------------------------------------+
|  smtpd :25  (smtpd_*_restrictions)            |
+-----------------------------------------------+
```

The position of `cidr:/etc/postfix/postscreen_access.cidr` matters: because it sits **before** `postscreen_dnsbl_sites` in `postscreen_access_list`, a `permit` entry here causes postscreen to short-circuit and skip every DNSBL lookup for that source. A `reject` entry closes the connection with no further checks at all.

Distinction from Relay Networks

This page is easy to confuse with Relay Networks — both store IPs and CIDRs against Postfix. They are not the same:

Page	Postfix destination	What an entry does
Network Block/Allow (this page)	`cidr:/etc/postfix/postscreen_access.cidr`, consulted by `postscreen_access_list`	`permit` = skip RBL scoring for this IP. `reject` = 550 at TCP accept. **No trust granted** — the source still passes through `smtpd_recipient_restrictions` and content scanning
Relay Networks	`mynetworks` directive in `main.cf`, also Amavis `@inet_acl`	Sets `permit_mynetworks` — sender is **fully trusted**: bypasses RBL, SPF, sender/recipient checks, *and* is allowed to relay outbound to any destination

A wrong entry on Relay Networks creates an open relay. A wrong entry here at worst lets a few extra messages through the perimeter into content scanning, where Amavis + SpamAssassin + ClamAV still apply. The two pages serve different jobs — gate the source vs. trust the source — and the postfix directives they write to are distinct.

When to add a `permit` entry

Scenario	Why allow here instead of Relay Networks
Trusted partner whose IP is listed in an RBL	You want their mail through, but you do **not** want to grant them open relay; the RBL bypass is enough
Shared-hosting sender whose IP also hosts a spammer	Same as above — bypass RBL scoring, let content checks still apply
Microsoft 365 outbound ranges	EOP IPs are already in the shipped seed list as `permit` (151 rows on a fresh install). They are inbound mail sources — they don't need relay trust
Internal monitoring sender whose IP randomly appears in CBL	RBL false positives caught by IP age or shared CGN

When to add a `reject` entry

Scenario	Why reject here instead of waiting for content scoring
Persistent spam source that consistently slips past RBLs	Cheapest possible reject — no DATA accepted, no Amavis cycles
Compromised CIDR block that the operator wants closed off entirely	One CIDR row handles a whole /24, /16, or /8
Manual ban after a Fail2ban-or-equivalent decision is escalated to permanent	A `reject` here outlasts any IP-table or jail-based ban

The two cards on the page

1. Add IP/Network

A textarea for bulk entry — one per line, `IP_or_Network [Note]`. The note is everything after the first space on each line; the IP/CIDR is everything before it. If a line has no space, the entry is its own note.

Validation runs per line:

- Plain IP: must match a strict IPv4 dotted-quad regex (`^(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}…$`).
- CIDR: split on `/`, validate the network half against the same regex, then validate the prefix is an integer in `1..32`.
- Both forms are normalized through `normalizeIP()` — strips leading zeros from each octet (`010.001.001.001/8` becomes `10.1.1.1/8`).
- Duplicates against `postscreen_access.sender` are skipped with a warning; processing continues for the rest of the batch.

The single **Action** radio applies to the whole textarea — every line in one submit gets the same `permit` or `reject`. To mix actions, submit twice.

On submit: rows are `INSERT`-ed into `postscreen_access` with `applied=1, action2='NONE'`, then `generate_postscreen_access.cfm` is included to write the new CIDR file and reload Postfix in the same request. The green "Entries Added" alert summarizes `added`, `skipped`, and any per-line errors.

2. Network Entries (DataTable)

Searchable, sortable, paginated; bulk-delete checkboxes, per-row Edit / Delete buttons.

Column	Source
IP/Network	`postscreen_access.sender`
Note	`postscreen_access.note` (free text from the second half of each Add line)
Action	`postscreen_access.action` rendered as a green "Allow" or red "Block" badge
Actions	Edit (modal), Delete (confirm)

The Edit modal lets the operator change the IP, the action (Allow / Block), or the note in one form post.

Save flow

```
Add / Edit / Delete
    |
    v
INSERT / UPDATE / DELETE on postscreen_access (datasource: hermes)
    |
```

```
    v
cfinclude generate_postscreen_access.cfm
    1. SELECT all enabled rows ORDER BY sender ASC
    2. Write /etc/postfix/postscreen_access.cidr
         <sender>\t<action>\n   per line
    3. docker exec hermes_postfix_dkim /usr/sbin/postfix reload  (30s timeout)
    |
    v
session.m = 1 / 2 / 5 (Added / Deleted / Updated)
On failure -> session.m = 4 ("Configuration Error")
```

The file is written via a direct `cffile action="write"` from the CommandBox container — possible because `/etc/postfix/` is a host-bind-mounted volume shared between `hermes_commandbox` and `hermes_postfix_dkim`. The reload then runs inside the postfix container via `docker exec`. No `postmap` is required for `cidr:` tables — Postfix reads them as text at load time.

The `postscreen_access` table

Column	Type	Role
`id`	`int AUTO_INCREMENT`	Primary key (used as form `delete_id` / `edit_id`)
`sender`	`varchar(255)`	The IP or CIDR string (the column is named `sender` for historical reasons — it is **not** an envelope sender)
`action`	`varchar(255)`	`permit` or `reject`
`action2`	`varchar(255)`	Always `NONE` — legacy two-phase apply column kept for compatibility
`applied`	`int`	`1` once the row is live in the generated `.cidr` file
`note`	`varchar(255)`	Free-text label shown in the table

Engine is MyISAM (matches other operator-curated tables in the schema); collation `latin1_swedish_ci`. The shipped seed includes a large block of Microsoft 365 / Exchange Online Protection ranges as `permit` so EOP-fronted senders are never RBL-scored on a fresh install.

Failure semantics

Failure	Behavior

Empty textarea on Add	`session.m = 30`, redirect, no DB write
Invalid IP or CIDR on a line	Line skipped, `entries_skipped` incremented, error appended; other lines still process
Duplicate against existing `sender`	Same as invalid — skipped with a `Duplicate:` error line
`cffile` cannot write `/etc/postfix/postscreen_access.cidr`	`cfcatch` -> `session.m = 4` ("Configuration Error")
`postfix reload` fails inside the container	Same `session.m = 4` path

If the SQL inserts succeed but the file write or reload fails, the database state has advanced but the live CIDR file lags. The next successful save (or any Edit / Delete) re-renders the file from the current table contents, so the page does not strand split-brain state permanently.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_network_block_allow.cfm`	`hermes_commandbox`	The page
`config/hermes/var/www/html/admin/2/inc/get_network_block_allow.cfm`	`hermes_commandbox`	Loads active rows
`config/hermes/var/www/html/admin/2/inc/network_add_entries.cfm`	`hermes_commandbox`	Per-line validate, INSERT, regen + reload
`config/hermes/var/www/html/admin/2/inc/network_edit_entry.cfm`	`hermes_commandbox`	UPDATE, regen + reload
`config/hermes/var/www/html/admin/2/inc/network_delete_entry.cfm`	`hermes_commandbox`	DELETE single or bulk, regen + reload
`config/hermes/var/www/html/admin/2/inc/generate_postscreen_access.cfm`	`hermes_commandbox`	Rewrites `/etc/postfix/postscreen_access.cidr` and reloads Postfix
`postscreen_access` table	`hermes_db_server` (`hermes` DB)	Source of truth
`/etc/postfix/postscreen_access.cidr` (volume mount)	`hermes_postfix_dkim`	Live CIDR file consumed by postscreen
`hermes_postfix_dkim` container	—	Where `postfix reload` runs

Related

- Perimeter Checks — postscreen toggles and the DNSBL threshold; this page's `permit` / `reject` short-circuits the scoring that page configures
- RBL Configuration — the DNSBL list that a `permit` entry on this page **skips entirely**; the canonical RBL-false-positive override

- Sender/Recipient Rules — envelope-level block/allow applied later in the pipeline (Amavis), not at TCP accept
- Global Sender Rules — envelope-sender block/allow that applies to every recipient on the system
- Relay Networks — the **trust** list (`mynetworks` / `permit_mynetworks`); explicitly different from this page's gate-only semantics
- Relay Recipients — the authenticated path that supersedes IP-based trust for senders that can authenticate
- Intrusion Prevention — the Fail2ban-equivalent layer that maintains short-lived IP bans; this page is where bans get promoted to permanent
- System Logs — postscreen `permit` / `reject` decisions surface in the postfix log under `postscreen[...]`

Revision #48
Created 2026-05-31 12:52:29 UTC by Dino Edwards
Updated 2026-06-20 13:33:16 UTC by Dino Edwards