LOGGING

Linux systems like Rocky use two main logging layers: classic plain-text log files under **/var/log** and the newer systemd journal managed by systemd-journald. Understanding how these pieces fit together makes the hands-on labs much easier.

1. Why logging matters

Linux logs record security events, service failures, configuration mistakes, and general system activity that administrators need for troubleshooting and auditing. On Rocky/RHEL systems, logging is built around systemd-journald plus a traditional syslog daemon such as rsyslog that still writes many logs to text files.

- Logs help answer "what happened, when, and by whom" after an incident or outage.
- Different components (kernel, services, user applications) log in different ways, but the platform tries to present a unified view.

2. Traditional log files under /var/log

Before systemd, most Linux logging flowed through a syslog daemon (rsyslog, syslog-ng) that wrote plain-text files in **/var/log**. Rocky still uses this model, so many important logs remain there.[1][2]

Typical Rocky log files students will see:

- **/var/log/messages** – general system messages from many services and the kernel, excluding some sensitive or high-volume facilities.[4]
- **/var/log/secure** – authentication and authorization events such as ssh logins and sudo attempts.[4]
- **/var/log/dnf.log** – package manager activity including installs and updates handled by DNF.[1]
- **/var/log/cron** – messages from cron scheduled jobs.[4]
- **/var/log/audit/audit.log** – detailed security audit trail when the audit subsystem is enabled.[1]

Traditional logs are:

- Stored as human-readable text, easily inspected with tools like **less**, **tail**, **grep**, and **zgrep** for rotated, compressed logs.[4][1]
- Rotated and compressed over time by logrotate, so older files appear as names like *messages-20260121* or *secure-20260121.gz*.

3. The systemd journal and journald

Systemd introduced **systemd-journald**, a logging service that collects messages from many sources and stores them in a structured binary format called the journal. On Rocky, journald is always present because systemd is pid 1.[5][6][1]

Key characteristics of the journal:

- Receives logs from the kernel, initrd, early boot, systemd units, and applications using stdout/stderr or the systemd logging APIs.[6][5]
- Stores rich metadata with each entry, such as unit name, PID, UID, boot ID, and priority, which journalctl can filter on later.[5][6]

Storage modes (controlled in **/etc/systemd/journald.conf** via **Storage=**):[7][5]

- **Storage=volatile** – keep logs only in memory under **/run/log/journal**; everything is lost on reboot.[7]
- **Storage=persistent** – keep logs in **/var/log/journal**, surviving reboots as long as disk space is available.[5][7]
- **Storage=auto** – default; use persistent if **/var/log/journal** exists and is usable, otherwise fall back to volatile storage.[7][5]

The **journalctl** command is the main interface:

- **journalctl -xe** shows recent high-priority messages with extra context, useful for live troubleshooting.[5]
- **journalctl -b** limits output to the current boot, while **-b -1** shows the previous boot once persistence is enabled.[7][5]
- **journalctl -k** focuses on kernel messages only, similar to dmesg but with timestamps and filtering.[5]

Some logs exist only in the journal—for example, very early boot messages before rsyslog starts or logs from units configured to log only to the journal. Others exist only as classic files under **/var/log**, particularly custom rsyslog outputs and audit logs that bypass journald.[6][4][5][1]

4. How journald and rsyslog work together

On a typical Rocky system, journald is the central collector and rsyslog is responsible for writing many of those messages to traditional files and for remote forwarding.[4][1]

The integration works roughly like this:

- Systemd-journald collects all local messages and exposes them via a socket (such as **/run/systemd/journal/syslog**) or via the **imjournal** input module.[4][1]
- Rsyslog loads **imjournal**, reads from the journal, and then applies its rules to write logs to text files like **/var/log/messages** and **/var/log/secure**.[4]

Rsyslog configuration:

- Main settings live in **/etc/rsyslog.conf**, with additional rules in **/etc/rsyslog.d/*.conf**.[4]
- Rules use a *facility.priority action* syntax, so you can, for example, send **local0.*** to **/var/log/app-debug.log** for application-specific debugging.[4]

Because rsyslog can both read from journald and forward logs over the network, it is often still required when building centralized logging solutions.[8][1]

5. Persistent logging and journal management

For lab 2, students focus on making sure journal logs survive reboots and learning how to manage journal size.[3][7]

Persistent configuration:

- Create **/var/log/journal** and ensure correct permissions, typically via **systemd-tmpfiles** and the tmpfiles configuration that ships with systemd.[7][5]
- Set **Storage=persistent** (or leave **auto** and just create the directory) in **/etc/systemd/journald.conf**, then restart **systemd-journald** or re-exec systemd.[5][7]

Verification and maintenance:

- **journalctl --disk-usage** shows how much space the journal currently uses.[7]
- **journalctl --vacuum-size=100M** or similar reduces stored logs to fit within the requested size, which is safe to experiment with in lab VMs.[5][7]

Once persistence is working, students can use **journalctl -b -1** after a reboot to confirm that previous boot logs are available.[7][5]

6. Querying, filtering, and when to use which tool

The journal's structured metadata makes it much easier to ask targeted questions than plain-text files allow. Labs 3–5 concentrate on practical querying and on deciding when to use **journalctl** versus grepping files.[6][5]

Common **journalctl** filters:

- **journalctl -n 50** or **-f** for "tail-like" views of the latest activity.[5]
- **journalctl --since "2026-01-21 13:00:00"** or **--since "10 minutes ago"** for precise time ranges, which is harder to do reliably with grep on text files.[5]

Service- and priority-based analysis:

- **journalctl -u sshd** shows logs for sshd across boots, pulling from all underlying sources without worrying which file they landed in.[5]
- **journalctl -p err -b** filters to error and higher priorities on the current boot, which is ideal for triaging after an incident.[5]

In contrast, grepping **/var/log/messages** or **/var/log/secure** is still convenient when:

- Investigating historical logs on systems where journald persistence was not enabled.[1][4]
- Working with custom rsyslog outputs or audit logs that are not fully mirrored into the journal.[1][4]

For the labs' discussion questions, students should recognize that:

- Some information (like very early boot logs, rich metadata, or logs from units configured as "journal only") appears only in the systemd journal.[6][5]
- Others (such as auditd's detailed audit trail or special rsyslog targets) exist only as text files under **/var/log**.[1][4]

⁂

1. https://docs.rockylinux.org/9/books/admin_guide/17-log/
2. https://www.redhat.com/en/blog/rsyslog-systemd-journald-linux-logs
3. https://reintech.io/blog/enabling-using-journald-system-logging-rocky-linux
4. https://www.server-world.info/en/note?os=Rocky_Linux_10&p=rsyslog

5. https://wiki.archlinux.org/title/Systemd/Journal
6. https://www.syslog-ng.com/community/b/blog/posts/systemd-journald-vs-syslog-ng
7. https://access.redhat.com/solutions/696893
8. https://itnixpro.com/configure-syslog-server-on-rocky-linux-8/
9. https://docs.rockylinux.org/8/books/admin_guide/17-log/
10. https://www.reddit.com/r/devops/comments/g20mte/rsyslog_and_journald_log_processing_in_linux/
11. https://stackoverflow.com/questions/16688671/configured-storage-parameter-to-be-persistent-in-journald-conf
12. https://www.youtube.com/watch?v=wF1awI26GiI
13. https://github.com/systemd/systemd/issues/26223
14. https://www.reddit.com/r/redhat/comments/n3b278/can_someone_briefly_explain_the_major_differences/
15. https://openobserve.ai/blog/journald-vs-syslog/
16. image.jpg