Firewall Configuration Report and Setup Procedure

Objective: To configure and test basic firewall rules on Windows and Linux to allow or block network traffic.

Windows Firewall Configuration

1. Setup Procedure:

* **Accessing the Firewall:** I accessed the "Windows Firewall with Advanced Security" by running `wf.msc`.
* **Creating an Inbound Rule (Block RDP):**
  * I created a new inbound rule.
  * Rule Type: Port
  * Protocol and Ports: TCP, port 3389
  * Action: Block the connection
  * Profile: All (Domain, Private, Public)
  * Name: "Block RDP Access"
* **Creating an Outbound Rule (Allow Application):**
  * I created a new outbound rule.
  * Rule Type: Program
  * Program Path: [Path to the application's .exe file]
  * Action: Allow the connection
  * Profile: All (Domain, Private, Public)
  * Name: "Allow [Application Name] Internet"

2. Testing and Verification:

* To test the "Block RDP" rule, I attempted to connect from another machine on the network using Remote Desktop, and the connection failed as expected.

*   To test the "Allow Application" rule, I ran the application and confirmed it could access the internet.

3. Configuration Output:

[Insert screenshot of the Windows Firewall rules or a note that the .wfw policy file is attached.]

Linux (UFW) Firewall Configuration

1. Setup Procedure:

*   **Installation and Initial Setup:** I installed UFW using `sudo apt install ufw` and checked its status.
*   **Default Policies:** I set the default policies to deny incoming traffic and allow outgoing traffic using:
    *   `sudo ufw default deny incoming`
    *   `sudo ufw default allow outgoing`
*   **Creating Rules:**
    *   I allowed SSH traffic: `sudo ufw allow ssh`
    *   I allowed HTTP and HTTPS traffic: `sudo ufw allow http` and `sudo ufw allow https`
    *   I blocked traffic on port 8080: `sudo ufw deny 8080`
*   **Enabling the Firewall:** I enabled UFW with `sudo ufw enable`.

2. Testing and Verification:

*   I confirmed I could still connect via SSH after enabling the firewall.
*   I attempted to access a service on port 8080 from another machine, and the connection was refused, as expected.
*   I could access web services on ports 80 and 443.

3. Configuration Output:

Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From

22/tcp ALLOW IN Anywhere
80/tcp ALLOW IN Anywhere
443/tcp ALLOW IN Anywhere
8080 DENY IN Anywhere
22/tcp (v6) ALLOW IN Anywhere (v6)
80/tcp (v6) ALLOW IN Anywhere (v6)
443/tcp (v6) ALLOW IN Anywhere (v6)
8080 (v6) DENY IN Anywhere (v6)

Outcome and Skills Gained:

Through this task, I have learned the fundamental principles of network traffic filtering. I am now able to set up, configure, and manage basic firewall rules on both Windows and Linux systems to enhance security by controlling inbound and outbound network connections. I have also gained practical experience in testing firewall rules to ensure they are functioning as intended.