SPF Settings

SPF Settings

Admin path: **Content Checks > SPF Settings** (`view_spf_settings.cfm`, `inc/get_spf_settings.cfm`, `inc/spf_save_settings.cfm`, `inc/spf_generate_config_file.cfm`, `inc/spf_add_whitelist.cfm`, `inc/spf_edit_whitelist.cfm`, `inc/spf_delete_whitelist.cfm`, `inc/generate_postfix_configuration.cfm`).

This page controls **inbound** SPF policy enforcement. SPF (RFC 7208) lets the owner of a domain publish, in DNS, the list of IP addresses authorized to send mail using that domain in the envelope `MAIL FROM` (and optionally the SMTP `HELO`). When Postfix accepts a connection, Hermes consults the published record for the connecting client and decides whether to accept, defer, or reject the message based on the result.

Hermes is responsible only for the **verification** side. Publishing your own organization's SPF record (the `v=spf1 ...` TXT record at your sending domain) is a one-time DNS operation done at your authoritative DNS host — it is not managed from this page.

Where SPF sits in the flow

```
+----------------------+
| Remote SMTP peer     |
+----------+-----------+
           |
           v
+----------+--------------------------------+
| smtpd :25 (hermes_postfix_dkim)           |
|   smtpd_recipient_restrictions = ...,     |
|     check_policy_service unix:private/    |
|       policy-spf                          |
|       |                                   |
|       v                                   |
|   Postfix spawns policyd-spf (python)     |
```

```
|   from master.cf "policy-spf unix" entry   |
|   - reads /etc/postfix-policyd-spf-python/ |
|       policyd-spf.conf                     |
|   - queries DNS for the sender's SPF TXT   |
|   - returns Pass / Fail / Softfail /       |
|     Neutral / None / TempError / PermError |
|   - returns Postfix action verb            |
|     (DUNNO / REJECT / DEFER_IF_REJECT)     |
+----------+--------------------------------+
           |
           v
+----------+--------------------------------+
| OpenDKIM milter :8891 (DKIM verify)        |
| OpenDMARC milter :54321 (DMARC eval)       |
+----------+--------------------------------+
           |
           v
   Amavis / SpamAssassin / ClamAV
```

The policy daemon is a **Postfix policy delegate** — a separate process that Postfix spawns from `master.cf`:

```
policy-spf  unix  -  n  n  -  -  spawn
            user=nobody argv=/usr/bin/policyd-spf
```

`smtpd_recipient_restrictions` invokes it via `check_policy_service unix:private/policy-spf`. The daemon's configuration file at `/etc/postfix-policyd-spf-python/policyd-spf.conf` is what this admin page writes; the entire file is regenerated on every save from the template at `/opt/hermes/templates/policyd-spf.conf.HERMES`.

SPF result classes and their typical meaning

Result	Meaning	Default Hermes behavior
`Pass`	Connecting IP is in the published `v=spf1` record	Accept
`Fail`	Sender has published `-all`; this IP is explicitly disallowed	Reject

Result	Meaning	Default Hermes behavior
`SoftFail`	Sender has published `~all`; this IP is not authorized but the owner is in monitoring mode	**Reject** (Hermes recommended) — see Operational consequence below
`Neutral`	Sender published `?all`; owner expresses no opinion	Accept (treated as `None`)
`None`	No SPF record exists for the sender	Accept
`TempError`	DNS timeout / SERVFAIL during the lookup	Accept (treat as no record) — operator can switch to defer
`PermError`	SPF record is malformed or exceeds the 10-DNS-lookup limit	Accept (treat as no record) — operator can switch to reject

SPF is **checked twice per message** by the daemon: once against the SMTP `HELO` identity (before `MAIL FROM`), and once against the envelope sender domain after `MAIL FROM`. Each check has its own rejection policy on this page.

The two cards on the page

1. SPF Settings (master toggle + policy daemon controls)

The master **SPF Enabled** dropdown flips a single child row in the `parameters` table — the row whose `parameter` value is `check_policy_service unix:private/policy-spf` under the `smtpd_recipient_restrictions` parent. When SPF is disabled the page also forces DMARC off (DMARC requires both an SPF and a DKIM result; without SPF the DMARC milter has nothing to align against). The in-page callout warns about this dependency.

When SPF is enabled, the policy section exposes six controls, each written to a `parameters2` row in the `dkim`/`spf` module rows:

Control	`policyd-spf.conf` directive	Effect
Logging Level	`debugLevel`	`0`–`4` verbosity; `-1` disables logging. Higher levels log every DNS lookup and the full SMTP envelope data — useful for diagnosing federal / M365 GOV / Proofpoint Government chain issues

Control	`policyd-spf.conf` directive	Effect
Test Mode	`TestOnly`	`1` adds the SPF result to message headers but **never** rejects, regardless of the rejection policies below. Use to evaluate impact before enforcing
HELO Check Rejection Policy	`HELO_reject`	What to do with the SPF result for the SMTP `HELO`/`EHLO` identity. Options: `Fail`, `SPF_Not_Pass` (Reject All), `Softfail` (Recommended), `Null` (reject HELO of null-sender bounces only), `False` (header only), `No_Check`
Mail From Check Rejection Policy	`Mail_From_reject`	Same option set, but applied to the envelope `MAIL FROM` domain
Permanent Error Policy	`PermError_reject`	`True` rejects when the published SPF record is broken; `False` (recommended) treats it as no record
Temporary Error Policy	`TempError_Defer`	`True` issues a 4xx defer on DNS timeout; `False` (recommended) accepts and continues

> **Operational consequence — single point of SPF truth.** The Hermes baseline disables SpamAssassin's redundant SPF re-check. SA's in-process SPF scoring runs after Amavis has reinjected the message over a local hop, so SA sees an IP path that does not include the original sender — on government/M365 GOV/Proofpoint Government mail the wrong IP gets scored, producing false-positive `SPF_SOFTFAIL` hits. The policy daemon on this page is the **single authoritative SPF verifier**; it sees the real connecting client IP. To preserve the spam-coverage SA's `SPF_SOFTFAIL` rule provided, set both HELO and Mail From Check Rejection Policy to **Reject SoftFail**. This is the in-page recommendation and the shipped baseline.

2. SPF Whitelist Entries

Per-row bypass list written to four `Whitelist` directives in `policyd-spf.conf`:

Entry type	`policyd-spf.conf` directive	What it matches	Typical use
IP / Network Address	`Whitelist`	The connecting client IP (single address or CIDR)	Trusted secondary MX, known forwarders, partner relays

Entry type	`policyd-spf.conf` directive	What it matches	Typical use
HELO/EHLO Host Name	`HELO_Whitelist`	The hostname announced in `HELO` / `EHLO`. Daemon DNS-checks the connecting IP against an A/AAAA for that name to prevent forgery	Mailing-list providers that consistently HELO with their own domain
Domain Name	`Domain_Whitelist`	The envelope `MAIL FROM` domain	Senders with broken `~all` records whose mail you still need to receive
PTR Domain	`Domain_Whitelist_PTR`	The reverse-DNS (PTR) domain of the connecting IP	Hosts whose forward DNS is unstable but whose reverse DNS is well-controlled

Entries are stored in the `spf_bypass` table (`entry`, `entry_type`, `entry_note`). The save handler joins all enabled rows of each type with commas and substitutes them into the template at `IP-NETWORK-WHITELIST`, `HELO-WHITELIST`, `DOMAIN-WHITELIST`, `PTR-WHITELIST` placeholders.

A whitelist hit completely **skips** SPF evaluation for that connection — the daemon returns `Pass` without consulting DNS. Use IP-based whitelisting when possible; HELO / Domain / PTR entries incur extra DNS lookups per message.

The DataTable supports add (textarea — one entry per line, validated and deduplicated), inline edit modal, single delete, and bulk delete via checkbox selection.

What this page does NOT control

- **Per-sender allow/block.** Address-level rules live on Sender/Recipient Rules and apply later in the pipeline.
- **The SPF record for your own sending domain.** That is a DNS TXT record you publish at your authoritative DNS host. A correct outbound SPF for a Hermes-served sending domain typically looks like `v=spf1 mx ip4:<hermes-egress-ip> include:<isp-relay> ~all` — see Domains (Email Relay) and Domains (Email Server) for the egress IP your record needs to authorize.
- **Network-level allow.** Trusted SMTP source ranges (`mynetworks`) short-circuit before any policy check via Relay Networks.

Save flow

```
1. Validate form fields exist when SPF is being enabled
   - Missing fields -> session.m = 20, redirect, no DB write
2. UPDATE parameters child row for SPF on/off
3. UPDATE parameters2 rows for the six policy daemon directives
4. cfinclude spf_generate_config_file.cfm
     a. Read /opt/hermes/templates/policyd-spf.conf.HERMES
     b. REReplace placeholders (DEBUG-LEVEL, TEST-ONLY, HELO-REJECT,
        MAIL-FROM-REJECT, PERMERROR-REJECT, TEMPERROR-REJECT)
     c. SELECT all enabled spf_bypass rows by entry_type, comma-join,
        substitute *-WHITELIST placeholders
     d. Backup current /etc/postfix-policyd-spf-python/policyd-spf.conf
        as policyd-spf.conf.HERMES
     e. Move generated tmp file into place
5. cfinclude generate_postfix_configuration.cfm
     - Regenerates main.cf so smtpd_recipient_restrictions reflects
       SPF on/off
     - Reloads Postfix inside hermes_postfix_dkim
6. If SPF was DISABLED: also disable the OpenDMARC milter rows,
   clear FailureReports, deactivate the DMARC report Ofelia job,
   regenerate opendmarc.conf, restart OpenDMARC
7. session.m = 9 -> green "SPF settings saved" alert on redirect
```

Files and containers touched

Path	Owner	Role
`config/hermes/var/www/html/admin/2/view_spf_settings.cfm`	`hermes_commandbox`	The page
`config/hermes/var/www/html/admin/2/inc/get_spf_settings.cfm`	`hermes_commandbox`	Loads current parameters / parameters2 / spf_bypass values
`config/hermes/var/www/html/admin/2/inc/spf_save_settings.cfm`	`hermes_commandbox`	Validates form, updates rows, calls config + Postfix regen; disables DMARC if SPF off
`config/hermes/var/www/html/admin/2/inc/spf_generate_config_file.cfm`	`hermes_commandbox`	Renders `policyd-spf.conf` from the template + DB
`config/hermes/opt/hermes/templates/policyd-spf.conf.HERMES`	`hermes_commandbox` (read) → `hermes_postfix_dkim` (live `/etc/postfix-policyd-spf-python/policyd-spf.conf`)	Canonical template with `DEBUG-LEVEL`, `TEST-ONLY`, etc. placeholders

Path	Owner	Role
`parameters` table (`check_policy_service unix:private/policy-spf` row)	`hermes_db_server` (`hermes` DB)	SPF on/off
`parameters2` table (rows where `module='spf'`)	`hermes_db_server` (`hermes` DB)	The six daemon settings
`spf_bypass` table	`hermes_db_server` (`hermes` DB)	Whitelist entries
`hermes_postfix_dkim` container	—	Runs `smtpd` , spawns `policyd-spf` , hosts the live `policyd-spf.conf`
`hermes_unbound` container	—	Resolves every SPF DNS query the daemon makes

Failure semantics

Failure	Behavior
Missing form fields when enabling SPF	`session.m = 20` , redirect, no DB write
`spf_generate_config_file.cfm` throws (template missing, write fails, etc.)	Surfaces as a `cfcatch` from the inline include — the save aborts
Empty whitelist entry on Add	`session.m = 13` , redirect, no DB write
Whitelist entry fails IP / hostname syntax check	`session.m = 17` , redirect, no DB write
Duplicate whitelist entry	`session.m = 14` , redirect, no DB write
`postfix reload` fails inside the container	Standard `generate_postfix_configuration.cfm` failure path

Related

- DKIM Settings — the second authentication service whose result is consumed by DMARC; paired conceptually with SPF as a "DNS-based outbound sender authentication" mechanism
- DMARC Settings — the policy layer that consumes SPF and DKIM results; disabling SPF here automatically disables DMARC
- ARC Settings — chain-of-custody for authentication results across forwarders; participates only after SPF / DKIM / DMARC have produced their verdicts
- Trusted ARC Sealers (M365) — for M365 customers whose downstream verifiers escalate when SPF fails on forwarded mail
- Perimeter Checks — the rest of the `smtpd_recipient_restrictions` chain; the SPF / DKIM / DMARC status badges on its fourth card link back to the dedicated pages

- Sender/Recipient Rules — per-address bypass applied after the SPF verdict
- DNS Resolver — every SPF lookup flows through `hermes_unbound`; resolver mode (recursive vs. forwarding through a public provider) directly affects SPF reliability and the 10-DNS-lookup limit timing
- Domains (Email Relay), Domains (Email Server) — where the egress IP that your authoritative SPF record needs to authorize is documented

---

Revision #64
Created 2026-05-31 12:52:33 UTC by Dino Edwards
Updated 2026-08-22 11:48:48 UTC by Dino Edwards