Proof of Concept (PoC) Report

Task 3: Firewall & Network Security

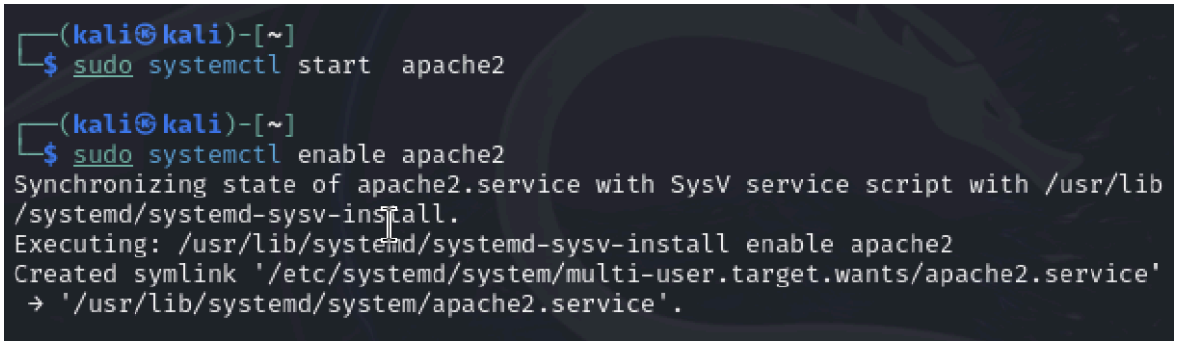
📌 Objective:
Demonstrate how an improperly configured firewall can expose services and how to implement network security measures to protect the system.

1️⃣ Setup: Installing & Configuring a Web Server with Weak Security

Step 1: Install Apache Web Server

Command:

- sudo apt update && sudo apt install apache2 -y(command for installing)
- Then we begin by starting and enabling apache2 and available to use



What Does It Do?

- Installs and enables the Apache web server.
- Starts the Apache service on boot.

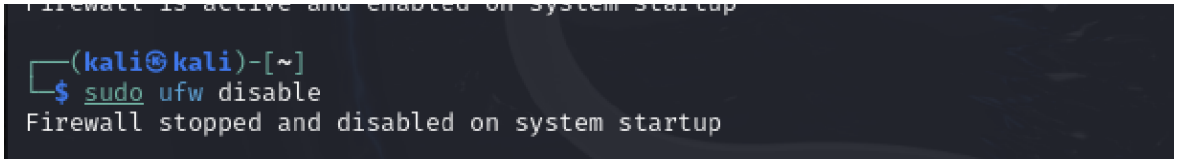
Step 2: Disable Firewall (Allow All Traffic)

Command:

- if the ufw tool is not installed use command

  (sudo apt update && sudo apt install ufw -y) then enable it

  -y:without asking yes or no prompt for installation



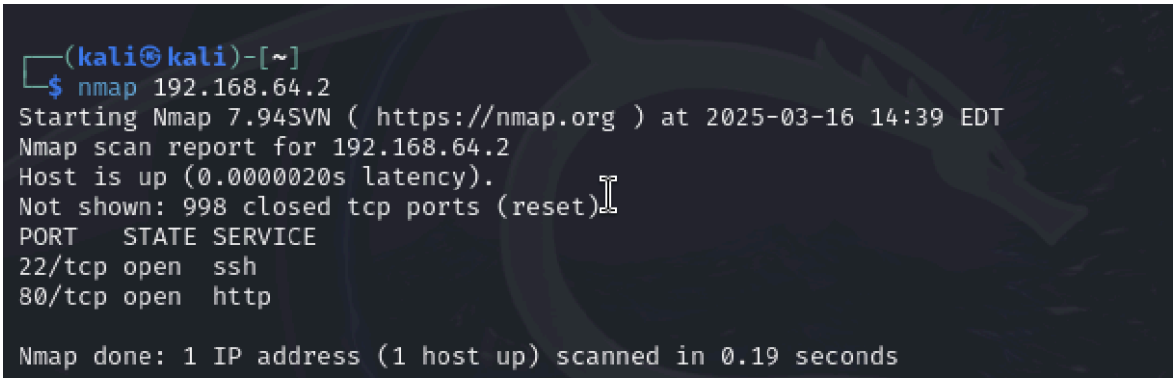
Security Risk:

- Without firewall protection, all ports remain open to attackers.
- Exposed services can be easily discovered and exploited.

2️⃣ Exploit: Scanning for Open Ports & Services

Step 3: Perform an Nmap Scan

Command:



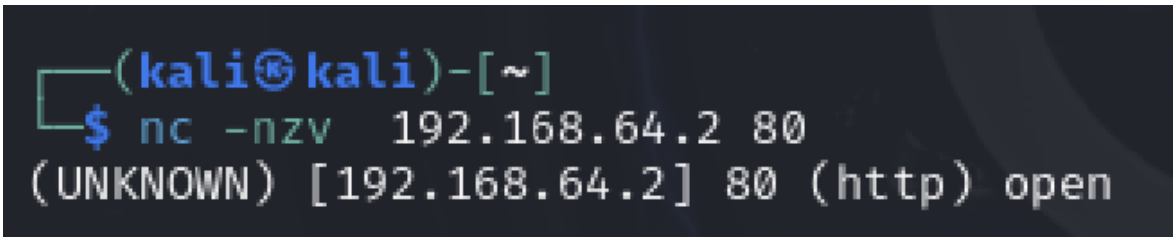
What Does It Do?

- Detects open ports and running services
- Identifies software versions and potential vulnerabilities.

🔴 Critical Risk: Attackers can discover and target these services.

Step 4: Use Netcat to Enumerate Services

Command:



Flag Breakdown

- **-n** → No DNS resolution (avoids delays from hostname lookups).
- **-z** → Zero-I/O mode (only checks if a port is open, without sending data).
- **-v** → Verbose mode (provides detailed output)

  What Does It Do?

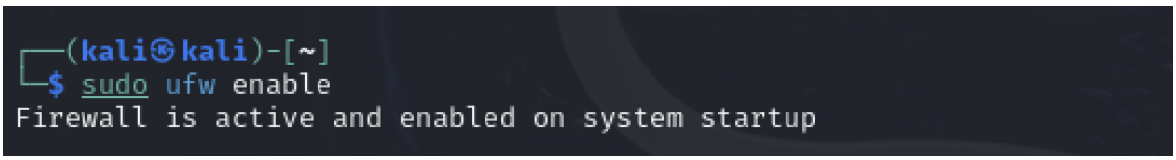
- Connects to the web server to check for service responses.
- Can be used to manually interact with exposed ports.

🔴 Issue: Attackers can send malicious requests to exploit vulnerabilities.

3️⃣ Mitigation: Securing the Network with a Firewall

Step 5: Enable UFW and Restrict Traffic

Command:(setting rules)



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

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

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

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

What Does It Do?

1. Enables firewall protection.
2. Allows only SSH (22) and HTTP (80) traffic.
3. Blocks all other unauthorized connections.

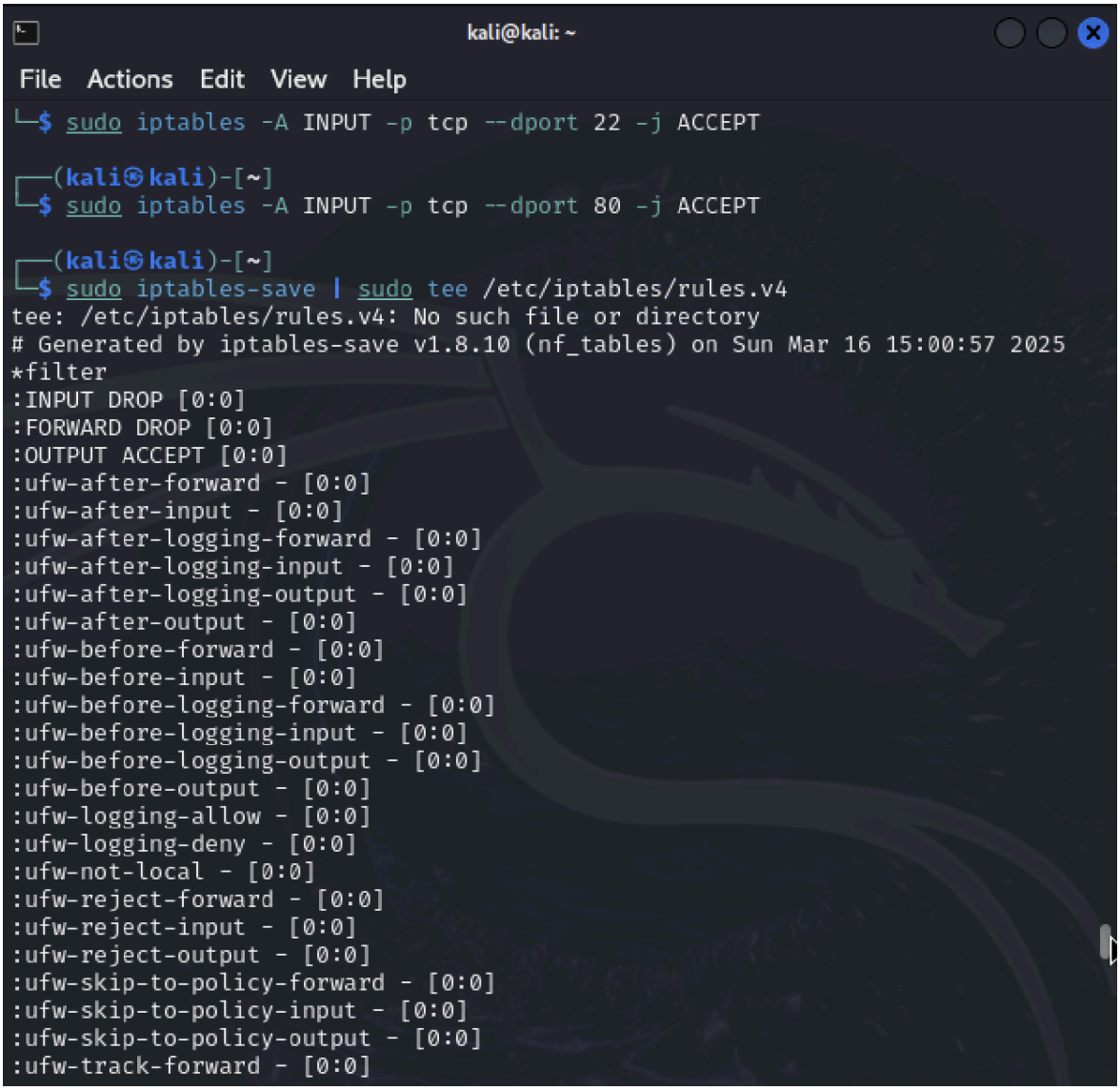
4. Reload the ufw

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw reload
Firewall reloaded
```

Step 6: Implement iptables Rules

Command:

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

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

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

┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -m --ctstate ESTABLISHED,RELATED -j ACEEPT
iptables v1.8.10 (nf_tables): Couldn't load match `--ctstate':No such file or
 directory

Try `iptables -h' or 'iptables --help' for more information.

┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -m conntrack  --ctstate ESTABLISHED,RELATED -j ACE
EPT
iptables v1.8.10 (nf_tables): Chain 'ACEEPT' does not exist
Try `iptables -h' or 'iptables --help' for more information.

┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -m conntrack  --ctstate ESTABLISHED,RELATED -j ACC
EPT

┌──(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-save | sudo tee /etc/iptables/rules.v4
```



What Does It Do?

1.Allows only SSH and HTTP traffic.
2.Drops all other incoming connections, preventing unauthorized access.

📝 Conclusion:

Exploitation: Demonstrated how an open firewall exposes services.

Mitigation: Applied firewall rules to limit access.

Outcome: System security enhanced, reducing risk of network attacks.

📍 Status: Fixed & Secured✅