PROOF OF CONCEPT

LINUX SECURITY - EXPLOITATION & HARDENING

Task 3: Firewall & Network Security

1.EXECUTIVE SUMMARY

This PoC demonstrates security risks related to Linux user and permission misconfigurations. The task involves identifying misconfigurations, exploiting them to escalate privileges, and applying mitigation strategies to secure the system.

2. OBJECTIVES

Setup: Identify existing user and permission misconfigurations.

Exploitation: Utilize privilege escalation techniques to exploit weak permissions.

Mitigation: Implement security best practices to prevent unauthorized access.

3. SETUP

3.1. Install and Configure Apache Web Server

1. Update and Install Apache:

```
┌──(kali㉿kali)-[~]
└─$ sudo apt update && sudo apt install apache2 -y
[sudo] password for kali:
```

2. Start SSH and Apache:

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl start ssh

┌──(kali㉿kali)-[~]
└─$ sudo systemctl start apache2
```

3. Enable the Apache:

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-ins
tall.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
```

4. verify Apache Status:

```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: disabled)
     Active: active (running) since Tue 2025-03-25 17:36:49 IST; 36min ago
 Invocation: 78b4e25581e3466587439acc90dd91c8
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 125391 (apache2)
      Tasks: 6 (limit: 6798)
     Memory: 18.8M (peak: 19.3M)
        CPU: 339ms
     CGroup: /system.slice/apache2.service
             ├─125391 /usr/sbin/apache2 -k start
             ├─125394 /usr/sbin/apache2 -k start
             ├─125395 /usr/sbin/apache2 -k start
             ├─125396 /usr/sbin/apache2 -k start
             ├─125397 /usr/sbin/apache2 -k start
             └─125398 /usr/sbin/apache2 -k start

Mar 25 17:36:49 kali systemd[1]: Starting apache2.service - The Apache HTTP Server ...
Mar 25 17:36:49 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
```

5. Disable Firewall

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw disable
[sudo] password for kali:
Firewall stopped and disabled on system startup
```

4. EXPLOITATION

4.1 Scan for Open Ports using nmap

```
┌──(kali㉿kali)-[~]
└─$ nmap 127.0.1.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-25 18:15 IST
Nmap scan report for kali.kali (127.0.1.1)
Host is up (0.0000060s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

4.2 Access Web Server

```
┌──(kali㉿kali)-[~]
└─$ echo -en "GET / HTTP/1.1\r\nHost: 127.0.0.1\r\nConnection: close\r\n\r\n" | nc 127.0.0.1 80
HTTP/1.1 200 OK
Date: Tue, 25 Mar 2025 12:49:13 GMT
Server: Apache/2.4.63 (Debian)
Last-Modified: Mon, 17 Mar 2025 03:54:27 GMT
ETag: "29cf-63081bea94d79"
Accept-Ranges: bytes
Content-Length: 10703
Vary: Accept-Encoding
Connection: close
Content-Type: text/html
```

5. MITIGATION

5.1 Enable Firewall

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw enable
Firewall is active and enabled on system startup
```

5.2 Allow SSH and HTTP Traffic

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow ssh && sudo ufw allow http
Rule added
Rule added (v6)
Rule added
Rule added (v6)
```

5.3 Verify Firewall Rules

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
80/tcp                     ALLOW IN    Anywhere
22/tcp (v6)                ALLOW IN    Anywhere (v6)
80/tcp (v6)                ALLOW IN    Anywhere (v6)
```

5.4 Implement iptables Rules

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT

┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT

┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -j DROP
```

6. CONCLUSION

This PoC successfully demonstrated the security risks arising from user and permission misconfigurations. By enforcing stricter access controls and auditing user activities, the system was secured against unauthorized privilege escalation.

7. RECOMMENDATIONS

- Regularly review and restrict file and directory permissions.
- Remove unnecessary sudo privileges for non-admin users.
- Enable logging and auditing to monitor suspicious activities.