

TWo-factor authentıcatıon

Implementation and Testing

A grey logo on a black background

Description automatically generated

December 25, 2024

salih doganer

B201202015

Overview

This is a web application using Flask that provides support for two-factor authentication . It provides a web application where the user is forced to secure access with both a password and one-time password (OTP), then redirects to the dashboard displaying sensitive, randomly generated data upon successful verification.

A screenshot of a cell phone

Description automatically generated

What is 2fa?

2 factor authentication is the process of providing 2 of the 3 form of verification factor required for authentication before accessing the system or application.

This project implements these 2 factors:

1. What You Know: A username and password for initial authentication.

2. What You Have: A one-time password (OTP) generated and verified during the login process.

Features

1. Username and Password Authentication

2. One-Time Password (OTP) Verification

3. Secure Session Management

4. Sensitive Data Display

5. Logout (End Session) Functionality

Requiriments

- Python 4+

- Flask

- Pyotp

To install these;

pip install flask pyotp

Default Credentials

- Username: user

- Password: password

These can be updated in the app.py file:

valid\_username = "user"

valid\_password = "password"

Application Workflow

1. Login: Users authenticate with their username and password.

2. OTP Verification: A one-time password is generated and displayed in the **console** for testing purposes. Users must enter the correct OTP to proceed.

3. Dashboard: On successful OTP verification, the user is granted access to the dashboard.

4. Logout: Users can log out via the Logout button on the dashboard, clearing their session.

Security Notes

- Session Management: Flask sessions are secured with a secret key. Update app.secret\_key to a strong, unique value.

- Sensitive Data: The sensitive data displayed on the dashboard is randomly generated and for demonstration purposes only.

Testing Procedures

The developed 2FA application has gone through to a testing process to ensure functionality and security. The key test procedures included:

1.Functionality Testing

- Verifying that the application correctly validates username and password combinations.

A screenshot of a login screen

Description automatically generated

- Ensuring the OTP generation and validation processes work easily.

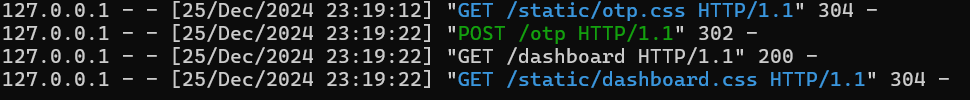
A screen shot of a computer screen

Description automatically generated

A screenshot of a phone

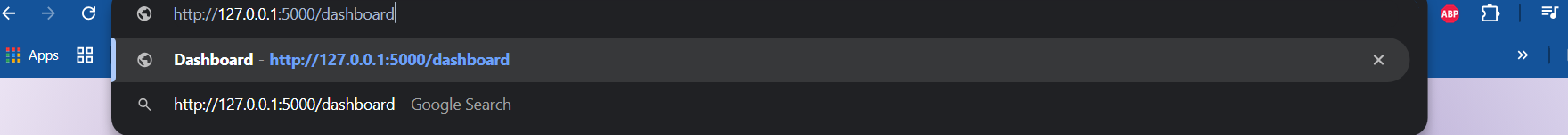
Description automatically generated

- Confirming that only users with both valid credentials and OTP can access the sensitive information.



2.Security Testing

- Attempting to bypass the OTP requirement to verify resistance to replay attacks and testing for vulnerabilities in session management to prevent unauthorized dashboard access via URL manipulation.



A screenshot of a login screen

Description automatically generated

Test Result Analysis

All functionalities performed as expected. Users could log in only with valid credentials and a correct OTP. Invalid attempts were correctly redirected with appropriate error messages. The application successfully resisted unauthorized access attempts. Session-based authentication was confirmed to prevent URL-based exploitation.

Project Results

The MFA application successfully combined What You Know (username and password) with What You Have (OTP) to provide secure access control. Testing validated its effectiveness in protecting sensitive data and ensuring usability.

Salih Doğaner

B201202015