PRIFYSGOL Glyndŵr Wrecsam | Wrexham Glyndŵr UNIVERSITY

MODULE SPECIFICATION PROFORMA

Module Code:	COM737

Module Title:	Developing Secure Software

Level:	7	**Credit Value:**	20

Cost Centre(s):	GACP	JACS3 **code**: HECoS code:	I300 100374

Faculty:	Arts, Science and Technology	**Module Leader:**	Nigel Houlden

Scheduled learning and teaching hours	21 hrs
Guided independent study	179 hrs
Placement	0 hrs
Module duration (total hours)	200 hrs

Programme(s) in which to be offered (not including exit awards)	Core	Option
MSc Cyber Security	✓	□

Pre-requisites
None

Office use only
Initial approval: 28/11/2018 Version no:1
With effect from: 01/09/2019
Date and details of revision: Version no:

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.

Intended Learning Outcomes

Key skills for employability

KS1	Written, oral and media communication skills
KS2	Leadership, team working and networking skills
KS3	Opportunity, creativity and problem solving skills
KS4	Information technology skills and digital literacy
KS5	Information management skills
KS6	Research skills
KS7	Intercultural and sustainability skills
KS8	Career management skills
KS9	Learning to learn (managing personal and professional development, self-management)
KS10	Numeracy

At the end of this module, students will be able to		Key Skills	
1	Research, comparing contrast various approaches to software and/or system security	KS1	
		KS5	
		KS6	
2	Demonstrate secure programming techniques	KS2	
		KS3	
		KS10	
3	Demonstrate an understanding of weaknesses in software and/or systems	KS3	
		KS5	
		KS1	
4	Express an understanding of approaches, methods and techniques to secure software	KS3	
		KS6	
5	Demonstrate an understanding of obfuscation, encryption and signing in software and system security	KS3	
		KS6	

Transferable skills and other attributes

Derogations

None

Assessment:

Indicative Assessment Tasks:

Assessment 1 will comprise of a portfolio of weekly practical exercises carried out over a minimum of six weeks. The exercises will be based on various aspects of module content such as development of secure programs, exploitation and mitigation of vulnerabilities. Each week will be submitted within allocated time for that week's activity such that continuous feedback can be provided for improvement.

Assessment 2 will be an in-class test hosted on the virtual learning environment which will test students on their understanding and knowledge of the module content.

Assessment number	Learning Outcomes to be met	Type of assessment	Weighting (%)	Duration (if exam)	Word count (or equivalent if appropriate)
1	1-5	Portfolio	70		4000
2	3,4	In-class test	30	1.5 hours	

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.

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:

Essential 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.

Other indicative reading

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.

Wu, H.and Zhao, L. (2017), *Web Security: A Whitehat Perspective*. Boca Raton, FL: Auerbach Publications.

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