Proof Of Concept

Linux Security- Exploitation & Hardening

Task 5: Automated Security Auditing & Scripting

1. Executive Summary

This PoC demonstrates how to create and execute a Bash script for automated security auditing. The script checks user login attempts, detects running services, monitors disk usage, and identifies potential security misconfigurations. Additionally, the task includes setting up automated monitoring using cron and implementing security alerts for unauthorized SSH login attempts.

2. Objectives

- *Setup*: Write a Bash script to perform security audits, including checking login attempts, running services, and disk usage.
- *Exploit*: Run the script to identify weak configurations and demonstrate potential exploits.
- *Mitigation*: Automate system monitoring using cron and implement security alerts.

3. Setup

3.1 Create Security Audit Script

A Bash script (security_audit.sh) was created to perform the following tasks:

- Check recent user login attempts.
- Detect failed SSH login attempts.
- List active services.
- Monitor disk usage.
- Check for inactive user accounts.
- Send email alerts for SSH brute-force attempts.

Script Content:

```
#!/bin/bash

# Security Audit Script

echo "Running Security Audit Script..."
```

```
echo "---"

# Check recent login attempts
echo "Checking User Login Attempts..."
last -n 10
echo "---"

# Check failed SSH login attempts
echo "Checking Failed SSH Login Attempts..."
FAILED_ATTEMPTS=$(journalctl -u ssh --no-pager | grep "Failed password" | tail -10)

if [[ -n "$FAILED_ATTEMPTS" ]]; then
   echo -e "Unauthorized SSH login attempts detected!\n$FAILED_ATTEMPTS"
else
   echo "No failed SSH login attempts found."
fi
echo "---"

# List active services
echo "Listing Active Services..."
systemctl list-units --type-service --state-running
echo "---"

# Monitor disk usage
echo "Checking Disk Usage..."
df -h
echo "---"

# Check for inactive user accounts (no login in 30+ days)
echo "Checking Inactive Users (No Login in 30+ Days)..."
sudo lastlog -b 30
```

echo "---"

Send email alert if SSH brute-force attempts are found

ALERT_EMAIL="your-email@example.com"

if [[-n "$FAILED_ATTEMPTS"]]; then

    echo -e "SSH Brute Force Detected on $(hostname)\n\n$FAILED_ATTEMPTS" | mail -s "Security Alert: SSH Attack" $ALERT_EMAIL

fi

echo "Security Audit Completed!"

```
┌──(llamafart㉿jaivanti)-[~]
└─$ cat security_audit.sh
#!/bin/bash

# 🛡️ Security Audit Script
echo "🔹 Running Security Audit Script..."
echo "---------------------------------"

# 1️⃣ Check recent login attempts
echo "🔍 Checking User Login Attempts..."
last -n 10
echo "---------------------------------"

# 2️⃣ Check failed SSH login attempts
echo "🔍 Checking Failed SSH Login Attempts..."
FAILED_ATTEMPTS=$(journalctl -u ssh --no-pager | grep "Failed password" | tail -10)

if [[ -n "$FAILED_ATTEMPTS" ]]; then
    echo -e "🚨 Unauthorized SSH login attempts detected!\n$FAILED_ATTEMPTS"
else
    echo "✅ No failed SSH login attempts found."
fi
echo "---------------------------------"

# 3️⃣ List active services
echo "🔍 Listing Active Services..."
systemctl list-units --type=service --state=running
echo "---------------------------------"

# 4️⃣ Monitor disk usage
echo "🔍 Checking Disk Usage..."
df -h
echo "---------------------------------"

# 5️⃣ Check for inactive user accounts (no login in 30+ days)
echo "🔍 Checking Inactive Users (No Login in 30+ Days)..."
sudo lastlog -b 30
echo "---------------------------------"

# 6️⃣ Send email alert if SSH brute-force attempts are found
ALERT_EMAIL="your-email@example.com"

if [[ -n "$FAILED_ATTEMPTS" ]]; then
    echo -e "🚨 SSH Brute Force Detected on $(hostname)\n\n$FAILED_ATTEMPTS" | mail -s "🔴 Security Alert: SSH Attack" $ALERT_EMAIL
fi

echo "✅ Security Audit Completed!"
```

3.2 Execute Security Audit Script

The script was executed to perform a security audit on the system.

Commands Used:

bash security_audit.sh

```
┌──(llamafart㉿jaivanti)-[~]
└─$ bash security_audit.sh
🔹 Running Security Audit Script...
---------------------------------
🔍 Checking User Login Attempts...
llamafar tty2                         Mon Mar 17 16:32 - 11:37 (213503982+03:06)
Debian-g tty1                         Mon Mar 17 16:32 - 16:32  (00:00)
Debian-g tty1                         Mon Mar 17 14:28 - 08:59 (213503982+02:32)
llamafar tty2                         Mon Mar 17 13:49 - still logged in
Debian-g tty1                         Mon Mar 17 13:49 - 13:49  (00:00)
llamafar tty2                         Mon Mar 17 08:59 - 10:05  (01:06)
llamafar ssh          ::1             Sun Mar 16 23:10 - 23:20  (00:09)
llamafar ssh          ::1             Sun Mar 16 23:05 - 23:20  (00:14)
llamafar tty2                         Sun Mar 16 19:16 - 23:46  (04:29)
Debian-g tty1                         Sun Mar 16 19:16 - 19:17  (00:00)
```

```
24 loaded units listed.
---------------------------------
🔍 Checking Disk Usage...
Filesystem      Size  Used Avail Use% Mounted on
udev            7.7G     0  7.7G   0% /dev
tmpfs           1.6G  1.9M  1.6G   1% /run
/dev/nvme0n1p5   55G   29G   24G  55% /
tmpfs           7.7G  4.0K  7.7G   1% /dev/shm
efivarfs        192K  118K   69K  64% /sys/firmware/efi/efivars
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-journald.service
tmpfs           7.7G     0  7.7G   0% /tmp
/dev/nvme0n1p1  256M  105M  152M  41% /boot/efi
tmpfs           1.6G  128K  1.6G   1% /run/user/1000
---------------------------------
🔍 Checking Inactive Users (No Login in 30+ Days)...
sudo: lastlog: command not found
---------------------------------
security_audit.sh: line 42: mail: command not found
✅ Security Audit Completed!
```

4. Exploitation

4.1 Identify Weak Configurations

The script identified recent user login attempts, active services, and disk usage. It also checked for failed SSH login attempts and inactive user accounts.

```
┌──(llamafart㉿jaivanti)-[~]
└─$ bash security_audit.sh
🔹 Running Security Audit Script...
---------------------------------
🔍 Checking User Login Attempts...
llamafar tty2                         Mon Mar 17 16:32 - 11:37 (213503982+03:06)
Debian-g tty1                         Mon Mar 17 16:32 - 16:32  (00:00)
Debian-g tty1                         Mon Mar 17 14:28 - 08:59 (213503982+02:32)
llamafar tty2                         Mon Mar 17 13:49 - still logged in
Debian-g tty1                         Mon Mar 17 13:49 - 13:49  (00:00)
llamafar tty2                         Mon Mar 17 08:59 - 10:05  (01:06)
llamafar ssh          ::1             Sun Mar 16 23:10 - 23:20  (00:09)
llamafar ssh          ::1             Sun Mar 16 23:05 - 23:20  (00:14)
llamafar tty2                         Sun Mar 16 19:16 - 23:46  (04:29)
Debian-g tty1                         Sun Mar 16 19:16 - 19:17  (00:00)
```

5. Mitigation

5.1 Automate System Monitoring

The script was scheduled to run periodically using cron.

Commands Used:

crontab -e

```
┌──(llamafart㉿jaivanti)-[~]
└─$ crontab -e
no crontab for llamafart - using an empty one
Select an editor.  To change later, run select-editor again.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /usr/bin/code

Choose 1-4 [1]: 1
No modification made
```

5.2 Implement Security Alerts

Email alerts were configured to notify the administrator of unauthorized SSH login attempts.

```
24 loaded units listed.
---------------------------------
🔍 Checking Disk Usage...
Filesystem      Size  Used Avail Use% Mounted on
udev            7.7G     0  7.7G   0% /dev
tmpfs           1.6G  1.9M  1.6G   1% /run
/dev/nvme0n1p5   55G   29G   24G  55% /
tmpfs           7.7G  4.0K  7.7G   1% /dev/shm
efivarfs        192K  118K   69K  64% /sys/firmware/efi/efivars
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           1.0M     0  1.0M   0% /run/credentials/systemd-journald.service
tmpfs           7.7G     0  7.7G   0% /tmp
/dev/nvme0n1p1  256M  105M  152M  41% /boot/efi
tmpfs           1.6G  128K  1.6G   1% /run/user/1000
---------------------------------
🔍 Checking Inactive Users (No Login in 30+ Days)...
sudo: lastlog: command not found
---------------------------------
security_audit.sh: line 42: mail: command not found
✅ Security Audit Completed!
```

6. Conclusion

This PoC successfully demonstrated how to create and execute a Bash script for automated security auditing. By automating system monitoring and implementing security alerts, potential security threats can be detected and mitigated promptly.