Practical Guide to Linux Logs (RHEL & SUSE) --With Real Examples

Logging is the backbone of Linux troubleshooting. When something fails—services, storage, authentication, patching, kernel, or networking, the very first clue always comes from log files.

Based on my real experience working across Red Hat & SUSE production servers, here are the most important log files along with real-life examples you can immediately relate to.

1. /var/log/messages — The First Log to Check on Any Issue

What it contains:

- Hardware issues

- Network errors

- Service failures

- Storage warnings

- Kernel-level messages

- System-wide alerts

This is the main system log file. Every important system activity hardware event, kernel warnings, service messages come here. If something goes wrong and you don't know which log to check, start here.

Real Example:

Issue: Application team reported server became slow suddenly.

Troubleshooting:

```
grep -i error /var/log/messages
```

Found repeated logs:

```
kernel: sd 0:0:0:1: I/O error, dev sdb
```

Root cause: One of the SAN disks in a multipath path failed.

Outcome: Storage team replaced faulty disk; performance normalized.

 /var/log/messages gives the first clue for 90% of issues.

2. systemd Journal — journalctl (Binary logs)

What it contains:

- Complete systemd service logs

- Failures during boot

- Timestamps + metadata

- Kernel logs

- User sessions

The systemd Journal helps check service issues, boot problems, and system events quickly using journalctl. To view a specific service log, use **journalctl -u sshd** or **journalctl -u httpd**. For detailed errors with context, use **journalctl -xe**. To view logs from the current boot run **journalctl -b**, or **use journalctl -b -1** for the previous boot. For real-time streaming similar to tail, run **journalctl -f.** To filter logs by time, use **journalctl --since "1 hour ago**" or **journalctl --since "2025-12-01" --until "2025-12-02**". For error-only logs, use **journalctl -p err**. To check logs of a specific PID, run **journalctl _PID=1234**, and to check logs of a specific systemd unit, use **journalctl _SYSTEMD_UNIT=network.service**.

Real Example:

Issue: Service httpd failed to start after patching.

Troubleshooting:

```
journalctl -u httpd -xe
```

Found:

```
permission denied: /var/www/html/logs
```

```
SELinux blocked the service.
```

Fix:

```
restorecon -Rv /var/www/html
```

journalctl gives detailed colored output + timestamps, perfect for debugging systemd services.

3 /var/log/secure (RHEL) Authentication & Security Log

What it contains:

- SSH logins

- Failed login attempts

- sudo commands

- Privilege escalations

- User/password changes

Tracks login attempts, sudo commands, SSH access, and authentication failures. Very important for security monitoring and checking unauthorized access.

Real Example:

Issue: Security team reported suspicious login spikes.

Troubleshooting:

```
grep "Failed password" /var/log/secure
```

Found:

```
Failed password for root from 185.209.xxx.xxx
```

Bruteforce attempt detected.

Fix: Added firewall rule + disabled direct root SSH login.

/var/log/secure is the first place to check during security incidents.

4. /var/log/boot.log -Boot Sequence Issues

What it contains:

- Services that failed during boot

- Kernel/init messages

- Hardware initialization

Contains logs of services that start during the boot process. Useful when the system boots slowly or a service fails to start.

Real Example:

Issue: Server took 10 minutes to boot after patching.

Troubleshooting:

```
cat /var/log/boot.log
```

Found:

```
FAILED: Starting NetworkManager-wait-online
```

Network service waited for an unreachable gateway.

Fix: Disabled NetworkManager-wait-online.

 Perfect for slow-boot troubleshooting.

5. /var/log/dmesg - Kernel & Hardware Errors

What it contains:

- HBA/NIC issues

- Hardware initialization

- Kernel warnings

- Memory errors

- Disk/USB issues

Kernel-level messages. Shows hardware initialization, disk/CPU/driver errors. Very useful for troubleshooting hardware or kernel-related issues.

Real Example:

Issue: Server frequently hung during load tests.

Troubleshooting:

```
dmesg | grep -i oom
```

Found:

```
Out of memory: Kill process 2324 (java)
```

Fix: Increased RAM & tuned JVM heap.

 dmesg is your best friend for hardware + kernel issues.

6. yum.log (RHEL) - Patch/Update History

What it contains:

- Installed packages

- Removed packages

- Kernel upgrades

Keeps history of all package installations, patches, updates, and removals done using yum or dnf/zypper.

Real Example:

Issue: After patching, the application stopped working.

Troubleshooting:

```
cat /var/log/yum.log | tail
```

Found:

```
Updated: openssl-1.1.1k
```

Fix: Application owner updated their config.

 Critical for post-patching issue debugging.

7. zypper.log (SUSE)

What it contains:

- Patch installation history

- Repo usage

- Dependency conflicts

Real Example:

Issue: SUSE server failed to install patches.

Troubleshooting:

```
tail -f /var/log/zypper.log
```

Found:

```
conflict: file from package python3-tools conflicts with python38-tools
```

Fix: Resolved dependency → patching succeeded.

8. /var/log/cron -Cron Job Execution

What it contains:

- Cron job start/run time

- Job failures

- Missed jobs

Shows all cron jobs scheduled and executed. Used to troubleshoot scheduled task failures.

Real Example:

Issue: Backup script didn't run overnight.

Troubleshooting:

```
grep backup.sh /var/log/cron
```

Output:

```
(cron) USER root failed (incorrect permissions)
```

Fix: chmod +x /scripts/backup.sh

 Check /var/log/cron when scripts silently fail.

9. /var/log/maillog Postfix/Sysmail Logs

What it contains:

- Email delivery

- SMTP errors

- Rejections

Real Example:

Issue: Monitoring alerts not received.

Troubleshooting:

```
tail -f /var/log/maillog
```

Found:

```
Relay access denied
```

Fix: Added IP to relay list.

10 /var/log/audit/audit.log - Security & SELinux Events

What it contains:

- SELinux denials
- File access logs
- Policy violations
- Privileged user activity

Real Example:

Issue: App unable to write files despite correct permissions.

Troubleshooting:

```
ausearch -m avc
```

Found:

```
avc: denied { write } for pid=3224 comm="nginx"
```

Fix:

```
setsebool -P httpd_unified 1
```

11 /var/log/wtmp, btmp - Login History

Commands:

last → successful logins

lastb → failed logins

Real Example:

Issue: Who rebooted the server?

Troubleshooting:

last | grep reboot

Found:

reboot system boot by sandeep

12 Application Logs

Common logs:

Apache: /var/log/httpd/

Nginx: /var/log/nginx/

MySQL: /var/log/mysqld.log

Pacemaker: /var/log/pacemaker.log

Example:

tail -f /var/log/httpd/error_log

Found:

client denied by server configuration

Fix: Corrected Apache config.

Final Words

Linux logs are your first point of truth when something goes wrong.

Mastering these logs helps troubleshoot:

- OS patching failures
- Boot issues
- Storage failures
- Authentication errors
- Service crashes
- Network problems
- SELinux denials
- CPU/Memory issues