IE3042 – SSS

Class Project
2024

Assignment Description

Secure Software Implementation

Deadline : 19th May 2024

IE3042 – Class Project

Course Overview:

This course is designed to provide students with the knowledge and skills to develop secure software applications. Students will learn about software development, secure coding practices, and security testing. The focus of the course is on developing a research-based secure software project that leverages machine learning and other software technologies to improve security.

Assignment Objective:

The objective of this assignment is to develop a research-based secure software project that leverages machine learning and other software technologies to improve security. Students will be required to write a White Paper/Technical Paper, develop the software with the MVP (Minimal Viable Product) to sell in a software store, and conduct a secure audit using OWASP or similar framework.

Assignment Description: The assignment will consist of three main components:

1. Technical Paper or White Paper:

- Produce a technical paper or white paper detailing the research, design, and implementation aspects of the secure software project.
- Discuss the significance of integrating machine learning and other software technologies in enhancing security measures.
- Provide an overview of the software architecture, algorithms employed, and anticipated security benefits.
- Present any novel approaches or contributions made to the field of secure software development.
- Include references to relevant literature, research papers, and patents that influenced the project.

2. Software Development:

- Develop a Minimum Viable Product (MVP) of the secure software project, suitable for commercialization in a software store.

  Feature Implementation

Utilize machine learning techniques or other new **technology/framework** to introduce advanced security features into the software. Examples include but are not limited to anomaly detection, threat prediction, and behavior analysis, which enhance the system's ability to identify and respond to potential security threats.

- Adhere to secure coding practices and incorporate industry-standard security controls to mitigate potential vulnerabilities.
- Ensure the usability and scalability of the software product to cater to a wide range of users and environments.
- Document the development process, including challenges faced, design decisions made, and lessons learned.

3. Secure Audit:

- Conduct a secure audit of the developed software project using established frameworks like OWASP or an equivalent.
- Identify and assess potential security vulnerabilities, including but not limited to injection flaws, authentication issues, and data exposure.
- Evaluate the effectiveness of implemented security measures and suggest improvements where necessary.
- Generate a comprehensive audit report outlining findings, recommendations, and proposed remediation strategies.
- Validate the security posture of the software project through rigorous testing and analysis.

Deliverables:
Students are expected to submit the following deliverables:

1. Technical Paper or White Paper detailing the research, design, and implementation aspects of the secure software project.
2. Minimum Viable Product (MVP) of the software project.
3. Secure Audit Report summarizing the findings, recommendations, and remediation strategies identified during the secure audit process.
4. Presentation highlighting key aspects of the project, including the significance of machine learning integration, software development challenges, and security audit outcomes. (Presentation schedule will be informed in due course).

Deadline : 19th May 2024

Marking Rubric

Technical Paper or White Paper (Out of 30):

Technical depth: 10
Clarity of writing and organization: 10
Coverage of design and implementation aspects: 10

Minimum Viable Product (MVP) (Out of 30):

Functionality and features implemented: 10
Usability and user interface design: 10
Integration of machine learning for security enhancement: 10

Secure Audit Report (Out of 20):

Identification and analysis of security findings: 8
Quality of recommendations and remediation strategies: 8
Clarity and organization of the report: 4

Presentation (Out of 20):

Presentation content and organization: 8
Viva – 12

Extracted Products from topics. [This is to get an idea about specific software and tech stack]

Secure Chat Application- A web or mobile application for secure messaging.
Possible Tech Stack: React, Node.js, Socket.io for real-time communication, and WebRTC for video calls.

Secure File Storage and Sharing- A web application for securely storing and sharing files.
Possible Tech Stack: React, Node.js, AWS S3 for file storage, and encryption libraries like CryptoJS or SJCL for secure file transfer.

Secure Email Client- A web or mobile application for sending and receiving encrypted emails.
Possible Tech Stack: React, Node.js, and OpenPGP.js for email encryption.

Secure Content Management System- A web application for securely managing and publishing content.
Possible Tech Stack: React, Node.js, MongoDB, and encryption libraries like bcrypt for user authentication.

Secure Code Repository- A web application for securely storing and sharing code.
Possible Tech Stack: React, Node.js, Git for version control, and encryption libraries like libsodium for secure communication.

Secure Passwordless Authentication- A web or mobile application for passwordless authentication.
Possible Tech Stack: React, Node.js, Auth0 for authentication, and WebAuthn for secure hardware-based authentication.

Secure VPN Service- A VPN service for secure internet browsing.
Possible Tech Stack: OpenVPN for secure communication, and AWS EC2 for hosting.

Secure Video Conferencing- A web application for secure video conferencing.
Possible Tech Stack: React, Node.js, WebRTC for video conferencing, and encryption libraries like AES for secure communication.

Secure Cloud Storage- A cloud storage service for securely storing files.
Possible Tech Stack: AWS S3 for storage, encryption libraries like SJCL for secure file transfer.

Secure Digital Wallet- A mobile application for securely storing and managing payment information.
Possible Tech Stack: React Native, Node.js, and encryption libraries like CryptoJS for secure payment information.

Secure IoT Device Management- A web application for securely managing and monitoring IoT devices.
Possible Tech Stack: React, Node.js, and encryption libraries like OpenSSL for secure communication.

Secure Online Voting System- A web application for securely conducting online voting.
Possible Tech Stack: React, Node.js, and encryption libraries like AES for secure communication.
Secure E-commerce Platform- A web application for securely buying and selling products.
Possible Tech Stack: React, Node.js, and encryption libraries like bcrypt for secure payment and user authentication.

Secure Document Signing- A web application for securely signing and verifying documents.
Possible Tech Stack: React, Node.js, and encryption libraries like OpenPGP.js for secure document signing.

Secure Health Record Management- A web application for securely managing and sharing health records.
Possible Tech Stack: React, Node.js, and encryption libraries like AES for secure communication.

Secure Blockchain Platform- A blockchain platform for secure and decentralized applications.

Possible Tech Stack: Ethereum, Solidity, and encryption libraries like libsodium for secure communication.

Secure Code Analysis Plugin- A plugin for secure code analysis in popular IDEs.
Possible Tech Stack: Java, Python, and static code analysis libraries like Checkmarx for

Audit Frameworks

There are several secure standards and frameworks that can be used for secure software development. Some of the major ones are:

OWASP

NIST (National Institute of Standards and Technology): NIST provides a comprehensive set of guidelines and best practices for securing information systems. The NIST Cybersecurity Framework is a widely adopted framework for managing cybersecurity risks.

> Implementation: Developers can follow the NIST guidelines and use the NIST Cybersecurity Framework to develop and implement a comprehensive security program.

CIS Controls: The Center for Internet Security (CIS) provides a set of 20 security controls that are widely used to help organizations improve their cybersecurity posture.

PCI DSS (Payment Card Industry Data Security Standard): PCI DSS is a set of security standards that are designed to ensure that all companies that accept, process, store or transmit credit card information maintain a secure environment.

In summary, developers can use one or more of these secure standards and frameworks to ensure that their software is developed and deployed securely. The implementation process involves following the guidelines and best practices provided by each framework and using the tools and resources provided by the organizations that maintain the standards.

Tech Stack for the Topics

1. Encryption libraries: OpenSSL, Cryptography, Bouncy Castle, NaCl

2. Authentication and authorization frameworks: OAuth2, OpenID Connect, JWT, OAuth1

3. Security testing tools: OWASP ZAP, Burp Suite, Metasploit, Nessus

4. Secure coding standards and frameworks: OWASP ASVS, OWASP Top 10, SANS Secure Coding Guidelines

5. Secure software development lifecycle tools: Microsoft SDL, BSIMM, OWASP SAMM

6. Static and dynamic code analysis tools: SonarQube, Fortify, Veracode, Checkmarx

7. Penetration testing tools: Kali Linux, Parrot OS, BlackArch, Samurai Web Testing Framework

8. Secure configuration management tools: Ansible, Puppet, Chef, SaltStack

9. Secure containerization tools: Docker, Kubernetes, OpenShift, Nomad

10. Secure cloud computing tools: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform

11. Machine learning and artificial intelligence (AI) frameworks: TensorFlow, Keras, PyTorch, Scikit-learn

12. Quantum cryptography libraries and frameworks: Qiskit, IBM Quantum, Microsoft Quantum

13. Blockchain and distributed ledger technology (DLT) platforms: Ethereum, Hyperledger, Corda, Stellar

14. Multi-factor authentication (MFA) tools: Google Authenticator, YubiKey, Duo Security, Authy

15. Secure programming languages: Rust, Swift, Kotlin, TypeScript, Python

16. Secure development environments: Visual Studio Code, Eclipse, IntelliJ IDEA, NetBeans

17. Github Copilot: An AI-powered code suggestion tool that can assist developers in writing secure code.

19. Cryptocurrency wallets: Ledger Nano X, Trezor, KeepKey, Exodus

20. Virtual private networks (VPNs): NordVPN, ExpressVPN, Surfshark, ProtonVPN.