Universidade de Aveiro
Mestrado em Cibersegurança
Código: 41783 - Identification, Authentication and Authorization
Responsible: João Paulo Silva Barraca

Flexible, Risk Aware Authentication System

Friday 22nd March, 2024

Ricardo Covelo (102668) - Telmo Sauce (104428)

Contents

1 Introduction

In this project, our objective was to apply our classroom-acquired knowledge to develop an Identity Provider (IdP) supporting services with different criticalities, implementing multi-factor authentication methods, and addressing the issue of MFA fatigue.

2 Applications

As was asked we will implement 3 different applications with different security levels and different clearance roles, all will use similar applicational code, with the also shared IDP and Resources servers.

- **Cooking Fórum (Less Secure, Biba Model)**
  This application will be a fórum for sharing recipes. This application will have 4 different roles (*Looker, Eater, Cook, Admin*). Both Looker and Eater Roles **can't make any posts**, the Cooker can make **recipe posts**, and the admins can make **announcements/Rules**. The admins can't see recipes so as not to get distracted.

- **Religious Fórum (Secure, Biba Model)**
  This application will be a Religious fórum with 4 different roles *(Believer, Priest, Bishop, and Pope)* that correspond to 4 different levels of integrity of posts *(Comment, Recommendation, Teaching, Religious law)* each role can post with a tag corresponding with the role or lower. Everyone can see every post with an integrity tag corresponding to, or above their own.

- **Military News(Very Secure, Bell-La Model)**
  "g This application will be a fórum for military-related news and so, methods for stopping information leakage will be put in place. There will be 4 different roles *(Private, Corporal, Sergeant, Major)* and correspondent clearance tags *(Unclassified, Restricted, Confidential, Top-Secret)*. Each individual can write a post with a clearance tag one level above their own and only access the ones with clearance the same level or below theirs

3 Architecture and Flow

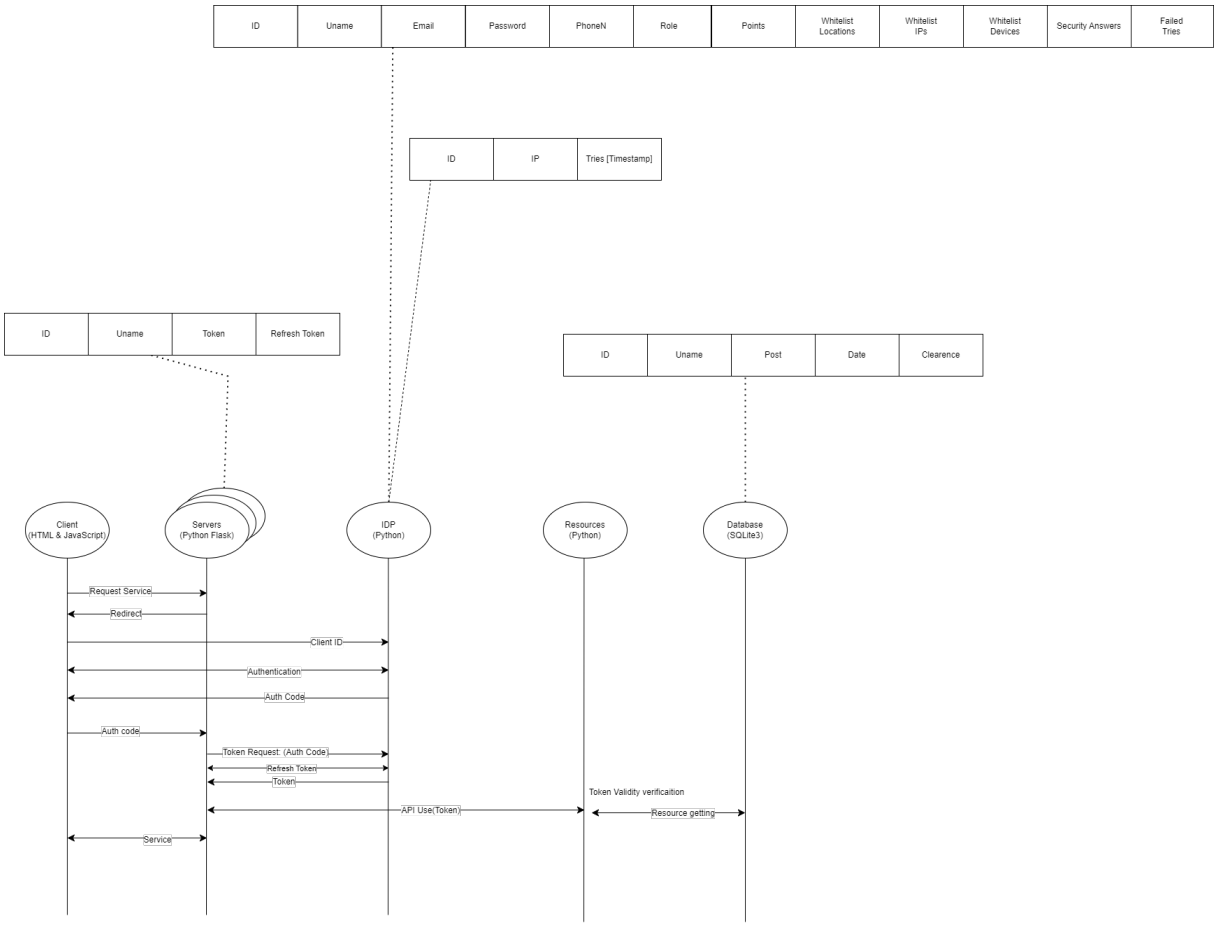

Figure 1: Architecture

3.1 Tokens

We will use session and refresh Tokens as seen in the diagram above. The refresh tokens will have a relatively long TTL(Time to Live). This token will be used by the client app every time the session token, used by the application to communicate with the API and with a short TTL, is about to expire and the application requests a new token so it can still use the API. This approach enables the negation of further tokens making it impossible to use the stolen token for unauthorized use of the API.

3.2 Databases

In this project, we will need:

- 3 databases for the servers *ID, Uname, Token, Refresh Token*, one for each service, this database is needed so each service can associate each user with the correspondent Token and Refresh token.

- 3 databases for the IDP(2 tables each)
  - 1st table *ID, Uname, Email, Password, PhoneN, Role, Points, Whitelist location, Whitelist IPs, Whitelist Devices, Failed tries* This table has all the information that will help retrieve the information to decide how many authentication factors will be asked, and some information to help enforce them.
  - 2nd table*ID, IP, Tries* This table will have the information to help to prevent Password Spraying.
- 3 databases for the resources one for each service,*Id, Uname, Post, Date, Clearance* This database will have all the data needed for the correct utilization of the website.

4 Authentication methods

Depending on the application and the score points of the account, the IDP can ask, for up to 4 different authentication methods.

- **Login**
  The user will be prompted to enter their User-Name and Password. If the entered credentials match the ones in the database, the user will have successfully logged in.
- **Email OTP**
  This method sends a unique code to the user's registered email address. The user needs to enter this code for final verification. Libraries like "smtplib" will be used in the implementation.
- **SMS OTP**
  This method is similar to the previous one with the only difference being the fact that the OTP Is sent through SMS. This factor will be implemented with "twilio"
- **Security question**
  This is a knowledge-based approach where pre-defined security questions are presented to the user during login. The user must answer these questions correctly for additional verification.

5 Risk Factors

A Scoring method will be implemented, which will increase depending on various factors such as location, IP, and others.

The score will be **persistent** which means the user will not lose the score after each login, however, he will lose 10 points each day, until 0 is reached.

Some of the Factors that will be explained in more detail will be added to an **allowList** in the database. It's important to note that elements added to this list 6 months prior will be **removed**, and when these elements are utilized, their timestamp will be **updated** with each use.

Failed Login Attempts (20 Points per attempt)

This implementation **avoids brute forces** on a password, to gain access to an unauthorized account. The more a user enters a wrong password, the more it will increase by 20 points on its overall score.

It was taken into consideration having a score for the **combination** between IP and the account accessed, this was to avoid a user needing to go through all the authentication processes after being targeted by a brute force attack, however, this possibility was **discarded** since an attacker can change its IP, making this option obsolete.

New Device (200 Points)

By implementing this risk factor, the possibility of falling victim to phishing attacks or having sensitive information leaked from other companies can be significantly reduced. This is because users often use the same login credentials across multiple services, which can make them vulnerable to security breaches. With this additional layer of authentication provided by the MFA, even if an attacker obtains valid credentials, they would need to successfully pass another authentication step to gain access to the account or compromise the user's device.

New IP (20 Points)

While implementing this risk factor can help mitigate some of the issues mentioned earlier (5). We also took into account the users' workflow. We gave it a low score because users often change locations and switch between different WiFi networks.

New Schedule (10 Points)

This factor may suggest that the user's account has been compromised. If a user uses the application at a time that does not align with the normal flow of people in the same area, it is another indication of the account being compromised.

Access with a new IP and unusual time may mean that the user's device has been stolen or compromised, hence the combination of the two requires an additional authentication method in the average application.

New Location (60 Points)

This factor can indicate an account being compromised, however, this can also be a false alarm since he can be only going on a trip, this is the reason the score is medium to not disturb

the user on a false alarm.

Setting Tempering (200 Points)

This factor will only be implemented in The Services of Medium and High Risk, and will at least activate one of the authentication methods since users can't ever modify any password of sensitive information without an MFA method to verify them.

Password Spraying (10 Points per attempt)

This measure is implemented to deter attackers from attempting to access multiple accounts from the same IP address. The scoring system for this factor operates somewhat differently from the previous factors, as the score is assigned to each IP address.

If the user attempts to log in from an IP that failed various times the Login, their score will temporarily increase by the points associated with that IP address.

We will only retain up to 10 attempts for each IP, accumulating to a maximum of 100 points. These attempts will be deleted after 2 weeks.

5.1 Scoring State Machine

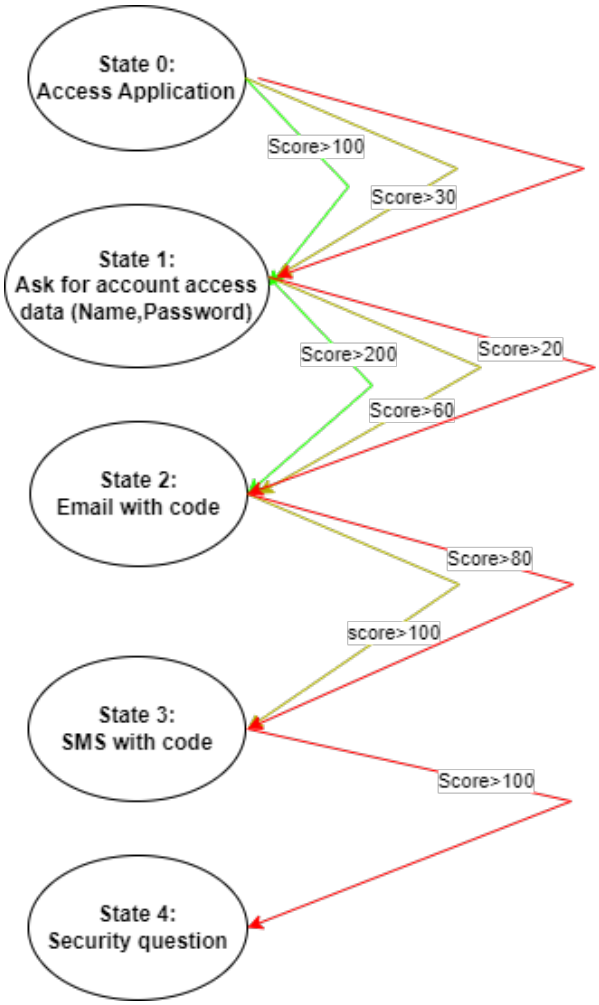

Figure 2: Red - Very Secure, Yellow - Secure, Green - Less Secure

6 Conclusion

In conclusion, we believe this authentication flow implementation is robust and capable of mitigating common attacks that target such systems. By prioritizing both security and user experience, we believe our approach not only enhances overall security but also reduces the likelihood of user frustration with the authentication system.