CSci 4271W: Development of Secure Software Systems

Instructor: Stephen McCamant
Office: 4-225E Keller Hall
E-Mail: mccamant@cs.umn.edu
Home page: http://www.cs.umn.edu/~mccamant
Office hours: Mondays 2-3pm, in person. Also by appointment.

Lecture Schedule: Tuesdays and Thursdays, 11:15am-12:30pm, 231 Smith Hall

Lab Schedule: Section 002: Mondays 10:10-11:00am, 1-262 Keller
Section 003: Mondays 11:15am-12:05pm, 1-262 Keller
Section 004: Mondays 12:20-1:10pm, 1-262 Keller

Course Overview:
Computer Science 4271W is an elective-level undergraduate course that introduces concepts of computer security with particular emphasis on the design and implementation of secure software. The course covers techniques to identify and exploit security threats, as well as prevent or mitigate them, during software design, coding, testing, and maintenance. It describes general patterns of vulnerabilities as well specific problems in low-level software, operating system interaction, use of cryptography, networked and web systems, identity and authentication, and usability. As a writing intensive course, students will also learn and practice techniques for effective written communication about software security.

Goals and Objectives:
By the end of this course you should be able to:

- Describe a system and its threat model using appropriate diagrams and threat classification.
- Recognize common vulnerabilities in protocols, designs and programs.
- Describe methods to detect and mitigate these vulnerabilities, and their limitations.
- Apply principles and standard processes to design more secure software components.

These objectives address two student learning outcomes:

Can identify, define, and solve problems: In assignments and exercises you will be given descriptions of software systems and asked to identify potential security vulnerabilities, and methods to mitigate these vulnerabilities.

Can communicate effectively: It is not enough that individual developers understand security when coding. Issues such as the organization of a complex software system, how potentially adversarial inputs and actors can affect the system, and methods to mitigate these threats require communication among developers and between developers and other stakeholders. If a security-conscious developer cannot convey why a threat should be considered, or the way it works, or how to address a threat, the larger organization may still fail to devote enough effort to security improvement. Developing skills to communicate about software and threats is thus just as important as developing skills to find and address these threats. In this class you will regularly practice writing such descriptions in problem sets and project reports.

Prerequisites:
The prerequisite for this course is CSci 3081W, Software Design and Development, and the course also depends on 3081's own transitive prerequisites. Students should be comfortable reading and writing code; especially in C, but other languages will also be used on occasion such as in labs. Students should also have experience working with the Unix/Linux command line. The portions of the course that deal with low-level attacks and defenses on software (buffer overflows, etc.), depend on some familiarity with assembly language and how C programs are compiled. These are covered in CSci 2021, which is transitively a prerequisite for 3081, but this may be an area you should consider reviewing if a while has passed since you took 2021. EE 2361 covers many of the same relevant topics from CSci 2021, but because it uses a micro-controller architecture, you may want to take some time outside of class to learn about the x86-64 architecture we'll use. Portions of the course cover security as applied to operating systems, networks, web programming, and databases, but the courses in those areas (CSci 4061, CSci 4211, CSci 4131, and CSci 4707), while potentially helpful, are not prerequisites. We will introduce some basic principles of these areas as relevant to security threats against them.

Reading Schedule:
The course web site will have a schedule of the topics that will be covered in each lecture. For some subject areas there will also be relating reading materials covering the same topics. You will get the most out of the lecture by reading the materials before coming to class. The material in the readings and the materials in lectures will not match exactly: while the most important material will appear in both, some points may appear only in the reading or only in the lecture, and they're all fair game to appear on exams. We'll post the slides from lectures on the web site, but reviewing the slides is intended to supplement, not replace, attending lecture.

Textbooks:
Because the area of computer security is changing quickly, it is hard to produce a good textbook. I haven't yet found a textbook that I like and which covers most or all of the material in the class, so the course has no required textbooks. We will draw readings from a variety of other public

source which are either free of charge to everyone or available via University licensing. However if you would like to look over books relevant to the course material, here are two optional recommendations:

Ross Anderson's *Security Engineering*, Third Edition (John Wiley & Sons) is a great book for understanding the adversarial perspective in security; it covers many of the course's main topics at a high level, as well as drawing on examples of security thinking outside strictly computer applications. It's also a lot more fun to read than the average CS textbook. However it doesn't go into as much technical detail as we will in many areas. I use it more heavily in 5271, so you may be able to find it via the campus bookstore, or various places online. The third edition includes some notable improvements since the previous second edition, but the majority of relevant material is also found in that edition. An advantage of the previous edition is that all the chapters are available online from the author's home page. Also, the campus libraries have acquired ebook access to an HTML edition of the book through the publisher's online portal. As the first edition is even older, it isn't recommended.

Adam Shostack's *Threat Modeling: Designing for Security* (Wiley), as the title suggests, focuses on threat modeling and security design concerns. The approach to threat modeling and dataflow diagrams we follow is based on Shostack's, so you can read this book if you are interested in a lot more detail on these topics than we'll have time for in class.

Mark Dowd, John McDonald, and Justin Schuh's *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities* has the most in-depth coverage I'm aware of on the process of auditing code for vulnerabilities, and related topics, so in some ways its the book that is closest to the course in topic coverage. However it is too out of date (2006), too long (1200 pages across two paper volumes), and too out of print (only an ebook is available for a reasonable price) for me to wholly recommend. I'll assign parts of one of the most relevant chapters as a reading.

Including chapters from Anderson as discussed above, all the class readings are available to download and/or read online. Many are public downloads; a few, will have course-specific links from the Canvas page, or others, such as some academic papers, are licensed to the University via the libraries so you can access them directly if you're coming from a campus IP address, or from off campus you can use the library's proxy service and bookmarklet. Some other online resources will be mentioned in lecture notes, assignments, or on the class forum, and of course there's lots of information, some of it good, available online that you can find yourself.

Grading:
Grading for this course will be based on the following components:

- Lab assignment participation (10%): Weekly on Mondays, a 50-minute lab section will give you hands-on experience with security concerns and tools. We recommend that you work in groups on for this labs: this semester there should be enough computers in the lab for everyone who is registered. The lab sessions are graded based on participation, meaning that you don't have to achieve any particular level of success to get credit: you just have to show to a TA that you're spending time working on the lab material. Please take advantage of the TAs to answer questions and get advice anyway; this will also be an opportunity for them to record your participation. If you're able to complete all the lab activities with no help and time to spare, just let a TA know about your success before you leave. Conversely, there may be more questions brought up in the lab than you have time to get to; it is in your interest to keep thinking about the material from the lab beyond the lab period, since the labs introduce skills and concepts you'll use elsewhere. Ask questions about lab material on Piazza or in office hours. Lab participation counts for 10% of your overall course grade, and the three lowest lab participation scores will be dropped from the computation, so in other words you can miss up to three labs without penalty.
- Online reading questions (6%): To check you understanding of assigned readings, some readings will be followed up with short, automatically-graded online quizzes with questions about what you've read. You'll need to complete the quiz no later than a week after the date when we recommend you do the reading, but you can repeat the quiz to improve your score.
- Problem sets (10%): Problem sets are individual written homework assignments that primarily ask you to engage with topics from the course and write about them, rather than programming or working primarily on the computer. Questions will be posted on the assignments page of the public course web site. Please type up your answers with your favorite word processor or text editor, and submit them as a PDF document online (via a submission page reachable from Canvas). The greater part of problem set evaluation (75%) will be based on the technical correctness of your answers, but take care to also express your ideas clearly; 25% of the score will be assigned based on the effectiveness of your writing. There will be either 2 or 3 problem sets over the semester. Problem sets that are late, but less than 48 hours late, are eligible for half credit; no credit is available after 48 hours from the due date.
- In-class midterm exams (14%): There will be two midterm exams held during the normal class period: one on Tuesday February 20th, and one on Tuesdayy, April 9th. The exams will be open book and open notes, and in fact any paper resources may be used, but no electronic devices may be used. Note there will be no final exam.
- Security assessment projects (60%): The most important but probably also most time-consuming part of the course will be series of projects in which you will work in depth on the security of a software system. Depending on the project, these will include modeling the possible threats against a system, finding security vulnerabilities in software and testing how they can be exploited, writing a document describing your findings, and fixing the bugs you have found. The projects are also the part of the course where writing is most important: in total, half of your project grade will be based on whether your writing conveys your ideas clearly and professionally. We will give you detailed feedback on your writing in all of the projects, and for project 1 you will also get a chance to submit a revised written report later incorporating feedback. The project reports will be 4-5 pages long in a format that is double-spaced but with a small font size, so you will have to write a substantial quantity of text but also express your results concisely. The written project submissions will all be individual assignments. The normal deadlines for project report submissions will be on Friday evenings, but for one submission over the course of the semester you will be able to receive an extension to the following Monday evening.

Letter grades will be assigned using the following scale:

Grade	Minimum Score
A	92.00
A-	88.00
B+	84.00

B	80.00
B-	76.00
C+	72.00
C	68.00
C-	64.00
D+	60.00
D	56.00
F	(below 56.00)

Collaboration and External Sources:

Discussing the course content with other students can be a very useful part of your learning experience: that is part of why we encourage you to collaborate in the labs, for instance. However it is up to you to structure that collaboration for the best results and to maximize everyone's learning.

On the other hand, we ask that you be more circumspect in discussing problem sets and projects with other students, so that everyone has the chance to grapple with the challenges on their own. The class isn't graded on a curve, so you aren't competing with other students: their failure will not improve your grade, and we'd like to maintain a generally friendly atmosphere. For instance it's okay to discuss ideas from the course at a general level that might apply to a homework (e.g., the concept of a buffer-overflow attack), or to help another group with a technical problem unrelated to the project itself ("I can't get my VM to boot"). But you should avoid giving specific suggestions that another student could use in place of figuring something out on their own ("you can overflow buffer foo by passing a 200-byte string to function bar"). And sharing code or prose that would be given as an answer to an assignment question is of course never okay.

Many assignments in the class will allow or even encourage the use of resources beyond the course readings and lecture notes, such as you might find in the library or on the Internet. However it is an important academic value, which we enforce rigorously in this class, that it is never acceptable to use another's work without properly acknowledging it. In problem sets and projects, you should acknowledge any external sources of inspiration or code directly in your answer. Failure to do so constitutes plagiarism.

Use of Software Tools and Artificial Intelligence:

Software tools can be an important part of work in computer security. But in some cases we will specifically ask that you do assignments without the most powerful tools, so you can get a better understanding for yourself of what's going on. Check the instructions for individual assignments or ask before using a tool if you are unsure. You should be particularly careful with AI-based tools that provide a general question-answering capability (include large language models like ChatGPT), since they can more easily be used in ways that would bypass the learning intent of an assignment. A good starting point is to think about consulting an AI tool in the same way you would think about asking another person who is not the course staff a question. It is fine to ask questions of an AI for your general understanding of a concept, or on a non-graded activity like a lab where we also encourage human collaboration. But don't use an AI tool in a way that substitutes for your won understanding or effort in a graded assignment. Of course it is also in your interest to watch out for the possibility of an AI-tool providing incorrect information. Our default policy for written assignments that are submitted electronically is that you can use spell-checking or grammar-checking tools that given advice on the mechanics of writing, but not more sophisticated AI tools that could understand the subject matter of the assignment. As an exception, you will be allowed to make free use of AI-based tools on the projects, as long as you explain what help they provided. This will be described in more detail in those assignments.

Academic Integrity Policies: By the nature of this class, we will often discuss techniques that could be used to compromise the security of certain computer systems. However, IT IS VERY IMPORTANT THAT YOU NEVER APPLY THESE TECHNIQUES TO A COMPUTER WITHOUT THE PERMISSION OF THE COMPUTER'S OWNER. In particular you should never attempt to attack the security of computers that belong to CSE Labs, the department, the University, or an unsuspecting classmate. If we learn that a student has unethically exploited a vulnerability discussed in class, THAT STUDENT WILL FAIL. This is in addition to any University-level, department-level or legal penalties such an action may be subject to.

You are also expected to do your own academic work and cite sources as appropriate. Failing to do so is scholastic dishonesty. Scholastic dishonesty includes, but is not limited to: plagiarizing; cheating on assignments or examinations; engaging in unauthorized collaboration on academic work; taking, acquiring, or using test materials without faculty permission; submitting false or incomplete records of academic achievement; acting alone or in cooperation with another to falsify records or to obtain dishonestly grades, honors, awards, or professional endorsement; altering, forging, or misusing a University academic record; or fabricating or falsifying data, research procedures, or data analysis. A student found responsible for scholastic dishonesty will at a minimum receive a grade of 0 for the assignment in question and be reported to the campus-wide Office for Community Standards (OCS). More serious offenses will receive a grade of F (or N) for the course and be subject to additional sanctions from the University.

Other Applicable Policies: There are a number of other University-wide policies that apply to this course which you should be familiar with. This list is an abridged summary of longer policies which you can find linked from a University-wide page:

- Students are required to abide by the Student Conduct Code, which among other things prohibits disruptive classroom conduct.
- Personal electronic devices should be used with caution in the classroom lest they interfere with your or other students' learning.
- Students will not be penalized for absence during the semester due to unavoidable or legitimate circumstances. Such circumstances include verified illness, participation in intercollegiate athletic events, subpoenas, jury duty, military service, bereavement, and religious observances. The requirement of verification for absences due to illness is waived for a single episode absence that did not require professional treatment, and did not lead to missing an important in-class event such as an exam.

- The University considers that accepting compensation for taking and distributing classroom notes violates shared norms and standards of the academic community.
- Sexual harassment is not acceptable in the University setting.
- The University provides equal access to and opportunity in its programs and facilities, without regard to race, color, creed, religion, national origin, gender, age, marital status, disability, public assistance status, veteran status, sexual orientation, gender identity, or gender expression.
- The University of Minnesota is committed to providing equitable access to learning opportunities for all students, including making reasonable accommodations. If you have, or think you may have, a disability that might affect your participation in class please contact the Disability Resource Center. If you are registered with the DRC and have a current letter requesting reasonable accommodations, please contact your instructor as early in the semester as possible to discuss how the accommodations will be applied in the course. If you are not registered with the DRC and are experiencing or think you may be experiencing disability related to a mental health, attention, learning, chronic health, sensory, or physical condition, and would like to discuss accommodations and/or resources, please contact the DRC. If you have a short-term medical condition, such as a broken arm, I may be able to assist in minimizing classroom barriers directly. But if any additional assistance is needed, you should contact the DRC.
- As a student you may experience a range of mental health concerns or stressful events which may interfere with learning. You can learn more about the broad range of confidential mental health services available on campus via the Student Mental Health website. You can also refer to a list of mental health resources maintained by the CS&E department, this search site with international resources, or this list of international and online resources.
- Within the scope and content of the course as defined by the instructor, academic freedom includes the freedom to discuss relevant matters in the classroom and conduct relevant research. Students are free to take reasoned exception to the views offered in any course of study and to reserve judgment about matters of opinion, but they are responsible for learning the content of any course of study for which they are enrolled. (Adapted from The AAUP Joint Statement on Rights and Freedoms of Students.)

(This syllabus is based in part on documents used in previous editions of 4271 and 5271 written by the instructor and by Prof. Nick Hopper.)

- Last modified on February 23, 2024