Report : Windows Firewall Configuration and Testing

By : Pranav Vijayakumar

Objective

The objective of this task is to configure and test basic firewall rules using **Windows Defender Firewall** to allow or block specific types of network traffic. Specifically, this report demonstrates how to block inbound traffic on **port 23 (Telnet)**, test the rule, and then restore the firewall to its original state by removing the rule.

Table of Content

1. Tools Used

- **Windows Defender Firewall with Advanced Security**
- **PowerShell (optional for command-line configuration)**
- **Telnet Client** (for testing blocked port)

2. Steps Performed

Step 1: Open Firewall Configuration Tool

- Accessed the Windows Firewall GUI via:
- Control Panel → System and Security → Windows Defender Firewall → Advanced Settings
- This opened the **Windows Defender Firewall with Advanced Security** console where rules can be viewed, created, modified, or deleted.

Step 2: List Current Firewall Rules

- Navigated to **Inbound Rules** to view all existing rules applied to incoming network traffic.
- Verified that no existing rule was blocking port 23.
- Alternatively, used PowerShell:
- Get-NetFirewallRule | Format-Table Name, Enabled, Direction, Action, DisplayName

Step 3: Add Rule to Block Port 23 (Telnet)

Using the GUI:

1. Clicked **New Rule** in the right panel under Inbound Rules.
2. Selected **Port** as the rule type.
3. Chose **TCP**, and specified port **23**.
4. Selected **Block the connection**.
5. Applied the rule to **Domain**, **Private**, and **Public** profiles.
6. Named the rule: **Block Telnet Port 23**.

Alternatively, using PowerShell:

New-NetFirewallRule -DisplayName "Block Telnet Port 23" -Direction Inbound -LocalPort 23 -Protocol TCP -Action Block

Step 4: Test the Firewall Rule

- Enabled the **Telnet Client** using:
- dism /online /Enable-Feature /FeatureName:TelnetClient
- Attempted to connect to port 23 using:
- telnet localhost 23
- Result: **Connection failed**, confirming that the firewall rule successfully blocked inbound traffic on port 23.

Step 5: Remove the Block Rule

Using GUI:

- Located the rule **Block Telnet Port 23** under Inbound Rules.
- Right-clicked and selected **Delete**.

Using PowerShell:

```
Remove-NetFirewallRule -DisplayName "Block Telnet Port 23"
```

- Rule was successfully removed, restoring the firewall to its previous state.

3. Firewall Filtering Summary

Windows Firewall filters network traffic based on a set of defined **rules**. These rules can specify:

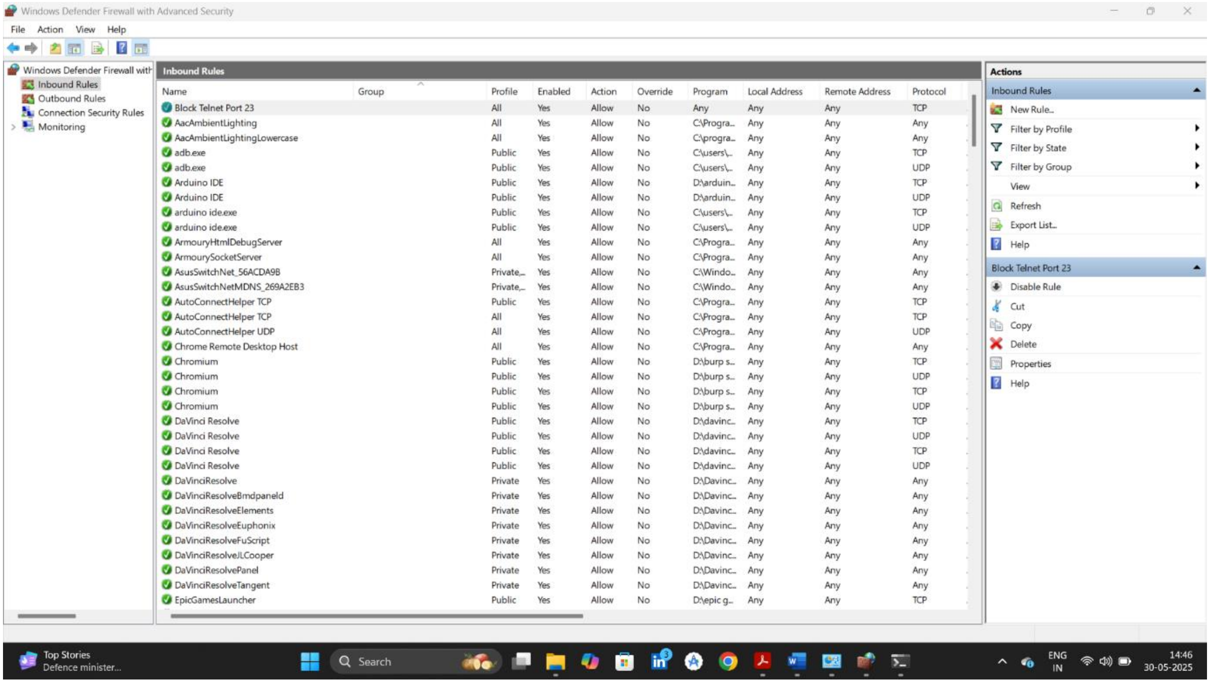
- **Direction**: Inbound or Outbound
- **Protocol**: TCP, UDP, etc.
- **Port Number**: Such as 23 for Telnet, 22 for SSH, etc.
- **Profile**: Domain, Private, Public

By defining these rules, Windows Firewall can:

- Allow or block traffic based on security requirements.
- Prevent unauthorized access from external networks.
- Protect against malware and intrusion attempts that exploit open ports.

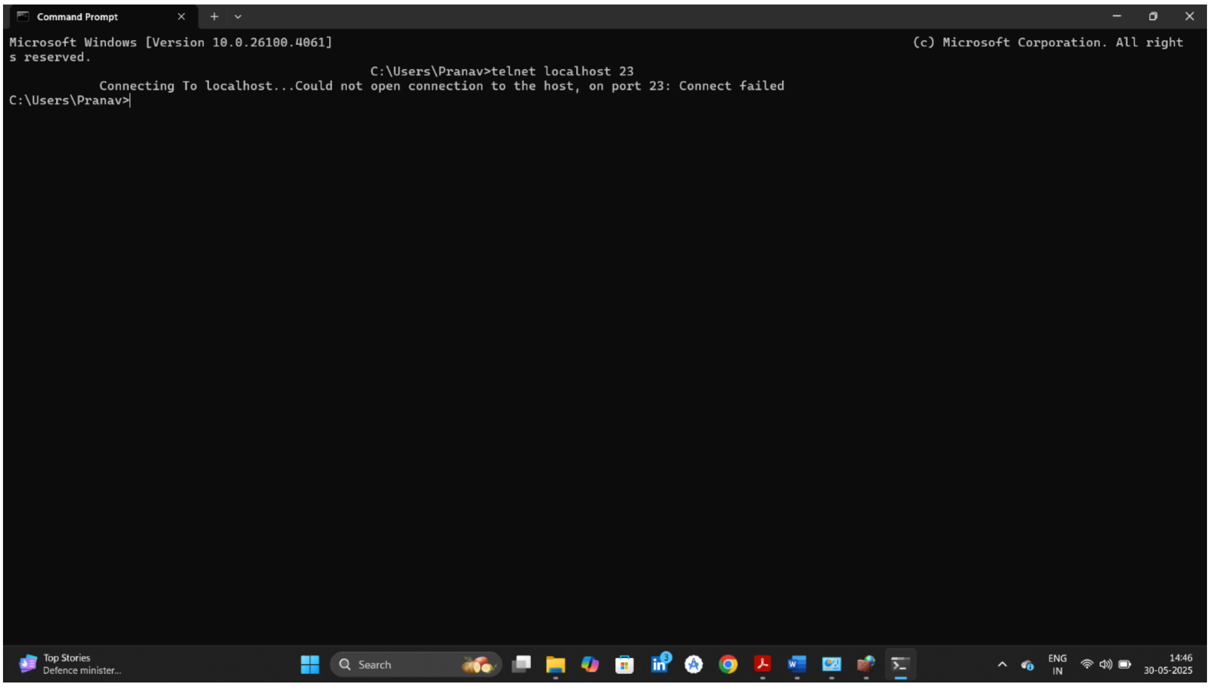
In this task, port 23 (commonly used by the Telnet protocol, which is insecure) was blocked to simulate a basic security policy that helps mitigate risk.

4. Screenshots



Inbound Rules

Name	Group	Profile	Enabled	Action	Override	Program	Local Address	Remote Address	Protocol
Block Telnet Port 23		All	Yes	Allow	No	Any	Any	Any	TCP
AacAmbientLighting		All	Yes	Allow	No	C:\Progra...	Any	Any	Any
AacAmbientLightingLowercase		All	Yes	Allow	No	C:\progra...	Any	Any	Any
adb.exe		Public	Yes	Allow	No	C:\users\...	Any	Any	TCP
adb.exe		Public	Yes	Allow	No	C:\users\...	Any	Any	UDP
Arduino IDE		Public	Yes	Allow	No	D:\arduin...	Any	Any	TCP
Arduino IDE		Public	Yes	Allow	No	D:\arduin...	Any	Any	UDP
arduino ide.exe		Public	Yes	Allow	No	C:\users\...	Any	Any	TCP
arduino ide.exe		Public	Yes	Allow	No	C:\users\...	Any	Any	UDP
ArmouryHtmlDebugServer		All	Yes	Allow	No	C:\Progra...	Any	Any	Any
ArmourySocketServer		All	Yes	Allow	No	C:\Progra...	Any	Any	Any
AsusSwitchNet_56ACDA9B		Private,...	Yes	Allow	No	C:\Windo...	Any	Any	Any
AsusSwitchNetMDNS_269A2EB3		Private,...	Yes	Allow	No	C:\Windo...	Any	Any	Any
AutoConnectHelper TCP		Public	Yes	Allow	No	C:\Progra...	Any	Any	TCP
AutoConnectHelper TCP		All	Yes	Allow	No	C:\Progra...	Any	Any	TCP
AutoConnectHelper UDP		All	Yes	Allow	No	C:\Progra...	Any	Any	UDP
Chrome Remote Desktop Host		All	Yes	Allow	No	C:\Progra...	Any	Any	Any
Chromium		Public	Yes	Allow	No	D:\burp s...	Any	Any	TCP
Chromium		Public	Yes	Allow	No	D:\burp s...	Any	Any	UDP
Chromium		Public	Yes	Allow	No	D:\burp s...	Any	Any	TCP
Chromium		Public	Yes	Allow	No	D:\burp s...	Any	Any	UDP
DaVinci Resolve		Public	Yes	Allow	No	D:\davinc...	Any	Any	TCP
DaVinci Resolve		Public	Yes	Allow	No	D:\davinc...	Any	Any	UDP
DaVinci Resolve		Public	Yes	Allow	No	D:\davinc...	Any	Any	TCP
DaVinci Resolve		Public	Yes	Allow	No	D:\davinc...	Any	Any	UDP
DaVinciResolve		Private	Yes	Allow	No	D:\Davinc...	Any	Any	Any
DaVinciResolveBmdpaneld		Private	Yes	Allow	No	D:\Davinc...	Any	Any	Any
DaVinciResolveElements		Private	Yes	Allow	No	D:\Davinc...	Any	Any	Any
DaVinciResolveEuphonix		Private	Yes	Allow	No	D:\Davinc...	Any	Any	Any
DaVinciResolveFuScript		Private	Yes	Allow	No	D:\Davinc...	Any	Any	Any
DaVinciResolveJLCooper		Private	Yes	Allow	No	D:\Davinc...	Any	Any	Any
DaVinciResolvePanel		Private	Yes	Allow	No	D:\Davinc...	Any	Any	Any
DaVinciResolveTangent		Private	Yes	Allow	No	D:\Davinc...	Any	Any	Any
EpicGamesLauncher		Public	Yes	Allow	No	D:\epic g...	Any	Any	TCP



```
Microsoft Windows [Version 10.0.26100.4061]                                          (c) Microsoft Corporation. All right
s reserved.
                                                C:\Users\Pranav>telnet localhost 23
            Connecting To localhost...Could not open connection to the host, on port 23: Connect failed
C:\Users\Pranav>
```

5. Conclusion

The experiment successfully demonstrated the process of configuring and testing a custom firewall rule in Windows. By blocking Telnet traffic on port 23, the system effectively denied potential remote access using an insecure protocol. This exercise reinforces the importance of using firewall rules to safeguard systems from unauthorized or risky connections.