Secure User Authentication System

(Internship Task 3 – Internee.pk)

1. Introduction

User authentication is a crucial component of modern apps. Because of the increase in cyber threats, using usernames and passwords alone is no longer sufficient. By using weak passwords, phishing scams, or hacked credentials, attackers can gain unauthorized access to user accounts.

To lessen these risks, organizations employ multi-layered authentication and strong data protection techniques. This assignment focuses on improving user authentication and safeguarding sensitive user data using Two-Factor Authentication (2FA), OAuth 2.0, and AES-256 encryption. The task was completed as part of the cybersecurity internship offered by Internee.pk.

2. Objective of the Task

The main objective of this task is to enhance authentication security and protect sensitive user data. The specific goals include:

- Implementing Two-Factor Authentication (2FA) to add an extra layer of login security.
- Using OAuth 2.0 to enable secure and password less third-party authentication
- Encrypting sensitive data using AES-256 encryption
- Safely testing authentication mechanisms using fake user data instead of real user information

This task helps in understanding how real-world applications secure user identities and prevent unauthorized access.

3. Overview of Secure Authentication

Secure authentication ensures that only authorized users can access a system. A strong authentication system usually includes:

- Something the user knows (password)
- Something the user has (OTP, mobile device)
- Secure handling of credentials and tokens

In this task, authentication security was strengthened by combining 2FA, OAuth 2.0, and encryption techniques.

4. Fake User Data Generation (Mockaroo)

To avoid using real user data, fake user datasets were generated using **Mockaroo**. Mockaroo is a data generation tool that creates realistic but synthetic data for testing purposes.

The generated dataset included:

- Usernames
- Email addresses
- Passwords (dummy values)
- User roles

Using fake data ensures privacy and allows safe testing of authentication features such as 2FA, OAuth login, and encryption without exposing sensitive information.

5. Two-Factor Authentication (2FA)

Two-Factor Authentication (2FA) adds an additional verification step during login. In this task, 2FA was implemented using **Google Authenticator**.

How 2FA Works:

1. The user enters a username and password.
2. The system requests a one-time password (OTP)
3. The OTP is generated by Google Authenticator on the user’s mobile device.
4. Access is granted only if the OTP is correct.

This approach significantly improves security because even if a password is compromised, an attacker cannot log in without access to the user’s authenticator device.

---

6. OAuth 2.0 Authentication

OAuth 2.0 is a secure authorization system that allows users to log in using trustworthy third-party services like Google or GitHub.
In this work, OAuth 2.0 was investigated using a demo authentication flow. Instead of exchanging credentials with the program, users authenticate directly with the supplier. The application receives an access token, which is used to grant restricted and temporary access.

Benefits of OAuth 2.0:

- No password sharing with third-party applications.
- Improved security and user experience

- Widely used in modern web and mobile applications

7. AES-256 Encryption

Sensitive user data must be protected both during transmission and storage. For this, AES-256 encryption was used.
The potent symmetric encryption algorithm AES-256 converts readable data (plain text) into unreadable cipher text using a 256-bit secret key. Only authorized systems with the appropriate key can decode the data.
In order to show how encryption protects data even when it is accessed by unauthorized individuals, example user data was encrypted in this work using AES-256 encryption.

8. Security Benefits of the Implemented System

The combined use of 2FA, OAuth 2.0, and AES-256 encryption provides multiple security benefits:

- Strong protection against password-based attacks
- Reduced risk of unauthorized access
- Secure handling of authentication credentials and tokens
- Protection of sensitive user data even in case of data leakage

This layered approach reflects best practices used in real-world applications.

9. Tools and Technologies Used

- Google Authenticator (Two-Factor Authentication)
- OAuth 2.0 (Secure authentication framework)
- AES-256 Encryption (Data protection)
- Mockaroo (Fake user data generation)
- GitHub (Documentation and version control)

10. Conclusion

I gained hands-on experience implementing secure user authentication systems thanks to this assignment. OAuth 2.0, Two-Factor Authentication, and AES-256 encryption were used to show a reliable and secure authentication solution.
The work facilitates understanding of how modern programs safeguard user accounts and sensitive data. All things considered, our study enhanced practical understanding of authentication security and illustrated its importance in actual cybersecurity contexts.