NETWORK SECURITY

FIREWALL IMPLEMENTATION & USAGE

Abstract

Implement the Firewall Rules and Understand them how they work and secure the endpoints and network from intrusions to protect the network infrastructure of organization to protect the confidential and sensitive information from threat actors

Abhinav Tiwari

Setup and Use a Firewall on Windows/Linux

Objective

The objective of this task is to configure and test basic firewall rules on both Windows Firewall and Linux UFW. The purpose is to understand how firewalls filter traffic, manage ports, and secure systems by controlling inbound and outbound communication.

Tools Used

1. Windows Firewall with Advanced Security
2. Command Prompt / PowerShell (for firewall commands)
3. Linux UFW (Uncomplicated Firewall)

Procedure

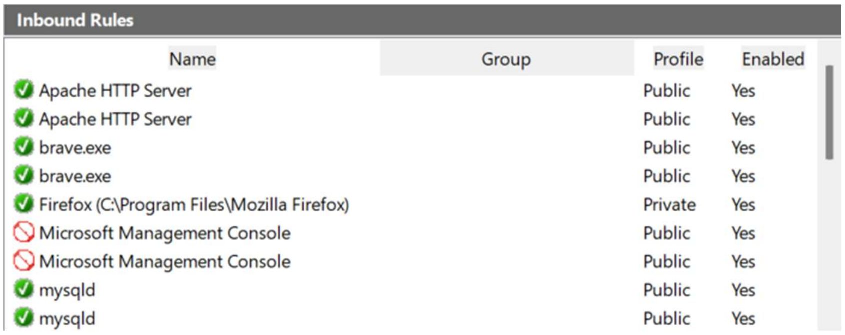
Windows Firewall Configuration

1. Open Windows Firewall with Advanced Security.

   Win + R -> wf.msc

2. Listed all current inbound and outbound rules.

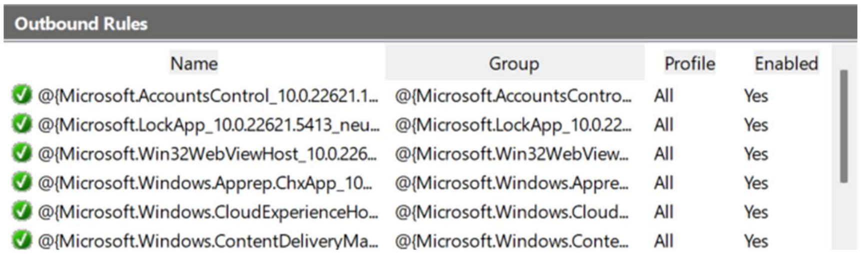
   Inbound Rules



Inbound Rules

Name	Group	Profile	Enabled
✓ Apache HTTP Server		Public	Yes
✓ Apache HTTP Server		Public	Yes
✓ brave.exe		Public	Yes
✓ brave.exe		Public	Yes
✓ Firefox (C:\Program Files\Mozilla Firefox)		Private	Yes
⊘ Microsoft Management Console		Public	Yes
⊘ Microsoft Management Console		Public	Yes
✓ mysqld		Public	Yes
✓ mysqld		Public	Yes

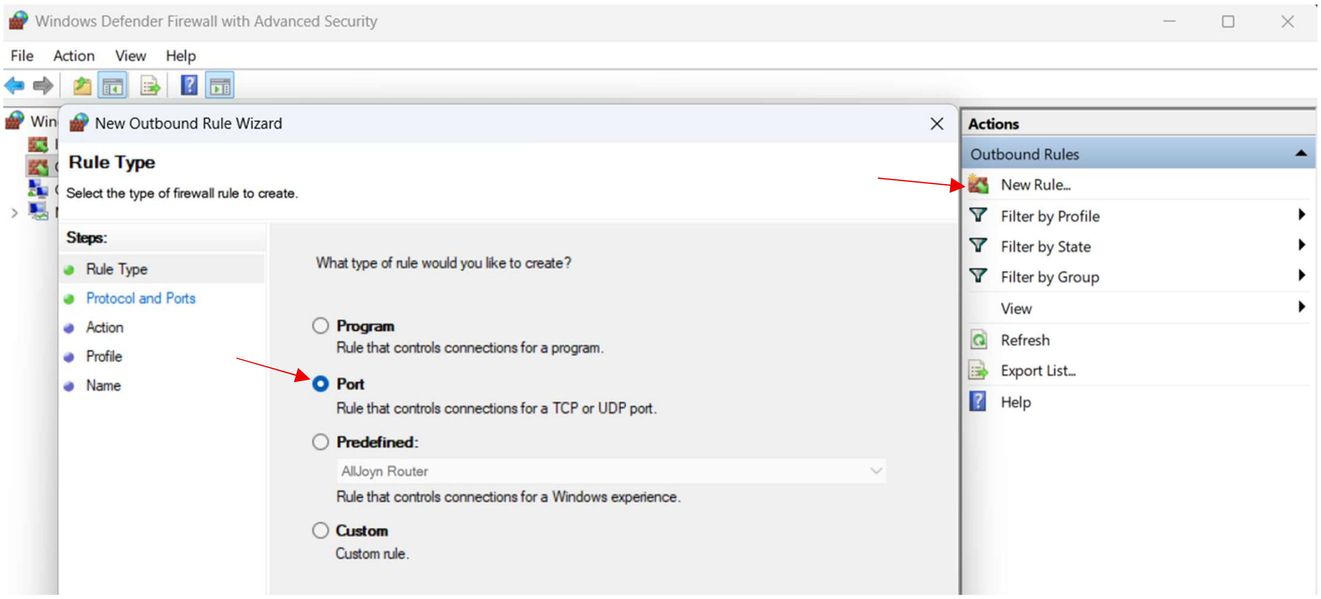
   Outbound Rules



Outbound Rules

Name	Group	Profile	Enabled
✓ @{Microsoft.AccountsControl_10.0.22621.1...	@{Microsoft.AccountsContro...	All	Yes
✓ @{Microsoft.LockApp_10.0.22621.5413_neu...	@{Microsoft.LockApp_10.0.22...	All	Yes
✓ @{Microsoft.Win32WebViewHost_10.0.226...	@{Microsoft.Win32WebView...	All	Yes
✓ @{Microsoft.Windows.Apprep.ChxApp_10...	@{Microsoft.Windows.Appre...	All	Yes
✓ @{Microsoft.Windows.CloudExperienceHo...	@{Microsoft.Windows.Cloud...	All	Yes
✓ @{Microsoft.Windows.ContentDeliveryMa...	@{Microsoft.Windows.Conte...	All	Yes

3. Added a **new inbound rule** to block Telnet (Port 23).
    - **Steps:**
        - New Rule → Port → TCP → Port 23 → Block Connection → Apply to Domain, Private, Public → Name it Block_Telnet.



New Outbound Rule Wizard

Rule Type

Select the type of firewall rule to create.

Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

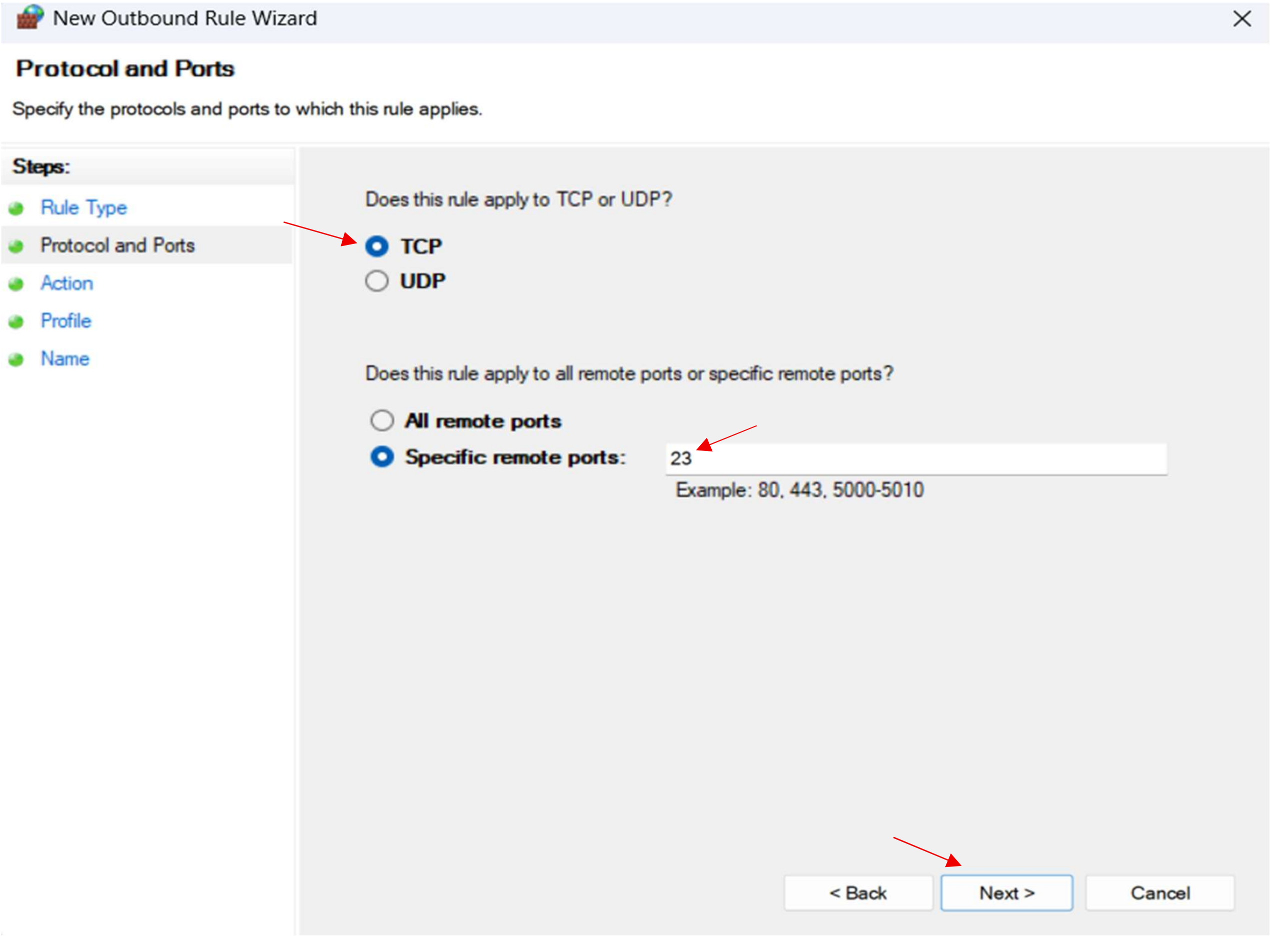
What type of rule would you like to create?

○ **Program**
Rule that controls connections for a program.

● **Port**
Rule that controls connections for a TCP or UDP port.

○ **Predefined:**
AllJoyn Router
Rule that controls connections for a Windows experience.

○ **Custom**
Custom rule.



New Outbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

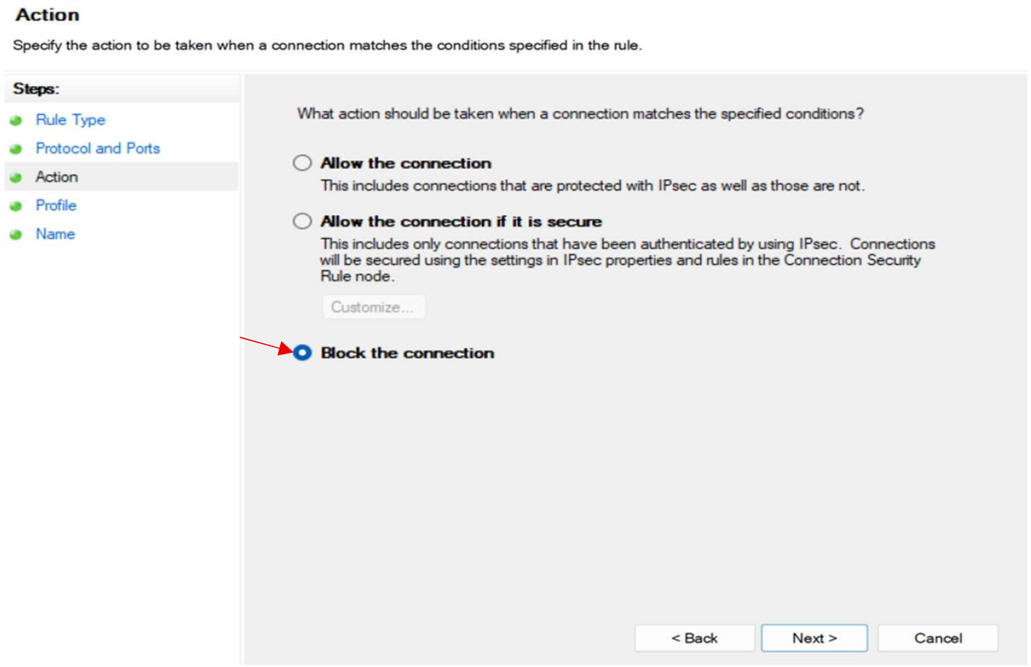
Steps:
- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Does this rule apply to TCP or UDP?

● **TCP**
○ **UDP**

Does this rule apply to all remote ports or specific remote ports?

○ **All remote ports**
● **Specific remote ports:** 23
Example: 80, 443, 5000-5010



Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

What action should be taken when a connection matches the specified conditions?

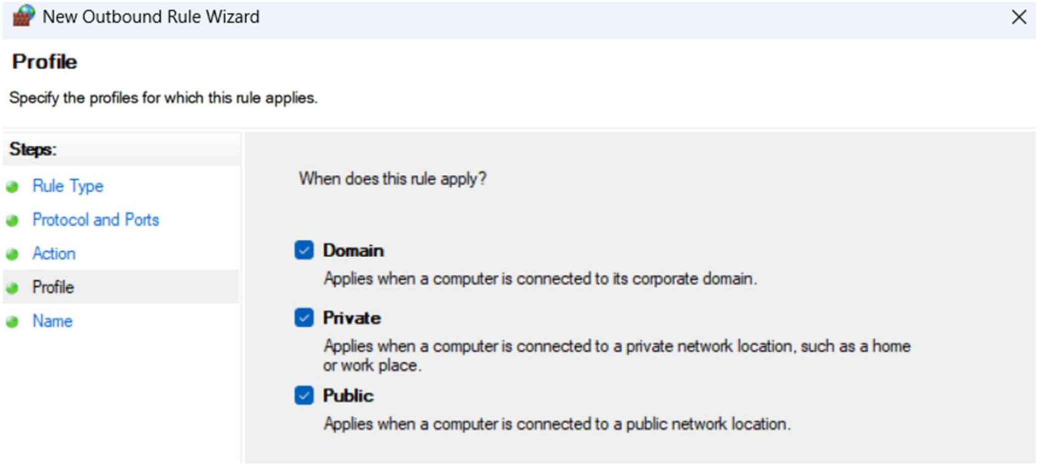
○ **Allow the connection**

This includes connections that are protected with IPsec as well as those are not.

○ **Allow the connection if it is secure**

This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

● **Block the connection**



Profile

Specify the profiles for which this rule applies.

When does this rule apply?

☑ **Domain**

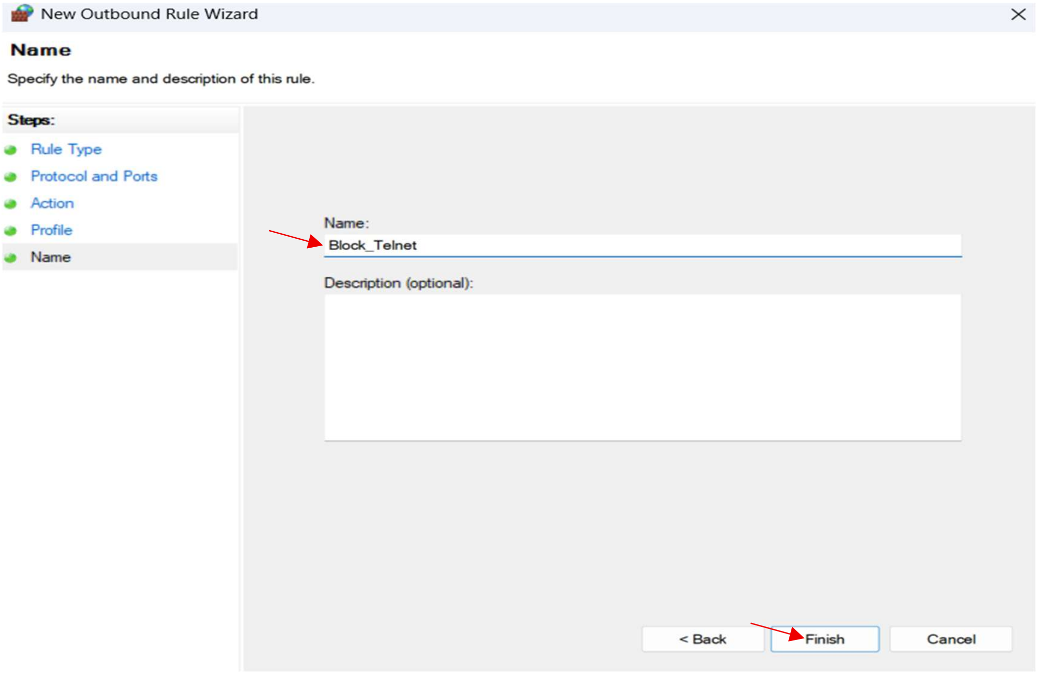
Applies when a computer is connected to its corporate domain.

☑ **Private**

Applies when a computer is connected to a private network location, such as a home or work place.

☑ **Public**

Applies when a computer is connected to a public network location.



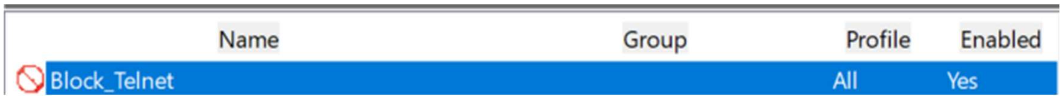
Name

Specify the name and description of this rule.

Name:

Block_Telnet

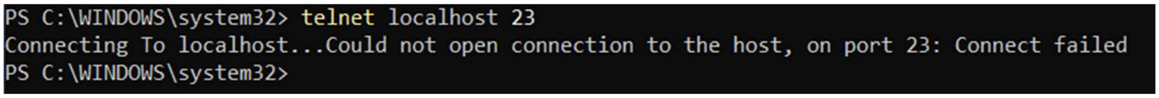
Description (optional):



Name	Group	Profile	Enabled
🚫 Block_Telnet		All	Yes

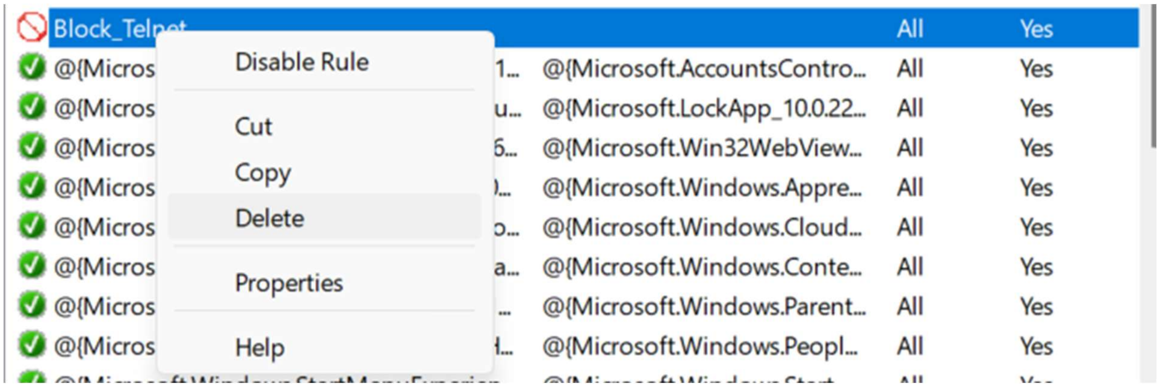
Now Rule has created as you can see in the above image.

4. Tested the rule by attempting to connect on port 23 → Connection failed as expected.



```
PS C:\WINDOWS\system32> telnet localhost 23
Connecting To localhost...Could not open connection to the host, on port 23: Connect failed
PS C:\WINDOWS\system32>
```

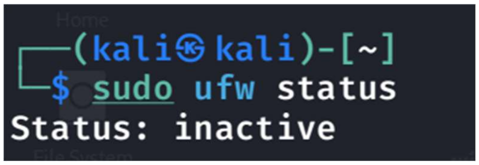
5. Now, We can create other rules (by following proper steps as we have create for Telnet above) as well according to our needs or purposes when we have to use them.
6. Removed the Telnet blocking rule (Block_Telnet) to restore default configuration.



🚫 Block_Tel			All	Yes
✓ @{Micros	1...	@{Microsoft.AccountsContro...	All	Yes
✓ @{Micros	u...	@{Microsoft.LockApp_10.0.22...	All	Yes
✓ @{Micros	6...	@{Microsoft.Win32WebView...	All	Yes
✓ @{Micros	)...	@{Microsoft.Windows.Appre...	All	Yes
✓ @{Micros	o...	@{Microsoft.Windows.Cloud...	All	Yes
✓ @{Micros	a...	@{Microsoft.Windows.Conte...	All	Yes
✓ @{Micros	...	@{Microsoft.Windows.Parent...	All	Yes
✓ @{Micros	1...	@{Microsoft.Windows.Peopl...	All	Yes

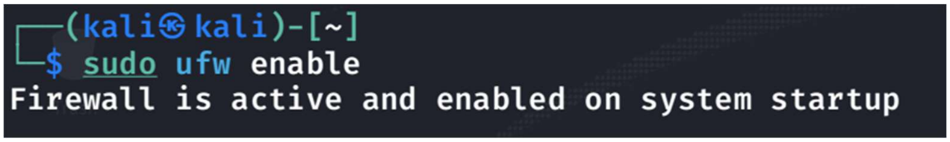
Linux Firewall Configurations Via UFW (Uncomplicated Firewall)

1. Verified if UFW was installed:



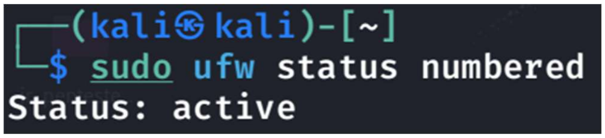
```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status
Status: inactive
```

2. Enabled UFW if not active:



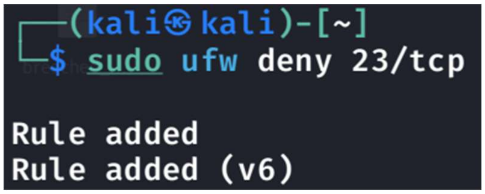
```
┌──(kali㉿kali)-[~]
└─$ sudo ufw enable
Firewall is active and enabled on system startup
```

3. Checked existing rules:



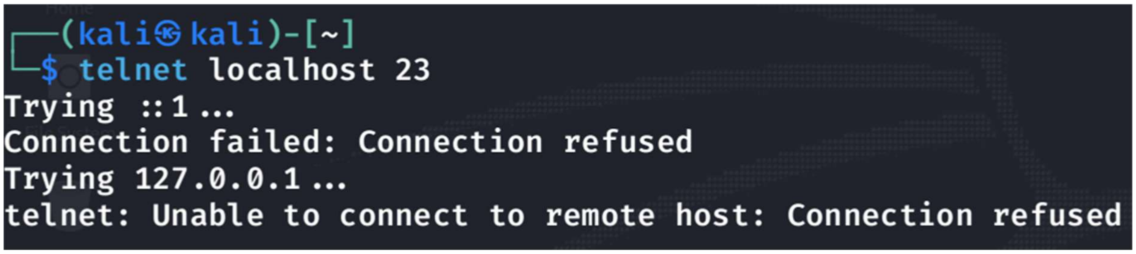
```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status numbered
Status: active
```

4. Blocked inbound Telnet traffic (Port 23):



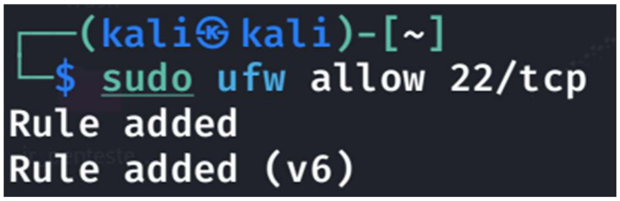
```
┌──(kali㉿kali)-[~]
└─$ sudo ufw deny 23/tcp

Rule added
Rule added (v6)
```

5. Tested the rule using ***telnet localhost 23*** → Connection refused.



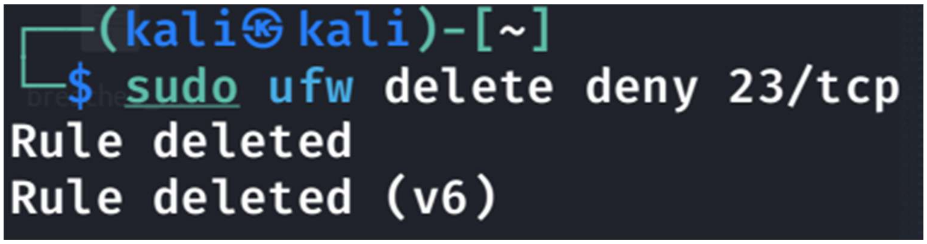
```
┌──(kali㉿kali)-[~]
└─$ telnet localhost 23
Trying ::1...
Connection failed: Connection refused
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection refused
```

6. Allowed SSH (Port 22) to ensure remote management:



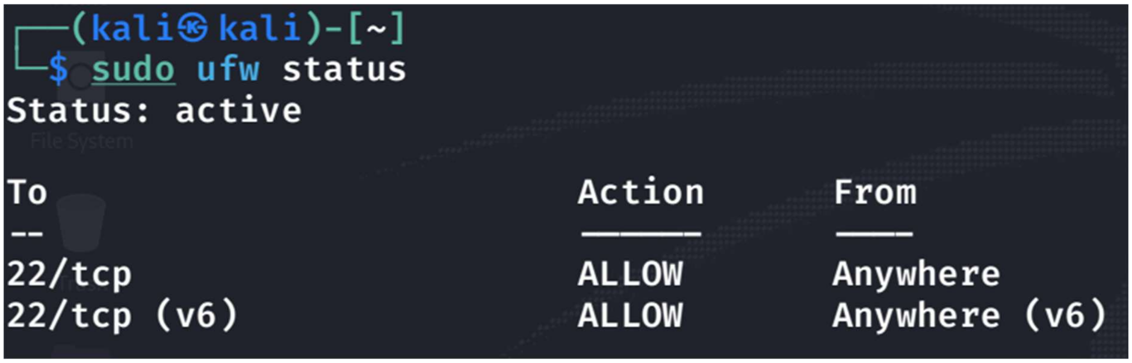
```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow 22/tcp
Rule added
Rule added (v6)
```

7. Deleted the Telnet blocking rule to restore original state:



```
┌──(kali㉿kali)-[~]
└─$ sudo ufw delete deny 23/tcp
Rule deleted
Rule deleted (v6)
```

8. Confirmed applied rules again:



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

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

Commands/Configuration

Windows Firewall

- Open rules: wf.msc
- Create inbound rule (Block Telnet): Port → TCP → 23 → Block
- Create inbound rule (Allow SSH): Port → TCP → 22 → Allow
- Delete rule: Right-click rule → Delete

Linux UFW

- sudo ufw status
- sudo ufw enable
- sudo ufw deny 23/tcp
- sudo ufw allow 22/tcp
- sudo ufw delete deny 23/tcp
- sudo ufw status

Observations & Results

- On **Windows**, blocking Telnet immediately stopped connections on port 23, while allowing SSH (22) kept remote management functional.
- On **Linux**, UFW effectively denied Telnet traffic and allowed SSH. The ufw status command clearly displayed active rules.
- Both firewalls provided **granular control** over inbound/outbound traffic.

Conclusion

This task successfully demonstrated how to configure, apply, and test firewall rules on both **Windows Firewall** and **Linux UFW**. Blocking Telnet and allowing SSH helped understand the importance of filtering network traffic. The task improved practical skills in traffic filtering, port management, and highlighted firewalls as a core defense mechanism in cybersecurity.