Prifysgol Wrecsam
Wrexham University

Module specification

When printed this becomes an uncontrolled document. Please access the Module Directory for the most up to date version by clicking on the following link: Module directory

Module Code	COM577
Module Title	Secure Software Development
Level	5
Credit value	20
Faculty	FACE
HECoS Code	100956
Cost Code	GACP

Programmes in which module to be offered

Programme title	Is the module core or option for this programme
BSc (Hons) Computer Science	Core
BSc (Hons) Computer Science with Industry Placement	Core
BSc (Hons) Cyber Security	Core
BSc (Hons) Cyber Security with Industry Placement	Core
BSc (Hons) Software Engineering	Core
BSc (Hons) Software Engineering with Industry Placement	Core

Pre-requisites

N/A

Breakdown of module hours

Learning and teaching hours	15 hrs
Placement tutor support	0 hrs
Supervised learning e.g. practical classes, workshops	15 hrs
Project supervision (level 6 projects and dissertation modules only)	0 hrs
Total active learning and teaching hours	**30** hrs
Placement / work based learning	0 hrs
Guided independent study	170 hrs
Module duration (total hours)	**200** hrs

For office use only	
Initial approval date	08/11/2023
With effect from date	Sept 2025
Date and details of revision	
Version number	1

Module aims

This module's main aim is for students to understand the fundamental concepts of software security and its significance in modern applications. Identify the potential risks and consequences of insecure software development practices. Acquaint students with industry-standard methodologies for secure software development. Introduce secure coding practices, secure design principles and secure development life cycle (SDLC) models. Introduce students to various security testing techniques, including penetration testing and vulnerability scanning. Gain insights into the legal, ethical and regulatory aspects of secure software development.

Module Learning Outcomes - at the end of this module, students will be able to:

1	Identify the principles of secure software development.
2	Apply secure software development methodologies and practices.
3	Analyse and mitigate common security vulnerabilities in software applications.
4	Demonstrate critical thinking and problem-solving skills in secure software development.

Assessment

Indicative Assessment Tasks:
This section outlines the type of assessment task the student will be expected to complete as part of the module. More details will be made available in the relevant academic year module handbook.

Coursework could involve the completion of multiple tasks; one task could be developing a secure code snippet or module that demonstrates proper input validation and output encoding techniques. Assess the code for potential vulnerabilities and provide recommendations for improving security. Perform a security assessment of a provided software application, identifying potential vulnerabilities. Use penetration testing tools and techniques to exploit vulnerabilities and document the findings. Evaluate the security design and architecture of a software system, such as an online shopping platform or a social media application.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1,2,3,4,	Coursework	100%

Derogations

None

Learning and Teaching Strategies

In line with the Active Learning Framework, this module will be blended digitally with both a VLE and online community. Content will be available for students to access synchronously and asynchronously and may indicatively include first and third-party tutorials and videos, supporting files, online activities any additional content that supports their learning.
As this module progresses, the strategies will change to best support a diverse learning environment. Initially, the module will start with a heavier reliance on engaging tutor-led lectures, demonstrations, and workshops to ensure that the students get the relevant threshold concepts. As the module continues experiential and peer learning strategies will be encouraged as the students' progress with their portfolio work.
Assessment will occur throughout the module to build student confidence and self-efficacy in relation to applying secure software development concepts.

Indicative Syllabus Outline

Yearly content will be updated to represent the most appropriate content for current industry technologies, but a list of indicative topics could include:

- Importance of software security
- Secure Software Development Life Cycle (SDLC)
- Secure Design Principles
- Secure Coding Practices
- Cryptography and Secure Data Handling
- Web Application Security (OWASP Top 10)
- Secure Software Testing and Vulnerability Assessment
- Secure Software Deployment and Maintenance
- Cross-site scripting (XSS)
- Secure database access and protection against SQL injection
- Secure handling of user authentication and authorization

Indicative Bibliography:

Please note the essential reads and other indicative reading are subject to annual review and update.

Essential Reads

J. Davies, The Secure Coder's Handbook: A Practical Guide to Secure Development, Independently published, 2023.

Other indicative reading

J. Davies, The Secure Coder's Handbook: A Practical Guide to Secure Development, Independently published, 2023.
A. Klause, Python for Web Development: The Definitive Guide to Building Scalable, Secure, and High-Performance Web Apps with Django, Independently published, 2023.
F. Hissen, Secure Programming of Web Applications: Web Application Security for Software Developers and Project Managers, Independently published, 2019.