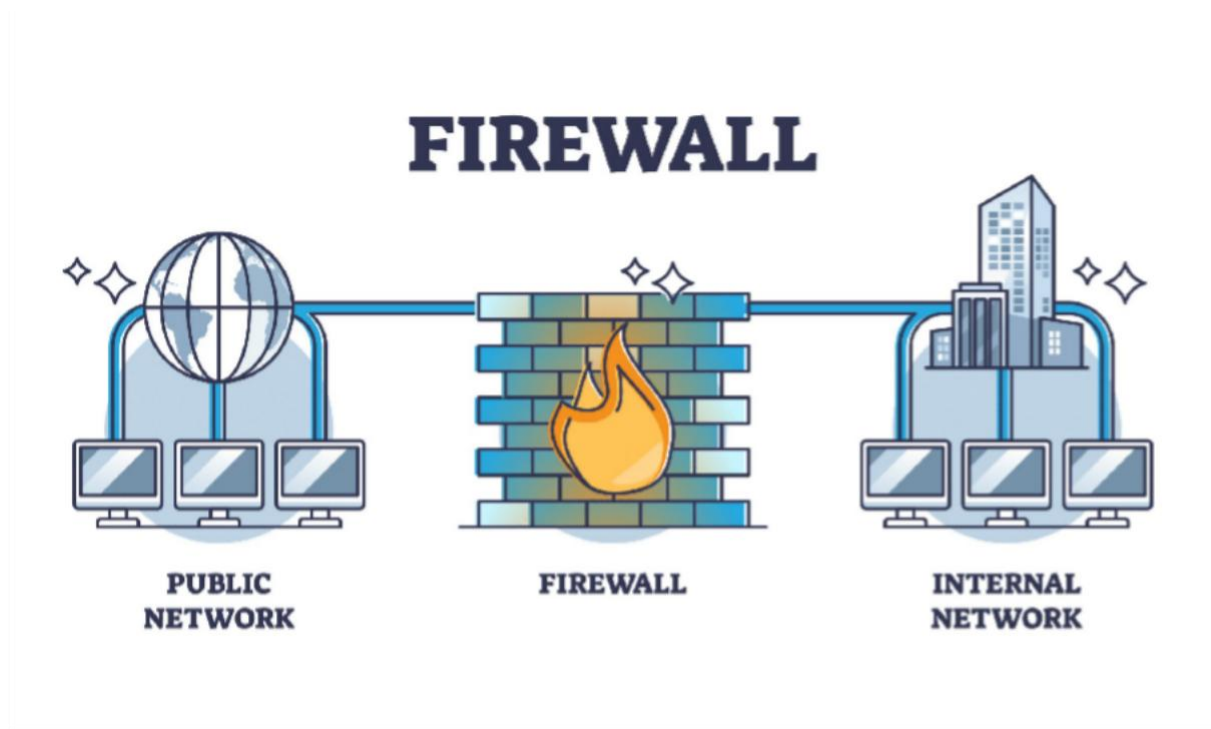


Setup and Use a Firewall on Windows/Linux



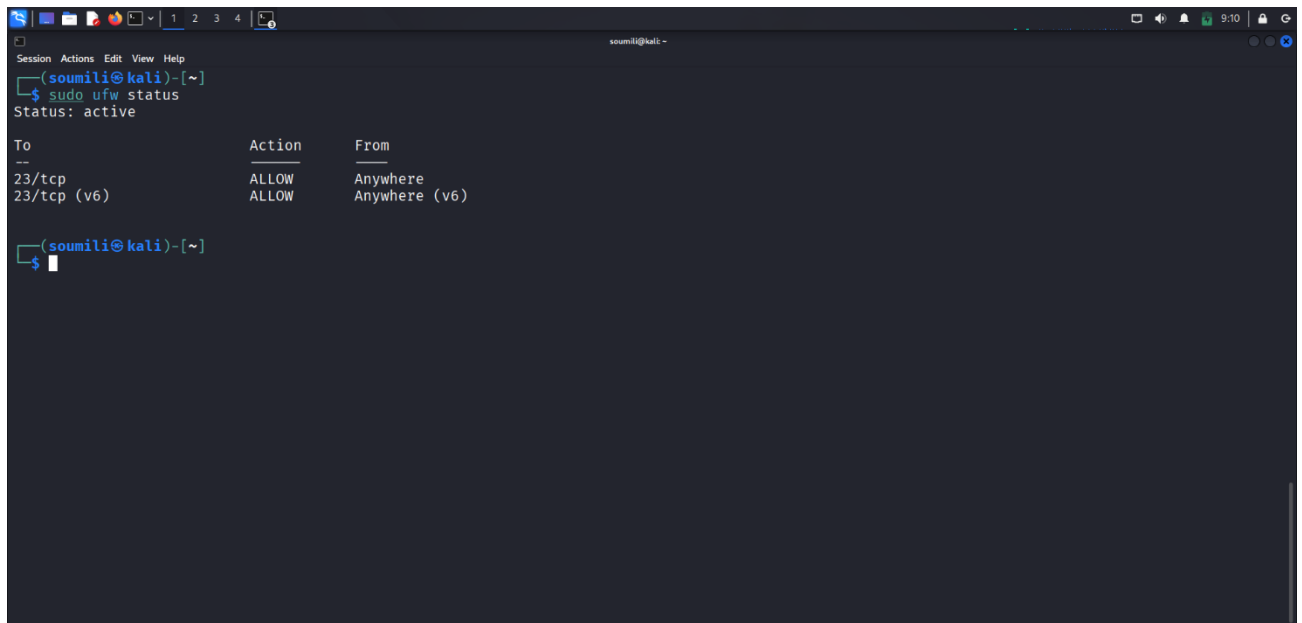
We setup and use firewall on Linux.

Open Firewall Configuration Tool

On Linux (UFW):

Open firewall configuration tool (**Windows Firewall or terminal for UFW**)

Open a terminal and run “**sudo ufw status**” to check if it's active.
If not, enable it with “**sudo ufw enable**”.

A terminal window on a Kali Linux system. The prompt is (soumili@kali)-[~]. The command 'sudo ufw status' has been executed, showing 'Status: active'. Below this, a table of firewall rules is displayed. The table has three columns: 'To', 'Action', and 'From'. There are two rules listed: one for '23/tcp' with 'ALLOW' action from 'Anywhere', and another for '23/tcp (v6)' with 'ALLOW' action from 'Anywhere (v6)'. The prompt is now \$.

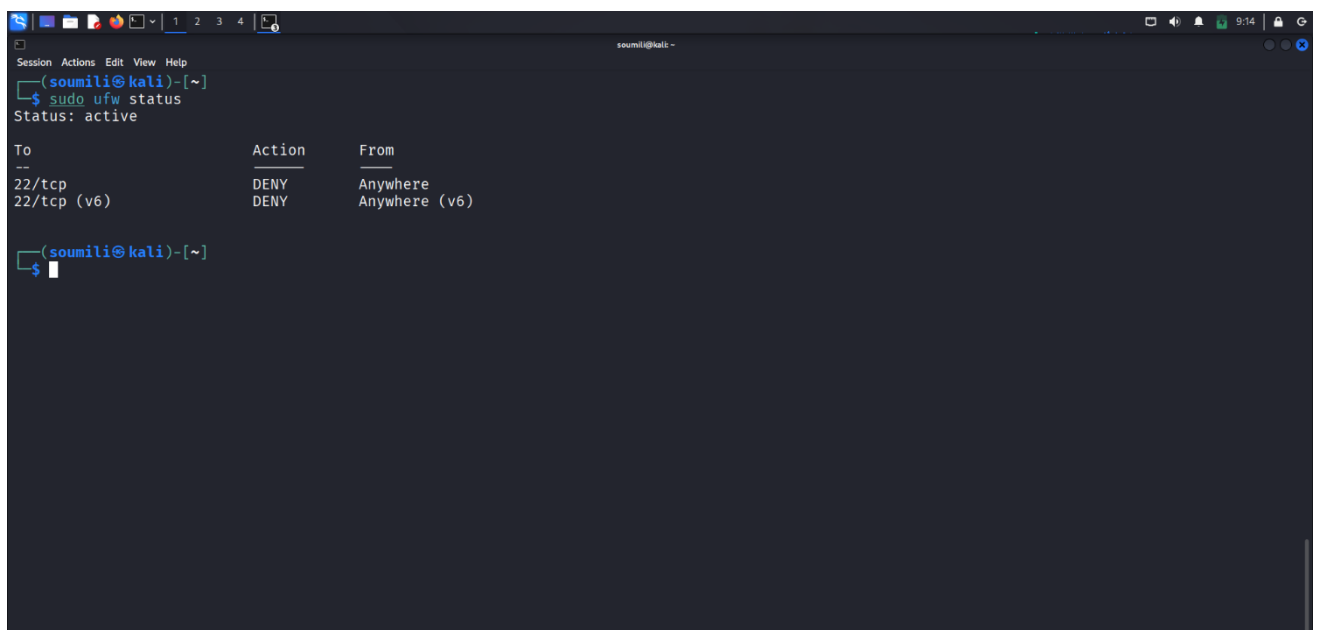
```
(soumili@kali)-[~]  
$ sudo ufw status  
Status: active  
  
To      Action      From  
--      -  
23/tcp  ALLOW       Anywhere  
23/tcp (v6)  ALLOW       Anywhere (v6)  
  
(soumili@kali)-[~]  
$
```

List Current Firewall Rules

On Linux (UFW):

Run “**sudo ufw status verbose**” in the terminal. This lists active rules, including allowed/denied ports and directions (e.g., “**22/tcp ALLOW IN Anywhere**”).

First status 22/tcp Deny

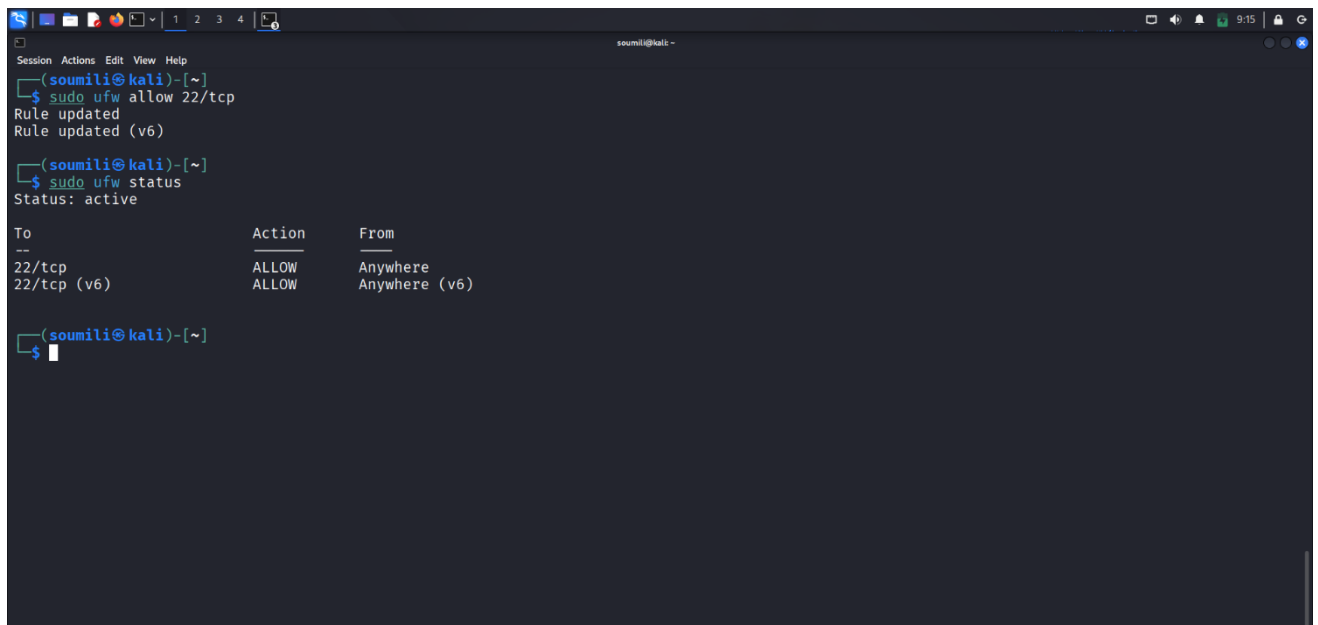


```
Session Actions Edit View Help
(soumili@kali)-[~]
$ sudo ufw status
Status: active

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

(soumili@kali)-[~]
$
```

Then Allow 22/tcp



```
Session Actions Edit View Help
(soumili@kali)-[~]
$ sudo ufw allow 22/tcp
Rule updated
Rule updated (v6)

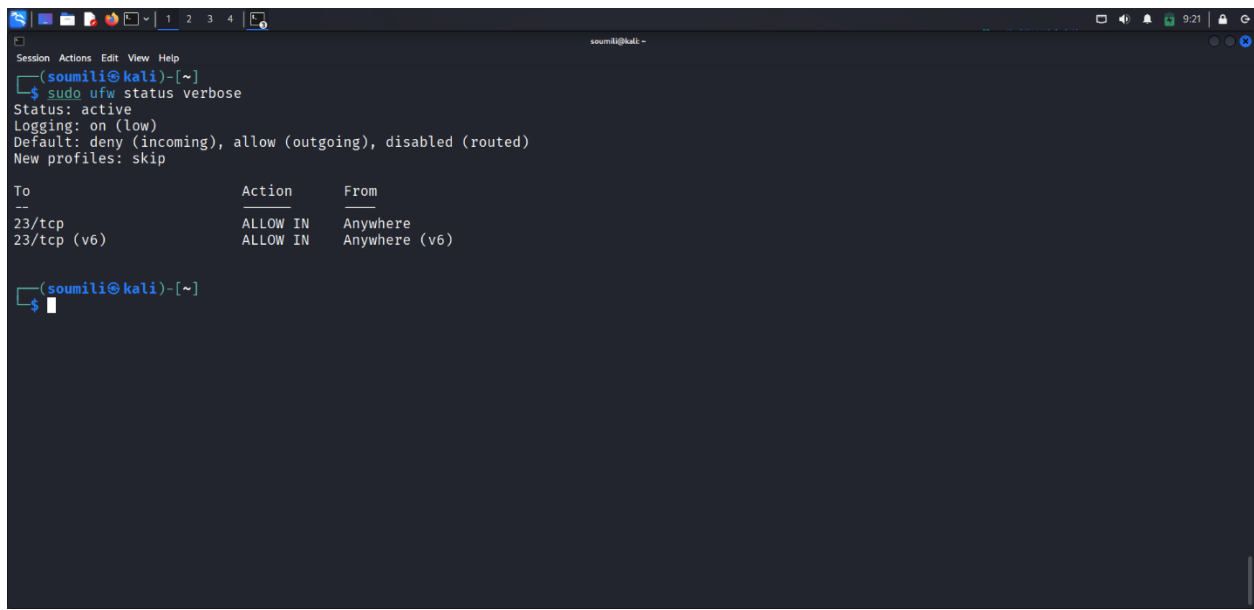
(soumili@kali)-[~]
$ sudo ufw status
Status: active

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

(soumili@kali)-[~]
$
```

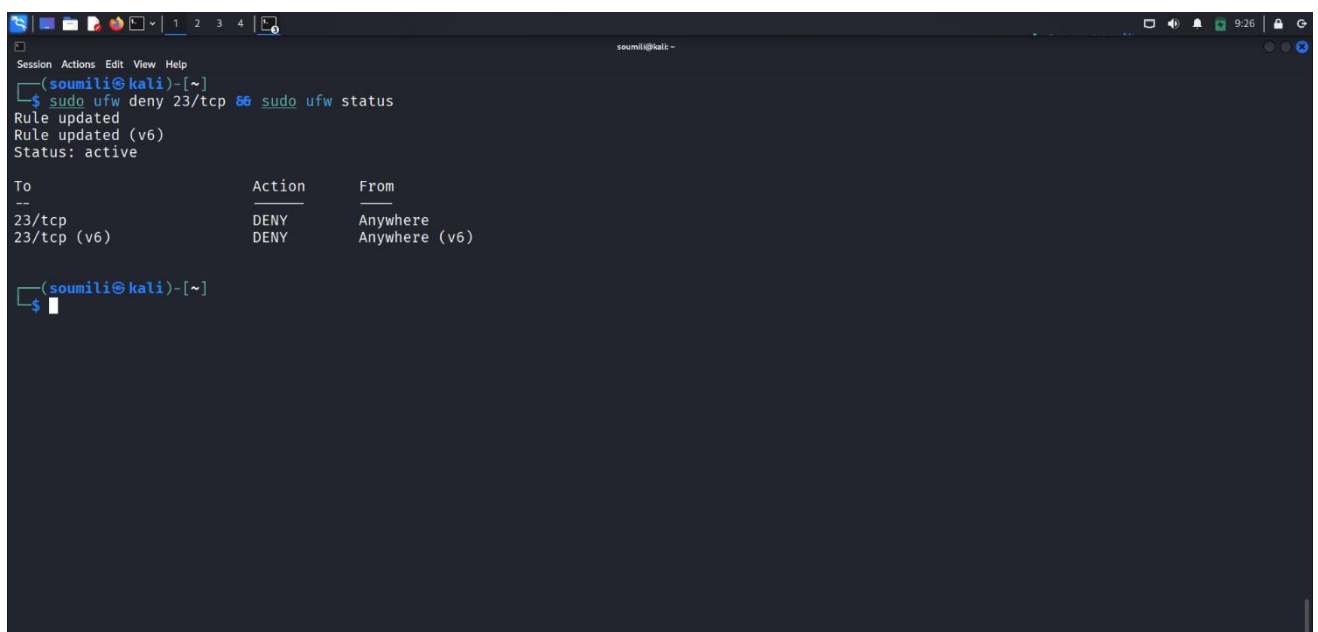
Add a Rule to Block Inbound Traffic on a Specific Port (e.g., 23 for Telnet)

On Linux (UFW):

A terminal window on a Kali Linux system. The user is at the prompt (soumili@kali)~. They have run 'sudo ufw status verbose'. The output shows UFW is active, logging is on (low), default is deny for incoming, and new profiles are skipped. A table lists current rules: 23/tcp and 23/tcp (v6), both with 'ALLOW IN' action from 'Anywhere' and 'Anywhere (v6)' respectively.

```
(soumili@kali)~  
$ sudo ufw status verbose  
Status: active  
Logging: on (low)  
Default: deny (incoming), allow (outgoing), disabled (routed)  
New profiles: skip  
  
To Action From  
--  
23/tcp ALLOW IN Anywhere  
23/tcp (v6) ALLOW IN Anywhere (v6)  
  
(soumili@kali)~  
$
```

Run “**sudo ufw deny 23/tcp**” (this blocks inbound on port 23).
To confirm, run “**sudo ufw status**”.

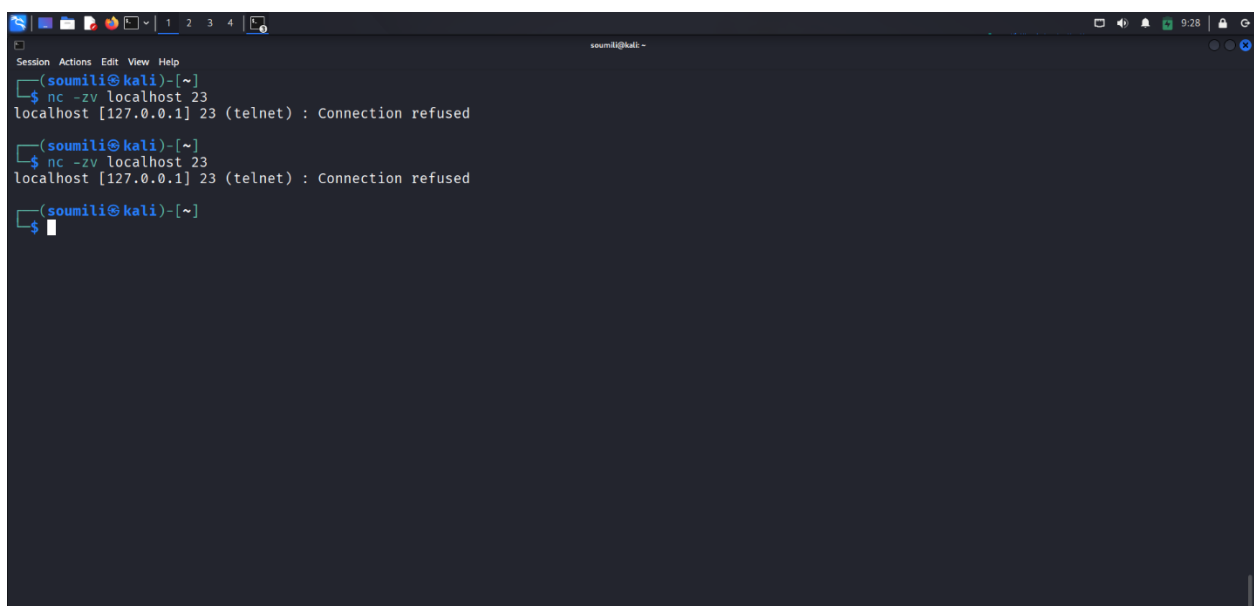
A terminal window on a Kali Linux system. The user is at the prompt (soumili@kali)~. They have run 'sudo ufw deny 23/tcp' and then 'sudo ufw status'. The output shows the rule has been updated to DENY for 23/tcp and 23/tcp (v6). The status is active. A table lists current rules: 23/tcp and 23/tcp (v6), both with 'DENY' action from 'Anywhere' and 'Anywhere (v6)' respectively.

```
(soumili@kali)~  
$ sudo ufw deny 23/tcp  
Rule updated  
Rule updated (v6)  
Status: active  
  
To Action From  
--  
23/tcp DENY Anywhere  
23/tcp (v6) DENY Anywhere (v6)  
  
(soumili@kali)~  
$
```

Test the Rule by Attempting to Connect to That Port Locally or Remotely

- **Local Test (on the same machine):**

Using Netcat on Linux & Try to connect.

A terminal window titled 'soumili@kali' showing a netcat listener on port 23. The user enters 'nc -zv localhost 23' twice, both times receiving the output 'localhost [127.0.0.1] 23 (telnet) : Connection refused'. The prompt returns to the user's shell after each attempt.

```
(soumili@kali)-[~]  
$ nc -zv localhost 23  
localhost [127.0.0.1] 23 (telnet) : Connection refused  
  
(soumili@kali)-[~]  
$ nc -zv localhost 23  
localhost [127.0.0.1] 23 (telnet) : Connection refused  
  
(soumili@kali)-[~]  
$
```

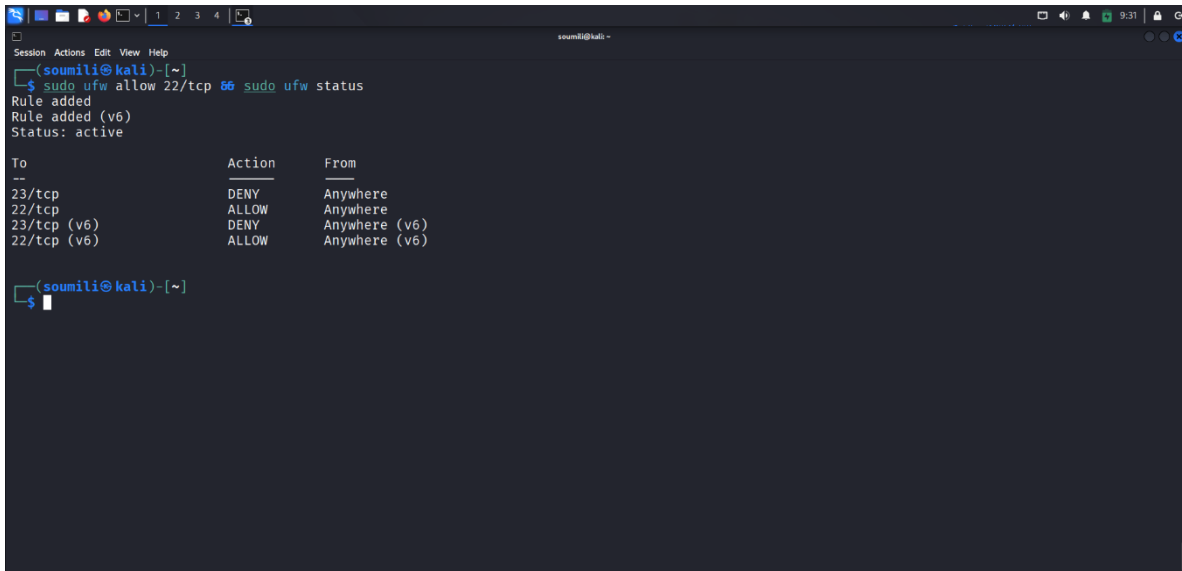
Reports:

Expected result: Connection should fail (e.g., "Connection refused" or timeout), confirming the block.

Add Rule to Allow SSH (Port 22) If on Linux

On Linux (UFW):

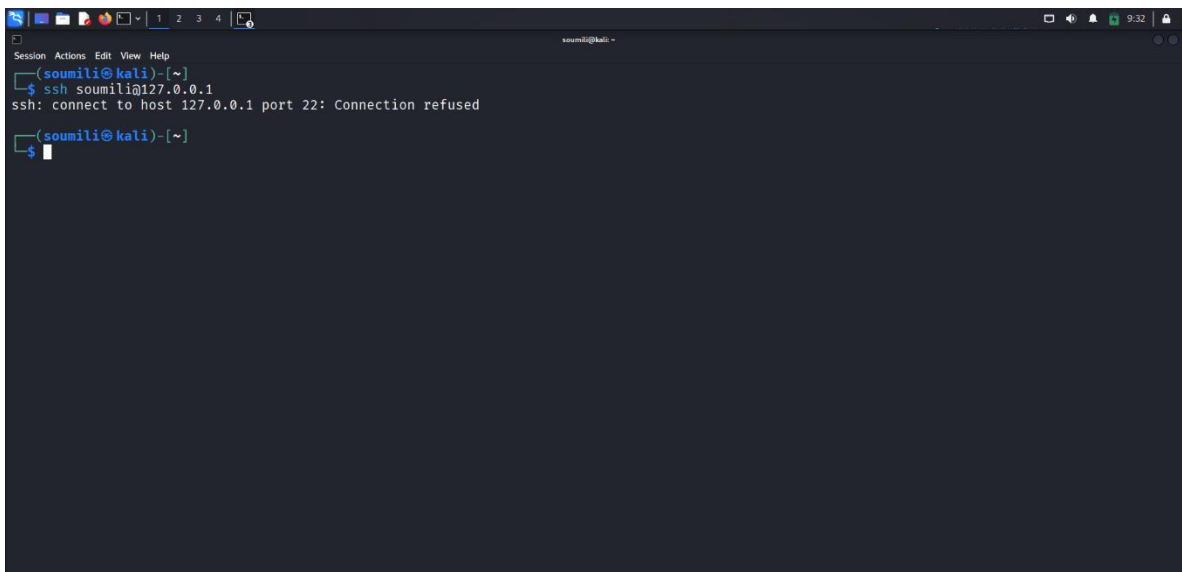
Run “**sudo ufw allow 22/tcp**” (allows inbound SSH). Confirm with “**sudo ufw status**”.

A terminal window on a Kali Linux machine. The user runs 'sudo ufw allow 22/tcp' and then 'sudo ufw status'. The output shows the rule is added and active. A table displays the current firewall rules.

```
soumili@kali ~  
$ sudo ufw allow 22/tcp  
Rule added  
Rule added (v6)  
Status: active  
  
To Action From  
--  
23/tcp DENY Anywhere  
22/tcp ALLOW Anywhere  
23/tcp (v6) DENY Anywhere (v6)  
22/tcp (v6) ALLOW Anywhere (v6  
  
soumili@kali ~  
$
```

Test by connecting via SSH from another machine: ssh user@<your-IP>

Result:

A terminal window on a Kali Linux machine. The user runs 'ssh soumili@127.0.0.1'. The output shows the connection is refused.

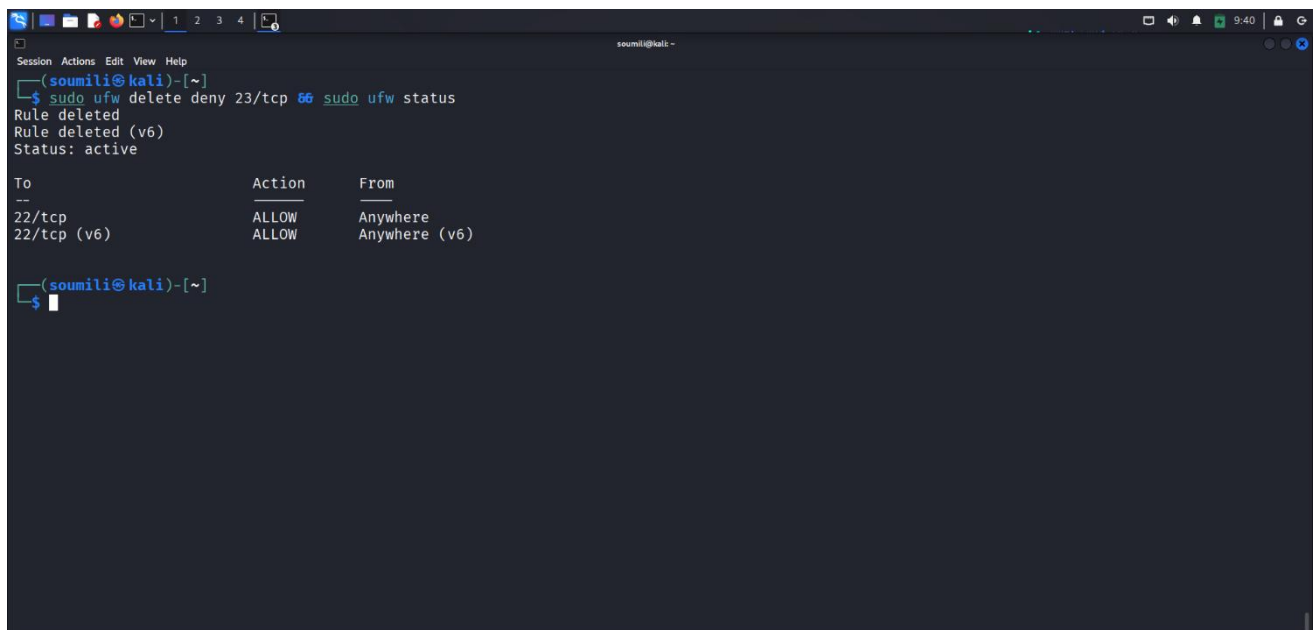
```
soumili@kali ~  
$ ssh soumili@127.0.0.1  
ssh: connect to host 127.0.0.1 port 22: Connection refused  
  
soumili@kali ~  
$
```

Connection Refused.

Remove the Test Block Rule to Restore Original State

On Linux (UFW):

Run “**sudo ufw delete deny 23/tcp**”. Confirm with “**sudo ufw status**”.

A terminal window on a Kali Linux system. The user enters the command 'sudo ufw delete deny 23/tcp' and then 'sudo ufw status'. The output shows the rule being deleted and the current firewall status. The status is active, and the rules table shows two rules: '22/tcp' with action 'ALLOW' from 'Anywhere', and '22/tcp (v6)' with action 'ALLOW' from 'Anywhere (v6)'.

```
(soumili@kali)-[~]  
$ sudo ufw delete deny 23/tcp  
Rule deleted  
Rule deleted (v6)  
Status: active  
  
To Action From  
--  
22/tcp ALLOW Anywhere  
22/tcp (v6) ALLOW Anywhere (v6)  
  
(soumili@kali)-[~]  
$
```

Document Commands or GUI Steps Used

Create a log file (e.g., **text document**) and record each step, including exact commands/GUI actions, timestamps, and outcomes.

Example:

Step 1: Opened firewall.cpl on Windows.

Step 2: Added inbound rule to block port 23 via GUI.

Step 3: Tested with `nc -zv localhost 23`; result: Connection refused.

This documentation helps track changes and troubleshoot.

Summarize How Firewall Filters Traffic

Firewalls act as a barrier between your system and the network, controlling inbound and outbound traffic based on rules. They filter packets by criteria like source/destination IP, port, protocol (**e.g., TCP/UDP**), and state (**e.g., new vs. established connections**). For example:

- **Inbound Filtering**: Blocks unwanted incoming connections (**e.g., denying port 23 prevents Telnet access**).
- **Outbound Filtering**: Restricts what your system can send out (**less common but useful for malware prevention**).
- **Default Behavior**: Most firewalls deny all inbound by default and allow outbound, requiring explicit "**allow**" rules for services like SSH.
- **Stateful Inspection**: Tracks connection states, allowing responses to outbound requests while blocking unsolicited inbound traffic. This layered approach enhances security by reducing attack surfaces, but over- restrictive rules can break functionality.

Thank You