# Firewall Configuration and Traffic Filtering Report

**Platform:** Linux (UFW - Uncomplicated Firewall)

---

## Steps and Commands Used

1. **Open Firewall Configuration Tool**

- Opened the terminal and ensured UFW is installed and enabled:

```
sudo apt install ufw
sudo ufw enable
```

2. **List Current Firewall Rules**

- Command to display existing rules:

```
sudo ufw status numbered
```

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

- Block incoming traffic on port 23:

```
sudo ufw deny 23
```

4. **Test the Rule**

- Used `telnet` or `nc` (netcat) locally or from a remote system to confirm port 23 is blocked:

```
telnet <IP_address> 23
          or
nc -vz <IP_address> 23
```

5. **Allow SSH Traffic on Port 22**

- Ensured SSH access is not blocked:

```
sudo ufw allow 22
```

6. **Remove the Test Block Rule**

- First, identify rule number for port 23 using:

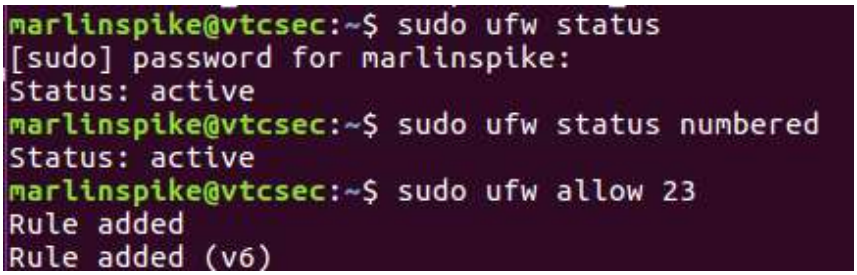
```
sudo ufw status numbered
```

- Then delete the rule (example: if it was rule #3):

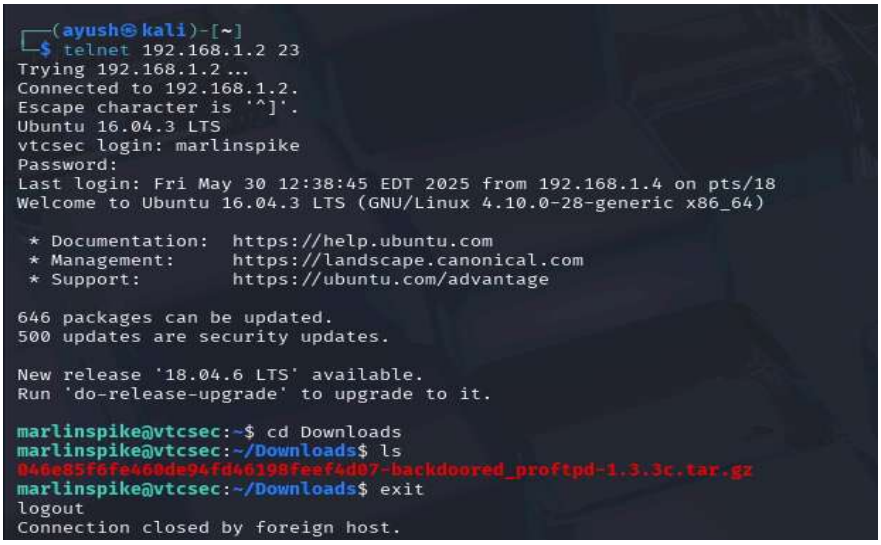
```
sudo ufw delete 3
```

## Screenshots Of Practicals

**Victim Machine:**

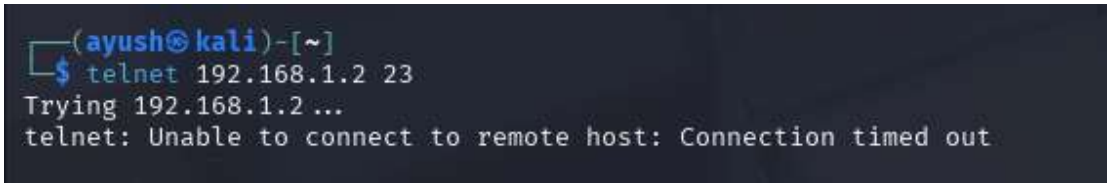



**Attacker Machine:**

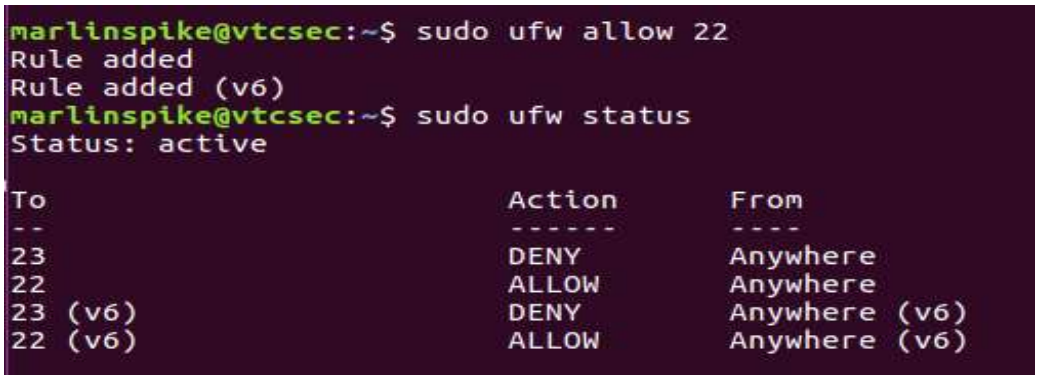


**Victim Machine:**

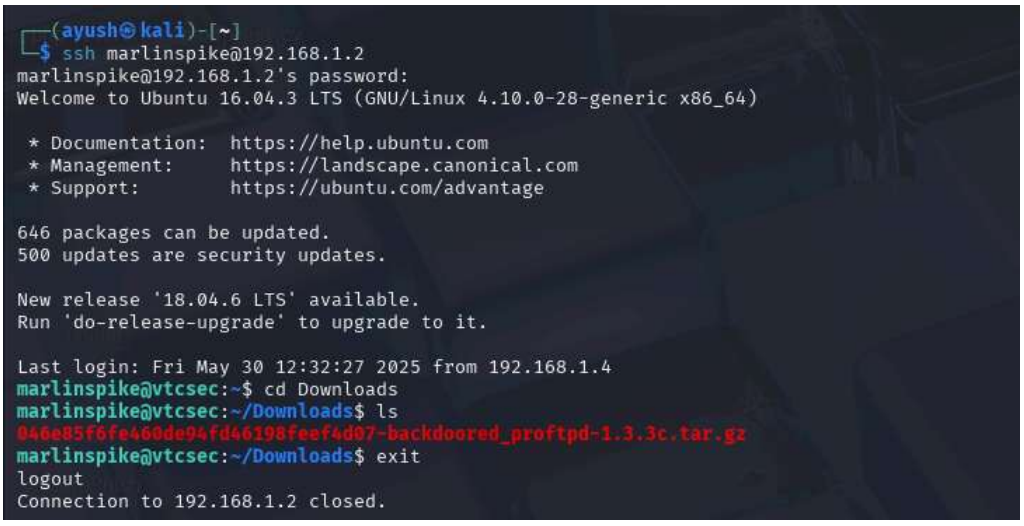

**Attacker Machine:**




**Victim Machine:**




**Attacker Machine:**



**Victim Machine:**

```
marlinspike@vtcsec:~$ sudo ufw deny 22
Rule updated
Rule updated (v6)
marlinspike@vtcsec:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
23                         DENY        Anywhere
22                         DENY        Anywhere
23 (v6)                    DENY        Anywhere (v6)
22 (v6)                    DENY        Anywhere (v6)
```

**Attacker Machine:**

```
┌──(ayush㉿kali)-[~]
└─$ ssh marlinspike@192.168.1.2
ssh: connect to host 192.168.1.2 port 22: Connection timed out
```

**Victim Machine:**

```
marlinspike@vtcsec:~$ sudo ufw reset
Resetting all rules to installed defaults. Proceed with operation (y|n)? y
Backing up 'before.rules' to '/etc/ufw/before.rules.20250530_135757'
Backing up 'before6.rules' to '/etc/ufw/before6.rules.20250530_135757'
Backing up 'user6.rules' to '/etc/ufw/user6.rules.20250530_135757'
Backing up 'after6.rules' to '/etc/ufw/after6.rules.20250530_135757'
Backing up 'user.rules' to '/etc/ufw/user.rules.20250530_135757'
Backing up 'after.rules' to '/etc/ufw/after.rules.20250530_135757'

marlinspike@vtcsec:~$ sudo ufw status
Status: inactive
marlinspike@vtcsec:~$
```

## How UFW Firewall Filters Traffic

UFW (Uncomplicated Firewall) acts as a front-end for iptables, providing a simplified interface to manage firewall rules. It filters network traffic based on:

- **Default Policies:** By default, UFW denies all incoming connections and allows all outgoing connections, minimizing exposure to threats.
- **Rule-Based Filtering:** Custom rules allow or deny traffic based on port numbers, protocols, and IP addresses. For example, you can block all traffic to port 23 or allow only SSH on port 22.
- **Direction Control:** Rules can be set for inbound or outbound traffic.
- **Stateful Inspection:** UFW tracks connection states to ensure only legitimate, established connections are allowed.

### In summary:

UFW firewall controls access to your Linux system by allowing or denying network connections according to user-defined rules. This protects the server from unauthorized access and restricts exposure to only necessary services, thereby enhancing overall security.