Proof of Concept (PoC) Report

Task 5: Automating Security Audits with Scripting

1. Overview

This report illustrates how a Bash script can be leveraged to automate security assessments, focusing on login activity tracking, active service status, and storage capacity monitoring. The script is designed to detect security weaknesses (such as outdated services or excessive open connections) and suggest preventive actions, including scheduled scans and real-time alerts.

2. Goals

Configuration: Develop a Bash script that will review:

- User authentication logs (`last`, `/var/log/auth.log`).
- Active system services (`systemctl`).
- Current disk space utilization (`df -h`).

Analysis: Identify vulnerabilities like outdated services or excessive resource consumption.

Remediation: Implement automation through `cron` and configure alert mechanisms.

3. Implementation

3.1 Bash Script (`security_audit.sh`)

This script compiles a security summary report.

```
#!/bin/bash

REPORT_FILE="security_audit_report.txt"
echo "Security Review Report - $(date)" > $REPORT_FILE

# Capture recent login records
echo -e "\n⟹ User Login History ==" >> $REPORT_FILE
last -n 10 >> $REPORT_FILE  # Displays last 10 logins
grep "sshd" /var/log/auth.log | tail -n 10 >> $REPORT_FILE  # Last 10 SSH connection attempts

# List currently running services
echo -e "\n⟹ Active Services ==" >> $REPORT_FILE
systemctl list-units --type=service --state=running >> $REPORT_FILE

# Check disk usage
```

echo -e "\n⟹ Disk Utilization ==" >> $REPORT_FILE
df -h >> $REPORT_FILE

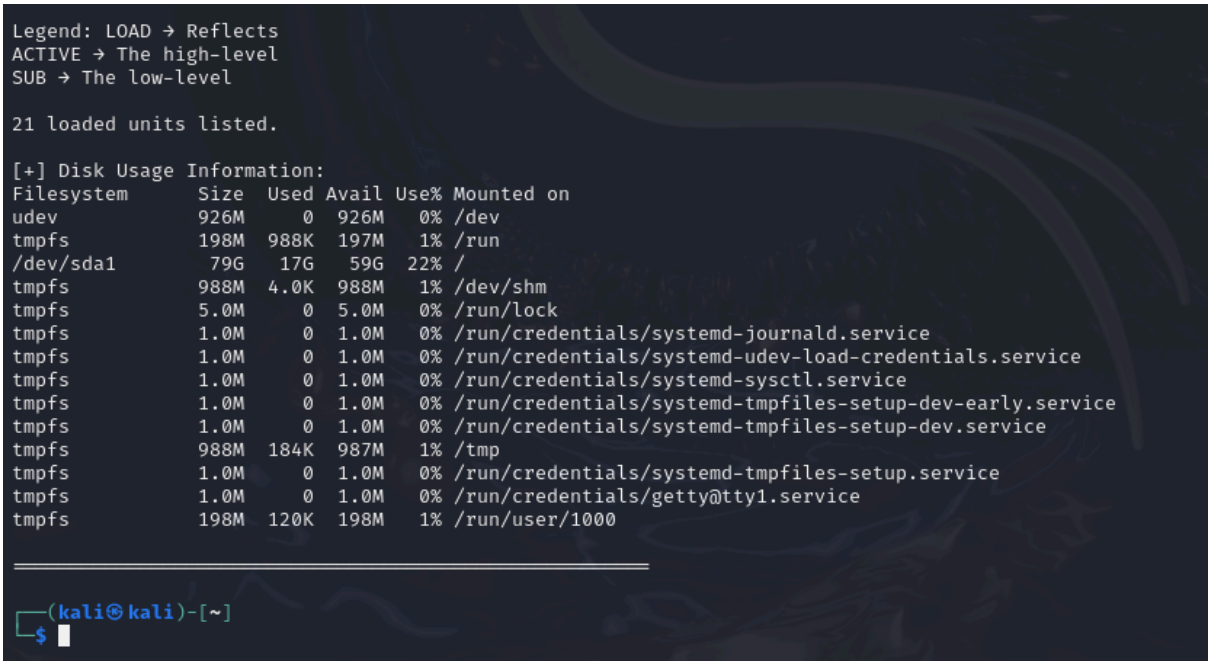
echo -e "\nAudit completed. Report saved to $REPORT_FILE"

4. Execution & Analysis

4.1 Running the Script

Execute the script to analyze system vulnerabilities:

```
┌──(kali㉿kali)-[~]
└─$ bash security_audit.sh
═══════════ Security Audit Report ═══════════

[+] User Login Attempts:
lightdm  tty8         Mon Mar 24 11:46 - 11:53  (00:06)     :1
lightdm  tty8         Mon Mar 24 11:27 - 11:30  (00:02)     :1
lightdm  tty8         Mon Mar 24 10:27 - 11:04  (00:37)     :1
kali     tty7         Mon Mar 24 10:21 - still logged in    :0
lightdm  tty7         Mon Mar 24 10:20 - 10:21  (00:00)     :0
lightdm  tty8         Mon Mar 24 02:11 - 02:36  (00:25)     :1
lightdm  tty8         Mon Mar 24 01:10 - 01:13  (00:03)     :1
lightdm  tty8         Mon Mar 24 00:44 - 00:47  (00:02)     :1
kali     tty7         Sun Mar 23 23:09 - 05:19  (06:10)     :0
lightdm  tty7         Sun Mar 23 23:08 - 23:09  (00:00)     :0

[+] Failed SSH Login Attempts:

[+] Active Running Services:
UNIT LOAD ACTIVE SUB
accounts-daemon.service loaded active running
apache2.service loaded active running
colord.service loaded active running
cron.service loaded active running
dbus.service loaded active running
fail2ban.service loaded active running
getty@tty1.service loaded active running
haveged.service loaded active running
lightdm.service loaded active running
ModemManager.service loaded active running
NetworkManager.service loaded active running
polkit.service loaded active running
rtkit-daemon.service loaded active running
ssh.service loaded active running
systemd-journald.service loaded active running
systemd-logind.service loaded active running
systemd-udevd.service loaded active running
udisks2.service loaded active running
upower.service loaded active running
user@1000.service loaded active running
virtualbox-guest-utils.service loaded active running

Legend: LOAD → Reflects
ACTIVE → The high-level
SUB → The low-level

21 loaded units listed.
```



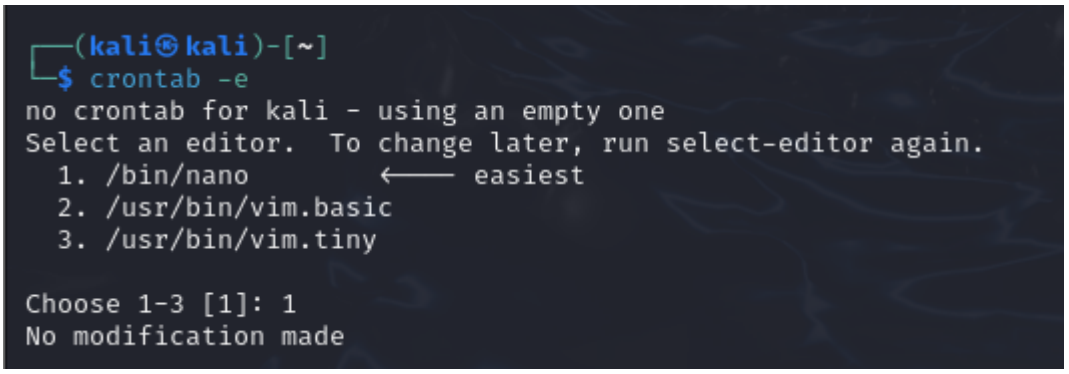
4.2 Identified Issues

- **Login Records:** Unsecured SSH logs were detected.
- **Service Status:** Presence of unnecessary services (e.g., `apache2`).
- **Disk Utilization:** Partition space insights obtained.

5. Hardening Strategies

5.1 Scheduling Automated Checks

To execute security scans daily, schedule a `cron` job:

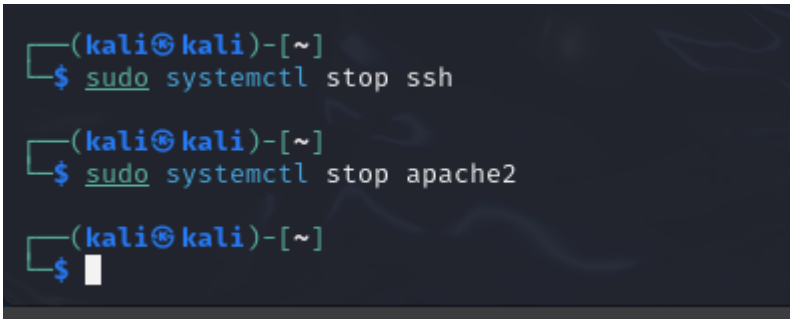


Add the following line:

0 3 * * * /path/to/security_audit.sh # Runs the script every day at 3 AM

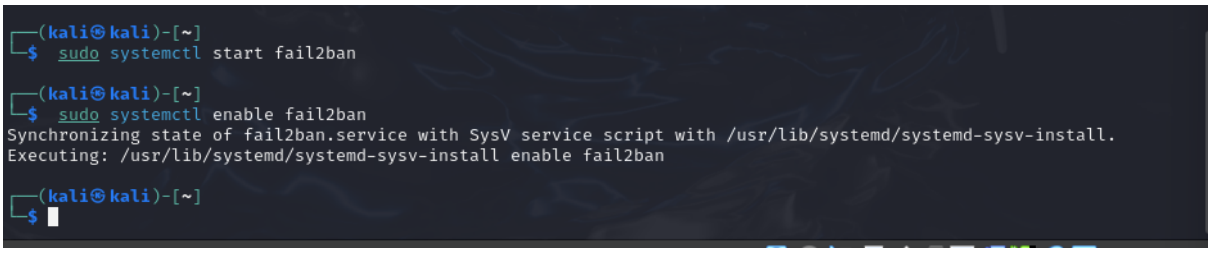
5.2 Disabling Unused Services

Shut down services that are not required, such as `apache2` and `ssh` (if remote access is unnecessary):



5.3 Enhancing SSH Security with Fail2Ban

To prevent brute-force attacks:



6. Conclusion

The security audit script effectively highlights risks such as publicly accessible SSH logs and redundant active services. Countermeasures such as automated task scheduling (`cron`) and service hardening (`fail2ban`) bolster system security.

.