



**Veermata Jijabai Technological Institute, Mumbai 400019**

**Experiment No.: 04**

**Aim:** Practical Experiment on Firewall

**Name:** Kiran K Patil

**Enrolment No.:** 211070904

**Branch:** Computer Engineering

**Batch:** D

## Theory:

### Firewall

A firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.

Firewalls have been a first line of defense in network security for over 25 years. They establish a barrier between secured and controlled internal networks that can be trusted and untrusted outside networks, such as the Internet.

A firewall can be hardware, software, software-as-a service (SaaS), public cloud, or private cloud (virtual).

#### 1. Firewall Software:

- **Firewall Functionality:** Firewalls act as a barrier between your network and external networks, controlling incoming and outgoing traffic based on a set of predefined security rules.
- **Packet Filtering:** Most firewalls use packet filtering to allow or block traffic based on IP addresses, ports, and protocols.
- **Stateful Inspection:** This feature keeps track of the state of active connections and makes decisions based on the context of the traffic.
- **Proxy Services:** Some firewalls offer proxy services, which involve acting as an intermediary between internal and external clients to enhance security.
- **Deep Packet Inspection (DPI):** DPI firewalls inspect the actual data within packets to identify and block specific content or threats.
- **Application Layer Filtering:** Modern firewalls often include application-layer filtering to control traffic based on specific applications and services.
- **Logging and Reporting:** Comprehensive logging and reporting capabilities help in monitoring and auditing network activity.

#### Popular commercial firewall software includes:

- Cisco ASA (Adaptive Security Appliance)
- Palo Alto Networks
- Fortinet FortiGate
- Check Point Firewall
- SonicWall

## 2. Intrusion Detection System (IDS) Software:

- **IDS Functionality:** IDS software monitors network traffic for suspicious activity or known attack patterns. It can be classified into two main types: Network-based (NIDS) and Host-based (HIDS).
- **Signature-Based Detection:** This method uses predefined patterns (signatures) to identify known threats, making it effective against known attacks.
- **Anomaly-Based Detection:** Anomaly-based IDS software establishes a baseline of normal network behavior and triggers alerts when deviations occur.
- **Heuristic-Based Detection:** These IDS systems use rules or heuristics to identify suspicious behavior that may indicate an attack.
- **Real-time Alerting:** IDS software generates alerts when it detects potentially malicious activity, which can be sent to administrators for further action.
- **Logging and Reporting:** Similar to firewalls, IDS systems offer logging and reporting features to help in post-incident analysis.

### Notable commercial IDS software includes:

- Snort
- Suricata
- McAfee Network Security Platform
- IBM QRadar
- Trend Micro Deep Security

### When studying commercial firewall and IDS software, consider the following aspects:

- **Licensing and pricing:** Understand the cost structure and licensing terms.
- **Scalability:** Assess whether the software can scale to meet your network's needs.
- **Integration:** Check if the software integrates with other security tools and SIEM systems.
- **Management and ease of use:** Evaluate the user interface and management capabilities.

- **Support and updates:** Look into the availability of support, documentation, and regular updates.
- Ultimately, the choice of commercial firewall and IDS software should align with your organization's security requirements, network architecture, and budget constraints. Conduct thorough research, consider trial versions, and, if possible, consult with experts in the field to make an informed decision.

### 3. Graphical Uncomplicated Firewall

Gufw, which stands for "Graphical Uncomplicated Firewall," is a user-friendly graphical interface for managing the Uncomplicated Firewall (UFW) on Linux systems. UFW is a front-end to the netfilter firewall, which is built into the Linux kernel and provides packet filtering and network address translation (NAT).

#### 1. Purpose:

- Gufw is designed to simplify the process of configuring and managing firewall rules on a Linux system. It provides an easy-to-use, graphical interface for users who may not be comfortable with the command-line interface.

#### 2. Features:

- **Rule Management:** Gufw allows users to create, modify, and delete firewall rules to control network traffic. This includes both incoming and outgoing traffic.
- **Application Profiles:** Gufw can create and manage profiles for different applications or services. This feature simplifies the process of allowing or blocking network traffic for specific applications.
- **Preconfigured Profiles:** It comes with a set of preconfigured profiles for common services such as SSH, web servers, and file sharing. This simplifies the setup for users who don't want to create custom rules.
- **Logging:** Gufw provides logging capabilities, allowing users to monitor firewall activity and track allowed or denied traffic.
- **Status Toggle:** You can easily enable or disable the firewall with a simple toggle switch within the Gufw interface.

#### 3. Basic Concepts:

- **Firewall Rules:** Gufw allows you to define rules that specify what network traffic should be allowed or denied. Rules typically include information like the direction of traffic (inbound or outbound), the protocol (e.g., TCP or UDP), and the port or application involved.

- **Default Policies:** You can set the default policies for incoming and outgoing traffic, specifying whether traffic not explicitly allowed by a rule should be denied or allowed by default.
- **Profiles:** Gufw uses profiles to group rules related to specific applications or services. For example, you might have a profile for a web server with rules allowing HTTP and HTTPS traffic.

#### **4. How Gufw Works:**

- Gufw is essentially a graphical front-end to UFW, which itself interacts with the netfilter framework in the Linux kernel. When you create or modify a rule in Gufw, it translates your settings into UFW commands that are then used to configure the underlying firewall.

#### **5. Advantages:**

- **User-Friendly:** Gufw makes firewall management more accessible to users who are not comfortable with the command line.
- **Easy Setup:** Preconfigured profiles and an intuitive interface simplify the setup of common firewall rules.
- **Logging:** Gufw provides logging capabilities to monitor and track firewall activity.

#### **6. Considerations:**

- While Gufw simplifies firewall management, it's important to understand the basics of networking and firewall rules to use it effectively.
- Misconfigured firewall rules can impact network connectivity and security, so it's essential to be cautious when setting up rules.

In summary, Gufw is a valuable tool for managing firewall rules on Linux systems, especially for users who prefer a graphical interface. It simplifies the process of configuring firewall rules and helps enhance the security of Linux systems by controlling network traffic. However, users should have a basic understanding of networking and firewall concepts to use Gufw effectively.

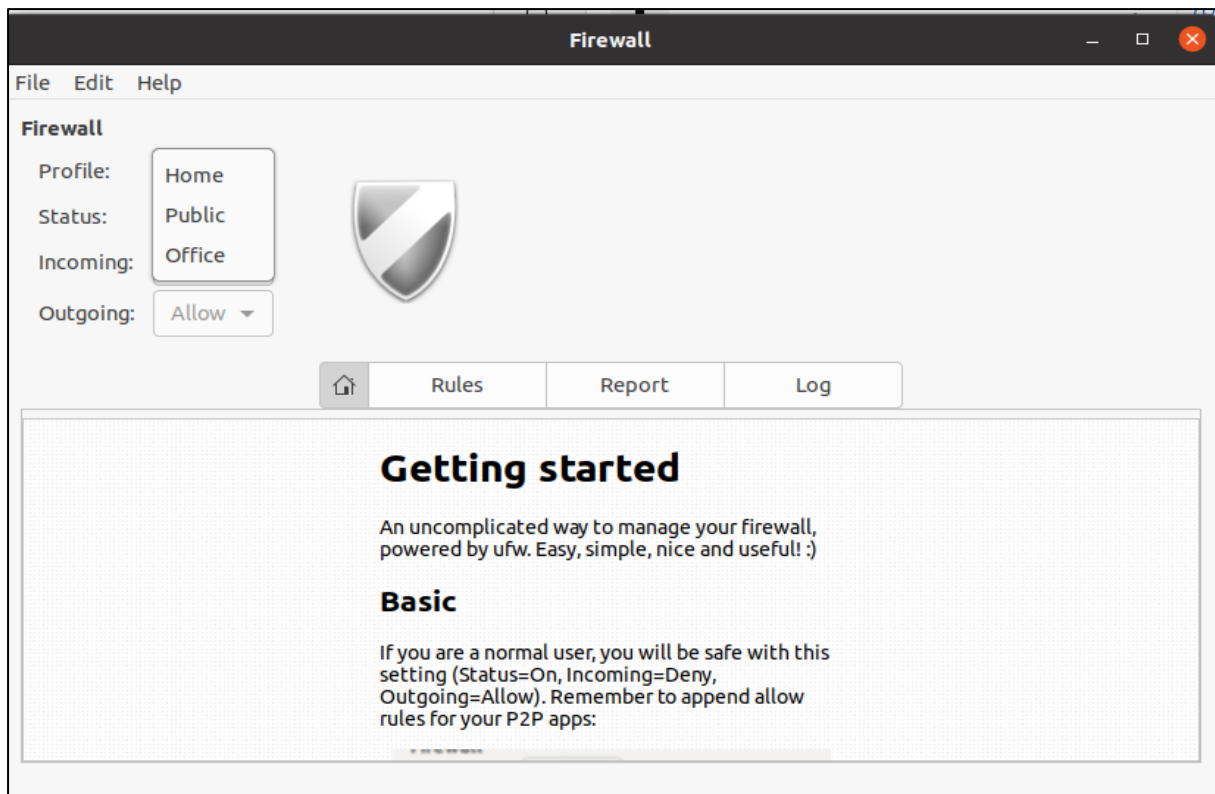
## Implementation:

### Install gufw tool

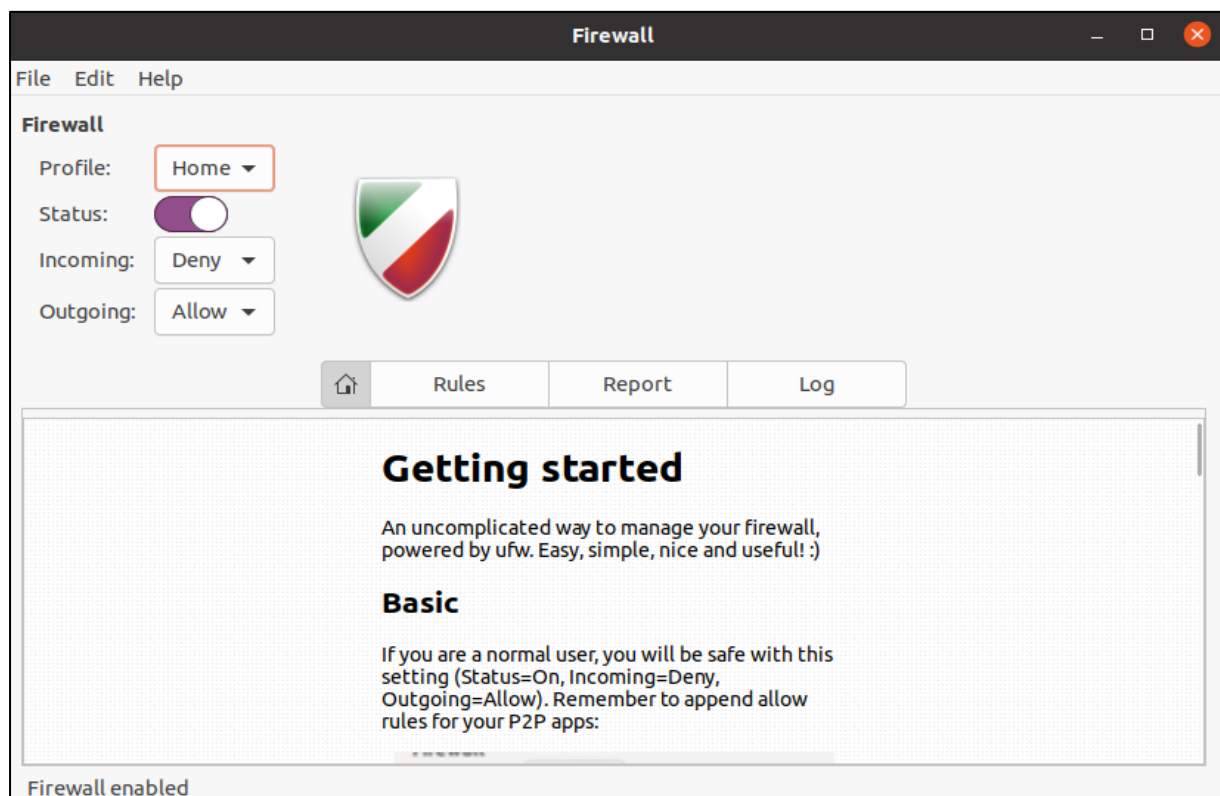
sudo apt-get install gufw

```
kiran@kiran-VirtualBox: ~  
kiran@kiran-VirtualBox:~$ sudo apt-get install gufw  
[sudo] password for kiran:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  lam-runtime liblam4  
Use 'sudo apt autoremove' to remove them.  
The following NEW packages will be installed:  
  gufw  
0 upgraded, 1 newly installed, 0 to remove and 531 not upgraded.  
Need to get 860 kB of archives.  
After this operation, 3,539 kB of additional disk space will be used.  
Get:1 http://in.archive.ubuntu.com/ubuntu focal-updates/universe amd64 gufw all 20.04.1-1ubuntu1 [860 kB]  
Fetched 860 kB in 2s (570 kB/s)  
Selecting previously unselected package gufw.  
(Reading database ... 230051 files and directories currently installed.)  
Preparing to unpack .../gufw_20.04.1-1ubuntu1_all.deb ...  
Unpacking gufw (20.04.1-1ubuntu1) ...  
Setting up gufw (20.04.1-1ubuntu1) ...  
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...  
Processing triggers for mime-support (3.64ubuntu1) ...  
Processing triggers for hicolor-icon-theme (0.17-2) ...  
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...  
Processing triggers for man-db (2.9.1-1) ...  
kiran@kiran-VirtualBox:~$
```

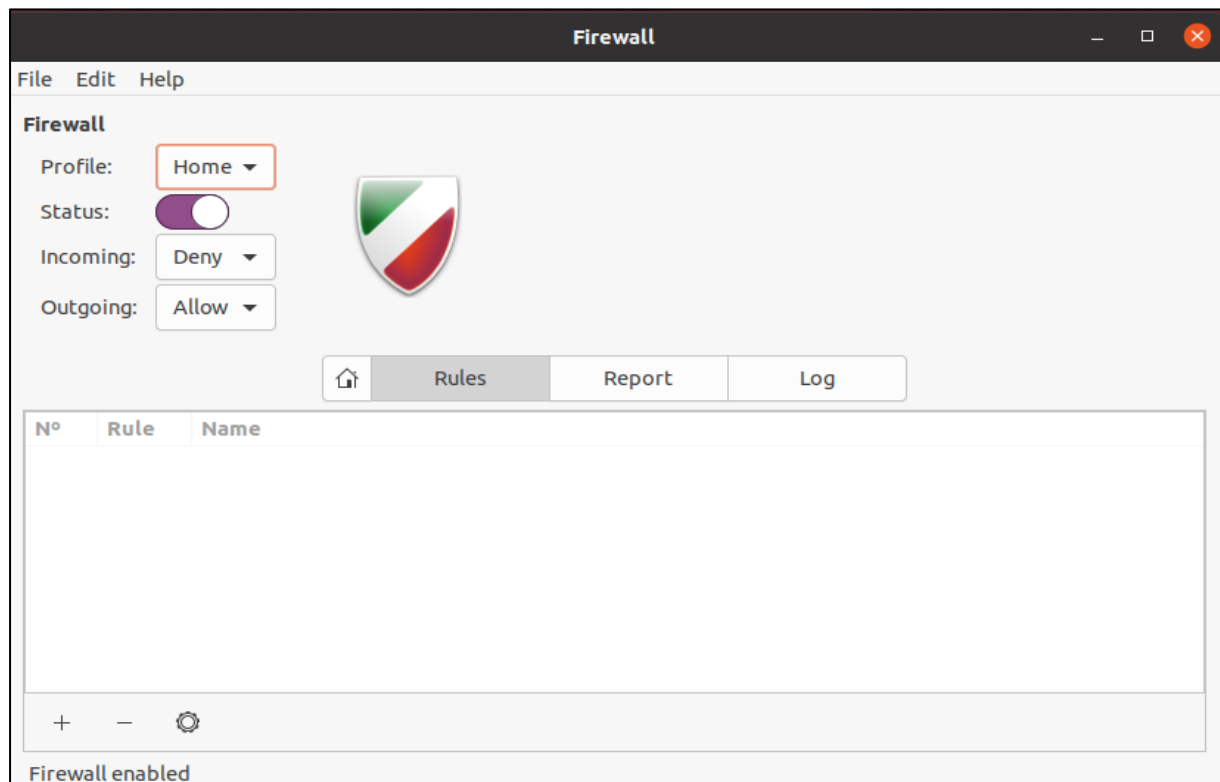
### UI for the gufw tool



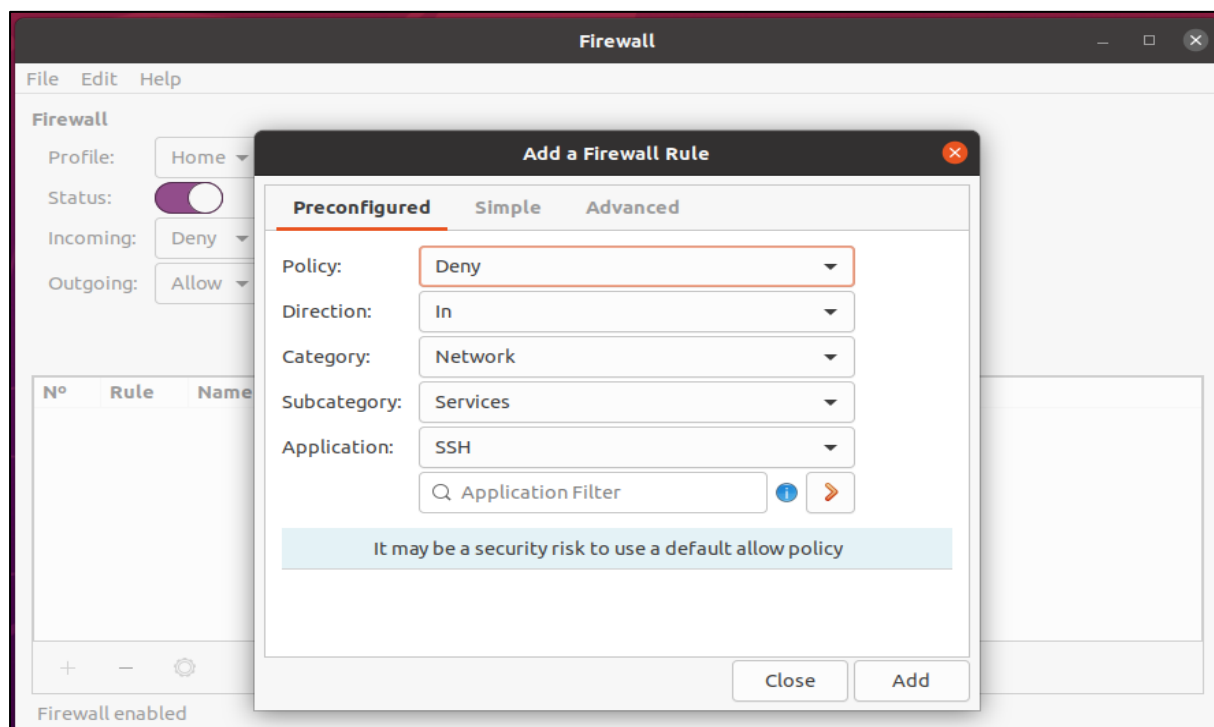
## Enable the firewall



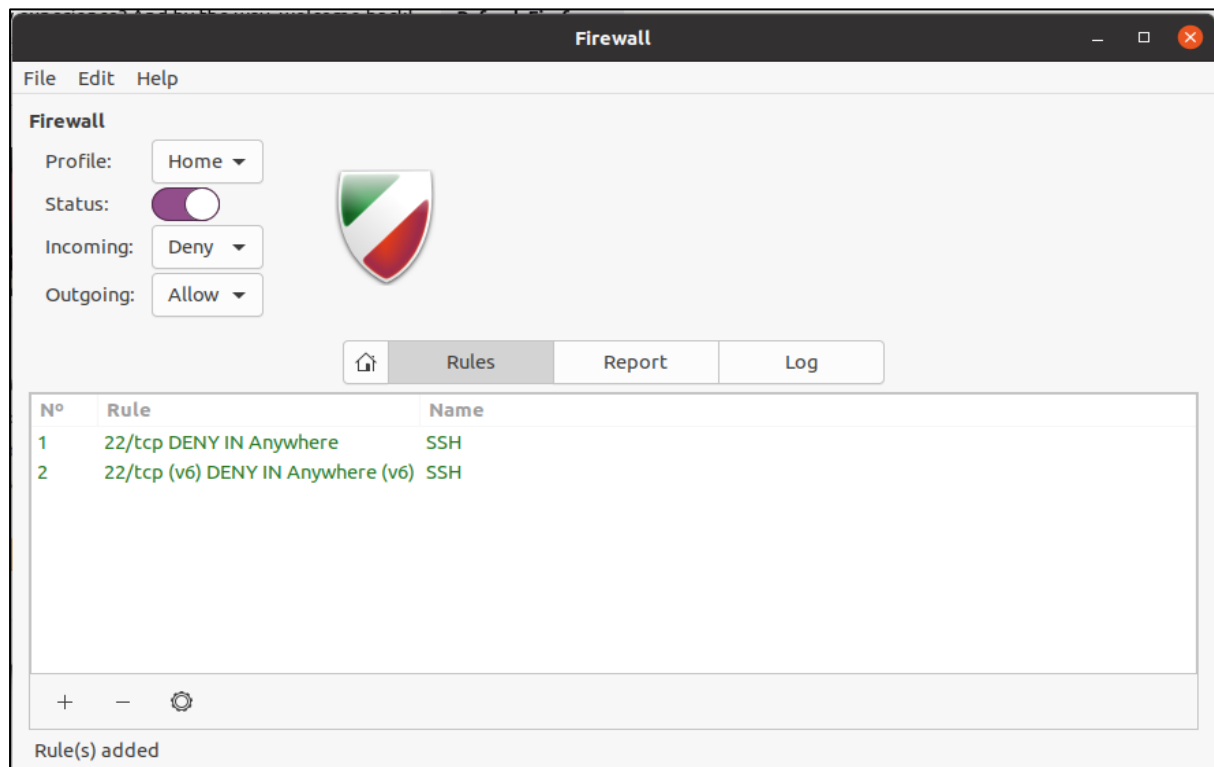
Set the options as below and click on “+” under Rules tab to add firewall rules



## Adding a Firewall rule to deny the SSH



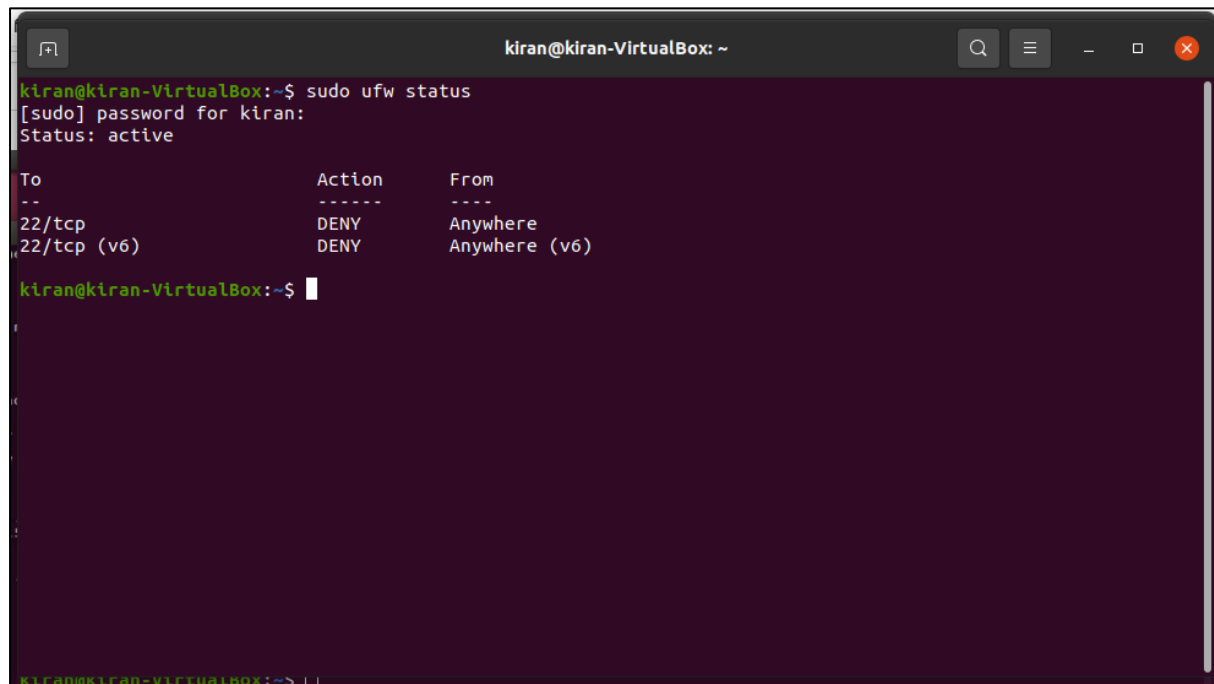
## The Rules are added





## Verify the the firewall rules from trminal

Sudo ufw status

A terminal window titled 'kiran@kiran-VirtualBox: ~' with search, menu, and window control icons in the title bar. The terminal shows the command 'sudo ufw status' being executed. The output indicates the firewall is active and lists two rules: '22/tcp' and '22/tcp (v6)', both with a 'DENY' action and 'Anywhere' as the source. The prompt returns to the user's shell.

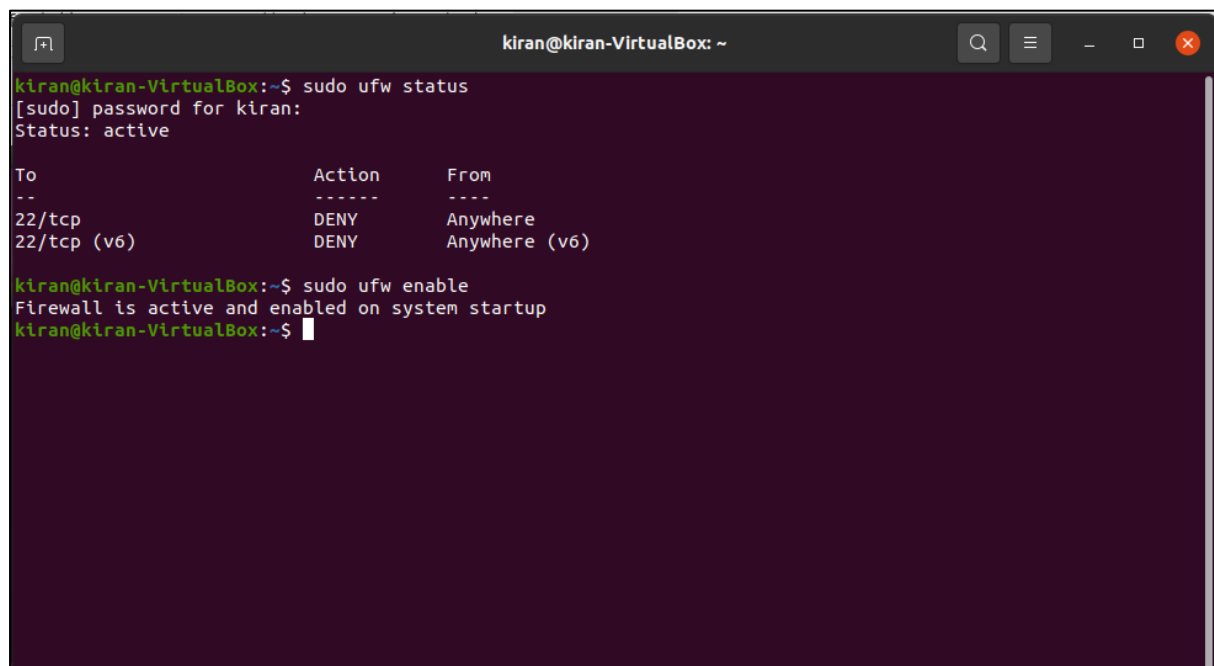
```
kiran@kiran-VirtualBox:~$ sudo ufw status
[sudo] password for kiran:
Status: active

To Action From
--
22/tcp DENY Anywhere
22/tcp (v6) DENY Anywhere (v6)

kiran@kiran-VirtualBox:~$
```

## Enable the firewall

Sudo ufw enable

A terminal window titled 'kiran@kiran-VirtualBox: ~' with search, menu, and window control icons in the title bar. The terminal shows the command 'sudo ufw status' being executed, followed by 'sudo ufw enable'. The output of the first command shows the firewall is active. The second command results in the message 'Firewall is active and enabled on system startup'. The prompt returns to the user's shell.

```
kiran@kiran-VirtualBox:~$ sudo ufw status
[sudo] password for kiran:
Status: active

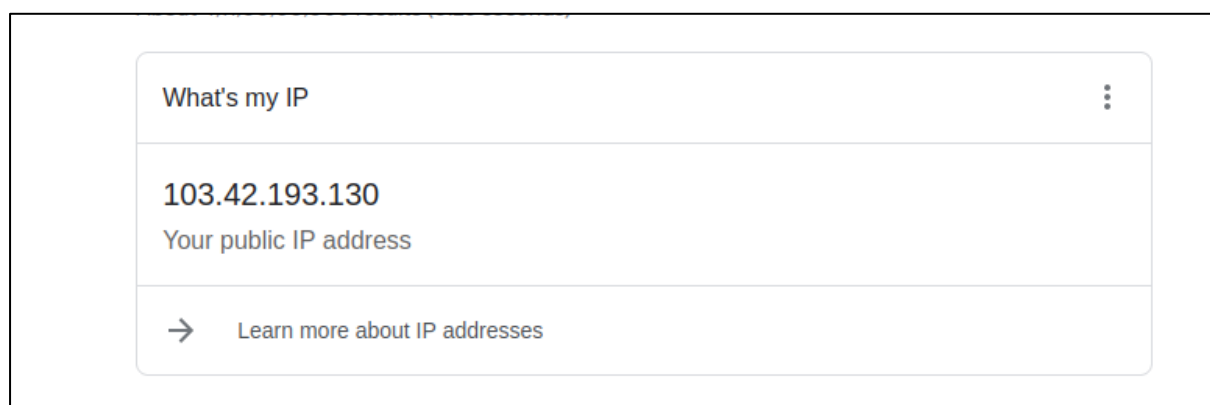
To Action From
--
22/tcp DENY Anywhere
22/tcp (v6) DENY Anywhere (v6)

kiran@kiran-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
kiran@kiran-VirtualBox:~$
```

## Checking the IP address of the system to validate the firewall rules

```
kiran@kiran-VirtualBox: ~  
kiran@kiran-VirtualBox:~$ sudo ufw status  
[sudo] password for kiran:  
Status: active  
  
To Action From  
-- -- --  
22/tcp DENY Anywhere  
22/tcp (v6) DENY Anywhere (v6)  
  
kiran@kiran-VirtualBox:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
kiran@kiran-VirtualBox:~$ hostname -I  
10.0.2.15  
kiran@kiran-VirtualBox:~$
```

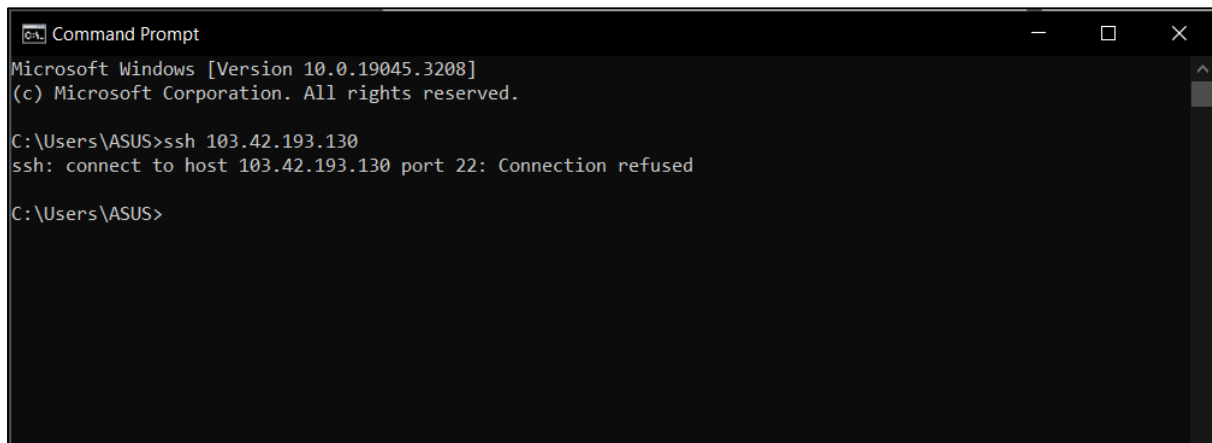
```
kiran@kiran-VirtualBox: ~  
kiran@kiran-VirtualBox:~$ sudo ufw status  
[sudo] password for kiran:  
Status: active  
  
To Action From  
-- -- --  
22/tcp DENY Anywhere  
22/tcp (v6) DENY Anywhere (v6)  
  
kiran@kiran-VirtualBox:~$ sudo ufw enable  
Firewall is active and enabled on system startup  
kiran@kiran-VirtualBox:~$ hostname -I  
10.0.2.15  
kiran@kiran-VirtualBox:~$ nmcli -p device show  
=====  
Device details (enp0s3)  
=====  
GENERAL.DEVICE: enp0s3  
-----  
GENERAL.TYPE: ethernet  
-----  
GENERAL.HWADDR: 08:00:27:11:9A:C7  
-----  
GENERAL.MTU: 1500  
-----  
GENERAL.STATE: 100 (connected)  
-----
```



Here 103.42.193.130 is my public Ip address

### Verifying from other machine :

SSH 103.42.193.130

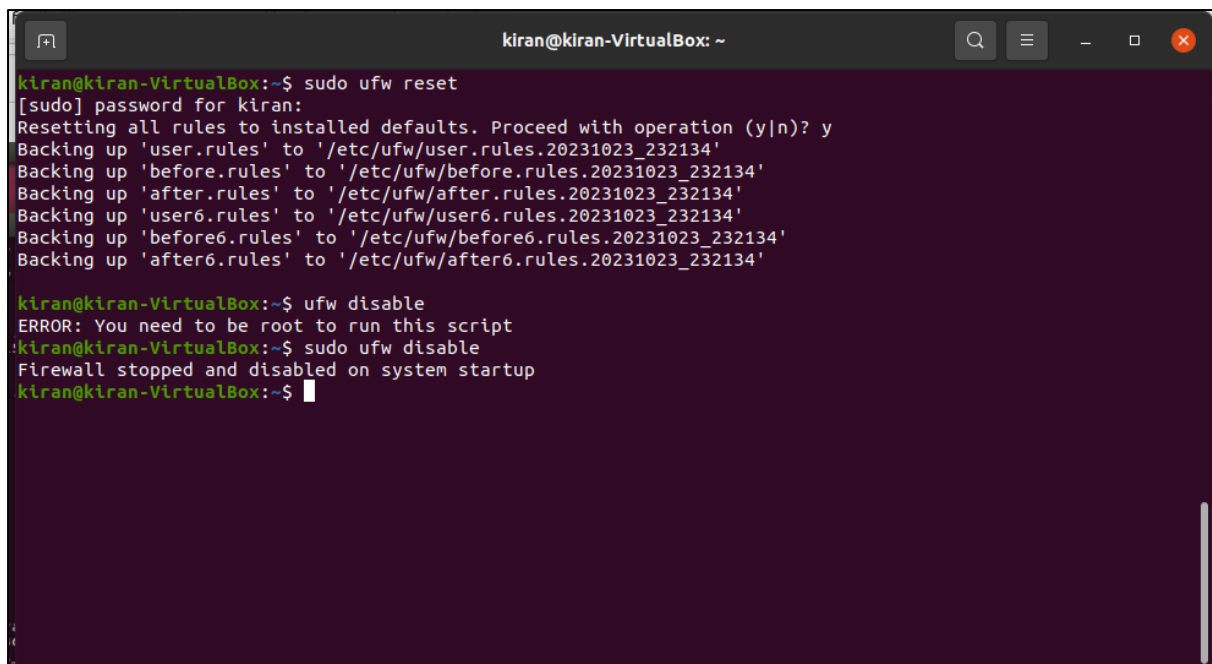


```
Command Prompt
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>ssh 103.42.193.130
ssh: connect to host 103.42.193.130 port 22: Connection refused

C:\Users\ASUS>
```

### Reset The firewall rules and disabling it



```
kiran@kiran-VirtualBox: ~$ sudo ufw reset
[sudo] password for kiran:
Resetting all rules to installed defaults. Proceed with operation (y|n)? y
Backing up 'user.rules' to '/etc/ufw/user.rules.20231023_232134'
Backing up 'before.rules' to '/etc/ufw/before.rules.20231023_232134'
Backing up 'after.rules' to '/etc/ufw/after.rules.20231023_232134'
Backing up 'user6.rules' to '/etc/ufw/user6.rules.20231023_232134'
Backing up 'before6.rules' to '/etc/ufw/before6.rules.20231023_232134'
Backing up 'after6.rules' to '/etc/ufw/after6.rules.20231023_232134'

kiran@kiran-VirtualBox:~$ ufw disable
ERROR: You need to be root to run this script
kiran@kiran-VirtualBox:~$ sudo ufw disable
Firewall stopped and disabled on system startup
kiran@kiran-VirtualBox:~$
```

### Conclusion:

In Conclusion, the practical experiment with the UFW (Uncomplicated Firewall) tool demonstrated its user-friendly and effective approach to network security and setting up the firewall. While UFW offers straightforward firewall management, it's vital to remember that it should be integrated into a broader security strategy.