CYBER SECURITY INTERNSHIP

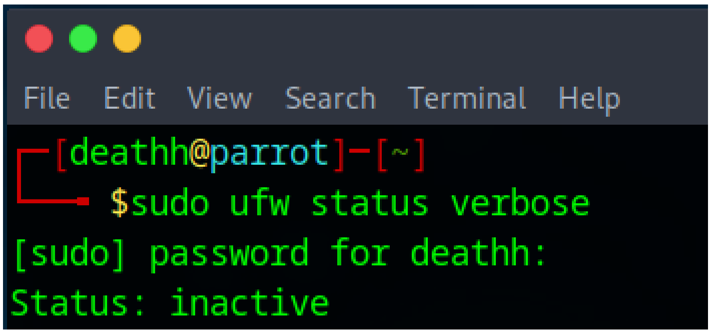
Name – Saimon Soren

Gmail – saimonsoren200206@gmail.com

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

For Linux (UFW - Uncomplicated Firewall)

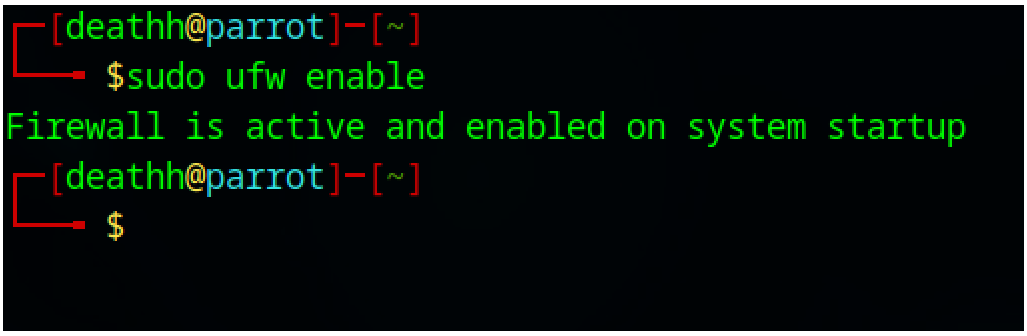
ufw is a command-line tool



```
┌─[deathh@parrot]─[~]
└──╼ $sudo ufw status verbose
[sudo] password for deathh:
Status: inactive
```

It is a tool for managing the firewall in linux. In the above screenshot we can see that the ufw status is inactive.

Now I will enable the ufw tool:

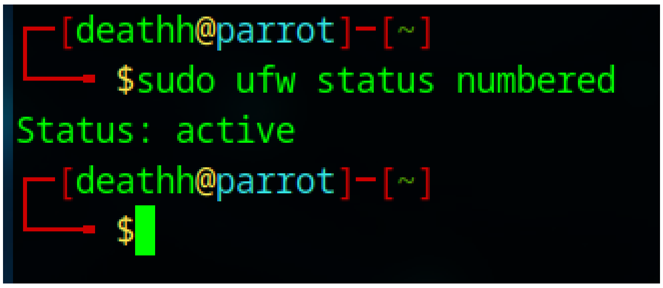


```
┌─[deathh@parrot]─[~]
└──╼ $sudo ufw enable
Firewall is active and enabled on system startup
┌─[deathh@parrot]─[~]
└──╼ $
```

It shows that it has enabled the firewall on the machine.

To list the Current firewall:

I will write this command "sudo ufw status numbered"

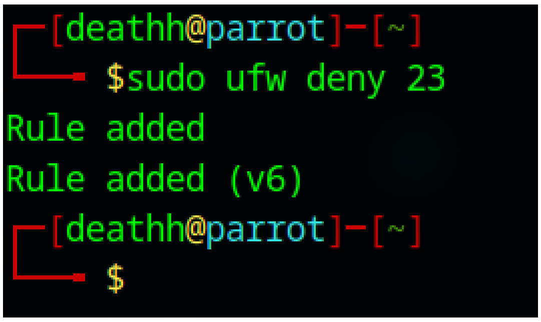


```
┌─[deathh@parrot]─[~]
└──╼ $sudo ufw status numbered
Status: active
┌─[deathh@parrot]─[~]
└──╼ $
```

In the above screenshot we can notice that no custom rules have been added yet.

It is only applying its default policy.

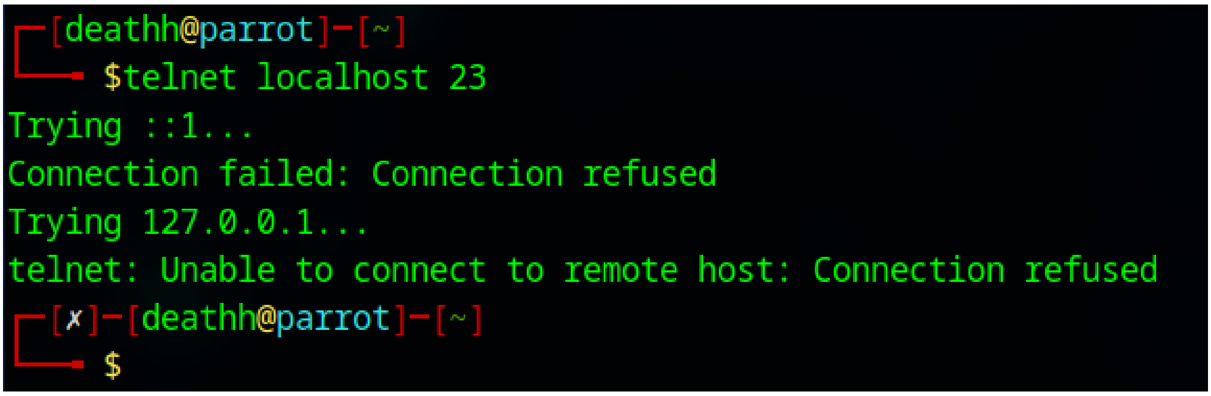
Now we will add a rule to block the Inbound traffic on port 23



```
┌─[deathh@parrot]─[~]
└──╼ $sudo ufw deny 23
Rule added
Rule added (v6)
┌─[deathh@parrot]─[~]
└──╼ $
```

The above screenshot shows the command to block Telnet Port no. 23;

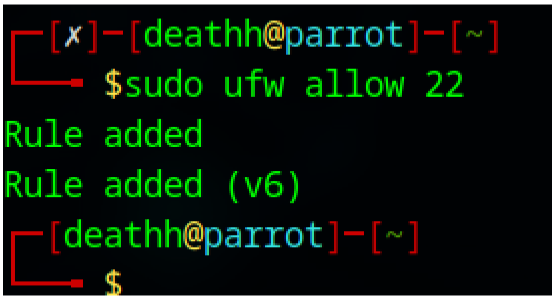
TO TEST THE RULE, I WILL WRITE THE COMMAND BELOW:



```
┌─[deathh@parrot]─[~]
└──╼ $telnet localhost 23
Trying ::1...
Connection failed: Connection refused
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection refused
┌─[✗]─[deathh@parrot]─[~]
└──╼ $
```

After running the “telnet localhost 23” command it shows “**Unable to connect to remote host: Connection refused**”

Let’s add a rule to allow port number 22 –



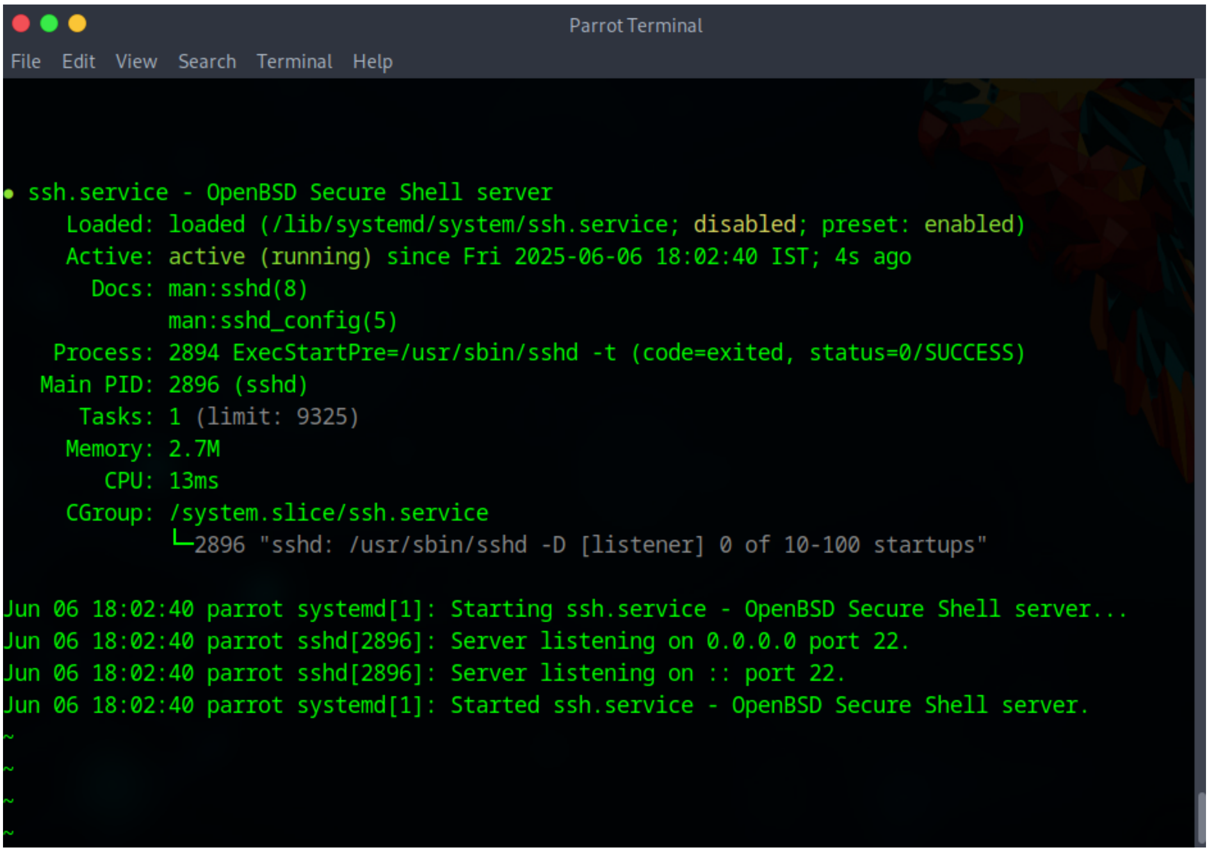
```
┌─[✗]─[deathh@parrot]─[~]
└──╼ $sudo ufw allow 22
Rule added
Rule added (v6)
┌─[deathh@parrot]─[~]
└──╼ $
```

Successfully added the SSH Port on firewall rule.

First check if the ssh service is disabled or not; if disabled write this command on terminal

sudo systemctl status ssh

To test the rule we can write the – sudo system status ssh



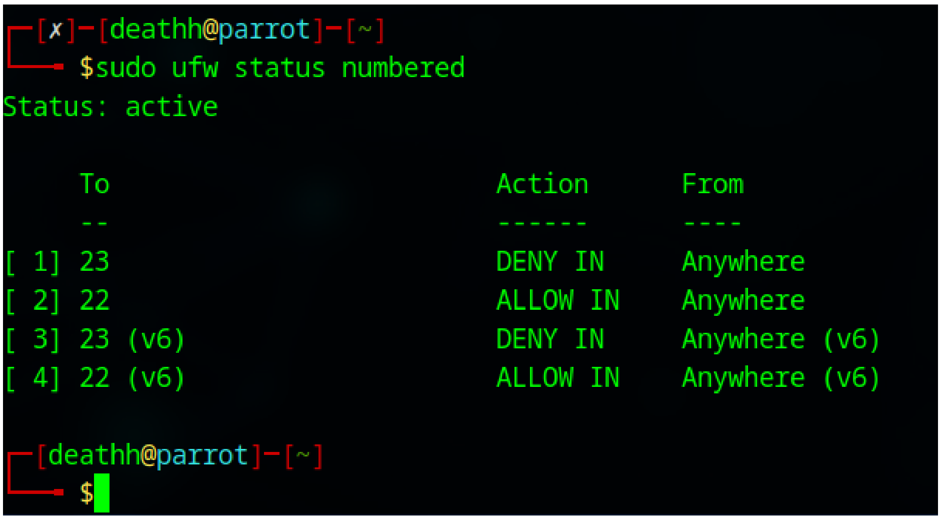
```
• ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset: enabled)
     Active: active (running) since Fri 2025-06-06 18:02:40 IST; 4s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 2894 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 2896 (sshd)
      Tasks: 1 (limit: 9325)
     Memory: 2.7M
        CPU: 13ms
     CGroup: /system.slice/ssh.service
             └─2896 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jun 06 18:02:40 parrot systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Jun 06 18:02:40 parrot sshd[2896]: Server listening on 0.0.0.0 port 22.
Jun 06 18:02:40 parrot sshd[2896]: Server listening on :: port 22.
Jun 06 18:02:40 parrot systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
~
~
~
~
```

It shows that the SSH service is running properly and not dead.

Now we can easily connect through a remote desktop on the same network.

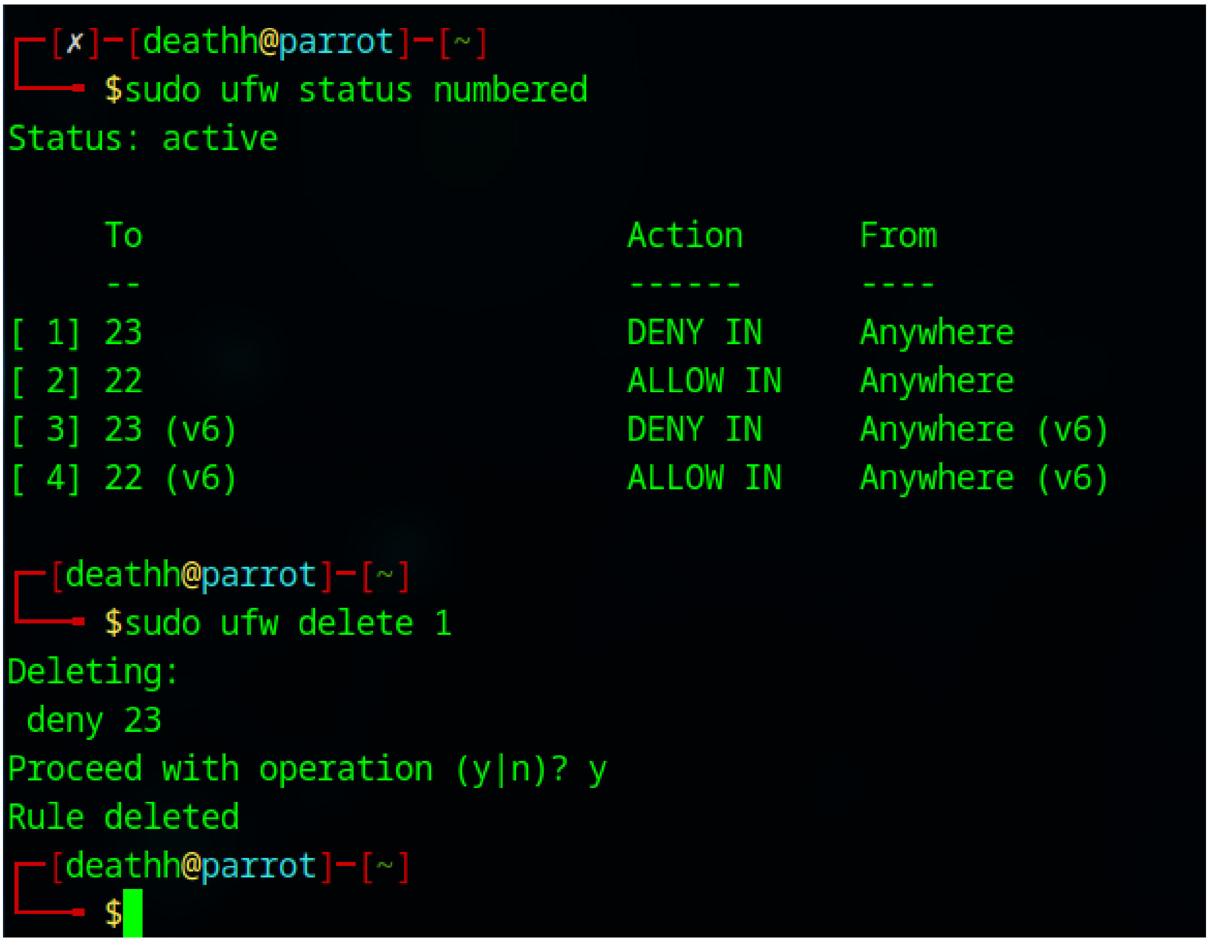
<u>Now, to remove the Block Rule to Restore Original State</u>:

- List the rules:
  sudo ufw status numbered



```
┌─[✗]─[deathh@parrot]─[~]
└──╼ $sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 23                         DENY IN     Anywhere
[ 2] 22                         ALLOW IN    Anywhere
[ 3] 23 (v6)                    DENY IN     Anywhere (v6)
[ 4] 22 (v6)                    ALLOW IN    Anywhere (v6)

┌─[deathh@parrot]─[~]
└──╼ $
```

- Delete the rule by its number:
  sudo ufw delete 1



```
┌─[✗]─[deathh@parrot]─[~]
└──╼ $sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 23                         DENY IN     Anywhere
[ 2] 22                         ALLOW IN    Anywhere
[ 3] 23 (v6)                    DENY IN     Anywhere (v6)
[ 4] 22 (v6)                    ALLOW IN    Anywhere (v6)

┌─[deathh@parrot]─[~]
└──╼ $sudo ufw delete 1
Deleting:
 deny 23
Proceed with operation (y|n)? y
Rule deleted
┌─[deathh@parrot]─[~]
└──╼ $
```

Above is the demonstration showed for the rule deletion.

Summary on How UFW Filters Traffic

- UFW is a straightforward layer on top of iptables used to manipulate firewall rules.

- By default, it blocks all incoming traffic unless allowed.

- You can add rules to allow or deny specific ports, protocols, or IP addresses.

- Rules are acted on in order.

- It helps to minimize exposure of services to unintended communication.

Commands used:

- *sudo ufw enable*
- *sudo ufw status verbose*
- *sudo ufw deny 23*
- *sudo ufw allow 22*
- *sudo ufw status numbered*
- *sudo ufw delete <rule_number>*