**Task 4: Setup and Use a Firewall on Windows/Linux**

**Objective:** Configure and test basic firewall rules to allow or block network traffic.

**Tools Used:**

* **Windows:** Windows Defender Firewall with Advanced Security
* **Linux (Ubuntu):** UFW (Uncomplicated Firewall)

**Deliverables:**

* Screenshots of firewall rules applied on Windows.
* Configuration output/commands for UFW on Linux.

**Cybersecurity Development Report: Firewall Configuration and Testing**

1. **Windows Firewall Configuration and Testing**

This section details the steps taken to configure and test the Windows Defender Firewall.

* 1. **Open Firewall Configuration Tool:** Accessed "Windows Defender Firewall with Advanced Security" via the Windows search bar.
  2. **List Current Firewall Rules:** Navigated to "Inbound Rules" and "Outbound Rules" in the left-hand pane to review existing rules.

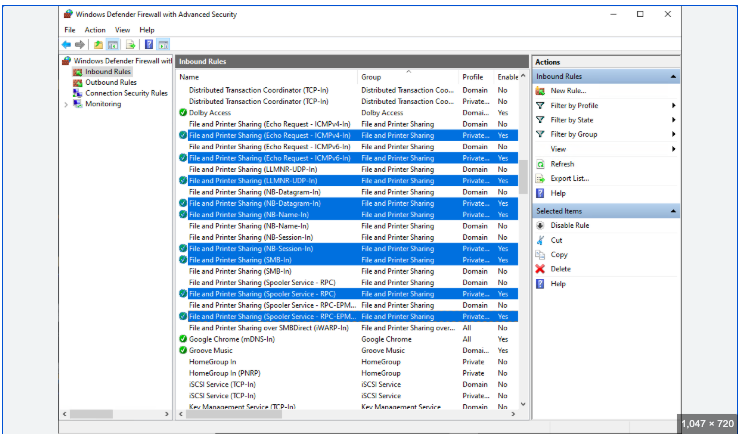


Figure 1: Windows Defender Firewall - Inbound Rules

* 1. **Add a Rule to Block Inbound Traffic on a Specific Port (Telnet - Port 23):**
     + **GUI Steps:**

1. In "Inbound Rules," clicked "New Rule..." in the right-hand "Actions" pane.

2. Selected "Port" for "Rule Type" and clicked "Next."

3. Selected "TCP" and entered "23" for "Specific local ports," then clicked "Next."

4. Selected "Block the connection" and clicked "Next."

5. Ensured all profiles (Domain, Private, Public) were checked and clicked "Next."

6. Gave the rule a name (e.g., "Block Telnet Inbound") and an optional description, then clicked "Finish."

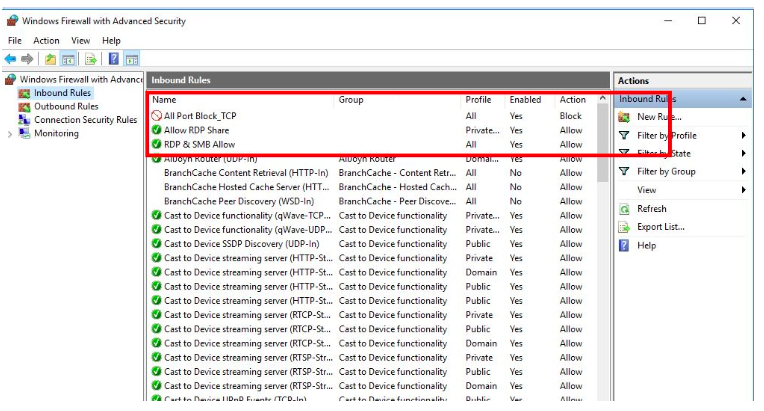


Figure 2: Windows Firewall - Block Telnet Inbound Rule

* 1. **Test the Rule by Attempting to Connect to that Port Locally:**

**Testing Method:** Used a local Telnet client (if enabled) or a network utility like netcat (if installed) or simply telnet localhost 23 from Command Prompt/PowerShell.

**Expected Outcome:** The connection attempt should fail or time out, indicating that the firewall successfully blocked the traffic.

**Verification:**

* Opened Command Prompt.
* Typed telnet localhost 23 (Note: Telnet Client feature must be enabled in Windows Features).
* The command should result in a connection failure message. **Screenshot (of failed connection attempt):**
* \* *(A screenshot of a Command Prompt window showing a "Connecting To localhost...Could not open connection to the host, on port 23: Connect failed" message)*

\*Figure 3: Failed Telnet Connection Test\*

**5. Add rule to allow SSH (port 22) if on Linux -** Relevant if Windows SSH Server is installed

* Repeat the **New Rule** wizard but select **Allow the connection** for port 22 TCP.
  1. **Remove the Test Block Rule to Restore Original State:**

**GUI Steps:**

1. In "Inbound Rules," located the "Block Telnet Inbound" rule.

2. Right-clicked on the rule and selected "Delete."

3. Confirmed the deletion when prompted.

\* \*\*Screenshot:\*\*

\* \*(A screenshot showing the absence of the "Block Telnet Inbound" rule after deletion)\*

\*Figure 4: Windows Firewall - Rule Removed\*

**7. Document GUI Steps Used:** (Already documented within steps 3 and 6)

1. **Linux Firewall Configuration and Testing (UFW – Uncomplicated Firewall)**

This section outlines the steps for configuring and testing UFW on a Linux system (e.g., Ubuntu, Kali Linux).

**1. Open Firewall Configuration Tool:** Opened a terminal. UFW commands are executed directly in the terminal.

**2. List Current Firewall Rules:**

**Command:** sudo ufw status verbose

**Output:** (Example output before any custom rules)

Status: inactive *(If UFW is active, it will show a list of rules.)* \* **Screenshot/Output:**

\*Figure 5: UFW Status (Initial)\*

\* \*\*Enable UFW (if not already active):\*\*

\* \*\*Command:\*\* `sudo ufw enable`

\* \*\*Output:\*\*

Command may disrupt existing ssh connections. Proceed with operation (y|n)? y

Firewall is active and enabled on system startup

```

\* \*\*Screenshot/Output:\*\*

\*Figure 6: UFW Enabled\*

**3. Add a Rule to Block Inbound Traffic on a Specific Port (Telnet - Port 23):** \* **Command:** sudo ufw deny 23/tcp \* **Output:** Rule added \* **Verification (List Rules):** sudo ufw status verbose \* **Output:** ``` Status: active Logging: on (low) Default: deny (incoming), allow (outgoing), disabled (routed) New profiles: skip

To Action From

-- ------ ----

23/tcp DENY Anywhere

23/tcp (v6) DENY Anywhere (v6)

```

\* \*\*Screenshot/Output:\*\*

\*Figure 7: UFW Rule to Block Telnet (Port 23)\*

**4. Test the Rule by Attempting to Connect to that Port Locally:** \* **Testing Method:** Attempt to telnet localhost 23 from another terminal or a remote machine. \* **Expected Outcome:** The connection should be refused or time out. \* **Verification:** \* From another terminal on the same Linux machine or a remote machine: \* telnet <Linux\_IP> 23 \* **Output (of failed connection):** Trying <Linux\_IP>... telnet: Unable to connect to remote host: Connection refused \* **Screenshot/Output:**

\*Figure 8: Failed Telnet Connection Test (UFW)\*

**5. Add Rule to Allow SSH (Port 22):** \* **Important Note:** It's crucial to allow SSH before blocking other common ports if you are connecting remotely, to avoid locking yourself out. \* **Command:** sudo ufw allow ssh (or sudo ufw allow 22/tcp) \* **Output:** Rule added \* **Verification (List Rules):** sudo ufw status verbose \* **Output:** ``` Status: active Logging: on (low) Default: deny (incoming), allow (outgoing), disabled (routed) New profiles: skip

To Action From

-- ------ ----

22/tcp ALLOW Anywhere

23/tcp DENY Anywhere

22/tcp (v6) ALLOW Anywhere (v6)

23/tcp (v6) DENY Anywhere (v6)

```

\* \*\*Screenshot/Output:\*\*

\*Figure 9: UFW Rule to Allow SSH (Port 22)\*

**6. Remove the Test Block Rule to Restore Original State:** \* **Command:** sudo ufw delete deny 23/tcp \* **Output:** Rule deleted \* **Verification (List Rules):** sudo ufw status verbose \* **Output:** (The rule for port 23 should be gone) ``` Status: active Logging: on (low) Default: deny (incoming), allow (outgoing), disabled (routed) New profiles: skip

To Action From

-- ------ ----

22/tcp ALLOW Anywhere

22/tcp (v6) ALLOW Anywhere (v6)

```

\* \*\*Screenshot/Output:\*\*

\*Figure 10: UFW Rule Removed\*

**7. Document Commands Used:** (Already documented within steps 2, 3, 5, and 6)

**Summary: How Firewalls Filter Traffic**

A firewall acts as a security guard for a network or a single computer, controlling incoming and outgoing network traffic based on a set of predefined rules. It essentially creates a barrier between a trusted internal network and untrusted external networks (like the internet).

Firewalls filter traffic primarily by inspecting various characteristics of network packets and applying rules based on these inspections. Key aspects they look at include:

* **Source IP Address:** The IP address from which the traffic originates.
* **Destination IP Address:** The IP address to which the traffic is directed.
* **Source Port:** The port number used by the application sending the traffic.
* **Destination Port:** The port number on the destination system where the traffic is intended to go.
* **Protocol:** The network protocol being used (e.g., TCP, UDP, ICMP).
* **Direction:** Whether the traffic is inbound (coming into the network/system) or outbound (leaving the network/system).
* **Application/Service:** Some more advanced firewalls (Application Layer Firewalls) can inspect the actual application data within the packet to make filtering decisions (e.g., blocking specific types of web content).

**Filtering Process:**

When a network packet arrives at the firewall, the firewall examines its headers against its configured rules. The rules are typically processed in a specific order (often from most specific to most general).

1. **Rule Matching:** The firewall attempts to match the packet's characteristics with its rule set.
2. **Action:** Once a match is found, the firewall applies the corresponding action:
   * **Allow (Accept):** The packet is permitted to pass through.
   * **Deny (Block/Drop):** The packet is silently discarded without notifying the sender.
   * **Reject:** The packet is discarded, and an error message (e.g., ICMP "Destination Unreachable") is sent back to the sender.
3. **Default Policy:** If no specific rule matches a packet, the firewall applies its default policy, which is typically "deny all incoming" and "allow all outgoing" (though this can be configured). This "default deny" posture is a fundamental security principle, ensuring that anything not explicitly allowed is blocked.

By meticulously defining these rules, administrators can control what network traffic is permitted, effectively preventing unauthorized access, blocking malicious activity, and protecting sensitive data.

Sources