Secure Software Design & Engineering(CY-321)

Project Proposal

Anti-Phishing Browser Extension

Group Members

1. Shameer Awais (2022428)
2. Rooshan Riaz (2022506)
3. Naqi Raza (2022574)
4. M. Yasir (2022455)

Submission Date: 07/03/2025

Ghulam Ishaq Khan Institute of Engineering Sciences and Technology

Introduction

Phishing attacks are one of the most common and dangerous cybersecurity threats, especially for non-technical users. These attacks often lead to financial losses, identity theft, and data breaches. To address this issue, we propose developing an **Anti-Phishing Browser Extension** that detects and blocks phishing websites in real-time. This extension will leverage machine learning and secure software design principles to protect users from malicious websites.

Problem Statement

Phishing attacks are becoming increasingly sophisticated, and many users, especially those without technical expertise, fall victim to these attacks. Existing solutions often rely on static blacklists, which are not effective against new or evolving phishing sites. There is a need for a dynamic, real-time solution that can identify and block phishing websites before they cause harm.

Solution

We will develop a browser extension that:

- Uses machine learning to detect phishing patterns in real-time.
- Scans URLs as users browse the web and alerts them if a phishing site is detected.
- Encrypts user data to ensure privacy and security.
- Provides a user-friendly interface for managing settings and viewing alerts.

Security Requirements

To ensure the security and effectiveness of the extension, we will implement the following security measures:

- **No Tracking of Browsing History:** The extension should collect only necessary data, such as URLs and site content for phishing detection purposes. Sensitive user information (e.g., login credentials, personal data) should never be collected or stored.
- **User Consent:** Ensure that users explicitly agree to the collection and processing of data for phishing detection purposes, with the option to opt out of any data sharing features..
- **Secure Communication:** All communication between the extension and backend servers (if applicable) must use **HTTPS (SSL/TLS encryption)** to prevent man-in-the-middle attacks (MITM) and eavesdropping..
- **No Injection of Malicious Scripts:** Ensure that the extension does not inject any malicious code (e.g., JavaScript) into webpages or compromise the integrity of a webpage's content.

- **User Customization:** Users should be able to toggle certain features on/off (e.g., phishing site blocking, user feedback features), and control how data is used (e.g., whether they wish to contribute data to improve the extension).

Security Planning

1. **Threat Modeling & Risk Assessment:** Analyze and identify potential attack vectors against the extension, such as

   - Phishing sites bypassing detection algorithms.
   - Data leakage of user information (e.g., browsing history, URL information).
   - Reverse engineering of the extension.

   Identify mitigation strategies for each identified threat such as using secure and trusted third-party services for phishing database updates, and implementing obfuscation and signing for extension code to prevent tampering.

2. **System Architecture & Secure Design:**

   - High-level diagram showing user interactions, phishing detection processes, and communication with the backend.
   - Component-level diagram for browser extension features (e.g., phishing detection, alerts, communication with database).
   - Secure session management if the extension includes any form of user authentication or personalization.

3. **Secure Coding & Initial Implementation:**

   - Ensure that any sensitive data (such as user credentials or preferences) is stored securely, either locally in the browser or in a secure database.
   - Implement the phishing detection mechanism, including both URL analysis and heuristic checking.

4. **Security Testing & Vulnerability Analysis:**

   - Review the source code for potential vulnerabilities using tools like SonarQube.
   - Simulate attacks to test the robustness of the extension.
   - Identify any vulnerabilities discovered during the testing phase such as missing input sanitization, and flaws in the phishing detection mechanism.

5. **Final Implementation & Secure Code Review:**

   - Implementing any necessary security fixes based on the findings from the security testing phase.
   - Ensuring that the extension's code is optimized for performance and security.
   - Performing a thorough code review to ensure that all security best practices are followed, there are no vulnerabilities or security holes, and all input and output are validated or sanitized.

Conclusion

The A**nti-Phishing Browser Extension** will provide a robust and user-friendly solution to protect users from phishing attacks. By leveraging machine learning and secure software design principles, this project aims to reduce the success rate of phishing attacks and enhance online security for all users.