Firewall Implementation and Simulation

CS-373L Computer Networks

Session 2022-2026

Submitted To

Mr. Syed Tehseen Ul Hassan Shah

Submitted By

Rafiya Rehan	2022-CS-182
Ayesha Shafqat	2022-CS-213

Department of Computer Science

University of Engineering and Technology

Acknowledgment

We would like to express our heartfelt gratitude to all those who contributed to the successful completion of this project. Firstly, we extend our sincere thanks to our instructor, Mr. Syed Tehseen Ul Hassan Shah for his invaluable guidance and support throughout this journey. His knowledge and expertise in the field of Computer Networks have been instrumental in shaping our understanding and approach. Additionally, we are deeply grateful to the authors of the book **Computer Networking: A Top Down Approach**, which served as an excellent reference and provided us with the essential knowledge and skills needed to complete this project. Their work has played a significant role in helping us achieve our goals, and for that, we are truly thankful.

Contents

1 Introduction

This project aims to implement a basic firewall system that enhances network security by managing and controlling network traffic in real-time. The primary purpose of this project is to allow users to block or unblock specific IP addresses and ports, thereby preventing unauthorized access and ensuring safe communication within a network.

The significance of this project lies in its ability to provide users with hands-on experience in understanding how firewall mechanisms operate and their role in protecting network infrastructure. By simulating real-time scenarios, such as testing IP connectivity through ping operations and managing port access, this project addresses the critical need for network security in today's digital environment.

The problem being addressed is the lack of user-friendly, real-time tools for managing basic network security measures. Many users, especially beginners in computer networking, find it challenging to interact with complex firewall systems. This project is motivated by the need to bridge this gap by creating a straightforward, command-based firewall implementation that demonstrates the fundamental principles of network protection in an accessible manner.

2 Project Description

The primary goal of this project is to implement a basic firewall setup along with its simulation, enabling users to manage device traffic in real-time through an intuitive and user-friendly graphical interface. The project focuses on simplifying network security management by providing accessible features for controlling IP and port activity.

2.1 Background and Motivation

2.1.1 Real-World Relevance

Basic firewall management is a critical component of network security, protecting devices and networks from unauthorized access and cyber threats. This project highlights the importance of firewalls in ensuring secure and efficient communication by managing traffic at the IP and port levels.

2.1.2 Educational Value

The project serves as a valuable learning tool, bridging the gap between theoretical concepts and practical application of networking and security fundamentals. By offering real-time simulations and visualizations, it provides users with an interactive platform to understand and implement essential firewall mechanisms.

2.2 Scope of Work

2.2.1 Included

- Blocking and unblocking of IP addresses and ports.
- Application and clearing of firewall rules dynamically.
- Real-time simulation for both IPs and ports, including connectivity checks.
- Visualization of firewall statistics through a donut chart, with the option to download.
- A simple and attractive graphical user interface (GUI) for enhanced usability.
- **Persistent Blocking:** File handling using JSON ensures that changes to blocked IPs and ports are saved, providing a real-world experience where firewall settings are retained even after the application is closed.

2.2.2 Excluded

- Advanced firewall capabilities, such as deep packet inspection or protocol-specific rules.
- Multi-device management or integration with external tools.

The project bridges the gap between theoretical concepts of firewalls and practical implementation, providing an interactive platform to understand and manage basic network security operations effectively.

3 Methodology

3.1 Blocking IPs and Ports in Real-Time

3.1.1 Core Functionality

This firewall system leverages the Windows commands to interact directly with the operating system's firewall. The core functionality focuses on dynamically blocking and unblocking specific IP addresses and ports.

3.1.2 Blocking and Unblocking IPs

Users interact with the system through a user-friendly GUI. Upon initiating a block, the system executes the appropriate Windows commands (e.g., using the netsh command) to create firewall rules that deny traffic from the specified IP address on the TCP protocol. Unblocking an IP involves removing the corresponding firewall rule.

3.1.3 Blocking and Unblocking Ports

Similar to IP blocking, users can specify ports to be blocked. The system then creates firewall rules to deny traffic on the specified port for all sources. Unblocking a port involves removing the respective firewall rule.

3.2 Real-Time Simulation

3.2.1 IP Blocking Simulation

The system validates the effectiveness of IP blocking by utilizing the ping command. By attempting to ping the blocked IP address, the system analyzes the ping response. If the ping is unsuccessful, it indicates successful IP blocking. This real-time feedback is displayed to the user within the GUI.

3.2.2 Port Blocking Simulation

To verify port blocking, the system checks the list of blocked ports. If the specified port is found within this list, the system indicates that the port is currently blocked. This real-time check provides immediate feedback to the user.

3.3 User Interface and Interaction

3.3.1 GUI Implementation

A user-friendly GUI is implemented using Streamlit. This provides an intuitive interface for users to interact with the firewall system.

3.3.2 Command-Line Interaction

While the user interacts with the GUI, the system executes the necessary Windows commands in the background to implement the requested actions (blocking/unblocking).

3.3.3 User Validation

1. **IP Address Validation**

   Regular expressions are employed to ensure the validity of entered IP addresses.

2. **Port Number Validation**

   Input is checked to ensure it is an integer within the valid port number range.

3. **Syntax Validation**

   Input is checked for proper syntax to prevent errors in command execution.

4. **Error Handling**

   The system includes mechanisms to handle invalid inputs and display appropriate error messages to the user.

3.4 Data Visualization

3.4.1 Firewall Statistics

A donut chart is generated using Plotly to visually represent firewall statistics.

3.4.2 Data Representation

The chart displays:

- The number of currently blocked IP addresses.
- The number of currently blocked ports.

3.4.3 Export Chart

The chart can be downloaded for further analysis or documentation.

3.5 Technology Stack

1. **Python**
   The core of the system is built using Python.
2. **Streamlit**
   Used for creating the interactive GUI.
3. **Libraries**
   - **re:** For regular expression-based input validation.
   - **subprocess:** For executing system commands (e.g., ping, netsh).
   - **os, platform:** For system-level interactions.
   - **json** For potential data storage and retrieval.
   - **plotly.graph_objects:** For creating and displaying the donut chart.
4. **Windows Commands**
   The system leverages Windows command-line tools, primarily netsh and advfirewall to interact with the operating system's firewall.

3.6 Challenges and Solutions

3.6.1 Administrative Privileges

Running the system requires administrative privileges on the Windows machine. This was addressed by informing users of the requirement and guiding them through the necessary steps.

3.6.2 Computer Networking Knowledge

The project requires some understanding of computer networking concepts. To make the system more accessible to non-technical users, the GUI was designed to be as intuitive and user-friendly as possible.

3.7 Testing and Validation

3.7.1 IP/Port Blocking/Unblocking

The functionality of blocking and unblocking IP addresses and ports was rigorously tested by:

- Blocking IPs and attempting to ping them.
- Blocking ports and verifying that applications using those ports are unable to connect.
- Unblocking IPs and ports and verifying that connectivity is restored.

3.7.2 Real-Time Validation

The real-time simulation and feedback mechanisms were thoroughly tested to ensure their accuracy and responsiveness.

3.7.3 GUI Testing

The GUI was tested for usability, responsiveness, and error handling.

3.7.4 Donut Chart Validation

The donut chart was tested to ensure accurate data representation and proper updates upon changes in the firewall rules.

4 Key Features and Functionalities

4.1 Dynamic Firewall Rule Management

- Enables real-time creation and removal of firewall rules for both IP addresses and ports.
- Automates the process of interacting with the Windows firewall through the netsh command.

4.2 User-Friendly GUI with Real-time Feedback

- Provides an intuitive Streamlit-based interface for easy interaction with the firewall system.
- Offers real-time feedback on IP blocking attempts through ping checks.
- Displays the status of port blocking by checking against the list of blocked ports.

4.3 Enhanced Network Visibility

- Presents a dynamic donut chart to visualize the number of blocked IPs and ports, providing a clear overview of the firewall's current state.
- Allows users to easily download the chart for further analysis or documentation.

4.4 Robust Input Validation and Error Handling

- Ensures the accuracy and validity of user input through rigorous checks on IP addresses and port numbers.
- Provides informative error messages to guide users in correcting invalid inputs.

4.5 Streamlined Workflow

Simplifies complex firewall management tasks, making it accessible to users with varying levels of technical expertise.

4.6 Innovation and Unqiue Features of System

4.6.1 Combination of Real-Time Control and Visualization

The system effectively integrates real-time firewall rule manipulation with a dynamic and informative visualization of the firewall's current state.

4.6.2 User-Centric Design

The Streamlit-based GUI prioritizes user experience with intuitive interactions, clear feedback, and robust error handling.

4.6.3 Proactive Security Posture

The system encourages a proactive approach to network security by providing real-time feedback on blocking attempts and enabling users to quickly assess the impact of their actions

4.7 Examples of User Interaction

4.7.1 Blocking an IP Address

1. The user enters the target IP address in the designated field within the GUI.
2. The system validates the IP address and, upon confirmation, executes the necessary commands to block traffic from that IP address.
3. The system attempts to ping the blocked IP address and displays the result (successful block or failure) to the user.

4.7.2 Unblocking a Port

1. The user selects the port to be unblocked from the list of blocked ports.
2. The system removes the corresponding firewall rule.
3. The donut chart and other relevant displays are updated to reflect the changes in real-time.

5 Technologies and Tools

5.1 Programming Language : Python

Python's versatility, extensive libraries (like subprocess for system interaction, re for regular expressions, json for data handling, os for system-level interactions, and platform for system information), and ease of use made it the ideal choice for this project.

5.2 Streamlit

Streamlit's simplicity and rapid development capabilities allowed for the quick creation of an interactive and user-friendly GUI. Its ability to easily integrate with Python code made it a perfect fit for this project.

5.3 Windows Commands

The netsh command provides direct and powerful interaction with the Windows Firewall. This allows for precise control over firewall rules and efficient implementation of blocking/unblocking functionalities.

5.4 Libraries

1. **subprocess:** Used to execute system commands (e.g., ping, netsh) and interact with the operating system.
2. **re:** Employed for regular expression-based input validation of IP addresses.
3. **plotly.graph_objects:** Utilized to create and display the interactive donut chart for visualizing firewall statistics.
4. **json:** Potentially used for data storage and retrieval
5. **os:** Used for system-level interactions and file handling
6. **platform:** Used to gather system information.

6 Flow Chart Diagram

6.1 System Architecture Diagram

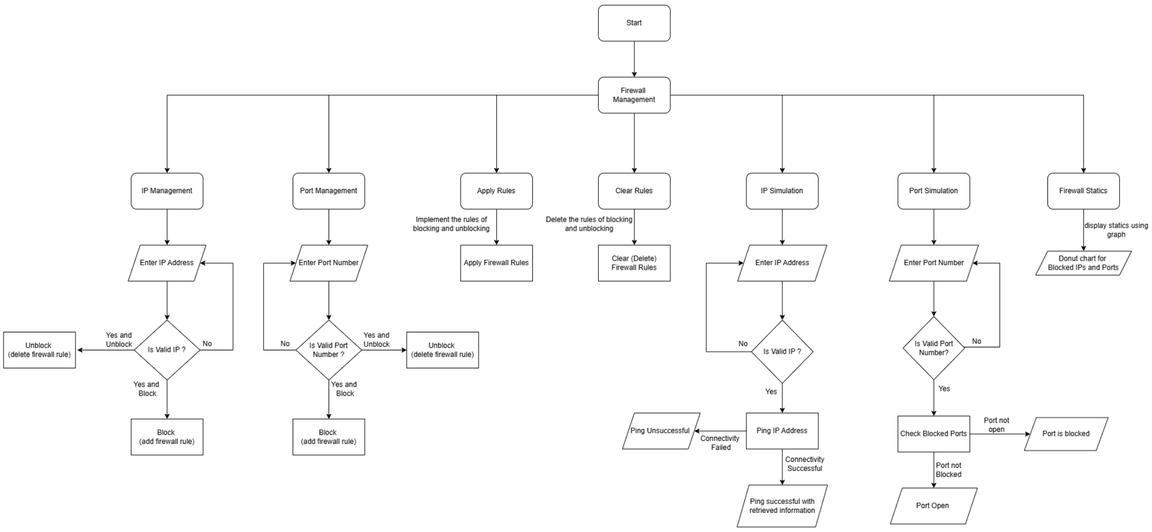


Figure 1: System Workflow Representation

6.2 Components Explanation

6.2.1 Firewall Management

This is the ”Home Page” or the main starting point where the user can choose between different functionalities.

6.2.2 IP Management

This section encompasses all actions related to managing IP addresses.

1. **Blocking IPs**
   Adding rules to the firewall to block traffic from specific IP addresses.
2. **Unblocking IPs**
   Removing rules to allow traffic from previously blocked IP addresses.

6.2.3 Port Management

This section deals with managing port-level traffic.

1. **Blocking Ports**
   Adding rules to the firewall to block traffic on specific ports.
2. **Unblocking Ports**
   Removing rules to allow traffic on previously blocked ports.

6.2.4 Apply Rules

This component applies the configured rules to the firewall.

6.2.5 Clear Rules

This component removes all existing firewall rules, effectively resetting the firewall settings.

6.2.6 IP Simulation

This section allows users to simulate network connectivity by pinging target IP addresses and displaying the results.

6.2.7 Port Simulation

This section allows users to simulate network connectivity by attempting to connect to specific ports on target IP addresses.

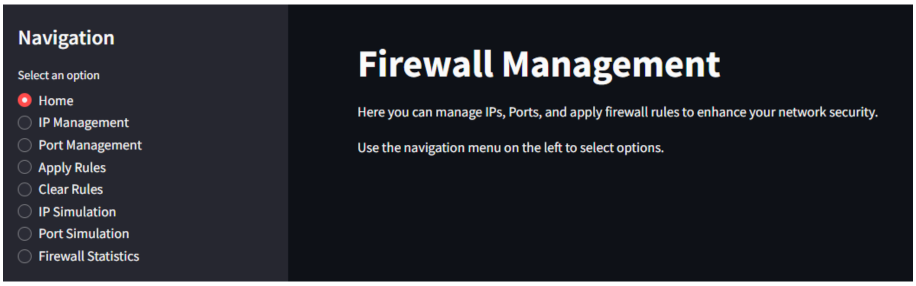
6.2.8 Firewall Statistics

This section provides a visual representation of the current firewall state, such as the number of blocked IPs and ports, typically displayed using a donut chart. This also allow users to download the chart for further analysis.

7 Tool Screenshots

The snippets for some user interfaces and results are given.

7.1 Home Page: Firewall Management



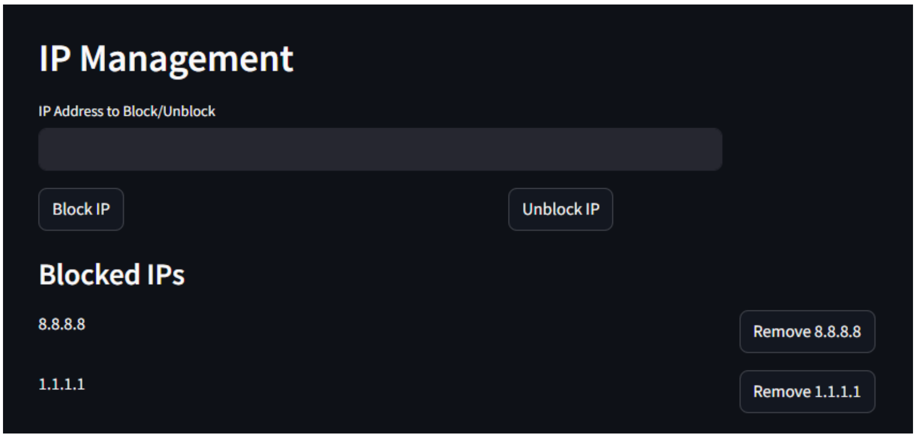
Firewall Management

Here you can manage IPs, Ports, and apply firewall rules to enhance your network security.

Use the navigation menu on the left to select options.

Figure 2: Firewall Management

7.2 IP Management



IP Management

IP Address to Block/Unblock

Blocked IPs

8.8.8.8

1.1.1.1

Figure 3: IP Management

7.3 Port Management

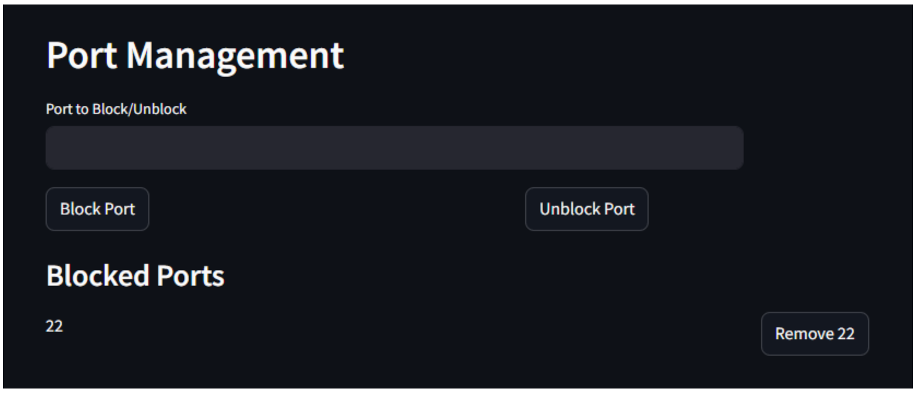
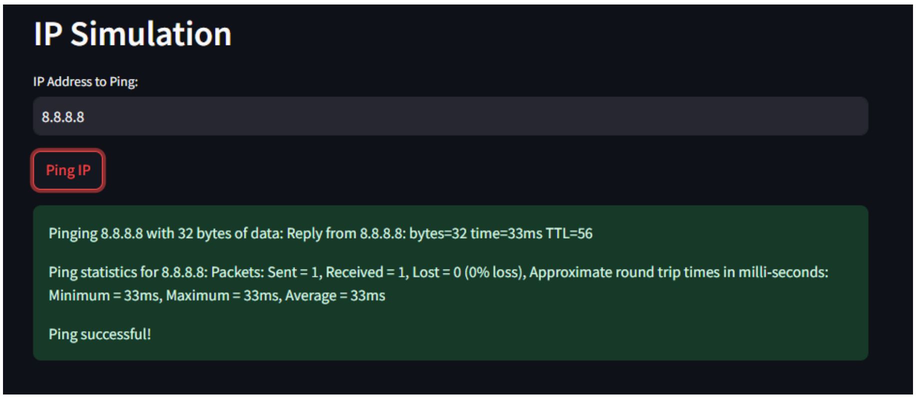


Figure 4: Port Management

7.4 IP Simulation



```
Pinging 8.8.8.8 with 32 bytes of data: Reply from 8.8.8.8: bytes=32 time=33ms TTL=56

Ping statistics for 8.8.8.8: Packets: Sent = 1, Received = 1, Lost = 0 (0% loss), Approximate round trip times in milli-seconds:
Minimum = 33ms, Maximum = 33ms, Average = 33ms

Ping successful!
```

Figure 5: IP Simulation using Ping

7.5 Analysis using Graph

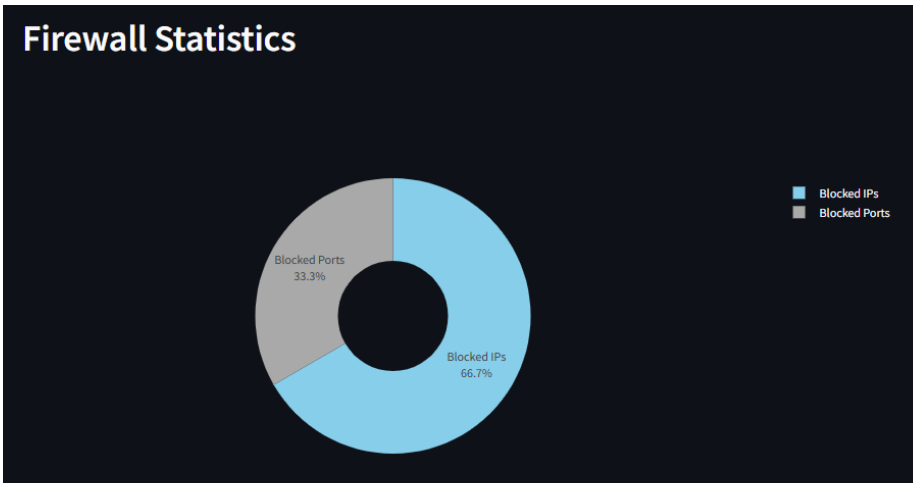


Figure 6: Firewall Statics

8 Results and Analysis

8.1 Project Objectives

- To develop a user-friendly firewall system with real-time capabilities.
- To implement features for blocking/unblocking IP addresses and ports.
- To provide real-time feedback on blocking/unblocking actions.
- To visualize firewall statistics using a dynamic donut chart.

8.2 Results

The project successfully implemented a functional firewall system with the following features.

1. Real-time blocking and unblocking of IP addresses and ports.
2. Real-time feedback on IP blocking attempts through ping checks.
3. A user-friendly GUI for easy interaction.
4. A dynamic donut chart to visualize firewall statistics.

8.3 Functionality

1. **IP/Port Blocking and Unblocking**

   The system effectively blocks and unblocks IP addresses and ports by interacting with the Windows Firewall using the netsh command.

2. **IP Simulation**

   The ping functionality accurately verifies the success of IP blocking.

3. **Port Simulation**

   The system can simulate connections to specific ports to check for connectivity.

4. **Statistics Visualization**

   The donut chart provides a clear and concise visual representation of the number of blocked IPs and ports.

8.4 Insights and Interpretations

1. **User Experience**

   The user-friendly GUI and real-time feedback enhance the user experience, making the system easy to use and understand.

2. **System Performance**

   The system demonstrated real-time responsiveness in most cases, providing timely feedback to user actions.

3. **Data Visualization**

   The donut chart effectively visualizes the current state of the firewall, providing a quick and intuitive understanding of the security posture.

8.5 Alignment with Objectives

The project successfully achieved its objectives by:

- Developing a user-friendly system with a well-designed GUI.
- Implementing core functionalities for blocking/unblocking IP addresses and ports.
- Providing real-time feedback through ping checks and visual updates.
- Visualizing firewall statistics effectively using a dynamic donut chart.

9 Conclusion

This project successfully demonstrated the development of a basic firewall system with key features like IP/Port blocking/unblocking, real-time feedback, and a user-friendly interface. The project highlighted the power of Python and libraries like Streamlit for building interactive applications and interacting with the operating system. Key takeaways include the importance of clear user interfaces, real-time feedback mechanisms, and effective data visualization for network security tools. Future enhancements could include advanced filtering, log management, and integration with cloud-based security services.

10 Github Repository Link

https://github.com/Rafiya-Rehan21/basic_firewall.git

11 References

1. https://en.wikipedia.org/wiki/Firewall_(computing)
2. https://gaia.cs.umass.edu/kurose_ross/lectures.php