Steam Stream Team Supreme
Daniel Jacobo
Emil Evangelista

Phase I

Application Properties

Our chat is a 1 to 1 user messaging application. The features included will be user accounts, clickable web links in the chat, and basic chat. Our extra features planned are pictures, 2-factor authentication, and group chat.

The platform is planned to be a web application. The language we will be using is Spring for Java.

High Overview: We plan to implement our RESTful server with a class for the server and user. For the database, we will use SQL. For encryption, we will use AES-256. For message integrity, we will use HMAC-SHA256. For key exchange, we will use the Diffie-Hellman method. For client authentication, we will use TLS handshakes.

Stakeholders and Assets

The stakeholders for our application are the users.

The assets include:
Messages, User Accounts, Passwords, private emails, the server itself, and the encryption/decryption keys used

Adversarial model

Possible Vulnerabilities

Client Side

- Keyloggers
- Evil Maid Attack
- Shoulder Surfing
- Weak Security Questions
- Weak passwords
    - Bruteforcing Passwords
    - Password Reuse
    - Easily Guessed

Server Side

- DDOS
- Cross-Site Request Forgery
- Cookie Theft
- Harmful Links
- Man in the Middle
    - Tampered Messages
    - Eavesdropping

Possible Related Previous Work

There are many similar services, all with their own philosophies, and therefore their own pros and cons. The two biggest fish are Signal, and Whatsapp, and the main difference between them is their inception.

Whatsapp was a simple messaging system that was bought out by Facebook in 2014. As of April 2016 Whatsapp, with the help of Open Whisper Systems, implemented the signal protocol in order to provide end-to-end encryption. Even though all messages are now secure in transit, they aren't completely safe from prying eyes. Whatsapp stores all messages locally on the device, therefore someone with an insecure phone can become an easy target. Local storage also creates the problem of backups, any backups of your device will also backup all saved messages, meaning that service now has access to all of your communications. On one hand Whatsapp protects your messages, but on the other hand it automatically collects information about you, including your contacts, and metadata. Whatsapp, being a proprietary client, breeds doubt within many security experts. With no way to check and verify the code itself the only thing that is ensured is the open-source signal protocol used to encrypt messages.

Signal released in 2014, developed by Open Whisper Systems. The client is open-source and easily reviewable by anyone, meaning any security holes are quickly found and dealt with. Signal employs its own signal protocol in order to encrypt all messages from end-to-end. Similarly to Whatsapp it requests all of your contacts to compare with its user base. That's where the similarities end. Your contacts are hashed, and therefore more secure, your messages are not transferred when your phone is backed up, and they collect as little metadata as possible.

Complete Description << Describe attacks in detail, and your solutions

We will use OpenSSL for comunication,, AES for encryption, (authentication?), and Implement the server through Java.

Full Rigorous Analysis << Why do your assumptions work? Include assumptions

CHANGES