COMP 4432: Secure Coding and Testing **Spring 2018**
Syllabus

Instructor:	Andrew Neel (aneel@memphis.edu)	
Office hours:	By Appointment only (Please arrange by email 2-3 days in advance)	
Location:	Dunn Hall Room 124	
Time:	Mon/Wed 7:00PM-8:30PM	
Key Dates:	Term:	Jan 16 – May 3
	Last Day:	April 25th
	Exam 1:	February 28th
	Exam 2:	March 28th
	Exam 3:	April 23rd
	Final Exam:	Mon, April 30 @ 7PM
Holidays:	Spring Break:	March 5-11
	Study Day:	April 26th
Text:	Software Security: Building Security In by Gary R. McGraw [ISBN-13: 9780321356703]	
Text (Opt):	Secure Coding in C and C++ (2nd Edition) by Robert C. Seacord [ISBN-13: 9780321822130]	
Text (Opt):	Building Secure Software by John Viega and Gary McGraw [ISBN: 0201721152X]	

Course Description

This course covers secure programming practices necessary to develop applications against attacks and exploits. Topics covered include fundamental concepts of secure software development, defensive programming techniques, secure design and testing, and secure development methodologies. Penetration Testing Concepts: Server-side Attacks, Client-side Attacks, Web Application Testing—Fuzz Testing, File Inclusion Vulnerabilities, etc. PREREQUISITE: COMP 4081

Professional Conduct:

Students are expected to conduct themselves in a professional manner. Each student will further be held accountable to The University of Memphis's code of conduct.

Classroom Expectations

I expect each student to appear in class prepared to discuss the topics of this course. Appropriate preparation includes but is not limited to reading the text, and reviewing recommended online materials, review of source code when needed. I further expect that each student will participate in classroom discussions.

Grading:

Mastery of this courses material will be evaluated as follows:

Three (3) exams	60%
Class project	20%
Homework	20%

NOTE: I require all students to bring one blue exam booklet for themselves on exam day.

COMP 4432: Secure Coding and Testing **Spring 2018**
Syllabus

Limited Collaboration Policy:

Students are permitted and encouraged (but not required) to discuss the ideas and concepts of any classroom topic or assignment. Unless otherwise specified, the product of each assignment and test is expected to be sole, individual work each student. Specifically, students can discuss ideas and concepts but one student is not permitted to write code or prose for another student. All help is expected to be documented and credited appropriately.

> *Warning 1: Each student should accept help with care. It is very easy to mislead yourself into believing that you understand a concept when others are providing aid or assisting. In a crunch (such as an exam), this error can prove fatal.*
>
> *Warning 2: Please give help with care. Collaboration is intended to improve the classes understanding of a concept. If too much help is given, students may be enabled to fail!*

Plagiarism or cheating behavior in any form is unethical and detrimental to proper education and will not be tolerated. All work submitted by a student (projects, programming assignments, lab assignments, quizzes, tests, etc.) is expected to be a student's own original work. The plagiarism is incurred when any part of anybody else's work is passed as your own (no proper credit is listed to the sources in your own work) so the reader is led to believe it is therefore your own effort. Students are allowed and encouraged to discuss with each other and look up resources in the literature (including the internet) on their assignments, but appropriate references must be included for the materials consulted, and appropriate citations made when the material is taken verbatim.

If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructor's discretion) a failing grade in the course. The course instructor may also decide to forward the incident to the University Office of Student Conduct for further disciplinary action. For further information on U of M code of student conduct and academic discipline procedures, refer to: http://www.memphis.edu/studentconduct/academic-misconduct/process.php

Course Outline:

0. Introduction
1. Defining a discipline
   (Aside) Principles of Software Security
2. A Risk Management Framework
3. Introduction to Software Security Touchpoints
4. Code Review with a Tool
5. Architectural Risk Analysis
6. Software Penetration Testing
7. Risk-Based Security Testing
8. Abuse Cases
9. Software Security Meets Security Operations
10. An Enterprise Software Security Program
11. Knowledge for Software Security
12. A taxonomy of Coding Errors

** I reserve the right to change this course outline at any time.*