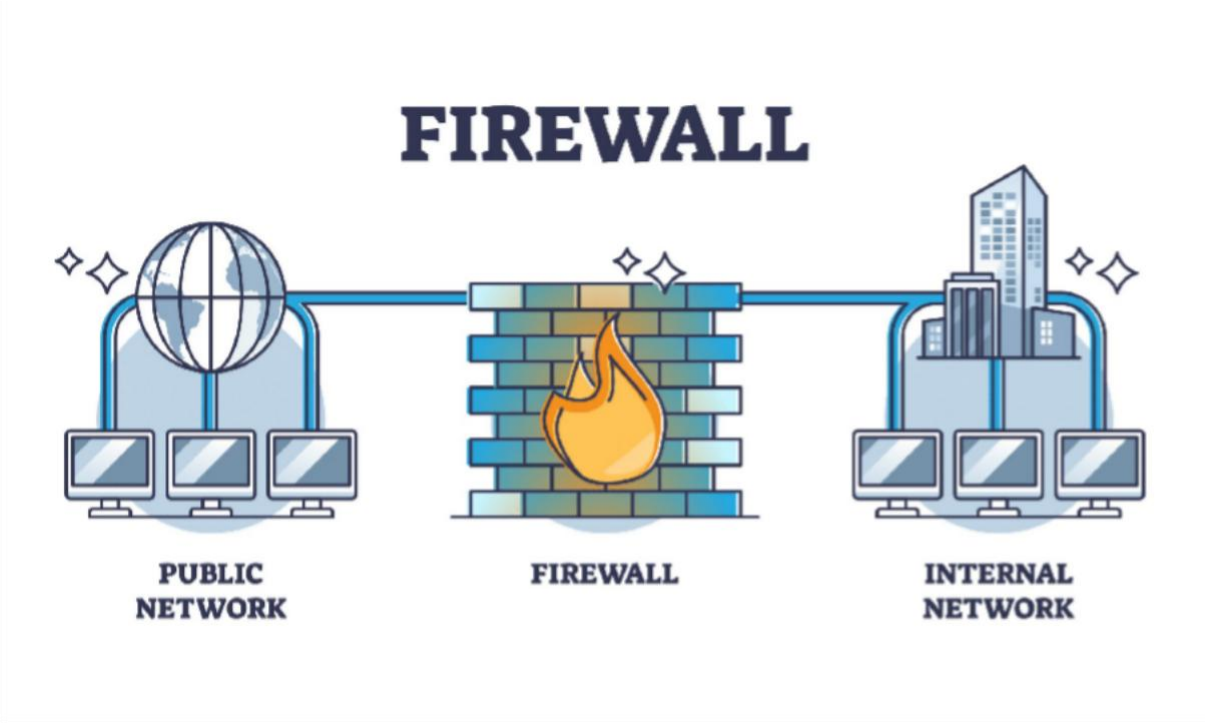
Setup and Use a Firewall on Windows/Linux



We setup and use firewall on Linux.

Open Firewall Configuration Tool

On Linux (UFW):

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

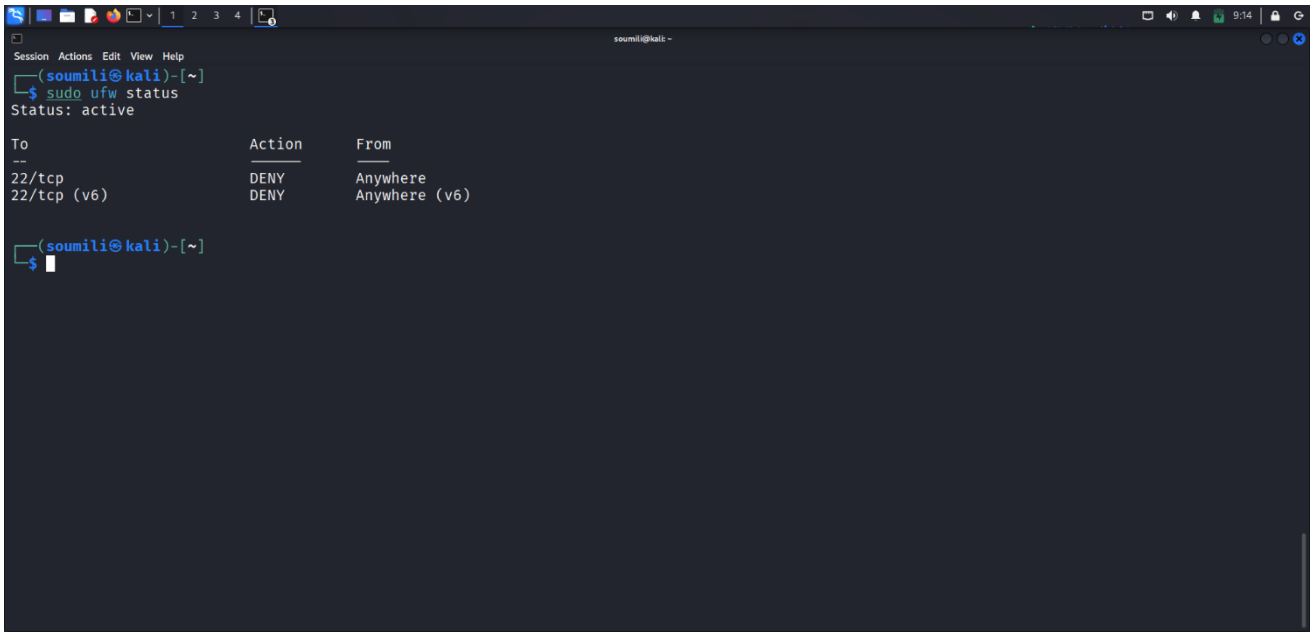
Open a terminal and run **“sudo ufw status”** to check if it's active. If not, enable it with **“sudo ufw enable”.**

List Current Firewall Rules

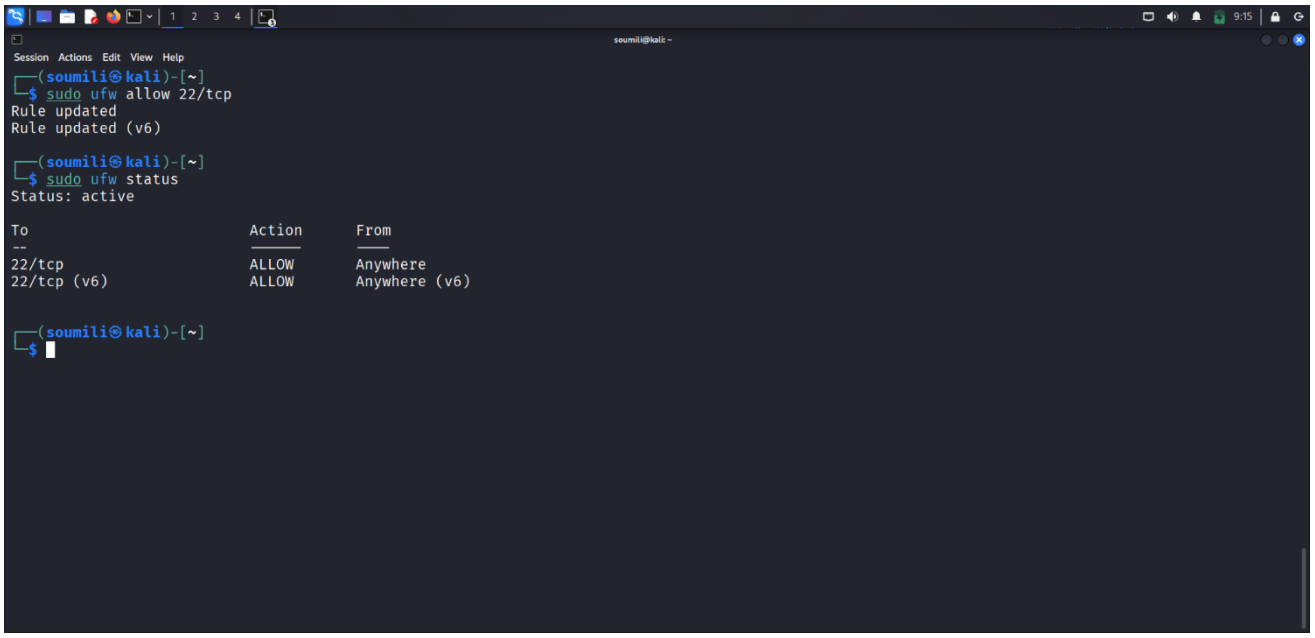
On Linux (UFW):

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



Then Allow 22/tcp



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

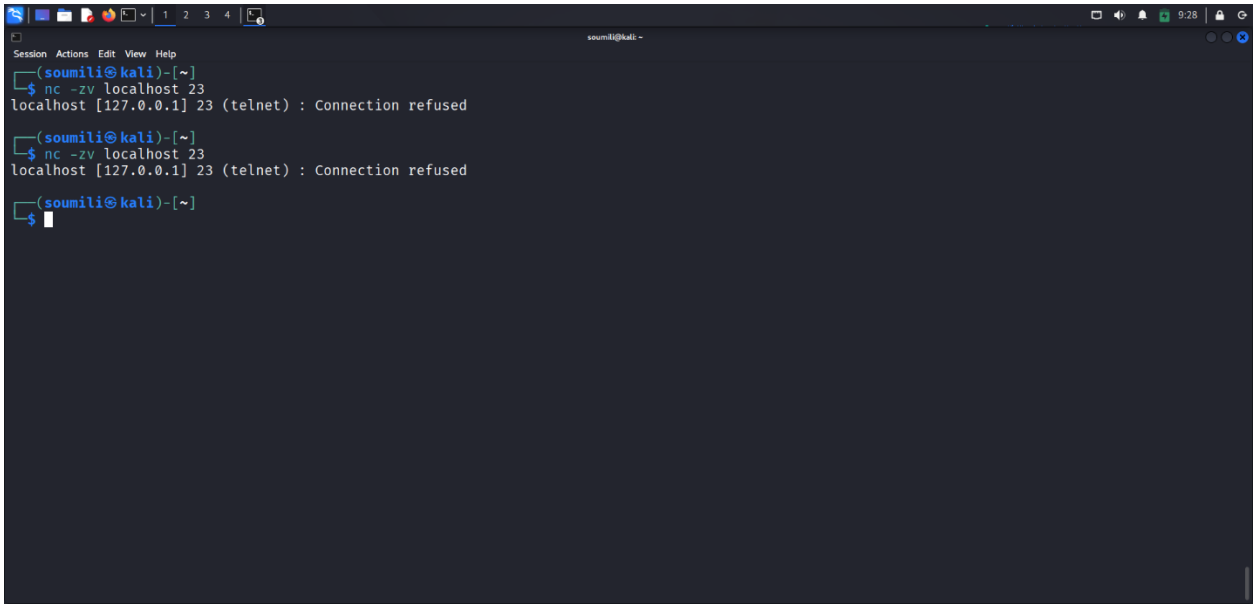
On Linux (UFW):

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

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.



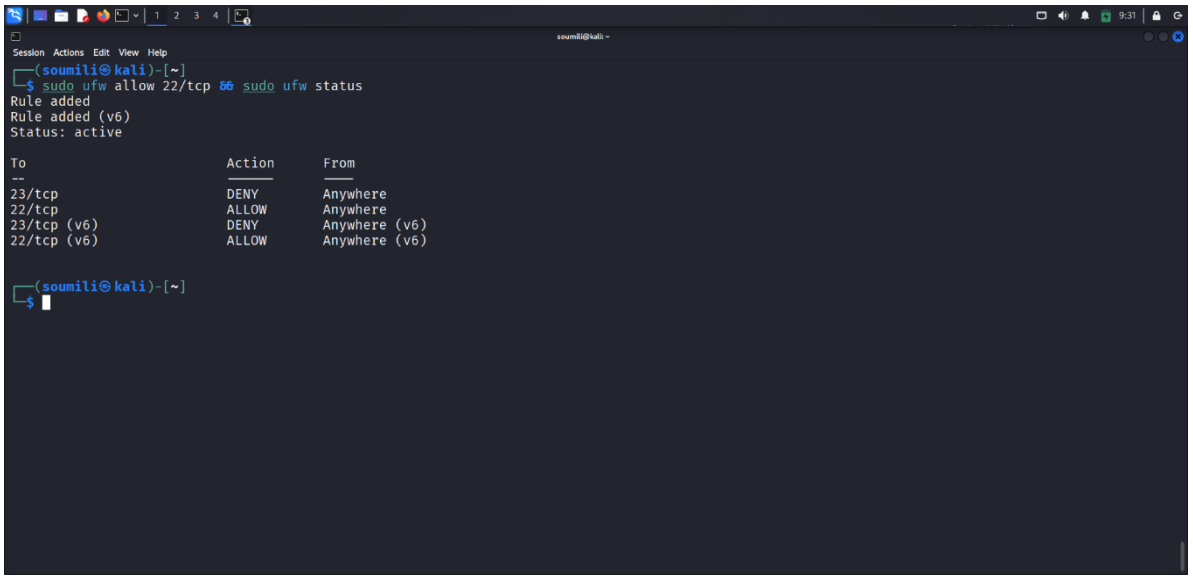
Reports:

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

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

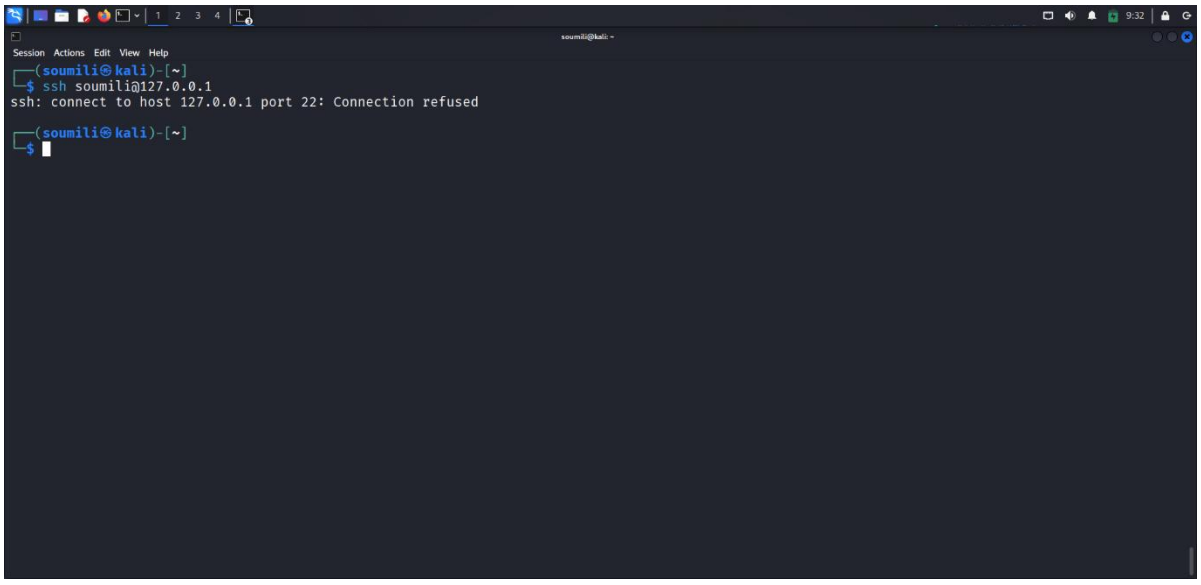
On Linux (UFW):

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



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

Result:



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”.**

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