Module Specification

Module Summary Information

1	**Module Title**	Software Security
2	**Module Credits**	20
3	**Module Level**	5
4	**Module Code**	CMP5355

5	**Module Overview**

Software is ubiquitous. Not only is it deployed on traditional computing platforms (such as workstations and servers) but it is also embedded in network equipment (such as routers and firewalls) and consumer devices (such as hearing aids and smartphones). This module introduces you to a system programming language and the edit-compile-link-execute process. It continues the software theme that you began at Level 4 when you studied Computer Programming and Applied Operating Systems.

Complex software inevitably contains unforeseen defects. By exploiting these software vulnerabilities, an attacker might be able to corrupt or extract data, or to disrupt or take control of a system, for example. In this module, you will encounter common attack mechanisms (such as command injection and buffer manipulation) that exploit programming errors. You will also learn language-specific guidelines that help programmers avoid certain types of vulnerability; this will enable you to appreciate the warnings generated by compilers and analysis tools that indicate programs are potentially using language features insecurely.

Once a vulnerability is discovered, it must be fixed in order to avoid harm. For example, the Heartbleed Bug was a vulnerability in an earlier release of a cryptographic software library that is in widespread use, for example, in Cisco products, Minecraft gameplay and Amazon Web Services. You will also learn from this module how vendors of products and services should manage vulnerabilities in order to reduce the exposure their customers to risk.

6	**Indicative Content**

- Basics of C programming and the edit-compile-link-execute process
- Common software vulnerabilities and their remediation
- Static and dynamic analysis tools (e.g., Cppcheck and American fuzzy lop)
- Approaches to vulnerability reporting and handling (e.g., bug bounties, CVE identification) and the challenges of patch management

7	Module Learning Outcomes On successful completion of the module, students will be able to:	
	1	Write simple programs that compile into executable files
	2	Demonstrate common software vulnerabilities and how to avoid them
	3	Interpret and communicate reports generated by software analysis tools
	4	Justify the need for responsible disclosure of vulnerabilities and regular patching of software

8	Module Assessment		
Learning Outcome			
	Coursework	**Exam**	**In-Person**
1-4	**X**		

9	Breakdown Learning and Teaching Activities
Learning Activities	**Hours**
Scheduled Learning (SL) includes lectures, practical classes and workshops, peer group learning, Graduate+, as specified in timetable	48
Directed Learning (DL) includes placements, work-based learning, external visits, on-line activity, Graduate+, peer learning, as directed on VLE	112
Private Study (PS) includes preparation for exams	40
Total Study Hours:	200