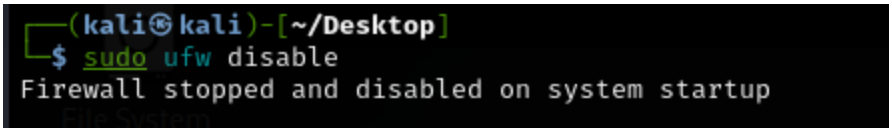
TASK 3:

Firewall & Network Security:

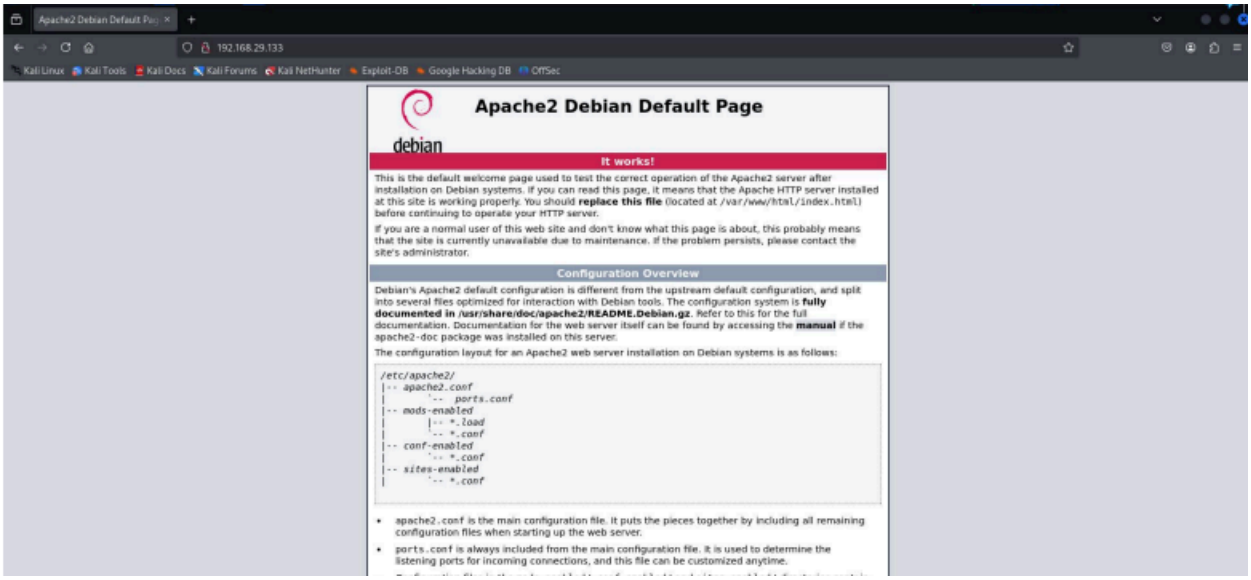
Setup:

1. We begin by starting and enabling apache2 server by sudo systemctl start apache2 and sudo systemctl enable apache2 to ensure it is active and available for use.

2. Then we disable the firewall by using `sudo ufw disable` command.



3. We initiate the apache2 server .



Exploit:

1. Now we perform a basic Nmap scan nmap to check for open ports.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap 192.168.29.133
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-25 08:23 EDT
Nmap scan report for 192.168.29.133
Host is up (0.0035s latency).
All 1000 scanned ports on 192.168.29.133 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 7.74 seconds
```

2. Then we use nc -nzv to check if any specific port is open.

```
┌──(kali㉿vbox)-[~/Desktop]
└─$ nc -nzv 192.168.29.133 80

(UNKNOWN) [192.168.29.133] 80 (http) open
```

Mitigation:

1. Enable firewall for defence by sudo ufw enable are used.

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

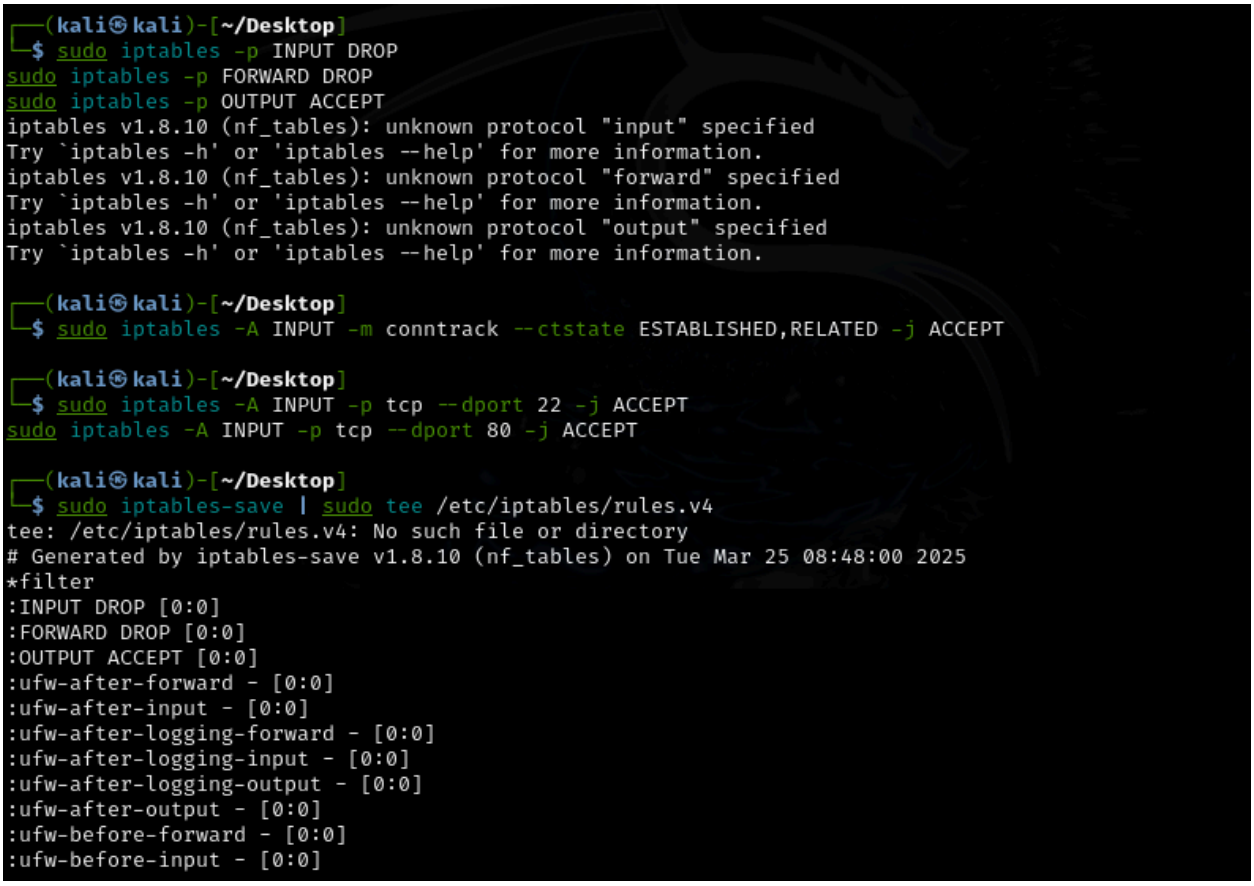
2. Rules for necessary ports are setted by sudo ufw default deny incoming, sudo ufw default allow outgoing, sudo ufw allow ssh and sudo ufw allow http are used.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow ssh
sudo ufw allow http
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
Skipping adding existing rule
Skipping adding existing rule (v6)
Skipping adding existing rule
Skipping adding existing rule (v6)
```

3. Atlast reload the firewall to update the changes made.

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

Iptables protection:



To configure iptables for enhanced security, set the default policies to drop incoming and forwarded traffic while accepting outgoing connections, allow established and related connections, permit incoming traffic for SSH (port 22) and HTTP(port 80), and finally save the rules using sudo iptables-save | sudo tee /etc/Iptables/rules.v4 to ensure they persist across reboots.