Rsyslog Load-Balancer (RHEL 8/9) — FIPS-Compatible (GnuTLS) — Final Procedure

Author: Prepared for ESA by Gaetan De Dobbeleer (Logpoint)

Goal Implement a robust rsyslog relay on **RHEL 8 and RHEL 9** that:

- Accepts **UDP 514**, **TCP 514**, and **TCP 6514 (TLS)** from sources
- **Load-balances** outgoing events to multiple Logpoint collectors/backends over **TCP/TLS (6514)** using the native **`omfwd` target pool** (round-robin)
- Uses the **GnuTLS** driver (`gtls`) and is **compatible with system FIPS** mode
- Can be installed **manually (RPMs)** or **via the Adiscon repo**
- Provides safe **buffering** (disk-assisted queues) and clean **auto-recovery**
- Enables **observability** with `impstats`
- Provides two **mutually exclusive egress options** with **identical parameters**: (1) **GnuTLS (default)** and (2) **CLEAR (optional for debugging)**
- **Optional filter** capability: **regex blacklist** drop-on-match before egress

> ⚠ Enable **only one** egress action at a time (TLS by default). The CLEAR option is for troubleshooting in controlled environments.

---

1) Version & design decisions

- Use **rsyslog v8.2502+ (recommended)**; **≥ 8.2408 required** for native `target=["ip1","ip2",...]` load-balancing in `omfwd`.
- Prefer **Adiscon builds** for up-to-date v8 on RHEL.
- Transport/ports (by Logpoint policy): **UDP 514**, **TCP 514 (clear)**, **TCP 6514 (TLS)**.
- Listener on the relay: UDP 514, TCP 514, and TLS 6514.
- Egress: **TLS 6514** with **round-robin** to multiple backends (default). Optional **CLEAR 514/tcp** egress with the **same LB/queue** settings.
- **GnuTLS path**: install **`rsyslog-gnutls`** and configure `StreamDriver="gtls"` and `defaultNetstreamDriver="gtls"`.
- **FIPS compatibility**: enable **system FIPS mode** on RHEL (no custom cipher overrides in rsyslog; inherit system crypto policies).

---

2) Prerequisites

- RHEL 8 or RHEL 9 (x86_64), root access.

- Network reachability to Logpoint backends on 6514/tcp (and 514/tcp if used).

- **TLS material** on the relay:

  - `/etc/rsyslog.d/tls/ca.crt`
  - `/etc/rsyslog.d/tls/server.crt`
  - `/etc/rsyslog.d/tls/server.key` (0600)

- Placeholders to replace with production targets: `<BACKEND_1>`, `<BACKEND_2>` (add more as needed).

Quick OS check

```
source /etc/os-release 2>/dev/null; echo "$NAME $VERSION_ID"; rpm -E %rhel; uname -m
```

---

3) Install options (choose **one**)

3A. Repo-based installation (Adiscon)

```
cd /etc/yum.repos.d/
sudo curl -O https://rpms.adiscon.com/v8-stable/rsyslog-rhel.repo
# Optional: daily-stable for faster fixes
sudo curl -O https://rpms.adiscon.com/v8-stable-daily/rsyslog-daily-rhel.repo

# Security hygiene
echo "Ensuring gpgcheck=1"; sudo sed -i 's/^gpgcheck=.*/gpgcheck=1/' /etc/yum.repos.d/rsyslog-*.repo

sudo dnf clean all && sudo dnf makecache
sudo dnf install -y rsyslog rsyslog-gnutls   # GnuTLS backend
rsyslogd -v                                  # verify v8.2502+ ideally
```

3B. Manual (RPM) installation — offline-friendly

> Use this when repos are not allowed. Pick the correct directory for your EL version.

- **EL8**: `https://rpms.adiscon.com/v8-stable/epel-8/x86_64/RPMS/`
- **EL9**: `https://rpms.adiscon.com/v8-stable/epel-9/x86_64/RPMS/`

Required packages

- Base daemon: `rsyslog-<version>-1.el8|el9.x86_64.rpm`
- **TLS (GnuTLS)**: `rsyslog-gnutls-<version>-1.el8|el9.x86_64.rpm`

Verify & install

```
sudo rpmkeys --import https://rpms.adiscon.com/RPM-GPG-KEY-Adiscon
rpm -K rsyslog-*.rpm            # should report a good signature
sudo dnf install ./rsyslog-*.rpm

# Offline bundle option on a connected box, then copy to the target:
sudo dnf download --resolve rsyslog rsyslog-gnutls
sudo dnf install ./*.rpm
```

Dependencies are normally satisfied by `dnf` from enabled RHEL repos (`libfastjson`, `libestr`, `openssl-libs`, etc.).

---

5) Configuration layout

Files

- `/etc/rsyslog.conf` — core, inputs, TLS defaults, `impstats`
- `/etc/rsyslog.d/05-pre-filter.conf` — **optional** regex blacklist (drop-on-match)
- `/etc/rsyslog.d/blacklist.d/` — directory of regex rules (one file per rule)
- `/etc/rsyslog.d/10-esa-lb.conf` — outbound to Logpoint via **LB + buffering + failover** (TLS default / CLEAR optional)
- `/etc/rsyslog.d/tls/` — TLS material

5.0 **Optional — Regex Filter (Blacklist) explained**

This option **drops** messages that match **any** regex, and forwards the rest. It can target the **entire syslog packet** via `$rawmsg` (header+body) or specific header fields (`$hostname`, `$programname`, `$fromhost-ip`, etc.).

Create `/etc/rsyslog.d/05-pre-filter.conf`:

```
# Pre-filter: drop-on-match, else forward
ruleset(name="pre_filter") {
  # Load all blacklist rules (each rule may 'stop' to drop)
  $IncludeConfig /etc/rsyslog.d/blacklist.d/*.conf

  # If no rule matched -> forward to egress
  call to_logpoint
}
```

Examples in `/etc/rsyslog.d/blacklist.d/` (one regex per file):

- `10-drop-link-flaps.conf`

```
if re_match($rawmsg, "(?i)link (is )?(up|down)") then { stop }
```

- `20-drop-noisy-host.conf`

```
if ($hostname startswith "noisy-fw") then { stop }
```

- `30-drop-by-source-ip.conf`

```
if ($fromhost-ip startswith "10.10.20.") then { stop }
```

- `40-drop-debug-verbosity.conf`

```
if ($syslogseverity >= 6 and re_match($msg, "(?i)debug|trace")) then { stop }
```

> **Tip:** Keep one pattern per file, prefix filenames with numbers for ordering. You can temporarily log dropped events to `/var/log/rsyslog_dropped_by_blacklist.log` if you need an audit trail.

5.1 `/etc/rsyslog.conf` (excerpt, **inputs** & TLS defaults)

```
# Work directory for persistent queues (disk-assisted)
global(workDirectory="/var/spool/rsyslog")

# Inputs
module(load="imudp")
module(load="imtcp")
module(load="impstats" interval="60" format="cee")  # stats/health

# TLS defaults (used by listener on 6514 and egress)
global(
  defaultNetstreamDriver="gtls"
  defaultNetstreamDriverCAFile="/etc/rsyslog.d/tls/ca.crt"
  defaultNetstreamDriverCertFile="/etc/rsyslog.d/tls/server.crt"
  defaultNetstreamDriverKeyFile="/etc/rsyslog.d/tls/server.key"
)

# === Choose ONE of the two wiring options below ===
# A) WITHOUT filters (send inputs straight to egress)
# input(type="imudp" port="514" ruleset="to_logpoint")
# input(type="imtcp" port="514" ruleset="to_logpoint")
# input(type="imtcp" port="6514" StreamDriver.name="gtls" StreamDriver.mode="1"
StreamDriver.authmode="anon" ruleset="to_logpoint")

# B) WITH filters (enable regex blacklist before egress)
input(type="imudp" port="514" ruleset="pre_filter")
input(type="imtcp" port="514" ruleset="pre_filter")
input(type="imtcp" port="6514" StreamDriver.name="gtls" StreamDriver.mode="1"
StreamDriver.authmode="anon" ruleset="pre_filter")

# Include drop-ins
$IncludeConfig /etc/rsyslog.d/*.conf
```

> Notes:
>
> - `omfwd` is builtin; no need to load it explicitly.
> - Template used later: `RSYSLOG_SyslogProtocol23Format` (RFC5424-like).
> - **Do not pin ciphers** in rsyslog; inherit **system FIPS** crypto policies.

5.2 `/etc/rsyslog.d/10-esa-lb.conf` — **LB + buffering + failover (two egress options: TLS default, CLEAR optional)**

```
# Keep impstats out of the forward stream (optional)
if ($syslogtag == 'impstats:') then {
  action(type="omfile" file="/var/log/rsyslog_stats.json")
  stop
}

# Main forwarding ruleset to Logpoint
ruleset(name="to_logpoint") {
  ########################################################################
  # 5.2.A DEFAULT — GnuTLS (TLS) egress with round-robin LB (ENABLE THIS)
  ########################################################################
  action(
    name="lp_tls_rr"
    type="omfwd" protocol="tcp"
    StreamDriver="gtls" StreamDriverMode="1" StreamDriverAuthMode="anon"
    target=["<BACKEND_1>","<BACKEND_2>"]   # SAME SYNTAX used for CLEAR option below
    port="6514"
    template="RSYSLOG_SyslogProtocol23Format"

    # --- Buffering (disk-assisted action queue) — identical across options
    queue.type="LinkedList"
    queue.filename="q_logpoint_rr"
    queue.maxdiskspace="10g"
    queue.size="50000"
    queue.highwatermark="40000"
    queue.lowwatermark="10000"
    queue.dequeuebatchsize="1024"
    queue.workerthreads="2"
    queue.saveonshutdown="on"

    # --- Availability / retry policy — identical across options
    action.resumeRetryCount="-1"
    action.resumeInterval="30"
  )

  # (Optional) Local fallback while the main action is suspended
  action(
    name="local_fallback" type="omfile"
    file="/var/log/esa_fallback-buffer.log"
    execOnlyWhenPreviousIsSuspended="on"
  )

  ########################################################################
  # 5.2.B OPTIONAL — CLEAR egress with round-robin LB (DISABLE BY DEFAULT)
  # Use ONLY for debugging in controlled environments (no TLS encryption)
  ########################################################################
  # action(
  #   name="lp_clear_rr"
  #   type="omfwd" protocol="tcp"
  #   target=["<BACKEND_1>","<BACKEND_2>"]   # SAME SYNTAX as TLS option
  #   port="514"
  #   template="RSYSLOG_SyslogProtocol23Format"
  #
  #   # --- Buffering (disk-assisted action queue) — identical to TLS
  #   queue.type="LinkedList"
  #   queue.filename="q_logpoint_rr"
  #   queue.maxdiskspace="10g"
  #   queue.size="50000"
  #   queue.highwatermark="40000"
  #   queue.lowwatermark="10000"
  #   queue.dequeuebatchsize="1024"
  #   queue.workerthreads="2"
  #   queue.saveonshutdown="on"
  #
  #   # --- Availability / retry policy — identical to TLS
  #   action.resumeRetryCount="-1"
  #   action.resumeInterval="30"
  # )
}
```

Behavior

- **Filters (optional)**: messages matching **any** blacklist regex are **dropped**; others are forwarded.
- **TLS (default)** and **CLEAR (optional)** egress actions are **parameter-for-parameter identical**, except for TLS driver/port.
- If any backend is **down**, `omfwd` skips it and uses the next in the pool.
- If **all** are down, the action becomes **suspended**; events buffer until a target returns.

---

6) Customize backends, filters & tests

Replace placeholders with production targets

```
sudo sed -i 's/<BACKEND_1>/192.0.2.10/' /etc/rsyslog.d/10-esa-lb.conf
sudo sed -i 's/<BACKEND_2>/192.0.2.11/' /etc/rsyslog.d/10-esa-lb.conf
```

Create filter directory & sample rules

```
sudo install -d -m 755 /etc/rsyslog.d/blacklist.d
cat <<'EOF' | sudo tee /etc/rsyslog.d/blacklist.d/10-drop-link-flaps.conf
if re_match($rawmsg, "(?i)link (is )?(up|down)") then { stop }
EOF
```

Connectivity smoke (TLS 6514)

```
timeout 3 bash -c 'cat < /dev/null > /dev/tcp/<BACKEND_1>/6514' && echo "OK" ||
echo "FAIL"
timeout 3 bash -c 'cat < /dev/null > /dev/tcp/<BACKEND_2>/6514' && echo "OK" ||
echo "FAIL"
```

7) Enable & verify

```
# Syntax check
sudo rsyslogd -N1

# Enable & start
sudo systemctl enable --now rsyslog
sudo systemctl status rsyslog --no-pager

# Send test events
logger -t ESA_SMOKE "rsyslog LB end-to-end OK"
logger -t test "Link is Down on sw-01"   # should be DROPPED by blacklist example
```

Operational checks

- **Round-robin**: stop one backend; confirm flow continues; restore and observe alternation.
- **Buffering**: stop **both** backends; generate traffic; verify spool growth under `/var/spool/rsyslog/` and `impstats` queue metrics; restore and confirm automatic drain.
- **Filters**: verify that blacklisted patterns are dropped; tune/add rules in `blacklist.d`.

8) Monitoring / operations with `impstats`

We emit `impstats` (JSON/CEE) every 60s to `/var/log/rsyslog_stats.json`.

- Latest 20 lines:

```
tail -n 20 /var/log/rsyslog_stats.json
```

- Follow live:

```
tail -f /var/log/rsyslog_stats.json
```

- Summarize action/queue health (requires `jq`):

```
sed 's/^@cee: //' /var/log/rsyslog_stats.json | \
  jq -r 'select(.name=="action" and .actionName=="lp_tls_rr") |
         "\(.timegenerated) submitted=\(.submitted) failed=\(.failed) suspended=\(.suspended) queuesize=\(.queuesize)"'
```

What to watch

- `submitted` rising; `failed` ≈ 0 in normal conditions
- `suspended=true` when all targets are unavailable
- `queuesize` and `/var/spool/rsyslog` growth during outage, then **drain** after recovery

---

9) RHEL 8 vs RHEL 9 notes (GnuTLS + FIPS)

- Both EL8 and EL9 inherit **system crypto policies** in FIPS mode; avoid pinning ciphers in rsyslog.
- EL9 uses OpenSSL 3 system-wide; EL8 uses OpenSSL 1.1, but rsyslog **GnuTLS** driver relies on GnuTLS/its crypto backend and remains FIPS-compatible when system FIPS is enabled.
- If policy later mandates **mTLS**, switch `StreamDriverAuthMode` from `anon` to `x509/name` and deploy client certs appropriately (CA trust, cert/key paths).

---

10) Rollback / change control

- To **pause LB** quickly: replace the `target=[...]` pool with a single backend.
- To **revert packages**: disable Adiscon repo and `dnf downgrade` to the previous build, or remove `rsyslog-gnutls` if needed.
- Configuration is self-contained under `/etc/rsyslog*`; keep a backup before changes.

---

11) Quick checklist (copy/paste)

1. Install (repo or manual) `rsyslog` + `rsyslog-gnutls` (v8.2502+).
2. Enable **FIPS** (if required) and **reboot**.
3. Create TLS dir + certs under `/etc/rsyslog.d/tls/`.
4. **(Optional)** Enable **regex blacklist**: create `05-pre-filter.conf`, `blacklist.d/`, and wire inputs to `pre_filter`.
5. Apply configs in §5.1 and §5.2.
6. Open firewalld ports 514/tcp, 514/udp, 6514/tcp.
7. `rsyslogd -N1` → fix errors.
8. `systemctl enable --now rsyslog`.
9. Replace `<BACKEND_1/2>` and run smoke tests.
10. Validate **round-robin**, **buffering**, and **filters** behavior.