Firewall Configuration and Testing Report

Student Name: Putta Honna

Tools Used: windows firewall

Operating System: Windows 11

Objective: To configure and test basic firewall rules on a Windows operating system to allow or block network traffic, and to understand the fundamental principles of firewall filtering.

Tools Used:

- Windows 11 Operating System
- Windows Defender Firewall with Advanced Security
- Command Prompt (for Telnet client testing),powershell
- Telnet Client (Windows Feature)

Step-by-Step Configuration and Testing

1. Open Firewall Configuration Tool

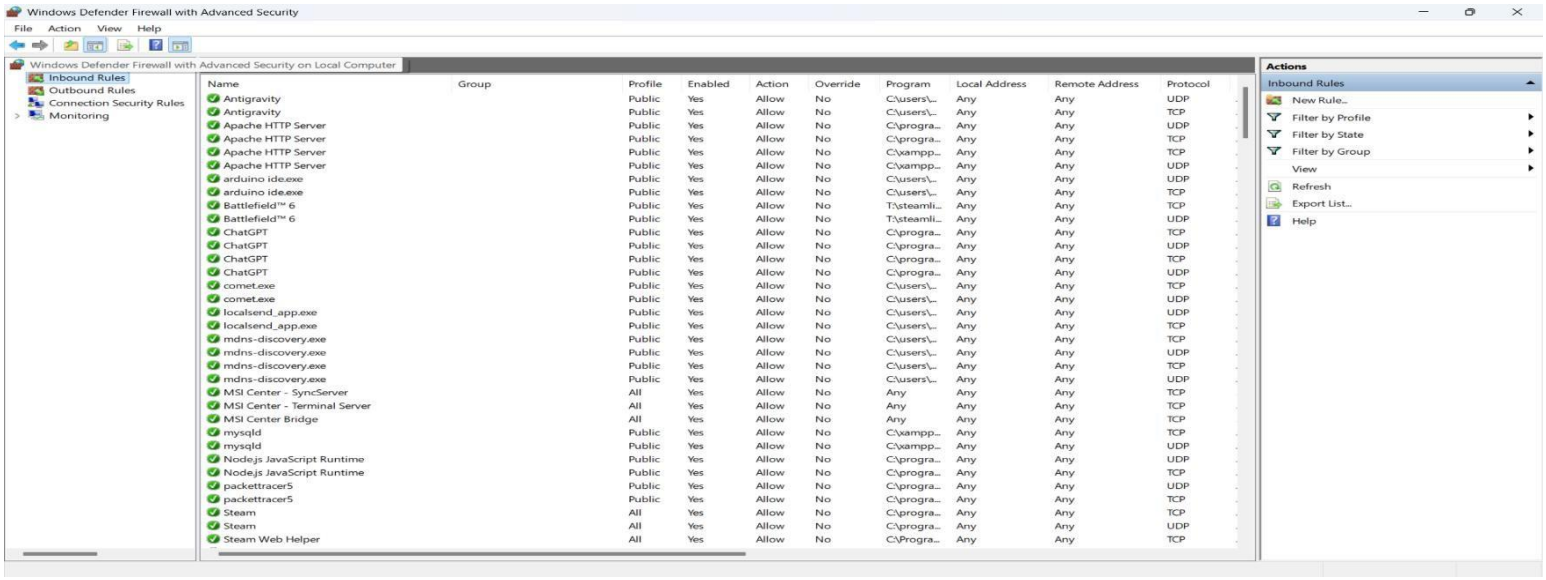
Action: Accessed "Windows Defender Firewall with Advanced Security" via the Start menu search to manage advanced firewall settings.

Screenshot/Configuration: "Windows Defender Firewall with Advanced Security" windows

2. List Current Firewall Rules

Action: Navigated to "Inbound Rules" to view the default and existing rules on the system, establishing a baseline before making changes.

Screenshot/Configuration:

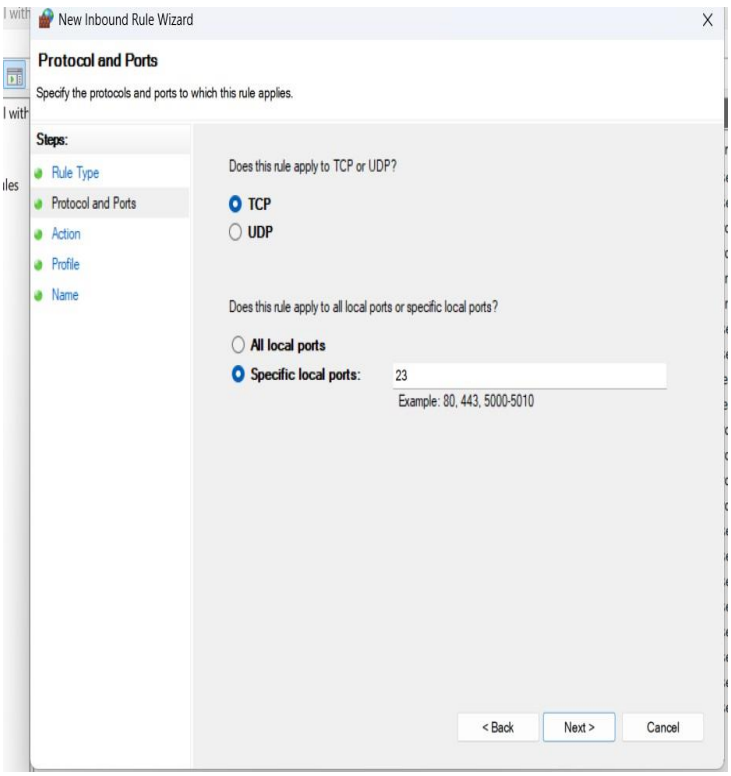


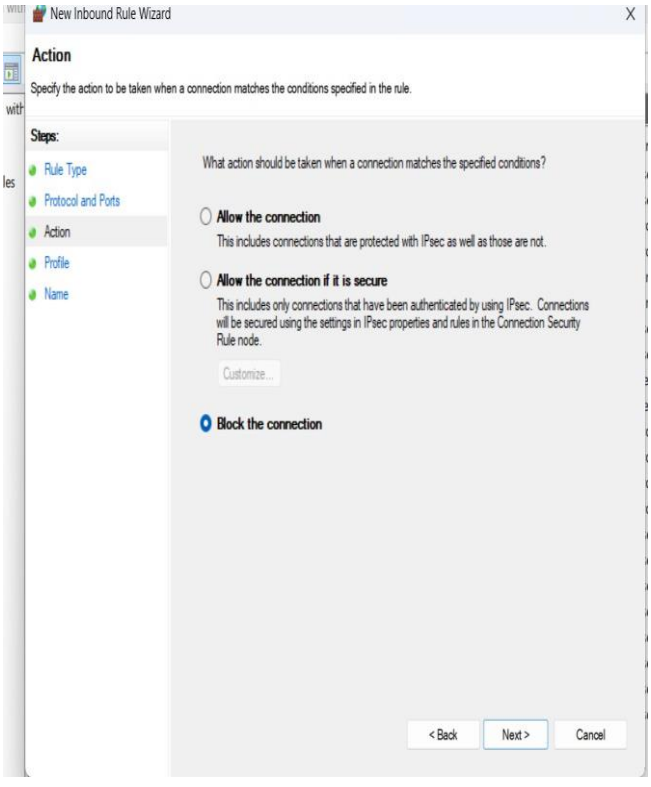
3. Add a Rule to Block Inbound Traffic on a Specific Port (Port 23 - Telnet)

Action: Created a new inbound rule to explicitly block all incoming TCP traffic on Port 23 (Telnet), a commonly targeted insecure protocol.

Steps Taken:

- Right-clicked "Inbound Rules" and selected "New Rule...".
- Selected "Port" as the Rule Type.
- Specified "TCP" protocol and "23" as the specific local port.
- Chosen "Block the connection" as the action.
- Applied the rule to all profiles (Domain, Private, Public).
- Named the rule "Block Inbound Telnet (Port 23)".





4. Test the Rule by Attempting to Connect to That Port Locally

Action: Installed the Telnet Client feature (if not already present) and then used the telnet command in an administrative Command Prompt to attempt a connection to Port 23 on the local machine (127.0.0.1).

Expected Outcome & Result: The connection was successfully blocked by the firewall, confirming the rule's effectiveness. The telnet command reported "Could not open connection to the host, on port 23: Connect failed".

Screenshot/Configuration: *(Please insert your screenshot of the Command Prompt showing the telnet 127.0.0.1 23 command and the "Connect failed" message here.)*

```
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bhuva>telnet localhost 23
Connecting To localhost...Could not open connection to the host, on port 23: Connect failed

C:\Users\bhuva>
```

```
PS C:\Users\bhuva> Test-NetConnection -ComputerName localhost -Port 23
WARNING: TCP connect to (::1 : 23) failed
WARNING: TCP connect to (127.0.0.1 : 23) failed


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

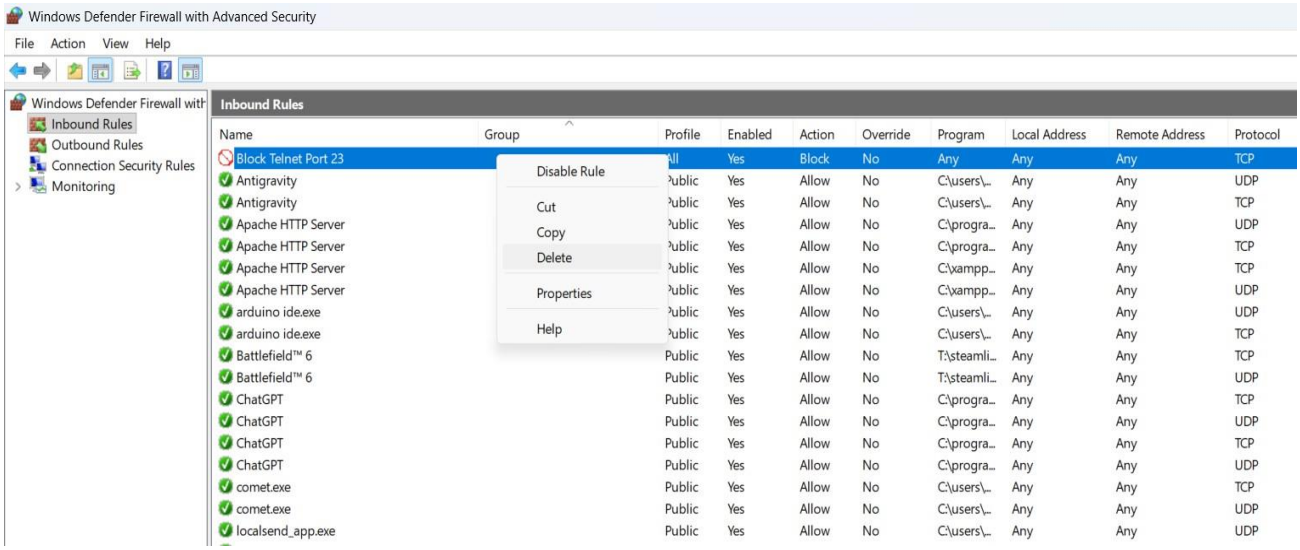
5. Remove the Test Block Rule to Restore Original State

Action: Deleted the "Block Inbound Telnet (Port 23)" rule to ensure the system's firewall configuration was returned to its pre-test state, preventing unintended blocking of services.

Steps Taken:

- Navigated to "Inbound Rules".
- Located the "Block Inbound Telnet (Port 23)" rule.
- Right-clicked and selected "Delete", then confirmed the deletion.

Screenshot/Configuration:



Summary: How Firewalls Filter Traffic

A firewall acts as a security guard for a network or system, inspecting incoming and outgoing network traffic against a set of predefined rules to decide whether to allow or block specific data packets.

Here's a breakdown of how firewalls filter traffic:

1. **Rule Sets:** Firewalls operate based on a comprehensive set of rules, typically configured by an administrator. These rules specify criteria such as:
    - **Source IP Address:** The origin of the traffic.
    - **Destination IP Address:** The intended recipient of the traffic.
    - **Source Port:** The port number used by the sending application.
    - **Destination Port:** The port number for the receiving application (e.g., port 80 for web traffic, 443 for secure web traffic, 22 for SSH, 23 for Telnet).
    - **Protocol:** The network protocol in use (e.g., TCP, UDP, ICMP).
    - **Direction:** Whether the traffic is inbound (entering the system/network) or outbound (leaving the system/network).
2. **Packet Inspection:** As network packets attempt to cross the firewall's boundary, the firewall thoroughly inspects their headers to extract relevant information based on the criteria listed above.
3. **Rule Matching:** The firewall then systematically compares the packet's extracted information against its configured rule set. Rules are usually processed in a specific order (often top-down). The first rule that matches all the packet's characteristics is applied.
4. **Action (Allow/Block/Reject):**
    - **Allow:** The packet is permitted to pass through the firewall.
    - **Block (or Deny):** The packet is silently dropped without any notification sent back to the sender. This action is typically used for malicious or unwanted traffic.
    - **Reject:** The packet is dropped, but an error message (like "Destination Unreachable" or "Connection Refused") is sent back to the source. While useful for troubleshooting, this can sometimes provide information to an attacker.
5. **Default Policy:** Most firewalls have a default policy that is applied if no specific rule matches a packet. This is often a "deny all" for inbound traffic (meaning everything is blocked unless explicitly allowed) and "allow all" for outbound traffic.

By diligently configuring these rules, firewalls play a critical role in protecting systems and networks from unauthorized access, various cyberattacks, and unwanted communication, thus forming a secure boundary between internal and external networks (like the internet).