✅ Task 4: Setup and Use a Firewall on Linux Using UFW

🔧 Objective:

Configure and test basic firewall rules to allow or block traffic.

🛠️ Tools:

- UFW (Uncomplicated Firewall) on Linux

📁 Deliverables:

- Screenshots of terminal with commands and outputs
- Summary of commands used
- Brief explanation of how firewall filters traffic

🔍 Task Overview:

This report documents the process of configuring and testing basic firewall rules using **UFW on Linux**. The goal was to allow or block specific network traffic, verify rule application, and understand how firewalls help in securing systems.

🧩 Steps with Commands:

1. Open UFW (Firewall Configuration Tool)

Make sure UFW is installed:

(There's no need to use the “sudo” command if you are already logged in as the root user.)

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

```
┌──(root㉿kali)-[~]
└─# sudo apt update
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [51.4 MB]
Get:4 http://kali.download/kali kali-rolling/non-free amd64 Packages [197 kB]
Get:5 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [911 kB]
Get:6 http://kali.download/kali kali-rolling/contrib amd64 Packages [120 kB]
Get:7 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [327 kB]
Fetched 74.0 MB in 14s (5271 kB/s)
1431 packages can be upgraded. Run 'apt list --upgradable' to see them.

┌──(root㉿kali)-[~]
└─# apt install ufw -y
Installing:
  ufw

Suggested packages:
  rsyslog

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1431
  Download size: 169 kB
  Space needed: 880 kB / 4299 MB available

Get:1 http://mirror.aktkn.sg/kali kali-rolling/main amd64 ufw all 0.36.2-9 [169 kB]
Fetched 169 kB in 1s (120 kB/s)
Preconfiguring packages ...
Selecting previously unselected package ufw.
(Reading database ... 434730 files and directories currently installed.)
Preparing to unpack .../archives/ufw_0.36.2-9_all.deb ...
Unpacking ufw (0.36.2-9) ...
Setting up ufw (0.36.2-9) ...
Creating config file /etc/ufw/before.rules with new version
Creating config file /etc/ufw/before6.rules with new version
Creating config file /etc/ufw/after.rules with new version
Creating config file /etc/ufw/after6.rules with new version
Update-rc.d: We have no instructions for the ufw init script.
Update-rc.d: It looks like a non-network service, we enable it.
Created symlink '/etc/systemd/system/multi-user.target.wants/ufw.service' → '/usr/lib/systemd
/system/ufw.service'.
Processing triggers for kali-menu (2025.1.1) ...
Processing triggers for man-db (2.13.0-1) ...
```

Enable UFW if it's not already enabled:

sudo ufw enable

```
┌──(root㉿kali)-[~]
└─# ufw enable
Firewall is active and enabled on system startup
```

---

2. List Current Firewall Rules

sudo ufw status verbose

```
┌──(root㉿kali)-[~]
└─# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
```

---

3. Add Rule to Block Inbound Traffic on Port 23 (Telnet)

sudo ufw deny 23

```
└─# ufw deny 23
Rule added
Rule added (v6)
```

---

4. Test the Rule

Try to connect to port 23 using telnet (or simulate a scan with nmap):

telnet localhost 23

or

nmap -p 23 localhost

```
┌──(root㉿kali)-[~]
└─# nmap -p 23 localhost
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-27 13:16 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000074s latency).
Other addresses for localhost (not scanned): ::1

PORT   STATE  SERVICE
23/tcp closed telnet

Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds
```

You should see **connection refused** or **filtered**.

🔍 What Does "Closed" Mean?

If nmap or telnet shows:

- **"Port 23 is closed"** → it means:
  - There is **no service (e.g., Telnet)** running on that port
  - AND your firewall is not explicitly **blocking** it
  - Instead, your system is just **not accepting** connections to it (passively rejecting)

- Since there was no Telnet service running on port 23, the system showed the port as **closed** even before the firewall rule.
- After applying the firewall rule to deny port 23, the port status remained closed — but now it is explicitly filtered by the firewall (even though no service listens on it).
- This confirms that the firewall rule was applied but the port was already non-responsive

---

5. Add Rule to Allow SSH (Port 22)

sudo ufw allow 22

```
┌──(root㉿kali)-[~]
└─# ufw allow 22
Rule added
Rule added (v6)
```

---

6. Remove the Test Block Rule (Restore State)

sudo ufw delete deny 23

```
┌──(root㉿kali)-[~]
└─# ufw delete deny 23
Rule deleted
Rule deleted (v6)
```

---

7. Documented Commands Used

 # UFW Installation & Setup

sudo apt update

sudo apt install ufw -y

sudo ufw enable

 # Rule Management

sudo ufw status verbose

sudo ufw deny 23

nmap -p 23 localhost

sudo ufw allow 22

sudo ufw delete deny 23

8. Summary: How Firewall Filters Traffic

UFW works by defining rules that either **allow** or **deny** network traffic based on port numbers, IP addresses, or services.

- **Inbound traffic** can be controlled by allowing or denying access to specific ports.
- UFW modifies **iptables** under the hood, making it easy to manage with simple commands.
- For example, blocking port 23 stops remote systems from connecting via Telnet, improving system security.

✅ **Outcome:**

- Listing existing firewall rules
- Creating new allow/deny rules
- Testing rule enforcement
- Removing temporary rules
- Understanding how traffic filtering works on Linux

✅ **Recommendations:**

1. **Always enable only necessary ports** to minimize the attack surface.
2. **Regularly audit firewall rules** to ensure they align with current security policies.
3. **Use logging options** in the firewall to monitor suspicious or repeated blocked attempts.
4. **Combine firewall rules with other security mechanisms** such as intrusion detection systems (IDS) for layered protection.