**Proof of Concept (PoC) Report**

Task 3: Firewall & Network Security

# 1. Executive Summary

This PoC demonstrates the risks of having a web server exposed without proper firewall rules. The task involves setting up a basic web server, scanning for open ports, and mitigating risks by configuring a firewall to restrict access.

# 2. Objectives

* **Setup:** Install and configure a basic web server (Apache) and disable the firewall.
* **Exploit:** Use nmap and netcat to scan for open ports and services.
* **Mitigation:** Restrict access using ufw and implement iptables rules to block unnecessary traffic.

# 3. Setup

## 3.1 Install and Enable Apache

Apache was installed and enabled to start on boot using the systemctl command.

**Commands Used:**

sudo systemctl start apache2 sudo systemctl enable apache2

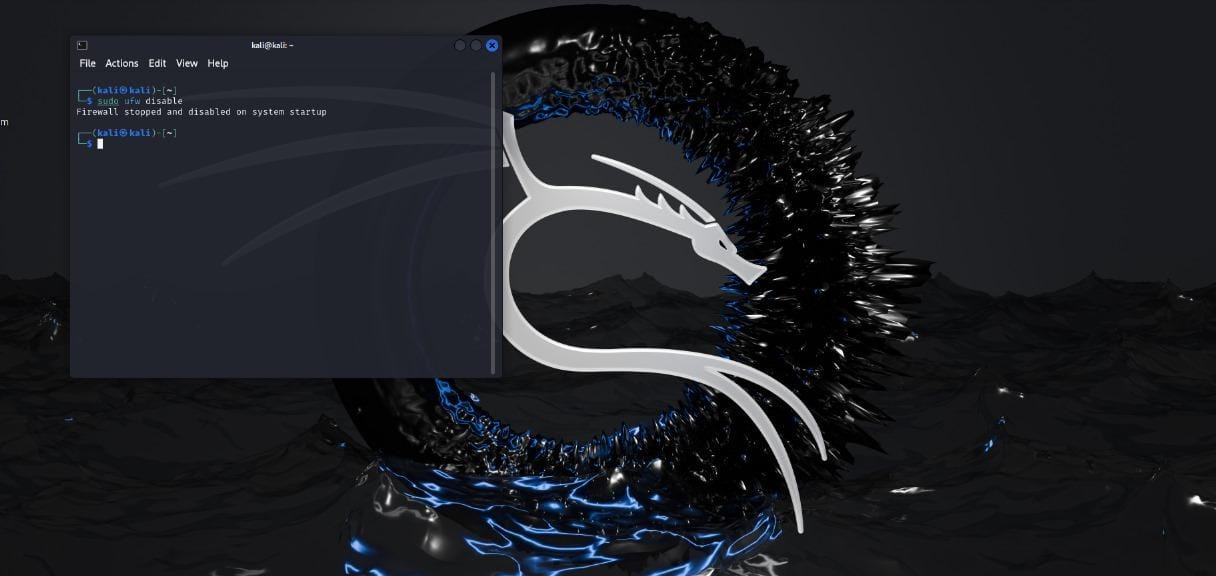


## 3.2 Disable Firewall

The firewall (ufw) was disabled to simulate an insecure environment.

**Commands Used:**

sudo ufw disable



## 4. Exploitation

### 4.1 Scan for Open Ports

nmap was used to scan the local machine for open ports. The scan revealed that ports 22 (SSH) and 80 (HTTP) were open.

**Commands Used:**

nmap 127.0.1.1

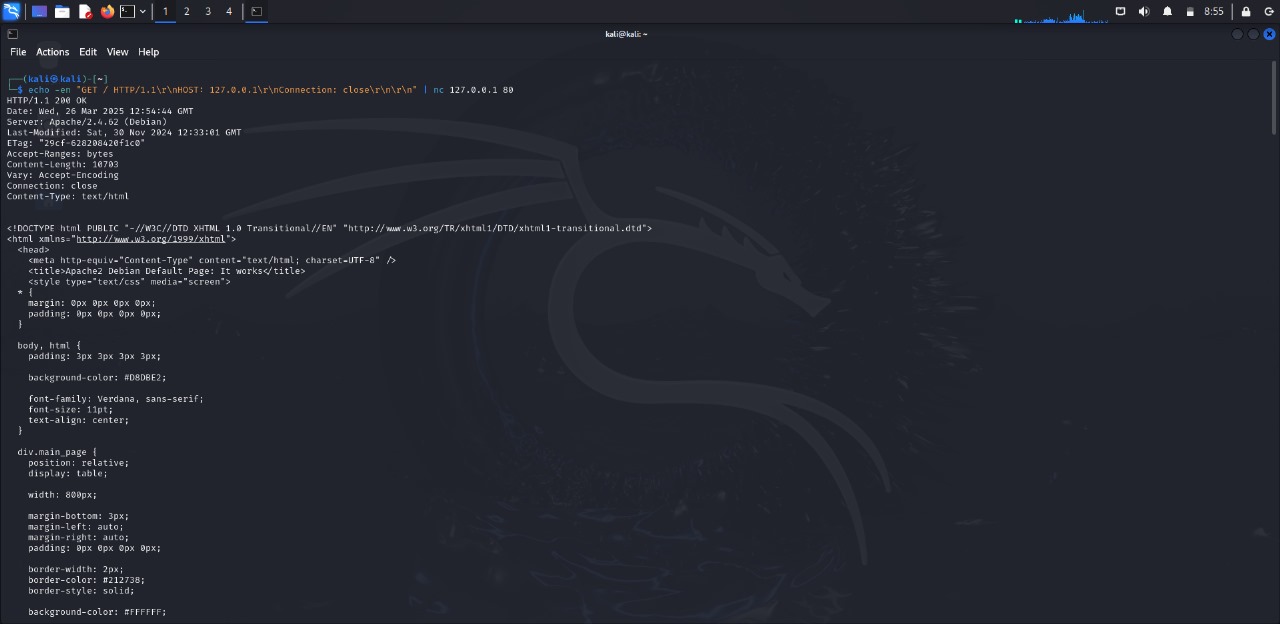


### 4.2 Access Web Server

netcat was used to send an HTTP request to the web server, confirming that it was accessible.

**Commands Used:** 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



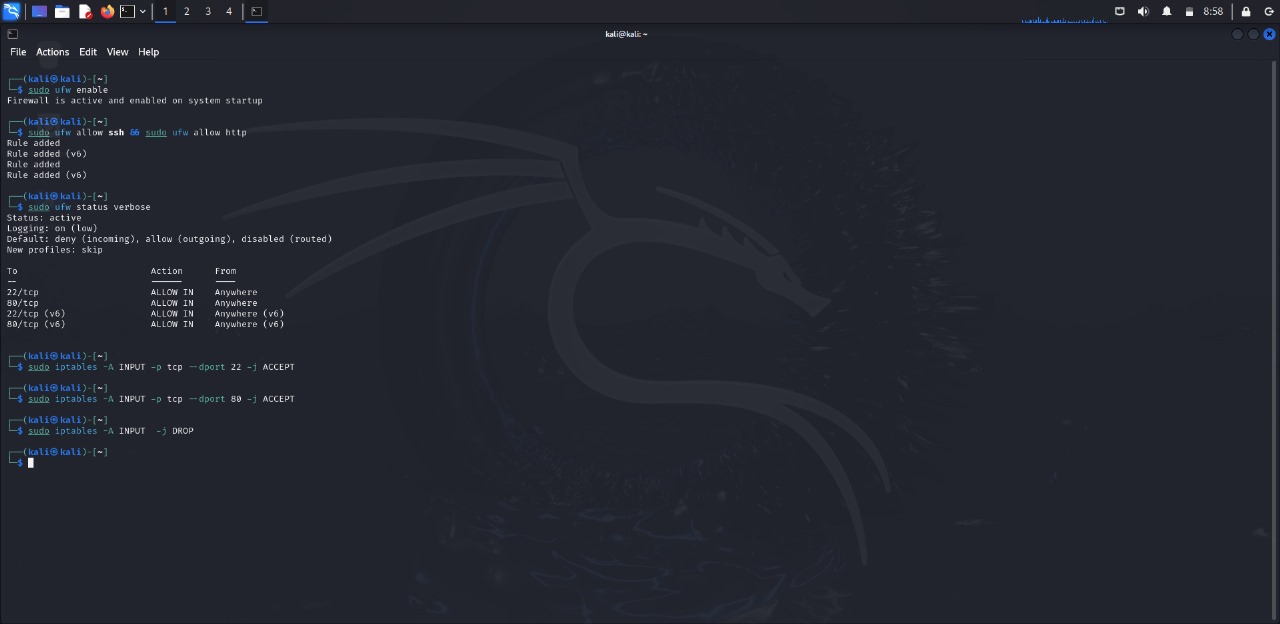
## 5. Mitigation

### 5.1 Enable Firewall

The firewall (ufw) was enabled to restrict incoming traffic.

**Commands Used:**

sudo ufw enable

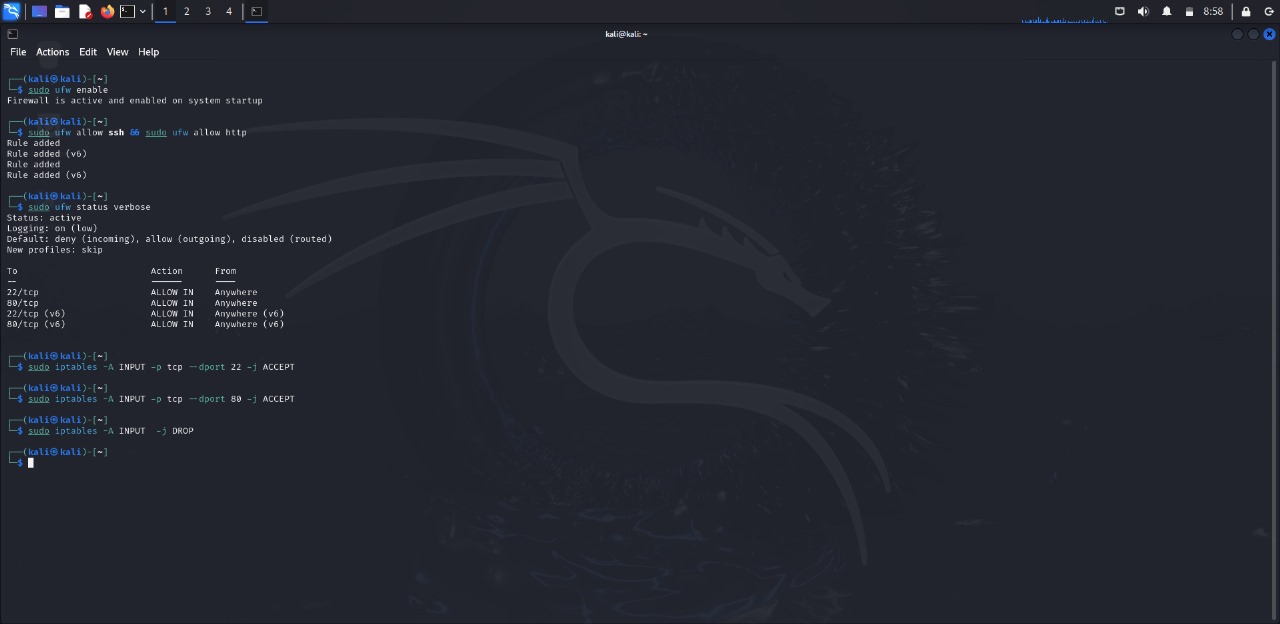


### 5.2 Allow SSH and HTTP Traffic

Firewall rules were updated to allow only SSH and HTTP traffic.

**Commands Used:**

sudo ufw allow ssh sudo ufw allow http

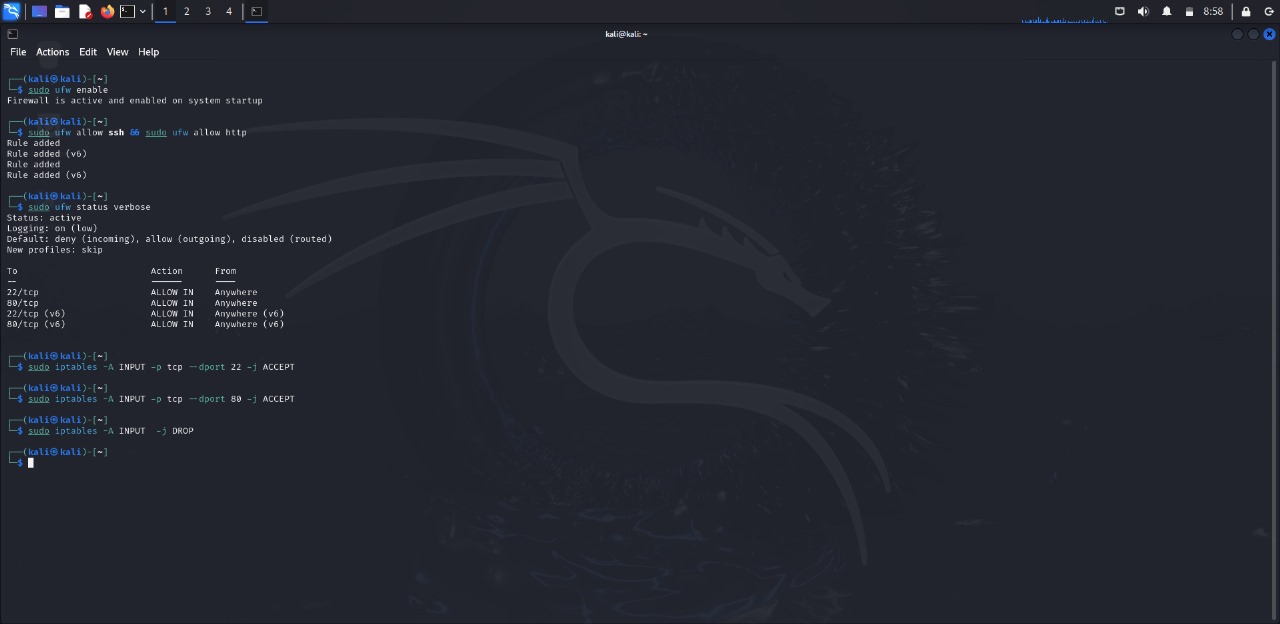


### 5.3 Verify Firewall Rules

The status of the firewall was checked to confirm that only SSH and HTTP traffic were allowed.

**Commands Used:**

sudo ufw status verbose

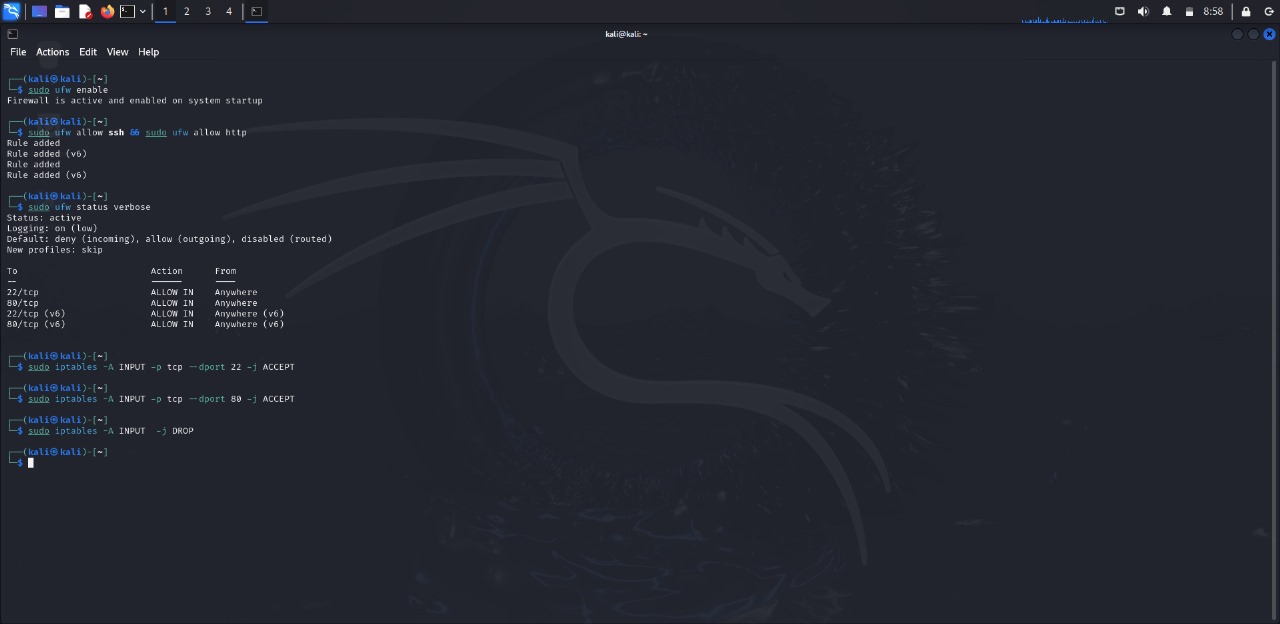


### 5.4 Implement iptables Rules

Additional iptables rules were added to explicitly allow SSH and HTTP traffic and drop all other incoming traffic.

**Commands Used:**

sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT sudo iptables -A INPUT -j DROP



## 6. Conclusion

This PoC successfully demonstrated the risks of exposing a web server without proper firewall rules. By enabling the firewall and restricting access to only necessary services, the server was secured against unauthorized access.

## 7. Recommendations

* Regularly review and update firewall rules to ensure only necessary ports are open.
* Use tools like fail2ban to block repeated failed login attempts.
* Monitor network traffic for suspicious activity.