1. **Title of the module**

COMP6640 (CO664) Secure Programming

1. **School or partner institution which will be responsible for management of the module**

School of Computing

1. **The level of the module (Level 4, Level 5, Level 6 or Level 7)**

Level 6

1. **The number of credits and the ECTS value which the module represents**

15 credits (7.5 ECTS)

1. **Which term(s) the module is to be taught in (or other teaching pattern)**

Autumn or Spring

1. **Prerequisite and co-requisite modules**

Prerequisites:

CO557 Computer Systems

1. **The programmes of study to which the module contributes**

BSc Computer Science (Cyber Security), both with and without Year in Industry

1. **The intended subject specific learning outcomes.  
   On successfully completing the module students will be able to:**
   1. understand programming principles and best practices to implement secure-by-design systems, i.e., software which is robust and resilient to attacks;
   2. develop analytical and practical skills (e.g., testing techniques and tools) to identify and avoid security vulnerabilities during all phases of software development;
   3. understand reverse engineering methods and techniques helpful for malware analysis;
   4. understand how security vulnerabilities in software can be exploited.
2. **The intended generic learning outcomes.  
   On successfully completing the module students will be able to:**
   1. Apply learned skills in the general IT environment
   2. Critically evaluate and communicate the behaviour of software and its security implications.
   3. Justify design choices.
3. **A synopsis of the curriculum**

The module focuses on providing foundations of theory and practice for security-aware software development in common architectures. It involves exposing students to common software vulnerabilities (e.g., implementation errors, logic flaws, and security weaknesses) and methods used in malware to exploit security (e.g., privilege escalation, memory leak, injection techniques, obfuscation and code mutation), and how to fix them across the software development lifecycle. As part of the module, students will learn reverse engineering techniques and get familiar with testing techniques and tools such as fuzzing, static analysis, and anti-debugging.

1. **Reading list (Indicative list, current at time of publication. Reading lists will be published annually)**

OWASP, “Security by Design Principles”, n.d., [Online].

Hoglund, G. and McGraw, G., “Exploiting Software: How to Break Code”, 2004, Addison-Wesley.

Howard, M. and LeBlanc, D., “Writing Secure Code”, 2002, Microsoft Press.

Chess, B. and West, J., “Secure Programming with Static Analysis”, 2007, Addison-Wesley.

Eagle, C., The IDA Pro Book, 2nd Edition, 2017, No Starch Press.

1. **Learning and teaching methods**

Total contact hours: 30

Private study hours: 120

Total study hours: 150

1. **Assessment methods**
   1. Main assessment methods

2 hour written exam (50%)

2 practical assessments (2x25%)

* 1. Reassessment methods

Like for like.

1. **Map of module learning outcomes (sections 8 & 9) to learning and teaching methods (section12) and methods of assessment (section 13)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Module learning outcome** | *8.1* | *8.2* | *8.3* | *8.4* | *9.1* | *9.2* | *9.3* |
| **Learning/ teaching method** |  |  |  |  |  |  |  |
| Lectures | x | x | x | x | x | x | x |
| *Private study* | x | x | x | x | x | x | x |
| **Assessment method** |  |  |  |  |  |  |  |
| *Assessments* | x | x | x | x | x | x | x |
| *Examination* | x | x | x | x | x | x | x |

1. **Inclusive module design**

The School recognises and has embedded the expectations of current equality legislation, by ensuring that the module is as accessible as possible by design. Additional alternative arrangements for students with Inclusive Learning Plans (ILPs)/declared disabilities will be made on an individual basis, in consultation with the relevant policies and support services.

The inclusive practices in the guidance (see Annex B Appendix A) have been considered in order to support all students in the following areas:

a) Accessible resources and curriculum

b) Learning, teaching and assessment methods

1. **Campus(es) or centre(s) where module will be delivered**

Canterbury

1. **Internationalisation**

The topics addressed by this module relate to a field which is of international importance, given the global role of computers in today's technological innovation. The topics covered by this module are international in nature, being identical worldwide and independent of traditional spoken language.

**FACULTIES SUPPORT OFFICE USE ONLY**

**Revision record – all revisions must be recorded in the grid and full details of the change retained in the appropriate committee records.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date approved | Major/minor revision | Start date of the delivery of revised version | Section revised | Impacts PLOs (Q6&7 cover sheet) |
|  |  |  |  |  |
|  |  |  |  |  |

Revised FSO Jan 2018