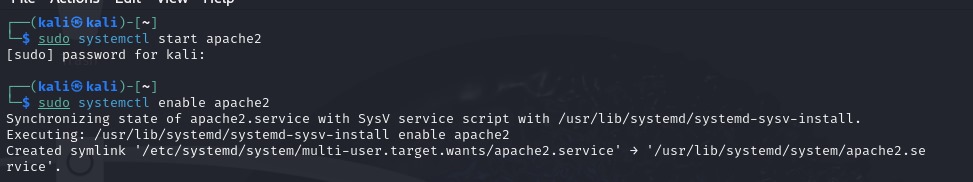
# **Task 3: Firewall & Network Security**

## **✅ Initial Setup: Deploying a Web Server**

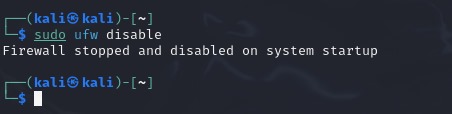
### **Step 1: Activating and Enabling Apache2**



**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**

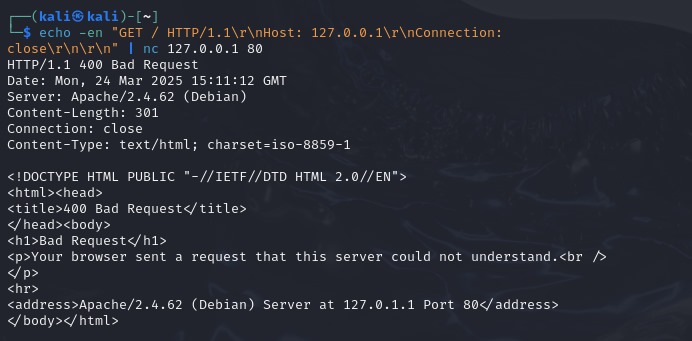
### **Step 1: Conducting a Port Scan with Nmap**



**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**

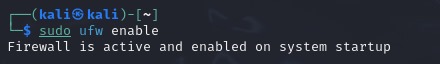


**Description:**

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**

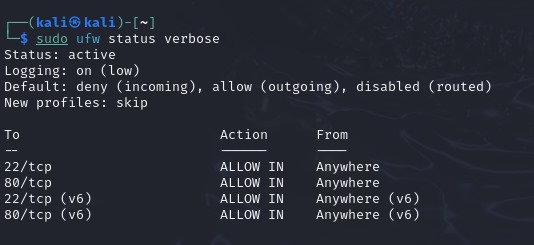




**Description:**

* 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.

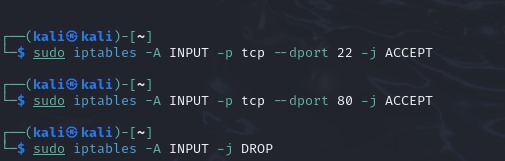
### **Step 2: Checking Firewall Rules**



**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.