Proof of Concept (PoC) Report

Task 5: Automated Security Auditing & Scripting

📌 Objective:

Demonstrate how automated security auditing scripts can help identify misconfigurations and enhance system security through monitoring and alerting.

1️⃣ Setup: Creating a Security Auditing Script

Step 1: Write a Bash Script to Automate Auditing

Create a script named `security_audit.sh` that checks key security aspects:

```
kali@kali: ~
File  Actions  Edit  View  Help
  GNU nano 8.2                    security_audit.sh
#!/bin/bash

#security audit script
echo "═══ security audit report ═══"
#check the recent user logins
echo -e "\n═══ recent user logins ═══"
last -n 10

#check failed login attempts
echo -e "\n═══ failed login attempts ═══"
grep "failed password" /var/log/auth.log |tail -10

# list running serivces
echo -e "\n═══ running  services ═══"
systemctl list-units--type=service --state=running

#Monitor disk usage
echo -e "\n═══Disk usage ═══"
df -h

                    [ Read 19 lines ]
^G Help      ^O Write Out  ^F Where Is   ^K Cut      ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste    ^J Justify
```

Save and give execution permissions:

Step 2: Run the Script

```
kali@kali: ~
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~]
└─$ chmod +x security_audit.sh

┌──(kali㉿kali)-[~]
└─$ ./security_audit.sh
═══ security audit report ═══

═══ recent user logins ═══
./security_audit.sh: line 7: last: command not found

═══ failed login attempts ═══
grep: /var/log/auth.log: No such file or directory

═══ running  services ═══
Unknown command verb 'list-units--type=service'.

═══Disk usage ═══
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           392M  1.3M  391M   1% /run
/dev/vda3        27G   13G   13G  50% /
tmpfs           2.0G  4.0K  2.0G   1% /dev/shm
efivarfs        256K   25K  232K  10% /sys/firmware/efi/efivars
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-journald.servi
ce
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-udev-load-cred
entials.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-tmpfiles-setup
-dev-early.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-sysctl.service
```

2 Exploitation: Identifying Security Weaknesses

Step 3: Running the Script to Detect Issues

Running `security_audit.sh` may reveal:

```
kali@kali: ~
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~]
└─$ chmod +x security_audit.sh

┌──(kali㉿kali)-[~]
└─$ ./security_audit.sh
═══ security audit report ═══

═══ recent user logins ═══
./security_audit.sh: line 7: last: command not found

═══ failed login attempts ═══
grep: /var/log/auth.log: No such file or directory

═══ running  services ═══
Unknown command verb 'list-units--type=service'.

═══Disk usage ═══
Filesystem      Size  Used Avail Use% Mounted on
udev            1.9G     0  1.9G   0% /dev
tmpfs           392M  1.3M  391M   1% /run
/dev/vda3        27G   13G   13G  50% /
tmpfs           2.0G  4.0K  2.0G   1% /dev/shm
efivarfs        256K   25K  232K  10% /sys/firmware/efi/efivars
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-journald.servi
ce
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-udev-load-cred
entials.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-tmpfiles-setup
-dev-early.service
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-sysctl.service
```

- Old user accounts are still active.
- Multiple failed login attempts (possible brute force attack).
- Unnecessary services running (increasing attack surface).
- Low disk space (can lead to DoS attacks if the system crashes).

3 Mitigation: Automating Security Monitoring & Alerts

Step 5: Schedule the Script with Cron

To run the script daily at midnight:

sudo crontab -e

Add the following line:

0 0 * * * /path/to/security_audit.sh >> /var/log/security_audit.log 2>&1

```
kali@kali: ~
File  Actions  Edit  View  Help
  GNU nano 8.2                /tmp/crontab.qGhvQi/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
00***/path/to/security_audit.sh >> /var/log/security_audit.log 2>&1

                                [ Read 28 lines ]
^G Help      ^O Write Out   ^F Where Is    ^K Cut       ^T Execute
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify
```

Step 6: Implement Security Alerts

To get email alerts when unauthorized SSH login attempts occur, modify the script

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install mailutils
```

📌 Conclusion:

Exploitation: We demonstrated how security misconfigurations can expose a system to attacks.
Mitigation: Implemented automation to audit, monitor, and alert system administrators about security threats.
-Outcome:A proactive approach to securing Linux systems against unauthorized access and misconfigurations.

📍 Status: Secured & Automated ✅