2024 / 25

School of Science and Computing
☎ +353 (0)51 302037
✉ Eleanor.Reade@setu.ie
🌐 www.wit.ie/schools/science_computing

SE
TU

Ollscoil
Teicneolaíochta
an Oirdheiscirt

South East
Technological
University

Module Descriptor

Secure Programming and Scripting (Computing and Mathematics)

Secure Programming and Scripting (A13740)

Short Title: Secure Programming
Department: Computing and Mathematics
Credits: 5 **Level:** Introductory

Description of Module / Aims

This module aims to equip students with the knowledge and skills to apply best security practices when programming in a range of languages and environments. In addition, they will gain an appreciation of risks and learn how to carry out basic threat modelling and avoid common vulnerabilities.

Programmes

		stage/semester/status
PROG-0083	BSc (Hons) in Applied Computing (WD_KACCM_B)	2 / 4 / E
PROG-0083	BSc (Hons) in Applied Computing (WD_KCOMP_B)	2 / 4 / E
PROG-0083	BSc (Hons) in Computer Forensics and Security (WD_KCOFO_B)	2 / 4 / M
PROG-0083	BSc (Hons) in Computer Science (WD_KCMSC_B)	2 / 4 / E

Indicative Content

- Software vulnerabilities
- Threat modelling
- Secure programming principles
- Memory allocation, memory leaks, overflows
- Web application security
- Secure software engineering; requirements; design; code auditing & review; testing; deployment

Learning Outcomes

On successful completion of this module, a student will be able to:

1. Describe a selection of security vulnerabilities caused by software development flaws.
2. Show using code examples how memory overflows can cause programs to behave unexpectedly.
3. Demonstrate specific security problems that can arise with web applications and how to address them.
4. Model security threats in the specification of requirements for a software system.
5. Explain how to build security measures into the software development process.

Learning and Teaching Methods

- This module will be presented by a combination of lectures and practical classes.
- The lectures will be used to introduce new topics and their related concepts.
- The practical element allows the student to put into practice the theoretical concepts covered in the lectures.
- The practical element involves a selection of laboratory exercises and related tasks. For example, students will look for vulnerabilities in applications provided and also write/adapt their own code to demonstrate common vulnerabilities and mitigation techniques. They will also carry out a high-level threat modelling exercise.

Learning Modes

Learning Type	F/T Hours	P/T Hours
Lecture	24	
Practical	24	
Independent Learning	87	

Assessment Methods

	Weighting	Outcomes Assessed
Continuous Assessment	100%	
Case Studies	30%	1,4,5
Lab Report	25%	2,5
Assignment	45%	3,5

Assessment Criteria

<40%: Unable to distinguish between different types of vulnerabilities or present instances of them in a clear manner.

40%–49%: Able to present instances of software vulnerabilities and carry out threat modelling on a basic system.

50%–59%: Able to explain in context and present instances of both low level software vulnerabilities and higher level web application vulnerabilities. Able to model threats in a software system with multiple usage scenarios and actors.

60%–69%: Also able to present and explain how to address both low level and web application vulnerabilities.

70%–100%: All the above to an excellent level. Able to present and explain in detail various ways to address both low level and web application vulnerabilities.

Essential Material(s)

- "Open Web Application Security Project." https://www.owasp.org

Supplementary Material(s)

- "Computer Emergency Response Team." https://www.cert.org
- "Security Focus." http://securityfocus.com
- McGraw, G. *Software Security: Building Security In.* NY: Addison-Wesley, 2006.
- Sullivan, B. and V. Liu. *Web Application Security, A Beginner's Guide.* NY: McGraw-Hill, 2012.

Requested Resources

- Computer Lab: BYOD Lab