

# POC TASK 5

## Step 1: Creating a Security Audit Bash Script

### 1. Writing the Script

Create a Bash script using nano and name it security\_audit.sh:

```
#!/bin/bash

# Log File
LOG_FILE="/var/log/security_audit.log"

# Function to check login attempts
check_logins() {
    echo -e "\n==== Recent User Logins =====" | tee -a "$LOG_FILE"
    last -n 10 | tee -a "$LOG_FILE"

    echo -e "\n==== Unauthorized SSH Attempts =====" | tee -a "$LOG_FILE"
    grep "Failed password" /var/log/auth.log | tail -n 10 | tee -a "$LOG_FILE"
}

# Function to check running services
check_services() {
    echo -e "\n==== Running Services =====" | tee -a "$LOG_FILE"
    systemctl list-units --type=service --state=running | tee -a "$LOG_FILE"
}

# Function to monitor disk usage
check_disk_usage() {
    echo -e "\n==== Disk Usage =====" | tee -a "$LOG_FILE"
    df -h | tee -a "$LOG_FILE"
}

# Function to send security alerts
send_alert() {
    ATTACK_COUNT=$(grep "Failed password" /var/log/auth.log | wc -l)
    if [ "$ATTACK_COUNT" -gt 10 ]; then
        echo "Security Alert: Multiple failed SSH login attempts detected!" | mail -s "Security Alert: SSH Login Attempts" root@localhost
    fi
}
```

```
# Main function
security_audit() {
    echo -e "\n==== Security Audit Report =====" | tee -a "$LOG_FILE"
    date | tee -a "$LOG_FILE"

    check_logins
    check_services
    check_disk_usage
    send_alert
}

# Execute the script
security_audit
```



```
# Log File
LOG_FILE="/var/log/security_audit.log"

# Function to check login attempts
check_logins() {
    echo -e "\n==== Recent User Logins =====" | tee -a "$LOG_FILE"
    last -n 10 | tee -a "$LOG_FILE"
    echo -e "\n==== Unauthorized SSH Attempts =====" | tee -a "$LOG_FILE"
    grep "Failed password" /var/log/auth.log | tail -n 10 | tee -a "$LOG_FILE"
}

# Function to check running services
check_services() {
    echo -e "\n==== Running Services =====" | tee -a "$LOG_FILE"
    systemctl list-units --type=service --state=running | tee -a "$LOG_FILE"
}

# Function to monitor disk usage
check_disk_usage() {
    echo -e "\n==== Disk Usage =====" | tee -a "$LOG_FILE"
    df -h | tee -a "$LOG_FILE"
}

# Function to send security alert
send_alert() {
    ATTACK_COUNT=$(grep "Failed password" /var/log/auth.log | wc -l)
    if [ "$ATTACK_COUNT" -gt 10 ]; then
        echo "Security Alert: Multiple failed SSH login attempts detected!" | mail -s "Security Alert: SSH Login Attempts" root@localhost
    fi
}

# Main function
security_audit() {
    echo -e "\n==== Security Audit Report =====" | tee -a "$LOG_FILE"
    date | tee -a "$LOG_FILE"
    check_logins
    check_services
    check_disk_usage
    send_alert
}
```

## 2. Making the Script Executable

Run the following command to give execution permissions:

- `chmod +x security_audit.sh`

## Step 2: Checking Security Parameters

### 1. Checking User Login Attempts

Command:

- `last -n 10`

### 2. Detecting Unauthorized SSH Attempts

- `grep "Failed password" /var/log/auth.log | tail -n 10`

### 3. Checking Running Services

- `systemctl list-units --type=service --state=running`

### 4. Monitoring Disk Usage

- `df -h`

### 5. Sending Security Alerts

- `grep "Failed password" /var/log/auth.log | wc -l`

```
(kali㉿kali)-[~]
└─$ nano security_audit.sh

(kali㉿kali)-[~]
└─$ last -n 10
Command 'last' not found, but can be installed with:
sudo apt install wtmpdb
Do you want to install it? (N/y)y
sudo apt install wtmpdb
The following packages were automatically installed and are no longer required:
  cpp-13          libical3t64          libmsgraph-0-1      libpython3.12-dev    libswscale7
  cpp-13-x86-64-linux-gnu libimobiledevice6    libnsl2             libpython3.12-minimal libtag1v5
  gcc-13-base     libjim0.82t64        libpaper1           libpython3.12-stdlib libtag1v5-vanilla
  imagemagick-6-common libldap-2.5-0        libperl5.38t64     libpython3.12t64     libtagc0
  libassuan0      libllvm17t64        libplacebo338      libqt6dbus6t64       libusbmuxd6
  libavfilter9    libmagickcore-6.q16-7-extra libplist3          libqt6gui6t64        libutempter0
  libavformat60   libmagickcore-6.q16-7t64 libpoppler134      libqt6network6t64    libwebrtc-audio-processing1
  libconfig++9v5  libmagickwand-6.q16-7t64 libpostproc57     libqt6opengl6t64     linux-image-6.8.11-amd64
  libdirectfb-1.7-7t64 libmbedcrypto7t64   libpython3.11-minimal libqt6widgets6t64    perl-modules-5.38
  libgspell-1-2   libmfx1             libpython3.11-stdlib libssh-gcrypt-4      python3-pexpect

Use 'sudo apt autoremove' to remove them.

Installing:
  wtmpdb

Installing dependencies:
  libpam-wtmpdb libwtmpdb0

Summary:
  Upgrading: 0, Installing: 3, Removing: 0, Not Upgrading: 360
```

## Step 3: Automating the Script with Cron

To run the script automatically every day at midnight, configure a cron job:

Open the crontab editor:

- `crontab -e`

Add this line to schedule execution at midnight:

- `0 0 * * * /path/to/security_audit.sh`

```
(kali@kali)~[~]
$ chmod +x security_audit.sh
./security_audit.sh

tee: /var/log/security_audit.log: Permission denied

===== Security Audit Report =====
tee: /var/log/security_audit.log: Permission denied
Wed Mar 12 21:50:49 IST 2025
tee: /var/log/security_audit.log: Permission denied

===== Recent User Logins =====
tee: /var/log/security_audit.log: Permission denied
open_database_ro: Cannot open database (/var/lib/wtmpdb/wtmp.db): unable to open database file
tee: /var/log/security_audit.log: Permission denied

===== Unauthorized SSH Attempts =====
tee: /var/log/security_audit.log: Permission denied
grep: /var/log/auth.log: No such file or directory
tee: /var/log/security_audit.log: Permission denied

===== Running Services =====
tee: /var/log/security_audit.log: Permission denied
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
accounts-daemon.service            loaded active running Accounts Service
apache2.service                    loaded active running The Apache HTTP Server
bluetooth.service                  loaded active running Bluetooth service
colord.service                      loaded active running Manage, Install and Generate Color Profiles
cron.service                       loaded active running Regular background program processing daemon
dbus.service                       loaded active running D-Bus System Message Bus
fail2ban.service                   loaded active running Fail2Ban Service
getty@tty1.service                 loaded active running Getty on tty1
haveged.service                    loaded active running Entropy Daemon based on the HAVEGE algorithm
lightdm.service                     loaded active running Light Display Manager
```

## Expected Script Output

After execution, the script generates a security audit report with details like:

===== Security Audit Report =====

Wed Mar 11 12:30:00 UTC 2025

===== Recent User Logins =====

root pts/0 192.168.1.100 Mon Mar 11 12:00 still logged in

===== Unauthorized SSH Attempts =====

Mar 11 12:30:01 server sshd[12345]: Failed password for invalid user admin from 192.168.1.200

===== Running Services =====

apache2.service loaded active running The Apache HTTP Server

===== Disk Usage =====

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	50G	45G	5G	90%	/

## Expected Script Output

This script helps in real-time security monitoring, allowing administrators to:

- ✓ Detect unauthorized login attempts
- ✓ Identify unnecessary running services
- ✓ Monitor disk space usage
- ✓ Receive security alerts for brute-force attempts