Ethical Phishing Simulation Platform

-Sai Aditya

Introduction

This report details the development and implementation of an Ethical Phishing Simulation Platform. The project, chosen from the Elevate Labs internship list, focuses on building a web-based tool to simulate phishing campaigns for educational purposes. As a cybersecurity aspirant with experience in ethical hacking, this project was a practical application of my knowledge in social engineering, web development, and network security. The platform's objective is to demonstrate the mechanics of a phishing attack in a controlled environment, allowing for the safe capture and analysis of user interaction to better understand and defend against such threats.

Abstract

The Ethical Phishing Simulation Platform is a web application built with Python and the Flask framework. The platform provides a user-friendly interface to send customized phishing emails to a list of approved test users. It leverages an SMTP server (Mailtrap) to handle email delivery and captures user responses, such as link clicks and credential submissions, in an SQLite database. A key feature of the platform is its analytics dashboard, which visualizes the results of the simulation, including email delivery logs and captured credentials. This project serves as a proof-of-concept for how organizations can conduct non-malicious simulations to measure and improve employee cybersecurity awareness.

Tools Used

- Python: The core programming language used to build the application logic.
- Flask: A micro-web framework for Python, used to create the web interface and manage all application routes.
- HTML/CSS: Used to design the templates for the simulated phishing email and the fake login page to ensure they are realistic and convincing.
- SQLite: A lightweight, file-based database used to securely store captured credentials and email delivery logs.
- Mailtrap: A safe, fake SMTP server used to capture and analyze outgoing emails without sending them to real users, ensuring the simulation remains ethical and contained.
- Python-dotenv: A library to manage environment variables, allowing for the secure storage of sensitive data like Mailtrap credentials.

Steps Involved in Building the Project

1. Environment Setup: A dedicated project directory was created with a Python virtual environment to manage dependencies. The required libraries, including Flask and python-dotenv, were installed.
2. SMTP Configuration: An account was created on Mailtrap to serve as a fake SMTP server. The provided SMTP credentials (Host, Port, Username, and API token) were securely stored in a .env file to prevent hardcoding sensitive information.
3. Core Application Logic: The application was built using Flask, with separate functions for handling the homepage, email sending, credential capture, and the analytics dashboard. The code was structured into app.py and database.py for modularity.
4. Database Integration: An SQLite database was integrated to manage data. The database.py script was created to handle the creation of two tables: one for captured credentials and another for logging email delivery status.
5. Template Design: Realistic HTML templates were designed for the phishing email (email_template.html) and the credential harvesting landing page (landing_page.html). The email template was designed to mimic a security alert from a well-known service, linking to the fake login page.
6. Functionality Testing: The application was tested in a controlled environment by sending a simulated email to a test address. The process was meticulously debugged, with issues ranging from TemplateNotFound errors to Authentication failed SMTP errors being resolved. The final successful test proved that the platform could send emails, capture credentials, and log results.
7. Dashboard Development: A dashboard page (dashboard.html) was created to retrieve and display data from the SQLite database. This dashboard shows a table of all captured credentials and a log of all email sending attempts, providing a clear visual summary of the simulation's results.

Conclusion

The Ethical Phishing Simulation Platform is a successful project that effectively demonstrates key cybersecurity concepts. It not only showcases my technical skills in web development and scripting but also my understanding of ethical hacking principles and responsible security practices. The project provides a tangible, working example of how security professionals can proactively test and train for cyber threats. The insights gained from building and debugging this application, particularly around email protocols and secure data handling, have been invaluable for my professional development as a cybersecurity aspirant.