Report on Ethical Phishing Simulation Platform

Introduction

This project involves developing a Phishing Simulation Platform that mimics real-world phishing attacks to raise awareness among users and employees. It aims to enhance cybersecurity readiness by demonstrating how easily users might fall for phishing attempts and by educating them on recognizing red flags.

Abstract

Phishing remains one of the most common and dangerous social engineering attacks. This lightweight simulation app replicates phishing techniques in a safe and controlled environment using Flask, Mailtrap, and SQLite. It allows administrators to send fake password reset emails and monitor interactions through an admin panel. Once users interact with the fake login page, they are shown an educational awareness message, reinforcing security best practices.

Tools & Technologies Used

- Frontend: HTML, CSS (Glassmorphism design), Bootstrap

- Backend: Python, Flask

- Database: SQLite

- Email Service: Mailtrap.io (for secure simulation)

- Others: GitHub for version control, .env for config security

Steps Involved in Building the Project

1. Designed UI components including login form, simulation launcher, awareness page, and admin dashboard.

2. Set up Flask routes and logic to manage user input, email sending, and database operations.

3. Integrated Mailtrap SMTP service for secure email delivery during testing.

4. Created .env configuration for secret variables and SMTP credentials.

5. Built a logging and analytics system to track user interactions in a SQLite database.

6. Implemented awareness and red-flag detection tips after users fell for simulated attacks.

7. Deployed styling with Glassmorphism for a modern look.

Security & Testing Considerations

To ensure data safety and ethical simulation, the app is designed to never store real passwords or send actual phishing emails. Instead, it uses Mailtrap—a safe email testing platform that captures simulated messages in a sandbox environment. Environment variables are used for sensitive credentials to prevent accidental leaks, and only non-sensitive user actions are logged. All testing was conducted in a controlled local environment without involving real users.

Conclusion

This project demonstrated how phishing simulations can be a practical tool in security awareness training. It reinforced key concepts in cybersecurity, Flask development, and ethical testing practices. The platform is lightweight, secure, and visually engaging—making it ideal for internal organizational use and training.