CSG3101 Applied Project
Project Proposal Form (Semester 2, 2024)

Project Title: Phishing Detection Browser Extension

*** Discipline Alignment (please select more than one for cross-disciplinary project):**

- *Cyber security*

*** Project Summary:**

- *Project information:*

This project aims to develop a Chrome browser extension for real-time phishing and malware detection. The extension monitors visited URLs, analyzes them using APIs (Google Safe Browsing, VirusTotal), and employs heuristic techniques to detect suspicious patterns. Key features include real-time URL monitoring, phishing detection, customizable alerts, whitelist/blacklist management, email phishing analysis, and basic malware detection. Advanced options like sandbox file analysis and adjustable sensitivity settings enhance user control. The extension provides a secure browsing experience by promptly alerting users to potential threats, helping safeguard sensitive information and promoting safer online practices.

- *Project type:* Development

*** Scope of work:**

This project involves developing a Chrome browser extension for real-time phishing and malware detection. The extension will monitor URLs, detect suspicious activity using APIs (Google Safe Browsing, VirusTotal) and heuristic techniques, and provide customizable user alerts. Advanced features include email phishing analysis, basic malware detection, and sandbox file analysis. The goal is to enhance online security and protect users from phishing attacks and malicious content.

Requirements

1. Functional Requirements:
   - The extension must monitor all visited URLs in real-time and analyze them for phishing or malware.
   - Provide alerts for flagged URLs, offering actionable options like "Block" or "Proceed."
   - Allow users to manage a whitelist/blacklist of URLs.

    - Analyze email content for phishing indicators, including embedded links and sender details.
    - Provide malware detection using APIs like VirusTotal.
    - Include customizable settings for detection sensitivity and feature toggles.
    - Enable users to upload files or links for sandbox-based threat analysis.
2. Non-Functional Requirements:
    - The extension must be lightweight and not significantly affect browser performance.
    - User data must remain secure, and no sensitive information should be stored without explicit permission.
    - The user interface should be intuitive and accessible.
3. Performance Requirements:
    - URL and email analysis results should be delivered within 3 seconds for real-time efficiency.
    - Notifications must be immediate upon detecting a threat.

*** Success criteria:**

- The extension accurately detects phishing attempts with minimal false positives/negatives.
- Users receive real-time alerts with actionable recommendations.
- The system demonstrates adaptability to new phishing strategies.
- Positive feedback from users during testing and deployment.

*** Opportunities for ECU Students:**

- Gain hands-on experience in developing cybersecurity tools.
- Enhance problem-solving skills by tackling real-world challenges.
- Build expertise in web development and machine learning integration.
- Strengthen teamwork and collaboration skills.
- Develop industry-relevant skills and a portfolio-worthy project.

ECU Supervisor(s): Mr. Jude Myuran

ECU Co-Supervisor(s): Ms. Ann Appuhami

External contacts (where relevant):

Skills/experience required: Programming skills in JavaScript, HTML, and CSS, experience with JSON and API integration, knowledge of Chrome Extension APIs, familiarity with third-party APIs like Google Safe Browsing, PhishTank, and

VirusTotal, understanding of phishing techniques such as typosquatting and malicious TLDs, basic knowledge of heuristic analysis and machine learning frameworks like scikit-learn, experience in threat detection concepts and malware analysis, UI/UX design for creating user-friendly interfaces, problem-solving and debugging skills, cybersecurity knowledge about phishing and malware behaviors, familiarity with tools like Visual Studio Code, Git/GitHub, Postman, and browser developer tools, and strong time management and documentation skills.

Project team: 4 students

Location dependencies: ***(does the project require access to on-campus only systems? – No)***

Any other information: All intellectual property developed and created in the project shall be assigned to ECU.