University of South Wales
Prifysgol De Cymru

Secure Chat Server

Jessica Clara Fealy - 18024092

Introduction

This project researches and determines the feasibility of developing a secure chat server using C# and related industry standard protocols. For the success of this project research is required into 3 primary fields of both secure programming and network security, these being attack vectors, encryption and general industry tools; using these researched topics this project is able to develop a solution for a secure chat server. The success of this project this weighted against the included features, user feedback, and testing of critical systems.

Objectives

This project had 5 primary goals; 3 research topics into threats and mitigations, encryption methodologies, and general industry standard tools and processes. 1 development objective to implement research into a secure chat server this also includes the prototyping stages. Final objective to evaluation and assess the project.

Project planning and management

3 primary deadlines were asserted for the completion of this project being sectioned into 3 milestones. Each milestone had a set of required activates that were required to be completed for the smooth progress of the project, to keep track of project planning and progress a Gantt Chart was employed.

GitHub was used heavily throughout this project as a source control, with 2 instances relied on roll backs due to fatal errors in the primary folders containing this project.

Research

Threats and Mitigations

Two primary threats were discussed in the research sections of this project, these being account jacking and man in the middle attacks.

Where account jacking relates to unauthorised access into an account via password theft or brute force attacks, with mitigation for this being to ensure only hash values are saved and a strict password policy (Bamiah and Brohi, 2011). Man in the middle attacks relates to the security between client and server where data can be stolen in transit, mitigations for this to be using encrypted messages and SSL (Swinhoe, D., 2021).

Cryptography Methodology

Research into the 2 different methodologies in cryptography, these being symmetric and asymmetric (Information Commissioners Office, n.d.). Where asymmetric uses a set of keys of keys known commonly as public and private keys; public keys can be freely shared and is used for the encryption of data, where private keys must be kept secured and are used for the decryption of data (Simmons, 1979). Symmetric uses a single key for both encryption and decryption operations (Smirnoff and Turner, 2019) , this methodology isn't secure to be transmitted over open air. The solutions for this is a encryption handshake, where you use asymmetric to encrypt a symmetric key for transmission then decrypt the symmetric key. This allows both computers to know the symmetric key. RSA and AES were decided to be used in this project for asymmetric and symmetric (Anton, 2019).

Figure 1: Flow of encryption handshake

Industry technologies, libraries, and protocols

Research discovered various technologies that should be implemented into the project; these include, SSL, MongoDB, Log4Net, input validation, hashing, and the complete authorisation system.

Figure 2: SSL Logo

Prototyping and Implementation

Prototyping

Prototyping in this project was used to better understand the requirements and processes for certain applications of code. The main areas that required prototyping was the TCP connections between server and clients, hashing, loggings, and encryption. What was learnt was used for implementation.

Implementation

The implementation stage was the longest stage of this project and was required to satisfy objective 4 set out in the planning phases.

For the development of this project critical system were required to be implemented this included, JSON messaging, persistent account storage, logging, secure hashing, SSL, registration and login systems. These critical systems existed on both client side and server side.

Server

The server managed:

- Local admin control
- MongoDB management
- Client management
- Main authorisation
- RSA creation
- And much more

Client

The client managed:

- Data inputs from user
- Displaying messages
- AES Creation
- Sending data securely to server
  - Login and registration
- And more

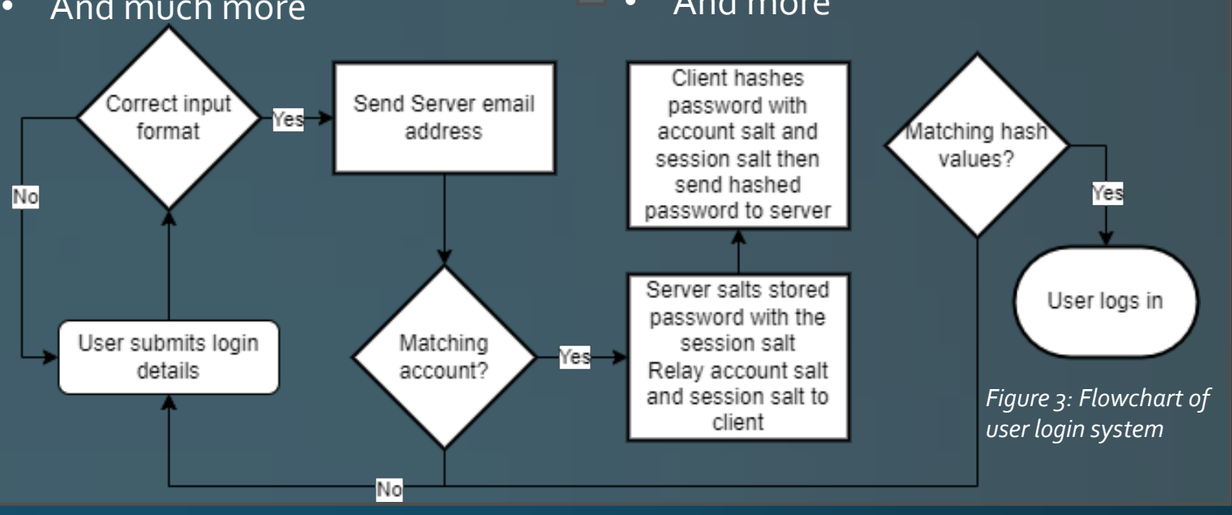

Figure 3: Flowchart of user login system

Results

To assess the success of this project both testing and user feedback was required to appropriate determine the feasibility of a secure chat server using C#. With questionnaires being used highlighting all major functionality operating without issue, where users are able to create account and login to them then they are shown the chat window. All tests preformed were successful.

Results from questionnaires did present some security vulnerabilities, where 2 respondents were capable of impeding the function or security of the server in considerable ways. These results do not significantly impose on the assessment of success, but does show where future research and improvement are required for a continued success.

This project was successful at showing the feasibility of researching and developing a C# secure chat server, with all researched attack vectors being appropriately responded to with SSL and encryption. There is a clear sight for improvements required in this project.

References

Bamiah, M. and Brohi, S., 2011. Seven Deadly Threats and vulnerabilities in Cloud Computing. INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES, 9(1), pp.087-090.

Swinhoe, D., 2021. What is a man-in-the-middle attack? How MitM attacks work and how to prevent them. Available at: https://www.csoonline.com/article/3340117/what-is-a-man-in-the-middle-attack-how-mitm-attacks-work-and-how-to-prevent-them.html (Accessed 19 October 2021).

Information Commissioners Office. n.d. What types of encryptions are there? Available at: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/encryption/what-types-of-encryption-are-there/ (Accessed 19 October 2021).

Simmons, G., 1979. Symmetric and Asymmetric Encryption. ACM Computing Surveys, 11(4), pp.305-330.

Smirnoff, P. and Turner, D., 2021. Symmetric Key Encryption - why, where and how it's used in banking. Available at: https://www.cryptomathic.com/news-events/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking (Accessed 19 October 2021).

Anton, T., 2019. The need to manage both symmetric and asymmetric keys. [online] Cryptomathic.com. Available at: <https://www.cryptomathic.com/news-events/blog/the-need-to-manage-both-symmetric-and-asymmetric-keys#:~:text=Typically%2C%20once%20a%20secure%20connection,is%20known%20as%20hybrid%20encryption.> [Accessed 23 December 2021].

Conclusion

This project aimed to determine the feasibility of developing a secure chat server in C# with standard industry security tools; this project researched various areas of cybersecurity and used that research to successfully develop a final solution in 2 programs (server and client). Security flaws were identified and evaluations were made with theses considerations. This project, which considered a success, has plentiful room for improvements.

Future work and improvements

Additional research into attack vectors will be needed for future work, as well as solving vulnerabilities raised in the results section. Also, additional features such as end-to-end encryption and remote admin controls would be considered for future improvements. As well as quality of life features raised through user feedback. Improvements could be endless as the threat landscape is forever changing.