H9SAD: Secure Application Development

Module Code:	H9SAD
Long Title	Secure Application Development APPROVED
Title	Secure Application Development
Module Level:	LEVEL 9
EQF Level:	7
EHEA Level:	Second Cycle
Credits:	5
Module Coordinator:	Arghir Moldovan
Module Author:	Andrea Del Campo Dugova
Departments:	School of Computing
Specifications of the qualifications and experience required of staff	PhD/Master's degree in a computing or cognate discipline. May have industry experience also.

Learning Outcomes	
On successful completion of this module the learner will be able to:	
#	**Learning Outcome Description**
LO1	Investigate and critically assess the impact of application security vulnerabilities on users of software products.
LO2	Analyse the security considerations associated with the state-of-the-art security toolchains for creating security controls that prevent common application security vulnerabilities.
LO3	Implement secure coding solutions that fix common software application security vulnerabilities.
LO4	Investigate the Secure Software Development Framework (SSDF) as supporting mechanisms for secure application development.
LO5	Critically assess secure coding guideline best practice and standards as applied to produce high level security controls for application development

Dependencies
Module Recommendations
No recommendations listed
Co-requisite Modules
No Co-requisite modules listed

Entry requirements	Programme entry requirements must be satisfied.

H9SAD: Secure Application Development

Module Content & Assessment

Indicative Content
Introduction & module Overview followed by Impact of vulnerability exploits Exploits: Examples of Identity Theft, data theft, ransom etc Organisational level: Economic Cost, Reputational Cost, Legal Consequences.
Secure Coding Standards Secure coding standards Cert Oracle or OWASP secure coding best practice, Principles of Secure Design Input Validation Output Encoding Authentication and Password Management Session Management Access Control Cryptographic Practices Error Handling and Logging
Secure Coding Standards Data Protection Communication Security System Configuration Database Security File Management Memory Management General Coding Practices
Understanding of how some exploits can happen Password misuse, directory traversal, access control prevention, broken authentication etc
State of the art Tool Chains: The stack as a whole Analyse the risk of applicable technology stacks (e.g., languages, environments, deployment models), and recommend or require the use of stacks that will reduce risk compared to others
State of the art Tool Chains: Individual tools Evaluate, select, and acquire tools, and assess the security of each tool. Regular verification of tools
Software Security Checks Define criteria for software security checks and track throughout the SDLC to prevent common application security vulnerabilities
Secure coding solutions The Problem: Identify and explain the occurrence and consequence of a variety of the following: • Bugs • Exposure of sensitive data • Flaws in Injection • Buffer overflow • Security misconfiguration • Broken access control • Insecure deserialization • Broken/Missing Authentication
Secure coding solutions: Examine a host of solutions to secure coding problems Part 1 with examples
Secure coding solutions: Examine a host of solutions to secure coding problems Part 2 with examples
Testing Techniques Testing Tools and Methodologies to find Bugs, Flaws, Black Box White Box Fuzz Testing Static Analysis & Dynamic Analysis
Secure Development Framework Part 1 Prepare The Organization Protect Software Produce Well Secured Software Respond To Vulnerabilities
All Content Recap Impact of application security vulnerabilities on users of software products State of the art security toolchains for creating security controls Implement secure coding solutions that fix common software application security vulnerabilities Critically assess Secure Software Development Framework (SSDF) as supporting mechanisms for secure application development.

Assessment Breakdown	%
Coursework	100.00%

Assessments

Full Time

Coursework

Assessment Type:	Formative Assessment	**% of total:**	Non-Marked
Assessment Date:	n/a	**Outcome addressed:**	1,2,3,4,5
Non-Marked:	Yes		

Assessment Description:
Formative assessment will be provided on the in-class individual or group activities. Feedback will be provided in written or oral format, or on-line through Moodle. In addition, in class discussions will be undertaken as part of the practical approach to learning.

Assessment Type:	Continuous Assessment	**% of total:**	30
Assessment Date:	n/a	**Outcome addressed:**	1,5
Non-Marked:	No		

Assessment Description:
This assessment will consist of a written academic report supported by relevant research and conclusions. This will assess learners' knowledge and competences on core secure application development concepts and methodologies covered so far.

Assessment Type:	Project	**% of total:**	70
Assessment Date:	n/a	**Outcome addressed:**	1,2,3,4,5
Non-Marked:	No		

Assessment Description:
The terminal assessment will consist of a project that will evaluate all learning outcomes. Learners will have to develop a software application to a given specification utilising appropriate secure supplication development techniques, tools / frameworks / services. The final submission will consist of a written report and the implemented securely developed application.

No End of Module Assessment

No Workplace Assessment

Reassessment Requirement

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

Reassessment Description
The reassessment strategy for this module will consist of a project that will assess all learning outcomes.

H9SAD: Secure Application Development

Module Workload

Module Target Workload Hours 0 Hours

Workload: Full Time

Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Classroom and demonstrations	24	Per Semester	2.00
Tutorial	Mentoring and small-group tutoring	12	Per Semester	1.00
Independent Learning	Independent learning	89	Per Semester	7.42
			Total Weekly Contact Hours	3.00

Workload: Blended

Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Classroom and demonstrations	12	Per Semester	1.00
Tutorial	Mentoring and small-group tutoring	12	Per Semester	1.00
Directed Learning	Directed e-learning	12	Per Semester	1.00
Independent Learning	Independent learning	89	Per Semester	7.42
			Total Weekly Contact Hours	3.00

Workload: Part Time

Workload Type	*Workload Description*	*Hours*	*Frequency*	*Average Weekly Learner Workload*
Lecture	Classroom and demonstrations	24	Per Semester	2.00
Tutorial	Mentoring and small-group tutoring	12	Per Semester	1.00
Independent Learning	Independent learning	89	Per Semester	7.42
			Total Weekly Contact Hours	3.00

Module Resources

Recommended Book Resources

Daniel Deogun,Dan Bergh Johnsson,Daniel Sawano. (2019), Secure By Design, Manning Publications, [ISBN: 978-1617294358].

Loren Kohnfelder. (2021), Designing Secure Software: A Guide for Developers, No Starch Press, p.330, [ISBN: 978-1718501928].

Supplementary Book Resources

Gerardus Blokdyk. (2020), Software Security Vulnerability A Complete Guide - 2020 Edition, 5STARCooks, p.310, [ISBN: 978-1867321460].

This module does not have any article/paper resources

Other Resources

[Website], OWASP Secure Coding Practices Quick Reference Guide (PDF), https://owasp.org/www-project-secure-cod ing-practices-quick-reference-guide/

[Website], SEI CERT Oracle Coding Standard for Java, https://wiki.sei.cmu.edu/confluence/disp lay/java/SEI+CERT+Oracle+Coding+Standard +for+Java

Discussion Note:	