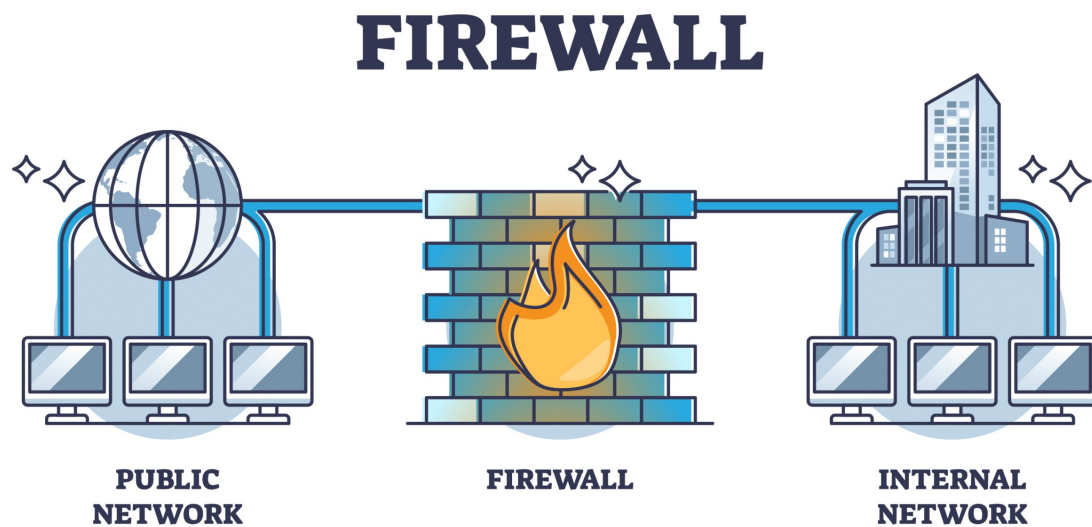


Setup and Use a Firewall on Windows/Linux



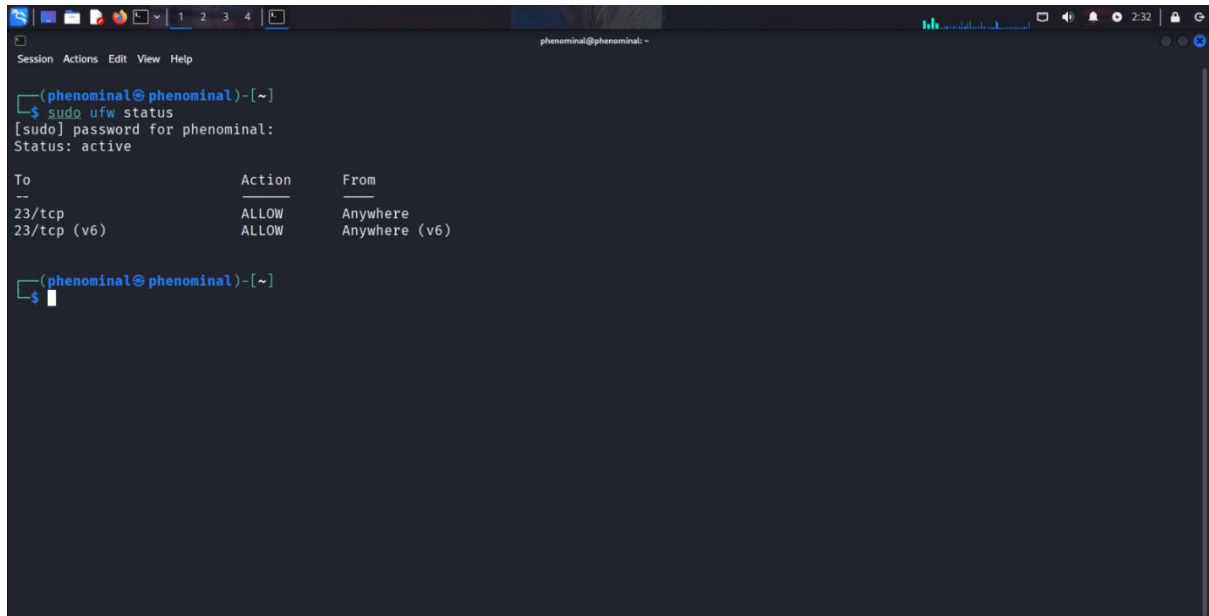
We setup and use firewall on Linux .

1. Open Firewall Configuration Tool

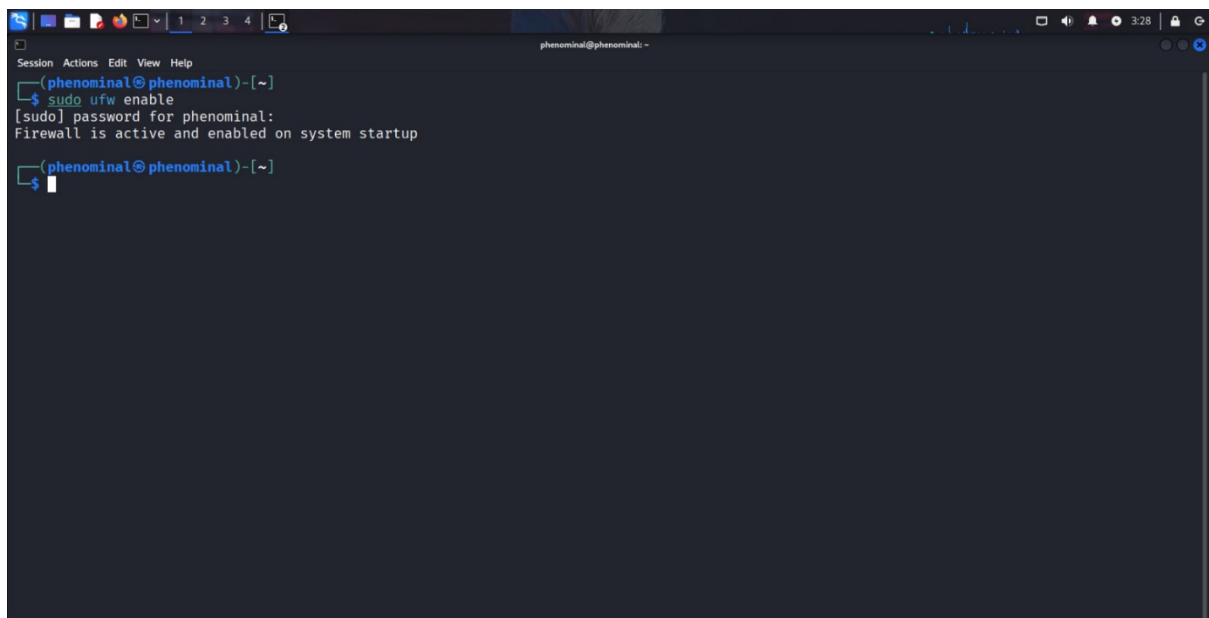
- *On Linux (UFW):*

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 titled 'phenominal@phenominal: ~' with a menu bar (Session, Actions, Edit, View, Help) and a toolbar. The terminal shows the command 'sudo ufw status' being executed. The output indicates the firewall is active and lists two rules: '23/tcp' and '23/tcp (v6)', both with an 'ALLOW' action and 'Anywhere' as the source. The prompt returns to the user after the command.

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

A terminal window titled 'phenominal@phenominal: ~' with a menu bar (Session, Actions, Edit, View, Help) and a toolbar. The terminal shows the command 'sudo ufw enable' being executed. The output indicates the firewall is active and enabled on system startup. The prompt returns to the user after the command.

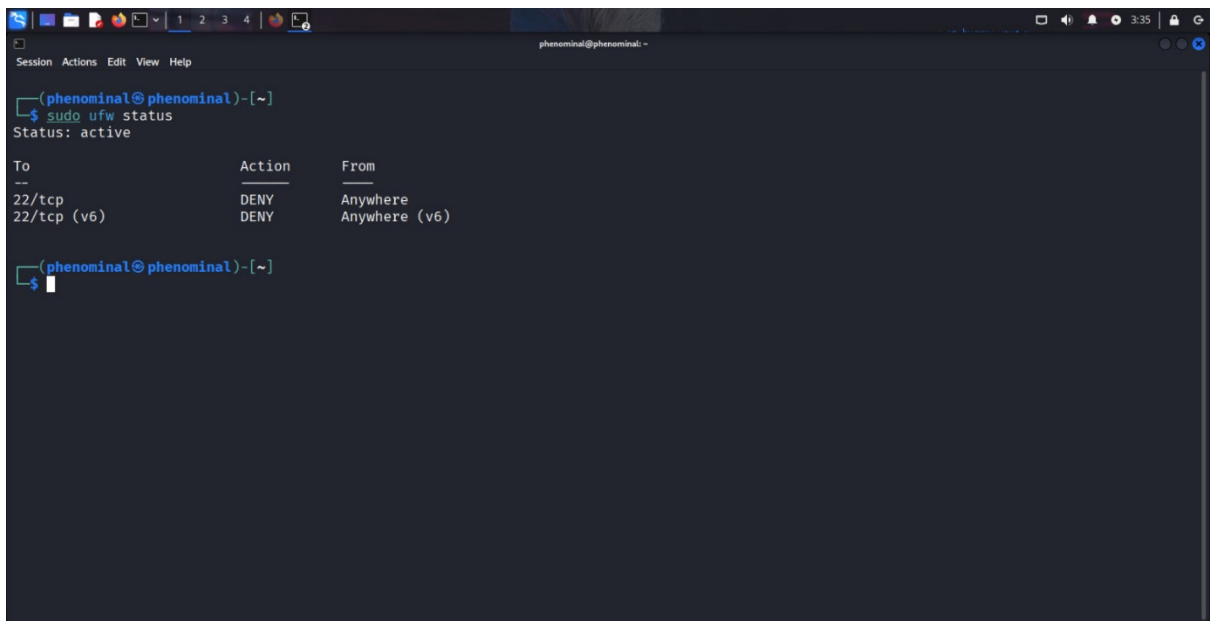
```
(phenominal@phenominal)~  
$ sudo ufw enable  
[sudo] password for phenominal:  
Firewall is active and enabled on system startup  
  
(phenominal@phenominal)~  
$
```

2. 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



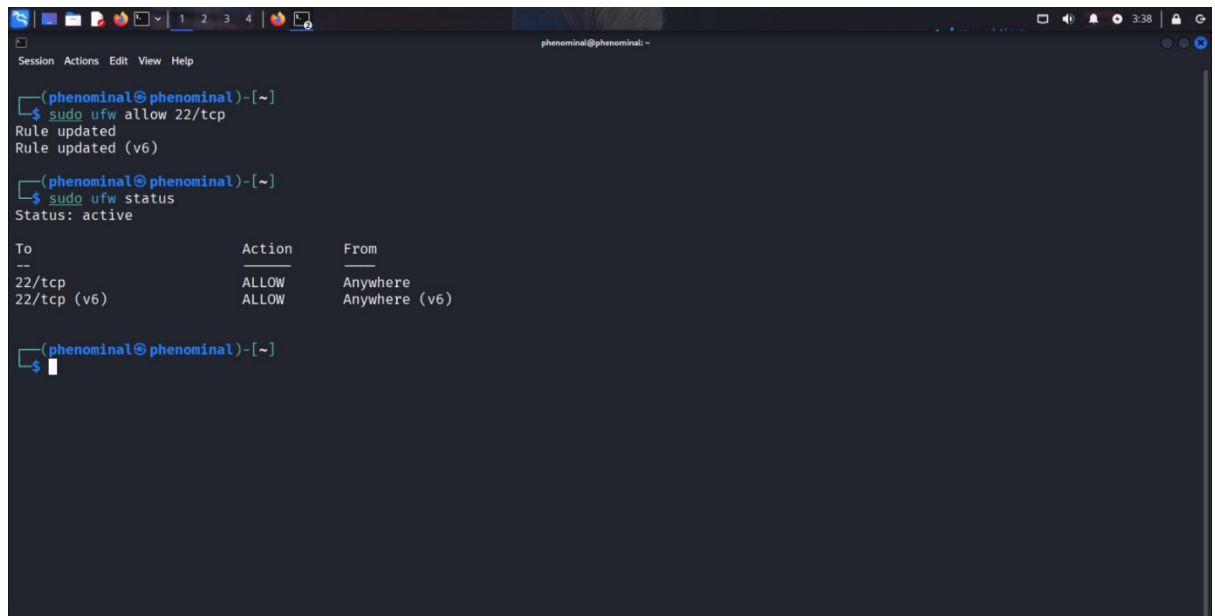
```
(phenominal@phenominal)~$ sudo ufw status
Status: active

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

(phenominal@phenominal)~$
```

The screenshot shows a terminal window with the command `sudo ufw status` executed. The output displays the UFW status as 'active' and lists two rules for port 22. Both rules are set to 'DENY' and apply to 'Anywhere' (IPv4 and IPv6).

Then Allow 22/tcp

A terminal window titled 'phenominal@phenominal: ~' showing the execution of UFW commands. The user runs 'sudo ufw allow 22/tcp', which results in 'Rule updated' and 'Rule updated (v6)'. Then, the user runs 'sudo ufw status', which shows 'Status: active' and a table of rules.

```
(phenominal@phenominal)~$ sudo ufw allow 22/tcp
Rule updated
Rule updated (v6)

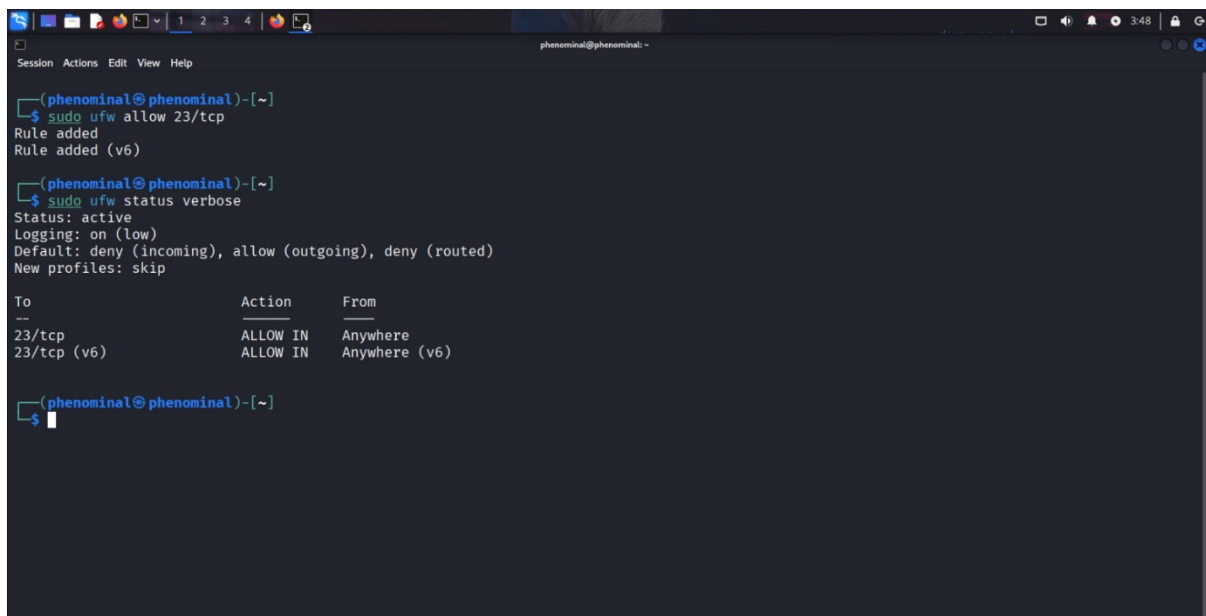
(phenominal@phenominal)~$ sudo ufw status
Status: active

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

(phenominal@phenominal)~$
```

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

- **On Linux (UFW):**

A terminal window titled 'phenominal@phenominal: ~' showing the execution of UFW commands. The user runs 'sudo ufw allow 23/tcp', which results in 'Rule added' and 'Rule added (v6)'. Then, the user runs 'sudo ufw status verbose', which shows 'Status: active', 'Logging: on (low)', 'Default: deny (incoming), allow (outgoing), deny (routed)', and 'New profiles: skip'. Finally, the user runs 'sudo ufw status', which shows a table of rules.

```
(phenominal@phenominal)~$ sudo ufw allow 23/tcp
Rule added
Rule added (v6)

(phenominal@phenominal)~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), deny (routed)
New profiles: skip

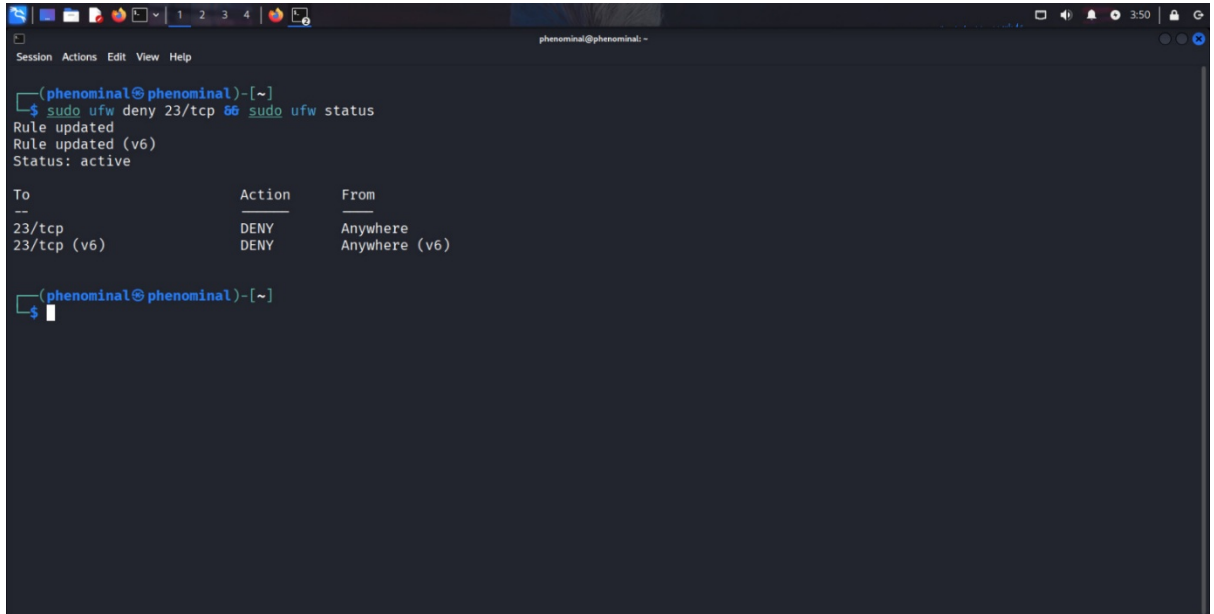
To Action From
--
23/tcp ALLOW IN Anywhere
23/tcp (v6) ALLOW IN Anywhere (v6)

(phenominal@phenominal)~$ sudo ufw status
Status: active

To Action From
--
23/tcp ALLOW IN Anywhere
23/tcp (v6) ALLOW IN Anywhere (v6)

(phenominal@phenominal)~$
```

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

A terminal window titled 'phenominal@phenominal: ~' showing the execution of ufw commands. The user runs 'sudo ufw deny 23/tcp' and 'sudo ufw status'. The output shows the rule being updated and its status as active. A table displays the rule details: To (23/tcp, 23/tcp (v6)), Action (DENY), and From (Anywhere, Anywhere (v6)).

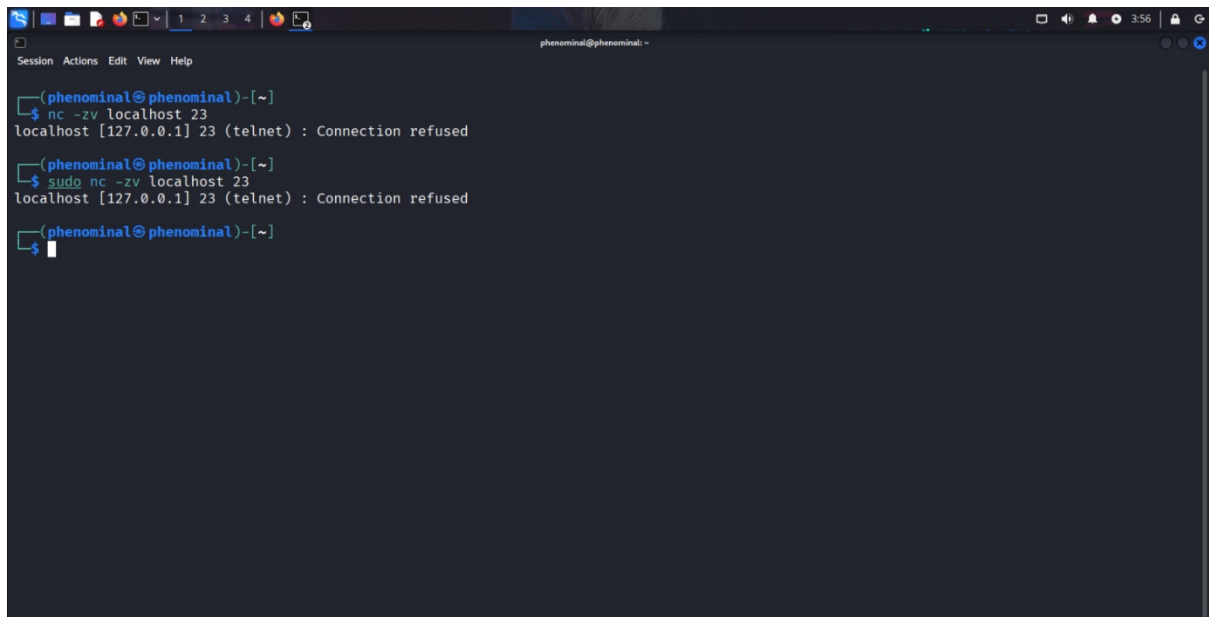
```
(phenominal@phenominal)~  
$ sudo ufw deny 23/tcp  
Rule updated  
$ sudo ufw status  
Rule updated (v6)  
Status: active  


| To          | Action | From          |
|-------------|--------|---------------|
| 23/tcp      | DENY   | Anywhere      |
| 23/tcp (v6) | DENY   | Anywhere (v6) |

  
(phenominal@phenominal)~  
$
```

4. 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 with a dark background and light blue text. The window title is 'phenominal@phenominal: ~'. The terminal shows three commands and their outputs: 1. Command: 'nc -zv localhost 23'. Output: 'localhost [127.0.0.1] 23 (telnet) : Connection refused'. 2. Command: 'sudo nc -zv localhost 23'. Output: 'localhost [127.0.0.1] 23 (telnet) : Connection refused'. 3. Command: '\$' (prompt). The terminal has a menu bar with 'Session', 'Actions', 'Edit', 'View', and 'Help'. The top of the window shows a standard Linux desktop environment with icons and a system tray.

- Reports:

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

5. 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**”.

```
phenominal@phenominal: ~  
$ sudo ufw allow 22/tcp 66 sudo ufw status  
[sudo] password for phenominal:  
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)  
  
phenominal@phenominal: ~  
$
```

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

Result:

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

Connection Refused.

6. 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 screenshot showing the execution of UFW commands. The user runs 'sudo ufw delete deny 23/tcp' and 'sudo ufw status'. The status output shows two active rules: '22/tcp' and '22/tcp (v6)', both with 'ALLOW' action and 'Anywhere' source. The terminal window has a dark background with light blue and green text. The top bar shows the user 'phenominal@phenominal' and the time '4:43'.

```
(phenominal@phenominal)-[~]
$ sudo ufw delete deny 23/tcp 66 sudo ufw status
[sudo] password for phenominal:
Rule deleted
Rule deleted (v6)
Status: active

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

(phenominal@phenominal)-[~]
$
```

7. 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.

8. 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.

Thankyou ..