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

Refer to the module guidance notes for completion of each section of the specification.

Module code	CONL711
Module title	Secure Software Development
Level	7
Credit value	15
Faculty	FAST
Module Leader	Nigel Houlden
HECoS Code	100374
Cost Code	GACP

Programmes in which module to be offered

Programme title	Is the module core or option for this programme
MSc Computer Science with Cyber Security	Core
MSc Computer Science with Software Engineering	Core

Pre-requisites

Studied CONL701 Critical Research for Postgraduate Study

Breakdown of module hours

Learning and teaching hours	15 hrs
Placement tutor support	0 hrs
Supervised learning e.g. practical classes, workshops	0 hrs
Project supervision (level 6 projects and dissertation modules only)	0 hrs
Total active learning and teaching hours	**15** hrs
Placement / work based learning	0 hrs
Guided independent study	135 hrs
Module duration (total hours)	150 hrs

For office use only	
Initial approval date	4/9/19
With effect from date	01/01/20
Date and details of revision	24/02/21 Revision to reading list effective from 08/3/21
Version number	3

Module aims

The module will allow students to understanding and apply the theory and practice of exploiting vulnerabilities in software as well as key skills of design and implementation of secure software. Students will learn the ability to implement secure systems and environments to support software security. Additionally, they will explore the use of secure programming languages and the effects on secure software. The use obfuscation and encryption in the protection of software will also be investigated.

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

1	Research, compare and contrast various approaches to software and/or system security.
2	Use and adapt secure programming techniques.
3	Critically evaluate approaches to obfuscation, encryption and signing in software and security.
4	Identify and evaluate weaknesses in computer software and systems.
5	Select, justify and document approaches, methods and techniques used to secure software.

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.

Throughout the module, students will develop a portfolio describing several aspects of secure software development, including appropriate techniques and design strategies. This will develop their understanding of appropriate practices and code assessment techniques.

Towards the end of the module, students will be given a case study, and asked to identify, document and present the potential security issues. This will involve analysing system documentation and evaluating samples.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1,2,3	Portfolio	60%
2	4,5	Case Study	40%

Derogations

None

Learning and Teaching Strategies

The overall learning and teaching strategy is one of guided independent study requiring ongoing student engagement. Online material will provide the foundation of the learning resources, requiring the students to login and engage on a regular basis throughout the eight-week period of the module. There will be a mix of suggested readings, discussions and interactive content containing embedded digital media and self-checks for students to complete as they work through the material and undertake the assessment tasks. The use of a range digital tools via the virtual learning environment together with additional sources of reading will also be utilised to accommodate learning styles. There is access to a helpline for additional support and chat facilities through Canvas for messaging and responding.

Indicative Syllabus Outline

Memory models.

Programming bugs and mistakes that lead to vulnerabilities.

Secure programming languages and frameworks.

Attacks against software.

Other software related attacks: e.g. XSS attacks, SQL injection, etc.

Programming for security.

Software and system protection methods.

‘Secure by design’ development.

Indicative Bibliography:

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

Essential Reads

Howard, M., LeBlanc, D. and Viega, J. (2009), 24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them. New York: McGraw-Hill.

Other indicative reading

Hoffman, A. (2020), Web Application Security: Exploitation and Countermeasures for Modern Web Applications. O'Reilly Media.

Azad, S. and Pahtan, A.S.K. (2014), *Practical Cryptography: Algorithms and Implementations Using C++.* Boca Raton, FL: Taylor & Francis.

Cachin, C., Geurraoui, R. and Rodrigues, L. (2011), *Introduction to Reliable and Secure Distributed Programming*. Springer.

Seacord, R.C. (2013), *Secure Coding in C and C++*. Upper Saddle River, NJ: AddisonWesley.

Shalloway, A., Bain, S., Pugh, K. and Kolsky, A. (2011), *Essentials Skills for the Agile Developer: A Guide to Better Programming and Design.* Boston: Addison-Wesley.

Appropriate web-based sources will be used to supplement the reading list.

Employability skills – the Glyndŵr Graduate

Each module and programme is designed to cover core Glyndŵr Graduate Attributes with the aim that each Graduate will leave Glyndŵr having achieved key employability skills as part of their study. The following attributes will be covered within this module either through the content or as part of the assessment. The programme is designed to cover all attributes and each module may cover different areas. Click here to read more about the Glyndwr Graduate attributes

Core Attributes

Engaged
Enterprising
Creative
Ethical

Key Attitudes
Commitment
Curiosity
Resilience
Confidence
Adaptability

Practical Skillsets
Digital Fluency
Organisation
Leadership and Team working Critical Thinking
Emotional Intelligence Communication