Title	Secure Software Development
Long Title	Secure Software Development
Credits	5
NFQ Level	Expert
Module Author	Dr.Anila Mjeda

Module Description:

This module focuses on secure software development practices integrated throughout the software development life cycle. The module explores various pitfalls and mitigation techniques for software-based attacks and how to prevent the inclusion of such vulnerabilities in software. This module was developed under the Cyber Skills HCI Pillar 3 Project. Please refer to consortium agreement for ownership.

Learning Outcomes

On successful completion of this module the learner will be able to:

LO1 Design a secure software development framework that identifies a company's security requirements and best practices to prevent and identify software security vulnerabilities.

LO2 Integrate secure software design best practices as part of the Software Development Life Cycle to mitigate various forms of software-based attacks.

LO3 Appraise traditional and emerging software attack techniques.

LO4 Integrate secure coding practices with a view of avoiding traditional and emerging security pitfalls.

LO5 Appraise the security of modern applications guided by traditional and modern techniques and tooling.

Indicative Content

A Secure Software Development Framework

Integration of secure software development practices (also called a Secure Software Development Framework (SSDF)) within the Software Development Life Cycle. Elicitation of relevant security requirements. Concepts such as a system's requirements in terms of encryption, malicious application inputs, security headers, password storage, backup and rollback, and, security features of frameworks.

Secure Software Design

Secure software design as part of a Secure Software Development Framework. Secure design concepts such as protecting sensitive data, a zero-trust and assume breach position, server-side security validation, segregation of production data, protection of source code, and threat modelling.

Common Attacks

Traditional and emerging attack techniques such as Buffer Overflow and Heap Overflow, Return-to-LibC, ROP, UAF, SEH Based Attacks. Traditional and emerging mitigation techniques such as Stack Cookies, ASLR, DEP, SafeSEH, CFG, and EMET.

Secure Code

Evaluation of development frameworks and programming languages. Secure code concepts such as inclusion of Public Key Infrastructure (PKI) (PKI encryption and digital signature capabilities into applications such as S/MIME email, SSL traffic, authentication (AuthN), authorization (AuthZ), error handling, logging and monitoring. Assessing code with a view to locating specific vulnerability patterns. Using tools as an aid to locate vulnerabilities in own and third-party code.

Security of Modern Applications

Traditional and modern security techniques such as OWASP Top 10, Cross-Site Scripting, SQL Injection, NoSQL Injection, Command Injection, Server-Side Request Forgery, Remote Code Execution, Cross-Site Request Forgery, Secure Session Handling, Security as Code (SaC), Infrastructure as Code (IaC) and modern tooling.

Course Work

Assessment Type	*Assessment Description*	*Outcome Addressed*	*% of Total*	*Assessment Date*
Project	The students will be asked to design a secure software development framework (SSDF) that identifies a company's security requirements and integrates best software design practices to prevent and identify software security vulnerabilities. The students will be asked to follow this SSDF to design a proof-of-concept software system that focuses on specified targeted areas of the system's security. The learner will document their work in a report produced to a professional standard.	1,2	40.0	Week 7
Project	The students will be asked to develop the software system of their design, focusing on key security mechanisms of interest and following secure coding practices, techniques and tooling. The learner will document their work in a report produced to a professional standard.	3,4,5	60.0	Sem End

No End of Module Formal Examination

Assessment Breakdown	%
Coursework	100

Re-Assessment Requirement

Coursework

This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.

Workload – Full Time

Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Leaner Workload*
Lecture	Lectures covering the theoretical concepts underpinning the learning outcomes.	2.0	Every Week	2.00
Lab	Lab to support the learning outcomes.	2.0	Every Week	2.0
Independent & Directed Learning (Non-contact)	Independent learning by the student	3.0	Every Week	3.00
		Total Hours		7
		Total Weekly Learner Workload		7
		Total Weekly Contact Hours		4

Workload – Part Time

Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Leaner Workload*
Lecture	Lectures covering the theoretical concepts underpinning the learning outcomes.	2.0	Every Week	2.00
Lab	Lab to support the learning outcomes.	2.0	Every Week	2.0
Independent & Directed Learning (Non-contact)	Independent learning by the student	3.0	Every Week	3.00
		Total Hours		7
		Total Weekly Learner Workload		7
		Total Weekly Contact Hours		4

Recommended Book Resources

- **Tanya Janca 2020, *Alice and Bob Learn Application Security*, Wiley [ISBN: 9781119687351]**