Firewall Simulator

Session: 2022-2026

Submitted by:

Saba Shahdin	2022-CS-112
Afeera Fatima	2022-CS-151

Submitted to:

Syed Tehseen Ul Hasan Shah

Course:

CS-373L Computer Networks C

Department of Computer Science

University Of Engineering And Technology,

Lahore, Pakistan

Contents

List of Figures

1 Introduction

The **Firewall Simulator** is a comprehensive solution for monitoring and managing web browsing activities, ensuring safe and controlled browsing. Its primary objective is to enhance online safety by restricting access to unauthorized or damaging websites while simultaneously revealing and logging blocked attempts.

1.1 Problem Statement

The unrestricted access to the internet increases vulnerability to malware infections, phishing sites, and exposure to inappropriate or abusive content. Traditional firewalls often lack user-friendliness and real-time monitoring functionality. Additionally, the absence of robust tools for trend analysis and targeted restrictions heightens these vulnerabilities. This project seeks to overcome these limitations with a user-centric solution.

1.2 Purpose and Significance

In an era where internet usage has become necessary across all demographics, challenges concerning online safety have increased. Organizations, parents, and individual users struggle to protect their browsing activities from harmful influences. This project addresses these concerns by offering a personalized browser integrated with an in-built firewall mechanism. Key features include content filtering, domain restrictions, and logging unauthorized access attempts. Moreover, the addition of statistical analysis and visualization empowers users with actionable insights into browsing patterns and potential threats.

1.3 Motivation

The inspiration for this project stems from the urgent need for a simple yet powerful tool to ensure the security and administration of browsing activities. With cyber threats becoming increasingly harmful and users easily distracted by the digital world, this project combines security, usability, and actionable insights. It bridges the gap between technical firewalls and user-friendly browsers, serving diverse audiences such as parents safeguarding their children and organizations enforcing safe browsing policies. This project's integrated analytics provide a foundation for informed decision-making, representing a significant contribution to internet safety and management.

2 Project Description

2.1 Background

This project addresses the growing need for tools that ensure safe and controlled web access. With the increasing prevalence of cyber threats, malicious websites, and distractions, a strong yet accessible solution is essential. Unlike traditional firewalls that often require technical expertise for configuration, this project offers a simpler, user-friendly approach.

By utilizing `PyQt5` for a highly interactive and responsive user interface and `matplotlib` for effective data visualization, the project focuses on core functionalities such as:

- Blocking harmful websites.
- Monitoring access attempts.
- Providing analytics for informed decision-making.

The balance between simplicity and functionality makes this project suitable for families and small businesses/organizations that aim to browse securely and productively.

2.2 Objectives

The **Firewall Simulator** aims to create a user-friendly web browser integrated with a firewall system to enhance internet security and control. The primary goal is to enable users to block unwanted or harmful websites, monitor attempts to access these sites, and provide analytical insights that helps in decision-making.

1. To define a list of blocked websites and deny access to specified domains.
2. To record and display all attempts to visit blocked sites.
3. To present complex analytics to users in the form of graphs and data visualizations regarding blocked attempts.
4. To design a user-friendly and interactive graphical user interface (GUI) to simplify the management of blocked websites and improve navigation.
5. To ensure a smooth browsing experience by integrating essential features such as navigation controls, an address bar, and options for setting up a homepage.
6. To enable users to view browsing data trends, patterns, and the frequency of attempts to access blocked sites.

2.3 Scope of Work

The scope of the project is described as follows:

1. Included Features
   - **Firewall Integration:** A customized web browser with a built-in firewall system, which blocks specific websites based on user input.
   - **Logging Mechanism:** Records all attempts made to access blocked sites along with timestamps.

- **Statistical Tools:** Displays graphs of access attempts to blocked sites and highlights the most frequently blocked websites.
- **User-Friendly Interface:** Provides a graphical user interface (GUI) with navigation controls, site management tools, and status monitoring dialog.
- **Custom Error Pages:** Displays a custom message when users attempt to access a blocked website.

2. Excluded Features
   - **Threat Detection:** The project does not include malware detection or advanced cybersecurity features.
   - **Network-Wide Firewall:** The firewall functionality is limited to the custom browser and does not extend to the entire network or other browsers.
   - **Cloud Storage Integration:** The project does not feature the ability to save logs or analytics to cloud storage.

3 Methodology

This project employs an iterative development methodology, emphasizing a structured and flexible approach. The key steps involved are:

1. **Requirement Gathering and Analysis:**Identification of core features, constraints, and potential challenges.Development of detailed use cases and user stories to guide the design and development process.
2. **System Design and Architecture:** Design of the system architecture, including:
   - User Interface (UI) Design: Creation of a user-friendly and intuitive interface using PyQt5, incorporating elements like navigation controls, address bar, blocklist management, and data visualization.
   - Firewall Logic Implementation:** Integration of firewall logic into the browser's request handling mechanism, defining rules for website blocking (e.g., by domain, IP address, keyword).
3. **Development and Implementation:** Incremental development of the browser application, starting with core functionalities and gradually adding features:
   - Basic browsing functionality (navigation, address bar, rendering web pages).
   - Implementation of the core firewall logic.
   - Integration of logging and monitoring features.
   - Development of the user interface with PyQt5.
   - Implementation of data visualization and analytics capabilities.

4. **Testing and Evaluation:**
    - Verifying that all core functionalities, such as website blocking, logging, analytics, and user interface interactions, operate as expected.
    - Evaluating the ease of use and user experience through user feedback and observation.

4 Features and Functionality

The Firewall Simulator carries a number of core features that enable users to safely, efficiently, browse across the internet and most importantly, in an end-user-friendly manner. Below are the main features with the innovations they bring:

1. **Website Blocking**
    - It has included a feature of blocking the selected websites so that users can't access them during the browsing process. It is directly available within the browser, so users can easily manage blocked sites without complicated configurations.
2. **Real-time Logging of Attempts Blocked**
    - Tracks and maintains logs of all attempts toward accessing blocked websites, including timestamps. A clear view of blocked activity allows the user to trace unauthorized access in real-time, essential for both individual and organizational use.
3. **Data Visualization and Analytics**
    - It shows graphical representations of blocked attempts over time and highlights the most frequently blocked sites. It not only blocks websites but also gives insights into browsing patterns. Users can take security-related decisions for their browsing activities.
4. **Custom Blocked Page**
    - Displays a customized page when attempting to access a blocked website, clearly explaining the reason for the block and lets users know why a site is blocked, improving user experience with clear and personalized messaging.
5. **Dynamic Management of Blocked Sites**
    - It is very easy for users to add or delete sites in real-time from the block list providing an easy solution to manage website access with great flexibility, and without need to restart the browser or perform additional configurations.
6. **Easy Interface**
    - A clean, intuitive interface combining basic web-browsing features like an address bar and navigation controls with the functionality of a firewall.The integration of

security features into a standard browser environment ensures users can manage their browsing without an extensive learning curve, merging usability and functionality.

5 Technologies and Tools Used

- **Programming Language:** Python
- **Framework:** PyQt5 for GUI development
- **Libraries:**
  - **Matplotlib:** Used for generating visualizations such as graphs and charts to display insights like top blocked websites and time-series analysis of blocked attempts.
  - **Collections (Counter, defaultdict):** Utilized for data processing, including counting blocked site occurrences and organizing data for visualizations.
- **Platform:** Cross-platform compatibility with Windows, Linux, and macOS
- **Supporting Tools:**
  - **Git and GitHub:** Used for version control and collaboration, hosting the project repository for easy access and contribution.
  - **IDEs:** Development was primarily done using IDEs like PyCharm or Visual Studio Code for efficient coding and debugging.

Why These Technologies?

- The combination of Python and PyQt5 ensured a balance of simplicity and power.
- Matplotlib facilitated effective data visualization, making complex data easier to understand.
- These tools collectively provided a robust framework for building, testing, and deploying the application efficiently.

6 Data Flow Diagram

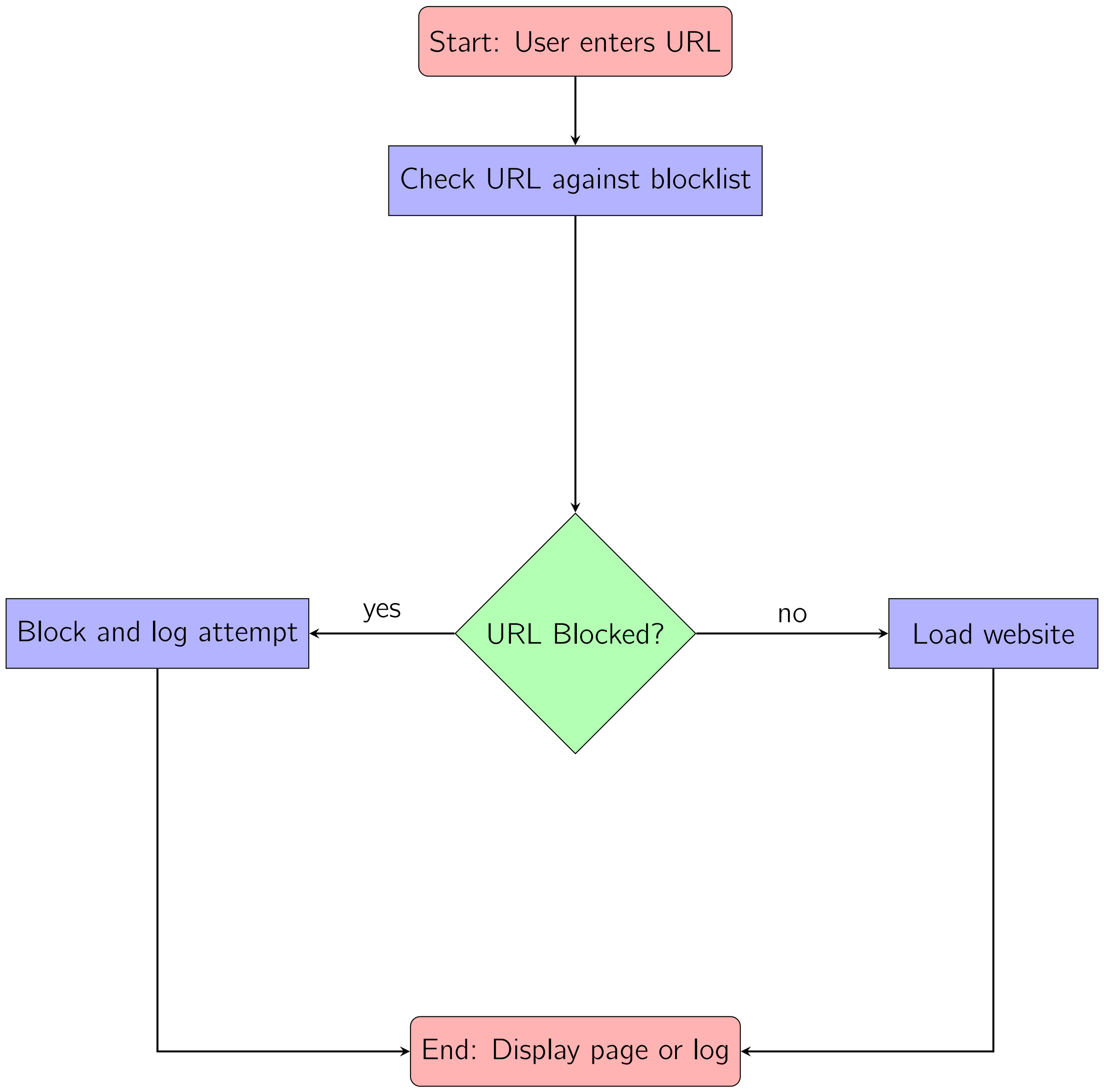
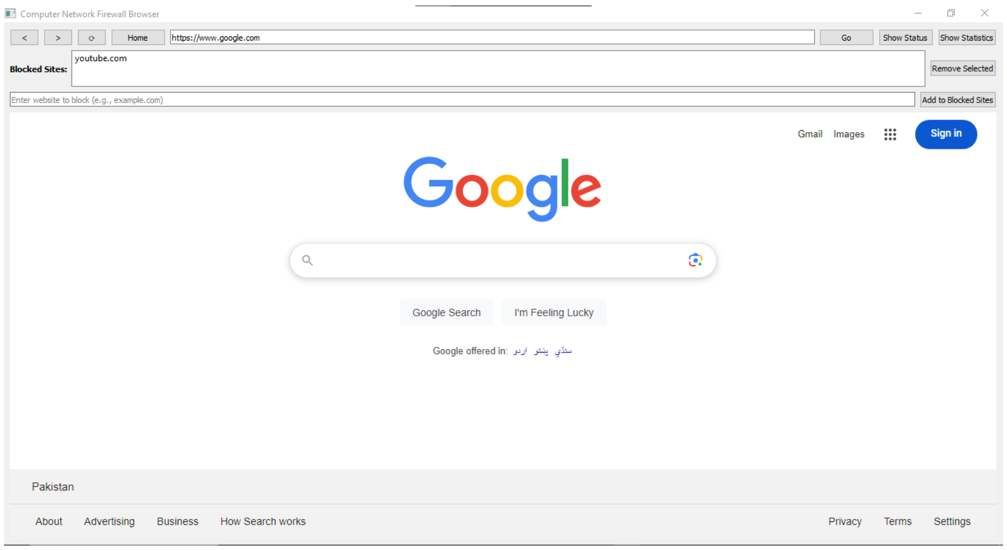


Figure 1: Data Flow Diagram for Firewall Web Browser Application

7 Tool Screenshots

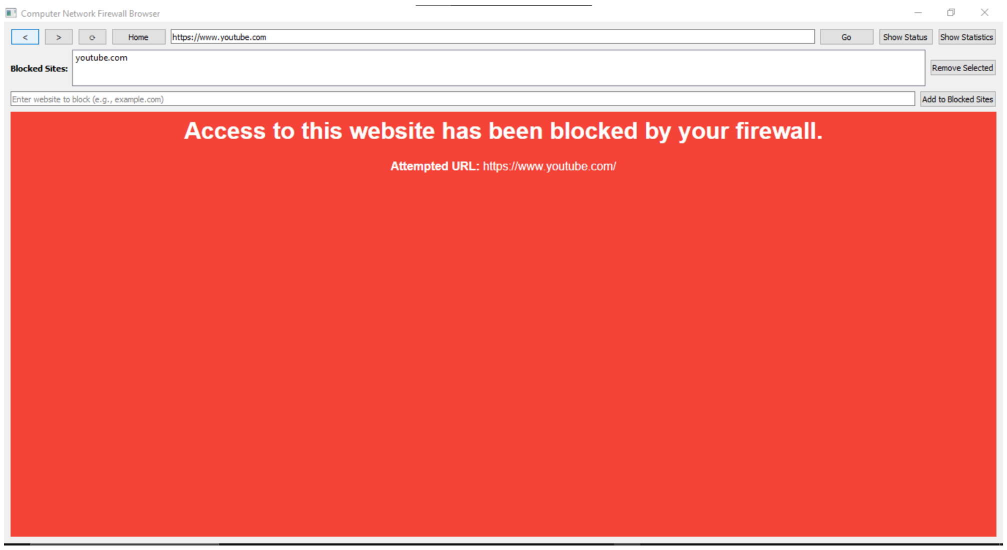
The following are screenshots from the system:



https://www.google.com

youtube.com

Figure 2: Home Screen of Browser



https://www.youtube.com

youtube.com

Access to this website has been blocked by your firewall.

Attempted URL: https://www.youtube.com/

Figure 3: Blocked Website

8 Results and Analysis

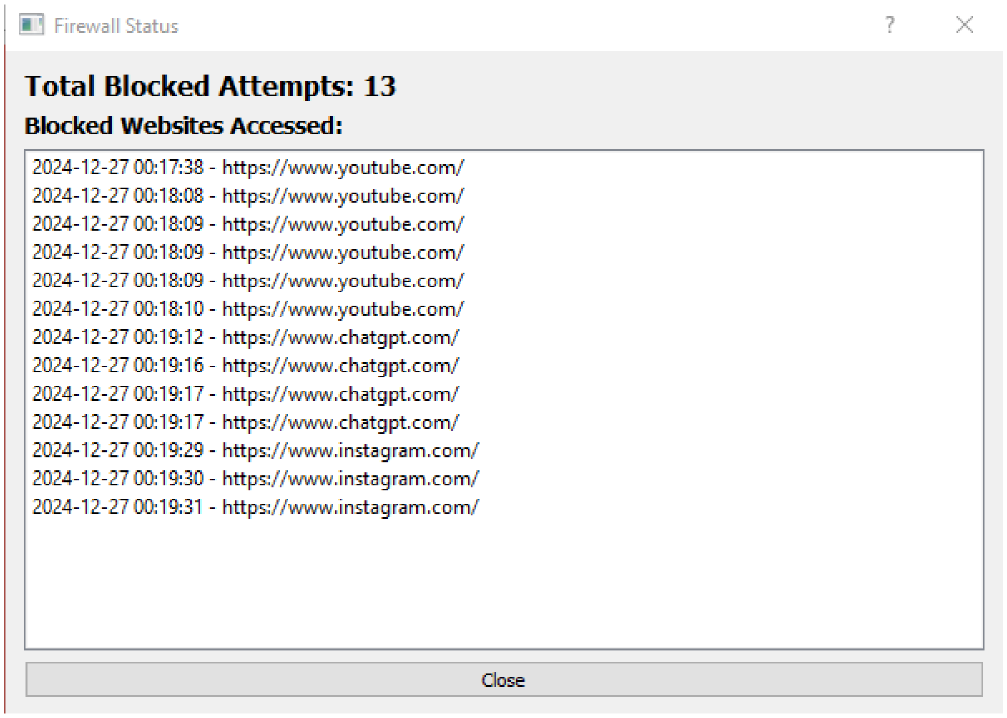
8.1 Project Outcomes

- A successful implementation of a web browser integrated with a firewall mechanism, capable of blocking access to specified websites.

- The simulator features user-friendly interface, providing seamless navigation, easy website management, and clear visualization of browsing data.
- The simulator effectively enforces website blocking rules, preventing access to unauthorized or harmful websites.
- The simulator provides valuable insights into user browsing behavior through real-time logging and comprehensive statistical analysis, including visualizations of blocked attempts and top blocked sites.

8.2 Alignment with Objectives

- The simulator successfully implements the ability to define and enforce a list of blocked websites, effectively denying access to specified domains.
- The simulator accurately records and displays all attempts to access blocked sites, providing a detailed audit trail for security analysis.

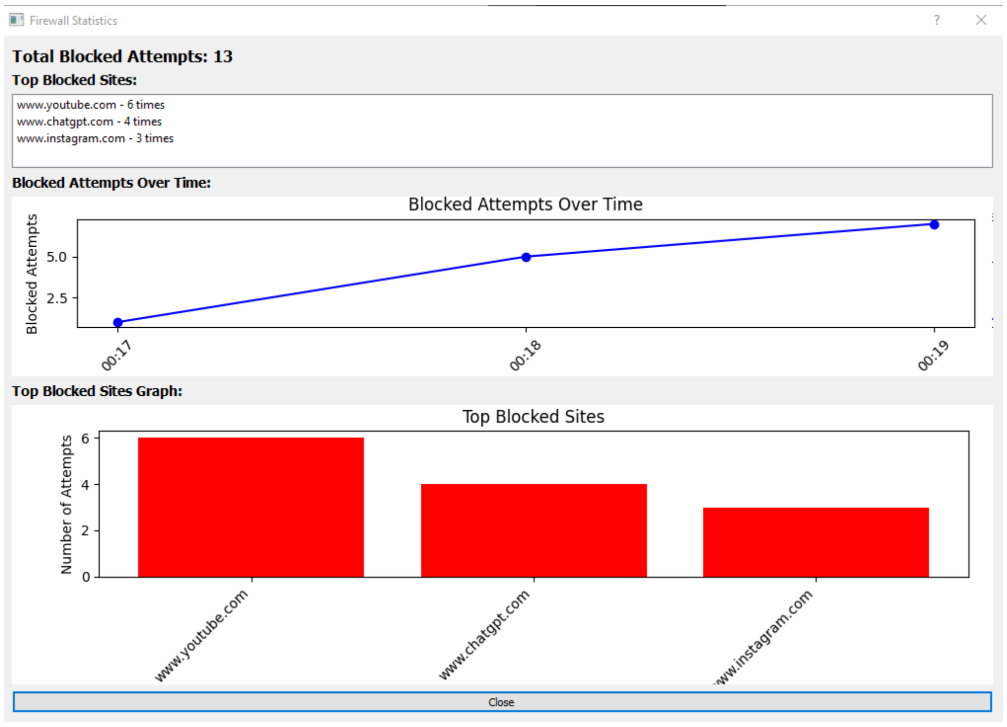


Total Blocked Attempts: 13

Blocked Websites Accessed:

```
2024-12-27 00:17:38 - https://www.youtube.com/
2024-12-27 00:18:08 - https://www.youtube.com/
2024-12-27 00:18:09 - https://www.youtube.com/
2024-12-27 00:18:09 - https://www.youtube.com/
2024-12-27 00:18:09 - https://www.youtube.com/
2024-12-27 00:18:10 - https://www.youtube.com/
2024-12-27 00:19:12 - https://www.chatgpt.com/
2024-12-27 00:19:16 - https://www.chatgpt.com/
2024-12-27 00:19:17 - https://www.chatgpt.com/
2024-12-27 00:19:17 - https://www.chatgpt.com/
2024-12-27 00:19:29 - https://www.instagram.com/
2024-12-27 00:19:30 - https://www.instagram.com/
2024-12-27 00:19:31 - https://www.instagram.com/
```

Figure 4: Attempts to Open Blocked Website



Firewall Statistics

Total Blocked Attempts: 13

Top Blocked Sites:

```
www.youtube.com - 6 times
www.chatgpt.com - 4 times
www.instagram.com - 3 times
```

Blocked Attempts Over Time:

Top Blocked Sites Graph:

Figure 5: Statistics for blocked Website Access

9 Conclusion

The Firewall Web Browser Application demonstrates the integration of firewall features into a custom web browser. Key takeaways include:

- The system effectively controls web access with user-defined rules.
- Visualization tools provide valuable insights for administrators.

Future improvements could include:

- Integration with machine learning for advanced threat detection.
- Support for category-based blocking (e.g., social media, gambling).
- Enhanced performance optimization for large-scale use.

10 Github Link

The project repository can be accessed at:

`https://github.com/afferafatima/Firewall-Simulator`

References

[1] GeeksforGeeks, “Introduction of Firewall in Computer Network,” Available: `https://www.geeksforgeeks.org/introduction-of-firewall-in-computer-network/`. [Accessed: Dec. 26, 2024].

[2] UMA Technology, “Firewall Rules Explained: From Basics to Best Practices,” Available: `https://umatechnology.org/firewall-rules-explained-from-basics-to-best-practices/`. [Accessed: Dec. 26, 2024].

[3] PyQT5, “PyPI,”Available:`https://pypi.org/project/PyQt5/`. [Accessed: Dec. 26, 2024].

[4] GeeksforGeeks, “What is a Web Application Firewall?,” Available: `https://www.geeksforgeeks.org/what-is-a-web-application-firewall/`. [Accessed: Dec. 26, 2024].

[5] “QWebEnginePage,”Available:`https://doc.qt.io/qtforpython-5/PySide2/QtWebEngineWidgets/QWebEnginePage.html`. [Accessed: Dec. 26, 2024].

[6] GeeksforGeeks,“Firewall Design Principles,”Available:`https://www.geeksforgeeks.org/firewall-design-principles/`. [Accessed: Dec. 26, 2024].