**Windows Firewall Configuration and Traffic Control Lab Report**

**Objective and Environment**

The goal of this lab was to configure and test **Windows Firewall** rules to control network traffic by blocking and allowing specific ports, demonstrating an understanding of firewall operations and packet filtering.

| Setting | Value |
| --- | --- |
| **Operating System** | Windows 10 Pro (Build 19045) |
| **Tools Used** | Windows Defender Firewall with Advanced Security, PowerShell, Command Prompt |
| **Testing Tools** | Test-NetConnection, Telnet Client, netstat |

**Part 1: Initial Firewall Status and Configuration**

The Windows Defender Firewall was accessed via wf.msc. The initial configuration was checked using PowerShell:

PowerShell

Get-NetFirewallProfile | Format-Table Name, Enabled, DefaultInboundAction, DefaultOutboundAction

| Profile | Enabled | Default Inbound Action | Default Outbound Action |
| --- | --- | --- | --- |
| Domain | True | Block | Allow |
| Private | True | Block | Allow |
| Public | True | Block | Allow |

**Observation:** All profiles are active, employing a secure default policy: **blocking inbound** connections while **allowing outbound** traffic. A total of **487** initial firewall rules were found.

**Part 2 & 3: Blocking Telnet Traffic (Port 23)**

A new, explicit **Block** rule was created to prevent all incoming Telnet connections, which use TCP Port 23.

**Rule Creation**

PowerShell

New-NetFirewallRule -DisplayName "Block Telnet Inbound" `

-Direction Inbound `

-LocalPort 23 `

-Protocol TCP `

-Action Block `

-Profile Any `

-Description "Lab exercise - blocking insecure Telnet protocol"

| Field | Value |
| --- | --- |
| **DisplayName** | Block Telnet Inbound |
| **Direction** | Inbound |
| **Action** | **Block** |
| **Port/Protocol** | 23 / TCP |

**Testing the Blocked Port**

Testing confirmed the rule was effective.

| Test Command | Result | Analysis |
| --- | --- | --- |
| Test-NetConnection -ComputerName localhost -Port 23 | TcpTestSucceeded : False | Connection failed as the firewall blocked the attempt. |
| telnet localhost 23 | Connect failed | Telnet client could not establish a connection, validating the block rule. |

**Part 4 & 5: Allowing SSH Traffic (Port 22)**

A custom **Allow** rule was created for SSH (TCP Port 22), a secure alternative to Telnet.

**Rule Creation**

PowerShell

New-NetFirewallRule -DisplayName "Allow SSH Inbound" `

-Direction Inbound `

-LocalPort 22 `

-Protocol TCP `

-Action Allow `

-Profile Any `

-Description "Lab exercise - allow SSH for secure remote access"

| Field | Value |
| --- | --- |
| **DisplayName** | Allow SSH Inbound |
| **Direction** | Inbound |
| **Action** | **Allow** |
| **Port/Protocol** | 22 / TCP |

**Testing the SSH Port**

The command Test-NetConnection -ComputerName localhost -Port 22 was used. The firewall rule successfully enabled the traffic, although connection success ultimately depends on whether an SSH server is actively **listening** on the port.

**Part 6: Removing Test Rules**

All custom rules were successfully removed to return the system to its initial security state, adhering to best practices.

PowerShell

Remove-NetFirewallRule -DisplayName "Block Telnet Inbound"

Remove-NetFirewallRule -DisplayName "Allow SSH Inbound"

**Part 7: Understanding Firewall Traffic Filtering**

Windows Firewall uses **stateful packet filtering** to inspect traffic based on packet headers (IPs, Ports, Protocol) and maintains a table of active connections.

**Rule Processing Order**

Windows Firewall processes rules by priority, with explicit blocks overriding explicit allows:

1. **Block Rules** (Highest Priority)
2. **Allow Rules**
3. **Default Profile Behavior** (Lowest Priority: Block Inbound, Allow Outbound)

**Stateful Inspection**

Because the firewall is stateful, it automatically **allows return traffic** for connections initiated internally (outbound). This is why the default policy can block inbound traffic but still allow users to browse the internet.

| Traffic Direction | Rule Action in Lab | Rationale |
| --- | --- | --- |
| **Inbound (23/Telnet)** | Block | Preventing remote access to an insecure service. |
| **Outbound** | Allow (Default) | Permitting internal users to connect to external services. |

**Observations and Learning Outcomes**

**Key Takeaways**

1. **Block Precedence:** Learned that **Block rules override Allow rules**, which is critical for security and troubleshooting.
2. **Rule Specificity:** Firewall rules must be precisely configured (Port, Protocol, Direction) to be effective.
3. **Testing Methods:** Verified rule effectiveness using multiple methods (Test-NetConnection, Telnet, netstat).
4. **Security Implications:** Confirmed the importance of blocking insecure protocols like Telnet (Port 23) and using encrypted alternatives like SSH (Port 22).

**Conclusion**

The lab successfully demonstrated practical firewall management skills, resulting in a deeper understanding of packet filtering, stateful inspection, and the importance of default policies and rule precedence in maintaining network security.