ESA CSOC — Rsyslog Load-Balancer (TCP/TLS Round-Robin) — **FIPS-Compliant Version** (RHEL 8/9)

Author: Gaetan De Dobbeleer (Logpoint)
Date: 13 Aug 2025
Scope: Implement a robust rsyslog relay on **RHEL 8/9** that load-balances outgoing syslog traffic to multiple Logpoint collectors/backends over **TCP/TLS (6514)** using the native `omfwd` **target pool**. This version enforces **FIPS compatibility**: OpenSSL backend, system FIPS policy, no custom ciphers.

---

1) Executive summary

- **OS**: RHEL 8 or RHEL 9 (x86_64).
- **Rsyslog**: **v8.2502+ recommended** (≥ 8.2408 required for native LB).
- **TLS/FIPS**: Use **OpenSSL backend** (`rsyslog-openssl`) and inherit system FIPS crypto policy.
- **Ingress on relay**: listen on **UDP 514**, **TCP 514**, **TCP 6514 (TLS)**.
- **Egress from relay**: **TCP 6514 (TLS)** with **round-robin** load-balancing to multiple backends.
- **Resilience**: disk-assisted queues; automatic retry; buffer to disk when **all** targets are down; auto-drain on recovery.
- **Security**: SELinux **Enforcing**; firewalld open only for 514/udp, 514/tcp, 6514/tcp.
- **Observability**: `impstats` every 60s in JSON/CEE for queue/action health.

> **Outcome**: transparent round-robin distribution to multiple Logpoint backends, safe buffering during outages, and clean automatic recovery.

---

2) Why this rsyslog version?

- Native `omfwd` **target pool** (`target=["ip1","ip2",...]`) landed in **8.2408.0**.
- RHEL stock repos may lag; use Adiscon builds or manual install to get **v8.2502+** (fixes TLS handshake stalls and improves reliability).

---

3) Prerequisites

- **RHEL 8/9** server (x86_64), root access.
- Network reachability to Logpoint backends.
- **Fixed Logpoint ports: UDP 514**, **TCP 514 (clear)**, **TCP 6514 (TLS)**.
- **TLS material** on the relay:
  - `/etc/rsyslog.d/tls/ca.crt`
  - `/etc/rsyslog.d/tls/server.crt`
  - `/etc/rsyslog.d/tls/server.key` (0600)
- **Backends:** replace placeholders with production targets: `<BACKEND_1>`, `<BACKEND_2>` (add more if needed).

OS check helper (on any host):

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

---

4) Manual package installation (FIPS-compliant set)

> Use this when the client wants to **download and install RPMs manually** (no repos). The same approach works for EL8 and EL9; just pick the right directory.

4.1 Choose the correct directory

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

4.2 Required packages

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

4.3 Verify signatures & install

```
# Import vendor GPG key and verify RPMs
sudo rpmkeys --import https://rpms.adiscon.com/RPM-GPG-KEY-Adiscon
rpm -K rsyslog-*.rpm  # should report a good signature

# Install locally (dnf resolves remaining deps from enabled RHEL repos)
sudo dnf install ./rsyslog-*.rpm

# Offline bundle option: stage everything including dependencies
# (run on a connected RHEL box, then copy to the target)
sudo dnf download --resolve rsyslog rsyslog-openssl
# then on the target:
sudo dnf install ./*.rpm
```

> **Dependencies** (auto-resolved by `dnf` in most cases): `libestr`, `libfastjson`/`libfastjson4`, `liblogging`, `openssl-libs` (OpenSSL), plus module-specific libs (`librelp`, `liblognorm`, etc.).

---

5) Alternative: repo-based install (if allowed by policy)

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

# Security hygiene
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-openssl
rsyslogd -v   # verify build; expect v8.2502+ and OpenSSL backend
```

---

6) System hardening for FIPS

- **Enable system FIPS** (if not already enforced by your image):
  - **RHEL 9/8**:

    ```
    sudo fips-mode-setup --enable
    sudo reboot
    ```

  - **RHEL 8 only** (if required by your baseline):

    ```
    sudo update-crypto-policies --set FIPS
    ```

- Keep **SELinux = Enforcing**.
- **Firewalld** — open only the required service ports:

  ```
  sudo firewall-cmd --permanent --add-port=514/tcp
  sudo firewall-cmd --permanent --add-port=514/udp
  sudo firewall-cmd --permanent --add-port=6514/tcp
  sudo firewall-cmd --reload
  ```

---

7) Configuration layout

Files

- `/etc/rsyslog.conf` — core, inputs, TLS defaults, `impstats`.
- `/etc/rsyslog.d/10-esa-lb.conf` — outbound to Logpoint via LB + buffering + failover.
- `/etc/rsyslog.d/tls/` — TLS material.

7.1 `/etc/rsyslog.conf` (excerpt, **OpenSSL/ossl**)

```
# 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="ossl"
  defaultNetstreamDriverCAFile="/etc/rsyslog.d/tls/ca.crt"
  defaultNetstreamDriverCertFile="/etc/rsyslog.d/tls/server.crt"
  defaultNetstreamDriverKeyFile="/etc/rsyslog.d/tls/server.key"
)

# Inputs from sources (fixed ports)
input(type="imudp" port="514" ruleset="to_logpoint")
input(type="imtcp" port="514" ruleset="to_logpoint")
input(type="imtcp" port="6514"
      StreamDriver.name="ossl" StreamDriver.mode="1" StreamDriver.authmode="anon"
      ruleset="to_logpoint")

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

> Notes:
>
> - `omfwd` is builtin.
> - Template used later: `RSYSLOG_SyslogProtocol23Format` (RFC5424-like).
> - No explicit cipher list → **inherit system FIPS** policy.

7.2 `/etc/rsyslog.d/10-esa-lb.conf` — **LB + buffering + failover**

```
# 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") {
  # --- Preferred: TLS 6514 with native round-robin load-balancing (OpenSSL
backend)
  action(
    name="lp_tls_rr"
    type="omfwd" protocol="tcp"
    StreamDriver="ossl" StreamDriverMode="1" StreamDriverAuthMode="anon"
    target=["<BACKEND_1>","<BACKEND_2>"]   # round-robin pool (add more if needed)
    port="6514"
    template="RSYSLOG_SyslogProtocol23Format"

    # --- Buffering (disk-assisted action queue)
    queue.type="LinkedList"                         # async, supports DA-queue
    queue.filename="q_logpoint_tls"                 # enables spooling under
workDirectory
    queue.maxdiskspace="10g"                        # cap for on-disk backlog
    queue.size="50000"                              # in-memory elements
```

```
    queue.highwatermark="40000"
    queue.lowwatermark="10000"
    queue.dequeuebatchsize="1024"
    queue.workerthreads="2"
    queue.saveonshutdown="on"                    # persist on planned reboot

    # --- Availability / retry policy
    action.resumeRetryCount="-1"                 # retry forever
    action.resumeInterval="30"                   # backoff between attempts (s)
  )

  # --- Local fallback (executes only if previous action is suspended)
  action(
    name="local_fallback" type="omfile"
    file="/var/log/esa_fallback-buffer.log"
    execOnlyWhenPreviousIsSuspended="on"
  )
}
```

> **Behavior**
>
> - If one backend is **down**, `omfwd` **skips it** and sends to the next one in the pool.
> - If **all** are down, the action becomes **suspended** and events are **buffered** (memory → disk) until a target returns.
> - On recovery, rsyslog **automatically resumes** and **replays** the backlog in order.
> - The **local fallback** only activates while the TLS action is suspended.

---

8) Verification & smoke tests

```
# Syntax check
sudo rsyslogd -N1

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

# Send a test event
logger -t ESA_SMOKE "rsyslog LB end-to-end OK"
```

Operational checks

- **Round-robin**: stop one backend; confirm flow continues; restore and observe alternation.
- **Buffering**: stop **both** backends; generate traffic; confirm spool growth under `/var/spool/rsyslog/` and impstats queue metrics; restore and confirm automatic drain.

---

9) Monitoring / operations with `impstats`

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

- **Quick view (latest 20 lines):**

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

- **Follow in real time:**

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

- **Parse JSON (strip the @cee: prefix) and summarize the 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` should steadily increase; `failed` near zero in normal conditions.
- `suspended=true` indicates the action is paused (e.g., all targets unavailable).
- `queuesize` and on-disk spool size (`du -sh /var/spool/rsyslog`) should **stabilize** then **drain** after recovery.
- For the fallback file, check growth of `/var/log/esa_fallback-buffer.log` during suspension.

Housekeeping

- Cap disk usage via `queue.maxdiskspace` (here `10g`); increase for longer outages.
- Review journald for rsyslog suspension/resume notices.

---

10) Rollback / change control

- To **pause LB** quickly, comment the `target=[...]` action and point to a single backend.
- To **revert packages**: remove the OpenSSL module if required or downgrade rsyslog.
- Configuration is self-contained under `/etc/rsyslog*`; keep a backup before changes.

---

11) Appendix

A. 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
sudo systemctl restart rsyslog
```

B. Connectivity smoke test (optional)

```
# 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"
```

C. FIPS verification

```
fips-mode-setup --check
rsyslogd -v      # confirm rsyslog built with OpenSSL
openssl version  # RHEL 9: provider-based OpenSSL 3; RHEL 8: OpenSSL 1.1
```

D. RHEL 8 vs 9 notes

- EL9 uses OpenSSL 3; EL8 uses OpenSSL 1.1. Keep defaults; do not pin ciphers in rsyslog.
- If policy later requires **mTLS**, switch `AuthMode` to `x509/name` and deploy client certs accordingly.