Detailed Test Plan for Rsyslog Load-Balancer Implementation

Environment Assumptions:

- RHEL 8 or 9 (x86_64) with root access.
- Logpoint backends accessible (replace `<BACKEND_1>`, `<BACKEND_2>` with actual IPs, e.g., 192.0.2.10, 192.0.2.11).
- Rsyslog acts as a forwarder to SIEM; tests simulate input via `logger -n 127.0.0.1 -P 514` (TCP clear on loopback).
- TLS materials in `/etc/rsyslog.d/tls/` (ca.crt, server.crt, server.key with 0600 perms).
- Test in a non-production environment to simulate outages (e.g., via firewall-cmd).

Success Criteria:

- All tests pass with expected outcomes.
- Buffering activates during full outages and drains on recovery.
- Impstats logs to `/var/log/rsyslog_stats.json` (not `/var/log/messages`).
- Load-balancing, failover, resume, and filtering work as described.
- FIPS and TLS configurations are compliant (no custom ciphers).

Tools: `rsyslogd`, `systemctl`, `logger`, `tcpdump`, `jq`, `sed`, `openssl`, `firewall-cmd`.

1. Prerequisites Verification

Objective: Validate OS, network, FIPS, TLS, and loopback connectivity (Doc: Section 2, Pages 1-2).

Prerequisites: Backends up; TLS files staged.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
1.1	OS Check: `source /etc/os-release; echo "$NAME $VERSION_ID"; rpm -E %rhel; uname -m`	"Red Hat Enterprise Linux 8" or "9", x86_64.	Check /etc/redhat-release; abort if incompatible.	5 min
1.2	FIPS Mode: `fips-mode-setup --check`	"FIPS mode is enabled" if required.	Enable: `fips-mode-setup --enable && reboot`; verify /proc/cmdline.	10 min
1.3	Backend Network: `timeout 3 bash -c 'cat < /dev/null > /dev/tcp/<BACKEND_1>/6514' && echo OK` (repeat for all, port 514 if CLEAR).	"OK" for all.	`firewall-cmd --list-all` (open 514/udp,tcp, 6514/tcp); ping backends.	10 min
1.4	Loopback (SIEM Input): `timeout 3 bash -c 'cat < /dev/null > /dev/tcp/127.0.0.1/514' && echo OK`	"OK" (rsyslog listens on 514/tcp).	Check imtcp config; no local firewall blocks.	5 min

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
1.5	TLS Materials: `ls -l /etc/rsyslog.d/tls/{ca.crt,server.crt,server.key}`; `chmod 0600 /etc/rsyslog.d/tls/server.key`; `openssl verify -CAfile /etc/rsyslog.d/tls/ca.crt /etc/rsyslog.d/tls/server.crt`	Files exist, key 0600, verify "OK".	Generate/copy certs; test handshake: `openssl s_client -connect <BACKEND_1>:6514 -CAfile ca.crt`.	10 min
1.6	Disk Space: `df -h /var`	>10GB free (for queues).	Free space or expand /var.	5 min

Explanation: Ensures foundational compatibility. Loopback simulates SIEM forwarding.

2. Installation Validation

Objective: Confirm rsyslog v8.2502+ and gnutls via Adiscon or RPM (Doc: Section 3, Pages 2-3).

Prerequisites: Choose repo or manual; internet or RPMs ready.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
2.1	Repo Setup (if repo): `cd /etc/yum.repos.d/; curl -O https://rpms.adiscon.com/v8-stable/rsyslog-rhel.repo`; `sed -i 's/^gpgcheck=.*/gpgcheck=1/' *.repo`; `dnf clean all && dnf makecache`.	Repo set, gpgcheck=1.	Use daily repo; check curl.	10 min
2.2	Install: `dnf install -y rsyslog rsyslog-gnutls` (or manual: `dnf install ./rsyslog-*.rpm`).	Installed successfully.	Verify GPG: `rpmkeys --import https://rpms.adiscon.com/RPM-GPG-KEY-Adiscon`; `rpm -K *.rpm`.	10 min
2.3	Version: `rsyslogd -v`	>=8.2408 (ideally 8.2502+), gnutls listed.	Manual RPM from Adiscon site.	5 min

Explanation: Version required for native LB; gnutls for FIPS/TLS.

3. Configuration Setup and Syntax Check

Objective: Apply core configs, inputs, TLS defaults, rulesets, queues (Doc: Sections 5-5.2, Pages 3-6).

Prerequisites: Decide wiring (A: no filters, B: with pre_filter); work dir ready.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
3.1	Work Dir: `mkdir -p /var/spool/rsyslog; chown root:root /var/spool/rsyslog; chmod 700 /var/spool/rsyslog`.	Dir exists, permissions OK.	SELinux: `restorecon -R /var/spool/rsyslog`.	5 min
3.2	Edit /etc/rsyslog.conf: Add global(workDirectory), load imudp/imtcp/impstats, TLS globals, inputs with ruleset (to_logpoint or pre_filter), $IncludeConfig.	Matches Doc page 4.	`rsyslogd -N1`; debug `rsyslogd -d -n` or `rsyslogd -d -n > debug.log 2>&1`.	20 min
3.3	Edit /etc/rsyslog.d/10-esa-lb.conf: Add impstats rule (to /var/log/rsyslog_stats.json), to_logpoint ruleset with lp_tls_rr action, queue params, fallback. Comment CLEAR.	Matches Doc pages 5-6.	Ensure stop after impstats; fix syntax.	20 min
3.4	Syntax: `rsyslogd -N1`	No errors.	Verbose debug: `rsyslogd -d -n`.	5 min

Explanation: Modular configs; impstats rule fixes routing issue.

4. Customization (Backends and Filters)

Objective: Replace placeholders, setup optional blacklist (Doc: Section 6, Page 6; Section 5.0, Page 3).

Prerequisites: Configs applied.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
4.1	Backends: `sed -i 's/<BACKEND_1>/192.0.2.10/' /etc/rsyslog.d/10-esa-lb.conf` (repeat).	Updated targets.	`grep target *.conf`.	5 min
4.2	Filters (if B): `mkdir -p /etc/rsyslog.d/blacklist.d -m 755`; Create 05-pre-filter.conf; Add samples like 10-drop-link-flaps.conf.	Rules loaded.	`rsyslogd -N1`; test regex.	15 min
4.3	Test Filter: `logger -n 127.0.0.1 -P 514 "Link is down"`; `logger -n 127.0.0.1 -P 514 "Normal test"`.	Dropped vs forwarded.	Temp log drops to file.	10 min

Explanation: Customizes for env; regex on $rawmsg.

5. Service Activation and Basic Functionality

Objective: Start and basic smoke tests (Doc: Section 7, Page 7).

Prerequisites: Syntax OK.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
5.1	Start: `systemctl enable --now rsyslog; systemctl status rsyslog`.	Active, no errors.	`journalctl -u rsyslog -e`.	5 min

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
5.2	Smoke: `logger -n 127.0.0.1 -P 514 -t ESA_SMOKE "end-to-end OK"`.	Appears on backends.	`tcpdump port 6514`.	10 min

Explanation: Verifies basic flow.

6. Input and Forwarding Tests

Objective: Confirm inputs/forwarding (Doc: Page 4).

Prerequisites: Service up.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
6.1	TCP 514: `logger -n 127.0.0.1 -P 514 "TCP test"`.	Forwarded.	Check imtcp.	10 min

Explanation: SIEM simulation.

7. Load-Balancing and Failover Tests

Objective: Verify LB, failover, resume (Doc: Page 6, Behavior).

Prerequisites: Backends; monitor impstats.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
7.1	Both Available (RR): Both up; `for i in {1..20}; do logger -n 127.0.0.1 -P 514 "RR $i"; sleep 0.5; done`; Check backends/impstats.	Alternates; submitted rising, failed=0.	`tcpdump`; target pool.	15 min
7.2	One Down (Failover): Drop <BACKEND_1>; `for i in {1..10}; do logger -n 127.0.0.1 -P 514 "Failover $i"; done`; Check remaining/impstats.	All to available; failed brief.	journalctl retries.	15 min
7.3	Resume: Restore; `for i in {1..20}; do logger -n 127.0.0.1 -P 514 "Resume $i"; sleep 0.5; done`; Check alternation/impstats.	Resumes; failed=0.	Wait 30s; connectivity smoke.	15 min
7.4	High-Load Failover: One down; `for i in {1..50000}; do logger -n 127.0.0.1 -P 514 "High $i"; done`.	Handles without drops.	Tune threads.	10 min

Explanation: Covers full/partial outages.

8. Buffering and Queue Tests

Objective: Validate buffering/resume (Doc: Pages 5-6).

Prerequisites: High traffic; monitor spool/impstats.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
8.1	Both Down (Buffer): Drop both; `for i in {1..60000}; do logger -n 127.0.0.1 -P 514 "Buffer $i"; done`; Check spool/impstats.	Suspended=true; files grow; queuesize >40000.	Lower highwatermark.	20 min
8.2	Draining: Unblock; Wait 2-5 min; `logger -n 127.0.0.1 -P 514 "Post"`.	Drains; queuesize=0; files gone.	journalctl drain.	15 min
8.3	Fallback: Check fallback.log during down.	Populates.	Config flag.	10 min
8.4	Max Limit: Extend outage; Generate >10g; Check.	Stops at 10g.	Disk usage.	10 min
8.5	Filters+Buffer: Send blacklisted during down.	Dropped pre-queue.	pre_filter.	10 min

Explanation: Tests full suspension/drain.

9. Monitoring with Impstats

Objective: Verify stats (Doc: Section 8, Pages 7-8).

Prerequisites: >60s running.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
9.1	File: `tail -n 20 /var/log/rsyslog_stats.json`.	JSON stats, not in messages.	Fix rule+stop.	10 min
9.2	Summarize: sed/jq command (Doc page 8).	Metrics as expected.	Install jq.	10 min
9.3	Live: `tail -f` during tests.	Updates 60s.	Interval.	10 min

Explanation: Integrates with outage tests.

10. FIPS and TLS-Specific Tests

Objective: Compliance (Doc: Sections 1,9, Pages 1,8-9).

Prerequisites: FIPS on.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
10.1	CLEAR Failover/Buffer: Uncomment CLEAR; Repeat 7.2/8.1.	Works (debug only).	Re-comment.	10 min
10.2	TLS Failover: Invalidate cert; Repeat 7.2.	Failovers.	journalctl.	10 min

Explanation: TLS edges.

11. Edge Cases and Debugging

Objective: Stress and fixes.

Prerequisites: Prior passed.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
11.1	High Load: `for i in {1..100000}; do logger -n 127.0.0.1 -P 514 "Load $i"; done`.	No drops.	Threads.	15 min
11.2	Buffer Debug: Lower watermark; retest.	Populates.	Permissions.	15 min
11.3	Impstats Debug: `grep impstats /var/log/messages`.	None.	Rule.	10 min
11.4	Retry: Outage; Check journalctl retries.	Every 30s.	resumeInterval.	10 min

Explanation: Additional retries.

12. Rollback and Final Validation

Objective: Reversibility (Doc: Sections 10-11, Page 8).

Prerequisites: All done.

Step	Commands/Tests	Expected Outcome	Debugging if Failure	Estimated Time
12.1	Rollback: Single target; `dnf downgrade rsyslog`.	Reverts.	Backup.	10 min
12.2	Checklist: Doc page 8 steps.	All valid.	Recap.	10 min
12.3	Final: Restart; `logger -n 127.0.0.1 -P 514 "Final"`.	Stable.	-	10 min

Explanation: Completes with checklist.