Sender/Recipient Rules

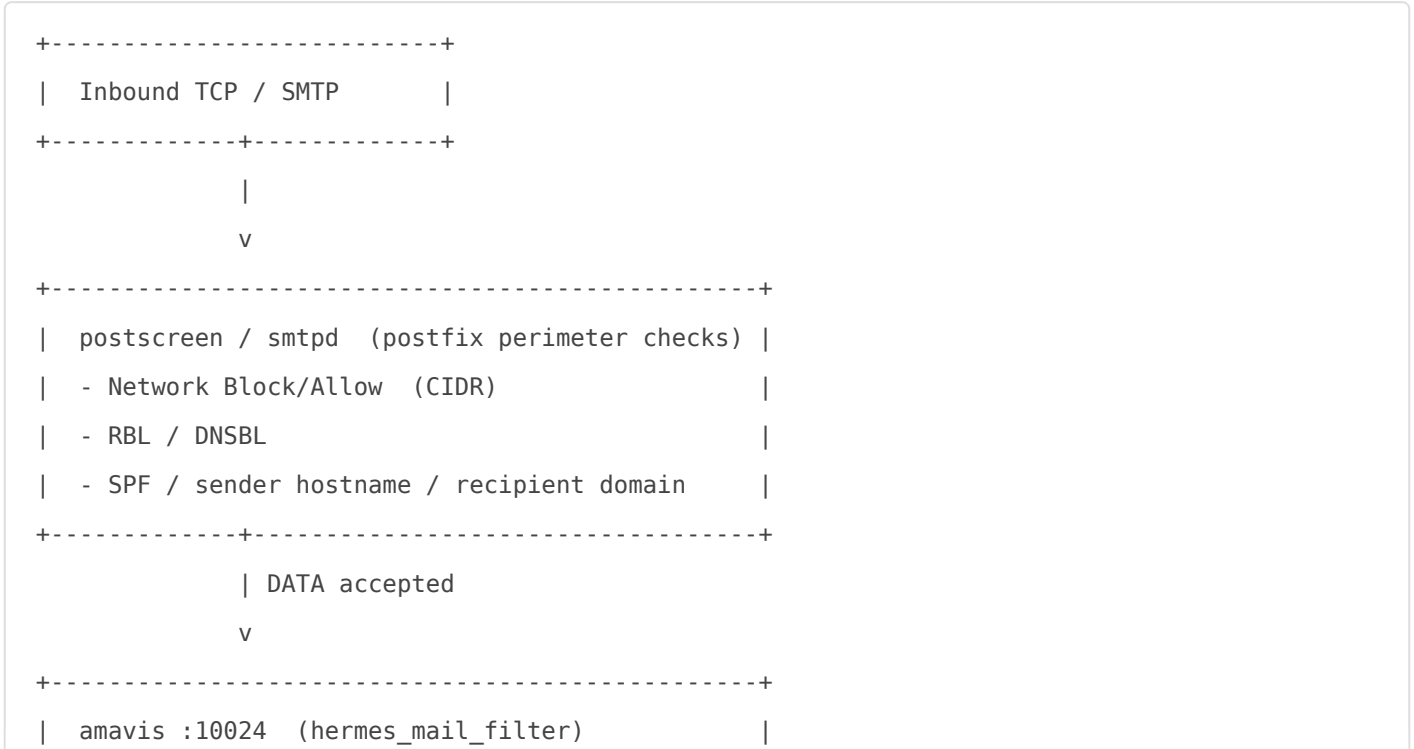
Sender/Recipient Rules

Admin path: **Content Checks > Sender/Recipient Rules** (`view_sender_recipient_block_allow.cfm`, `inc/get_sender_recipient_block_allow.cfm`, `inc/sender_add_entry.cfm`, `inc/sender_edit_entry.cfm`, `inc/sender_delete_entry.cfm`).

This page manages **per-recipient envelope-sender filters** — pairs of (sender, recipient) that Amavis honors when it scores an inbound message. Each row says "when this sender writes to this recipient, apply this rule" — `ALLOW` (skip spam scoring) or `BLOCK` (quarantine / reject). The rules live in Amavis's native `wblist` table and are read live on every message, so saves take effect on the next inbound delivery with no service reload.

This is the **envelope-level** half of the inbound-control story. Pairs with Network Block/Allow, which is the IP-level half evaluated much earlier in the SMTP pipeline.

Where this list sits in the flow



```
|                                                |
|  Per-recipient lookup:                         |
|  $sql_select_white_black_list                  |
|    SELECT wb FROM wblist, mailaddr, recipients |
|    WHERE recipients.id = wblist.rid            |
|      AND mailaddr.id   = wblist.sid            |
|      AND mailaddr.email IN (%k)                |
|                                                |
|  -> wb = 'W'  -> SKIP spam scoring             |
|                 (viruses + banned files +      |
|                  bad headers STILL apply)      |
|  -> wb = 'B'  -> mark as spam / quarantine     |
|  -> no row    -> normal scoring path           |
+------------------------------------------------+
```

The lookup is keyed on the envelope-sender address (`mailaddr.email`) **after** Amavis has already accepted the message from Postfix and started its scoring pass. That is the central operational fact: this page does not stop mail at SMTP time — it only changes how Amavis treats it once received.

Distinction from sibling pages

Three pages share overlapping vocabulary; they apply at three different points in the pipeline.

Page	Layer	Match key	Effect
Network Block/Allow	`postscreen` (TCP / pre-SMTP)	Source IP / CIDR	550 or RBL bypass; **no content-layer effect**
Global Sender Rules	Amavis (per-message)	Envelope sender only	Allow / block from this sender to **every recipient** on the system
Sender/Recipient Rules (this page)	Amavis (per-message)	Envelope sender **and** specific recipient	Allow / block from this sender to **one recipient** (or one recipient-domain)

Order of precedence within Amavis: a Global Sender Rules entry takes precedence over a per-recipient entry on this page — the in-page callout on Global Sender Rules states this explicitly. Use this page when the policy needs to be scoped to a specific person or mailbox; use Global Sender Rules only when the policy must apply to everyone.

`ALLOW` does not bypass virus, banned files, or bad headers

The in-page callout makes this explicit:

> Allow entries only bypass Spam checks. Emails with Viruses, Banned Files, and Bad Headers will still be blocked.

That is a property of Amavis itself — `wb='W'` in the `wblist` table short-circuits the SpamAssassin score path but does not exempt the message from virus scanning (ClamAV), banned-file extension rules (`@banned_filename_re`), or RFC-violation header checks. The operational consequence is that an `ALLOW` here is much narrower than the `permit` action on Network Block/Allow — there, RBL is skipped and the message enters Amavis on the same path as any other; here, only the spam-score gate is removed.

Sender match formats

The sender field accepts three formats, all distinguished by the position of `@`:

What you type	Stored as	Matches
`user@example.com`	`user@example.com`	A single full envelope-sender address
`example.com`	`@example.com`	Any envelope sender on `example.com` (the bare domain — exact match, **no** subdomains)
`.example.com`	`@.example.com`	`example.com` **and** any subdomain (`mail.example.com`, `sub.sub.example.com`, …)

The page accepts the bare domain form for convenience and rewrites it with the leading `@` before the `mailaddr` lookup. The leading-dot form is preserved as-is and stored as `@.example.com` — Amavis itself interprets the dot as the wildcard.

Recipient match formats

The recipient field is constrained to recipients already known to the system. It autocompletes from the `recipients` table via a `<datalist>` populated on page render. Two forms work:

What you type	What the lookup does	Effect
`user@example.com`	Matches a single row in `recipients`	One `wblist` row inserted (one rid)
`@example.com`	Matches a domain-level row in `recipients` (where `domain='1'`); the handler then enumerates every individual recipient under that domain	One `wblist` row per recipient in the domain — the rule fans out

If the typed recipient does not exist anywhere in `recipients`, the save fails with `session.m = 34` ("specified recipient was not found in the system"). The page does not create recipients on the fly — add the recipient on Relay Recipients or as a Mailbox first.

Same-domain sender / recipient is rejected

A guard rejects entries where the sender domain and recipient domain are the same (`session.m = 35`). Inbound mail from `user@example.com` to `boss@example.com` is normally outbound or internal, not the inbound-filtering case this page is designed for, and an `ALLOW` across that boundary would be a routine misconfiguration.

The two cards on the page

1. Add Sender/Recipient Entry

Four inputs across one form: **Sender Email or Domain**, **Recipient** (autocomplete from `recipients`), **Action** (BLOCK / ALLOW radios), and submit. Validation order on submit:

1. Sender non-empty (`session.m = 30` on fail).
2. Recipient non-empty (`session.m = 31`).
3. Action is BLOCK or ALLOW (`session.m = 32`).
4. Sender is a syntactically valid email *or* a syntactically valid domain — checked by `IsValid("email", ...)` against a stub address (`session.m = 33`).
5. Recipient resolves to a row in `recipients` (`session.m = 34`).
6. Sender domain != recipient domain (`session.m = 35`).
7. Sender+recipient pair is not already in `wblist` (`session.m = 36`, "already exists or already staged for addition").

On success, the handler:

1. Resolves or creates the `mailaddr` row for the sender (one row per distinct address — `mailaddr` is shared with the rest of the Amavis stack).
2. Inserts the `wblist` row(s):
   - Specific recipient: one row.
   - Domain-wide recipient: one row per individual recipient in that domain (the rule fans out at insert time, not at lookup time).
3. Sets `wb = 'W'` (ALLOW) or `wb = 'B'` (BLOCK).

There is no Postfix or Amavis reload — Amavis reads `wblist` live on every message via its SQL backend.

2. Sender/Recipient Entries (DataTable)

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

Column	Source
Sender	`mailaddr.email` joined via `wblist.sid`
Recipient	`recipients.recipient` joined via `wblist.rid`
Type	`wblist.wb` rendered as green "Allow" or red "Block" badge
Actions	Edit (modal), Delete (confirm)

Each row's checkbox value is a composite `rid:sid` (the `wblist` table's natural primary key — no surrogate `id` column). The bulk delete handler splits each entry on `:` and deletes the matching `wblist` row directly.

The **Edit** modal keeps the recipient read-only (with the inline note "Recipient cannot be changed. Delete and re-add if needed") — changing the recipient would change `rid`, which is the row's identity. The sender and the BLOCK/ALLOW type are editable; the save handler deletes the original row and inserts a new one, using the sender email strings to find the old row (no integer ID is needed from the form).

Save flow

```
Add / Edit / Delete
    |
    v
INSERT / UPDATE / DELETE on wblist (and mailaddr for new senders)
  All queries datasource = "hermes"
```

```
    |
    v
(Delete only) Garbage-collect orphaned mailaddr rows:
  DELETE FROM mailaddr WHERE id NOT IN (SELECT DISTINCT sid FROM wblist)
    |
    v
session.m = 1 / 2 / 5  (Added / Deleted / Updated)
On validation failure -> session.m = 30..36
```

No file write, no `postmap`, no service reload. Amavis picks the new rules up on the next message.

Tables involved

Table	Role	Engine
`wblist`	(`rid`, `sid`, `wb`) composite-key per-pair rule	MyISAM, utf8mb3
`mailaddr`	Distinct envelope-sender addresses; unique key on `email`	MyISAM, utf8mb3
`recipients`	Resolved at lookup time to find `rid`; populated from the rest of the system (Mailboxes, Relay Recipients, domain-level entries)	MyISAM

`wblist` and `mailaddr` are Amavis's own native tables — Hermes pre-creates them in `hermes_install.sql` because Amavis would otherwise lazily create them on its first SQL-backend write, after the CFML pages that reference them have already started to render.

The composite key `(rid, sid)` is enforced at the database layer, so the page's duplicate guard (`session.m = 36`) and the database itself will both refuse a true duplicate. `mailaddr` carries a `UNIQUE KEY` on `email`, so concurrent sender adds cannot create duplicate rows even mid-race.

Relationship to user-portal sender filters

End users in the recipients table see and manage their **own** subset of `wblist` rules from the user portal (`/users/2/`) — the "Allow this sender" and "Block this sender" buttons on a quarantined message, plus the explicit Sender Filters page, both write rows into the same `wblist` table with the user's own recipient `id` as `rid`.

This admin page sees those user-trained rules in the same table — they are not flagged separately in the UI. Operators editing or deleting from this page can affect user-trained rules; that is by design (this page is the operator's view of the entire `wblist` table).

Failure semantics

Failure	`session.m`	Behavior
Empty sender	`30`	Redirect, no DB write
Empty recipient	`31`	Redirect, no DB write
Invalid action (neither BLOCK nor ALLOW)	`32`	Redirect, no DB write
Sender not a valid email or domain	`33`	Redirect, no DB write
Recipient not found in `recipients`	`34`	Redirect, no DB write
Same sender and recipient domain	`35`	Redirect, no DB write
Pair already in `wblist`	`36`	Redirect, no DB write

There is no equivalent of `session.m = 4` ("Configuration Error") on this page — there is no Postfix / Amavis regen step that could fail. A SQL error would surface as an uncaught `cfcatch` and the standard 500-error page, not a friendly alert.

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_sender_recipient_block_allow.cfm`	`hermes_commandbox`	The page
`config/hermes/var/www/html/admin/2/inc/get_sender_recipient_block_allow.cfm`	`hermes_commandbox`	Joins `wblist` + `mailaddr` + `recipients` for the table
`config/hermes/var/www/html/admin/2/inc/sender_add_entry.cfm`	`hermes_commandbox`	Validate, resolve/insert `mailaddr`, INSERT `wblist` (fans out for domain recipients)
`config/hermes/var/www/html/admin/2/inc/sender_edit_entry.cfm`	`hermes_commandbox`	DELETE original row by email-join, INSERT new row, garbage-collect orphan `mailaddr`
`config/hermes/var/www/html/admin/2/inc/sender_delete_entry.cfm`	`hermes_commandbox`	DELETE single or bulk by `rid+sid`, garbage-collect orphan `mailaddr`
`wblist`, `mailaddr`, `recipients` tables	`hermes_db_server` (`hermes` DB)	Source of truth

Path	Owner	Role
`hermes_mail_filter` container (Amavis)	—	Consumes the rules live via `$sql_select_white_black_list` on every inbound message

Related

- Network Block/Allow — IP-level (`postscreen`) sibling; runs before any SMTP handshake, much earlier than this page in the pipeline
- Global Sender Rules — envelope-sender block/allow with no recipient scope; takes precedence over this page's per-pair rules
- Anti-Spam Settings — the scoring path that an `ALLOW` here short-circuits
- Anti-Virus Settings — runs even when this page sets `ALLOW`; the "bypass spam only" caveat exists because virus scanning is non-bypassable
- BCC Maps — sibling Content Checks page; takes the same per-recipient routing approach for a different purpose (silent copies vs. block/allow)
- Perimeter Checks — the SMTP-time checks that run **before** Amavis ever sees the message
- Relay Recipients — the recipient list this page's autocomplete draws from; an entry here presupposes a row there (or in Mailboxes)
- Message History — where the effect of ALLOW / BLOCK decisions on this page shows up after delivery
- System Logs — Amavis logs each `wblist` lookup result; the wb value (`W` / `B`) is visible in the per-message scoring trace

---

Revision #48
Created 2026-05-31 12:52:32 UTC by Dino Edwards
Updated 2026-06-20 13:33:18 UTC by Dino Edwards