This course covers: Fundamentals and techniques to design and implement software systems. Assessment of security vulnerabilities in software systems, exploitation of software vulnerabilities, and methods to secure vulnerable software. Secure coding practices, data analytics for security, microservices and cloud services security. Reverse engineering and security assessment of cyber-physical systems.

Department: Department of Computer Science and Computer Engineering

Credit Hours: 3

Current semester and year (start to end date): Spring 2024 (Jan 16 - May 15)

Format: face-to-face

Meetings schedule: Section 2 - TU/Th  2:30 to 3:50 pm in K150

Instructor: Dr. Lotfi ben Othmane

Office: Discovery Park E235R

Email: lotfi.benothmane@unt.edu

Office hours: Tu 4:00-5:00 / Th 4:00 - 6:00 pm and Fr 1:00 am to 22:00 (or arranged by emails)

Lab support: Mon/Wed/Fr from 2:20 pm to 5:20 pm in F223 (when there are lab assignments)

Course Goals and Learning Objectives

The goal of the course is to provide students with the knowledge and first-hand experience they need to develop secure software. The students will get familiar with exploiting software vulnerabilities and will experiment with the techniques to design secure software and to ensure the security of developed software. In addition, they will learn to use of empirical research methods to study software security challenges.

At the end of the course, the students will be able to:
- Assess the security in vulnerable software systems
- Exploit software vulnerabilities
- Apply best practices in secure software development
- Build effective cryptographic-based functionalities and assess their vulnerabilities
- Assess security implications for emerging software technologies.

Course Materials

There is no textbook. A set of papers and chapters will be distributed.


Learning Activities and Assessments

Learning Activities

To successfully complete this course, students will do the following:
• Attend the lectures or watch the recorded lectures.
• Watch additional media.
• Participate in discussion topics.
• Participate in assigned group projects.
• Complete quizzes and exams.
• Complete the project.

Assessments

Labs on software attacks - There will be 4 lab exercises that have equal weights. The labs work on Windows-based computer only. The labs count 25% of the grade. You can work on the labs in pairs or individual.

Assignments - There will be 3 practice assignments that have equal weights. The assignments count for 20% of the final grade. This is an individual work.

Project - Each team (composed by up to 4 students) should practice their knowledge in assessing the security of an open-source software or work on a research topic related to software security and submit a report at the end of the semester. The project counts 20% of the final grade.

Quizzes - There will be a set of quizzes that have equal weights, almost one quiz for each module. The dates of the quizzes will be announced as we progress in the semester. The score of the quizzes counts 25% of the final grades. Every student will have the opportunity to exclude their lowest quiz score or the score from a quiz they were absent for.

In-class exercises - There will be frequent in-class practice exercises and online participation activities. The students will submit their individual attempts to get grades and the answers will be discussed in-class. The activity counts for 10% of the grade. (The grades are given based on attempts and not correct answers.)

On certain occasions, students will be invited to come up to the board, solve problems, and get consequently rewarded with quiz bonus points.

Canvas assumes when computing the final grades that the scores of the activities within each assessment group are cumulative. Given that we will use equal weight assessment activities, Canvas grade will be an approximate of the final grade. The rules above will be applied when computing the final grades at the end of the semester.

Grading Policies

The grading schema is:

<table>
<thead>
<tr>
<th>Name</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>100 % to 93.0%</td>
</tr>
<tr>
<td>B</td>
<td>&lt; 93.0 % to 83.0%</td>
</tr>
<tr>
<td>C</td>
<td>&lt; 82.0 % to 73.0%</td>
</tr>
<tr>
<td>D</td>
<td>&lt; 73.0 % to 65.0%</td>
</tr>
<tr>
<td>F</td>
<td>&lt; 65.0 % to 0.0%</td>
</tr>
</tbody>
</table>
Grade Appeal Process

If you become concerned about the class management, please communicate your concerns with your instructor. Concerns sometimes relate to grading methods, paper turnaround time, and course policies, as examples.

Students have 7 days after returning the grades to contest their scores. Requests that come after 7 days will be ignored.

Course Policies

Feedback

All graded assessments will be returned with feedback within 10 days of the due date, when possible. Personalized feedback will be provided for each assignment and reflection. In addition, responses to common questions and unclear content will be posted at the conclusion of each module. Comments will be posted at the conclusion of each discussion.

Unclaimed student quiz sheets will be discarded one week after they are returned in class.

Missed and late coursework

It is important to keep up with the pace of this course, therefore late submissions will be reduced by a penalty of 5% for each late day up to 5 days.

Make sure to keep careful track of submission deadlines for all your work in this class.

Attendance

Attendance is not required.

Expectations

- Each student should have laptop that they could use for the in-class activities.
- Students are expected to focus on the lecture during the course sessions.

Course topics

The topics are:

1. Introduction to software security
2. Risk analysis
3. Security architecture
4. Implementing security features
5. Secure coding
6. Reverse engineering
7. Security assessment
8. Data analytics for security