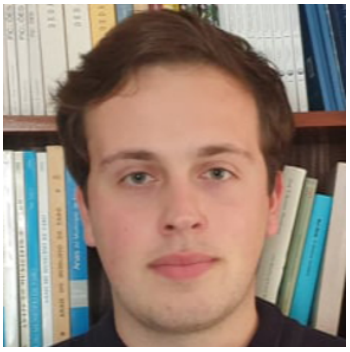
Smartphone as a security token - Group A14

Network and Computer Security 2021/2022



Bernardo Quinteiro
93692

Diogo Lopes
93700

Jérémy Breton
101360

1. Problem

The topic we've chosen for this project is "Smartphone as a security token". As a result of that, we've decided that we'll be developing a virtual wallet with two internal servers. The first one will be responsible for the critical system of the wallet, such as the login phase and checking the card's balance. The other server will be responsible for managing the communication with the exterior (web application and the smartphone) containing a firewall to prevent attacks.

The main security issue we need to address is the possibility of an unwanted entity gaining access to the critical server of the wallet. We must fully guarantee that attackers can't gain access to this system, by any means. Other than that, the information that is passed between the internal servers and the external entities needs to be confidential and keep its integrity, so these connections have to be secure.

1.1. Solution Requirements

R1: Confidentiality: The data passed between the server and the client, in the web application and in the smartphone, must be encrypted.

R2: Integrity: The data passed between the server and the client, in the web application and in the smartphone, can't be tampered with and should be preserved.

R3: Authentication: To connect to the virtual wallet server the user must perform two-factor authentication, a password and a token generated on the smartphone.

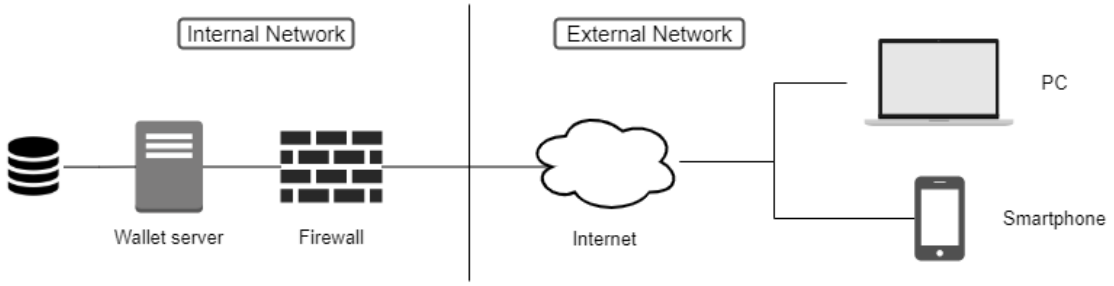
R4: Non-repudiation: The critical data passed between the server and the client, in the web application and in the smartphone, must be signed.

1.2. Trust Assumptions

In regard to the trust between entities of this system, if the client follows the protocol, our server must partially trust him in communications with the smartphone and with the web application.

2. Proposed Solution

2.1. Overview



This image represents a diagram of the pretended solution that consists of three elements. Firstly, we have a server that handles the operations, such as login and checking the card's balance. We also have a client that wants to do these operations and connects to the server via a web application and, finally, a smartphone that acts as a security token.

In order to be able to login to the web application, the client needs to perform a two-factor authentication, which could be the password to login and a randomly generated code that would only be accessible through the smartphone as a security manner. After logging in, the client has access to his stored cards but to be able to check each one's balance there will be the need to again grant authorization through the smartphone.

2.2. Deployment

We will deploy the server in a virtual machine and this will be responsible for serving the web application and performing the operations.

We will also deploy two virtual machines, one will be a smartphone emulator responsible for the token, and the other one will be the computer.

2.3. Secure channel(s) to configure

Our server will have:

- An OpenSSL generated self-signed certificate;
- Duo SDK to ensure the two factor authentication.
- Java Crypto library to ensure all cryptographic operations in the smartphone;
- Pycrypto library to make all cryptographic operations in the server possible;

2.4. Secure protocol(s) to develop

Who will communicate?

Analyzing the above diagram, we will have a server and a client communicating and we will use the Diffie-Hellman key exchange that is a method for safely developing and exchanging keys over an insecure channel.

Which security properties will be protected?

In regard to the security properties, we will be able to obtain confidentiality, integrity and authenticity. Confidentiality because nobody else can read the plain text. Authenticity because we use tags in order to check if the text has been changed, and therefore Integrity.

The token exchanged between the smartphone and the computer will be secure, and also the communications between the website and the client.

The use of nonces makes it impossible for hackers to impersonate the legitimate parties.

What keys will exist and how will they be distributed?

We will be using RSA2048 keys since we need safety and asymmetric encryption. Firstly the server and the client will generate pair keys. These last ones will be then exchanged. We will also use TLS and AES.

3. Considered technologies

At this point in time, we intend to create our server and web application using Python, the database using SQLite and the mobile app using Java.

4. Plan

4.1. Milestones

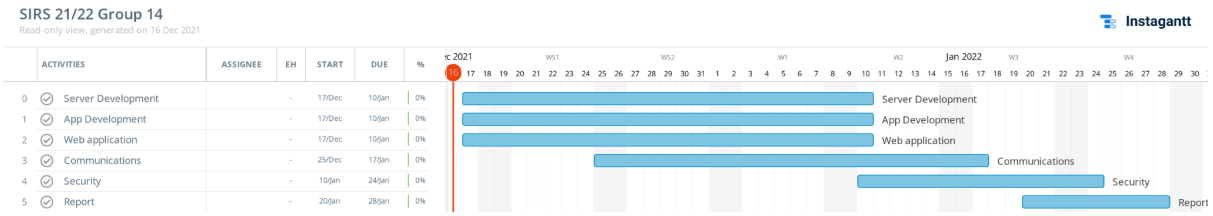
We divided the project in three versions, basic, intermediate and advanced stages. In the **basic version** we want to have the server communicating securely with the web application using the decided protocol. Afterwards, in the **intermediate version** we will be implementing the two-factor authentication and authorization via smartphone. Finally, in regard to the **advanced version** we will be setting up a firewall with DDoS mitigation.

4.2. Effort commitments

As for the organization of tasks among the group, we have identified 3 key stages to the development of this project:

- For the first delivery (10th of January), we intend to have the Server, App and Web App fully developed;
- For the second delivery (17th of January), we want to have fully functional communications protocols between these;
- By the end of the project(24th of January), we'll need to have our security protocols fully implemented, with the report being delivered four days later.

This organization is summarized in the image below:



For the first delivery, we'll divide the development of the Server, App and Web App between Bernardo, Jeremy and Diogo, respectively. As for the following stages, we'll be working together most of the time on the implementation of features. Therefore, we do not believe we should be dividing work among us. Each of us expect to have around 7 days of work each, by the end of the project, being 3 of those in the initial stage and the other four for the two final deliveries.

5. References

- Git overview
- Git topics
- Python
- Pycryptodome
- OpenSSL
- Duo SDK
- Java
- SQLite