Global Sender Rules

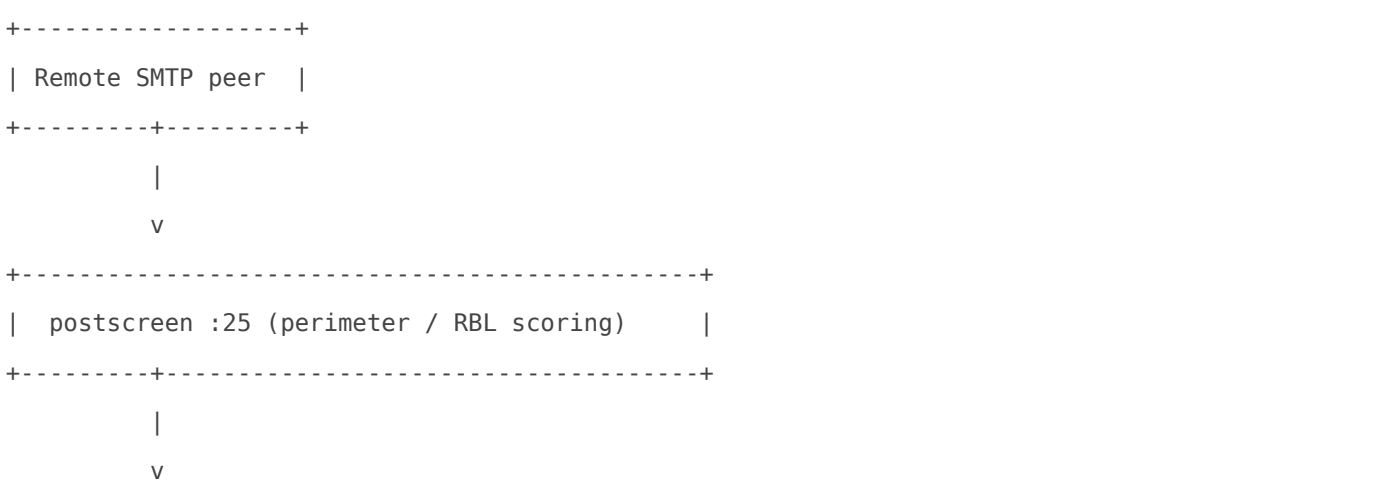
Global Sender Rules

Admin path: **Content Checks > Global Sender Rules** (`view_global_sender_block_allow.cfm`, `inc/get_global_sender_block_allow.cfm`, `inc/global_sender_add_entries.cfm`, `inc/global_sender_edit_entry.cfm`, `inc/global_sender_delete_entry.cfm`, `inc/global_sender_write_and_reload.cfm`).

This page manages **system-wide envelope-sender rules** that apply regardless of recipient. Every entry on this page is a single sender pattern (full address, exact domain, or domain + subdomains) paired with an action — **Block** or **Allow**. The rules are evaluated by Postfix at `MAIL FROM` time, before the message body is read; an Allow match additionally bypasses Amavis content filtering for that sender.

Global Sender Rules are the system-wide counterpart to Sender/Recipient Rules. A Global rule matches **all recipients** in the system; a Sender/Recipient rule requires both a sender and a recipient to match. A Global entry takes precedence over any Sender/Recipient entry for the same sender.

Where Global Sender Rules sit in the flow



```
+---------------------------------------------+
|  smtpd :25                                  |
|   smtpd_sender_restrictions =               |
|     check_sender_access                     |
|       hash:/etc/postfix/amavis_senderbypass |
|                                             |
|   match -> REJECT (block)                   |
|   match -> FILTER amavis:[127.0.0.1]:10030  |
|             (allow -> route past content    |
|              filtering)                     |
|   no match -> fall through to recipient rules |
+---------+-----------------------------------+
          |
          v
+---------------------------------------------+
|  Amavis (white.lst / black.lst consulted    |
|  again at content-filter tier)              |
+---------------------------------------------+
```

The same rule set is written to **two places** on each save: the Postfix `check_sender_access` table (`/etc/postfix/amavis_senderbypass`, `postmap`ed into a Berkeley DB) and the Amavis whitelist/blacklist files (`/etc/amavis/white.lst`, `/etc/amavis/black.lst`). Block entries surface at the Postfix tier — the connection is rejected at `MAIL FROM` and Amavis is never invoked. Allow entries route past Amavis content scoring via the `FILTER` transport hint, and are also written to Amavis's own whitelist as a safety net for any mail path that does reach Amavis (locally-injected, alias-rewritten, etc.).

Pattern formats

The page accepts three pattern formats. The save handler validates each line and auto-prepends `@` to bare domains so the stored row is always in one of the three canonical forms:

Format	Example	Matches
Full email	`user@example.com`	A single envelope sender
Exact domain (`@`)	`@example.com`	Every sender on `example.com` only — subdomains do not match
Domain + subdomains (`.`)	`.example.com`	`example.com` **and** every subdomain (`sub.example.com`, `mail.sub.example.com`, …)

Bare-domain input (`example.com`) is treated as a typo for `@example.com` and rewritten on insert. Email-syntax validation runs on the host portion of every pattern; entries that fail validation are collected into a "Invalid Entries" alert and the rest of the batch is still processed.

The page

A single warning callout, a multi-line Add form, and one DataTable.

Add Sender Entries

A textarea (one entry per line) plus a Block/Allow radio. The form processes the entire batch in one round-trip:

- Each line is trimmed, classified (`@domain`, `.domain`, full email, or bare domain), and validated.
- Valid lines are checked against `amavis_sender_bypass` for an exact-string duplicate; duplicates are collected separately.
- Surviving lines are inserted with `type = block` or `type = allow`. For Allow entries, the row's `transport` column is set to `FILTER amavis:[127.0.0.1]:10030` — this is the Postfix transport hint that bypasses content filtering when a sender match fires.
- If any entries were added, the page calls the write-and-reload include before redirecting.

The redirected page surfaces three separate inline alerts (green success, red invalid, red duplicate) so a mixed batch reports clearly on what happened to every line.

A small inline JS check flips a warning banner under the textarea when the operator types a domain (no `@`) — the consequence of allow-listing or block-listing an entire domain is significant enough to warrant the extra nudge.

Global Sender Entries (DataTable)

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

Column	Source
Sender	`amavis_sender_bypass.sender`
Format	Derived from the leading character — `@` -> Domain badge, `.` -> Domain + Subdomains badge, otherwise Email badge
Action	`amavis_sender_bypass.type` -> Allow (green) or Block (red)
Actions	Edit (modal), Delete (confirm)

Bulk delete posts a comma-separated list of row IDs from the wrapping form. Single Edit and Delete use separate hidden forms so they don't collide with the bulk submit handler.

Save flow

Every Add, Edit, and Delete runs the full regeneration path inline:

```
1. Validate input + INSERT / UPDATE / DELETE on amavis_sender_bypass
2. cfinclude global_sender_write_and_reload.cfm:
     a. SELECT all type='allow' rows (with transport column)
     b. SELECT all type='block' rows
     c. Write /etc/postfix/amavis_senderbypass    (allow rows + transport)
     d. Write /etc/amavis/white.lst               (allow rows, one per line)
     e. Write /etc/amavis/black.lst               (block rows, one per line)
     f. docker exec hermes_postfix_dkim postmap /etc/postfix/amavis_senderbypass
     g. docker exec hermes_postfix_dkim chown root:root <file + .db>
     h. docker exec hermes_postfix_dkim postfix reload
     i. docker exec hermes_mail_filter /etc/init.d/amavis force-reload
3. session.m = 1 / 2 / 5 (Added / Deleted / Updated)
   On failure -> session.m = 4 ("Apply Failed")
```

The Postfix `postmap` step is what makes Block entries actually take effect — `check_sender_access` reads the hashed `.db` file, not the plain-text source. Skipping the postmap (e.g. by editing the source file out-of-band) is a common cause of "I added a block but mail is still getting through".

> **Why both Postfix and Amavis get the list.** The Postfix tier handles the common case — Block rejects before DATA, Allow routes past Amavis via the `FILTER` transport. The Amavis-side `white.lst` / `black.lst` files are a defence in depth: any mail path that **does** reach Amavis (locally-injected mail, mail that was alias-rewritten after the sender check, mail from `permit_mynetworks` sources that skipped sender restrictions) still gets the same allow/block treatment at the content-filter tier. The two layers are kept in sync by the single save flow.

The `amavis_sender_bypass` table

Column	Purpose

`id`	Auto-increment primary key
`sender`	The pattern (`user@example.com`, `@example.com`, or `.example.com`)
`transport`	For Allow rows: `FILTER amavis:[127.0.0.1]:10030`. Empty for Block rows
`action`	Always `NONE` for active rows; reserved for future scheduled-action use
`type`	`allow` or `block`
`applied`	`1` once the row is live; future use for deferred apply

The duplicate check on insert is an exact string match on `sender`, so `@example.com` and `.example.com` are treated as separate (and both can legitimately coexist — they match different sets of addresses).

Failure semantics

Failure	Behavior
Empty textarea	`session.m = 30`, redirect, no DB write
Invalid email/domain on a line	Line skipped, accumulated into the Invalid Entries alert; other valid lines still processed
Exact-string duplicate on a line	Line skipped, accumulated into the Duplicate Entries alert; other valid lines still processed
`cffile` / `postmap` / reload failure	`session.m = 4` ("Apply Failed"); inserted rows remain in the DB and will be re-applied on the next successful save
Postfix container down	Reload fails -> `session.m = 4`; mail flow continues with the previously-loaded Berkeley DB until the container is back

The save is **not** transactional across the DB + file-write + reload steps. If the DB insert succeeds but the postmap or reload fails, the next Add/Edit/Delete will regenerate from the full DB state and reapply.

Operational guidance

- **Allow entries bypass every content filter** — Spam, Virus, Banned File, custom Amavis rules — for the matched sender, for every recipient in the system. The shipped warning callout on the page is not boilerplate; use Allow sparingly and prefer Sender/Recipient Rules for narrower exceptions.

- **Block is cheaper than content filtering.** A Block entry rejects the SMTP transaction at `MAIL FROM`. The body is never read, no spam score is computed, no virus scan runs. For known-phishing sender domains this is the right tier to act at.
- **Domain + subdomain (`.example.com`) carries a wide blast radius** — a Block entry on `.example.com` will reject mail from `support@example.com`, `noreply@news.example.com`, and every other subdomain. The textarea's live warning banner exists for exactly this case.
- **Order of precedence.** Global Sender Rules beat Sender/Recipient Rules. A Block on `@example.com` here will reject mail from that sender even if a per-recipient Allow exists on the Sender/Recipient Rules page for the same sender.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_global_sender_block_allow.cfm`	`hermes_commandbox`	The page
`config/hermes/var/www/html/admin/2/inc/get_global_sender_block_allow.cfm`	`hermes_commandbox`	Loads the active row set for the DataTable
`config/hermes/var/www/html/admin/2/inc/global_sender_add_entries.cfm`	`hermes_commandbox`	Batch validation + INSERT loop
`config/hermes/var/www/html/admin/2/inc/global_sender_edit_entry.cfm`	`hermes_commandbox`	Single-row UPDATE + regen
`config/hermes/var/www/html/admin/2/inc/global_sender_delete_entry.cfm`	`hermes_commandbox`	Single or bulk DELETE + regen
`config/hermes/var/www/html/admin/2/inc/global_sender_write_and_reload.cfm`	`hermes_commandbox`	Writes the three files, runs postmap, reloads Postfix and Amavis
`amavis_sender_bypass` table	`hermes_db_server` (`hermes` DB)	Source of truth
`/etc/postfix/amavis_senderbypass` (+ `.db`)	`hermes_postfix_dkim`	Postfix `check_sender_access` lookup
`/etc/amavis/white.lst`, `/etc/amavis/black.lst`	`hermes_mail_filter`	Amavis sender whitelist / blacklist
`hermes_postfix_dkim` container	—	Runs `postmap` + `postfix reload`
`hermes_mail_filter` container	—	Runs `amavis force-reload`

Related

- Sender/Recipient Rules — per-pair variant; narrower scope, lower precedence
- Perimeter Checks — the upstream `smtpd_*_restrictions` toggles a connection is evaluated against before sender-access lookup

- Network Block/Allow — the IP-level `postscreen_access.cidr` table consulted **before** any sender evaluation; an entry there can short-circuit a peer regardless of envelope sender
- RBL Configuration — third-party DNSBL scoring at the postscreen tier; runs before sender access lookup
- BCC Maps — sibling envelope-level rule table; the other half of the envelope-rule pair
- Anti-Spam Settings — the content-filter tier that Allow entries route around
- System Logs — `mail.log` is where block rejections and Amavis bypass decisions surface for audit
- Mail Queue — visible flow-of-mail diagnostics if a rule change has an unexpected effect

---

Revision #14
Created 2026-05-31 12:52:26 UTC by Dino Edwards
Updated 2026-06-13 12:30:20 UTC by Dino Edwards