Lab 1.1 – Setting Up a Firewall

1. Lab Overview

1.1 Lab Description

Welcome to your first hands-on cybersecurity lab!

In this exercise, you will learn how to enable and configure a basic firewall to protect your system from unauthorized access. Whether you're working on Windows or Linux, this lab will guide you through turning on the firewall, adding rules to allow or block specific traffic, and testing the effectiveness of those rules.

By the end of the lab, you'll have taken the first step toward controlling how data enters and exits your system—an essential foundation in cybersecurity defense.

1.2 Learning Objectives

After completing this lab, you will be able to:

- Understand the purpose and function of a firewall in cybersecurity.
- Enable and manage firewall settings on both Windows and Linux systems.
- Create rules to allow or block network traffic based on ports, IP addresses, or services.
- Test firewall effectiveness using SSH attempts, ping, and Nmap scans.
- Recognize the real-world importance of firewalls in personal and enterprise environments.

1.3 Prerequisites

To successfully complete this lab, you should have:

- Basic knowledge of computer operating systems (Windows and/or Linux).
- Access to a Windows PC or a Linux virtual machine (Ubuntu preferred).
- Administrative privileges on the system you're using.
- An internet connection for testing network traffic (or a second device/VM for SSH testing).
- Optional: Installation of Nmap for port scanning tests.

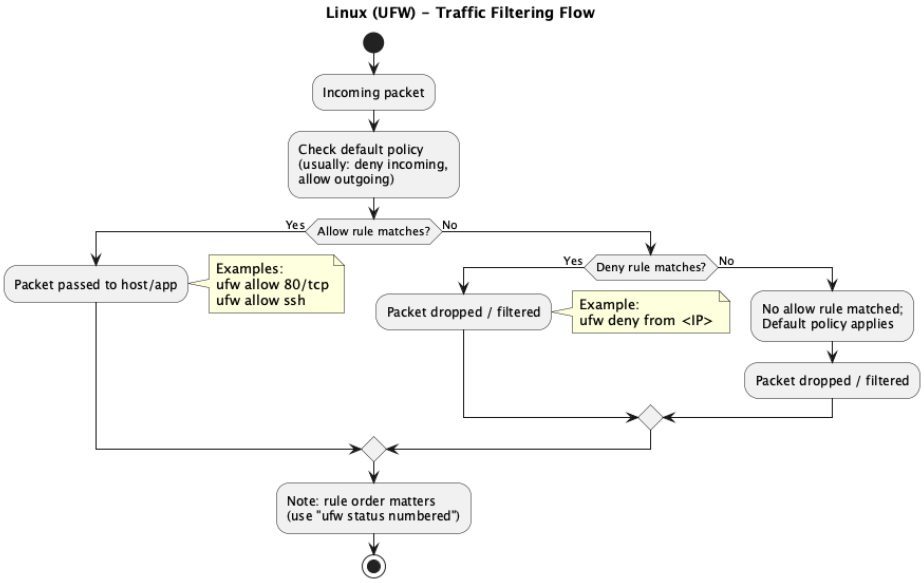
1.4 Estimated Completion Time

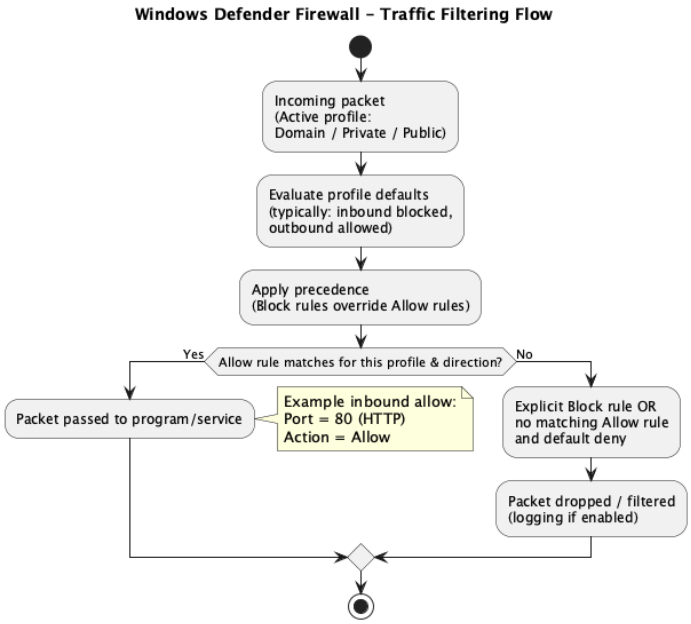
Activity	Estimated Time
Reading Instructions & Environment Setup	10 minutes
Enabling Firewall (Windows and/or Linux)	15 minutes
Configuring Custom Rules (Inbound/Outbound, Ports, IPs)	30 minutes
Testing the Configuration (SSH, Ping, Nmap, Logs)	35 minutes
Deep Dive: Advanced Windows Firewall Features (Optional)	25 minutes
Real-World Reflection: Use Cases and Threat Scenarios	25 minutes
Interactive Challenge: Allow HTTP, Block All Else	30 minutes
Documenting & Writing Your Firewall Configuration Report	10 minutes
Total Estimated Time	**3 hours**

2. Getting Started

2.1 What Is a Firewall?

A **firewall** is a system that monitors and controls incoming and outgoing network traffic based on pre-defined rules. It acts as a barrier between your trusted internal network and potentially dangerous external traffic (like the internet).





There are two main types:

- **Host-based firewalls** (the focus of this lab): Installed on individual machines (e.g., your laptop or server).
- **Network firewalls**: Deployed at the network level to control traffic across multiple devices.

Why this matters:
Firewalls are one of the first lines of defense in any secure system—used by enterprises, governments, and even home users to protect against attacks.

Did You Know?
Most operating systems include a built-in firewall that is disabled by default or minimally configured. This lab teaches you to take control of it.

2.2 Lab Setup Checklist

Before you begin, make sure you have the following ready:

A working computer with either:

- **Windows 10/11** with admin access
- **Ubuntu Linux (22.04 or later)** physical or virtual machine

Terminal access (Linux) or administrative GUI access (Windows)
Internet connection for testing connectivity and installing tools
Ability to install software (if needed: Nmap, OpenSSH)
(Optional) A second device or VM to simulate external SSH access

2.3 Notes for Windows vs Linux Users

This lab supports **both Windows and Linux** users. You can follow both paths or choose the one for your system.
We recommend completing both for deeper understanding.

Task	**Linux (UFW)**	**Windows**
Enable Firewall	sudo ufw enable	Use **Windows Security → Firewall & Network**
Create Rules	ufw allow / ufw deny commands	Use **Advanced Firewall Settings** GUI
Test Firewall	SSH, ping, Nmap	Ping, port scan, test blocked apps
Rule Visibility	ufw status verbose	GUI rule list in Windows Defender Firewall

Tip: If you're not sure which OS to use, start with **Linux UFW**, then compare it with Windows for broader experience.

3. Step 1 – Enabling the Firewall

In this section, you will enable the default host-based firewall on your operating system.

You'll learn:

- How to activate UFW (Uncomplicated Firewall) on Linux.
- How to check and enable the firewall using the Windows Security Center.
- How to verify that your firewall is now protecting your system.

3.1 Enabling the Firewall on Windows

Step-by-Step (Windows 10/11):

1. **Open Windows Security Center**
    - Click the **Start Menu** or press the **Windows key**
    - Type: Windows Security
    - Press **Enter** or click the top result
2. **Navigate to Firewall Settings**
    - In the left-hand menu, click **"Firewall & Network Protection"**
3. **Check the Status for Active Network**
    - Under the **active network** (usually Private or Domain), look for a green check mark and the label **"Firewall is On"**
    - If it's **Off**, click and toggle the firewall **On**
4. **Confirm Success**
    - When the firewall is active, Windows shows a green check mark
    - You're now protected by basic host-based filtering

Tip: Some third-party antivirus tools may disable Windows Firewall. If you don't see the option, check your antivirus settings.

3.2 Enabling the Firewall on Linux (UFW)

Step-by-Step (Ubuntu/Debian-based systems):

These instructions assume you are using Ubuntu 22.04 LTS or similar.

1. **Open the Terminal**
    - You can use **Ctrl + Alt + T** to open the terminal
2. **Check UFW Status (optional)**
3. sudo ufw status
    - If UFW is **inactive**, proceed to enable it
    - If UFW is **active**, skip to the next section
4. **Enable the Firewall**
5. sudo ufw enable
    - Type your **user password** when prompted
    - UFW will respond with:
      Firewall is active and enabled on system startup
6. **Confirm Status**
7. sudo ufw status verbose
    - This shows a detailed summary of current rules and status

Tip: UFW stands for "Uncomplicated Firewall" — it's designed to be beginner-friendly and easy to use on the command line.

3.3 Quick Recap: Windows vs Linux Setup

OS	Steps to Enable Firewall
Windows	Start → Search Windows Security → Firewall & Network Protection → Turn On
Linux	Open Terminal → sudo ufw enable → Check with sudo ufw status verbose

Checkpoint

Before continuing:

- Is your firewall **enabled**?
- Can you **see status output** confirming the firewall is running?
- Are you ready to start configuring rules?

4. Step 2 – Configuring Firewall Rules

Once the firewall is enabled, it doesn't do much until you start adding rules to allow or block specific types of network traffic.

In this section, you'll:

- Allow traffic to essential services (like SSH or HTTP).
- Block unwanted connections (e.g., from specific IPs).
- Learn to use both command-line tools (Linux) and graphical tools (Windows).

4.1 Linux – Managing Rules with UFW

Example 1: Allow SSH Access

Allow incoming connections on port 22 (default for SSH):

```
sudo ufw allow ssh
```

This enables remote login via SSH.

Example 2: Allow Web Traffic (HTTP)

Allow web server traffic on port 80:

```
sudo ufw allow 80/tcp
```

This is commonly used for hosting or accessing websites.

Example 3: Block a Specific IP Address

Block incoming connections from a malicious or suspicious IP:

```
sudo ufw deny from 192.168.1.100
```

Replace with any IP you want to block.

Example 4: Review All Current Rules

View active rules and firewall configuration:

```
sudo ufw status numbered
```

or for more detail:

```
sudo ufw status verbose
```

Note: Rules are applied in the order they are listed. You can later delete or adjust specific rules using their numbers.

4.2 Windows – Creating Rules with Advanced Firewall

Step 1: Open Advanced Firewall Settings

1. Open **Start Menu**
2. Search: Windows Defender Firewall with Advanced Security
3. Click to launch the advanced settings panel

Step 2: Create a New Inbound Rule

1. In the left pane, click **Inbound Rules**
2. On the right, select **New Rule...**
3. Choose **Port**, then click **Next**
4. Enter a port (e.g., 80), select **Allow** or **Block**
5. Apply to the correct profiles (Domain, Private, Public)
6. Name your rule (e.g., “Allow HTTP”), then finish

Step 3: Other Rule Types

You can also:

- Allow or block specific programs
- Filter based on IP addresses
- Choose between TCP or UDP
- Set conditions based on authentication or services

Tip: You can duplicate and edit existing rules to create custom variations quickly.

4.3 Quick Recap: Firewall Rules Summary

Platform	Action	Command / Steps
Linux (UFW)	Allow SSH	`sudo ufw allow ssh`
Linux (UFW)	Allow HTTP	`sudo ufw allow 80/tcp`
Linux (UFW)	Block IP Address	`sudo ufw deny from 192.168.1.100`
Linux (UFW)	Check Rules	`sudo ufw status verbose`
Windows	Open Advanced Settings	Start → Search Windows Defender Firewall with Advanced Security
Windows	Allow Port (e.g., HTTP)	Inbound Rules → New Rule → Port → 80 → Allow
Windows	Block Specific App or IP	New Rule → Program / Custom → Block

Checkpoint

Before moving on:

- Have you created at least two rules (one **allow**, one **block**)?
- Did you test creating rules on **both platforms**, if applicable?
- Are you familiar with how to **view and manage** existing rules?

5. Step 3 – Testing Your Firewall

Creating firewall rules is only half the job — you must also **test them** to ensure they behave as intended.

In this section, you'll:

- Attempt to connect using SSH and observe firewall behavior.
- Use ping to test basic network access.
- Run an **Nmap** scan to view which ports are visible from the outside.

Note: You may need a second device or virtual machine to simulate external access. Testing works best when one system is protected by the firewall and another acts as the "attacker" or external client.

5.1 SSH Blocking Test

Objective:

Test if SSH access is blocked or allowed by your firewall rules.

Steps:

1. On the **second machine**, open a terminal or SSH client.
2. Attempt to connect using:

```
ssh your-username@your-server-ip
```

3. If SSH is **blocked**, you should see:
    - Connection refused
    - Timed out
    - No route to host
4. If SSH is **allowed**, you'll be prompted for your password and can log in.

Tip: On Linux, block SSH using `sudo ufw deny ssh`, then retry the connection.

5.2 Ping Test (ICMP Reachability)

Objective:

Check if your system responds to basic **ICMP echo requests** (ping).

Steps:

1. On another machine, open a terminal.
2. Run:

```
ping your-server-ip
```

3. Observe the result:
    - Replies = ICMP is **allowed**
    - Request timed out = ICMP is **blocked**

Note: Some systems block ICMP by default; this is configurable using advanced firewall settings.

5.3 Nmap Port Scan

Objective:

Use **Nmap** to scan open ports and see which services are exposed.

Steps:

1. On the second system (or from a scanning machine), install Nmap if needed:

```
sudo apt install nmap      # On Debian/Ubuntu
brew install nmap          # On macOS
```

2. Scan the protected system:

```
nmap your-server-ip
```

3. Review the results:
    - Allowed ports (e.g., 22, 80) will appear as **open**
    - Blocked ports won't show or will appear as **filtered**
4. For more details:

```
nmap -v your-server-ip
```

Security Tip: Attackers use tools like Nmap too. Learning to test your own systems helps defend against scans and probes.

5.4 Recap Table – Firewall Testing Methods

Test	Command or Action	Expected Result if Rule Works
SSH Block Test	`ssh user@ip`	Connection refused / timeout
Ping Test	`ping your-server-ip`	Replies (allowed) or timeout (blocked)
Nmap Port Scan	`nmap your-server-ip`	Only allowed ports (e.g., 22, 80) visible
Nmap Verbose Scan	`nmap -v your-server-ip`	Detailed listing of filtered/blocked services

Checkpoint

Confirm before moving on:

- Did you test **at least one blocked and one allowed port**?
- Were you able to confirm firewall behavior with **ping or SSH**?
- Did you try a **port scan** using Nmap?

6. Real-World Relevance

Setting up a firewall may seem simple, but it's one of the most **critical steps in defending real-world systems** — from laptops to corporate data centers.

In this section, you'll learn:

- Why firewalls matter in the wild.
- How attackers are stopped by well-configured rules.
- Where firewalls fit in at home and at work.

6.1 Why Firewalls Matter in Cybersecurity

Firewalls aren't optional. They're **essential defensive tools** used in nearly every secure system today.

Firewalls protect systems by:

- Blocking unauthorized scans and connection attempts.
- Preventing access to vulnerable services.
- Filtering malicious traffic **before** it reaches the system.

6.2 Firewalls in the Wild – Use Cases

Home User:

- Your **home router** has a built-in firewall that silently blocks most inbound internet traffic.
- This protects your devices (laptop, TV, phone) from direct attacks.

Enterprise Systems:

- Large organizations deploy **multi-layered firewalls** to segment internal networks and control access to databases, applications, and employee devices.
- These firewalls are deeply monitored and updated constantly.

Personal Devices:

- Firewalls on laptops or workstations block dangerous apps, prevent backdoors, and isolate malware.

6.3 Visual Summary – Firewall as a Barrier

Firewall Function in 3 Steps:

1. **Inspect** all incoming and outgoing traffic.
2. **Match** traffic to configured rules.
3. **Allow or block** based on policy.

Checkpoint

Reflect on your system:

- What services do you want **open** to the internet?
- What services should always be **blocked**?
- How would a misconfigured firewall expose your system?

7. Lab Challenge

7.1 Scenario: Allow HTTP, Block Everything Else

Your Task:
Configure your system so that **only HTTP (port 80)** traffic is allowed — all other inbound traffic must be blocked.

This simulates a basic web server scenario where you want users to reach your website but block all other access (SSH, ping, etc.).

Instructions for Linux (UFW)

1. **Set the default policy to block all incoming traffic:**

```
sudo ufw default deny incoming
```

2. **Allow only HTTP (port 80):**

```
sudo ufw allow 80/tcp
```

3. **(Optional) Deny all other services explicitly:**

```
sudo ufw deny ssh
```

4. **Check your rules:**

```
sudo ufw status verbose
```

Instructions for Windows

1. **Open Advanced Firewall Settings**
2. **Create a new inbound rule:**
    - Type: **Port**
    - Port: 80
    - Action: **Allow**
    - Name: Allow HTTP
3. **Block all other inbound traffic (optional):**

- Go to **Inbound Rules** → Click **New Rule**
- Type: **All Programs**
- Action: **Block**
- Scope: Apply to all ports or specific ones like 22 for SSH

4. **Reorder or adjust conflicting rules if needed**

7.2 Checklist – Did You Succeed?

Task	Completed?
Set default deny for incoming traffic	[ ]
Allowed only HTTP (port 80)	[ ]
Verified rule status/output	[ ]
Confirmed SSH or ping is blocked	[ ]
Confirmed port 80 is open (via Nmap)	[ ]

7.3 Optional Tests

Try accessing your system from another device or a VM:

- Visit http://your-server-ip → Should load a web service (if one is running)
- Try SSH or Nmap → Should be blocked

Don't have a web server running? Try:

sudo apt install apache2

sudo systemctl start apache2

7.4 Reflection Questions

Answer these in your lab notebook or submit as part of your report:

1. What commands or steps did you use to configure your firewall?

2. How did you verify that non-HTTP traffic was blocked?
3. Were there any services still accessible that you didn't intend to allow?
4. What could happen if you accidentally allowed a vulnerable port?

Challenge Complete!

You've simulated the firewall profile of a **public-facing web server** — one of the most common use cases in cybersecurity operations.

8. Wrap-Up and What's Next

8.1 Key Takeaways

Congratulations — you've completed your first cybersecurity lab!

Here's what you accomplished in this session:

Enabled a host-based firewall on both Linux (UFW) and Windows.
Created inbound and outbound rules to allow or block traffic.
Tested the rules using tools like SSH, ping, and Nmap.
Explored real-world use cases of firewalls in both home and enterprise networks.
Configured a restricted firewall profile to simulate a web server in the challenge.

Why this matters:
Every secure system, from your personal laptop to a global cloud infrastructure, depends on well-configured firewalls. You've taken a professional step toward understanding and controlling digital defenses.

8.2 Skills You've Gained

By completing this lab, you now have hands-on experience with:

- Operating system–level security configuration.
- Command-line networking tools (ufw, ping, ssh, nmap).
- Defensive thinking: minimizing attack surface by restricting access.
- Firewall GUI management and rule logic (Windows Advanced Firewall).

These are skills you'll use throughout your cybersecurity journey — in labs, competitions, or the workplace.

8.3 Coming Up: Malware Scanning Lab

Next, we'll shift gears and look at **how to detect and remove malicious software** from a system using antivirus tools.

In **Lab 1.2 – Malware Scanning**, you'll:

- Perform a scan using built-in or open-source tools.
- Examine how malware behaves and hides.
- Learn what real-world detection and remediation looks like.

Prep Tip: Be ready to install a malware scanner (Windows Defender, ClamAV, etc.) and run both quick and full scans on test files.

Next Steps

- Complete your **reflection questions** from the challenge.
- Submit your **firewall configuration log/report** if required.
- Review the **commands used** — try them again from memory.
- Start preparing your system for **malware scanning**.

You've built a shield — now you'll learn to detect intruders.

9. Appendix

9.1 Linux Command Reference (UFW)

Command	Description
`sudo ufw status`	View basic firewall status
`sudo ufw status verbose`	View detailed rule set
`sudo ufw enable`	Activate the firewall
`sudo ufw disable`	Turn off the firewall
`sudo ufw default deny incoming`	Block all incoming traffic by default
`sudo ufw default allow outgoing`	Allow all outgoing traffic
`sudo ufw allow ssh`	Allow SSH (port 22)
`sudo ufw allow 80/tcp`	Allow HTTP (port 80)
`sudo ufw deny from [IP]`	Block all traffic from specific IP
`sudo ufw delete [rule-number]`	Remove a specific rule by number

9.2 Windows Firewall Tips

- Use the **Advanced Settings** panel for full control over rules.
- Always **test new rules** before deploying in critical environments.
- **Use "Custom" rules** to target ports, apps, or protocols precisely.
- **Default profiles** (Domain, Private, Public) can behave differently — always verify which one is active.
- Use **Event Viewer > Security logs** to track blocked/allowed connections (advanced users).

9.3 Troubleshooting Common Issues

Problem	Possible Cause	Solution
Can't connect via SSH	Rule is blocking port 22	sudo ufw allow ssh or verify Windows rule
Ping returns "Request timed out"	ICMP blocked	Allow ICMP (advanced), or confirm network reachability
Nmap shows all ports closed/filtered	All ports blocked or firewall enabled	Allow specific ports as needed
Windows rule doesn't apply	Wrong profile selected (Private/Public)	Check active network profile and apply to correct one
Firewall not working on Linux	UFW not enabled or missing	Run sudo ufw enable and ensure it's installed

9.4 Optional Tools & Resources

🛠 Recommended Tools

- **Nmap** – https://nmap.org
- **Wireshark** – https://www.wireshark.org
- **ClamAV (Linux antivirus)** – https://www.clamav.net

Further Reading

- UFW Manual: man ufw or https://help.ubuntu.com/community/UFW
- Microsoft Defender Firewall: https://learn.microsoft.com/en-us/windows/security/threat-protection/windows-firewall
- OWASP Firewalls Guide: https://owasp.org/www-community/Firewalls