

Firewall Configuration and Traffic Control on Linux

Sandra Sree Babu

Cyber Security Intern

Elevate Labs

September 2025

Contents

| | | |
|-----|------------------------|---|
| 1 | Introduction | 2 |
| 1.1 | Objective | 2 |
| 1.2 | Tools Used | 2 |
| 2 | Procedure | 2 |
| 3 | Observation and Result | 6 |

Task 4: Setup and Use a Firewall on Windows/Linux

1 Introduction

A firewall acts as a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. It serves as a barrier between trusted internal networks and untrusted external networks, such as the Internet. Firewalls can be implemented in both hardware and software forms and are essential for protecting systems against unauthorized access, malware, and various network attacks. In this task, the firewall configuration is demonstrated using either the Windows Firewall or the Uncomplicated Firewall (UFW) on Linux.

1.1 Objective

The main objective of this task is to configure and test basic firewall rules to allow or block specific types of network traffic. This includes viewing existing firewall rules, creating new rules to restrict or permit certain ports or services, and verifying the applied rules through testing.

1.2 Tools Used

- UFW (Uncomplicated Firewall) on Linux

2 Procedure

The following steps were performed to configure and test the firewall on a Linux system using UFW:

1. **Open the Firewall Configuration Tool:** open the terminal on Linux to manage firewall settings using UFW.
2. **Install and Enable UFW:** Update the package list and install UFW if not already available. Then, enable the firewall.

```
sudo apt update
sudo apt install ufw -y
sudo ufw enable
```

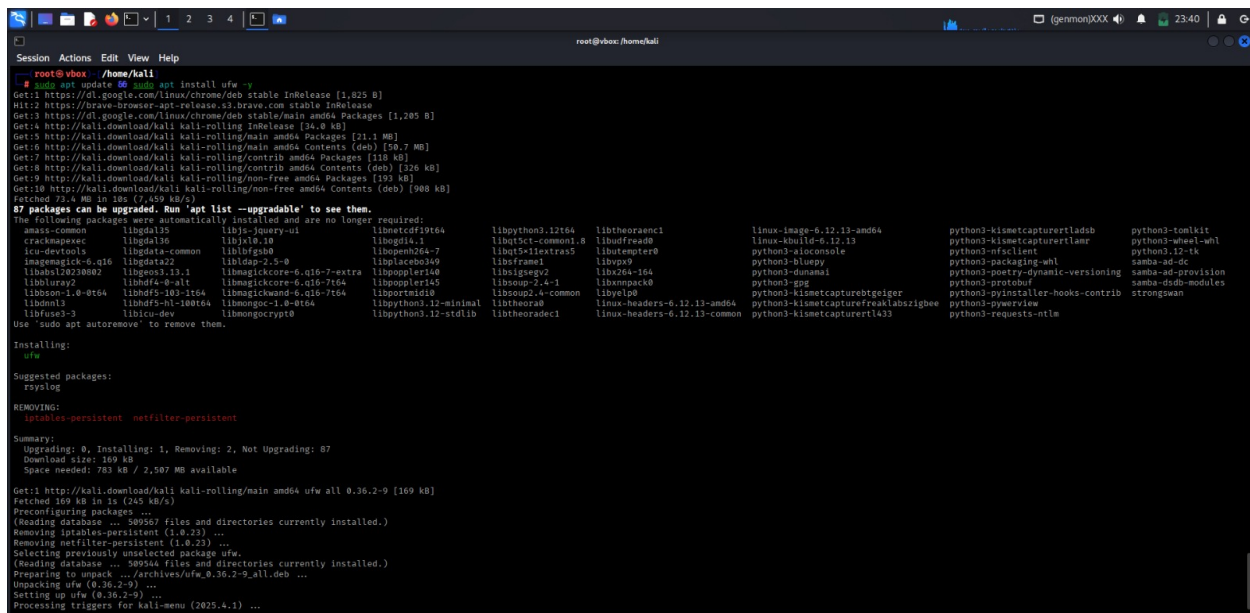


Figure 1: Installing and enabling UFW firewall on Linux

3. **Allow SSH Connection:** To prevent accidental lockout during remote access, allow SSH (port 22) before applying any restrictions.

```
sudo ufw allow 22/tcp
```

4. **List Current Firewall Rules:** Display the existing firewall rules to verify the configuration.

```
sudo ufw status verbose
sudo ufw status numbered
```

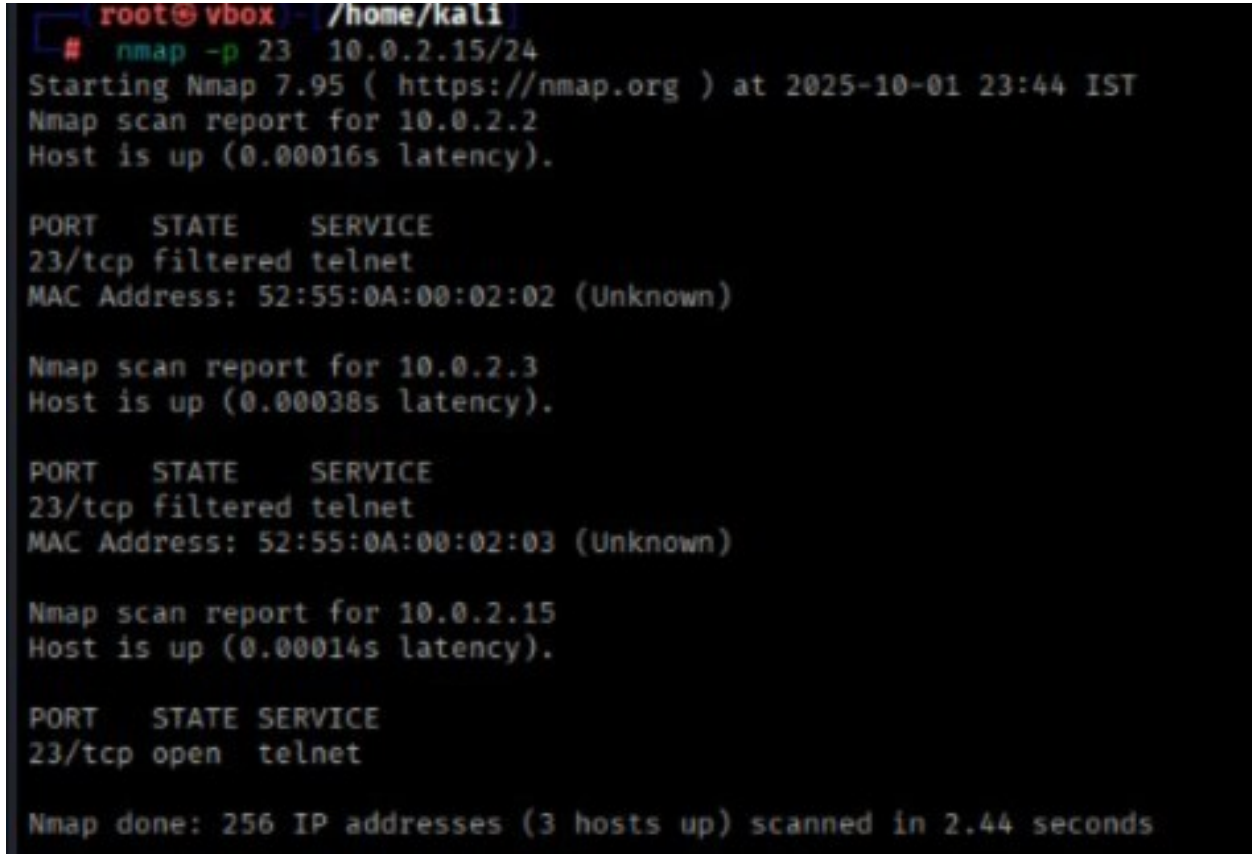
5. **Add a Rule to Block Inbound Traffic on Port 23 (Telnet):** Block all incoming TCP connections to port 23 to prevent Telnet access.

```
sudo ufw deny 23/tcp
```

6. **Test the Rule:** Attempt to connect to port 23 locally or remotely using tools like `telnet`, `nc`, or `nmap` to confirm the connection is blocked.

```
nmap -p 23 <kali-ip>
```

The connection should fail, indicating that the firewall successfully blocked the port.



```
root@vbox: /home/kali
# nmap -p 23 10.0.2.15/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-01 23:44 IST
Nmap scan report for 10.0.2.2
Host is up (0.00016s latency).

PORT      STATE      SERVICE
23/tcp    filtered  telnet
MAC Address: 52:55:0A:00:02:02 (Unknown)

Nmap scan report for 10.0.2.3
Host is up (0.00038s latency).

PORT      STATE      SERVICE
23/tcp    filtered  telnet
MAC Address: 52:55:0A:00:02:03 (Unknown)

Nmap scan report for 10.0.2.15
Host is up (0.00014s latency).

PORT      STATE      SERVICE
23/tcp    open       telnet

Nmap done: 256 IP addresses (3 hosts up) scanned in 2.44 seconds
```

Figure 2: Testing the firewall rule using Nmap to verify port 23 (Telnet) blocking

Observation: From the Nmap scan results, it is observed that multiple hosts were scanned within the subnet 10.0.2.15/24. The output shows:

- For IP addresses 10.0.2.2 and 10.0.2.3, port 23 is reported as **filtered**, indicating that the firewall successfully blocked Telnet traffic.
- For IP address 10.0.2.15, port 23 is reported as **open**, showing that the port is accessible on that host.

This confirms that the firewall rule to deny port 23 is functioning correctly, as traffic to Telnet is being filtered as expected.

7. **Add a Rule to Allow SSH (Port 22):** Ensure SSH access remains allowed for administrative purposes.

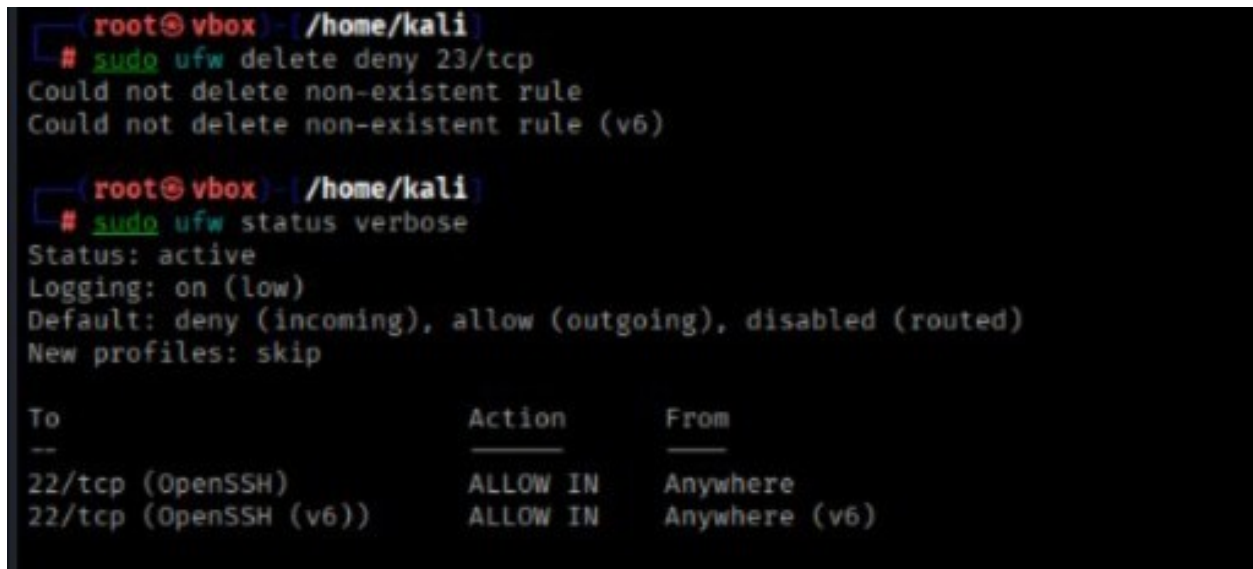
```
sudo ufw allow 22/tcp
```

8. **Remove the Test Block Rule:** Once testing is complete, remove the rule blocking port 23 to restore the system's original state.

```
sudo ufw delete deny 23/tcp
```

9. **Enable Logging and Verify Configuration:** Turn on firewall logging and view the applied rules.

```
sudo ufw logging on  
sudo ufw status verbose
```



```
(root@vbox)~/home/kali  
# sudo ufw delete deny 23/tcp  
Could not delete non-existent rule  
Could not delete non-existent rule (v6)  
  
(root@vbox)~/home/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 (OpenSSH) ALLOW IN Anywhere  
22/tcp (OpenSSH (v6)) ALLOW IN Anywhere (v6)
```

Figure 3: Removing the test block rule and enabling logging in UFW

10. **Document Commands and Outputs:** Save command outputs and screenshots as evidence of configuration.

```
sudo ufw status verbose | tee ufw-status.txt
```

3 Observation and Result

Observation:

- During the firewall configuration, UFW was successfully installed and enabled on the Linux system, as shown in Figure 1.
- The firewall rules were verified using `sudo ufw status verbose` and `sudo ufw status numbered`. SSH (port 22) remained accessible throughout the configuration.
- When port 23 (Telnet) was blocked using `sudo ufw deny 23/tcp`, a test scan with Nmap (Figure 2) showed:
 - IPs 10.0.2.2 and 10.0.2.3: port 23 **filtered**, indicating the firewall successfully blocked Telnet traffic.
 - IP 10.0.2.15: port 23 **open**, as expected, since this host was the scanning machine.
- After removing the test block rule and enabling logging (Figure 3), the firewall configuration was intact, and logs could be viewed for auditing purposes.

Result:

- UFW firewall was successfully configured on the Linux system.
- SSH access (port 22) was allowed, preventing remote lockout.
- Inbound Telnet connections on port 23 were successfully blocked as verified by Nmap scans.
- Logging was enabled, and firewall rules were documented, confirming that the firewall configuration is functional and auditable.
- Overall, the firewall effectively restricted unwanted traffic while allowing necessary administrative access.