TEXAS A&M UNIVERSITY

SYLLABUS

Course title and number	CSCE 413: Software Security
Term	Spring 2020
Meeting times and location	TR 12:45pm – 2:00pm on Zoom (https://tamu.zoom.us/j/837913136)

Course Description and Prerequisites

Defects in software are sources of vulnerabilities, which in turn are the avenues used by attackers to create and deploy exploits against software. Software defects occur along a continuum between the implementation-level and the design-level. Implementation defects, or bugs, are errors in the source code of software that can result in undefined or incorrect behavior. Design defects, or flaws, are errors in the architecture of software. Software with a flaw will have vulnerabilities even when it is implemented exactly as designed.

This course covers basic principles of design and implementation of defect-free software, code reviews including tool-assisted review by static and dynamic analysis, risk analysis and management, and methods for software security testing.

Prerequisites: CSCE 315 or approval of instructor.

Learning Outcomes or Course Objectives

Students will be able to…
list the first principles of security and **explain** why each is important to security and how it enables the development of security mechanisms that can implement desired security policies.
identify specific principles that have been violated in common security failures.
identify appropriate design principles to apply in each software development scenario.
explain the interaction between security and system usability and importance of human-computer interfaces to system usability.
explain the importance of secure software and the programming practices, development processes, and methodologies that lead to secure software.
explain techniques for specifying program behavior, the classes of well-known defects, and how they manifest themselves in various languages.
perform penetration testing on previously unknown software.
analyze existing source code for functional correctness.
analyze software for defects using industry standard tools.
develop test cases that demonstrate the existence of defects.
develop defect-free software components that satisfy their functional requirements.

Instructor Information

Name	Philip Ritchey
Telephone number	979-845-3510
Email address	pcr@tamu.edu
Office hours	T 3:30pm – 4:30pm, R 2:30pm – 4:30pm and by appointment (https://calendly.com/pcr)

Office location	HRBB 338D (Tuesdays on Zoom: https://tamu.zoom.us/j/835697225, Thursdays on Zoom: https://tamu.zoom.us/j/631454060)

Textbook and/or Resource Material

Required

24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them, Howard, LeBlanc, Viega. ISBN-13: 978-0071626750

Software Security: Building Security In. McGraw. ISBN-13: 978-0321356703

Recommended

Building Secure Software: How to Avoid Security Problems the Right Way, Viega and McGraw. ISBN-13: 978-0201721522

Exploiting Software: How to Break Code, Hoglund and McGraw. ISBN-13: 978-0201786958

Secure Coding in C and C++, Seacord, 2nd edition. ISBN-13: 978-0321822130

Grading Policies

Late and Missed Work

Excused absences allow for late work to be accepted without penalty and missed work to be made up. All other cases require prior arrangement with, and are at the discretion of, the instructor. See rule 07 of the student rules: https://student-rules.tamu.edu/rule07.

Grade Assignment and Weighting

Weight	Component	Date
10%	Participation	Weekly
5%	Homework	2 February~~, 23 February, 15 March~~
15%	Quizzes	Weekly
20%	Report	26 April
50%	Project	23 March, 19 April, 3 May

Attendance Policy

You are strongly encouraged to attend every class, arrive on time, and stay the whole time. You are responsible for learning the material covered in class regardless of your attendance.

Participation

Participation in discussion and interaction with other students are both important to your success in this course. I expect you to participate in online discussions on Piazza. Over the course of the semester, you should make at least five substantive, interesting posts to the discussion forum (either initiating a new topic or responding to someone else). These posts should be directly related to the course material. Be proactive in helping other students.

Homework

Homework assignments mirror the project, but do not share the dependency on prior work and are smaller in scope. The first homework will have the student build a secure software system that will be assessed using a combination of automated and manual testing. ~~The second homework will have the students break (find and exploit bugs) in a software system. The bugs will be verified by an automated system. The third homework will have the students fix a list of bugs found in a software system. The fixes will be verified through automated testing. The score on each assignment corresponds to the proportion of tests passed, bugs found, bugs fixed.~~

Quizzes
The topics of the quizzes will vary, including, but not limited to, the assigned reading, the homework, the project, the discussions on Piazza, and current events.

Report
The *Report* is for students to explore a topic in software security in greater depth. Students will choose a topic and select papers and other resources (e.g. blogs, webcasts, podcasts, videos) that pertain to that topic for review and summary. The report should introduce the background, present current issues in the area, discuss proposed solutions from academia and industry, and include the student's own thoughts and conclusions about the selected topic.

Project
The project is *Build it, Break it, Fix it*. Students will form teams of 3 or 4. The first phase, Build-It, is to build a secure software system (all teams implementing the same specifications). Completion of the first phase is verified by a set of automated correctness and performance tests. Team submissions must pass all required tests to "qualify" for Phase 2. Points are awarded for passing tests, efficiency, and for implementing extra features (from the specification). The second phase, Break-It) is to break the systems of other teams. All source code will be released to the teams for analysis. Teams earn points by finding, exploiting, and documenting bugs (flaws in design or implementation) in the code of other teams. Teams lose points for each bug found in their code. The third phase, Fix-It, is to fix the bugs in the team's own submission. All teams will receive the bug reports against their system and must fix as many as they can. Points are restored to the building team and deducted from the breaking teams for each non-unique bug fixed. For example, in the Break-It phase, 4 breaker teams find the same bug in Team A's code. Team A loses 400 points and each breaker team gains 100 points for that bug. During the Fix-It phase, Team A fixes that bug (and identifies all the teams which submitted the same bug) and recovers 300 points while the breaker teams each lose 75 points. That is, the builder team loses 100 points and each breaker team earn 25 points for that unique bug. This helps to emphasizes the value of building security into software rather than patching it in later. Teams which write fewer bugs will lose fewer points. At the end, the team with the most points wins bragging rights.
Doing the minimum required for each phase earns an amount of points equal to 80% of the possible points (not counting bonuses) available in that phase. Additional points are earned by going above and beyond minimum requirements.
You are **required** to use the Texas A&M institutional GitHub service (https://github.tamu.edu) for the project. You must add the instructor and TA/grader(s) to your repository. Your repository is documentation of your development process. You are required to have clear commit messages that document the bugs you identified, corresponding fixes, addition of test cases, etc. Your GitHub usage will impact your grade. For example, having all the code added to the repository and submitted within a short period of time (the magic of going from no code to a working solution quickly) will result in a very low grade.

Grading Scale

Final letter grades will be assigned according to the following cutoffs:

A ≥ 90	Superior
B ≥ 80	Satisfactory
C ≥ 70	Needs Improvement
D ≥ 60	Unsatisfactory
F < 60	Unacceptable

The cumulative numerical grades from graded assignments are normalized to 100 points.

Course Topics, Calendar of Activities, Major Assignment Dates

TAMU Academic Calendar
https://registrar.tamu.edu/Catalogs,-Policies-Procedures/Academic-Calendar

Tentative Course Topics and Calendar of Activities

Week	Topic	Required Reading
1	Introduction to Software Security	SSBSI 1: Defining a Discipline
2	**RVV**. Security Requirements	SSBSI 8: Abuse Cases Core Security Requirements Artefacts Requirements Engineering for Survivable Systems
3	**RM**. Risk Management Framework	SSBSI 2: A Risk Management Framework
4	**D**. Secure Design Principles	Principles of Computer System Design 11.1.4: Design Principles
5	**T**. Static Analysis	SSBSI 4: Code Review with a Tool A Few Billion Lines of Code Later: Using Static Analysis to Find Bugs in the Real World
6	**SC**. Risky Resource Management	SANS Top 25 Software Errors 24DSSS 5: Buffer Overruns 24DSSS 6: Format String Problems 24DSSS 7: Integer Overflows 24DSSS 18: The Sins of Mobile Code
7	**RVV**. Software Verification and Validation	The verifying compiler: A grand challenge for computing research (watch the lecture at Gresham College) Hacker-Proof Coding
8	**RM**. Architectural Risk Analysis / Threat Modeling	SSBSI 5: Architectural Risk Analysis Planning Poker Protection Poker (tutorial)

9	**D**. Secure Design Patterns	Secure Design Patterns Software-security patterns: degree of maturity
10	**T**. Peer Code Review	Best-Kept Secrets of Peer Code Review
11	**SC**. Porous Defenses	SANS Top 25 Software Errors 24DSSS 16: Executing Code with Too Much Privilege 24DSSS 17: Failure to Protect Stored Data 24DSSS 21: Using Cryptography Incorrectly
12	**T**. Dynamic Analysis, Fuzzing	KLEE SAGE: Whitebox Fuzzing for Security Testing
13	Thanksgiving Break	Alice's Restaurant
14	**SC**. Insecure Interaction Between Components	SANS Top 25 Software Errors 24DSSS 1: SQL Injection 24DSSS 2: Web Server-Related Vulnerabilities (XSS, CSRF, Response Splitting) 24DSSS 3: Web Client-Related Vulnerabilities (XSS)

Topic Legend
D: Design **SC**: Secure Coding
RM: Risk Management **T**: Testing
RVV: Requirements, Verification, Validation

Major Assignment Dates

Weekly:	Participation
	Quiz
2 Feb	Homework 1 due
~~23 Feb~~	~~Homework 2 due~~
~~15 Mar~~	~~Homework 3 due~~
23 Mar	Project Build-It due
19 Apr	Project Break-It due
26 Apr	Report Due
3 May	Project Fix-It due

Additional Pertinent Course Information

Typesetting
All written (i.e. non-coding) homework must be typed.
You are strongly encouraged to typeset your work using LaTeX.
Resources for LaTeX can be found on the course website and on the Internet.
Microsoft Word and OpenOffice Write are acceptable, yet vastly inferior, alternatives.

Version Control
You are **required** to use the Texas A&M institutional GitHub service (https://github.tamu.edu) for the project. You are strongly encouraged to use git for your homework and report, as well.

Submission to eCampus
All assignments will be submitted through eCampus (https://ecampus.tamu.edu). Written assignments must be submitted as PDFs. Submission of source code must follow the instructions in the homework or project specifications.

Piazza
Some discussions and announcements will take place on Piazza (https://piazza.com). You should check Piazza often (e.g. every weekday). All questions and comments about the course should be posted on Piazza. Piazza is designed and managed so that you can get help quickly and efficiently from classmates, the PTs, the graders, the TAs, and me. In this class, communication will work as with a large team in a company: everyone has an obligation to chime in and help others by providing information. We will use e-mail only for student-sensitive concerns. If you e-mail a question or comment about the course to me or a TA, you will very likely be redirected to Piazza.

E-mail Formatting
When you send email to me or a TA, the subject must be prefixed with [CSCE 413] and you must sign your name to the email. Putting [CSCE 413] in the subject will let us know in which course of ours you are enrolled. Signing your name will let us know who you are. If you do not sign your name, we may assign you one at random in our reply. You are encouraged to encrypt and sign all emails to me. My PGP public key is on my home page and the MIT key server (https://pgp.mit.edu).

Discussion of Grades
Federal law prohibits the instructor, TAs, and graders from discussing grades over non-TAMU email. If you have a question about your grade, the preferred mechanism is to meet with us in-person, such as during office hours.

Harassment and Discrimination
Texas A&M is committed to the fundamental principles of academic freedom, equality of opportunity and human dignity. To fulfill its multiple missions as an institution of higher learning, Texas A&M encourages a climate that values and nurtures collegiality, diversity, pluralism and the uniqueness of the individual within our state, nation and world. All decisions and actions involving students and employees should be based on applicable law and individual merit.

Texas A&M University prohibits harassment and discrimination against any member of the University community on the basis of race, religion, color, sex, age, national origin or ancestry, genetic information, marital status, parental status, sexual orientation, gender identity and expression, disability, or status as a veteran.

Students who believe they have experienced harassment or discrimination prohibited by this statement are encouraged to contact the Office of the Dean of Student Life at 979-845-3113.

Americans with Disabilities Act (ADA) Policy Statement
The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you believe you have a disability requiring an accommodation, please contact Disability Services, currently located in the Disability Services building at the Student Services at White Creek complex on west campus or call 979-845-1637. For additional information, visit http://disability.tamu.edu.
Academic Integrity
An Aggie does not lie, cheat, or steal, or tolerate those who do. For all academic work in this and every course, it is expected of you that you shall neither give nor receive any unauthorized aid. All violations of the Aggie code of Honor will be reported to the Aggie Honor System Office. For this course, a significant amount of work will require solving problems for which a solution or test data might be available or posted online. Unless otherwise specified, students are not allowed to seek out or examine code/data for these problems on their own, prior to turning in their own solutions. Doing so will be considered a violation of the honor code, and students caught doing so will be referred to the Aggie Honor System Office, regardless of whether the actual code or data is copied or not. For more information, see https://aggiehonor.tamu.edu/.