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	COM561
Module title	Secure software design
Level	5
Credit value	20
Faculty	FAST
Module Leader	Teri Birch
HECoS Code	100374
Cost Code	GACP

Programmes in which module to be offered

Programme title	Is the module core or option for this programme
BSc (Hons) Applied Cyber Security	Core
BSc (Hons) Applied Software Engineering	Core

Pre-requisites

None

Breakdown of module hours

Learning and teaching hours	30 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	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	10 Nov 2021
With effect from date	Jan 2021

For office use only	
Date and details of revision	
Version number	1

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, comparing contrast various approaches to software and/or system security
2	Demonstrate secure programming techniques
3	Demonstrate an understanding of weaknesses in software and/or systems
4	Express an understanding of approaches, methods and techniques to secure software
5	Demonstrate an understanding of obfuscation, encryption and signing in software and system security

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.

The module will have one assessment of a portfolio of practical and short reports based on various aspects of module content such as development of secure programs, exploitation, and mitigation of vulnerabilities.

Where practical the assessment will be related / carried out in the workplace.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)
1	1,2,3,4,5	Portfolio	100%

Derogations

None

Learning and Teaching Strategies

Students will develop understanding and practical skills based on weekly lectures, task orientated tutorials and supervised workshops. The teaching sessions will utilise examples/case studies as a platform for understanding software security principles.

Appropriate blended learning approaches and technologies, such as, the University's VLE

and computer security tools, will be used to facilitate and support student learning, in particular, to:

• deliver content;

• encourage active learning;

• provide formative and summative assessments, and prompt feedback;

• enhance student engagement and learning experience.

Indicative Syllabus Outline

- Introduction to Secure Systems
- Secure Development Lifecycles
- Access Control Problems and Controls
- Input Problems and Controls
- Memory Management Problems and Controls
- Reflected and Stored Script Problems and
- Dealing with External Data
- Keys and Encryption
- Testing and Compliance

Indicative Bibliography:

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

Essential Reads

Kohnfelder , L. Designing Secure Software: A Guide for Developers No Starch Press, US (2021)

Other indicative reading

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.

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.

Coffin, D. (2011), Expert Oracle and Java Security: Programming Secure Oracle Database Applications with Java. Apress.

Johnson, T.A. (2015), Cybersecurity: Protecting Critical Infrastructures from Cyber-attack and Cyber Warfare. CRC Press.

Manico, J. and Detlefsen, A. (2014), Iron-clad Java: Building Secure Web Applications. New York: McGraw Hill Education.

O'Connor, T.J. (2012), Violent Python: A Cookbook for Hackers, Forensic Analysists, Penetration Testers and Security Engineers. Syngess.

Seacord, R.C. (2013), Secure Coding in C and C++. Upper Saddle River, NJ: Addison-Wesley.

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.

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.

Core Attributes
Engaged
Enterprising
Creative
Ethical

Key Attitudes
Commitment
Curiosity
Resilience
Confidence
Adaptability

Practical Skillsets
Digital Fluency
Organisation
Critical Thinking
Communication