

GAZI UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING



A Secured TCP Chat Room using Python

Outline

Dr. Lecturer Member of FEYZA YILDIRIM OKAY

INTRODUCTION TO WIRELESS AND MOBILE NETWORKS (CENG473)

Selin Cansu AKBAŞ - 191180005

Anıl KOÇER - 191180057

Oğuz Kaan SUBAŞI - 191180076

Summary

In the age of the internet, communication has become fast and widespread, but it has also brought about security concerns. Some chat applications may fall short in ensuring the security of users. Therefore, we believe that developing a secure TCP chat room, which provides a secure communication environment, may be necessary. In our project, we plan to create a secure TCP chat room. In this room, we plan to use text encryption with security measures to protect users' data. In our initial research findings, we have seen that a secure TCP chat room is possible and that user data security can be ensured. In summary, this project aims to contribute to making communication on the internet more secure by preserving users' privacy and providing a secure communication environment.

1) Introduction

- A. **Problem Definition:** In areas of heavy internet usage, advanced internet protocols will be employed to prevent delays in communication and packet loss, thus ensuring the efficiency of the application. Another issue in chat room applications is inadequate security. Since the messages will be text-based, we aim to address this problem by utilizing text encryption protocols.
- B. **Methods:** After conducting our research, we have identified IPV6 as the most efficient communication protocol for addressing the aforementioned issues. Additionally, we have decided to utilize SSL/TLS for message encryption.
- C. **Motivation:** In today's world, the internet has become a tool that facilitates communication among people. However, the large amount of data exchange and information sharing on this communication network also brings security risks. Especially when sharing certain information and personal data, privacy and security are major concerns for users. Chat applications often encounter security vulnerabilities. Without security measures, users' personal information and messages can fall into the hands of unauthorized individuals. From this perspective, developing a secure TCP chat room is important for enhancing security on the internet and protecting users' data. This project is designed to meet the need for secure communication and raise awareness among users. A secure TCP chat room will enable users to communicate securely by using encryption techniques to protect their

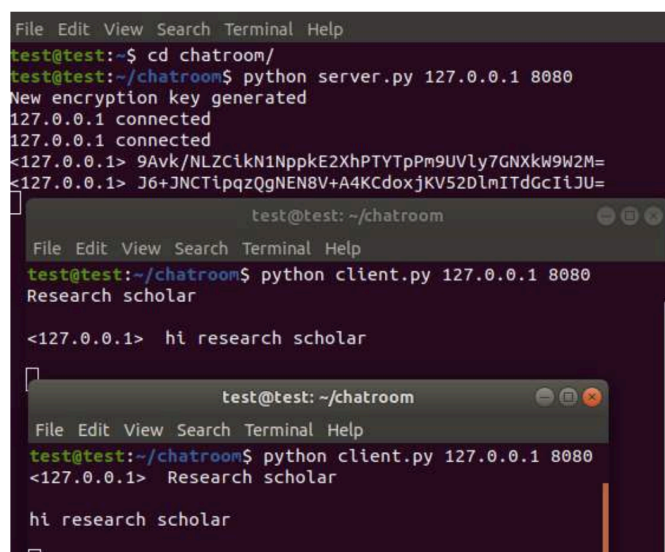
data. This will preserve the confidentiality and integrity of conversations between users and provide stronger defense against potential attacks. In summary, the primary motivation behind this project is to increase awareness of secure communication and provide guidance to those interested in developing similar projects.

- D. **Contribution:** Our project aims to contribute to the literature in the following ways. As a chat room project utilizing IPV6, we aim to develop an advanced project in terms of optimization. Additionally, since it will be a project with high security measures, we aim to enhance the SSL/TLS system.

2) Studies in the Literature

In this section, studies in the literature similar to this project will be examined.

- Secure Chat Room Application using Advanced Encryption Standard Algorithm



```
File Edit View Search Terminal Help
test@test:~$ cd chatroom/
test@test:~/chatroom$ python server.py 127.0.0.1 8080
New encryption key generated
127.0.0.1 connected
127.0.0.1 connected
<127.0.0.1> 9Avk/NLZCikN1NppkE2XhPTYTpPm9UVly7GNXkW9W2M=
<127.0.0.1> J6+JNCTipqzQgNEN8V+A4KCdoxjKV52DlmITdGcIiJU=

test@test: ~/chatroom
File Edit View Search Terminal Help
test@test:~/chatroom$ python client.py 127.0.0.1 8080
Research scholar

<127.0.0.1> hi research scholar

test@test: ~/chatroom
File Edit View Search Terminal Help
test@test:~/chatroom$ python client.py 127.0.0.1 8080
<127.0.0.1> Research scholar

hi research scholar
```

Figure 2.1. Chat Room

This study implements a secure group chat application using cryptographic algorithms. Socket programming is utilized for communication between two clients and a server. Clients need to establish a connection to the server before sending data. Data encryption is performed on the client side, and the encrypted message is then transmitted to the server. The server forwards the data to other clients except for the one that sent the message, and the data is decrypted on the receiver's end. Random is used for generating random keys. AES algorithm

and CBC mode are employed for encryption and decryption processes as they are known to provide higher security against known cryptographic attacks.

- Realtime Chat Application Using Client-Server Architecture

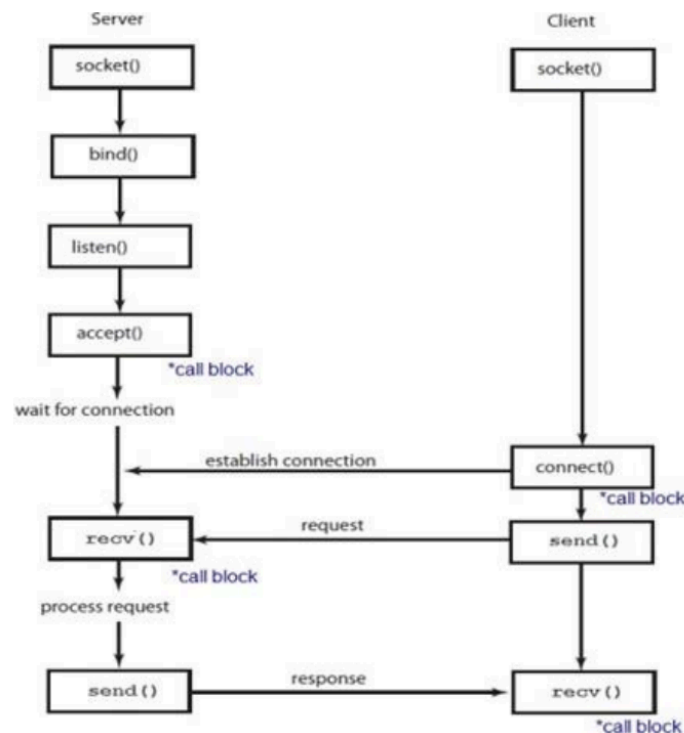


Figure 2.2. System Architecture

This study focuses on a chat system that originated in the mid-1980s and was quite popular at the time. The chat application represents communication between two entities (sender and receiver). In an era where security and internet penetration are increasing daily, these issues are emphasized. In this application, a server and several client connection points are created, and clients communicate with the server using a connection module. These connection points are endpoints for sending and receiving data. Each organization has two connection points. The program is implemented using TCP connection, which will connect to a specific port on the machine or local host. In the case of the client, a connection point will connect to the same port used on the server side. This application can assist many professional organizations, colleges, preferred schools, universities, and IT companies. Therefore, we aim to design this application for the local networks of these organizations. People can communicate and exchange ideas within the LAN using many features of this chat application. As a result, server-client applications can be used to perform various queries

such as medical helpline services and user reporting, and can be utilized in different scenarios. This provides an effective approach to creating a chat process today without UI or any other human intervention.

3) A Secured TCP Chat Room using Python

In our project, a secure chat environment will be created by integrating the TCP protocol and message encryption protocols to provide secure communication between users.

The details of the project are as follows:

- *TCP Protocol*: TCP protocol is widely used and successful in ensuring secure communication. The advantages it provides include the preservation of data integrity and the security of established connections.
- *SSL/TLC*: Since our application will be transmitting text-based data, there is a need for encryption for security purposes. This encryption will be performed before the message is sent and will be in a format that can only be decrypted by the receiver. We have decided to use the SSL/TLS protocol for our application.
- *Interface*: Users require an interface for comfortable and understandable messaging. Since we will be programming our application using the Python language, we have decided to design our interface with PyQt5, considering its compatibility for performance.
- *Secure Session Management*: We will implement user sessions to ensure the security of users, allowing each user to connect to the network with their own session. User login data will be stored in encrypted form. This adds multiple layers of security and reduces the likelihood of attacks on the application, as well as the possibility of unauthorized access to sent messages.
- *Performance*: In order for our project to operate with high performance and not to get congested when there is high traffic, with no packet loss or delays, we have decided to use the IPv6 protocol instead of IPv4.

4) Open Issues

TCP/IP protocol provides a basic framework for creating a secure TCP chat room. However, there are still unresolved issues. In this section, we will discuss some of these open issues.

- *Security*: The TCP/IP protocol offers various security mechanisms to maintain data confidentiality and integrity. However, these security mechanisms are not always flawless. For example, the TCP/IP protocol is vulnerable to attacks without additional security layers (such as authentication and access control), which can allow unauthorized users to access the chat room and compromise data.
- *Performance*: The TCP/IP protocol is designed for high performance and scalability. Nevertheless, bottlenecks can occur in some cases. For instance, if a large number of users join the chat room or there is a large amount of data input, delays and data loss may occur, negatively affecting the user experience.
- *Compatibility*: The TCP/IP protocol is supported by most operating systems and network devices. However, it is not fully compatible on all platforms. Therefore, establishing connections between users operating on different platforms can be challenging.
- *Error Handling*: The TCP/IP protocol provides mechanisms to handle network errors and data corruption. However, these mechanisms do not always work flawlessly, leading to interruptions or data loss in chat sessions.
- *Extensions*: The TCP/IP protocol supports the use of extensions to add additional features and functionalities to the chat room. However, not all extensions are compatible and may introduce security vulnerabilities.

Note: In this section, some general challenges and unresolved issues related to the TCP/IP protocol have been discussed. Depending on the specific requirements of the project, there may be additional open topics that need to be addressed.

5) Results

In conclusion, although communication has become rapid and widespread in the internet age, it has also brought along security concerns. Some chat applications may fall short in ensuring user security. Therefore, we believe it is necessary to develop a secure TCP chat room to protect user data. In our project, we plan to create a secure TCP chat room that provides a secure communication environment to safeguard user data. Our research has shown that it is possible to have a secure TCP chat room and ensure the security of user data. In summary, this project aims to contribute to making communication on the internet more secure, providing a secure communication environment while preserving user privacy. Our project aims to create a secure chat environment that enables secure communication among users by integrating TCP protocol and message encryption protocols. Among the advantages provided by the TCP protocol are data integrity and security of established connections. Additionally, message encryption will be performed using the SSL/TLS protocol since it involves text-based data communication. Users require an interface for comfortable and understandable communication. Therefore, we have decided to design the interface for our project using the Python language with PyQT5. By providing secure session management, we will ensure that each user connects to the network with their own session. In the conclusion section of this report, we summarized the general objectives of our project and the results of the literature review. As the project progresses, more detailed results and findings will be obtained, and a detailed presentation of the project will follow this report.

References

- 1- Hassan, G. M., Hussien, N. M., & Mohialden, Y. M. (2023). Python TCP/IP libraries: A Review. *International Journal Papier Advance and Scientific Review*, 4(2), 10-15.
- 2- SINGH, AJIT, PYTHON SOCKET PROGRAMMING, 2019.
- 3- Salihu, E., & Blakaj, G. (2021). Workplace Chat Application Using Socket Programming in Python.
- 4- Uchenna, U. I., Gregory, U. S., Virginus, U. N., Angela, O. A., Enyioma, C. K., Ezeora, N. J., ... & Michael, I. U. (2021). Exploring a Secured Socket Python Flask Framework in Real Time Communication System. *Asian Journal of Research in Computer Science*, 8(1), 77-87.
- 5- Shah, A., Servar, M. G., & Tomer, M. U. (2022). Realtime Chat Application using Client-Server Architecture. *International Journal for Research in Applied Science and Engineering Technology*, 10(5), 2575-2578.
- 6- Kagane, A. (n.d.). *Chatroom using python project report*. Scribd.