

CSCE 548: Building Secure Software

1. Course number and name: CSCE 548: Building Secure Software
2. Credit: 3-hrs; Contact: 3 lecture periods of 50 minutes or 2 periods of 75 minutes per week
3. Instructor: Csilla Farkas
4. Textbook: *Software Security: Building Security In* by Gary McGraw, Publisher: Addison-Wesley Professional, February 2, 2006, ISBN-10: 0321356705 ISBN-13: 978-0321356703
 - a. *19 Deadly Sins of Software Security* by Michael Howard, David LeBlanc, John Viega, Publisher: McGraw-Hill Osborne Media, July 26, 2005, ISBN-10: 0072260858, ISBN-13: 978-0072260854
5. Specific course information
 - a. Catalog description: Construction of software systems resistant to vulnerabilities and attacks. Cryptographic tools. Language, operating system, and network security. Case studies. Development of best practices through programming assignments.
 - b. Prerequisites: CSCE 510 or consent of instructor
 - c. Elective Course
6. Specific goals for the course
 - a. Specific outcomes of instruction are that students will be able to:
 1. Know potential threats and vulnerabilities to software
 2. Know techniques to protect software and computer systems
 3. Develop secure software
 - b. As an elective this course cannot be counted upon to contribute to the attainment of any student outcome
7. Topics covered and approximate weight (14 weeks, 4 hours/week, 56 hours total)
 1. Week 1 Software Engineering and Security
 2. Week 2 Risk Management
 3. Week 3 Use Cases and Misuse Cases
 4. Week 4 Software Reliability
 5. Week 5 Seven Touchpoints for Software Security
 6. Week 6 Code Review; Architectural Risk Analysis
 7. Week 7 Penetration Testing, Risk-Based Security Testing
 8. Week 8 Abuse Cases; Security Requirements
 9. Week 9 Security Analysis; Taxonomy of Coding Errors
 10. Week 10 Programming Flaws; Malicious Code
 11. Week 11 -
 12. Week 13 Programming Flaws; Language-Based Vulnerabilities

Computer Engineering

Relation of Course Outcomes to EAC Student Outcomes*

Course Objectives	Program Outcomes										
	1. Logic & Math	2. Computing Fundamentals	3. Apply Computing Principles	4. Work on teams	5. Communicate Effectively	6. Liberal Arts & Soc. Sciences	7. Basic Science and Lab Procedures	8. Learn New Tools & Processes	9. Employed upon Graduation	10. Application Area	11. Electronics and Digital Sys Design
1. Know potential threats and vulnerabilities		1	3	1	1			3	2		
2. Know techniques to protect software and computer systems	1	1	3	1	1			3	2		
3. Develop secure software		1	3	1	1			3	2		

* 3 = major contributor, 2 = moderate contributor, 1 = minor contributor; blank if not related

Estimated Computing Category Content (Semester hours):

Area	Core	Advanced	Area	Core	Advanced
Algorithms		1	Data Structures		
Software Design		2	Programming Languages		
Computer Architecture					

Estimated Information Systems Category Content (Semester hours):

Area	Core	Advanced	Area	Core	Advanced
Hardware and Software			Networking and Telecommunications		1
Modern Programming Language			Analysis and Design		1
Data Management		1	Role of IS in an Organization		

Quantitative Analysis			Information Systems Environment		
--------------------------	--	--	------------------------------------	--	--