

Windows Firewall Inbound Rules Test

Internship: Elevate Labs

Submitted by Menati Vyshnavi

Objective

This exercise demonstrates the use of Windows Firewall inbound rules to manage network traffic, specifically by blocking Telnet (port 23) and testing its behavior using Windows PowerShell.

What is Telnet?

Telnet is a network protocol that enables users to connect to and control remote computers over a TCP/IP network, such as the internet or a local area network (LAN). It creates a virtual terminal connection, allowing text-based interactions with a remote device. Telnet operates using plain text transmission, making it less secure than modern alternatives like SSH (Secure Shell). It is typically used for network device management, testing services, and troubleshooting.

What is PowerShell?

PowerShell is Microsoft's advanced command-line shell and scripting language built on the .NET framework. It is designed for system administration, automation of tasks, configuration management, and network diagnostics. PowerShell supports both interactive commands (cmdlets) and scripting, providing administrators with powerful tools to automate complex tasks and perform remoting, system monitoring, and troubleshooting.

What are Inbound Firewall Rules?

Inbound firewall rules control all incoming traffic to a device or network. These rules are critical for network security as they filter and restrict access to internal resources, protect against malware and hacking attempts, and regulate which ports/services are accessible from external sources. Common use-cases include allowing secure connections like HTTPS and

blocking potentially insecure or unneeded protocols such as Telnet.

Methodology

Navigated to the Windows Firewall interface and reviewed existing inbound rules.

Created custom rules:

Allowed incoming SSH traffic (Port 22)

Blocked incoming Telnet traffic (Port 23)

Checked the enabled/disabled status for these rules.

Used PowerShell to test TCP connectivity to Telnet port (23) on localhost.

Observations

Inbound Rules Configuration

The inbound rules panel lists several predefined and custom firewall rules for multiple applications and services.

The custom rule for "Block Telnet 23" is clearly visible and enabled, which should prevent connections over port 23.

PowerShell Test Result

Command executed:

*Test-NetConnection -ComputerName 127.0.0.1 -Port
23*

Outcome:

TCP connection attempt to port 23 failed
(TcpTestSucceeded: False), confirming the firewall rule
is working.

Network path was reachable (PingSucceeded: True) but
traffic was blocked at the specified port.

Refer to the PowerShell output screenshot for evidence.

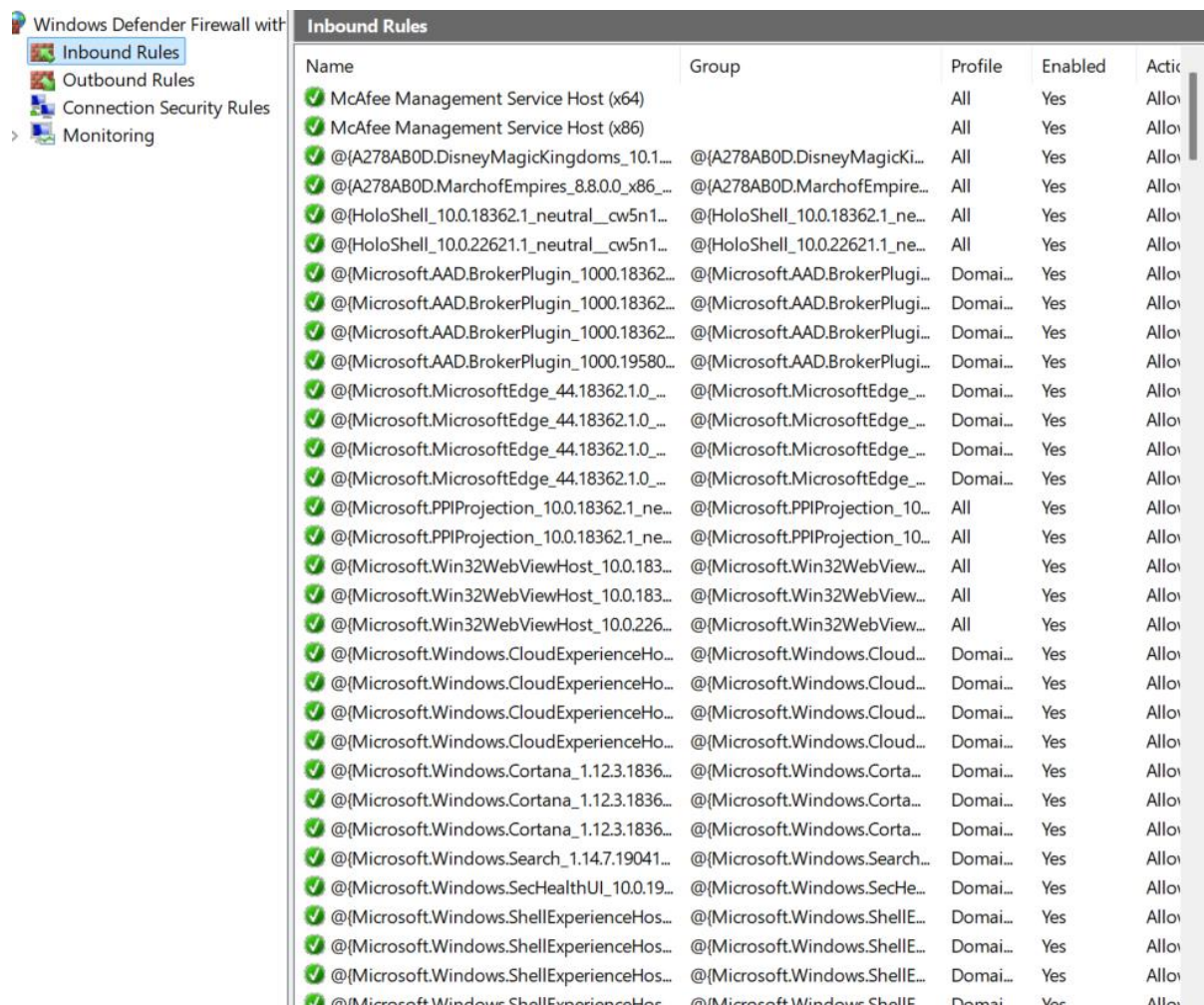
Analysis

The test validates that inbound traffic can be managed
effectively using firewall rules, enhancing system
security.

Blocking Telnet (an insecure protocol) prevents unauthorized access, while allowing other traffic (e.g., SSH) supports secure remote management.

PowerShell streamlines testing and troubleshooting of firewall configurations.

References



The screenshot shows the Windows Defender Firewall with Inbound Rules window. The left sidebar contains links to Inbound Rules, Outbound Rules, Connection Security Rules, and Monitoring. The main pane displays a list of inbound rules with columns for Name, Group, Profile, Enabled, and Action. The rules are sorted by Name and include various Microsoft and McAfee services.

Name	Group	Profile	Enabled	Action
McAfee Management Service Host (x64)		All	Yes	Allow
McAfee Management Service Host (x86)		All	Yes	Allow
@{A278AB0D.DisneyMagicKingdoms_10.1...	@{A278AB0D.DisneyMagicKi...	All	Yes	Allow
@{A278AB0D.MarchofEmpires_8.8.0.0_x86_...	@{A278AB0D.MarchofEmpire...	All	Yes	Allow
@{HoloShell_10.0.18362.1_neutral_cw5n1...	@{HoloShell_10.0.18362.1_ne...	All	Yes	Allow
@{HoloShell_10.0.22621.1_neutral_cw5n1...	@{HoloShell_10.0.22621.1_ne...	All	Yes	Allow
@{Microsoft.AAD.BrokerPlugin_1000.18362...	@{Microsoft.AAD.BrokerPlugi...	Domai...	Yes	Allow
@{Microsoft.AAD.BrokerPlugin_1000.18362...	@{Microsoft.AAD.BrokerPlugi...	Domai...	Yes	Allow
@{Microsoft.AAD.BrokerPlugin_1000.18362...	@{Microsoft.AAD.BrokerPlugi...	Domai...	Yes	Allow
@{Microsoft.AAD.BrokerPlugin_1000.19580...	@{Microsoft.AAD.BrokerPlugi...	Domai...	Yes	Allow
@{Microsoft.MicrosoftEdge_44.18362.1.0_...	@{Microsoft.MicrosoftEdge_...	Domai...	Yes	Allow
@{Microsoft.MicrosoftEdge_44.18362.1.0_...	@{Microsoft.MicrosoftEdge_...	Domai...	Yes	Allow
@{Microsoft.MicrosoftEdge_44.18362.1.0_...	@{Microsoft.MicrosoftEdge_...	Domai...	Yes	Allow
@{Microsoft.PPIProjection_10.0.18362.1_ne...	@{Microsoft.PPIProjection_10...	All	Yes	Allow
@{Microsoft.PPIProjection_10.0.18362.1_ne...	@{Microsoft.PPIProjection_10...	All	Yes	Allow
@{Microsoft.Win32WebViewHost_10.0.183...	@{Microsoft.Win32WebView...	All	Yes	Allow
@{Microsoft.Win32WebViewHost_10.0.183...	@{Microsoft.Win32WebView...	All	Yes	Allow
@{Microsoft.Win32WebViewHost_10.0.226...	@{Microsoft.Win32WebView...	All	Yes	Allow
@{Microsoft.Windows.CloudExperienceHo...	@{Microsoft.Windows.Cloud...	Domai...	Yes	Allow
@{Microsoft.Windows.CloudExperienceHo...	@{Microsoft.Windows.Cloud...	Domai...	Yes	Allow
@{Microsoft.Windows.CloudExperienceHo...	@{Microsoft.Windows.Cloud...	Domai...	Yes	Allow
@{Microsoft.Windows.CloudExperienceHo...	@{Microsoft.Windows.Cloud...	Domai...	Yes	Allow
@{Microsoft.Windows.Cortana_1.12.3.1836...	@{Microsoft.Windows.Corta...	Domai...	Yes	Allow
@{Microsoft.Windows.Cortana_1.12.3.1836...	@{Microsoft.Windows.Corta...	Domai...	Yes	Allow
@{Microsoft.Windows.Cortana_1.12.3.1836...	@{Microsoft.Windows.Corta...	Domai...	Yes	Allow
@{Microsoft.Windows.Search_1.14.7.19041...	@{Microsoft.Windows.Search...	Domai...	Yes	Allow
@{Microsoft.Windows.SecHealthUI_10.0.19...	@{Microsoft.Windows.SecHe...	Domai...	Yes	Allow
@{Microsoft.Windows.ShellExperienceHos...	@{Microsoft.Windows.ShellE...	Domai...	Yes	Allow
@{Microsoft.Windows.ShellExperienceHos...	@{Microsoft.Windows.ShellE...	Domai...	Yes	Allow
@{Microsoft.Windows.ShellExperienceHos...	@{Microsoft.Windows.ShellE...	Domai...	Yes	Allow
@{Microsoft.Windows.ShellExperienceHos...	@{Microsoft.Windows.ShellE...	Domai...	Yes	Allow

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\PRAGNYA MALIKA> Test-NetConnection -ComputerName 127.0.0.1 -Port 23
WARNING: TCP connect to (127.0.0.1 : 23) failed

ComputerName      : 127.0.0.1
RemoteAddress     : 127.0.0.1
RemotePort        : 23
InterfaceAlias    : Loopback Pseudo-Interface 1
SourceAddress     : 127.0.0.1
PingSucceeded     : True
PingReplyDetails (RTT) : 0 ms
TcpTestSucceeded  : False

PS C:\Users\PRAGNYA MALIKA> |
```

Conclusion

This experiment demonstrates how Windows Firewall inbound rules can be configured to restrict network access for certain protocols. Blocking Telnet improves security by preventing unencrypted remote sessions, and PowerShell provides a simple way to verify rule enforcement.