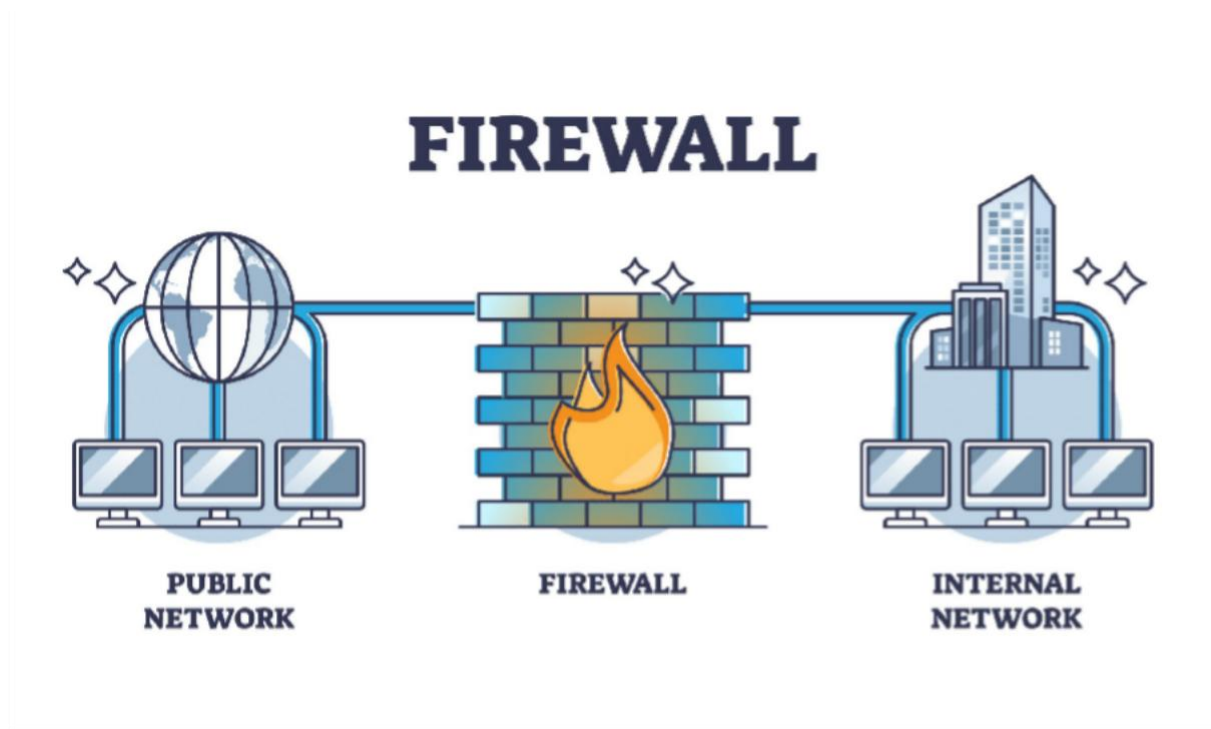


Setup and Use a Firewall on Windows/Linux



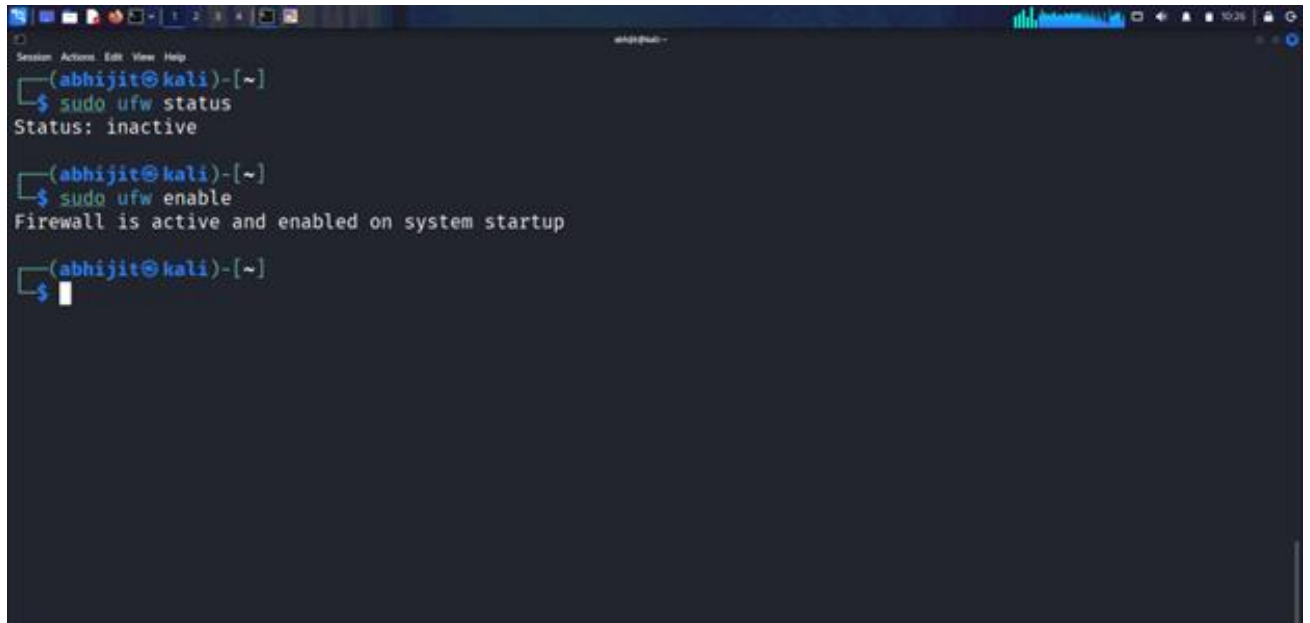
We setup and use firewall on Linux.

Open Firewall Configuration Tool

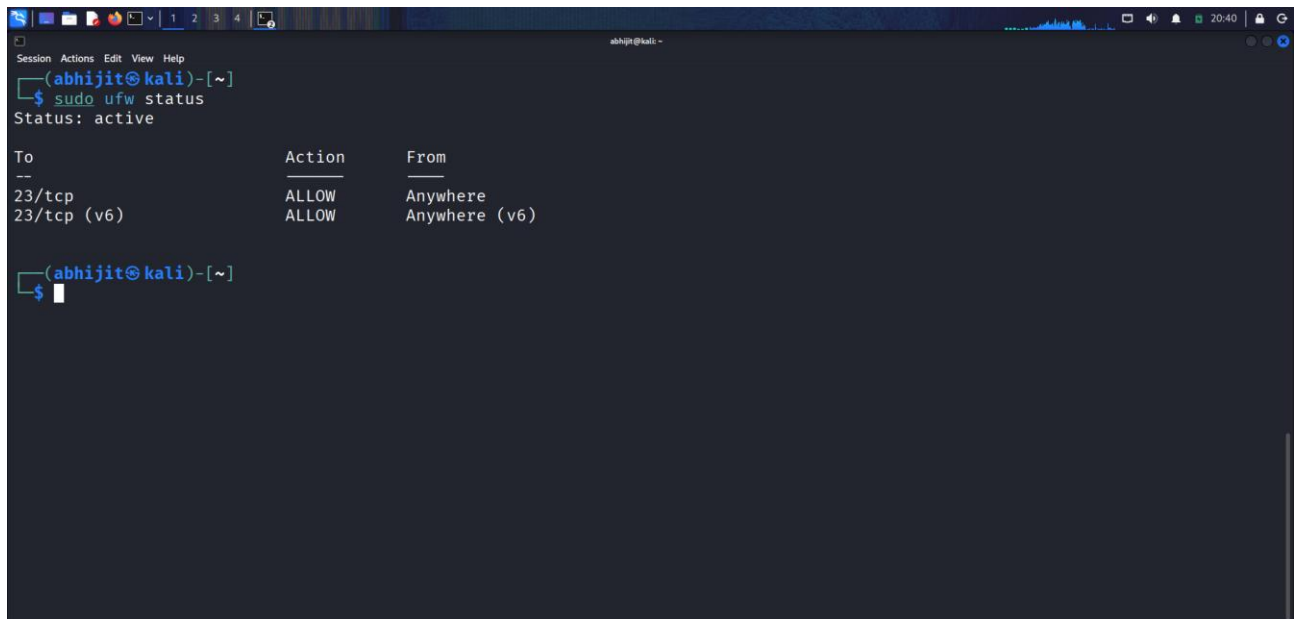
On Linux (UFW):

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 (abhijit@kali)-[~]. The user enters 'sudo ufw status' and the output is 'Status: inactive'. Then the user enters 'sudo ufw enable' and the output is 'Firewall is active and enabled on system startup'. The prompt returns to (abhijit@kali)-[~].

```
(abhijit@kali)-[~]  
$ sudo ufw status  
Status: inactive  
  
(abhijit@kali)-[~]  
$ sudo ufw enable  
Firewall is active and enabled on system startup  
  
(abhijit@kali)-[~]  
$
```

A terminal window on a Kali Linux system. The prompt is (abhijit@kali)-[~]. The user enters 'sudo ufw status' and the output is 'Status: active'. Below this, a table of rules is displayed. The prompt returns to (abhijit@kali)-[~].

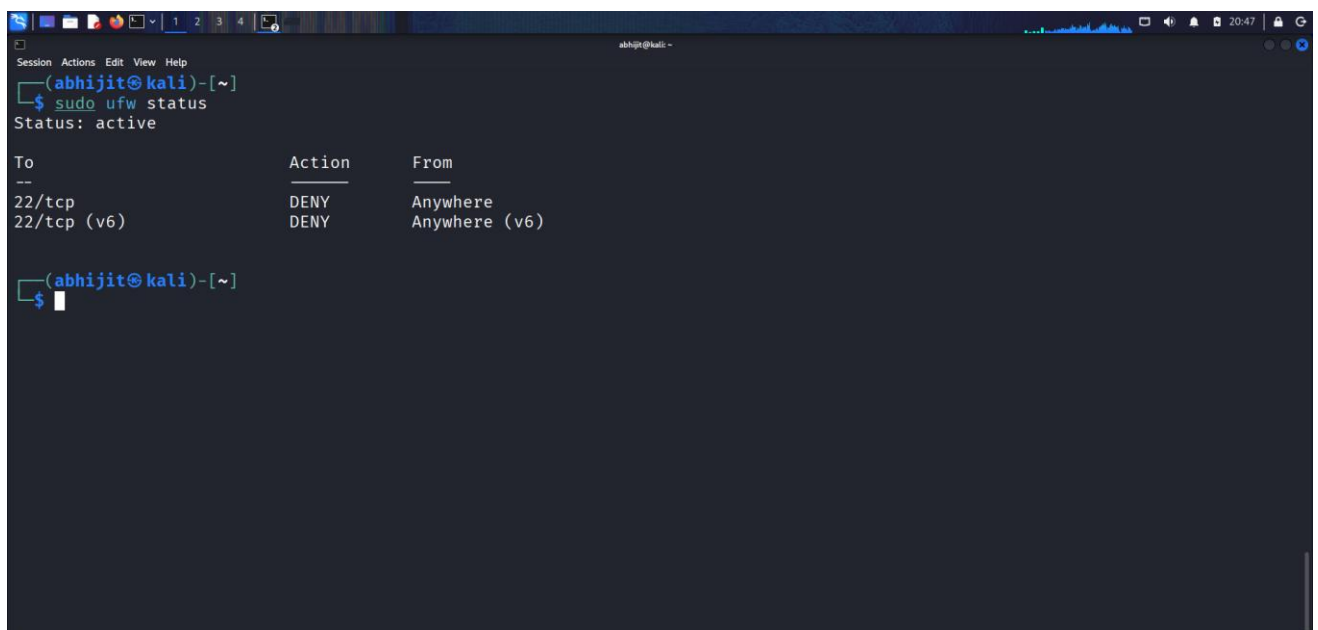
```
(abhijit@kali)-[~]  
$ sudo ufw status  
Status: active  
  
To Action From  
--  
23/tcp ALLOW Anywhere  
23/tcp (v6) ALLOW Anywhere (v6)  
  
(abhijit@kali)-[~]  
$
```

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

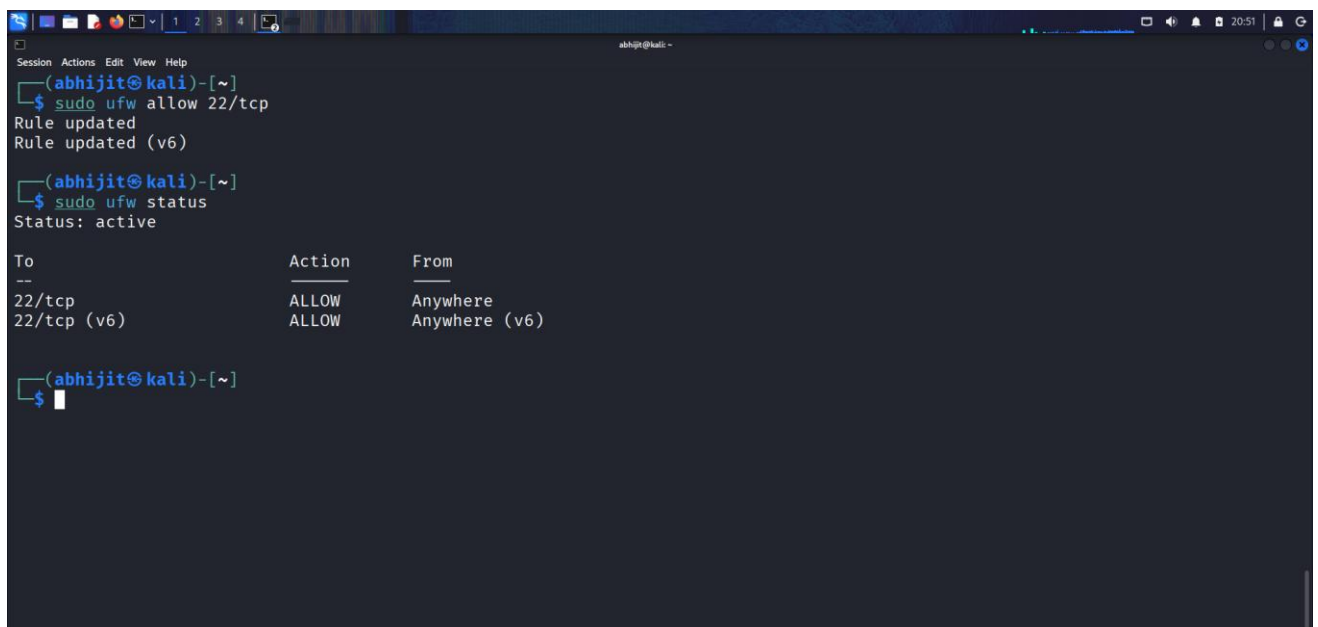


```
Session Actions Edit View Help
(abhijit@kali)~]
$ sudo ufw status
Status: active

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

(abhijit@kali)~]
$
```

Then Allow 22/tcp



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

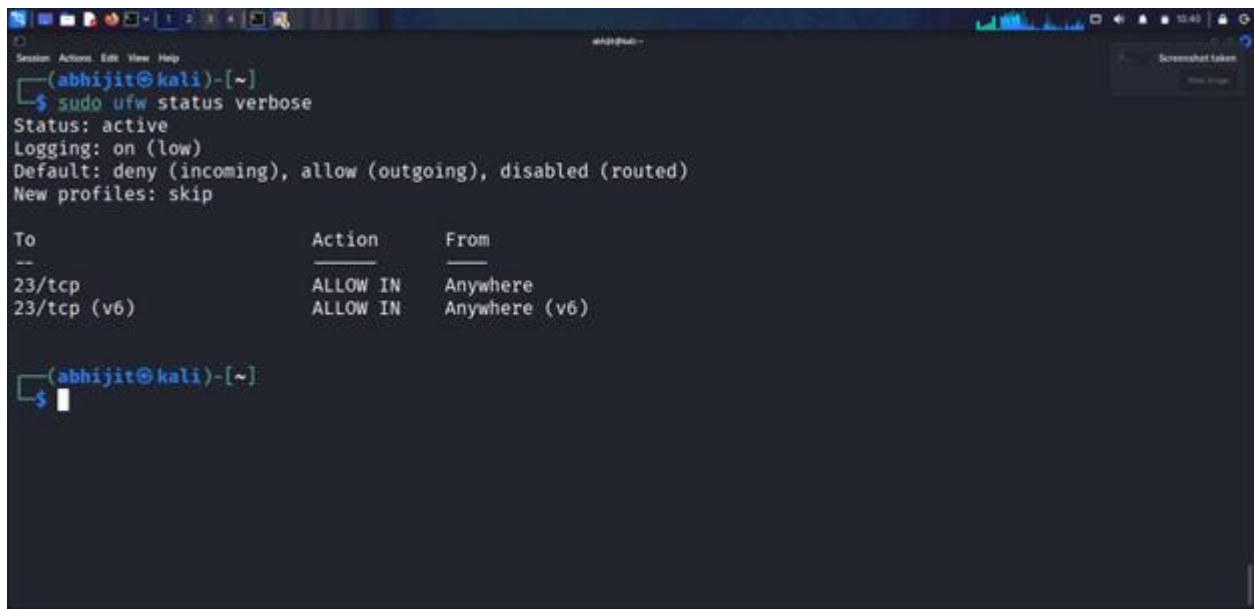
(abhijit@kali)~]
$ sudo ufw status
Status: active

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

(abhijit@kali)~]
$
```

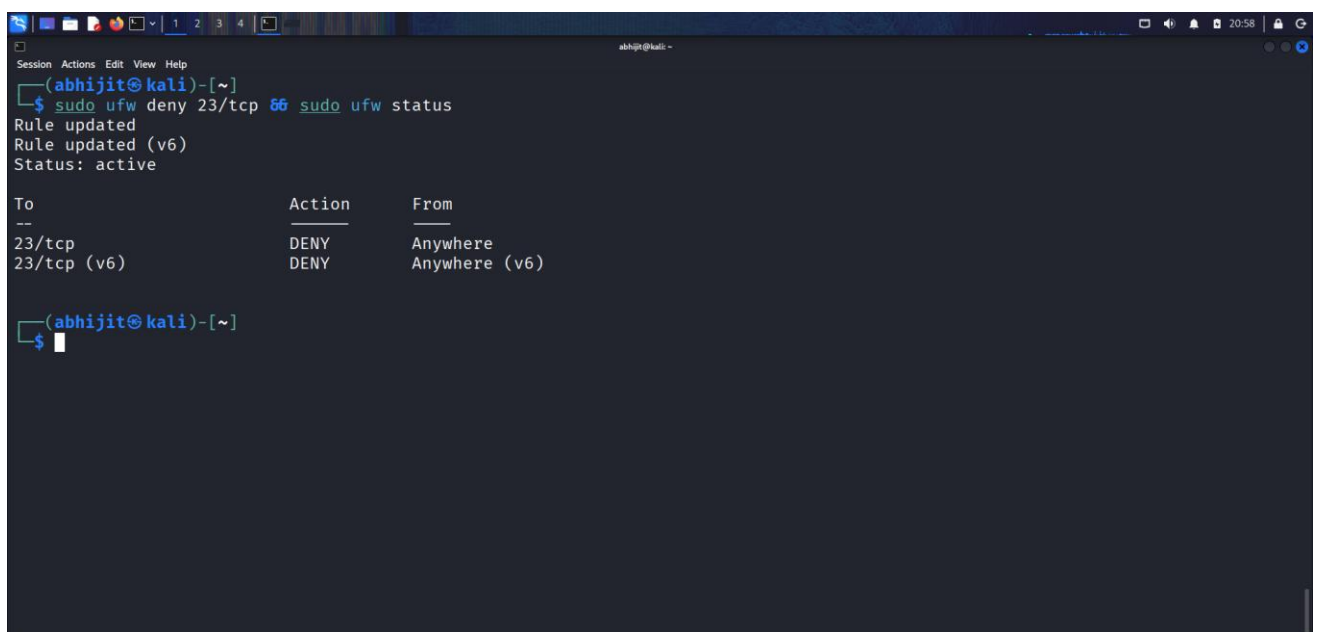
Add a Rule to Block Inbound Traffic on a Specific Port (e.g., 23 for Telnet)

On Linux (UFW):

A terminal window on a Kali Linux system showing the output of the command 'sudo ufw status verbose'. The output indicates that UFW is active, logging is on (low), the default is deny for incoming traffic, and new profiles are skipped. A table lists the current rules: port 23/tcp is allowed in from anywhere, and port 23/tcp (v6) is also allowed in from anywhere (v6).

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

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

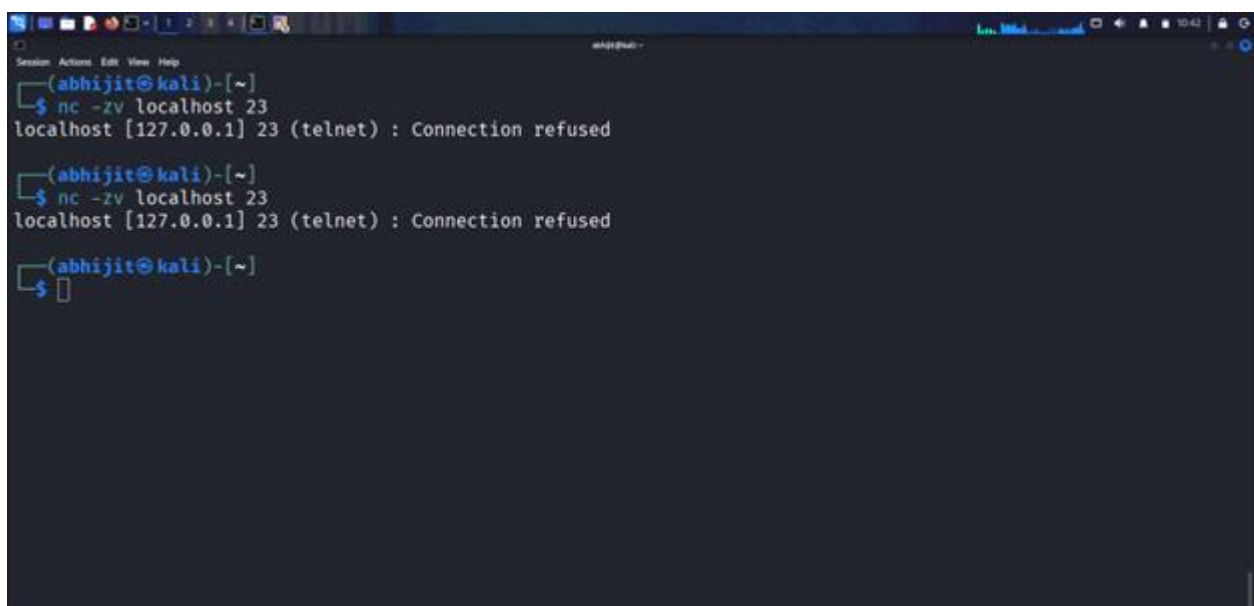
A terminal window on a Kali Linux system showing the output of the commands 'sudo ufw deny 23/tcp' and 'sudo ufw status'. The first command successfully updates the rule to deny incoming traffic on port 23/tcp. The second command shows the updated status: port 23/tcp is now denied in from anywhere, and port 23/tcp (v6) is also denied in from anywhere (v6).

```
(abhiжит@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)  
  
(abhiжит@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.



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

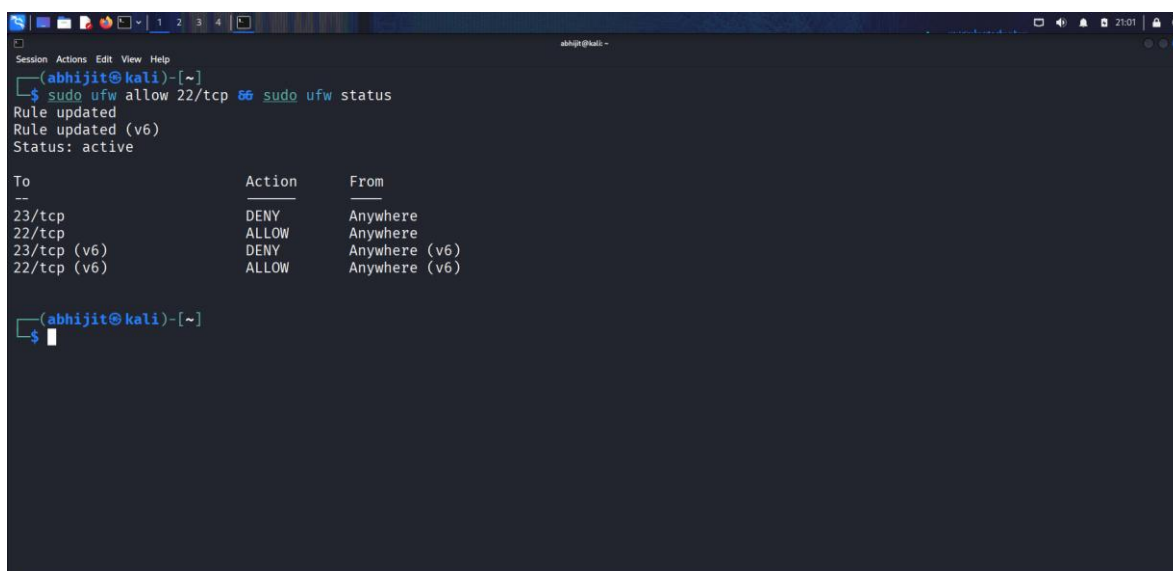
Reports:

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):

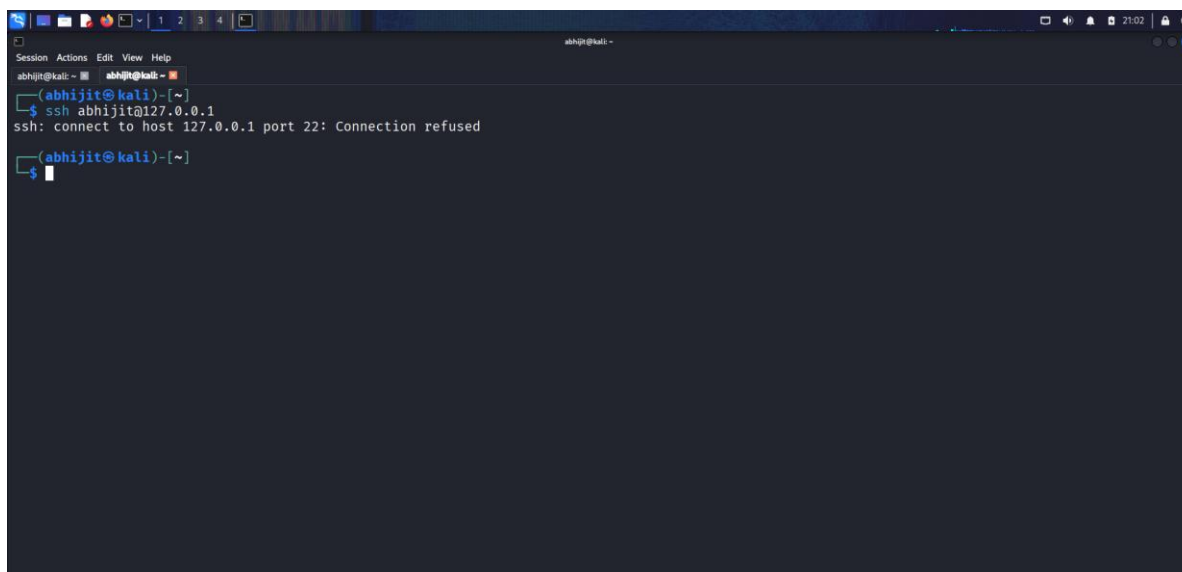
Run “**sudo ufw allow 22/tcp**” (allows inbound SSH). Confirm with “**sudo ufw status**”.

A terminal window on a Kali Linux machine. The user runs 'sudo ufw allow 22/tcp' and 'sudo ufw status'. The status output shows a table of rules. The rule for port 22/tcp is now 'ALLOW' from 'Anywhere'.

```
abhi@kali: ~  
$ sudo ufw allow 22/tcp 66 sudo ufw status  
Rule updated  
Rule updated (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  
  
abhi@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 abhi@127.0.0.1'. The output shows 'ssh: connect to host 127.0.0.1 port 22: Connection refused'.

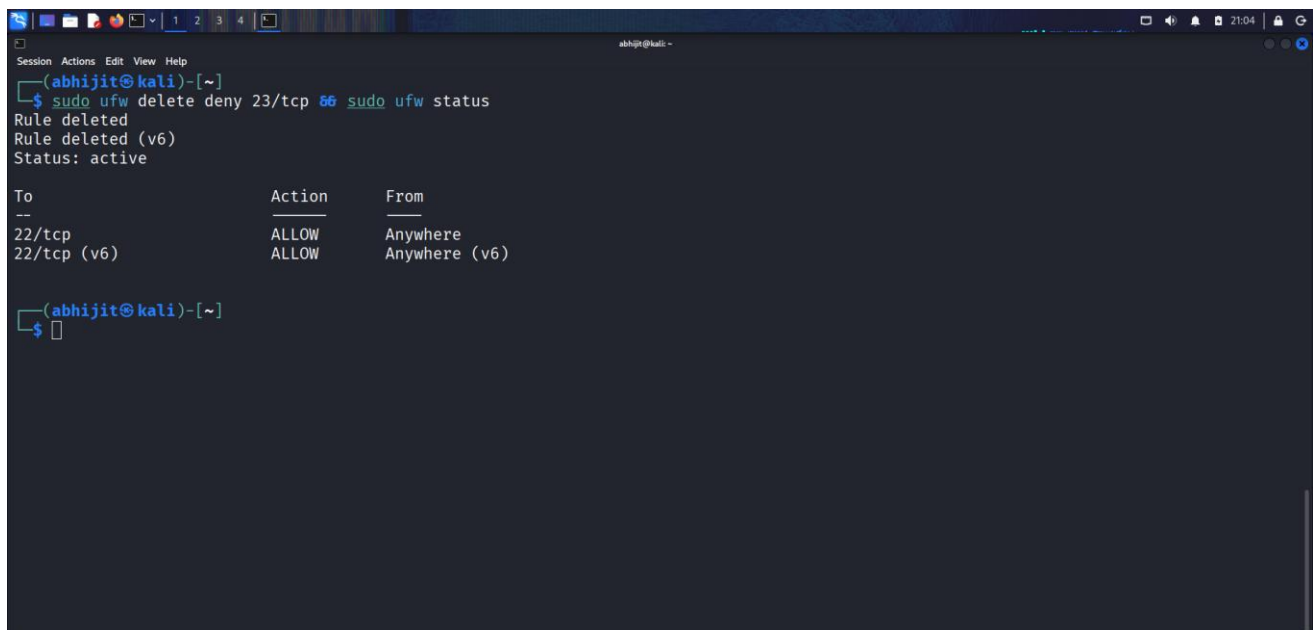
```
abhi@kali: ~  
$ ssh abhi@127.0.0.1  
ssh: connect to host 127.0.0.1 port 22: Connection refused  
  
abhi@kali: ~  
$
```

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 runs 'sudo ufw delete deny 23/tcp' and then 'sudo ufw status'. The output shows the rule is deleted and the firewall status is active. A table of active rules is displayed.

```
(abhiжит@kali)-[~]  
$ sudo ufw delete deny 23/tcp 86 sudo ufw status  
Rule deleted  
Rule deleted (v6)  
Status: active  
  
To Action From  
--  
22/tcp ALLOW Anywhere  
22/tcp (v6) ALLOW Anywhere (v6)  
  
(abhiжит@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