Lab A7 Overview: System Logging and Security Auditing on RHEL

Ayoub Zouargui
Master 1 – Networks and Embedded Systems
University of Algiers 1 (2025–2026)
https://github.com/AyoubSecurity

Introduction

Logging and auditing are essential components of Linux system security. Logging provides a chronological record of system and service events, while auditing enables precise tracking of security-sensitive actions performed by users and processes. Together, they form the foundation of accountability, incident response, and compliance in enterprise Linux environments.

Lab Objective

The objective of this lab is to explore system logging mechanisms using `journalctl` and `rsyslog`, and to implement security auditing using `auditd`. The lab demonstrates how to inspect system activity, trace authentication events, and monitor critical system files.

Lab Environment

Operating System: Red Hat Enterprise Linux

User Role: System administrator (`ayoub.admin`)

Logging Tools: `systemd-journald`, `rsyslog`

Auditing Tool: `auditd`

Execution Steps

Step 1: Verify systemd Journal Service

```
systemctl status systemd journald
```

This step confirms that the system journal service is active and collecting system events.

Step 2: View Recent System Logs

```
journalctl -n 20
```

This command displays the most recent system log entries, providing a snapshot of recent activity.

Step 3: Inspect Logs for a Specific Service (SSHD)

```
journalctl -u sshd
```

This allows examination of authentication-related events, including login attempts and SSH service behavior.

Step 4: Filter Logs by Time Range

```
journalctl --since "1 hour ago"
journalctl --since today
```

Time-based filtering is critical for incident investigation and timeline reconstruction.

Step 5: Verify rsyslog Service

```
systemctl status rsyslog
```

This ensures traditional logging to text files under `/var/log` is enabled.

Step 6: Inspect Log Directory and Authentication Logs

```
ls -l /var/log
sudo tail -n 20 /var/log/secure
```

Authentication logs are examined to identify successful and failed login attempts.

Step 7: Verify Audit Daemon Status

```
systemctl status auditd
```

The audit daemon must be running to capture security-relevant system events.

Step 8: Review Current Audit Rules

```
sudo auditctl -l
```

This displays the list of active audit rules applied to the system.

Step 9: Add an Audit Rule for a Critical File

```
sudo auditctl -w /etc/passwd -p wa -k passwd_changes
```

This rule monitors write and attribute changes to the user account database.

Step 10: Trigger an Audited Event

```
sudo nano /etc/passwd
```

A controlled modification is performed to generate an audit event.

Step 11: Search Audit Logs by Key

```
sudo ausearch -k passwd_changes
```

This retrieves detailed records of who accessed the file and when.

Step 12: Generate an Audit Summary Report

```
sudo aureport -f
```

This produces a high-level summary of audited file activity.

Outcome and Learning

By completing this lab, system logging and auditing mechanisms were examined and validated. The lab demonstrated how Linux systems record operational and security events, how administrators can trace user actions, and how auditing provides accountability for sensitive system modifications. These capabilities are fundamental for secure system administration, incident response, and regulatory compliance.