Proof Of Concept

Linux Security- Exploitation & Hardening

Task 3: Firewall & Network Security

1. Executive Summary

This PoC demonstrates the risks associated with improper firewall configurations and exposed network services. The task involves setting up a basic web server, scanning for open ports, and then hardening the system using ufw and iptables to restrict access and block unnecessary traffic.

2. Objectives

- **Setup:** Install and configure a basic web server (apache2) and disable the firewall (ufw disable).
- **Exploit:** Use nmap and netcat to scan for open ports and services, demonstrating how an attacker can discover exposed services.
- **Mitigation:** Restrict access using ufw (only allow SSH and HTTP) and implement iptables rules to block unnecessary traffic.

3. Setup

3.1. Install and Configure Apache Web Server

1. **Update and Install Apache**:

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

2. **Start SSH and Apache**:

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

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

3. **Enable and verify Apache Status**:

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
Created symlink '/etc/systemd/system/multi-user.target.wants/apache2.service' → '/usr/lib/systemd/system/apache2.service'.
```

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >
     Active: active (running) since Sun 2025-03-16 20:14:34 IST; 2min 9s ago
 Invocation: bd782fd7e678425b9a919657c6b51f64
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 7005 (apache2)
      Tasks: 7 (limit: 18730)
     Memory: 19.9M (peak: 20.6M)
        CPU: 85ms
     CGroup: /system.slice/apache2.service
             ├─7005 /usr/sbin/apache2 -k start
             ├─7008 /usr/sbin/apache2 -k start
             ├─7009 /usr/sbin/apache2 -k start
             ├─7010 /usr/sbin/apache2 -k start
             ├─7011 /usr/sbin/apache2 -k start
             ├─7012 /usr/sbin/apache2 -k start
             └─7013 /usr/sbin/apache2 -k start

Mar 16 20:14:34 jaivanti systemd[1]: Starting apache2.service - The Apache HTTP>
Mar 16 20:14:34 jaivanti apachectl[7004]: AH00558: apache2: Could not reliably >
Mar 16 20:14:34 jaivanti systemd[1]: Started apache2.service - The Apache HTTP >
```

3.2. Disable Firewall (UFW)

1. **Disable UFW**:

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo ufw disable
Firewall stopped and disabled on system startup
```

4. Exploitation

4.1. Scan for Open Ports Using Nmap

Scan the Local Machine:

```
┌──(llamafart㉿jaivanti)-[~]
└─$ nmap -A 192.168.29.149
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-16 20:25 IST
Nmap scan report for 192.168.29.149
Host is up (0.000019s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.9p1 Debian 3 (protocol 2.0)
| ssh-hostkey:
|   256 fd:41:27:c6:31:ea:72:ab:8e:38:f8:4f:f2:3d:d9:67 (ECDSA)
|_  256 ff:56:2b:d5:e5:89:16:f8:57:98:e7:59:c7:d2:68:f5 (ED25519)
80/tcp open  http    Apache httpd 2.4.63 ((Debian))
|_http-server-header: Apache/2.4.63 (Debian)
|_http-title: Apache2 Debian Default Page: It works
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.92 seconds
```

4.2. Use Netcat to Test Open Ports

```
┌──(llamafart㉿jaivanti)-[~]
└─$ nc -zv 192.168.29.149 1-1000
jaivanti [192.168.29.149] 80 (http) open
jaivanti [192.168.29.149] 22 (ssh) open
```

5. Mitigation

5.1. Enable and Configure UFW

1. **Enable UFW**:

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

2. **Set Default Policies**:

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)

┌──(llamafart㉿jaivanti)-[~]
└─$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

3. **Allow SSH and HTTP**:

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo ufw allow ssh
Rule added
Rule added (v6)

┌──(llamafart㉿jaivanti)-[~]
└─$ sudo ufw allow http
Rule added
Rule added (v6)
```

4. **Verify UFW Status**:

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo ufw status
Status: active

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

5.2. Implement IPTables Rules

1. **Set Default Policies**:

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo iptables -P INPUT DROP
[sudo] password for llamafart:

┌──(llamafart㉿jaivanti)-[~]
└─$ sudo iptables -P FORWARD DROP

┌──(llamafart㉿jaivanti)-[~]
└─$ sudo iptables -P OUTPUT ACCEPT
```

2. **Allow SSH and HTTP Traffic**:

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

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

3. **Allow Established Connections**:

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

4. **Save IPTables Rules**:

```
┌──(llamafart㉿jaivanti)-[~]
└─$ sudo iptables-save | sudo tee /etc/iptables/rules.v4
tee: /etc/iptables/rules.v4: No such file or directory
# Generated by iptables-save v1.8.10 (nf_tables) on Sun Mar 16 21:00:59 2025
*filter
:INPUT DROP [20:4432]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
:ufw-after-forward - [0:0]
:ufw-after-input - [0:0]
:ufw-after-logging-forward - [0:0]
:ufw-after-logging-input - [0:0]
:ufw-after-logging-output - [0:0]
```

```
-A ufw-not-local -m limit --limit 3/min --limit-burst 10 -j ufw-logging-deny
-A ufw-not-local -j DROP
-A ufw-skip-to-policy-forward -j DROP
-A ufw-skip-to-policy-input -j DROP
-A ufw-skip-to-policy-output -j ACCEPT
-A ufw-track-output -p tcp -m conntrack --ctstate NEW -j ACCEPT
-A ufw-track-output -p udp -m conntrack --ctstate NEW -j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 22 -j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 80 -j ACCEPT
-A ufw-user-limit -m limit --limit 3/min -j LOG --log-prefix "[UFW LIMIT BLOCK] "
-A ufw-user-limit -j REJECT --reject-with icmp-port-unreachable
-A ufw-user-limit-accept -j ACCEPT
COMMIT
```

Conclusion:

This PoC demonstrates the importance of proper firewall configuration and network security. By restricting access to essential services and blocking unnecessary traffic, system administrators can significantly reduce the attack surface and improve overall system security.