CSci 4271W
Development of Secure Software Systems
Day 1: Introduction and logistics

Stephen McCamant (he/him)
University of Minnesota, Computer Science & Engineering

Outline

Big-picture introduction

Discussion group greetings

Course logistics

What is computer security?

- Keep "bad things" from happening
- Distinguished by presence of an adversary

Two sides of security

- Defenders / white-hats / good guys[sic]
- Attackers / black-hats / bad guys[sic]
- Each side's strategy depends on the other
- In some ways like a game

Common security threats

- Spoofing
- Tampering
- Repudiation
- Information disclosure
- Denial of service
- Elevation of privilege

Threat modeling

- What are the relevant parts of your system?
- What threats are possible?
- How can you stop the threats?

Course areas

- Low-level software security
- OS interaction security
- Web software security
- Using cryptography
- User identities and usability

Outline

Big-picture introduction

Discussion group greetings

Course logistics

Say hello to your neighbors

- From time to time I'll ask you to do discussions or exercises in groups with people sitting near you
- For today, just introduce yourself to the folks sitting nearby

Outline

Big-picture introduction

Discussion group greetings

Course logistics

Instructor information

- Stephen McCamant
- Office: 4-225E Keller (most days)
- Office hours: today 5:30pm (after class), future weeks TBA
- Email: `mccamant@cs.umn.edu`

Teaching assistants

- Bowen Wang, Derek Wong
- Office hours: TBA

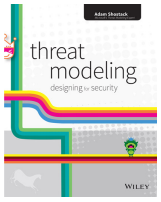
Prerequisites

- Software design and development (3081)
- C, machine code, and compilation
  - E.g. 2021, transitive for 3081

Reading materials

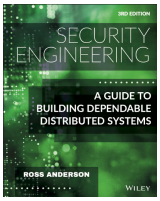
- Posted on the course web site
- Download, perhaps with library proxy
- Chosen to complement lecture discussions
- Comprehension questions on Canvas

Optional book 1



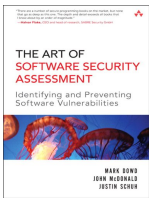
Provides more detail on threat modeling, but no assigned readings

Optional book 2



Source for several readings, but chapters are free online

Optional book 3



Lots of detail about software auditing, but old and out of print

Evaluation components

10% Lab participation
6% Online reading Qs (best scores)
10% Written homework / problem sets
14% Two in-class midterms
60% Projects

Online reading questions

- Auto-graded questions to check your understanding
- Due within a week from the material posting
- Can repeat to improve your score

Problem sets

- 2-3 sets, roughly by topic areas
- Done individually
- Mostly thinking and writing, not much programming
- Submit in PDF online
- 75% technical correctness, 25% writing

Midterm exams

- Two in-class exams, in October and November
- Open-book, open-notes, but no electronics
- No final exam

Projects

- Single most important and time-consuming part of course
- Each may cover:
  - Modeling possible threats against a system
  - Finding bugs and testing attacks
  - 4-5 page writeup of your results, with revision
  - Fixing the bugs
- Mostly individual, 50% of grade is writing

1.5 projects

- Proj 0.5: memory safety vulnerabilities, smaller
- Proj 1: memory safety vulnerabilities, full size

Project 0.5: BCBASIC

- Badly Coded BASIC interpreter
- Audit code to find a vulnerability and produce one proof-of-concept attack
- Vulnerability-finding in groups, writeup is individual

Project 1: BClmgView

- Badly Coded Image Viewer
- Larger, handles multiple file formats and has multiple vulnerabilities
- Earlier step: threat modeling
- Later steps: propose code fixes, revise your report
- All individual

Writing intensive

- A major focus is effectively communicating about security
- Writing techniques will be a periodic topic in lectures
- Lots of feedback (and grading) about writing assignments
  - Project 1 includes revision in response to feedback

Late assignments

- Problem sets: half credit for up to 48 hours late
- Projects: may request an extension (from Friday night to Monday night) for one project submission

Collaboration

- Be careful about bugs: "no spoilers"
- OK to discuss general concepts
- OK to help with side tech issues
- Sharing code or written answers is never OK

External sources

- Many assignments will allow or recommend outside (library, Internet) sources
- But you must appropriately acknowledge any outside sources you use
- Failure to do so is plagiarism

What about AI?

- General principle: what if you got similar help from a person outside the class?
  - Okay to use for concept understanding, or non-graded activities
  - Not okay to substitute for your own understanding or effort in graded assignments
- Also beware the AI's answers might not be right!

Exception: AI in the projects

- For now, the projects are beyond what AI can do on its own
- AI tools can also be a resource to help with writing
- So, AI tools will be allowed on projects, as long as you give credit for what they did
- Trying to make an AI do the whole project is not recommended, but you can try

Security ethics

- Don't use techniques discussed in class to attack the security of other people's computers!
- If we find you do, you will fail, along with other applicable penalties

Academic misconduct generally

- Don't cheat, plagiarize, help others cheat, etc.
- Minimum penalty: 0 on assignment, report to OCS
- More serious: F in course, other OCS penalties

Course web site

- Department web site is under `csci4271`
- Also linked from my home page `~mccamant`

On Canvas

- Online lecture/reading questions
- Assignment submissions (or Gradescope)
- Viewing grades
- Zoom links (only if needed)

Mostly Piazza

- Online Q&A
  - Can be anonymous and/or private
  - Both students and staff can answer
- Course announcements
  - Can control delivery preferences, defaults to email
- Reserve email for personal, administrative issues

In-person lecture/discussions

- TuTh 4:00–5:15pm in 231 Smith
- Mixture of lecture and discussions
  - Come prepared to participate
- Lecture slides posted

Lab sections

- Hands-on and collaborative practice with code and tools
- Wednesday afternoon/evenings in 1-250 Keller
- Graded on participation, meaning:
  - Be present and working on 4271 material
  - If you have a question, that interaction counts
  - No questions? Show off your progress

First lab tomorrow

- No security content, just practice with background and logistics skills
  - Recommended to work in small groups
- Vole (FastX) and SSH access to CSE Labs
- Read-only screen sharing via Zoom
- Interactive terminal sharing via `tmate`
- Off-campus access to library materials

4271 vs. 5271

- Designed so you can take either or both
  - 5271 easier but still worthwhile after 4271
- 4271 has more of: threat modeling, software engineering, writing support
- 5271 has more of: research perspectives, novel/difficult attacks

Challenging course aspects

- Stressing C, low-level, and Unix skills
- Thinking like an attacker
- Time/project management

Large language model Q&A

- Explore a bit about what questions are easy or hard

Detailed material starts Thursday

- I'll see in you in lab Wednesday and here again Thursday