FIREWALL AND SECURITY

Hints/Mini Guide:

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

2.List current firewall rules.

3.Add a rule to block inbound traffic on a specific port (e.g., 23 for Telnet).

4.Test the rule by attempting to connect to that port locally or remotely.

5.Add rule to allow SSH (port 22) if on Linux.

6.Remove the test block rule to restore the original state.

7.Document commands or GUI steps used.

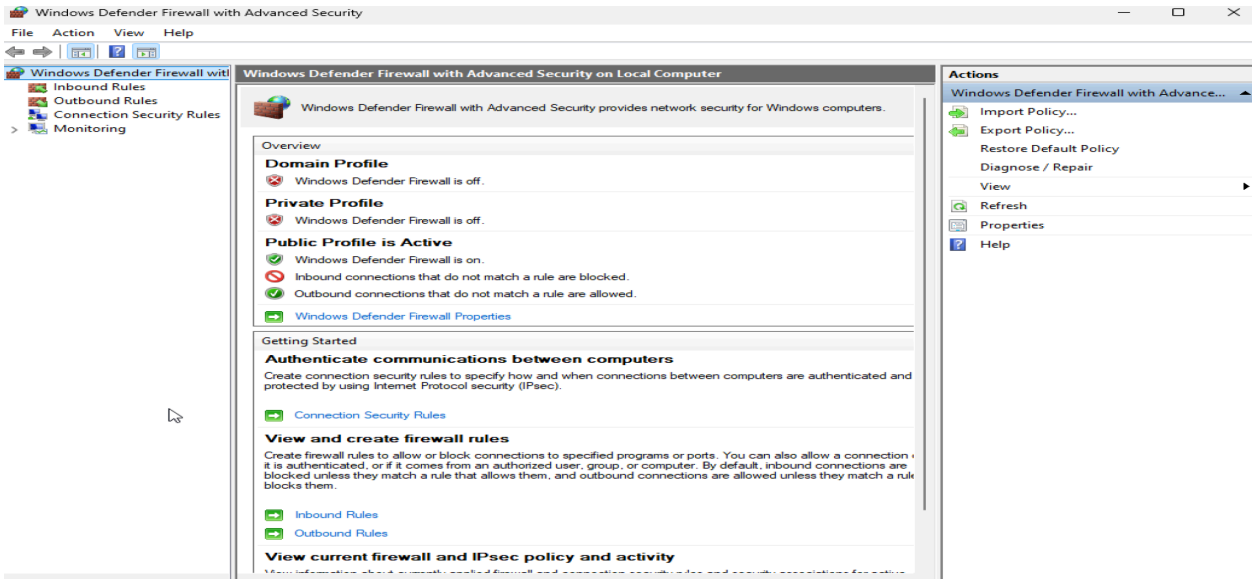
8.Summarize how a firewall filters traffic.

Outcome: Basic firewall management skills and understanding of network traffic filtering.

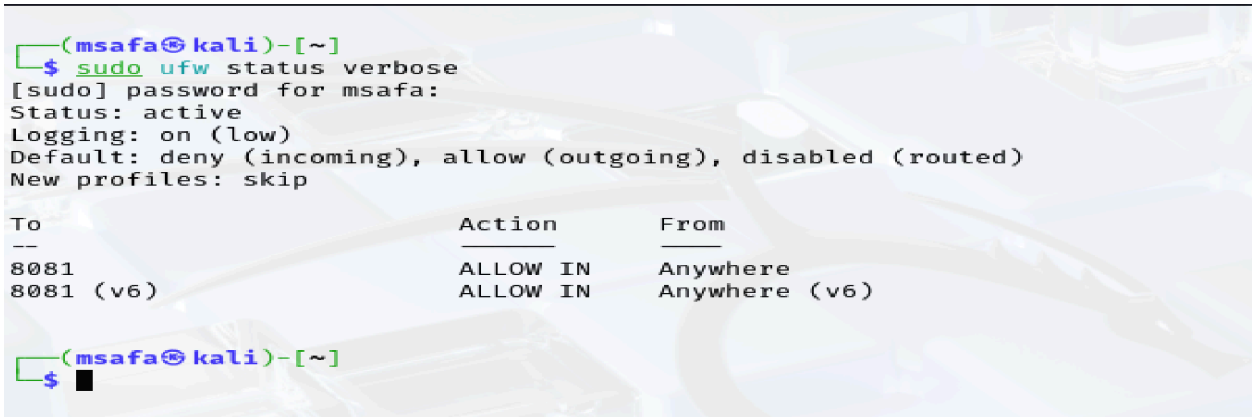
ANSWERS

1. Open firewall configuration tool

- **Windows:**
  - Go to **Control Panel → Windows Defender Firewall → Advanced Settings**.
  - Or search for **"Windows Defender Firewall with Advanced Security"** in the Start menu.



- **Linux (UFW):**
  - Open terminal and run:
    `sudo ufw status verbose`

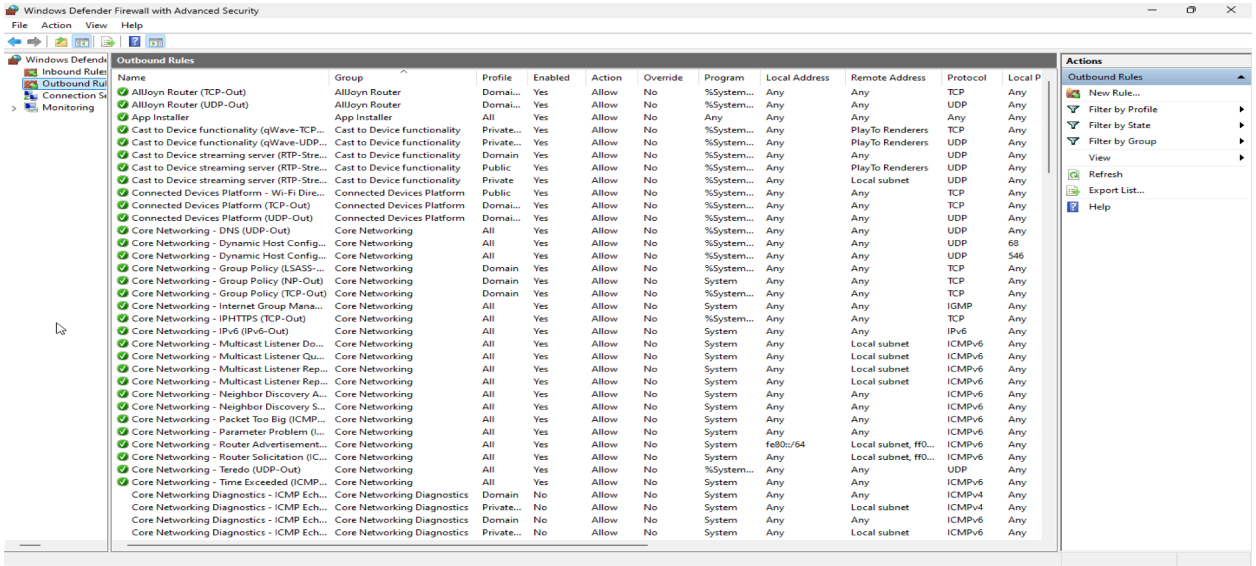


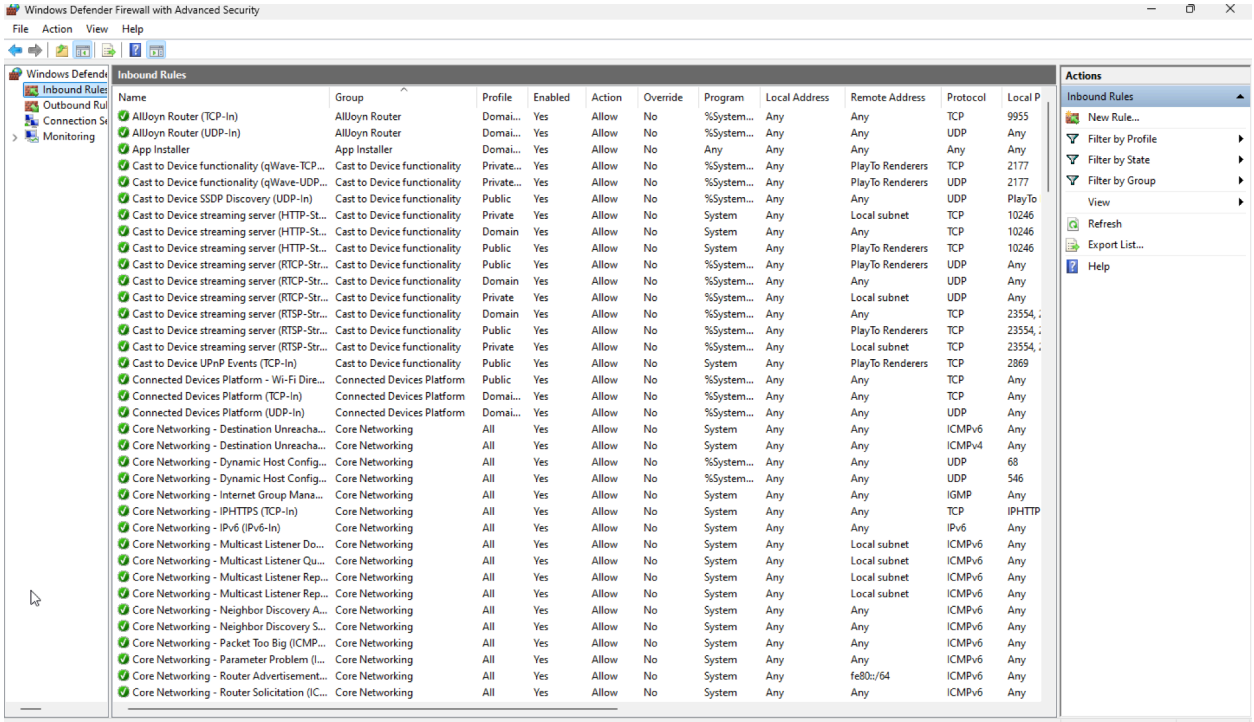
2. List current firewall rules

- **Windows:**
    - In the **Inbound Rules** and **Outbound Rules** sections, view all existing rules.
    - You can export the policy for documentation.

2. List current firewall rules

- **Windows:**
    - In the **Inbound Rules** and **Outbound Rules** sections, view all existing rules.
    - You can export the policy for documentation.

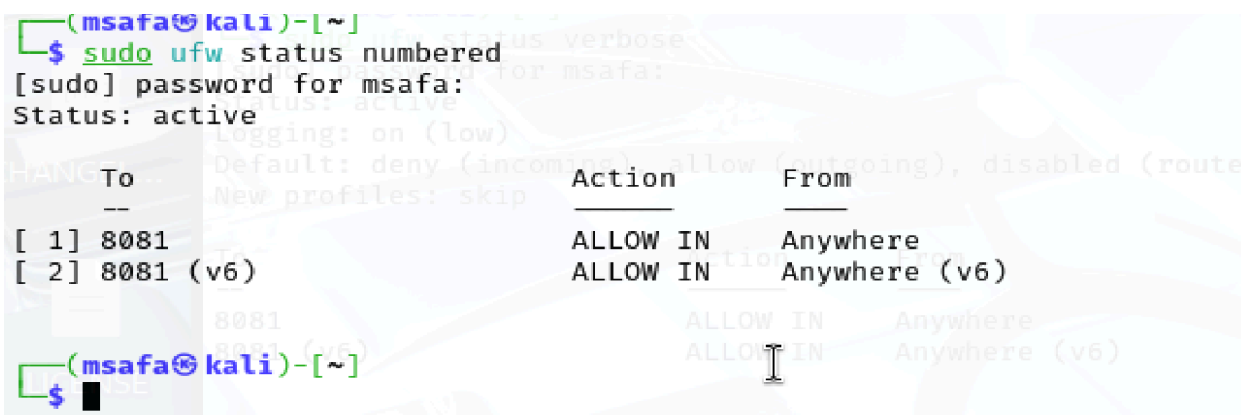




Linux (UFW):

sudo ufw status numbered

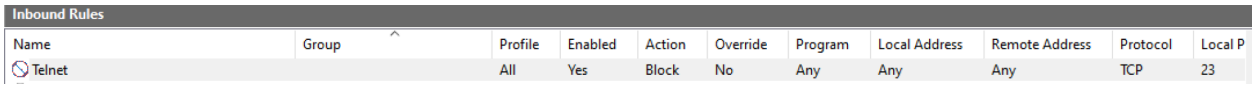
-
  - This shows each rule with its number, action (ALLOW/DENY), protocol, and port.



3. Add a rule to block inbound traffic on a specific port (e.g., 23 for Telnet)

- **Windows:**

  1. Open **Inbound Rules** → New Rule.
  2. Select **Port**, choose TCP or UDP (for Telnet, TCP).
  3. Specify **port 23**, select **Block the connection**, apply to all profiles.
  4. Name the rule **Block Telnet Port 23** and save.

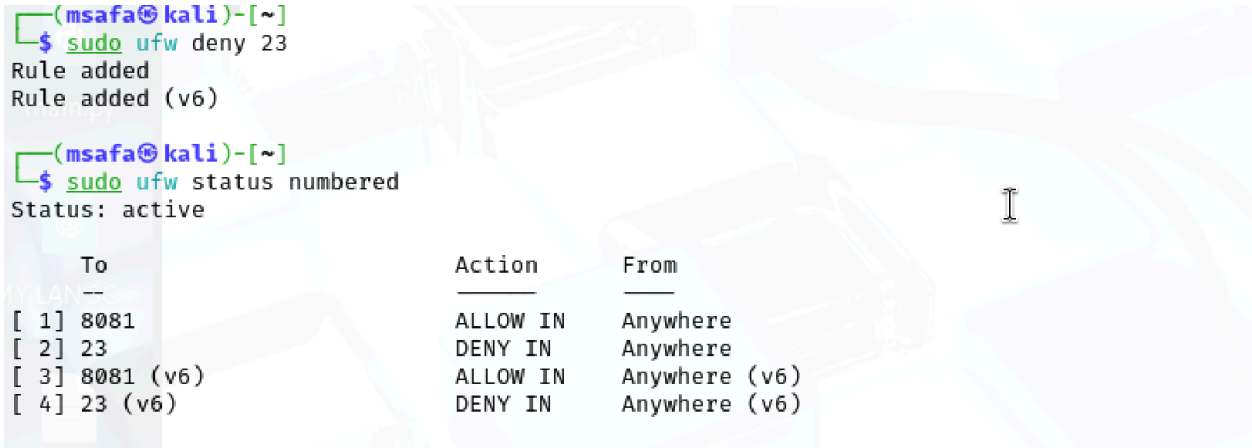


Inbound Rules

Name	Group	Profile	Enabled	Action	Override	Program	Local Address	Remote Address	Protocol	Local P
Telnet		All	Yes	Block	No	Any	Any	Any	TCP	23

Linux (UFW):

```
sudo ufw deny 23
```



```
┌──(msafa㉿kali)-[~]
└─$ sudo ufw deny 23
Rule added
Rule added (v6)

┌──(msafa㉿kali)-[~]
└─$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 8081                       ALLOW IN    Anywhere
[ 2] 23                         DENY IN     Anywhere
[ 3] 8081 (v6)                  ALLOW IN    Anywhere (v6)
[ 4] 23 (v6)                    DENY IN     Anywhere (v6)
```

4. Test the rule by attempting to connect to that port locally or remotely

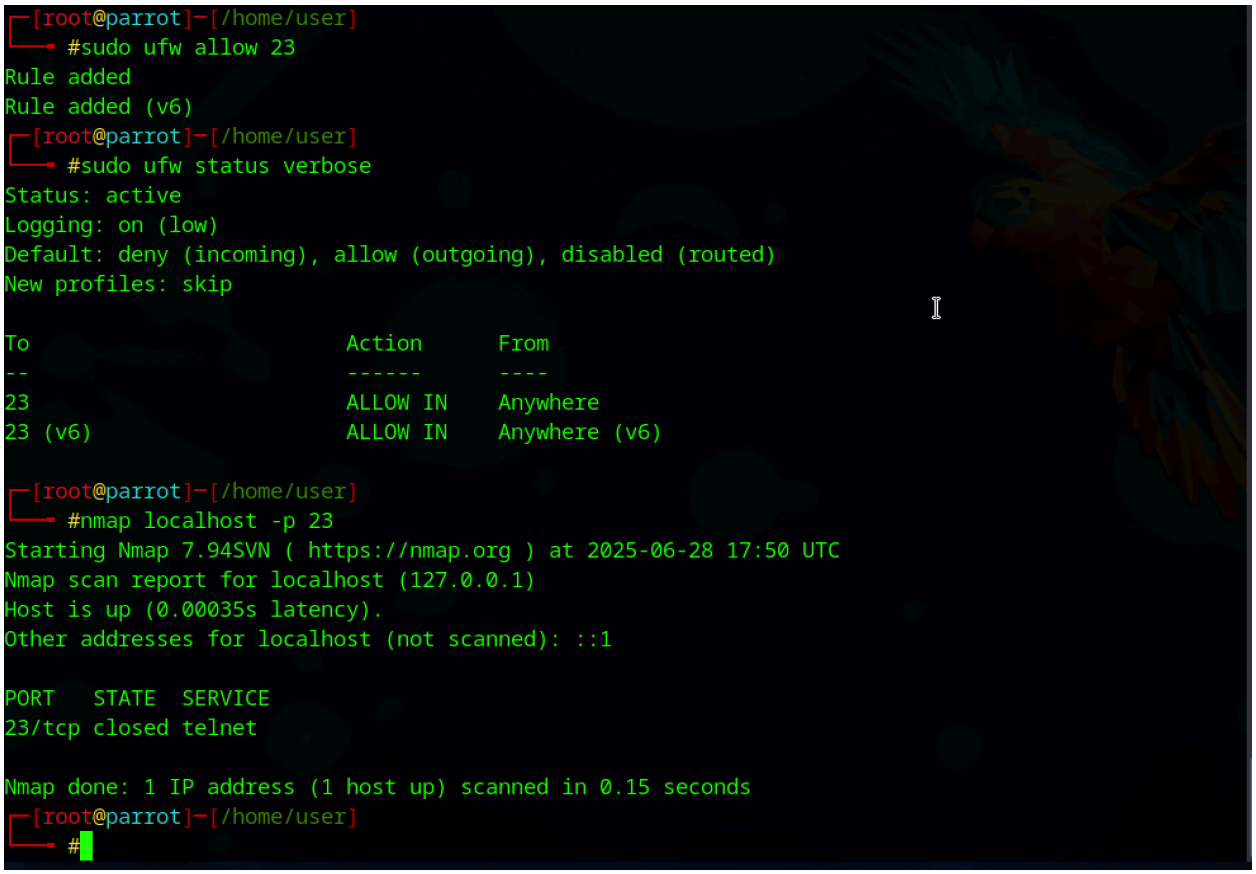
Use **telnet** or **nmap**:

```
telnet localhost 23
```

Or

```
nmap -p 23 localhost
```

- The connection should be refused or show the port as filtered/closed.



- **Result:**
  - `23/tcp closed telnet` – means your system's firewall **allows** connections on port 23, but there is **no service listening** on it.
  - So, even if someone tries to connect, they won't succeed because no Telnet server is running.

5. Add rule to allow SSH (port 22) if on Linux

To ensure remote management stays accessible:
`sudo ufw allow 22`

- This allows inbound SSH connections on port 22.

```
┌─[user@parrot]─[~]
└──╼ $sudo systemctl enable ufw
Synchronizing state of ufw.service with SysV service script with /lib/systemd/system
d-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ufw
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26
, <DATA> line 44.
insserv: Script `ssh' has overlapping Default-Start and Default-Stop runlevels (2 3
4 5) and (2 3 4 5). This should be fixed.
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26
, <DATA> line 44.
insserv: Script `ssh' has overlapping Default-Start and Default-Stop runlevels (2 3
4 5) and (2 3 4 5). This should be fixed.
┌─[user@parrot]─[~]
└──╼ $sudo systemctl status ufw
● ufw.service - Uncomplicated firewall
     Loaded: loaded (/lib/systemd/system/ufw.service; enabled; preset: enabled)
     Active: active (exited) since Sun 2025-06-29 05:21:19 UTC; 1min 24s ago
       Docs: man:ufw(8)
   Main PID: 443 (code=exited, status=0/SUCCESS)
        CPU: 33ms

Jun 29 10:51:19 parrot systemd[1]: Starting ufw.service - Uncomplicated firewal>
Jun 29 05:21:19 parrot systemd[1]: Finished ufw.service - Uncomplicated firewal>
┌─[user@parrot]─[~]
└──╼ $sudo systemctl start ufw
┌─[user@parrot]─[~]
└──╼ $sudo ufw allow 22
Skipping adding existing rule
Skipping adding existing rule (v6)
```

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

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

┌─[user@parrot]─[~]
└──╼ $nmap localhost -p 22
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-29 05:25 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000097s latency).
Other addresses for localhost (not scanned): ::1

PORT   STATE  SERVICE
22/tcp closed ssh

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
┌─[user@parrot]─[~]
└──╼ $suod systemctl start ssh
bash: suod: command not found
┌─[✗]─[user@parrot]─[~]
└──╼ $sudo systemctl start ssh
```

- You used ufw to allow inbound SSH on **port 22** → the rule is there.
- But your nmap scan still shows **port 22/tcp** as **closed**.
- That means: **Your firewall is not blocking it — but there's no SSH service running and listening on port 22.**
- **Start ssh service**

```
┌─[✗]─[user@parrot]─[~]
└──╼ $sudo systemctl start ssh
┌─[user@parrot]─[~]
└──╼ $sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 44.
insserv: warning: current start runlevel(s) (empty) of script `ssh' overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (2 3 4 5) of script `ssh' overrides LSB defaults (empty).
insserv: Script `ssh' has overlapping Default-Start and Default-Stop runlevels (2 3 4 5) and (2 3 4 5). This s
hould be fixed.
insserv: warning: current start runlevel(s) (empty) of script `ssh' overrides LSB defaults (2 3 4 5).
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 44.
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/systemd/system/ssh.service.
┌─[user@parrot]─[~]
└──╼ $nmap localhost -p 22
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-29 05:27 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00015s latency).
Other addresses for localhost (not scanned): ::1

PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
┌─[user@parrot]─[~]
└──╼ $
```

Now you can see the port is open and can be connected.

6. Remove the test block rule to restore original state

- **Windows:**
  - **Right-click the Block Telnet Port 23 rule → Delete.**

Linux (UFW):

```
sudo ufw status numbered

sudo ufw delete <rule_number>
```

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

     To                         Action      From
     --                         ------      ----
[ 1] 22                         ALLOW IN    Anywhere
[ 2] 22 (v6)                    ALLOW IN    Anywhere (v6)

┌─[user@parrot]─[~]
└──╼ $sudo ufw delete 1 2
Deleting:
 allow 22
Proceed with operation (y|n)? y
Rule deleted
┌─[user@parrot]─[~]
└──╼ $sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22 (v6)                    ALLOW IN    Anywhere (v6)

┌─[user@parrot]─[~]
└──╼ $sudo ufw delete 1
Deleting:
 allow 22
Proceed with operation (y|n)? y
Rule deleted (v6)
┌─[user@parrot]─[~]
└──╼ $
```

Summarize how a firewall filters traffic

✅ **Answer:**
A firewall is a security system that monitors and controls **incoming and outgoing network traffic** based on predefined security rules.

- **Host-based firewalls** (like Windows Firewall or UFW) act as a barrier between the trusted local system and untrusted networks.
- They use **rules** to filter packets by:
  - **Source IP address**
  - **Destination IP address**
  - **Protocol (TCP/UDP)**
  - **Port numbers**
  - **Direction (inbound/outbound)**

A firewall uses an **Access Control List (ACL)** to decide whether to **allow**, **deny**, or **drop** packets.
This minimizes the attack surface by blocking unwanted services and only permitting essential connections — enforcing the **principle of least privilege** and protecting systems from unauthorized access or exploits.

📌 Quick Example Summary

What it does	How it works
Allow traffic	If source, destination, and port match allowed rule.
Block/deny traffic	If the packet matches a block rule or no allow rule exists.
Log traffic (optional)	Some firewalls log matches for auditing.

Thanks