Liverpool John Moores University

Title:	SECURE SOFTWARE ENGINEERING
Status:	Definitive
Code:	**7138COMP** (122210)
Version Start Date:	01-08-2021
Owning School/Faculty:	Computer Science and Mathematics
Teaching School/Faculty:	Computer Science and Mathematics

Team	Leader
Nathan Shone	Y

Academic Level:	FHEQ7	**Credit Value:**	20	**Total Delivered Hours:**	36
Total Learning Hours:	200	**Private Study:**	164		

Delivery Options
Course typically offered: Semester 2

Component	Contact Hours
Lecture	12
Practical	24

Grading Basis: 50 %

Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Artefacts	AS1	Software testing and security task	100	

Aims

To develop students' analysis skills in identifying and understanding security problems and vulnerabilities, and the methods, tools, and techniques that can be used during software development to prevent them.
To develop students software development skills by applying a software development lifecycle in order to improve software security and robustness.

Learning Outcomes

After completing the module the student should be able to:

1 Apply best-practice security architectures and patterns to mitigate threats against software in different environments.
2 Demonstrate a comprehensive understanding of applying security techniques to software development.
3 Show critical awareness of the complexity of contemporary software vulnerabilities and the techniques to discover and mitigate them.

Learning Outcomes of Assessments

The assessment item list is assessed via the learning outcomes listed:

Software testing and security	1	2	3

Outline Syllabus

-Characteristics of large-scale software systems projects, team membership and activities.
-Networking vulnerabilities, access control, random number generation, cryptography, and authentication in software development.
-Understanding, exploiting, and mitigating common software vulnerabilities.
-Testing software to discover security vulnerabilities.
-Process models and lifecycles for secure software development.
-Threat modelling and formal techniques for vulnerability analysis.
-Secure deployment and post-deployment management of software.
-Understanding the implications of different computing environments on security and the software development process.
-Programming languages and security characteristics, decompilation, disassembly, and obfuscation.
-Recent examples from computing are used throughout and practical exercises used to illustrate the applications of these concepts.

Learning Activities

Students will undertake a software engineering task involving the application of secure software development lifecycles to a software development task. As part of this task, students will be expected to complete a report that demonstrates an understanding of how software should be designed, implemented, and tested to reduce the risk of security vulnerabilities. Students will also be expected to discover, through best practice security testing techniques, and mitigate vulnerabilities in software provided to them as part of this activity.

Students will participate in lectures, practical tutorials / lab sessions.

Notes

This module is intended to expose students to development practices that lead to reliable and secure software. Design models (secure development lifecycle model) as well technical skills including code vulnerability detection and testing will be explored. Students would benefit from prior programming experience but it is not essential as some experience will be gained during semester 1 modules, and extra tutorial support will be provided within the module. Students will work in small teams, mimicking the environment that most professional software engineers work in.