Report – Ethical Phishing Simulation Platform

1. Introduction

With phishing attacks among the most common and damaging threats to organizations and individuals, security awareness and education are vital in reducing risk. Realistic simulation of phishing scenarios, combined with targeted user education, is recognized as a best practice for improving resilience against social engineering attacks. The Ethical Phishing Simulation Platform was developed to provide a safe, controlled environment for simulating phishing campaigns and analysing user behaviour, enabling administrators to assess vulnerabilities and deliver impactful security training.

2. Abstract

The Ethical Phishing Simulation Platform is a web-based solution that facilitates the end-to-end process of phishing awareness training. Administrators can design and launch simulated phishing campaigns, track user engagement, and provide post-campaign education—all within a secure lab environment. The platform supports customizable phishing templates, safe email delivery, detailed analytics dashboards, and automated feedback to participants. Data is stored in an SQLite database, and the app leverages Python, Flask, and modern web technologies for robustness and usability. By offering actionable insights and educational interventions, the platform empowers organizations to proactively defend against phishing risks.

3. Tools Used

- Python 3: Primary programming language for backend logic and automation.
- Flask: Lightweight web framework for server-side routing, templating, and session management.
- SQLite: Embedded database for storing campaign data, results, and user interactions.
- Sendmail/Postfix (SMTP): For sending simulated phishing emails within a controlled lab.
- HTML5/CSS3: Structure and styling of web pages for admin and user interfaces.
- Jinja2: Template engine used with Flask for dynamic HTML rendering.

4. Steps Involved in Building the Project

4.1. Project Planning & Architecture

- Defined requirements for campaign management, email simulation, user tracking, analytics, and education modules.
- Designed a modular structure, separating static assets, templates, backend logic, and phishing email templates.

4.2. Backend Implementation

- Set up Flask application with clear routing for dashboard, campaign creation, analytics, education, and authentication.
- Integrated SQLite database for persistent storage of campaigns, user actions, and analytics.
- Developed models and utility functions for campaign lifecycle, template handling, and secure data interactions.

4.3. Phishing Simulation & Tracking

- Enabled administrators to create and manage campaigns, select or customize phishing templates, and define recipients.
- Configured SMTP (Sendmail/Postfix) for safe email delivery in a lab environment, ensuring no real-world harm.
- Embedded tracking mechanisms in phishing links to log user actions (clicks, form submissions, timestamps).

4.4. Analytics & Education

- Developed dashboard views to display campaign metrics: open rates, click rates, success rates, and timelines.
- Automated post-campaign feedback, educating users on recognizing phishing and following security best practices.

4.5. Frontend Development

- Designed a clear, responsive interface using HTML/CSS for both admins and test users.
- Implemented navigation, forms, and analytics visualization for a seamless user experience.

4.6. Testing & Documentation

- Conducted end-to-end testing of campaign creation, email delivery, user tracking, and analytics features.
- Documented setup, usage, and best practices within README.md and the included project report.

5. Conclusion

The Ethical Phishing Simulation Platform provides a practical, safe, and effective solution for security awareness training. By simulating realistic phishing attacks and analysing user behaviour, the platform enables organizations and educators to identify vulnerabilities and deliver targeted training interventions. The integration of customizable templates, robust tracking, and actionable analytics creates a comprehensive tool for boosting organizational resilience against phishing threats. The project demonstrates the power of open-source tools—Python, Flask, SQLite, and modern web standards—in building impactful cybersecurity education solutions.

6. References

- Phishing Simulation Best Practices – SANS Whitepaper. https://www.sans.org/white-papers/38547
- Flask Documentation. https://flask.palletsprojects.com
- OWASP Phishing Guide. https://owasp.org/www-community/Phishing
- SQLite Documentation. https://www.sqlite.org/docs.html
- Security Awareness and Phishing Simulations – NIST. https://csrc.nist.gov/publications/detail/sp/800-50/final

---