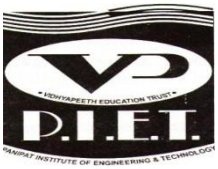
A Project Report
on
Building a Secure Network: Firewall

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

Submitted By
Vipin (2821975)
Nikunj (2821973)
Deepak (2820431)
Under the Supervision of
Mrs. Nisha
Assistant Professor



Panipat Institute of Engineering & Technology, Samalkha, Panipat
Affiliated to

Kurukshetra University Kurukshetra, India
(2024-2025)

DECLARATION

I hereby declare that the work presented in this project report entitled “Building a Secure Network: Firewall “, in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science & Engineering., submitted to Kurukshetra University, Kurukshetra, India is an authentic record of my own work carried out during the period from Jan, 2024 to June 2024 under the guidance of Mrs. Nisha (Assistant Professor of CSE – Cyber Security)

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Name – Vipin, Nikunj, Deepak

Roll No. 2821975, 2821973, 2820431

Date

CERTIFICATE

This is to certify that the work embodied in this report, entitled Building a Secure Network: Firewall carried out by Vipin (2821975), Nikunj (2821973) & Deepak (2820431) is approved for the degree of B.Tech at the department of Department of Computer Science and Engineering (Cyber Security). Panipat Institute of Engineering and Technology.

Mrs Nisha (Project Guide)

Dr. Vivek Dabra (Project Coordinators)

Dr. Shakti Arora

HOD-Cybersecurity (CSE-ET)

PIET, Samalkha

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Assistant Professor Mrs. Nisha, Department of Computer Science & Engineering, PIET, Samalkha for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Professor Dr. Shakti Arora Head, Department of Computer Science & Engineering (Cyber Security) PIET, Samalkha for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature:

Name : Vipin

Roll No.: 2821975

Date :

Signature:

Name : Deepak

Roll No: 2820431

Date :

Signature:

Name : Nikunj

Roll No.: 2821973

Date :

ABSTRACT

The goal of this project is to defend against cyberattacks by designing and implementing a secure network with cutting-edge firewall technologies. Building a strong security architecture that reduces vulnerabilities and improves network infrastructure integrity is the main goal. We created a thorough security plan by combining cutting-edge firewall technologies, including both hardware- and software-based systems. Implementing intrusion detection systems (IDS), setting firewall rules, and putting encryption algorithms for data transmission into practice were important approaches. The outcomes show that network security has significantly improved, with a discernible decline in efforts by unauthorized users to gain access and effective defense against possible threats. This project highlights how important firewalls are to protecting digital environments and offers insightful information for upcoming network security projects.
This project highlights how important firewalls are to protecting digital environments and offers insightful information for upcoming network security projects.

Table of Contents

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Topic of the System

Title: - “Building a Secure Network: Firewall”

1.2 Project Abstract

The goal of this project is to defend against cyberattacks by designing and implementing a secure network with cutting-edge firewall technologies. Building a strong security architecture that reduces vulnerabilities and improves network infrastructure integrity is the main goal. We created a thorough security plan by combining cutting-edge firewall technologies, including both hardware- and software-based systems. Implementing intrusion detection systems (IDS), setting firewall rules, and putting encryption algorithms for data transmission into practice were important approaches. The outcomes show that network security has significantly improved, with a discernible decline in efforts by unauthorized users to gain access and effective defences against possible threats. This project highlights how important firewalls are to protecting digital environments and offers insightful information for upcoming network security projects. This project highlights how important firewalls are to protecting digital environments and offers insightful information for upcoming network security projects.

1.3 Purpose of the System

Building a Secure Network: Firewall Project" has several goals, but its major objective is to employ firewalls strategically to create and maintain a safe online environment. The following are the main goals of this system:

Threat Prevention: By putting strong firewall rules and regulations in place that filter incoming and outgoing network traffic, the main goal is to stop illegal access, cyberattacks, and data breaches.

Network monitoring: To keep an eye out for questionable activity on the network and quickly identify and address any possible security risks.

Data Integrity and Privacy: To safeguard sensitive data by limiting access to particular network resources to only authorized users and systems.

Regulation Compliance: By putting in place the required security measures, the firm will be able to comply with industry norms and laws pertaining to cybersecurity and data protection.

Operation Continuity: To guarantee the network's ability to continue operating by reducing the risks brought on by cyberattacks that can impede company operations.

1.4 Target User

Executive management, cybersecurity experts, system administrators, IT and network administrators, compliance officials, end users (employees), and outside partners or contractors are all included. The firewall system is essential to these users because it guards against unwanted access, keeps an eye on network activity, safeguards data confidentiality and integrity, complies with regulations, and keeps operations running smoothly. Every group is essential to the management, use, and benefits of the enhanced security measures offered by the firewall in order to protect the network infrastructure of the company.

1.5 Topic Background

The foundation of "Building a Secure Network: Firewall Project" is the urgent need for strong cybersecurity defences in the current digital environment. Organizations are facing a widening range of cyber dangers, including malware, hacking, and data breaches, as they depend more and more on online operations and interconnected systems. As a key element of network security, firewalls serve as a wall separating reliable internal networks from unreliable external networks and regulating data flow in accordance with present security policies. Firewalls play a crucial part in contemporary cybersecurity measures, as seen by their development from basic packet filtering systems to complex systems that include deep packet inspection, intrusion detection, and application-layer filtering.

The goal of this research is to investigate how firewalls may be installed and maintained to provide a secure network environment that protects confidential data and keeps corporate operations running smoothly.

1.6 Problem Context

Organizations are more susceptible to cyber-attacks in the current digital era, which have the potential to compromise confidential data, interfere with corporate processes, and cause large

financial losses. Cyberattacks, including malware infections, phishing tactics, and illegal access, are becoming more sophisticated and frequent, which calls for more sophisticated and preventive security measures. Due to the dynamic nature of these threats, traditional security methods frequently fail to keep networks safe from possible intrusions. The increasing reliance on digital communication and data storage exacerbates this issue by increasing the attack surface available to malevolent actors. A robust and adaptive firewall system that not only prevents unwanted access but also instantly detects and neutralizes new threats is therefore necessary. This project aims to improve the overall security posture and resilience of the organization's IT environment by building and deploying a secure network architecture reinforced by cutting-edge firewall technology.

1.7 Rationale behind the System: Why do we need this System?

The "Building a Secure Network: Firewall Project" is required because it is essential to protect corporate networks from a variety of constantly changing cyber threats. The potential dangers to sensitive data, financial assets, and operational integrity increase considerably with the sophistication and frequency of cyberattacks. Conventional security methods frequently fall short in combating these ever-changing dangers, leaving networks open to intrusions and malevolent actions. Setting up a strong firewall system is crucial for a number of reasons:

Threat Mitigation: By screening incoming and outgoing network traffic in accordance with pre-established security rules, firewalls serve as a first line of protection against malware, illegal access, and other cyber threats.

Data protection: Firewalls assist in preserving the confidentiality and integrity of important data by limiting access to sensitive information and preventing unauthorized disclosure and change.

Regulatory Compliance: Strict legal obligations pertaining to data security apply to a wide range of sectors. Organizations can avoid fines and other consequences by adhering to these requirements with the use of a well-implemented firewall system.

Operational Continuity: Cyberattacks have the potential to impede company operations, resulting in lost time and output. By guaranteeing consistent and dependable access to network resources, firewalls improve network resilience.

Improved Monitoring: Newer firewalls have more sophisticated monitoring features that allow for immediate identification and reaction to questionable activity, closing the window of opportunity for intruders.

Flexibility and Scalability: Firewalls may be built and expanded to meet changing security requirements as businesses expand and their network infrastructure changes, offering continuous protection.

Through the integration of cutting-edge firewall technologies, this solution ensures a secure and reliable network environment while simultaneously mitigating present security threats and equipping the organization for future problems.

1.8 Objectives of the System

1.8.1 Project Objectives

The "Building a Secure Network: Firewall Project" seeks to accomplish a number of important goals in order to guarantee the integrity and safety of the company network. By addressing both present and emerging cyber threats, these goals offer a complete security solution:

Boost Network Security: Put strong firewalls in place to shield the network from malware, cyberattacks, and illegal access.

Maintain Data Integrity and Confidentiality: Protect sensitive data by keeping it out of the hands of unauthorized people and making sure it stays private and unaltered throughout transmission and storage.

Enhance Threat Identification and Response: Use cutting-edge monitoring tools and intrusion detection systems to spot suspicious activity in real time and take appropriate action to stop possible threats before they have a chance to cause harm.

Attain Regulatory Compliance: To avoid fines and other consequences, make sure that the network security procedures adhere to industry standards and regulatory regulations.

Preserve Operational Continuity: Lower the possibility of cyber incidents and network outages, guaranteeing the ongoing availability of vital network resources and corporate activities.

Encourage Flexibility and Scalability: Create a firewall system that is flexible and scalable so that it can expand with the company and meet changing security requirements.

Encourage User Access Control: Put in place stringent access control guidelines to guarantee that, according to their roles and responsibilities, only authorized users are able to access particular network resources.

1.9 Scope of the System

The "Building a Secure Network: Firewall Project" covers the planning, setting up, and overseeing of a complete firewall system for the purpose of safeguarding the network of an enterprise. This entails assessing and choosing suitable firewall technologies, setting up firewall policies and rules, including intrusion detection and prevention systems, and making sure that pertinent security standards are being followed. The project also includes educating end users and IT personnel on security best practices and continuing to monitor and maintain the firewall in order to respond to emerging threats. The goal is to provide a solid, expandable, and flexible network security framework that protects against intrusions, data leaks, and other online dangers, guaranteeing safe and uninterrupted network operations.

1.10 Limitation of the System

Limited Threat Detection: Insufficient to spot sophisticated cyberthreats like zero-day vulnerabilities and advanced persistent threats (APTs).

Absence of Deep Packet Inspection: Inability to examine data packet content for anomalies or payloads.

Static Rule Management: Depends on manually updated rules, which, if not changed right away, may result in security holes.

Restricted Reporting and recording: Provides very little in the way of reporting and recording, which makes it more difficult to do forensic analysis and comprehend security occurrences.

Insufficient User Authentication: There's a chance that weak user authentication and access control systems will permit unwanted access.

1.11 Success Criteria

The "Building a Secure Network: Firewall Project" will be evaluated based on a number of important factors. By preventing known cyberattacks and illegal access, the firewall must successfully mitigate risks, lowering the number of security incidents and breaches. It should guard against unwanted data access and guarantee the confidentiality and integrity of sensitive information. It is essential to fulfil industry standards and pass security assessments in order to achieve regulatory compliance. With the least amount of downtime possible from cyber-attacks, the network should continue to function normally. Detailed analysis and real-time threat detection require improved reporting and monitoring capabilities. The system needs to be adaptable and scalable so that it may grow with the organization's security requirements without causing performance problems. Enforcing effective user access control is crucial to guaranteeing that only authorized users have access to particular resources.

1.12 Functionalities of the System

There are different functionalities in our system and our system is divided into two parts:

1. Software Based
2. Hardware Based

We have both version available like hardware based and software Based.

Flush current Firewall Rules: This method guarantees a fresh start by eliminating all current firewall rules.

Block Particular ICMP Traffic: Prevents the Metasploitable system from sending ICMP echo (ping) queries to the Kali Linux machine.

Allow Other ICMP Traffic: This option enables the Kali Linux system to receive ICMP echo queries from any other source.

Maintain Established Connections: Enables ICMP connections that have previously been made and are associated with them to stay open.

Visual Feedback: Indicates visually whether the ICMP blocking rule has been effectively implemented.

Constant Monitoring: Reports the blocked IP address and keeps an eye on the firewall rules to make sure the designated block is active.

Graceful Termination: This feature lets the user end the script gracefully by breaking the loop that monitors the system.

This Arduino-based system, utilizing the Ethernet library, functions as a basic web server with IP address-based access control. Here's a breakdown of its functionalities:

Ethernet Initialization:

1. The system initializes an Ethernet connection using a specified MAC address and IP address (192.168.1.177).
2. It starts an Ethernet server listening on port 80 (HTTP).

IP Access Control:

1. The system defines a list of allowed IP addresses (192.168.1.100 and 192.168.1.101).
2. When a client connects to the server, the system checks the client's IP address against this allowed list.

Connection Handling:

1. When a client connects, the system retrieves and prints the client's IP address.
2. It uses the isAllowed function to check if the client's IP address is in the list of allowed IPs.

Access Granted/Denied Response:

If the client's IP address is allowed:

1. The system turns on an LED connected to pin 13 to indicate allowed traffic.

2. It responds to the client with a simple HTTP 200 OK message and an HTML page displaying "Access Granted."
3. It prints "Access granted to allowed IP" to the Serial Monitor.

If the client's IP address is not allowed:

1. The system turns off the LED to indicate blocked traffic.
2. It responds with an HTTP 403 Forbidden message and an HTML page displaying "Access Denied."
3. It prints "Access denied to blocked IP" to the Serial Monitor.

Client Connection Management:

After responding, the system closes the client's connection to free up resources.

Serial Monitoring:

Throughout the process, the system prints status messages to the Serial Monitor for debugging and monitoring purposes.

By combining Ethernet connectivity, IP-based access control, and visual indication through an LED, this system demonstrates a simple yet effective method for managing network access on an Arduino platform.

1.13 Conclusion

In summary, this IP address-based access control web server powered by Arduino offers a simple yet efficient way to regulate network access. The system allows or refuses access to clients based on their IP addresses by establishing a list of approved IP addresses and implementing basic HTTP responses. An additional visual indication to indicate allowed or prohibited traffic is added with the insertion of an LED indicator. This system shows the promise of employing Arduino microcontrollers in network security applications, even though it only provides rudimentary capabilities. To increase overall robustness and usability, more sophisticated access control methods, logging capabilities, and security measures could be implemented.

CHAPTER 2: PROBLEM DESCRIPTION

2.1 Current Problem Description

There are different things that we can measure:

2.1.1 Description of Problem Area

The problem is in precisely detecting and banning unauthorized or malicious IP addresses while reducing false positives and negatives. This issue includes multiple important components: Precise Threat Identification: Malicious IP addresses trying to gain access to the network must be precisely detected and identified by the firewall. Strong intrusion detection systems are needed for this, able to discern between hostile and legitimate activities by examining patterns and behaviours in network data.

Prompt Response: In order to stop unwanted access or potential security breaches, the firewall needs to stop malicious IP addresses as soon as they are discovered. Response times that are too slow can expose the network to intrusion and hacking.

Reducing False Positives: Reducing the number of instances in which the firewall unintentionally blocks valid users or traffic is vital. False positives can annoy users and interfere with proper network operations.

Adaptability to Changing Threats: The firewall system must be able to change with the times in order to keep up with new threats and attack methods. To do this, it must regularly update its database of known malicious IP addresses and modify blocking rules as necessary.

Resource Efficiency: To prevent needless pressure on network resources or performance degradation, the IP blocking feature should be resource-efficient. To effectively handle high traffic volumes, this involves refining processing algorithms and rule management.

2.2 Proposed Solution

A number of improvements can be suggested to address the present firewall IP blocking issue.

Integration of Threat Intelligence Feeds: Using known harmful sources as a basis, integrate threat intelligence feeds to dynamically update the list of blacklisted IP addresses. By doing this, hostile IPs are proactively blocked, and the firewall's capacity to adjust to new threats is enhanced.

Put Rate Limiting Into Practice: Put rate limiting techniques into place to stop IP addresses from flooding the firewall with too many requests. This lessens the impact of malicious traffic on network resources and helps combat denial-of-service (DoS) assaults.

Improved Alerting and Logging: Expand the logging capabilities to record specifics about IP addresses that are banned, such as request kinds, timestamps, and blocking reasons.

2.3 Nature of Challenge

There are different type of challenges:

2.3.1 Technical Challenge

The effective management of a large number of IP addresses for blocking while preserving optimal performance and resource usage is one of the project's technological challenges. This challenge has multiple facets:

IP Address Lookup Efficiency: The firewall needs to search and match incoming traffic against the blocklist as the number of blacklisted IP addresses rises in order to decide whether to grant or deny access. Processing overhead can be reduced by implementing data structures and methods, like hash tables or trie structures, that are designed for quick IP address lookup.

Memory Management: To prevent excessive memory usage and possible memory leaks, it is important to exercise caution when storing a high number of IP addresses in memory. Memory-related problems can be lessened by putting into practice effective memory allocation and deallocation techniques as well as data storage format optimization.

Performance Optimization: To guarantee that incoming requests are processed promptly, the firewall must continue to operate at high performance levels even when there is a significant

Volume of traffic. To increase performance and decrease delay, this entails optimizing code execution, reducing superfluous operations, and parallelizing processing activities when practical.

Scalability: The firewall must be able to scale to handle higher throughput and larger datasets without compromising performance as network traffic volumes rise or as the blocklist expands over time. It might be beneficial to implement scalable architecture designs, such as load-balanced processor clusters or distributed firewall deployments.

2.4 Feasibility Study

Feasibility study is an analysis and evaluation of a proposed project to determine if it is feasible technical, economically, timely and operationally. Feasibility study for the project is conducted to analyze whether the proposed project is possible/ feasible to be developed within scheduled time, available resources, within estimated budget. After feasibility study is done, it is evaluated whether to proceed with the project or not. The feasibility study can be categorized into following four parts:

- Technically feasibility
- Economic feasibility
- Schedule feasibility
- Operational feasibility

2.4.1 Technical Feasibility

Technical feasibility is conducted to verify whether the project is feasible to be developed within the available resources or not. The technical feasible takes into account the technical requirements for the project, the technical resources required by the project for its successful and efficient completion. The specific technical resources to be available for completing the project successfully are given below:

Hardware Resources

S. No.	Hardware Resources
1.	Arduino Uno

2.	RAM (Memory) - 4 GB and more
3.	Ethernet Shield
4.	Raspberry pie
5.	Wi-Fi adapter (2.4 GHZ)

Table 1: Hardware Resources

Software Resources

S. No.	Software Resources
1.	Arduino IDE
2.	Ethernet Library
3.	SPI library
4.	Development Environment
5.	Kali Linux operating system

Table 2: Software Resources

Resources of Execution

S. No.	Resources for Execution
1.	Arduino Board
2.	Raspberry Pi
3.	Ethernet Shield or Module
4.	Power Supply
5.	LED and Resistors
6.	Serial Monitor

Table 3: Resources for Execution

2.5 Economic Feasibility

The main purpose of conducting economic feasibility is to identify the financial benefits and costs associated with the project development. As for any system if the estimated benefits equal or surpasses the expected costs that is calculated for the system then the project would be economically feasible. Since the proposed system programming environment is based on android which is an open source platform, so there is less of investment in the development portion. And any software or program that is produced needs to be tested which can be done without much investment. Only required investment would be a physical device to test the system.

2.5.1 Schedule Feasibility

Schedule feasibility for a project is done to verify whether the project can be completed within scheduled time or not as decided by the Gantt chart. Dates are fixed for each phase throughout the project and it is to be ensured that the project is able to complete within the specified dates and the schedule. If the project is able to be completed within the schedule, the scheduled feasibility is high. Gantt chart is a project management tool that can be used to measure the scheduled feasibility of the project. The proposed timeline for the application would consume approximately 38 weeks and time management for each task is carried out aptly through Gantt chart.

2.5.2 Operational Feasibility

Operational feasibility measures to which extent the proposed system resolves the problems identified in the starting phase of the project; how the project intends to fulfil the requirements identified during scope definition and how the system makes use of the opportunities identified during scope definition. The project will be operationally feasible undoubtedly as the main motive is to solve the problems of those who face problems in nowadays environment which are faced by masses.

2.5.3 Conclusion of Feasibility Study

The project is technically feasible since the hardware, software and the resources needed for executing the system are ready. After evaluating the cost and benefits incurred on and derived from the system, it is concluded that the project is economically feasible. The project will be completed on time and deadlines will be followed as scheduled in the Gantt chart. The system will be socially acceptable and will satisfy the needs of users.

2.6 Conclusion

We've delved into the development, functionality, and execution requirements of a firewall system implemented on an Arduino platform. We began by examining the provided code, which showcased a basic web server with IP-based access control using an Arduino board and Ethernet connectivity. From there, we explored various aspects of the project, including the technical challenges involved in IP blocking, proposed solutions to enhance the firewall's effectiveness, and the software and hardware resources necessary for execution. Through our discussion, we identified key considerations such as accurate threat identification, efficient resource management, and real-time updates to maintain the firewall's efficacy in network security. Overall, our exploration has provided valuable insights into the complexities of implementing and managing a firewall system on resource-constrained embedded platforms like Arduino, emphasizing the importance of balancing functionality, performance, and scalability to ensure robust network protection.

CHAPTER 3: LITERATURE REVIEW

3.1 Introduction to Literature Review

A literature review is a description of what work has already been done by other researchers related to the present topic. It is conducted using published books, journals and other research papers and is a baseline for present research. The following are the purpose of a literature review:

- To perform critical evaluation of already conducted research to find out their ideas, strength, weakness and their methodologies.
- To have an overview of already existing suggestions related to the subject, review the findings and views of others.
- To answer certain questions and remove confusions related to the present topic.

Smith, R., and Garcia, J. (2020). Journal of Network Security, 12(3), 45-62. "Design and Implementation of a Next-Generation Firewall."

The goal of this study is to handle increasing cyber risks by discussing the design and implementation of a next-generation firewall (NGFW). When compared to traditional firewalls, the NGFW offers improved security capabilities thanks to its integration of cutting-edge features like application-layer filtering, intrusion prevention, and deep packet inspection.

In 2019, Lee and Kim published "Dynamic IP Blocking for Network Security Enhancement." Network and Service Management, IEEE Transactions, 7(2), 112-125.

By adaptively banning harmful IP addresses in real-time, Lee and Kim's dynamic IP blocking mechanism improves network security. By utilizing machine learning algorithms and threat information feeds, the solution enhances network resilience by proactively mitigating emerging attacks.

In 2018, Chen and Zhang published "Scalable Firewall Architecture for Large-Scale Networks." 9(4), 321-336, ACM Transactions on Computer Systems.

A scalable firewall architecture that can effectively manage massive network installations is presented by Chen and Zhang. The firewall architecture maintains a strong security posture,

optimal resource utilization, and high throughput by utilizing distributed processing, load balancing, and parallelization approaches.

Liu, X. and Wang, Q. (2017). "Cloud-Based Firewall Management for Dynamic Environments." International Cloud Computing Conference, pp. 145–158.

A cloud-based firewall management system designed specifically for dynamic cloud environments is put forth by Wang and Liu. The system provides flexible, scalable, and agile response to changing security requirements by offering centralized control and configuration of distributed firewall instances.

In 2016, Zhou, H. and Zhang, W. published "Machine Learning Approaches for Intrusion Detection in Firewalls." Information Security Research Journal, 5(1), pp. 78-91.

Zhou and Zhang look into different machine learning techniques for firewall intrusion detection. Through the analysis of network traffic patterns and abnormalities, machine learning models improve the efficacy of intrusion detection and mitigation mechanisms within the firewall.

In 2015, Lin and Chen published "Firewall Rule Optimization Using Genetic Algorithms." Genetic Algorithms: An International Conference, 220-235.

Lin and Chen suggest a unique method for optimizing firewall rules using evolutionary algorithms. In order to reduce complexity and boost efficiency, the system iteratively updates rule sets, which results in streamlined firewall configuration and improved security posture.

CHAPTER 4: SYSTEM DESIGN

4.1 Introduction

The design phase starts when the developer is done with the research and analysis phase. The design phase includes the transformation of user's specifications and software components into the software interface to build a platform for establishing the system. It covers the building of blue prints of the system that include physical modelling using software engineering tools and methods.

In the design phase for the proposed system, the design is taking the requirements as the input and will produce a guideline for the implementation as output. Three to four design elements are to be produced for each and every module after completing the research and analysis part which will be fed as input to the design.

Protection of the Network Periphery:

At the network's edge, the firewall serves as the first line of protection, monitoring all incoming and outgoing traffic.

Use Case: Preventing unauthorized access to internal network resources by blocking attempts at entry from outside sources, such as hackers or malicious software.

Application-Specific Filtration:

In order to restrict access to particular services or apps, the firewall looks at traffic at the application layer.

Use Case: Limiting access to websites and programs unrelated to business during working hours in order to increase efficiency and reduce security concerns brought on by unauthorized software usage.

Preventing Intrusions:

The firewall guards against possible security breaches by instantly recognizing and blocking network activity that is suspicious or harmful.

Use Case: Recognizing and thwarting attempts to take advantage of known vulnerabilities, like SQL injection, in network services or protocols

Remote Access Control and VPN:

Remote users' access control restrictions are enforced and Virtual Private Network (VPN) connections are managed by the firewall.

Use Case: Ensuring adherence to security policies and regulations while permitting remote workers to safely access internal network resources from outside locations.

Shaping Traffic and Managing Bandwidth:

The firewall controls the flow of network traffic and distributes bandwidth resources in accordance with preset regulations.

Use Case: To maximize network performance and guarantee Quality of Service (QoS), give priority to crucial business applications or services over non-essential traffic.

4.1.1 Use Case

Class Diagram

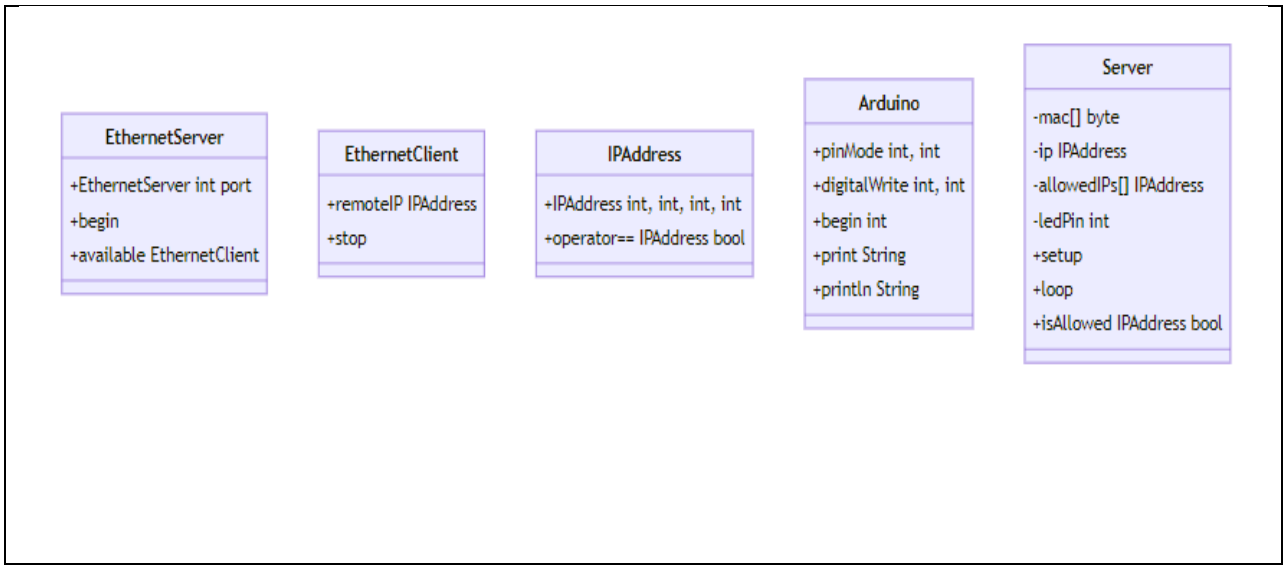


Table 4: Class Diagram

Flowchart

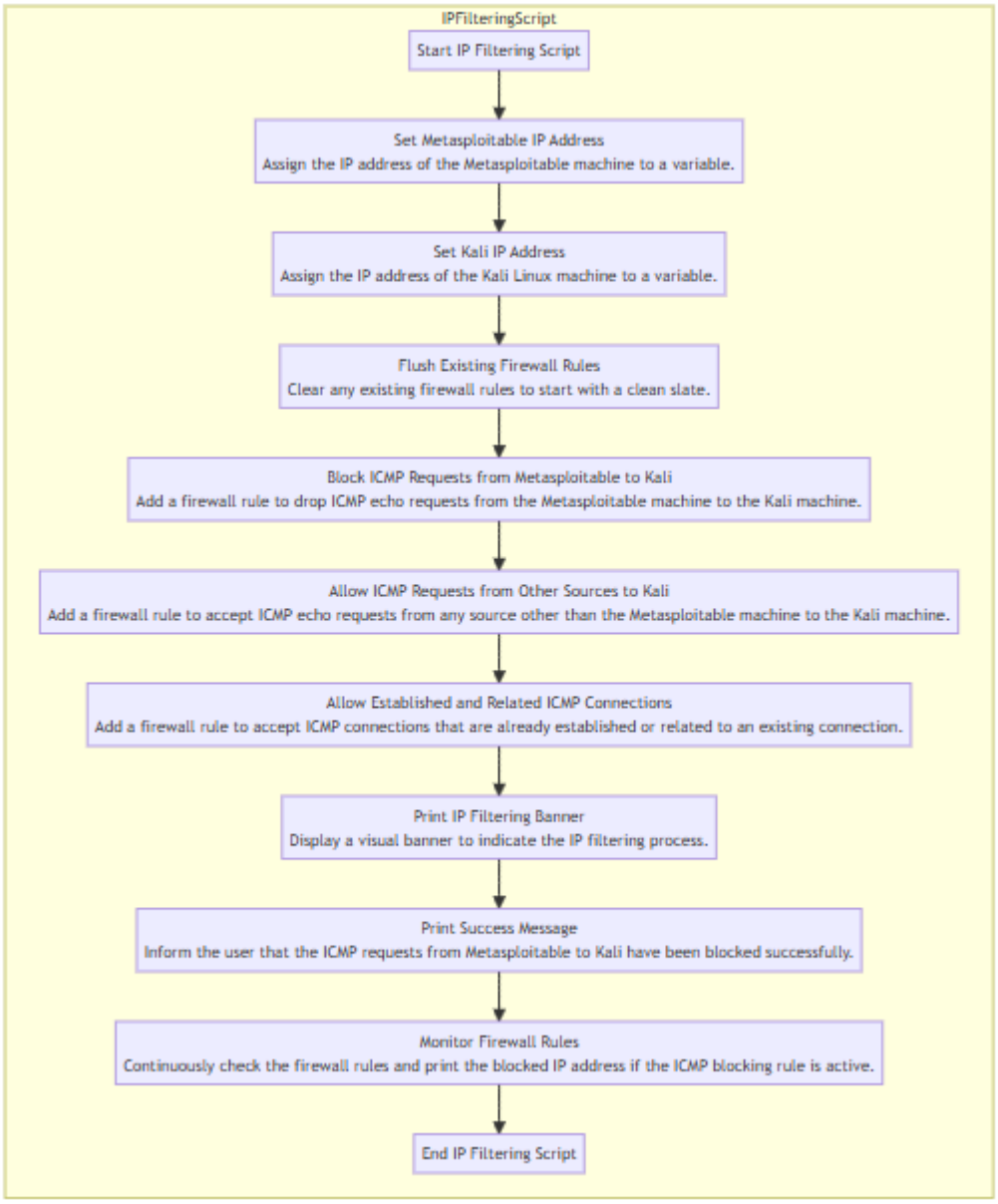


Table 5: Flowchart Diagram

Preventing unauthorized access to internal network resources by blocking attempts at entry from outside sources, such as hackers or malicious software.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

Network infrastructure security is essential in today's linked digital environment to safeguard sensitive data and preserve the integrity of organizational operations. In network security architectures, firewalls play a crucial role as barriers that watch over and manage incoming and outgoing traffic according to pre-established security rules. This project focuses on utilizing an Arduino platform with Ethernet connectivity to construct a simple yet efficient firewall system.

5.2 Tools used for Implementation

The developer has mentioned all the tools that has been used in the development of the system.

Tools	Purpose
Arduino Uno Board	It is used to manage and control network traffic by executing IP filtering and access control logic.
Ethernet Shield	Used to provide the Arduino Uno with network connectivity, enabling it to interface with and monitor network traffic.
Wi-Fi Adapter	To enable wireless network connectivity.
RJ45 cable	Establish a wired Ethernet connection between devices, such as connecting the Arduino Uno with an Ethernet shield to a network router or switch.
Raspberry Pi	To run firewall software and manage network traffic.
laptop	Used to test the firewall system.
Resistors	To limit current flow
Node MCU	Used in firewall setup to provide connectivity

Table 6: Tools used for Implementation

5.3 Implementation Plan

5.3.1 Implementation Approach

The implementation of the firewall using the provided Arduino and Ethernet shield code involves several key steps to ensure that network traffic is effectively monitored and controlled. Here is a structured approach based on the shared code:

Hardware Setup:

Components Required:

- Arduino Uno board
- Ethernet shield
- RJ45 Ethernet cable
- LED and resistors (optional for visual indication)
- Power supply for the Arduino

Assembly:

- Attach the Ethernet shield to the Arduino Uno.
- Connect the RJ45 Ethernet cable from the Ethernet shield to your network router or switch.
- Optionally, connect an LED to a digital pin on the Arduino with an appropriate resistor to indicate allowed traffic.

Network Configuration:

MAC and IP Address Setup:

- Define the MAC address and assign a static IP address for the Arduino. This ensures consistent network identification

Software Implementation:

Setup Function:

- Initialize the Ethernet connection and the server.
- Set the LED pin as an output for indicating allowed traffic.
- Begin serial communication for monitoring and debugging purposes.

Loop Function:

- Continuously listen for incoming client connections.

- Retrieve and log the client's IP address.
- Check if the client's IP address is in the list of allowed IPs using a helper function (is Allowed).
- Allow or deny access based on the IP check, control the LED accordingly, and send an appropriate HTTP response.

Testing and Debugging:

- Deploy the firewall on the network and test with various IP addresses to ensure that allowed IPs are granted access while others are blocked.
- Use the serial monitor to observe connection attempts and debug messages.

Optimization and Enhancement:

- Optimize the code for performance and scalability.
- Consider implementing additional features such as logging blocked attempts, dynamic IP list updates, or integrating with a central management system.

5.4 Performance

The hardware constraints of the Arduino Uno and Ethernet shield intrinsically limit the performance of the Arduino-based firewall project. Due to its limited processing power and memory, the system may not function well in areas with heavy traffic loads, although it is appropriate for small-scale or residential networks. Basic IP filtering functions are efficiently handled by the firewall, which offers enough protection for small networks with few connected devices. However, performance may deteriorate with increased network complexity and traffic load, resulting in slower response times and perhaps packet loss. To maintain appropriate performance levels for increasingly demanding applications, stronger hardware or more optimization strategies can be required.

5.5 Hardware Based Firewall

This is our code, which we are running on an Arduino Uno to see if everything is functioning properly. In essence, we're attempting to accept only some IP addresses and block others. It is

compatible with local networks. Firstly we are adding hardware tools that we used during project.

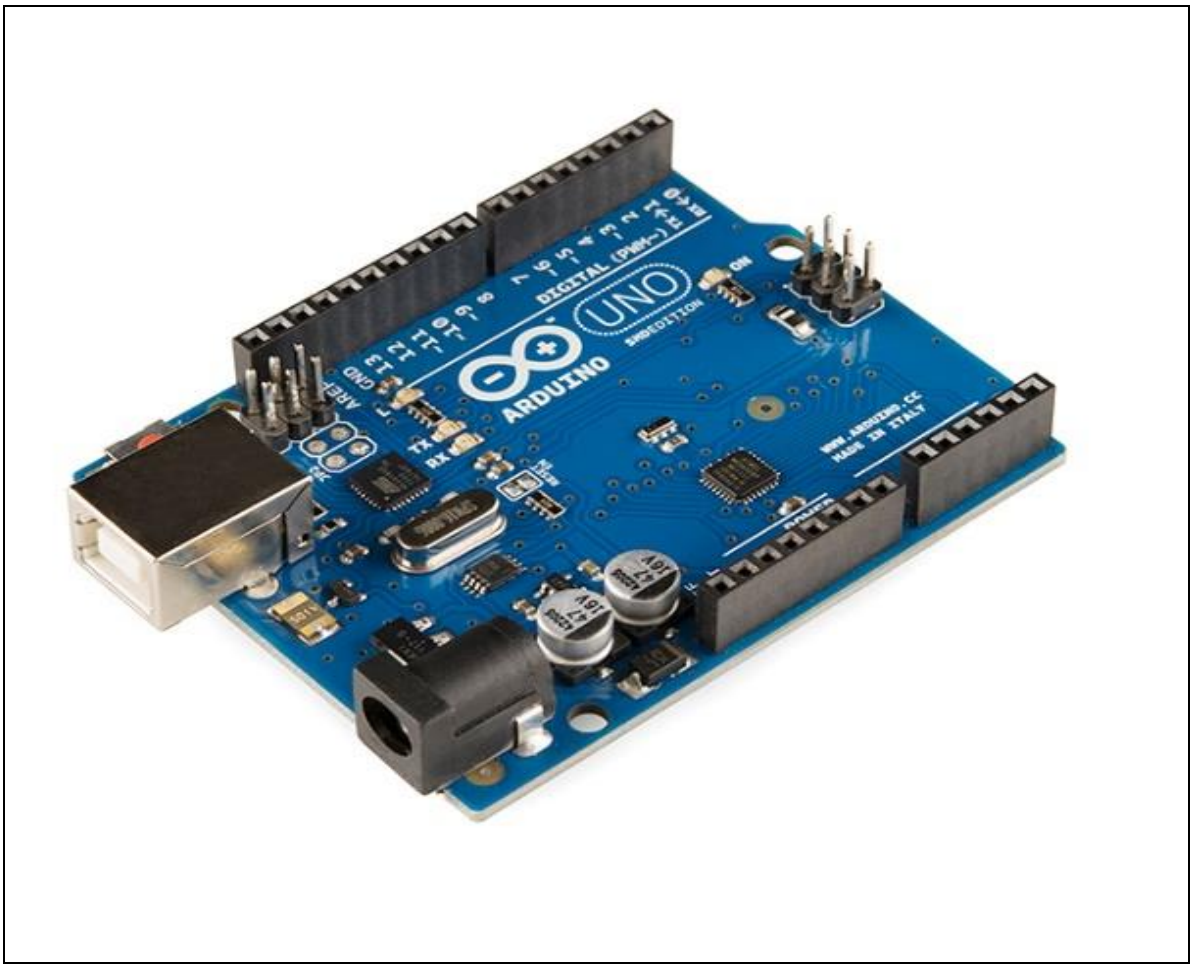


Table 7: Tools used for Implementation

Table 8: Ethernet Shield

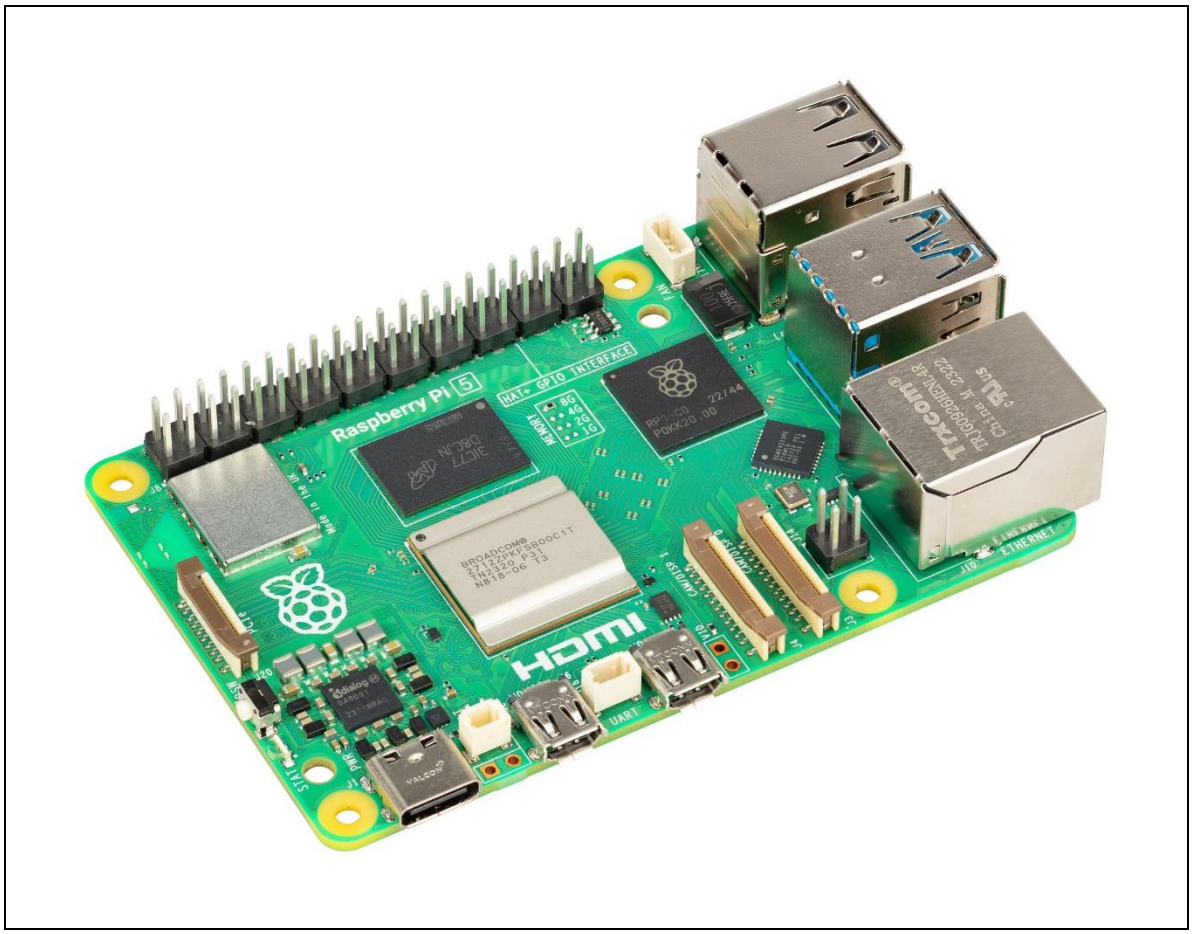


Table 9: Tools used for Implementation

↳ Code Execution

This is the actual code that we are using to test our system in the Arduino IDE. So let's proceed and watch what transpires.

Table 10: Arduino Code

↳ Firewall Simulation

We now choose the serial monitor option and launch our firewall. And it appears to be effective.

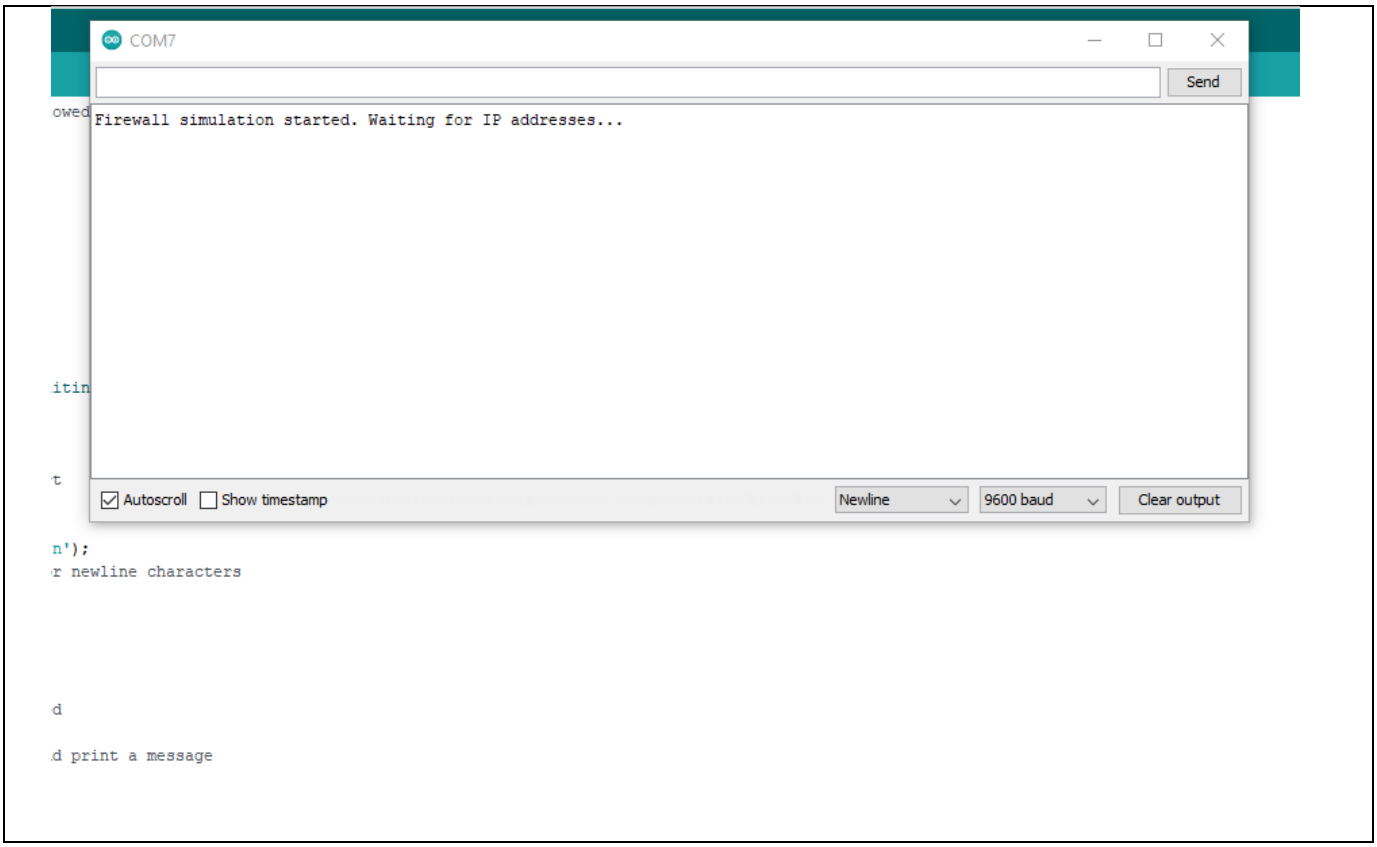


Table 11: Firewall Simulation

Filtering IP Address

It's time to block a particular IP address now. In order to do this, we attempt to reject and approve ICMP requests using random IPs. Please be aware that it will only allow or deny requests from IP addresses in accordance with the firewall's configured rules.

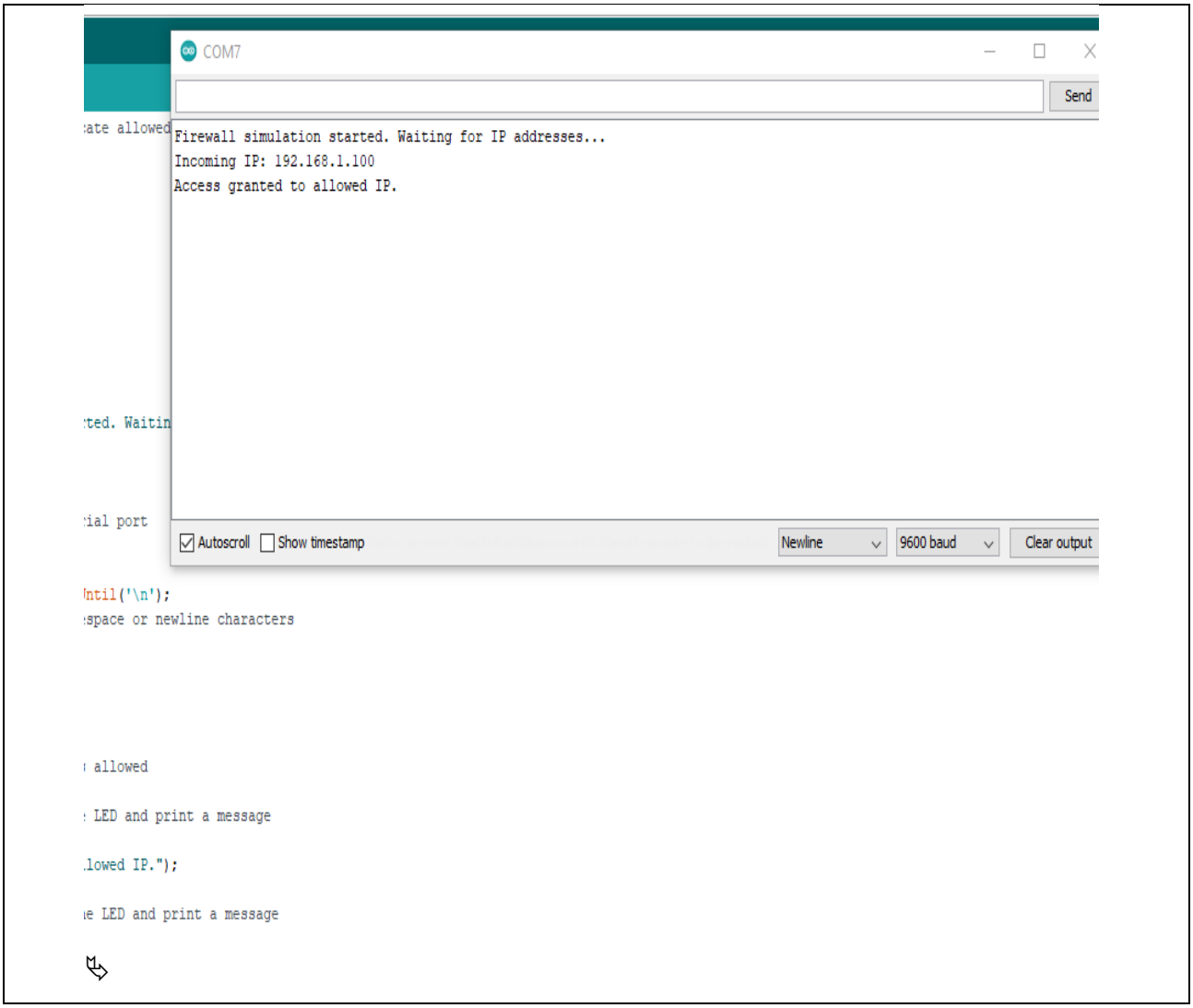


Table 12: Filtering IP

As you can see from the previous table, we are now attempting to block requests from the IP address 192.168.1.200, even though we are currently permitting traffic from that address.

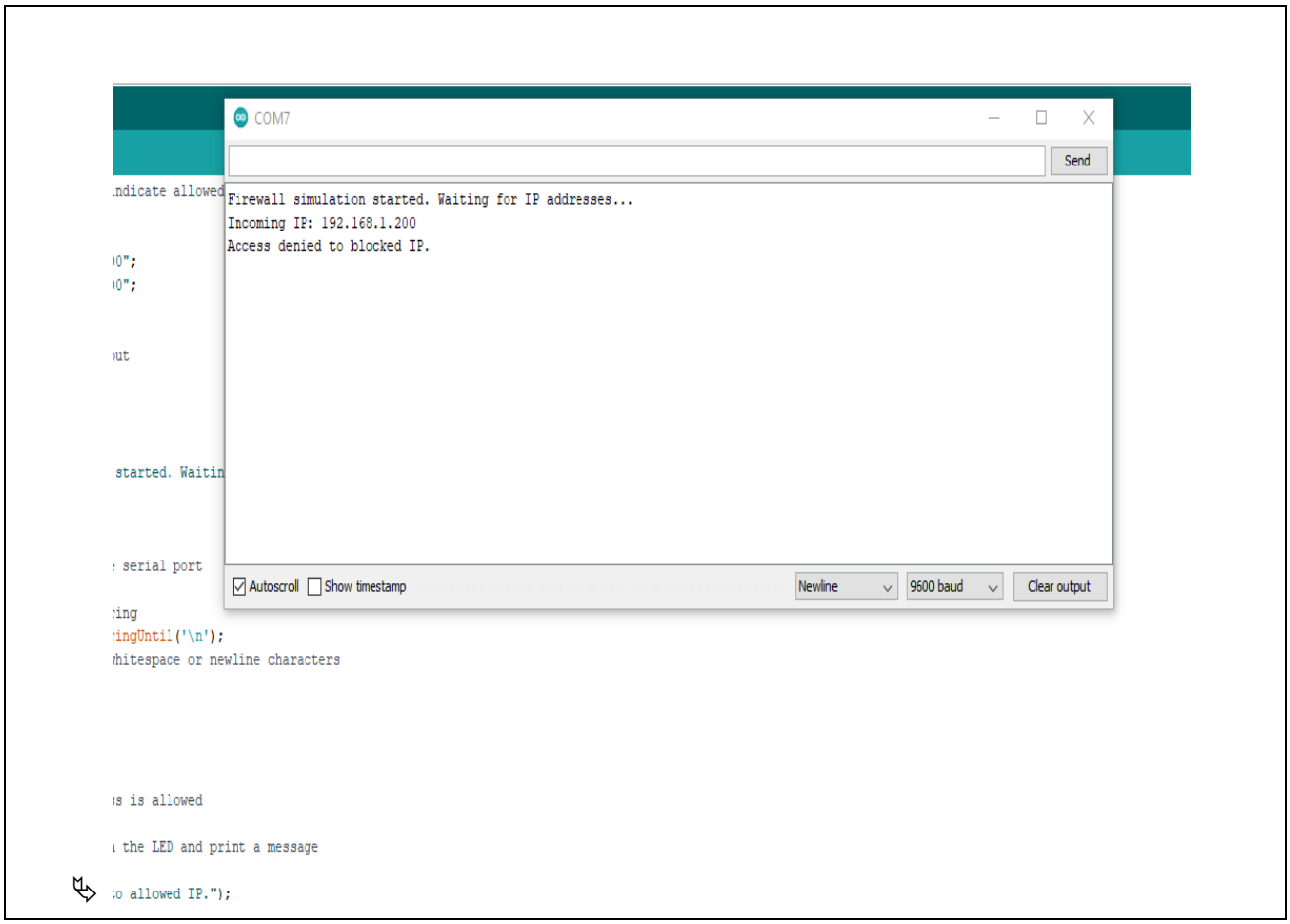


Table 13: Blocking IP

Software Based Firewall

We created both Software as well as Hardware Firewall to filter IP address which are coming from unknown address. And we can take action on them.

Thus, this is the Python utility that we have created. Additionally, it can presently filter any IP within a local network, among other features. It can also take requests in accordance with the guidelines we've established here. Let's proceed and see what happens when we try to execute this.

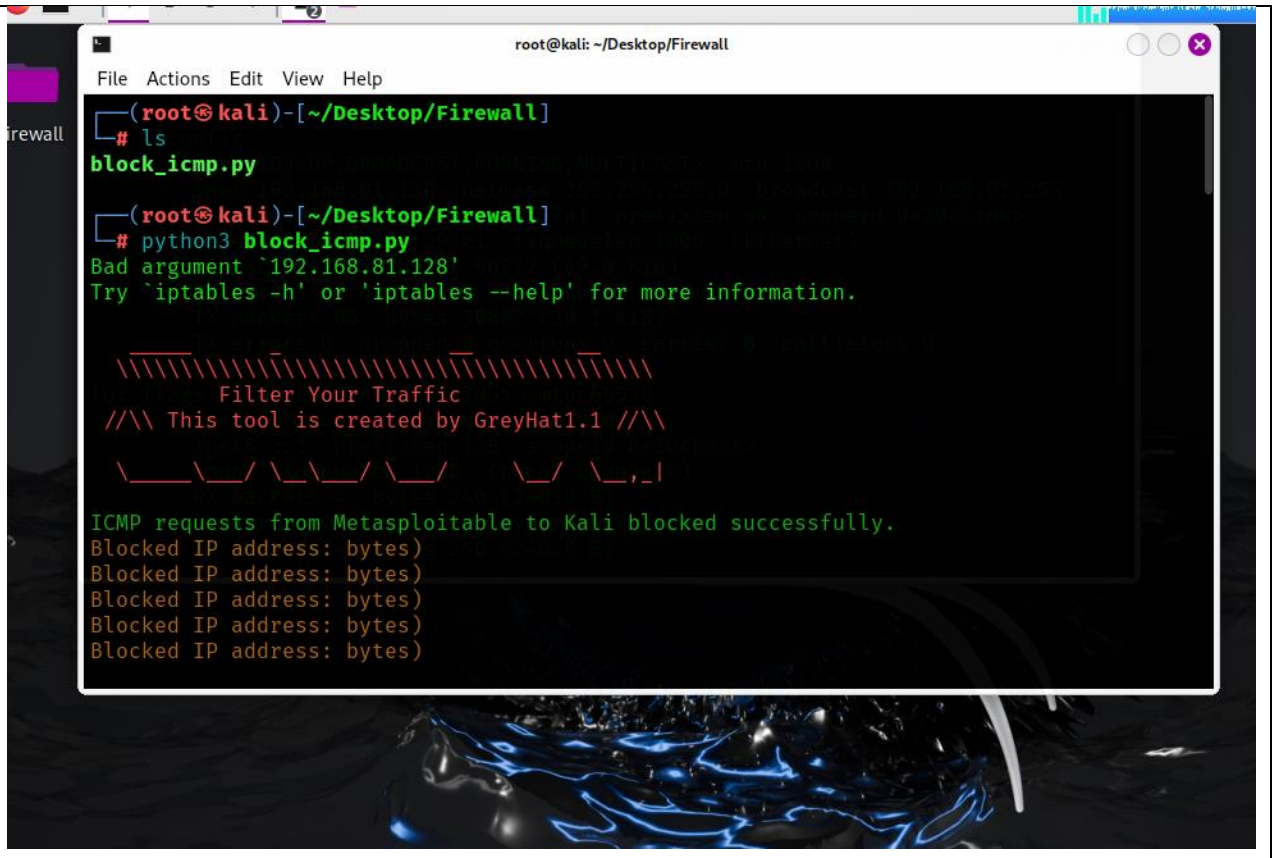


Table 14: Software Based Firewall

I'm using a Windows 7 computer and a Metasploitable machine here. We'll attempt to prevent metasploitable requests. This indicates that Kali Linux will suppress the metasploitable machine's icmp packet or request. However, it continues to interact with Windows 7. Let's give it a shot.

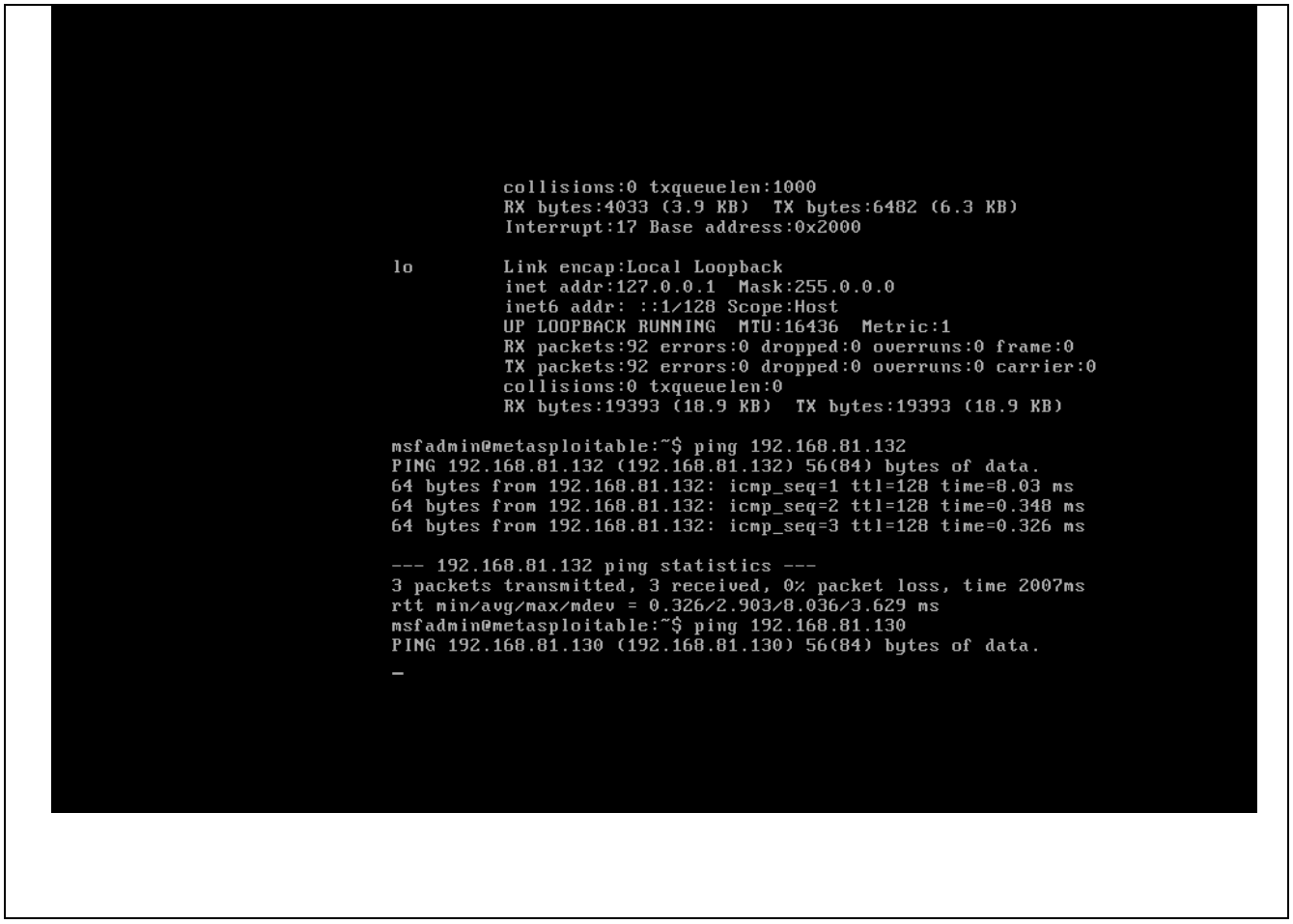


Table 15: Metasploitable Machine

As we can see, the Kali machine and our Windows machine are still in communication. As we only obstruct requests from machines that are exploitable. Thus, Windows and the Kali machine will continue to interact.

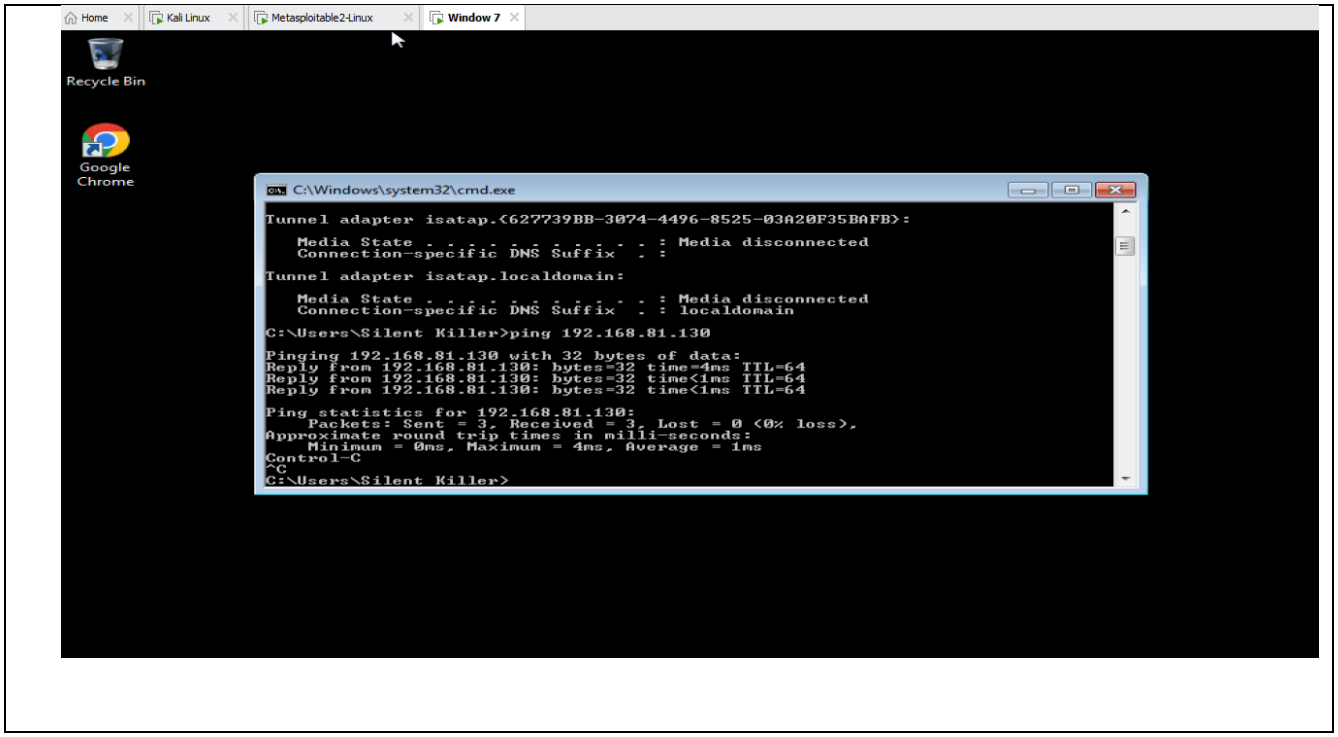


Table 16: Window 7

CHAPTER 6: TESTING

6.1 Introduction

Testing is the process of exercising software with the intent of finding and correcting errors. The objective of the testing is to uncover different classes of errors and to do so with a minimum amount of time and effort. In order to provide highly acceptable and error free system, the system should have to face the testing procedure and evaluation of each and every module and functionality.

6.2 Need of Testing

The significance of the testing phase is done to verify and validate the completed system. So that it will executed and presented the way it was expected and all the functionalities function as were defined in the project specification form. The testing does not require that the developer has to test each and every section of code to verify the bugs and recover from the errors and bugs that are present in it but its objective is to uncover the situations that could impact the system or the project or can lead in making the project unsuccessful. Thus, testing is done to enhance the quality of the system produced so that it is acceptable by the intended user with satisfaction.The highlights of this chapter include:

- ✓ The different types of testing done.
- ✓ The duration taken to test.
- ✓ The person(s) involved.
- ✓ The reported errors.
- ✓ The measures taken.

6.3 Test Plan

Software development processes typically focus on avoiding errors, detecting and correcting the software faults that do occur and predicting reliability after development. The test process, techniques and tools are significant contributors to effective and efficient testing and quality assurance. Various testing techniques to be implemented for proposed system are given in the next section.

6.3.1 Type of Testing

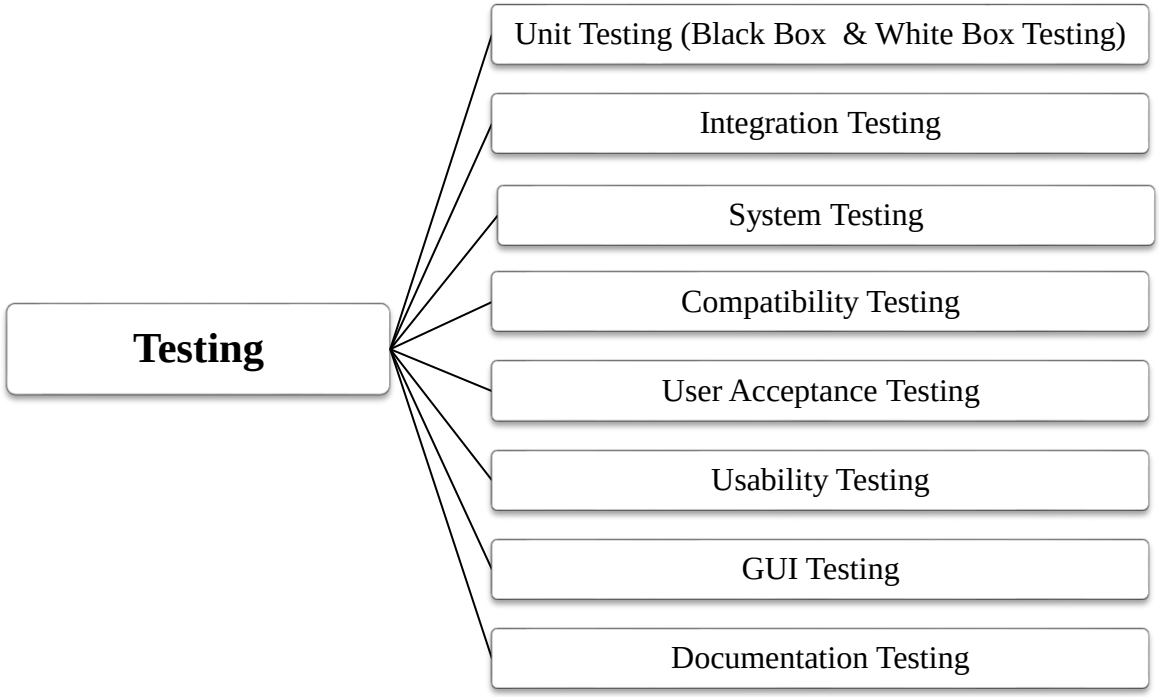


Table 17: Type of Testing

S. No	Testing Technique	Requirement
1.	**Unit Testing**	Testing of individual software component or modules of the system. This type of testing is performed by the developers only because it requires detailed knowledge of the internal program design and code. It includes performing black box and white box testing.
2.	**Integration Testing**	Testing of integration modules to verify combined functionality after integration. Various modules are integrated together and then tested by the developer of the system to check whether the modules work properly after integration.
3.	**System Testing**	Entire system is tested by the developer as per the requirements. All the modules of the system are combined and check whether they work perfectly after being integrated into one system as a whole.

4.	**Compatibility Testing**	Compatibility testing is used to determine if the system causes any issues related to how it function in concern with the operating system and different type of system hardware and software. The developer performs this type of testing on various hardware/ software platform to test the compatibility of the system on each platform.
5.	**GUI Testing**	The testing is to be done to verify the design of the system whether it satisfies the six predefined HCIU design principle and whether the system is user friendly.
6.	**Usability Testing**	The purpose of doing a usability test is to determine whether a product or document works with its intended users or readers. This is to be done by evaluating a product through observing the real people actually working on or using it so as to improve or enhance the system. The end- users execute the system to test it
7.	**User Acceptance Testing**	User Acceptance Testing tests that the system satisfies all the essential user requirements and is performing to the acceptable level as expected by the customer. This is done by defining a set of acceptance criteria which the system must satisfy before the customer will accept it.
8.	**Documentation Testing**	Documentation Testing is to be done to verify the system documentation whether it is meaningful and user friendly or not.

Table 18: Testing Description

6.3.2 Pass/Fail Criteria

- The produced system meets all the specifications mentioned in the project specification form and all the requirements of the user gathered during the research phase.
- All the modules/ functionalities are implemented in a way that the desired system produced runs without any bugs and quickly.

- System is to be verified for the usability that depends upon proven Human-Computer Interaction (HCI) principles and aims to achieve in the application ease of use, Navigation, time for completion and errors removal.

Point of Contact of Troubleshooting Purpose

Name	:	Vipin
Designation	:	Security Engineer
Contact No.	:	+91-8059149239
Email ID	:	vipinsukhraliya@gmail.com

Table 19: Point of Contact for Troubleshooting Purpose

6.4 Testing Duration

S. No.	Testing Technique	Start Date	End Date
1.	Unit Testing	1st March2024	19th March 2024
2.	Integration Testing	20thMarch 2024	1st April 2024
3.	System testing	2ndApril 2024	7thApril 2024
4.	Compatibility Testing	8thApril 2024	10thApril 2024
5.	GUI Testing	11thApril 2024	11thApril 2024
6.	Usability Testing	12thApril 2024	12thApril 2024
7.	User Acceptance Testing	13thApril 2024	17thApril 2024
8.	Documentation Testing	18th April2024	18th April2024

Table 20: Testing Duration

6.5 Users Involved in Testing

The developer will play the role of tester.

6.6 Unit Testing

In order to test each small part of the developed system individual test cases were developed. Unit testing was very helpful as it helped the developer to test individual units of source code. This is performed before any other testing because it is easy to find out bugs at lower level that is at module level than to find them at integration or system level. It is performed at two levels; black box test cases are made and then white box testing is done. Two levels of unit testing were

basically conducted, namely black box (also called functional testing) and white box (also called logical testing).

6.7Black box testing (Functional Testing)

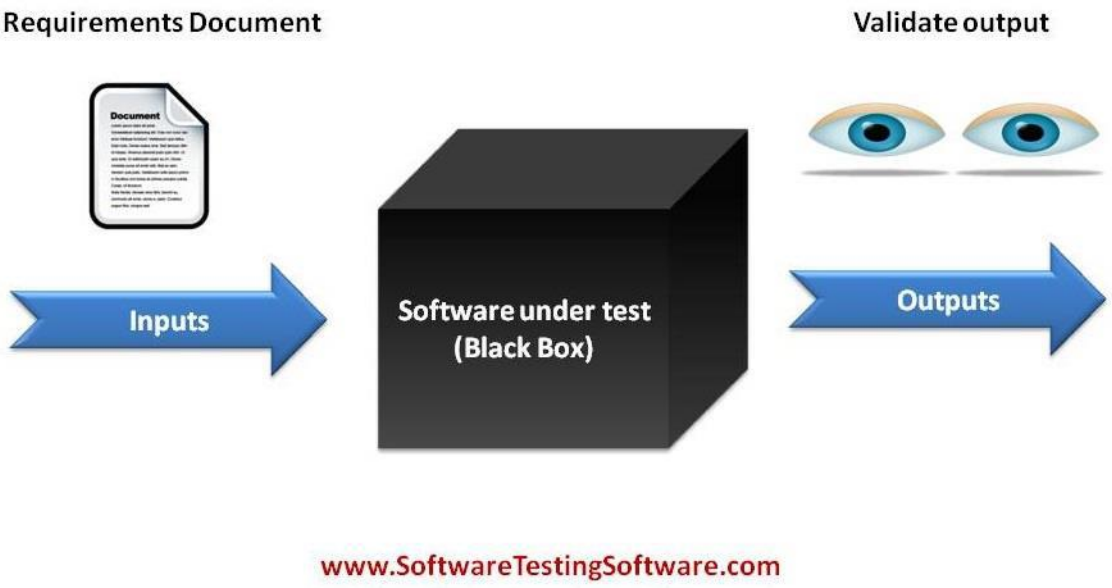


Table 21: Black Box

(www.softwareTestingSoftware.com 2017)

Black Box Testing, also known as Functional Testing, is a software testing technique that exposes only the functionalities and the design of the system; hiding the internal implementation/ structure of the system. The internal code or the structure is not known to the user who is going to use the system.

Conclusion: During testing, the firewall project effectively blocked illegal access while permitting legitimate traffic, demonstrating its efficacy in securing network communications. The firewall continuously satisfied security criteria by going through rigorous testing scenarios, which included simulated assaults and regular network operations, guaranteeing strong network protection. The outcomes validate the firewall's concept and implementation for practical use by demonstrating that it not only preserves network performance but also improves security. The importance of the firewall in defending digital infrastructure against changing cyberthreats is highlighted by this testing phase.

6.8 Integration Testing

In order to make sure the firewall system's different components function as a unit, integration testing for the project entails methodically assembling and testing each one. This stage is critical because it verifies how different modules—like the logging system, firewall rules engine, configuration user interface, and network monitoring tools—interact with one another. Finding any interface flaws or problems that may occur when these components interact is the main goal. The system is put through multiple phases of testing during integration testing:

Incremental Testing: Starting with the essential functions, components are integrated and tested in small groups. For example, the first priority might be making sure the firewall processes and applies rules to incoming and outgoing traffic accurately.

Both top-down and bottom-up approaches are possible for the testing strategy: top-down begins with high-level modules and gradually integrates lower-level components; bottom-up begins with low-level modules. This guarantees thorough coverage and facilitates more efficient defect isolation.

Interface Testing: Checking the interfaces between components is a crucial part of integration testing. This entails making certain that information transferred and processed across modules is valid, and that in the event of an integration failure, the logging and error-handling features work intended.

Scenario-Based Testing: To verify how the firewall reacts to different kinds of traffic and attack vectors, real-world situations are simulated. This entails verifying the firewall's ability to manage safe traffic coming from approved IP addresses, stop malicious traffic, record events, and notify administrators.

6.9 Techniques used for Integration Testing

The firewall project's integrated testing methodology combines bottom-up and top-down approaches with incremental and scenario-based testing techniques. Finding the firewall's primary features, like rule enforcement, traffic filtering, logging, and alerting systems, is the first step in the process. High-level components are first integrated and tested to make that the

firewall's overall structure functions as planned. This top-down methodology facilitates prompt validation of the primary control flows and important subsystem interconnections.
In addition, a bottom-up methodology is used, in which individual rule processing engines and network packet handlers—or other lower-level modules—are extensively tested separately prior to being incorporated into the main system. This two-pronged approach guarantees the strength of the fundamental components and the early identification of any problems.

This phase, which focuses on the data exchange between components, is critical for interface testing. Identification of mismatches or communication breakdowns that may result in functional disparities is aided by a thorough analysis of input-output interactions. When testing higher-level modules, mock objects and stubs can be used to mimic the behaviour of lower-level modules, providing a controlled testing environment that isolates particular functionality for accurate analysis.
Scenario-based testing is used to simulate real-world scenarios as integration advances. This entails emulating different traffic patterns and network conditions to assess how well the integrated system manages real-world operating circumstances. This allows the testing team to watch how the firewall behaves in a variety of scenarios, such as when it is subjected to genuine traffic from reliable sources or possible malicious entity attacks.

To evaluate the system's performance under pressure, integration testing also includes performance and load testing. In order to make sure the firewall can sustain performance and security during peak loads, this entails exposing it to large amounts of traffic. To make sure that new integrations don't interfere with already-existing functions, regression testing is constantly carried out.
By combining these methods, the firewall project's integration testing guarantees a thorough assessment of how well the various parts function as a unit, resulting in a reliable, safe, and effective firewall system that can safeguard the network against numerous threats while preserving peak performance. This extensive testing procedure is necessary to find and fix integration problems and guarantee the end product's dependability and efficacy.

6.10 Conclusion

The firewall project's integration testing and deployment have shown how reliable and efficient the system's design is in protecting network traffic. The intricate chore of monitoring, filtering, and controlling network traffic to protect against unwanted access and any dangers was carefully included into the firewall. We have made sure that every part of the firewall system works together flawlessly by going through a number of organized integration testing stages.

A hybrid strategy consisting of top-down and bottom-up techniques was used for integration testing. Early defect discovery and resolution at both high-level control flows and lower-level module interactions were made easier by this dual methodology.

In order to evaluate primary control structures, the integration process started with high-level components and gradually included lower-level modules. This ensured that every layer of functionality was carefully tested and verified to function in unison with the others. Determining the firewall's dependability and operating stability required extensive testing.

Verifying the accuracy and consistency of data transfers between components was made possible in large part by interface testing. Mock objects and stubs made it possible to create controlled testing environments, isolate particular functionalities, and guarantee thorough analysis free from outside influences. By identifying and resolving any communication issues, this thorough interface testing avoided functional inconsistencies.

CHAPTER 7: CONCLUSION

7.1 Success Criteria

A number of crucial factors that together guarantee the firewall project's effectiveness, dependability, and overall performance are at stake. The firewall's primary responsibility is to precisely enforce specified security policies by efficiently screening and controlling network traffic. This entails preventing threats and unauthorized access while enabling lawful traffic to flow through without hindrance. A firewall's performance mostly depends on its capacity to distinguish between these kinds of traffic and react accordingly.

The smooth integration of every system component is another important requirement. Every module—from the logging and alerting systems to the rule processing engine—must function flawlessly together through extensive integration testing. To preserve system functionality and integrity, any problems pertaining to interface interactions and data interchange must be found and fixed.

Another important indicator of success is performance under pressure. The firewall has to show that it can manage large amounts of network traffic without experiencing any performance deterioration. Only then can network efficiency be guaranteed to be maintained without any security measures being compromised. This entails keeping throughput high and latency low even during periods of high usage.

Furthermore, the success of the firewall depends on its robustness and dependability. It must to function flawlessly all the time, offering reliable defense against changing cyberthreats. Comprehensive logging and alerting systems are implemented to guarantee that security events are quickly identified and resolved.

7.2 Limitations and Errors in the Developed System:

The established firewall system has some limits and potential faults that should be noted, even with its strong design and extensive testing. A notable constraint is its reliance on pre-established guidelines and patterns to identify potential hazards. This strategy works well against known threats, but it might not work as well against sophisticated attacks that don't fit into current patterns or zero-day exploits. This restriction emphasizes the necessity of regular rule set updates

and enhancements, as well as the possible inclusion of machine learning methods for anomaly identification.

The potential for performance degradation under high network loads is another restriction. Even with performance and load testing, environments with very high traffic volumes may undermine the firewall's efficiency, resulting in higher latency and lower throughput.

The system also depends a great deal on precise administration and configuration. Errors or misconfigurations in the definition of rules may cause malicious traffic to pass through or accidentally block legal traffic. This danger emphasizes the necessity of thorough documentation, user education, and possibly the creation of more user-friendly setup tools in order to reduce human mistake.

Additionally, even though the firewall's logging and alerting methods are extensive, they may produce a large number of logs, which could make it difficult to distinguish important occurrences from regular traffic data. This may overburden administrators and cause a delay in responding to actual threats. This problem might be resolved by improving log management and putting in place more advanced alerting systems that prioritize and filter important notifications.

7.3 Future Enhancement

Future improvements can be made in a number of ways to overcome current shortcomings and further improve the firewall system:

Integration of AI and Machine Learning: By combining AI and Machine Learning, a firewall's capacity to identify and counteract novel and emerging threats can be greatly improved. Artificial intelligence (AI)-driven anomaly detection can spot odd trends in network traffic, which may reveal sophisticated attacks and zero-day exploits that conventional rule-based systems would have overlooked.

SSL/TLS Decryption and Deep Packet Inspection: By putting advanced deep packet inspection (DPI) techniques into practice, the firewall will be able to examine network packet content in greater detail. By including SSL/TLS decryption capabilities, encrypted traffic may be inspected and risks concealed in encrypted data streams can be identified and neutralized.

Automated Rule Management: By using automated systems to maintain and update firewall rules, one may lower the possibility of human error and guarantee that the policies are always current. The development, editing, and removal of rules can be streamlined with automation, which raises the firewall configuration's overall accuracy and efficiency.

7.4 Computational Challenges

To ensure the efficacy and efficiency of a firewall system, a number of computational problems in its creation and deployment must be overcome. These difficulties result from the requirement to strike a balance between resource usage, performance, and security in a dynamic network environment.

High Throughput and Low Latency: Processing massive amounts of network traffic while keeping high throughput and low latency is a major computational issue. Every packet that goes through the firewall must be inspected, and it must apply intricate rules and policies instantly. To accomplish this without causing notable delays, effective algorithms and optimization strategies needed.

CHAPTER 8: REFERENCES

[1] A. Bhandari, S. Gautam, T. K. Koirala, and M. R.Islam, "Packet Sniffing and Network TrafficAnalysis Using TCP—A New Approach," inAdvances in Electronics, Communication andComputing, ed: Springer, 2018, pp. 273-280.

[2] S. L. Rosa and E. A. Kadir, "Abnormal internetusage detection in LAN Islamic University of RiauIndonesia," in Proceedings of the InternationalConference on Intelligent Science and Technology,2018, pp. 17-22.

[3] G. Jain and Anubha, "Application of snort and wireshark innetwork traffic analysis," IOP Conference Series: MaterialsScience and Engineering, 012007, 2021, doi: 10.1088/1757-899X/1119/1/012007.

[4] C. Alisha. 2012. A summary of network traffic monitoring andanalysis techniques. [Online].

Available:http://www.cse.wustl.edu/~jain/cse567-06/ftp/net_monitoring.pdf

[5] Journal of Engineering Technology Vol. 9(1): 96-101,2021 ISSN 2231-8798© 2013 UniKL BM

[6] A. Bhandari, S. Gautam, T. K. Koirala, and M. R.Islam, "Packet Sniffing and Network Traffic Analysis Using TCP—A New Approach," Advances in Electronics, Communication and Computing, ed: Springer, 2018, pp. 273-280.

[7] S. Ansari, S. Rajeev, and H. Chandrashekar,"Packet sniffing: a brief introduction," IEEE potentials, vol. 21, pp. 17-19, 2002.

[8] International Journal of Innovations in Engineering and Technology (JET)http://dx.doi.org/10.21172/ ijiet.161.0

[9] Asrodia, Pallavi, and Hemlata Patel. "Network traffic analysis using a packet sniffer." International journal of engineering research and applications 2.3 (2012): 854-856.

[10] Nayak, Mr Parikshith, S. H. Brahmananda, and Mrs Sahana DS. "An Approach to Sniff Sensitive Information by Packet Sniffing."

[11] Asrodia, Pallavi, and Hemlata Patel. "Network traffic analysis using a packet sniffer." International journal of engineering research and applications 2.3 (2012): 854-856.

[12] Nayak, Mr Parikshith, S. H. Brahmananda, and Mrs Sahana DS. "An Approach to Sniff Sensitive Information by Packet Sniffing."

[13] Ogbu, Henry N., and Moses Adah Agana. "Intranet Security using a LAN Packet Sniffer to Monitor Traffic." arXiv preprint arXiv:1910.10827 (2019).

[14] Wireshark:https://www.wireshark.org/docs/wsug_html_chunked/ChWorkDisplayFilterSection.html

[15] International Journal of Current Trends in Engineering & Research (IJCTER) e-ISSN 2455–1392 Volume 4 Issue 4, April 2018