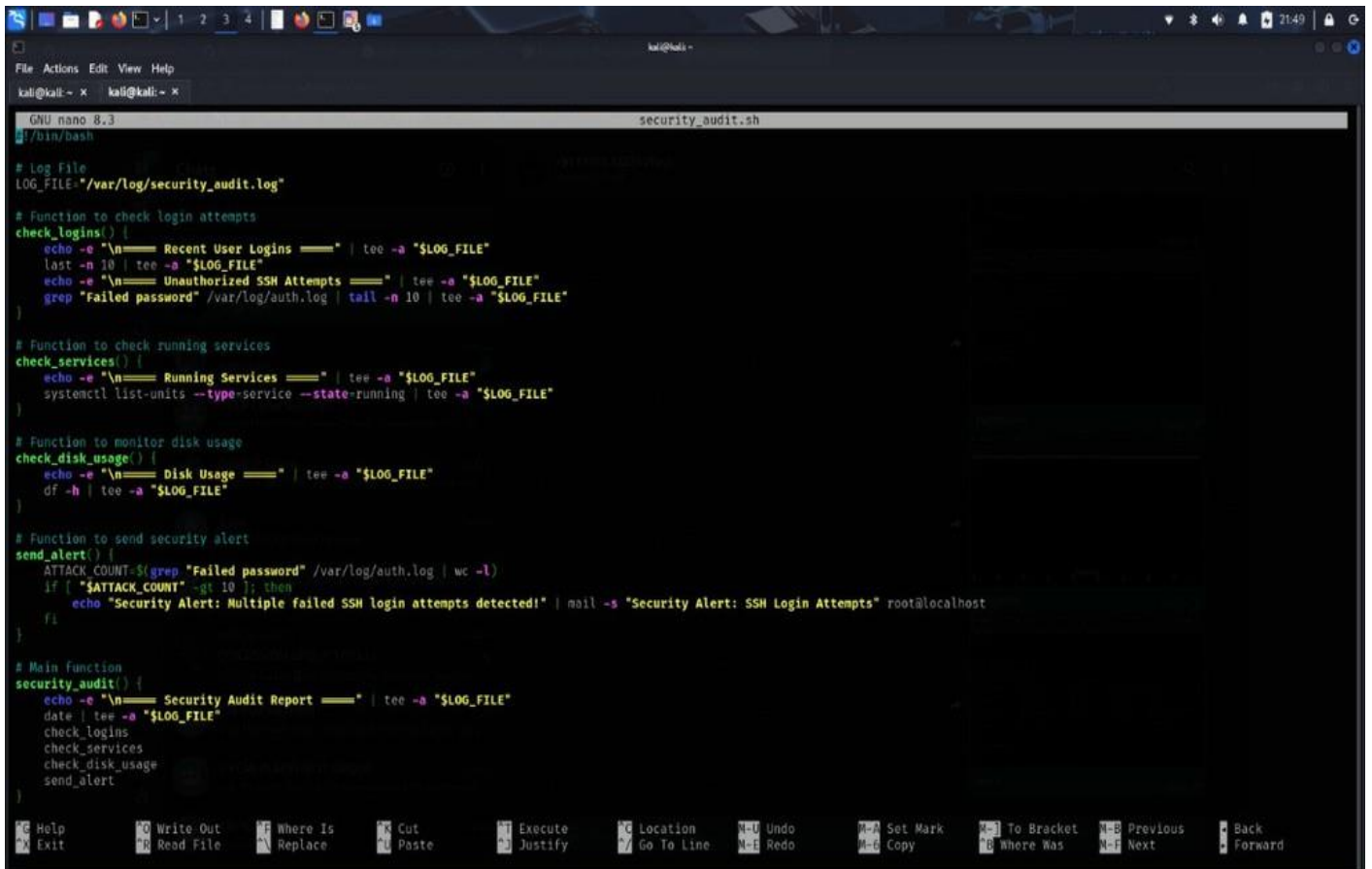


POC TASK 5

- **Bash Script:**

Write a bash script using nano called **security_audit.sh**



The screenshot shows a terminal window with the nano editor open, editing a file named `security_audit.sh`. The script content is as follows:

```
#!/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 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
}
```

- **Execute these following code:**

```
#!/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 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
```

```
}
```

```
# Execute the script
```

```
security_audit
```

- **Checking User Login Attempts Command:**

```
last -n 10
```

```
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)  
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 pyt  
  cpp-13-x86-64-linux-gnu libimobiledevice6 libnsl2 libpython3.12-minimal libtag1v5 pyt  
  gcc-13-base libjim0.82t64 libpaper1 libpython3.12-stdlib libtag1v5-vanilla pyt  
  imagemagick-6-common libldap-2.5-0 libperl5.38t64 libpython3.12t64 libtagc0 pyt  
  libassuan0 libllvm17t64 libplacebo338 libqt6dbus6t64 libusbmuxd6 pyt  
  libavfilter9 libmagiccore-6.q16-7-extra libplist3 libqt6gui6t64 libutempter0 pyt  
  libavformat60 libmagiccore-6.q16-7t64 libpoppler134 libqt6network6t64 libwebRTC-audio-processing1 pyt  
  libconfig+9v5 libmagicwand-6.q16-7t64 libpostproc57 libqt6opengl6t64 linux-image-6.8.11-amd64  
  libdirectfb-1.7-7t64 libmbcrypto7t64 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
```

- **Purpose:** Lists the last 10 user login attempts.
- **Example Output:**

- root pts/0 192.168.1.100 Mon Mar 11 12:00 still logged in
- user1 pts/1 192.168.1.101 Mon Mar 11 11:45 - 11:55 (00:10)

Security Risk: Identifies old, inactive accounts or unauthorized logins.

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

Purpose: Finds failed SSH login attempts from /var/log/auth.log.

- **Example Output:**

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

- **Security Risk:**

If there are multiple failed attempts, an attacker may be brute-forcing SSH.

- **Detecting Running Services Command:**

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

Purpose: Lists currently running system services.

Example Output:

```
UNIT LOAD ACTIVE SUB DESCRIPTION
apache2.service loaded active running The Apache HTTP Server
ssh.service loaded active running OpenBSD Secure Shell server
```

- **Security Risk:** Unnecessary services (e.g., old database servers) can expose vulnerabilities.

- **Monitoring Disk Usage**

Command: `df -h`

- **Purpose:** Displays disk space usage in a **human-readable** format.

- **Example Output:**

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

- **Security Risk:** If disk space is **over 90%**, attackers might try a **Denial-of-Service (DoS) Attack** by filling up logs or storage.

- **Sending Security Alerts**

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

Purpose: Counts the number of failed SSH login attempts.

Example Output:

15

- **Action:** If this count is greater than 10, an alert is sent.

Command: `mail -s "Security Alert: SSH Login Attempts" root@localhost`

Purpose: Sends an email alert.

Alternative: Install and configure mailutils for external

emails: `sudo apt install mailutils`

- **Running the Script**

Make the script executable:

`chmod +x 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
```

- **Run the script:**

`./security_audit.sh`

- **Expected output:**

`pgsql`

`===== 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

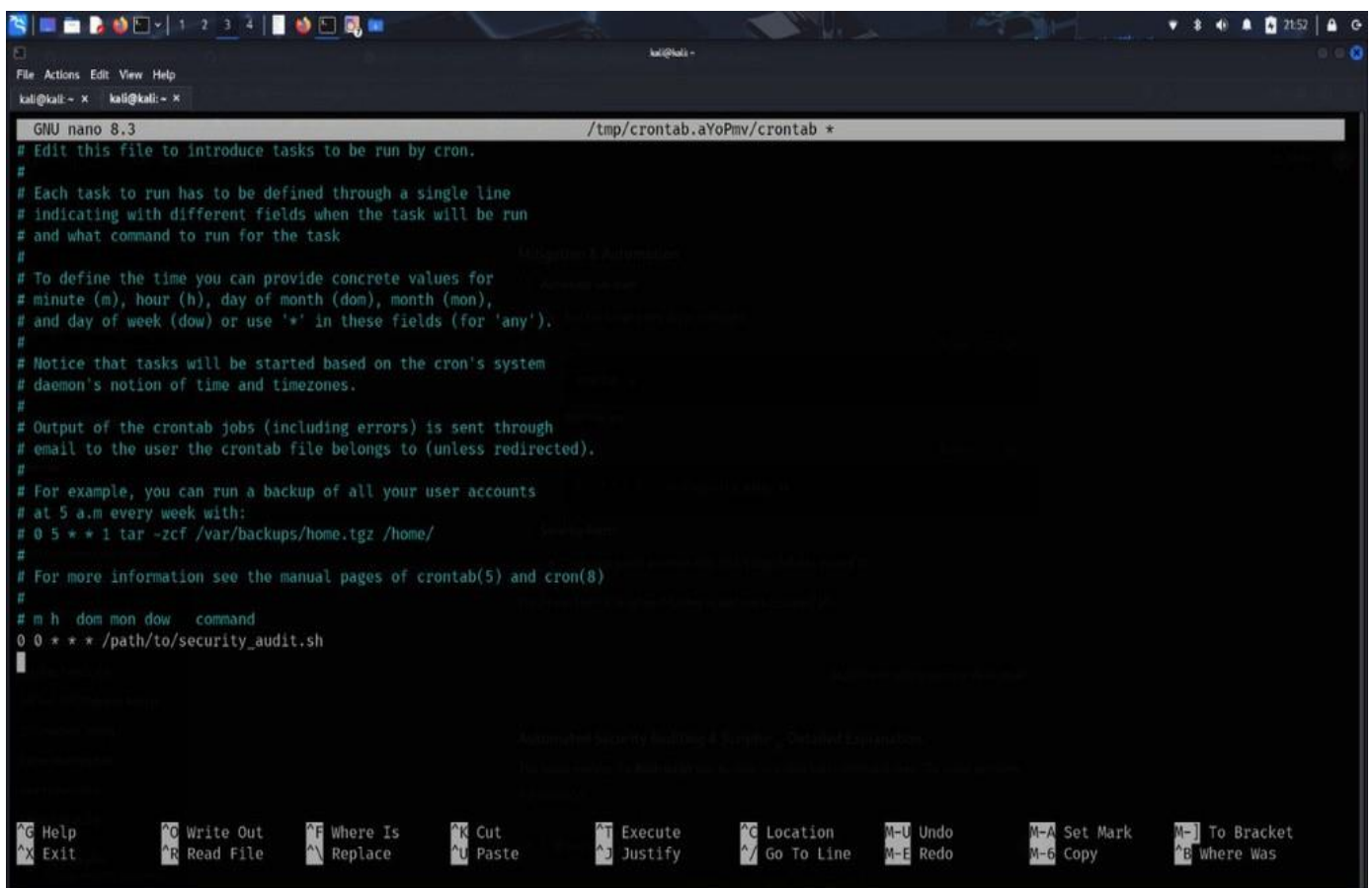
/dev/sda1 50G 45G 5G 90% /

- **Automating with Cron:**

To run the script automatically every day at

midnight, use: **crontab -e**

- **Add this line:**



```
GNU nano 8.3 /tmp/crontab.aYoPmv/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 0 * * * /path/to/security_audit.sh
```

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

This ensures the script runs **daily at midnight**.

