MODULE SPECIFICATION

Part 1: Information			
Module Title	Operating Systems Security and Defensive Programming		
Module Code	UFCFHU-30-2	Level	Level 5
For implementation from	2021-22		
UWE Credit Rating	30	ECTS Credit Rating	15
Faculty	Faculty of Environment & Technology	Field	
Department	FET Dept of Computer Sci & Creative Tech		
Module type:	Standard		
Pre-requisites	Operating Systems and Architecture 2020-21, Programming 2020-21		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description
Overview: This module introduces students to the tasks of operating systems such as controlling and allocating memory, prioritising system requests, controlling input and output devices, facilitating data networking and managing files, including security and protection. Students will learn the concepts of security protection in operating systems, such as hierarchical protection domains, and how they are employed to resist malware threats. A design pattern is a description of how to solve a problem that can be used in many different situations and can help deepen the understanding of object-orientated programming and help improve software design and reusability. They can also be used for secure programming and students will learn how to apply them along with other methods and tools. **Educational Aims:** This module contributes to cyber knowledge. It strengthens and deepens the computing underpinnings developed at earlier levels of study. **Outline Syllabus:** security and protection kernel security and protection

typical OS security features and how they may be exploited

approaches to defensive programming, for example input validation, least privilege, defence in depth, data sanitization, etc

authentication, authorisation and access control

resistance to malware techniques such as memory corruption, code injection, user/kernel space vulnerabilities, privilege escalation, etc.

design patterns for developing secure software

use of compiler features to support the creation of secure code

static and dynamic code analysis techniques

sources of secure programming practices, including employer or software development organisation, for different types of software systems (e.g., OWASP, CERT, etc.)

at least 1 formal method that may be applied to software development and its strengths and weaknesses when applied to development of software with security properties

Teaching and Learning Methods: Lecture sessions cover the technical knowledge required. Designated practical work is included to ensure that apprentices have absorbed and understood the key principles involved.

Part 3: Assessment

This module is assessed by a combination of techniques: a report (3,000 words) and a portfolio which includes a practical demonstration.

Component A

Students will be given a specification from which they have to produce a solution. This must incorporate the use of secure design patterns. They must show the research they have done to select the secure programming practices they will employ. As well as demonstrating the solution, apprentices will have to produce evidence of the design, development, implementation, test and debug.

Component B

Students will write a 3,000-word report that will require them to research the most prevalent threats to operating systems. They will then describe how operating system security features protect against these threats. Finally, they will show how operating systems should be configured to take the most advantage of these features.

First Sit Components	**Final Assessment**	**Element weighting**	**Description**
Portfolio - Component A		60 %	Portfolio showing evidence of design, development, implementation, testing and debugging of a solution to a security problem.
Report - Component B	✓	40 %	Report on contemporary operating system threats.
Resit Components	**Final Assessment**	**Element weighting**	**Description**

Portfolio - Component A		60 %	Re worked protfolio
Report - Component B	✓	40 %	Re-worked report

Part 4: Teaching and Learning Methods

Learning Outcomes

On successful completion of this module students will achieve the following learning outcomes:

Module Learning Outcomes	Reference
Configure an Operating System in accordance with security policy.	MO1
Identify threats and the features that mitigate the threats.	MO2
Identify appropriate secure programming principles and design patterns and analyse their fitness to address security issues	MO3
Research sources of secure programming practices and apply them	MO4

Contact Hours

Independent Study Hours:	
Independent study/self-guided study	135
Total Independent Study Hours:	135
Placement Study Hours:	
Placement	75
Total Placement Study Hours:	75
Scheduled Learning and Teaching Hours:	
Face-to-face learning	90
Total Scheduled Learning and Teaching Hours:	90
Hours to be allocated	300
Allocated Hours	300

Reading List

The reading list for this module can be accessed via the following link:

https://rl.talis.com/3/uwe/lists/20F18CAC-2867-6DAC-83CA-F36CCC443E5E.html

Part 5: Contributes Towards
This module contributes towards the following programmes of study: BSc (Hons) Cyber Security Technical Professional (integrated degree) BSc (Hons) 2020-21