CECS 478 – Introduction to Computer Security

Phase 1 – Design

HDcode

Holly Dinh

John Nguyen

September 23, 2016

Table of Contents

1 Application Properties

This chat messenger application will allow a maximized of two users at a time to communicate with one another through text messages. Text messaging is the process where users are allowed to type out a message in any written language they would prefer and then proceed to send the message through this messenger application. This application will be a web application that will use the internet and web browser to send messages back and forth between communicating users. In order to use this chat application, temporarily called “YuP”, the users must log on to the application with a personal email that they had used to create an account. After creating an account in the system, users will then have the option to find their friends on the application by either indicating their friends’ email addresses or with their unique identification numbers that had been distributed to each user after creating their accounts. After having successfully founded their friends, users will then have the option to add their friends into their buddy list. In order to communicate with one another, two users must be in each others’ personal buddy lists. YuP will work on any desktop and any operation system such as Windows, Mac, and Linux with browsers such as Google Chrome, Firefox, and Safari. Both the browser and platform must be able to access the website for the chat messenger application. In the development process, Java will be used to develop the web application.

2 RESTful Server

By using Java, there is the opportunity to implement the principles of representational state transfer in order to program a functional and simple client to server interaction. With REST, developers can take advantage of HTTP such as PUT, GET, POST, and DELETE in YuP. There will be a GET that will allow the users to request the message sent to them from a server that acts similar to a post office. The server will receive a POST command that is the message being sent out by a user. In return, the server will take and distribute that POST message to the indicated targeted friend.

3 Assets/Stakeholders

This messenger application is focused on providing security, privacy, and functional communication between users utilizing this application to send messages to each other. As a result, the stakeholders of YuP are the users. The users are putting their privacy at risk

while using YuP, therefore the developers and creators of this web application consider the stakeholders' privacy to be an asset.

4 Adversarial Model

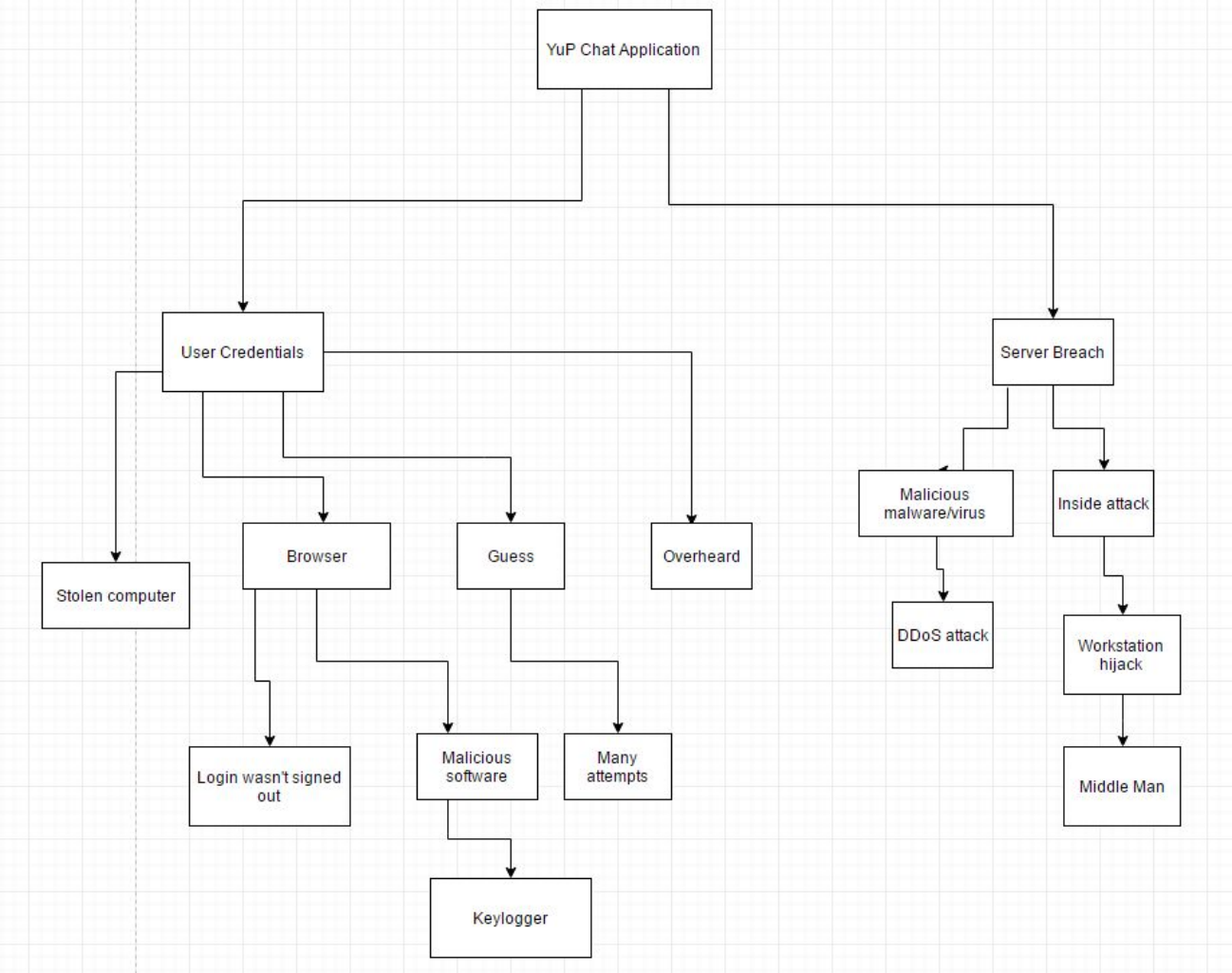

Figure 1.1 Attack Tree (V.1)

5 Possible Vulnerabilities

5.1 OpenSSL Vulnerability

While using OpenSSL as a public and private key encryption, a possible vulnerability is that "man-in-the-middle" attack (MITM). MITM is when an adversary pretends to be a user. For example in YuP case, MITM occurs when an adversary pretends to be user 1 and

communicate with user 2 or vice versa. To do this the adversaries could pretend to be the server and be sending out fake message with their fake keys.

5.2 Eavesdropper

Eavesdropper can see all the conversations between the server and the client.

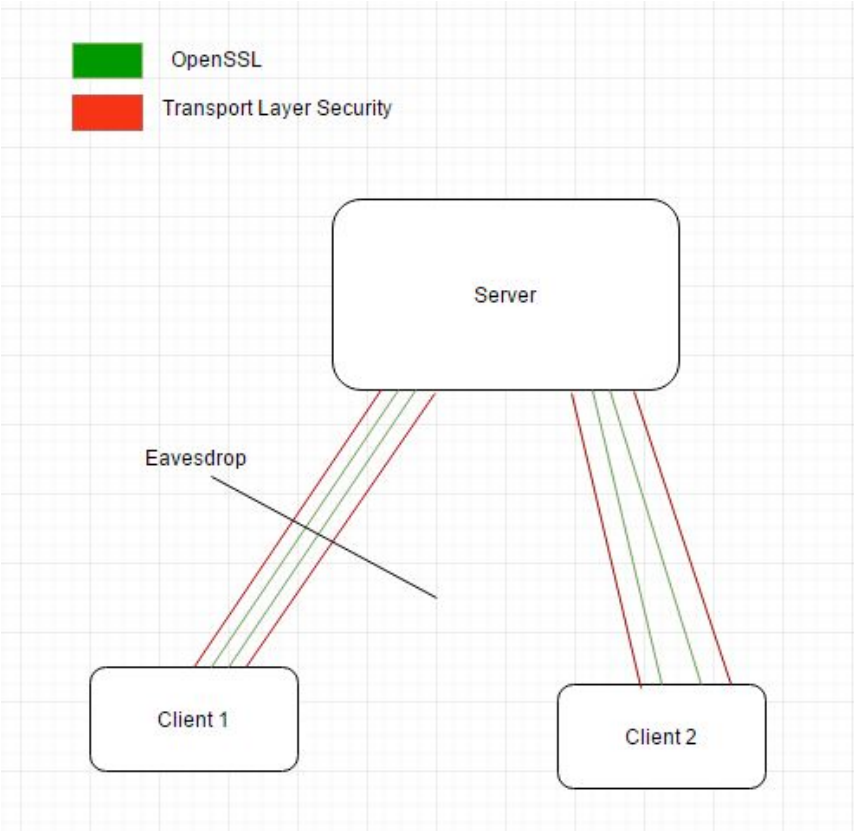

Figure 1.1 TLS and OpenSSL

6 Possible Related Previous Works

For this application, related previous works are many of the well-known messenger applications such as:

1. Facebook Messenger
2. Viber
3. WhatsApp

7 Solution

5. 1 OpenSSL Vulnerability Solution

In YuP we plan to assign a special ID to every users when they sign up. That special ID is then used as a public key and it has some unique information about the identity of the user – like the user email address. We plan to have clients acquire their friends public key right after they added one another on the buddy list, so the MITM can't fake the public key exchange. This is call identity-based encryption (IBE). Since MITM takes place in the key exchange phase of PGP, we can apply IBE where the public key is the recipient's ID.

5.2 Eavesdropper Solution

Transport Layer Security (TLS) 1.2 will be use. TLS 1.2 will create a secure tunnel from the client to the server, so no eavesdropper could attain data during the exchanging of the keys phase. In addition, there will be an additional usage of Let's Encrypt, which will further secure the tunnel. Let's Encrypt will introduce the usage of certificates that are being use for authentication. Certificates are being assigned when the server or user first contact one another. So for example, when the user first contacts the server, Let's Encrypt uses the public key to identify the user and then assign the user with a new key pair. So when the user wants to contact the server again, the user must have they key to be allow to pass through to the server – vice versa server to user.

8 Full Rigorous Analysis

For this application, Pretty Good Privacy (PGP), will be achieved with the signing, encryption, and decryption of the message being rallied between the users therefore improving the security of the system.

In this YuP chat application, we will build a RESTful server using Java. RESTful server allows a simple client to server interaction. By having a RESTful server, we have the opportunity to implement the HTTP such as PUT, GET, POST, and DELETE. PUT will be used for a way for users to be able to send out their messages. GET will be used for the users to acquire messages that were sent to them by friends in their buddy list only. GET would also be used to acquire the users' friends, where the users have the capability to search up their friend on the server. POST could be used to allow users the option to update certain information, such as their public user name or even their profile picture.

DELETE would allow users the option to delete certain information and even their profile picture.

TLS 1.2 will be implemented with OpenSSL in YuP to allow for a secure tunnel that will protect against eavesdropper. In addition to those two, project developers will also use Let's Encrypt, which will further secure our tunnels with the implementation of certifications.

With MITM attacks, there have been serious cases, such as with Facebook Messenger. For Facebook Messenger, they had problems with hackers having access to users' messages and then replacing links and file uploads with malware. Their way of solving this was by assigning unique identification numbers to every message between sent between users. As Facebook scans the messages, they look out for any duplicate keys. When seeing two identical messages, they will be notified of a message being suspiciously copied and then act to have it halted. To apply this to the YuP messenger application, similar provisions will be done to the messages of our system as well with the inclusion of TLS connections between users.

9 Citation Links

http://www.bostoncommons.net/facebook-messaging-vulnerabilities/?doing_wp_cron=1474644302.9796149730682373046875.

http://www.thewindowsclub.com/man-in-the-middle-attack

https://letsencrypt.org/how-it-works/