Phase 1

Nexus | Nexus Messenger

Team Nexus

Adonias Lopez Payan & Henna Gohil

2016

**Application Properties:**

We are developing a messenger app that allows a user to send messages to a single recipient. The messenger app will be developed under the Android JDK and we will be our RESTful webserver for our backend/server. We will be using Signal Protocol for authentication. OpenSSL will be used to implement Transport Layer Security 1.2 (TLS) to handle the security layer and establishes connection between the server and client. Furthermore, we will use MySQL database to store credentials of the users and messages. LetsEncrypt will be used to create key pairs to authorize certificates being passed.

**Assets and Stakeholders:**

The stakeholder for this project will be the user of the app. The asset of the stakeholder is the message being sent through the sever to another client (recipient).

**Adversarial Model:**

* **Outside Attacks:**
  + We will incorporate end-to-end encryption to combat eavesdropping from outside attackers.
  + To limit the effectiveness of a brute force attack, the private-key size is going to be set to 2048-bit.
* **Inside Attacks:**
  + We will use endpoint authentication to combat Man-in-the-Middle (MitM) attacks. OpenSSL will allow us to use TLS to implement an authentication process for both messengers using mutual certificate authority.

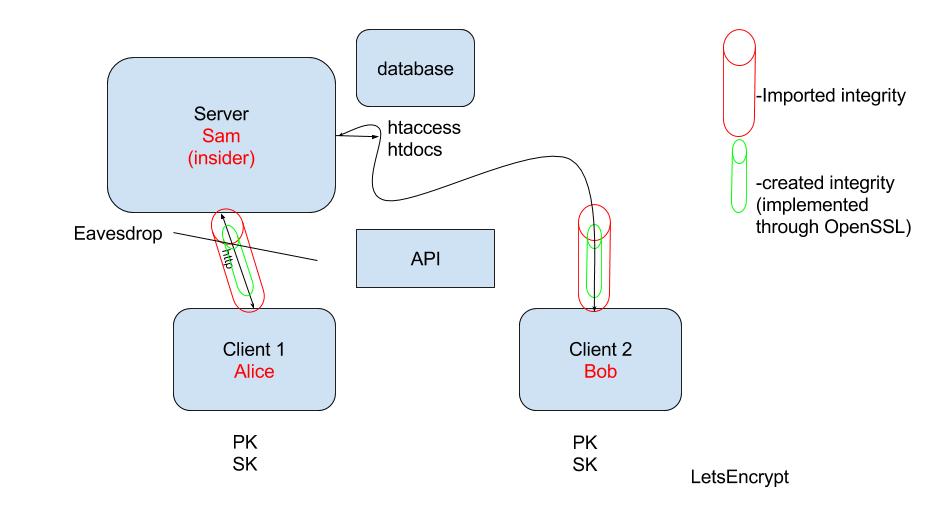
**Possible Vulnerabilities:**

Possible vulnerabilities of our system can consist of a workstation hijacking and DDOS attacks. Our app cannot guarantee security from someone stealing the device and gaining the messages that have been decrypted from the client side. In addition, no mechanisms will be set up to protect the dignity and availability of the server. The Signal protocol does not provide anonymity preservation and it requires servers for the relaying of the messages and storing of public key material.

**Possible Related Previous Work:**

WhatsApp Encryption Overview, https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf

**Use Cases/Diagrams:**



**Complete Description of Solution:**

For our project, we will be using the android API, due to how there are built in distinct components, which can maximize security. Since android is an open source platform, the security is strengthened by its community of android users. Android protects applications and user data, system resources including network, and provides application isolation from the system, and other applications. For the backend/server of our application, we will be using Signal Protocol from Open Whisper Systems, which will help us set up a connection between the client and the server (red tunnel). The green tunnel will be provided by PGP (Pretty Good Privacy) with a public and private key pair. We will be using Signal as our protocol which provides confidentiality, integrity, participant consistency, authentication, forward and backward secrecy, causality and asynchronous communication.

**Full Vigorous Explanation:**

We are usingandroid JDK because of how the open platform requires strong architecture and rigorous security programs. The android design allows us to be more flexible and supports an open platform while still protecting users of the platform. Android security had released a tool for testing SSL, which helps us developers find potential security issues on our application.

The reason why we chose to make a RESTful server is protect HTTP methods and create a secure HTTPS instead of HTTP. Creating a RESTful API for our server protects almost every aspect of client-to-server communication and it secures the integrity of the server.

Furthermore, we are choosing to encrypt our messages with Signal Protocol, which combines the Double Ratchet Algorithm, prekeys, and a 3-Diffie-Hellman (DH) handshake. The Double Ratchet Algorithm allows for end-to-end encryption for instant messaging by managing the ongoing renewal and maintenance of shot session keys after an initial key exchange; combines a cryptographic ratchet based on the Diffie-Hellman key exchange and a ratchet based key derivation function. We will protect our system from MitM attack by using this protocol. Since Signal uses Curve25519, AES-256, and HMAC-SHA256, our system is guaranteed to be protected from brute force attack for about 15 years, according to Signal.