Elliott Picker
Secure Chat
MSCS_630L_711_20S

Abstract

Secure Chat is an open-source lightweight desktop app that enables encrypted end-to-end communication through a centralized server. All encryption occurs on the client side so the messages broadcasted to and from the server are unreadable by anyone other than the Secure Chat users with the appropriate key. The symmetric encryption depends on the users sharing the key external to the application such that the server plays no part in identifying or authenticating any users.

Introduction

In today's world it seems that cyber hacking and data breaches are a recurring theme in the news. Properly securing all information including communication is a challenging task that can have severe consequences when not done properly. While some cases can lead to embarrassing conversations making it to the public, business confidential secrets can also be leaked and in some cases even compromise national safety. End-to-end encryption offers its own challenges and often leaves possible exposures where data may not be fully secured.

Secure Chat is a lightweight open-source desktop app that enables end-to-end encrypted communication through a centralized server. Because all encryption occurs on the client side and there is no authentication or key management, Secure Chat is extremely secure so long as the key file is protected. Because the program is very lightweight, any skeptics of the security of the program can easily read through the source code and ensure that there are no backdoors present and all content will be properly safeguarded. Because the server has no role in authenticating users, the server doesn't have the ability to share the encryption key or the content of messages exchanged, even in the event of some type of breach of hack.

Background and Related Work

While many existing end-to-end encrypted chat services exist today many people have doubts about the security and integrity of such services. For example, while the popular messaging service WhatsApp boasts end-to-end encryption, some people fear possible “backdoors” which could allow for someone other than the intended end users to possibly access this data. Because the source code is not publicly available, users are left to trust the developers when they claim that the data is properly safeguarded and secured. While many government agencies are in favor of these types of backdoors for authorized use in necessary circumstances, once such backdoors exist, they become a prime candidate for hackers and other malicious users attempting to gain unauthorized access to the information. Additionally because many services allow a user to logon from multiple devices and access their same message histories, the server must somehow store the private key, and is tasked with authenticating users prior to allowing use of this key. If someone can falsely convince the server they are an authorized user, they can gain access to the appropriate secret key and access the encrypted messages. For example, while someone's data may be properly encrypted and secured, if a malicious user can convince the server that they are that user, the server will allow use of the encryption key and expose all the otherwise safeguarded data.

As of Spring 2020 there is a bill in US Congress, called the Eliminating Abusive and Rampant Neglect of Interactive Technologies Act (EARN IT Act) that may threaten the non-liability of end-to-end encryption messaging apps. Because these services typically have no ability to access encrypted user data, they have no ability to police what type of content is exchanged on the application. This bill proposes that certain companies can be held responsible for user-uploaded content regardless of how safeguarded the information is, denying their inability to access the data as a valid defense of non-liability.

Methodology

A key file created from the Secure Chat program is vital to the integrity of the encrypted data. This key file contains both the 16 byte hex string used as a cipher key, as well as the ip address of the server. Because these fields are stored as plaintext in the key file, proper safeguarding of the key file is essential to the security of the chat. The key file can be shared between authorized users in a secured way such as physically transmitting on a flash drive. If the key file itself cannot be safely transmitted, a plain text passphrase can also be communicated that can be used by the keyCreator program to produce identical keys for each user.

The Key Creator program converts a unique passphrase into a bit encryption key. It does so by breaking the input passphrase into 128-bit sections, which are then combined to form one 128-bit hex cipher key. The key file produced consists of both this unique key as well as the path of the server. There is no use of public keys in the key sharing as the server takes no part in deciding who users are or who is authorized to the chat. Some alternative end to end encrpytion programs make use of public and private keys to securely produce a cipher session key. Authorization for Secure Chat is done completely external to the Secure Chat application, assuming that only authorized users will have access to the appropriate key or passphrase used. While this may seem like a disadvantage in that another means of communication is necessary for key sharing, it enables the server to know nothing of the session key being used. This prevents the server from being able allow untrusted access, either through a valid source such as a government under a warrant or an invalid source such as a malicious hacker.

Discussion and Analysis

The Secure Chat client program encrypts all messages before sending them to the server through use of the javax.crypto.* libraries performing symetric encryption using the cipher session key found in the key file. The client program additionally decrypts all messages received from the server with this same key, so that the end user can read the messages in plain text. This assures that the authorized users of the chat client are the only people able to access the content of the messages exchanged. The server itself and the communication sockets are unable to decipher the messages being sent back and forth. Additionally while the server will be aware of the number of connected users, it has no knowledge of the identity of the users. For this reason it is unable to determine which users are actively participating or which user logged off in the event of a logoff. Anyone with the valid key will be permitted to join the chat and use any username they wish. By using the javax.crypto libraries, secure chat is not tasked with implementing the encryption itself nor any possible bugs or vulnerabilities associated with that. As vulnerabilities are detected and fixed in the javax.crypto library, Secure Chat can be recompiled to include those changes. If some new encryption standard becomes commonplace in the future, Secure Chat is modularized in such a way that including a new encryption method should be a trivial effort.

Conclusion

Secure Chat is an easy to use solution for end-to-end encryption messaging. Because the keys are not passed through the application in any way, a properly protected and shared key file will ensure the security of messages transmitted through the application. Additionally since the program is lightweight and open-sourcem all of it's integrity claims can be easily validated by any skeptical users. Because users are free to recompile both the chatClient client program at any time, and host the chatServer server program themselves, users can be assured that they are in complete control of their conversations. For any users looking for a simple way to share a confidential chat with multiple users, Secure Chat is a simple solution where users can be assured their message content will only be accessible to those who have the key.

References / Bibliography

[1] Kenyon, Miles. “Secure Your Chats: Why Encrypted Messaging Matters.” *The Citizen Lab*, 8 May 2019, citizenlab.ca/2017/11/secure-your-chats-encrypted-messaging/.

[2] Thakkar, Jay. “End-to-End Encryption: The Good, the Bad and the Politics.” *Hashed Out by The SSL Store™*, 2 Dec. 2019, www.thesslstore.com/blog/end-to-end-encryption-the-good-the-bad-and-the-politics/.

[3] Strickland, Jonathan. "End to End Encryption Services." *Tech Stuff. HowStuffWorks, Inc*. 24 April, 2018 https://www.iheart.com/podcast/105-techstuff-26941194/episode/end-to-end-encryption-services-29237343/