CSS 577: Secure Software Development

The course is focused on techniques for developing secure software from beginning to end. Secure design and secure coding principles, practices, and methods including least privilege, threat modeling, and static analysis will be covered. Common vulnerabilities such as buffer overruns, integer overflows, injection attacks, cross-site scripting, and weak error handling will be covered in detail.

Learning Objectives

- Describe principles of secure design
- Implement software in compliance with secure design principles
- Evaluate software designs for potential vulnerabilities using formal methods such as threat modeling
- Use code reviews and static analysis to identify common secure coding vulnerabilities
- Understand and use formal methods for identifying secure coding violations
- Document software system vulnerabilities and recommend mitigations
- Understand research techniques as applied to secure development
- Understand and describe state of the art research in secure development

Grading

Activities/Homework/Projects	35%
Research	15%
Quizzes	25%
Exam	25%

Activities / Homework / Projects

You will be responsible will completing several individual and class activities including workshops, projects, and homework assignments. Individual activities will include both mandatory (required) and optional assignments.

Class activities may include small group activities to be accomplished both inside and outside of class sessions. You are expected to fully participate in small groups for all class activities. Unless otherwise instructed, you may turn in one write-up per group for most group activities. All members named in the group will receive the same score unless you choose to protest the effort made by fellow group members in writing.

Research

You will develop a research proposal in the area of secure development. You will be provided a more detailed description separately on the class website.

Quizzes and Exams

Each class will begin with a short quiz. These quizzes will be based on any material that we have covered to date in lecture or readings. They will mainly focus on material from the previous lecture or two, but I reserve the right to include anything we've covered on them.

Textbooks and Readings

- Howard, Michael; LeBlanc, David; and Viega, John. *24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them*, McGraw-Hill, 2009. (not required, but highly recommended)
- Microsoft Security Development Lifecycle, Version 5.2, Microsoft, 2010.
    - https://msdn.microsoft.com/en-us/library/windows/desktop/cc307748.aspx (Links to an external site.)Links to an external site.
- Additional Readings: Various articles, papers, and publications available through the internet or electronic reserve. Many of the links may use off-campus access of library resources. This reading list may be updated throughout the quarter.

Tentative Schedule

Week 1 -- Intro
Week 2 -- Threat Modeling and Attack Trees
Week 3 -- Secure Coding/Deadly Sins
Week 4 -- Secure Coding/Deadly Sins
Week 5 -- Secure Coding/Deadly Sins
Week 6 -- Secure Patterns
Week 7 -- Code Review
Week 8 -- Static Analysis
Week 9 -- Formal Methods
Week 10 -- Dynamic Analysis