Software Security COMP4384

Instructor: Ahmed Tamrawi
Fall 2020, Revision 1

E-mail: ahmedtamrawi@gmail.com Web: `https://atamrawi.github.io/teaching/comp4384_fall20`
Private Meetings: Available upon request Class: Tuesdays,Thursdays 12:50 – 2:05pm

This course syllabus should be interpreted as a contract of understanding between students, teaching assistants, and the instructor. By participating in the course you are agreeing to this understanding. Please review this document carefully and inform the instructor of any concerns. Revisions to this document will be made as needed then posted and announced in class.

Course Description

Software is pervasive, and for better or worse it now controls our daily lives. Almost every recent computer security issue has been rooted in software, so it is critical to develop and maintain secure software. This course provides a foundation for building secure software by applying the principles and the practices of secure programming. Secure software is the umbrella term used to describe software that is engineered such that it continues to function correctly under malicious attack. To write secure software, software practitioners need to write programs in a defensive fashion, to avoid vulnerabilities that can be exploited by attackers and use security features provided by libraries, such as authentication and encryption, appropriately and effectively. Specific course topics will include: operating systems and applications security, web security, secure design and development, malware, algorithmic complexity and side channel attacks, an introduction software analysis and penetration testing, among other topics.

This course provides students with a unique experience to hone their software development and software security skills in a hands-on and fast paced environment. Course materials are designed to prepare students for the workforce with practical skills while also providing a solid foundation to continue learning and research beyond the course.

If you do not feel my goals for the course align well with your personal goals, but you need to take this course anyway to satisfy a degree requirement, you should meet with me to figure out a way to make this course useful for satisfying your personal goals.

Expected Background & Prerequisites

Students entering this course are expected to be comfortable reading, designing, and writing *C* and *Java* programs that involve code distributed over many modules. Students are expected to have some familiarity with web applications and database concepts. You should be comfortable learning how to use new programming language features and APIs by reading their documentation (or source code when no documentation is available), and not be surprised when solving programming assignments that require you to seek documentation beyond what was provided in class.

Required Materials

We will closely follow the textbook from:

- **[B1]** Michael Goodrich and Roberto Tamassia 's *"Introduction to Computer Security,"* 1st Edition.

However, we will have several readings from many other resources including:

- **[B2]** Dieter Gollmann's *"Computer Security,"* 3rd Edition.
- **[B3]** Charles P. Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies 's *"Security in Computing,"* 5th Edition.
- **[B4]** Brian Chess and Jacob West 's *"Secure Programming with Static Analysis,"* 1st Edition.
- **[B5]** Greg Hoglund, Gary McGraw 's *"Exploiting Software How to Break Code,"* 1st Edition.

- **[B6]** David Le Blanc, John Viega, and Michael Howard 's *"24 Deadly Sins of Software Security,"* 1st Edition.
- **[B7]** David Le Blanc and Michael Howard 's *"Writing Secure Code Book,"* 2nd Edition.

Course Structure

Classroom Meetings

Based on the registrar's schedule, the lecture will last for 75 minutes. We will divide it into multiple parts upon class agreement.

Assignments

There will be 3-4 assignments through the course. Assignments will consist of two types of questions: *synthesis* and *conceptual*. Synthesis questions require you to analyze a program with respect to the software security concepts and practices studied in class. Conceptual questions will reinforce important concepts and may only be practically achievable with a working software implementation. Partial credit is always available so do your best to complete as much as you can in each assignment.

Exams

There will be a final exam. Most of the questions on the exams will be taken directly from questions on the provided course notes, with a few additional synthesis questions.

Final Project

There will be a programming or a case study project that is due by the end of the semester. No obligation about the topic, but we will have several deadlines to progress the project in proper direction. Project will be judged by other classmates and staff based on final presentations. Collaboration of at most 3 students are allowed. Given the students show a fare amount of code sharing and participating in the project.

Grading Policy

I prefer to spend my time focused as much as possible on teaching, and as little as possible on grading. The quizzes, the final exam, and the final project in this class are designed to maximize learning, rather than primarily for assessment. That said, I understand that students do need to be assigned grades at the end of the semester, and sometimes grades can be a powerful and effective motivator. Grades will be determined based on your performance on the assignments and exams. The grading breakup will be tentatively as follows:

Assignments	40%
Final Project	20%
Final Exam	40%

Course Policies

Attendance Policy

Attendance in lectures is expected, and we welcome active participation (by raising questions and participating in class discussions). Please do not disturb or interrupt other classmates. You are free to leave at any time, do not feel obligated to attend the whole class.

Late Policy

Late assignments will be accepted for no penalty if a valid excuse is communicated to the instructor before the deadline. It is left to the discretion of the instructor to determine if an excuse is valid. All students are allotted a one time, no questions asked, no penalty, two day extension of exactly 48 hours from the original due date time. If a student wishes to use the one time extension the intention must be communicated to the instructor within the 48 hour window before or after the original due date time. Aside from the aforementioned exceptions, late assignments will not be graded and will result in a zero score.

Academic Integrity and Honesty

As an undergraduate student, you are trusted to be honorable. We will take advantage of this trust to provide a better learning environment for everyone. In particular, students are expected to follow these rules:

- **I will not lie, cheat or steal**. If I am unsure whether something would be considered lying, cheating or stealing, I will ask before doing it.
- **I will carefully read and follow the collaboration policy on each assignment**. I will not abuse resources, including any submissions or solutions that would be clearly detrimental to my own learning.

In addition to the honor rules, students are expected to follow these behaviors:

- **I will do what I can to help my fellow classmates learn.** Except when specifically instructed not to, this means when other students ask me for help, I will attempt to provide it. I will look at their answers and discuss what I think is good or bad about their answers. I will help others improve their work, but will not give them my answers directly. I will try to teach them what they need to know to discover solutions themselves.
- **I will ask for help.** I will make a reasonable effort to do things on my own first (or with my partners for group assignment), but will ask my classmates or the course instructor for help before getting overly frustrated.
- **I grant the course instructor permission to reproduce and distribute excerpts from my submissions for teaching purposes.** If may opt-out of this by adding a comment to your code, but without an explicit opt-out comment we assume you agree to it.
- **I will provide useful feedback.** I realize that this is a new and experimental course, and it is important that I let the course staff know what they need to improve the course. I will not wait until the end of the course to make the course staff aware of any problems. I will provide feedback either anonymously or by contacting the course staff directly. I will fill out all requested surveys honestly and thoroughly.

Course Schedule

The schedule is tentative and subject to change. An up to date version of the schedule will be posted on the course website with additional materials.

Week 1

Course Overview	
History of Computer Security	Chapter 1 [B2]
The Software Security Problem	Chapter 1 [B3], Chapter 1 [B4], Chapter 1 [B5]

Week 2

Operating Systems Concepts	Section 3.1 [B1]
Compiling and Linking	Section 3.4.1 [B1]

Week 3

Simple Buffer Overflow Attacks (Arithmetic Overflow)	Section 4.2 [B1], Section 7.1 [B4]

Week 4

Stack-Based Buffer Overflow	Section 4.3 [B1], Section 6.1 [B4]
Heap-Based Buffer Overflow	Section 4.4 [B1]
Runtime Protection	Section 7.2 [B4]

Week 5

RFormat String Attacks	Section 4.5 [B1], Section 6.2 [B4], Chapter 6 [B6]

Week 6

Privileged Programs	Chapter 12 [B4], Section 4.6 [B4]

Week 7

Network Security Concepts	Section 5.1 [B1]
The World Wide Web	Section 7.1 [B1]

Week 8

Session Hijacking	Section 7.2.1 [B1]
Phishing	Section 7.2.2 [B1]
Click-Jacking	Section 7.2.3 [B1]
Vulnerabilities in Media Content	Section 7.2.4 [B1]
Privacy Attacks	Section 7.2.5 [B1]
Cross-Site Scripting (XSS)	Section 7.2.6 [B1], Section 18.3 [B2], Section 18.4 [B2]
Cross-Site Request Forgery (CSRF)	Section 7.2.7 [B1], Section 18.5 [B2]
Defenses Against Client-Side Attacks	Section 7.2.8 [B1]

Week 9

Server-Side Scripting	Section 7.3.1 [B1]
Server-Side Script Inclusion Vulnerabilities	Section 7.3.2 [B1]
Javascript Hijacking	Section 18.6 [B2]
Databases and SQL Injection Attacks	Section 7.3.3 [B1]
Denial-of-Service Attacks	Section 7.3.4 [B1]
Web Server Privileges	Section 7.3.5 [B1]
Defenses Against Server-Side Attacks	Section 7.3.6 [B1]
XML and Web Services	Chapter 10 [B4]

Week 10

The Proactive Security Development Process	Chapter 2 [B7]
Security Principals to Live By	Chapter 3 [B7]
Threat Modeling	Chapter 4 [B7]
Handling Input	Chapter 5 [B4]

Week 11

Malware Attacks	Chapter 4 [B1]

Week 12

Mobile Malware Attacks	Provided Articles

Week 13

Algorithmic Complexity Attacks	Provided Articles

Week 14

Side Channel Attacks	Provided Articles

Week 15

Presentations

Week 16

Course Wrap-up and QA