Title:	Software Vulnerabilities APPROVED
Long Title:	Software Vulnerabilities

Module Code:	COMP9056	**Duration:**	1 Semester

Credits:	5
NFQ Level:	Expert
Field of Study:	Computer Science
Valid From:	Semester 1 - 2017/18 (September 2017)
Module Delivered in	**4 programme(s)**
Module Coordinator:	Sean McSweeney
Module Author:	VINCENT RYAN
Module Description:	When developing code, developers and designers often unwittingly introduce software vulnerabilities because they are unaware how they are caused or how to avoid them. These vulnerabilities are often very difficult to find and while testers and developers are trained how to test for and locate software bugs, the same is not true for vulnerabilities. In this module students will learn how to design and develop software so that security is an integral part of the process. In addition, students will learn how to locate and identify software vulnerabilities and how to prevent the inclusion of such vulnerabilities in software.

Learning Outcomes	
On successful completion of this module the learner will be able to:	
LO1	Evaluate the security of a software system using a threat modelling approach.
LO2	Assess the security of a software system, where the source code is available.
LO3	Review the security of a software system, where only the binary code is available.
LO4	Apply fuzzing techniques to a software system and triage resulting crashed binaries.
LO5	Design a strategy for an software company that identifies best practices and procedures to prevent and identify software vulnerabilities.

Pre-requisite learning

Module Recommendations
This is prior learning (or a practical skill) that is strongly recommended before enrolment in this module. You may enrol in this module if you have not acquired the recommended learning but you will have considerable difficulty in passing (i.e. achieving the learning outcomes of) the module. While the prior learning is expressed as named CIT module(s) it also allows for learning (in another module or modules) which is equivalent to the learning specified in the named module(s).

Incompatible Modules
These are modules which have learning outcomes that are too similar to the learning outcomes of this module. You may not earn additional credit for the same learning and therefore you may not enrol in this module if you have successfully completed any modules in the incompatible list.

12867	COMP9031	Software Security

Co-requisite Modules

No Co-requisite modules listed

Requirements
This is prior learning (or a practical skill) that is mandatory before enrolment in this module is allowed. You may not enrol on this module if you have not acquired the learning specified in this section.

No requirements listed

Module Content & Assessment

Indicative Content
Threat Modelling White box and black box Threat Modelling with a view to locating architecture and design related threats and vulnerabilities; ATASM (Architecture, Threats, Attack Surfaces, Mitigations)
Source Code Review Reviewing code with a view to locating specific vulnerability patterns. User and Kernel mode vulnerability classes associated with pointers and integers (e.g. integer overflow, Use After Free)
Review of Binary Files Using tools such as IDAPro as an aid to locate vulnerabilities in an binary file. Often using a Black Box approach
Fuzzing Applying fuzzing techniques and tools to a binary with a view to discovering exploitable vulnerabilities.
Crash Triaging Examining memory dumps and using real-time memory analysis with a view to determining exploitability
Security Program Design Designing a strategy to create and manage a security team dedicated to software vulnerability location in a medium/large organisation.

Assessment Breakdown	%
Course Work	100.00%

Course Work

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Project	This project will focus on source code analysis, with a view to locating vulnerabilities.	2	30.0	Week 7
Project	This project will focus on examining binary files in order to locate vulnerabilities, and will include a proof of concept exploitation.	3,4	30.0	Week 11
Project	In this assessment students will use threat modeling techniques to locate threats and vulnerabilities and will also design a strategy for a software security team outlining best practices and procedures to prevent and identify the introduction of software vulnerabilities.	1,5	40.0	Sem End

No End of Module Formal Examination

Reassessment Requirement
Coursework Only *This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.*

The institute reserves the right to alter the nature and timings of assessment

Module Workload

Workload: Full Time				
Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Weekly lectures covering essential theory.	2.0	Every Week	2.00
Lab	Weekly labs to support lectures.	1.0	Every Week	1.00
Independent Learning	Weekly student independent learning.	4.0	Every Week	4.00
			Total Hours	7.00
			Total Weekly Learner Workload	7.00
			Total Weekly Contact Hours	3.00

Workload: Part Time				
Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Weekly lectures covering essential theory.	2.0	Every Week	2.00
Lab	Weekly labs to support lectures.	1.0	Every Week	1.00
Independent Learning	Weekly student independent learning.	4.0	Every Week	4.00
			Total Hours	7.00
			Total Weekly Learner Workload	7.00
			Total Weekly Contact Hours	3.00

Module Resources

Recommended Book Resources

- **Brook S. E. Schoenfield 2015, *Securing Systems: Applied Security Architecture and Threat Models*, CRC Press [ISBN: 9781482233971]**

Supplementary Book Resources

- **Mark Dowd,John McDonald and Justin Schuh 2006, *The Art of Software Security Assessment*, Addison Wesley [ISBN: 9780321444424]**
- **Tobias Klein 2011, *A Bug Hunter's Diary: A Guided Tour Through the Wilds of Software Security*, No Starch Press [ISBN: 9781593273859]**
- **Chris Eagle 2011, *The IDA Pro Book: The Unofficial Guide to the World's Most Popular Disassembler*, 2nd Ed., No Starch Press [ISBN: 9781593272890]**
- **Enrico Perla and Massimiliano Oldani 2010, *A Guide to Kernel Exploitation: Attacking the Core*, Syngress [ISBN: 9781597494861]**
- **Daniel Regalado, Shon Harris, Allen Harper, Chris Eagle, Jonathan Ness, Branko Spasojevic, Ryan Linn and Stephen Sims 2015, *Gray Hat Hacking The Ethical Hacker's Handbook*, 4th Ed., McGraw-Hill [ISBN: 9780071832380]**
- **Joxean Koret and Elias Bachaalany 2015, *The Antivirus Hacker's Handbook*, Wiley [ISBN: 9781119028758]**
- **Bill Blunden 2012, *The Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System*, 2nd Ed., Jones & Bartlett [ISBN: 9781449626365]**
- **Charlie Miller and Dino Dai Zovi 2009, *The Mac Hacker's Handbook*, Wiley [ISBN: 9780470395363]**

Supplementary Article/Paper Resources

- **Alexander Sotirov and Mark Dowd 2008, *Bypassing Browser Protection Mechanisms***
  https://www.blackhat.com/presentations/bh-usa-08/Sotirov_Dowd/bh08-sotirov-dowd. pdf
- **Nikita Tarakanov 2014, *The past, the present and the future of software exploitation techniques***
  http://2014.zeronights.org/assets/files/slides/the-past-the-present-and-the-futu re-of-software-exploitation-techniques.pptx
- **Dennis Yurichev 2017, *Intro to Reverse Engineers for Beginners***
  https://beginners.re/RE4B-EN.pdf

Other Resources

- **website: *Google's Project Zero Team***
  https://googleprojectzero.blogspot.com/
- **website: *Avoiding the Top 10 Software Security Design Flaws***
  http://cybersecurity.ieee.org/blog/2015/ 11/13/avoiding-the-top-10-security-flaws /
- **website: *From fuzzing to 0-day***
  https://blog.techorganic.com/2014/05/14/ from-fuzzing-to-0-day/
- **website: *15 minute guide to fuzzing***
  https://www.mwrinfosecurity.com/our-thin king/15-minute-guide-to-fuzzing/
- **website: *Exploit Dev Community***
  http://expdev-kiuhnm.rhcloud.com/
- **website: *Chris Rohlf Modern Memory Safety in C/C++***
  https://github.com/struct/mms
- **website: *Corelan Exploitation Tutorial Series***
  https://www.corelan.be/index.php/2009/07 /19/exploit-writing-tutorial-part-1-stac k-based-overflows/

Module Delivered in

Programme Code	Programme	Semester	Delivery
CR_KINSE_9	Master of Science in Cybersecurity	2	Elective
CR_KSADE_9	Master of Science in Software Architecture & Design	1	Group Elective 1
CR_KSADE_9	Master of Science in Software Architecture & Design	2	Group Elective 1
CR_KINSY_9	Postgraduate Diploma in Science in Cybersecurity	2	Elective