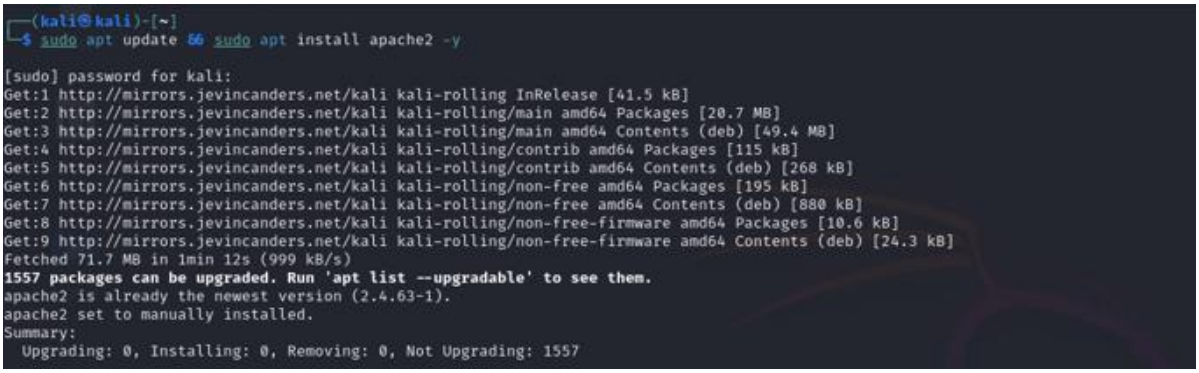
Task 3

Firewall & Network Security

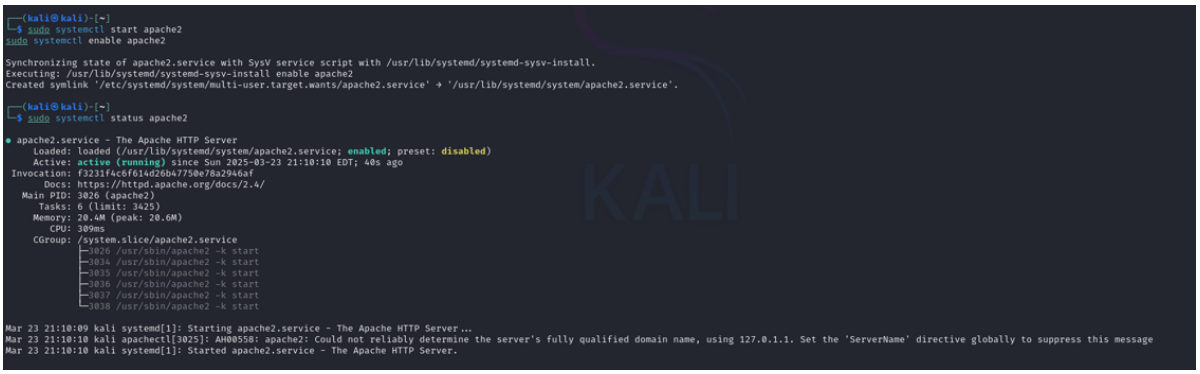
Setup

1. Install and configure a basic web server (apache2) and disable the firewall (ufw disable).



2. Setting up apache web server :

We begin by starting and enabling the Apache2 server to ensure it is active and available for use



3. Disable Firewall (UFW)

Install ufw then check **"inactive"**, it means the firewall is installed but not enabled

```
┌──(kali㉿kali)-[~]
└─$ sudo apt update && sudo apt install ufw -y

Hit:1 http://http.kali.org/kali kali-rolling InRelease
1557 packages can be upgraded. Run 'apt list --upgradable' to see them.
Installing:
  ufw

Suggested packages:
  rsyslog

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1557
  Download size: 169 kB
  Space needed: 880 kB / 62.9 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 ufw all 0.36.2-9 [169 kB]
Fetched 169 kB in 6s (28.8 kB/s)
Preconfiguring packages ...
Selecting previously unselected package ufw.
(Reading database ... 401537 files and directories currently installed.)
Preparing to unpack .../archives/ufw_0.36.2-9_all.deb ...
Unpacking ufw (0.36.2-9) ...
Setting up ufw (0.36.2-9) ...

Creating config file /etc/ufw/before.rules with new version

Creating config file /etc/ufw/before6.rules with new version

Creating config file /etc/ufw/after.rules with new version

Creating config file /etc/ufw/after6.rules with new version
update-rc.d: We have no instructions for the ufw init script.
update-rc.d: It looks like a non-network service, we enable it.
Created symlink '/etc/systemd/system/multi-user.target.wants/ufw.service' → '/usr/lib/systemd/system/ufw.service'.
Processing triggers for kali-menu (2024.4.0) ...
Processing triggers for man-db (2.13.0-1) ...
```

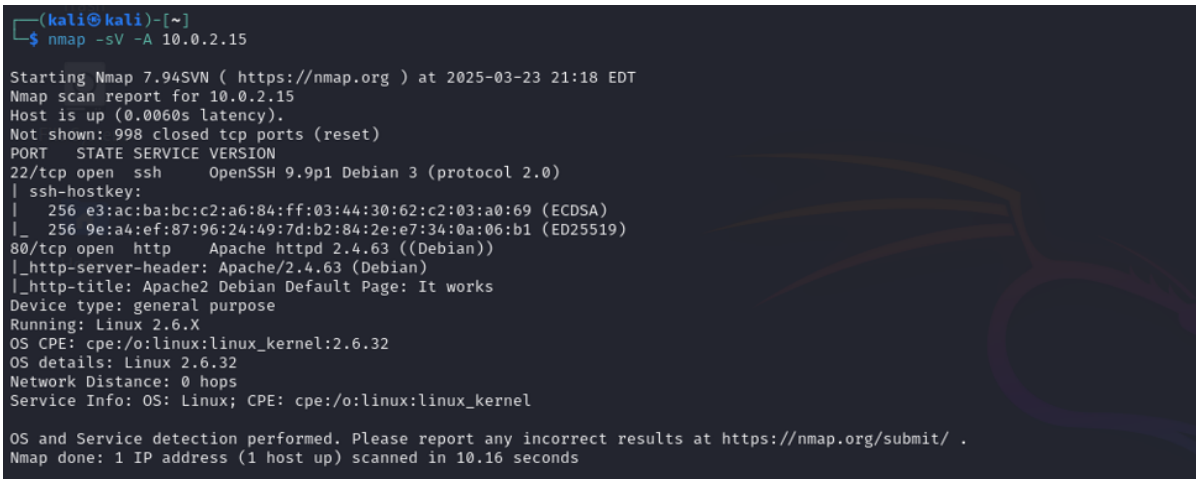
```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status

Status: inactive
```

Disable ufw

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw disable
```

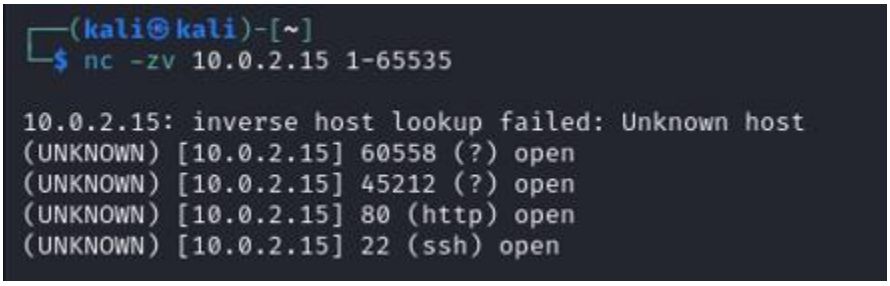
Exploit : port scanning with Nmap and Netcat

1. Perform a basic Nmap scan to check for open ports



2. Then we use Netcat to check if a specific port is open. Scans all possible ports on the server.

This helps an attacker find which ports are open and potentially vulnerable.



Mitigation: Firewall & Network Hardening

- Allows SSH connections (used for remote login).
- Allows HTTP traffic (used for websites).

After this, only SSH and HTTP will be accessible, blocking all other unnecessary services.

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow 22/tcp
sudo ufw allow 80/tcp

Rules updated
Rules updated (v6)
Rules updated
Rules updated (v6)
```

1. Turns on the firewall with the rules we just set. Any service that is **not explicitly allowed** will now be blocked.

2. First we enable firewall for active defense.

3. By checking status it will show that **only SSH (22) and HTTP (80) are allowed**, while everything else is blocked.

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw enable

Firewall is active and enabled on system startup

┌──(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)
```

4. To provide an extra layer of security, we can use iptables to block specific services.

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -p tcp --dport 23 -j DROP
sudo iptables -A INPUT -p tcp --dport 21 -j DROP
```

```
┌──(kali㉿kali)-[~]
└─$ sudo netstat -tulnp

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      709/sshd: /usr/sbin
tcp6       0      0 :::80                   :::*                    LISTEN      3026/apache2
tcp6       0      0 :::22                   :::*                    LISTEN      709/sshd: /usr/sbin
```

This shows a list of all open ports and the services running on them.

After enabling the firewall and iptables rules, run the same command again.Compare the results to show how many ports were blocked.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -A 10.0.2.15

Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-23 21:32 EDT
Nmap scan report for 10.0.2.15
Host is up (0.000098s 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 e3:ac:ba:bc:c2:a6:84:ff:03:44:30:62:c2:03:a0:69 (ECDSA)
|_  256 9e:a4:ef:87:96:24:49:7d:b2:84:2e:e7:34:0a:06:b1 (ED25519)
80/tcp open  http    Apache httpd 2.4.63 ((Debian))
|_http-title: Apache2 Debian Default Page: It works
|_http-server-header: Apache/2.4.63 (Debian)
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 8.38 seconds
```

Updated nmap & netstat results (showing only ports 22 & 80 open). Screenshot of sudo ufw status confirming restricted access.

```
┌──(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)
```