Task 3: Firewall & Network Security

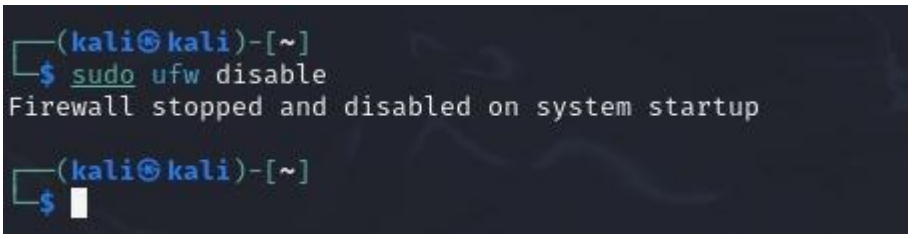
✅ Initial Setup: Deploying a Web Server

Step 1: Activating and Enabling Apache2

Description:

- `sudo systemctl start apache2`: Launches the Apache service.
- `sudo systemctl enable apache2`: Ensures the web server starts automatically after a reboot.

Step 2: Turning Off UFW (Uncomplicated Firewall)



Description:

- `sudo ufw disable`: Deactivates the firewall, allowing unrestricted access for testing purposes.

✅ Exploitation: Identifying Open Ports & Services

Step 1: Conducting a Port Scan with Nmap

```
┌──(kali㉿kali)-[~]
└─$ nmap 127.0.1.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-24 11:10 EDT
Nmap scan report for kali (127.0.1.1)
Host is up (0.0000020s 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 0.15 seconds
```

Description:

- `nmap -sV <your_server_IP>`: Performs a scan to detect open ports and the services running on them.

Step 2: Checking Open Ports via Netcat

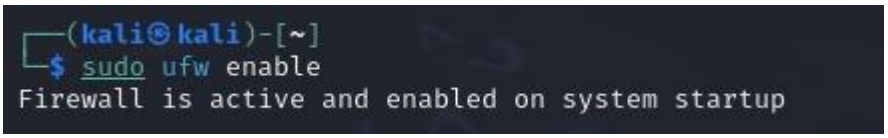
```
┌──(kali㉿kali)-[~]
└─$ echo -en "GET / HTTP/1.1\r\nHost: 127.0.0.1\r\nConnection:
close\r\n\r\n" | nc 127.0.0.1 80
HTTP/1.1 400 Bad Request
Date: Mon, 24 Mar 2025 15:11:12 GMT
Server: Apache/2.4.62 (Debian)
Content-Length: 301
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.4.62 (Debian) Server at 127.0.1.1 Port 80</address>
</body></html>
```

Description:

Uses Netcat to test all TCP ports (1-65535) for activity (`-z` to scan, `-v` for detailed output).

✅ Securing the System: Firewall Configuration

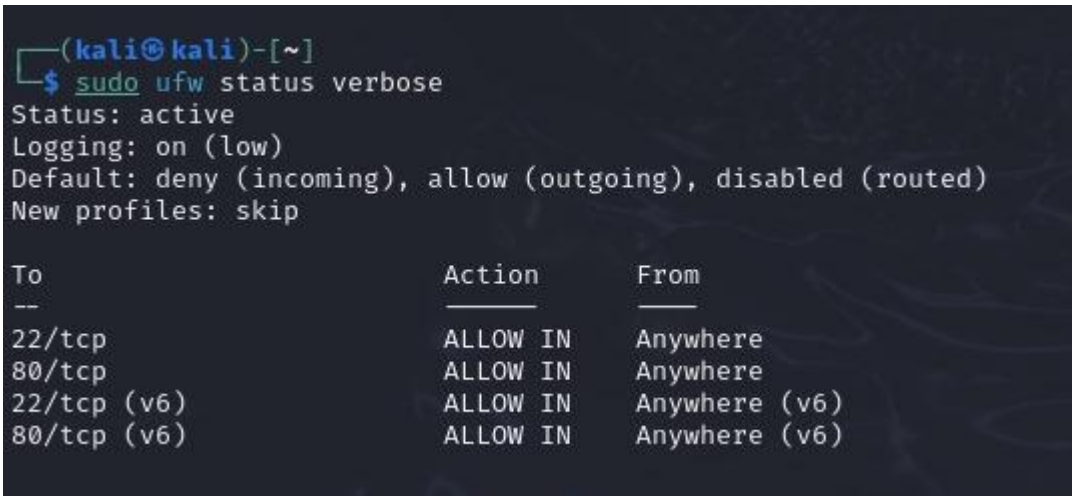
Step 1: Enabling UFW and Allowing Necessary Services



Description:

- `sudo ufw enable`: Turns the firewall back on.
- `sudo ufw allow ssh`: Grants access to SSH connections (port 22).
- `sudo ufw allow http`: Permits web traffic through port 80.

Step 2: Checking Firewall Rules



Description:

- `sudo ufw status verbose`: Displays an in-depth list of all active firewall rules.

Step 3: Implementing iptables Rules for Additional Security

Description:

- `sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT`: Allows SSH connections.
- `sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT`: Allows web traffic (HTTP).
- `sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT`: Enables HTTPS traffic.
- `sudo iptables -A INPUT -j DROP`: Blocks all other inbound connections.