Task 4 – Elevate Labs Cyber Security Internship

Ahmad Ghtraieb

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

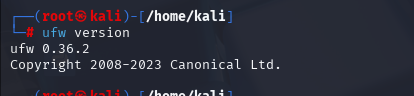
For this task, I used ufw from kali linux on a virtual machine, **UFW** (Uncomplicated Firewall) is a user-friendly, command-line interface for managing iptables rules on Linux. It simplifies the process of configuring a firewall by providing an easier way to allow or deny network traffic, making it simpler for users to secure their systems.

**Step 1: Open firewall configuration tool**

On Linux (Kali in my case), the tool controlled via the terminal.  
First, check if UFW is installed and enabled:

sudo apt update && sudo apt install ufw -y # install if not installed

here we can see ufw already installed with its version number



sudo ufw status # check status, shows whether the firewall is active.



sudo ufw enable # enable the firewall if inactive, (rules apply after this).

A screenshot of a computer program

AI-generated content may be incorrect.

**Step 2: List current firewall rules**

sudo ufw status numbered

A screen shot of a computer

AI-generated content may be incorrect.

shows active rules with numbers, which makes it easier to delete/edit them later. We can see no rules are set now.

**Step 3: Add a rule to block inbound traffic on a specific port (Telnet = 23)**

sudo ufw deny 23/tcp

A computer screen shot of a computer

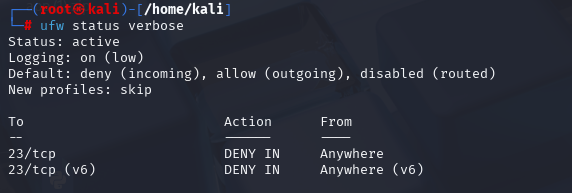
AI-generated content may be incorrect.

**Explanation**:

* deny → blocks traffic.
* 23/tcp → blocks TCP connections on port 23 (Telnet).

You can confirm by:

sudo ufw status numbered as mentioned earlier of ufw status verbose



**Step 4: Test the rule**

From the same machine or another on the network(here I used the same machine from a different terminal), try to connect to port 23:

nc -v 127.0.0.1 23

A close up of a computer screen

AI-generated content may be incorrect.

**Explanation**:

* nc (netcat) is a simple way to test open/closed ports.
* If blocked, you’ll see *Connection refused* or timeout.

**Step 5: Allow SSH (port 22)**

sudo ufw allow 22/tcp A screenshot of a computer program

AI-generated content may be incorrect.

**Explanation**:

* allow → permits traffic.
* 22/tcp → enables SSH so you don’t accidentally lock yourself out.
* We confirm the rule using status verbose

**Step 6: Remove the test block rule (restore state)**

List rules again:

sudo ufw status numbered

A screenshot of a computer

AI-generated content may be incorrect.

the Telnet(at port 23) is rule #1 and #3. Remove it with:

sudo ufw delete 1

A screenshot of a computer program

AI-generated content may be incorrect.

deleting by number is safer and clearer.

After the deletion we confirm it again

**Commands used:**

1. sudo ufw status

2. sudo ufw deny 23/tcp

3. nc -v 127.0.0.1 23

4. sudo ufw allow 22/tcp

5. sudo ufw status numbered

6. sudo ufw delete <rule-number>

**Summarize how firewall filters traffic**

A firewall (like UFW) works as a **traffic filter**.

* It examines incoming/outgoing packets.
* Rules decide whether to **allow** or **deny** them based on criteria like **port, protocol, IP address**.
* Example: Block Telnet (insecure) on port 23 but allow SSH (secure) on port 22.
* This helps prevent **unauthorized access** and reduces the system’s **attack surface**.