

DEPARTMENT OF CYBER SECURITY

PROJECT REPORT



SUBMITTED BY:

ABDUL RAFAY (233011)

SHAHZAIB ALI (232101)

ALI NASIR (232931)

SUBJECT: OOP

COURSE CODE: CS112L

SECTION: BSCYSev-2-A

SUBMITTED TO: DR MUHAMMAD IMRAN

SUBMITTED ON: 19TH MAY, 2024.

Table of Contents

1. Introduction

- 1.1. Introduction To Firewall Tutor**
- 1.2. Importance of Firewall**
- 1.3. Need for GUI based Firewall**

2. Design and Class Hierarchy

- 2.1. UML Class Diagram**
- 2.2. Description of Classes**

3. Program Flow

- 3.1. Flow Chart**
- 3.2. Processing of user inputs, rule storing and action**

4. User Manual

- 4.1. Explanation of the Forms Added**
- 4.2. Use of Application**

5. Settings and Resetting the GUI based app

- 5.1. Explanation of Settings**
- 5.2. Rules and Results Resetting**

6. Implementation Details

- 6.1. Task Division among the Group members**
- 6.2. Implementation Procedure**
- 6.3. Challenges faced in Implementation**

7. Analysis and Lesson Learned

- 7.1. Project Strength and Weaknesses**
- 7.2. Suggestions for Future improvement**

8. Conclusion

- 8.1. Potential Impact of the Project**
- 8.2. Summary of Project Objectives**

1. Introduction

1.1. Introduction to Firewall Tutor

The GUI-based Firewall Tutor is a Windows Forms application developed in C++/C# to teach firewall functionality to students. The application enables users to enter a set of firewall rules and data pertaining to network packets, and then check which rule, if any, applies to the given packet.

The Firewall Tutor is designed to be an educational tool that helps students learn the fundamentals of firewall functionality. By providing a visual interface for entering and testing firewall rules, students can gain a deeper understanding of how firewalls work and how to configure them for different network scenarios.

1.2. Importance of Firewall

Firewalls are a critical component of network security, providing a first line of defense against unauthorized access and malicious traffic. A firewall is a network security device that monitors incoming and outgoing traffic and based on a defined set of security rules accepts, rejects, or drops that specific traffic.

Firewalls are essential for securing a network from unauthorized access, preventing malware and other threats, controlling network access, monitoring network activity, and complying with industry regulations. They provide a critical layer of defense against cyber attacks, protecting sensitive data and systems from unauthorized access and theft.

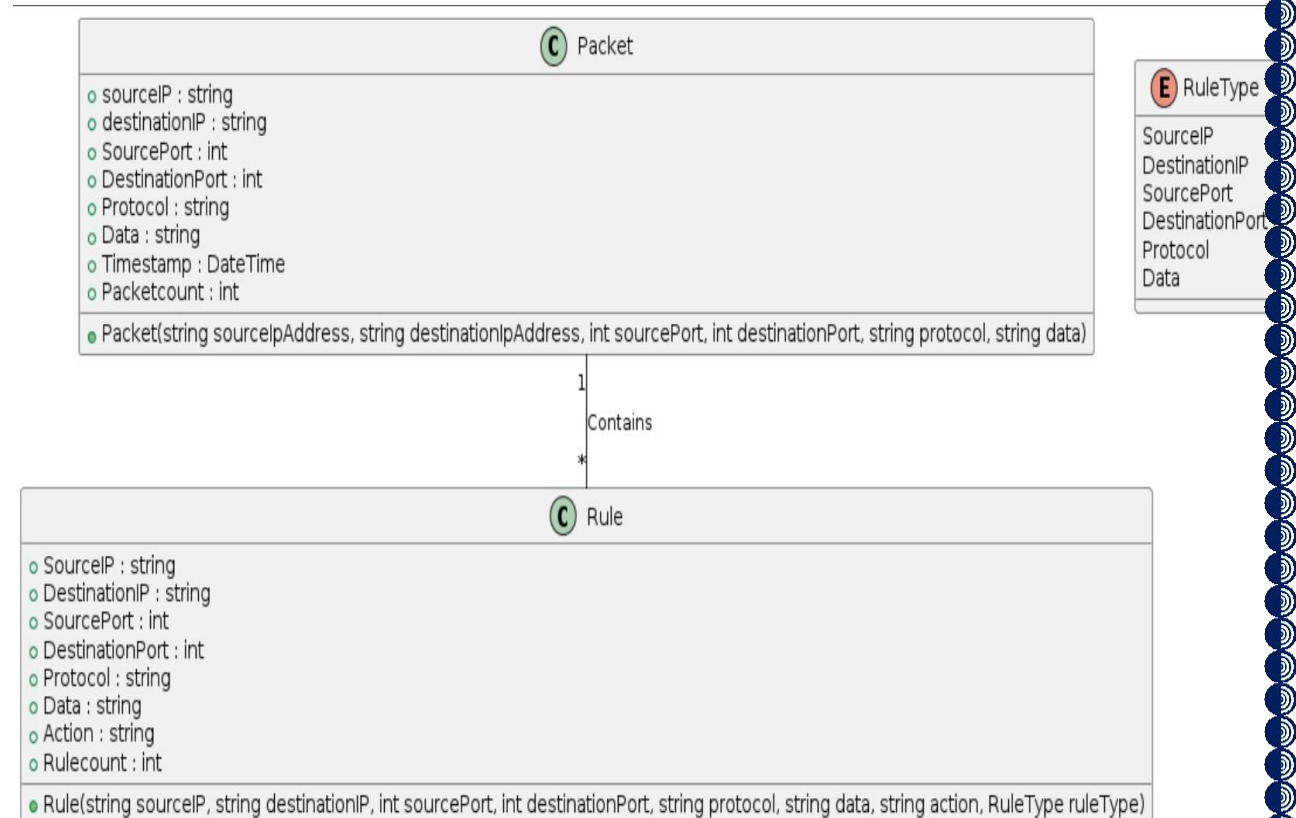
1.3. Need for GUI based Firewall

While command-line interfaces (CLI) have their advantages, GUI-based firewalls offer a more user-friendly and accessible way to manage and configure firewall settings. A GUI-based firewall provides a visual interface for managing firewall rules and policies, making it easier for users to understand and configure firewall settings without requiring extensive knowledge of command-line interfaces or network security protocols.

2. Design and Class Hierarchy

2.1. UML Class Diagram

The class diagram for the GUI-based Firewall Tutor application would include classes such as Firewall, Rule, Packet, and FirewallTutor, and their relationships. This would help in understanding the system's architecture, promoting reusability and maintainability, and facilitating communication between the development team and stakeholders.



2.2. Description of Classes

Rule Class:

The Rule class contains properties for source IP, destination IP, source port, destination port, protocol, data, and action. It also contains a static property Rulecount to keep track of the number of rules created. The constructor of the Rule class initializes these properties and increments the Rulecount.

The RuleType enumeration is used to specify the type of rule.

Packet Class:

The Packet class contains properties for source IP, destination IP, source port, destination port, protocol, data, and timestamp. It also contains a static property PacketCount to keep track of the number of packets created. The constructor of the Packet class initializes these properties and increments the PacketCount.

Reason of Making Classes Public:

These classes and enumeration are made public to allow other parts of the application to create and manipulate instances of these classes. The public members of the Rule and Packet class are the properties that are used to define the rules and packets.

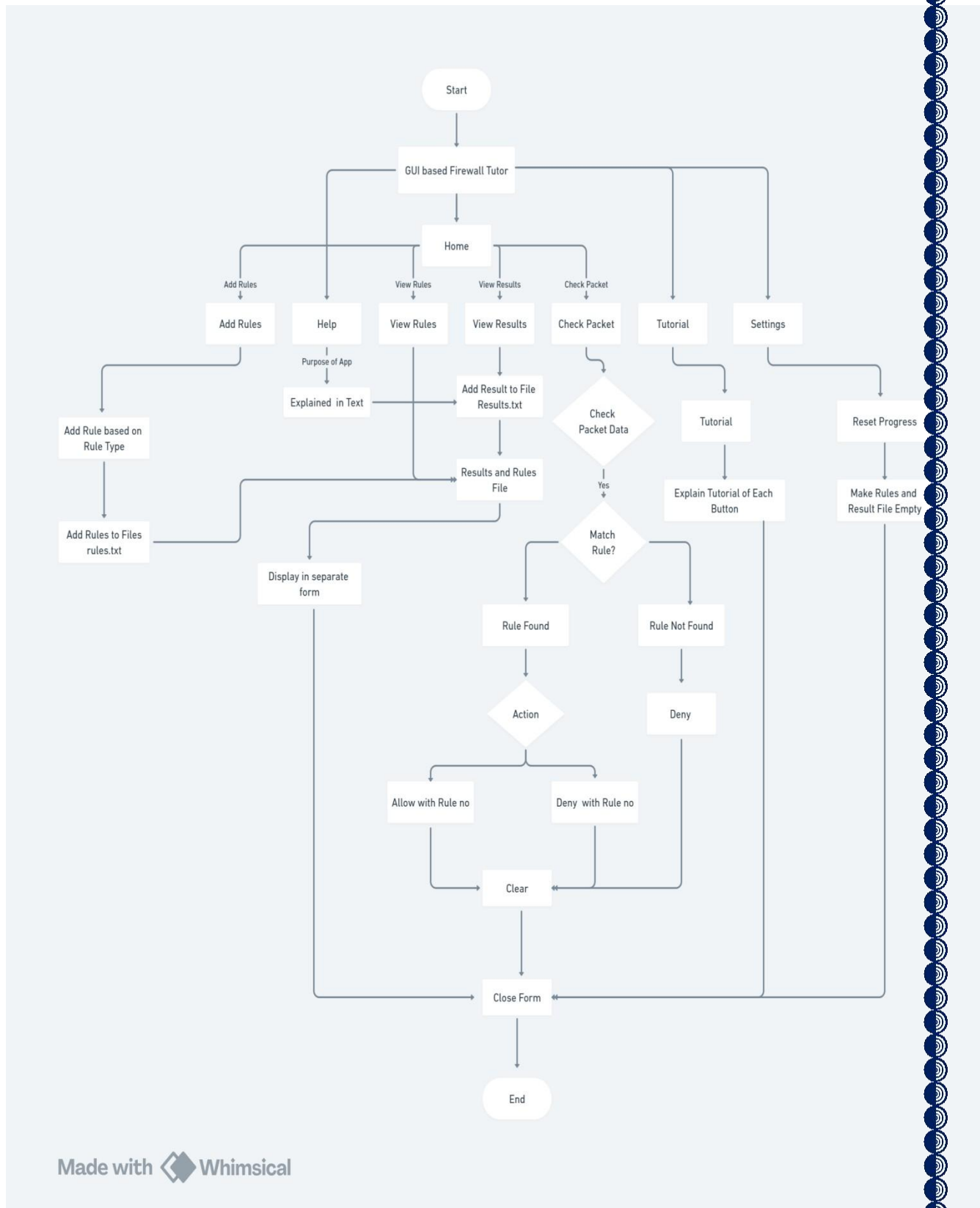
The RuleType enumeration is also made public to allow other parts of the application to specify the type of rule. The enumeration values are used in the Rule class to define the type of rule.

Overall, these classes and enumeration are essential for defining the rules and packets for a firewall system. The public members allow other parts of the application to create and manipulate instances of these classes and specify the type of rule.

3. Program Flow

3.1. Flowchart

A flowchart helps in planning and designing the application's workflow, making it easier to visualize the different steps involved in the application. It helps in identifying the different components of the application and how they interact with each other.



:: Zoom in using Word app for best view

3.2. Processing of user inputs, rule storing and action

The firewall tutor app you described is a tool that allows users to create and manage firewall rules. Here's a breakdown of how it processes user inputs, stores rules, and takes actions based on those rules:

User Input: The user selects a rule type from a dropdown menu (combo box). This rule type determines the type of firewall rule that will be created (e.g., allow or deny traffic based on IP address, port number, protocol, etc.). The user then enters the specific details of the rule in a textbox, such as the source and destination IP addresses, port numbers, and protocol.

Rule Fetching: Once the user has entered the rule details, the app fetches the rule from the textbox and adds it to the packet data. This packet data is used to determine whether incoming network traffic should be allowed or denied based on the rules that have been created.

Rule Storage: The app stores the rule in a file called "rules.txt" along with a rule count for each rule. This file serves as a database of firewall rules that can be easily accessed and modified by the app.

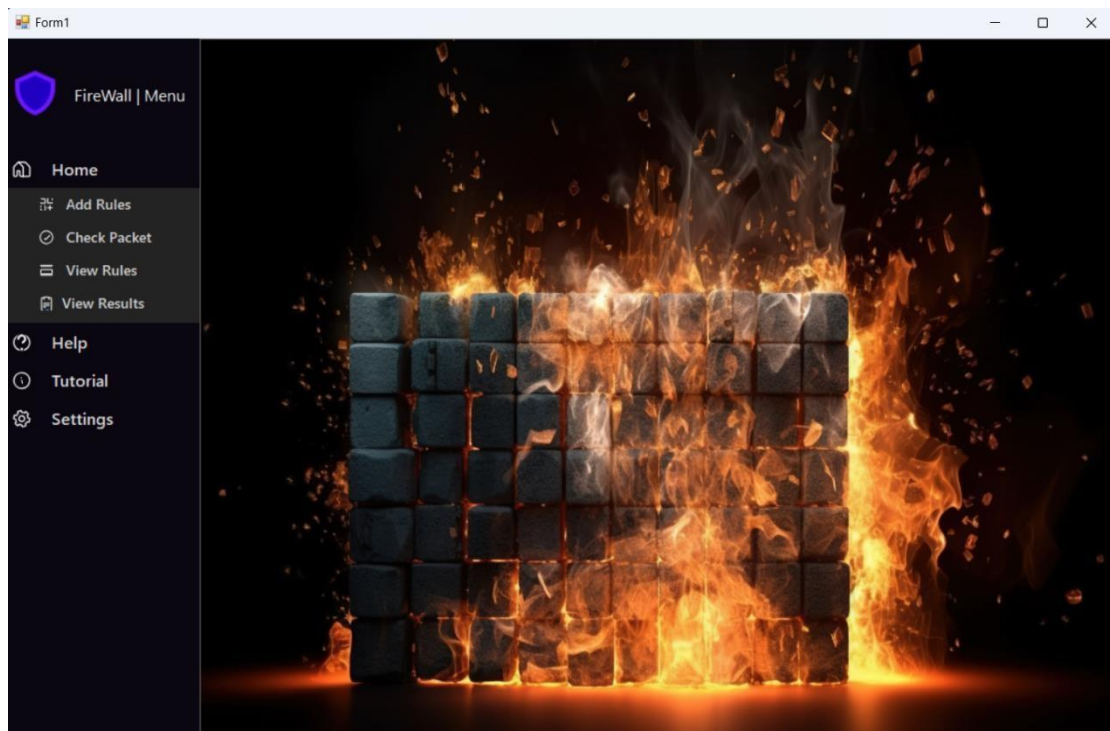
Rule Parsing: When the app needs to check whether incoming network traffic should be allowed or denied, it parses the rules from the "rules.txt" file and stores them in a list. This list is used to compare the packet data against the rules and determine whether the traffic should be allowed or denied.

Action Taking: Based on the comparison between the packet data and the rules in the list, the app takes a specific action. If the traffic matches a rule that allows it, the app permits the traffic to pass through the firewall. If the traffic matches a rule that denies it, the app blocks the traffic from passing through the firewall. If there is no rule that is present in the file then the packet is denied by default and message says "No rule found".

4. User Manual

4.1. Explanation of the Forms Added

Home page Form:



This is the main form for the application. It has animated icons, dropdown menu and a attractive background image. The child forms or other forms will open under the image panel to make sure application looks like a real GUI based app.

It uses a method called `openChildForm` which takes an variable of type `Form` and then checks if the active `Form` is current `Form`. If it isn't the current `Form` it opens new `Child Form`.

It uses another method called `Hidesubmenu` to make sure `subMenu` is hided before user clicks on `Home` button. The theme of the app is tried to be made more attractive for the users so they can have smooth experience and overall tone of the GUI based app.

Add Rule Form:

The screenshot shows a web application window titled 'Form1' with a sidebar menu containing 'Home', 'Help', 'Tutorial', and 'Settings'. The main content area is titled 'Enter FireWall Rules'. It contains several input fields with red asterisks indicating they are required: 'Enter Rule Type:', 'Enter Source IpAddress(Single/Range):', 'Enter Destination IpAddress(Single/Range):', 'Enter Source Port:', 'Enter Destination Port:', 'Enter Protocol:', 'Enter Data:', and 'Action:'. A note at the bottom states 'Note: IP range format: 192.168.0.1-192.168.0.10'. At the bottom right, there are three buttons: 'Close', 'Save', and 'Clear'.

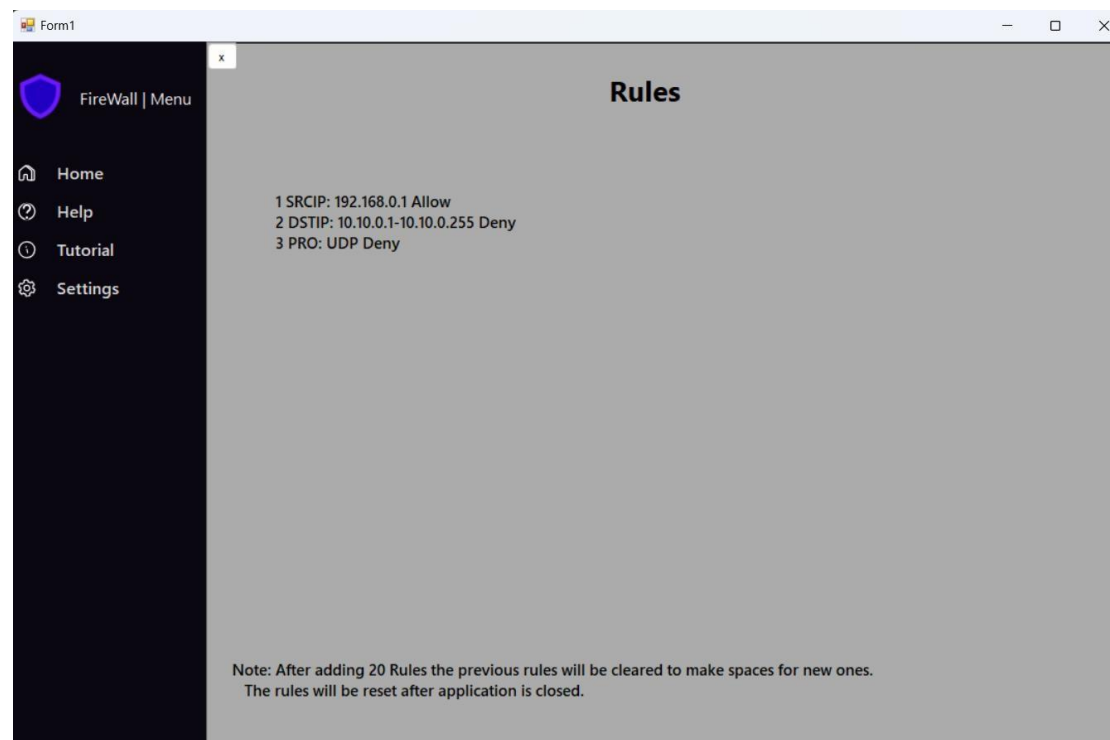
This is the Form where user will enter Rule based on ruleType. The rules will be saved to file rules.txt. The rules file will be made empty each time the user closes application to maintain rulecount. There is a check to make sure user doesn't leave any field empty.

Check Packet Form:

The screenshot shows a web application window titled 'Form1' with a sidebar menu containing 'Home', 'Help', 'Tutorial', and 'Settings'. The main content area is titled 'Check Packet'. It contains several input fields with red asterisks indicating they are required: 'Source IpAddress:', 'Destination IpAddress:', 'Source Port:', 'Destination Port:', 'Protocol:', and 'Packet Data:'. Below the input fields, there is a line of text showing packet details: '2024-05-19 01:08:20 SRCIP: 192.1688.0.1 DSTIP: 10.10.0.1 SRCPORT: 80 DSTPORT: 8080 PRO: UDP DATA: ABCDEFG Allowed Rule no 1'. A note at the bottom states 'Note: Even if no Rule is found the Packet will be denied by default'. At the bottom right, there are three buttons: 'Close', 'Check Packet', and 'Clear'.

This form enables the user to enter all the data about a Packet and takes action. If the rule matches it takes action and mentions the rule number which took action along with the packet data. It then stores this result and further result in Results.txt file which will be used for viewing the rules. If there are no rule present for packet then by default it is denied.

View Rules Form:



Here user can view all the rules added. The rules are fetched from the file and then based on packet count the label is made visible showing rules else labels says “No rule Found”.

After 20 rules added the rules are again set to zero to make sure the space is again free and user can view all the rules. Because if too many rules are added user will not be able to see them properly.

View Results Form:

The screenshot shows a web application window titled "Form1". On the left is a dark sidebar with a "FireWall | Menu" header and four items: Home, Help, Tutorial, and Settings. The main content area is titled "Results" and displays a table of firewall log entries. The table has columns for timestamp, source IP, destination IP, source port, destination port, protocol, data, action, and rule. Three entries are shown. A note at the bottom states: "Note: After 15 Packets Results the previous results will be cleared to make spaces for new ones. The results will be rest after the application is closed."

Timestamp	SRCIP	DSTIP	SRCPORT	DSTPORT	PRO	DATA	Action	Rule
2024-05-19 01:33:06	192.168.0.1	10.10.0.1	80	8080	UDP	ABCDEDCDCX	Allowed	Rule no 1
2024-05-19 01:33:25	10.10.0.1	2.3.6.5	443	84	TCP	KSSNSDKDSS	Denied	No rule found
2024-05-19 01:33:44	10.10.0.1	10.10.0.6	80	8080	TCP	ISKDVDSN	Denied	Rule no 2

Note: After 15 Packets Results the previous results will be cleared to make spaces for new ones.
The results will be rest after the application is closed.

This form allows the user to see all the Results on the packet. The results are displayed with a timestamp to keep a record of them. After 10 packets being checked it will be set to zero to make spaces for new ones. The results are fetched from Results.txt file.

Help Form:

The screenshot shows a web application window titled "Form1". On the left is a dark sidebar with a "FireWall | Menu" header and four items: Home, Help, Tutorial, and Settings. The main content area is titled "FireWall Tutor" and contains a welcome message, a "Getting Started" section, a "What You Can Do Here" section, and a "Contact Us" section.

FireWall Tutor

Welcome to the Firewall Tutor Help Section.
Thank you for using our GUI-based Firewall Tutor! We're happy to help you learn and understand the concepts of firewall configuration.

Getting Started

Our Firewall Tutor is designed to simulate a real-world firewall environment, allowing you to practice and learn in a safe and controlled space. The modern and intuitive interface makes it easy to navigate and understand the different features and settings.

What You Can Do Here

Learn about firewall rules and how to configure them.
Practice creating and managing firewall rules.
Understand how to block or allow specific traffic.
Get familiar with common firewall scenarios and how to handle them.

Contact Us

If you have any questions or feedback, please contact our team.
Thank you for using our Firewall Tutor! We hope you find it helpful in your learning journey.

This form explains the purpose of the app, what user can expect on the app.

Tutorial Form:

Form1

FireWall | Menu

Home

Help

Tutorial

Settings

FireWall Tutorial

Please select a button below to see its tutorial:

Home Help Settings

This button opens subMenu. Select the button:

*** Add Rules Button ***

Add Rules

This section allows you to define custom firewall rules to manage incoming and outgoing network traffic.

Check Packet

Choosing a Rule Type:

1. Select the desired rule type from the dropdown menu.
 - Available rule types include source IP address, destination IP address, source port, destination port, protocol, or data content.
2. Define the specific criteria for your chosen rule type in the corresponding fields.
 - At least one criterion must be specified.

View Rules

View Results

Creating a Rule:

1. **Rule Criteria:** After selecting the rule type, fill in the relevant details like source/destination IP addresses, ports, protocol, or data string (depending on the chosen rule type).

Remember:

- Only one rule type can be selected at a time.
- You must define at least one criterion for the chosen rule type.

This button opens a new form which explains the tutorial of each button in the Firewall. It is extremely necessary for those who don't know much about Firewall.

Settings Form:

Form1

FireWall | Menu

Home

Help

Tutorial

Settings

FireWall Settings

Welcome to FireWall Settings!

We hope that you enjoying the experience of the FireWall.

Need a new Start. Reset your progress.

Reset Progress

We hope that you liked the overall experience of the GUI based app. We continue to customize the app according to the user needs. If you report an issue while using the app contact our team.

Please leave a suggestion for our app:

Save Suggestion

This section allows user to reset all the rules added and results. On clicking this button rules and results file will be made empty.

4.2. Use of Application

"If you're new to the Firewall Tutor application or need a refresher on how to use it, don't worry! Our tutorial form is here to help. Simply click on the 'Tutorial' button on the main screen, and you'll be taken through a step-by-step guide on how to enter firewall rules, create packets, and check if they're allowed or denied. Additionally, if you have any questions or need further assistance, our Help form is just a click away. It provides detailed instructions and answers to common questions, ensuring you get the most out of our application."

5. Settings and Resetting the GUI based app

5.1. Explanation of Settings

"The Settings feature in our Firewall Tutor application allows you to reset all the rules and results added to the GUI. This is useful if you want to start fresh or if you've reached the maximum limit of 20 rules and 15 packet results. The Settings also includes a suggestion feature that takes into account the user's experience with the app. After the form is closed, the rules and results are automatically reset, ensuring a clean slate for your next session. This feature helps maintain the accuracy and efficiency of the application, making it easier for you to learn and understand firewall functionality."

5.2. Rules and Results Resetting

"To reset the rules and results in our Firewall Tutor application, simply click on the 'Reset' button in the Settings feature. This action empties the file where the rules and results are stored, effectively removing all the data. It's important to note that this action cannot be undone, so make sure to save any important data before resetting. Resetting the rules and results is useful if you want to start fresh."

6. Implementation Details

6.1. Task Division among the Group members

The task was divided among the group members to ensure efficient and timely completion of the project. Abdul Rafay was responsible for designing the home form, rules, and check packet form. Ali Nasir designed the settings tutorial and other forms. Shahzaib wrote the complete report, flowchart, class diagram, and developed the help button, view rules, and view results forms.

6.2. Implementation Procedure

The implementation procedure involved using file handling and lists to compare and store data. The team worked together to ensure that each component of the project was integrated seamlessly. The implementation process was iterative, with each member building upon the work of the others to create a cohesive and functional system.

6.3. Challenges faced in Implementation

One of the significant challenges faced during implementation was that, the previous rule count could not be maintained and would start from zero if the app was closed so if there are already preset rules the count would again start from one. To overcome this challenge, the team found that the only solution was to make the file empty when the app is closed. This ensured that the rule count was reset correctly, and the system functioned as intended.

7. Analysis and Lesson Learned

7.1. Project Strength and Weaknesses

The Firewall Tutor project had several strengths, including its user-friendly interface, clear instructions, and interactive tutorials. The project's design was intuitive, making it easy for users to navigate and understand the concepts being taught. However, there were also some weaknesses, such as the lack of customization options

and the limited range of topics covered. Additionally, the project could benefit from more comprehensive feedback mechanisms to help users identify areas where they need improvement.

7.2. Suggestions for Future improvement

To improve the Firewall Tutor project, the team could consider adding more customization options to allow users to tailor the tutorials to their specific needs. Additionally, expanding the range of topics covered would make the project more versatile and valuable to a wider audience. Finally, incorporating more sophisticated feedback mechanisms, such as real-time analytics and personalized recommendations, could help users identify their strengths and weaknesses and improve their learning outcomes. Overall, the Firewall Tutor project has the potential to be an effective educational tool, and with some strategic improvements, it could become even more valuable to users.

8. Conclusion

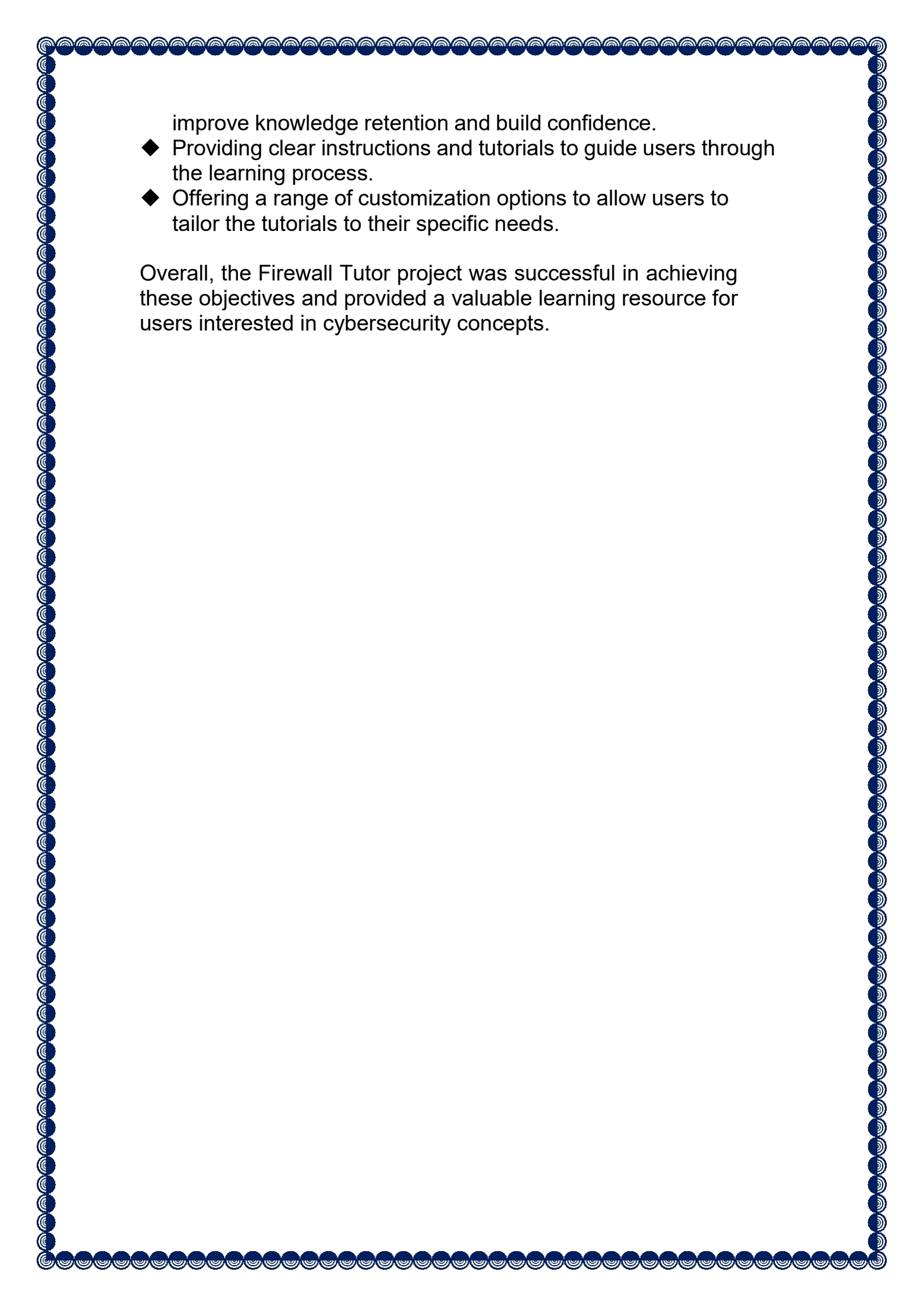
8.1. Potential Impact of the Project

The Firewall Tutor project has the potential to make a significant impact on the way that users learn about cybersecurity concepts. By providing an interactive and engaging platform for learning, the project can help users develop a deeper understanding of firewall fundamentals and improve their ability to apply these concepts in real-world scenarios. Additionally, the project's focus on practical application and hands-on learning can help users build confidence in their abilities and develop a more positive attitude towards cybersecurity.

8.2. Summary of Project Objectives

The Firewall Tutor project was designed to achieve several key objectives, including:

- ◆ Providing an interactive and engaging platform for learning about firewall fundamentals.
- ◆ Developing a user-friendly interface that is easy to navigate and understand.
- ◆ Incorporating practical application and hands-on learning to

- 
- improve knowledge retention and build confidence.
- ◆ Providing clear instructions and tutorials to guide users through the learning process.
 - ◆ Offering a range of customization options to allow users to tailor the tutorials to their specific needs.

Overall, the Firewall Tutor project was successful in achieving these objectives and provided a valuable learning resource for users interested in cybersecurity concepts.