What is computer security?
- Keep "bad things" from happening
- Distinguished by presence of an adversary

Two sides of security
- Defenders / white-hats / good guys
- Attackers / black-hats / bad guys
- 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

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: this Thursday 5:30pm (after class), future weeks TBA
- Email: mccamant@cs.umn.edu

Teaching assistants

- Bowen Wang, Jack Tschetter
- Office hours: TBA

Prerequisites

- Software design and development (3081)
- C, machine code, and compilation
  - E.g. 2021, transitive for 3081
- Occasionally relevant, not required: 4061 (OS), 4211 (networks), 4707 (databases), 4131 (web programming)

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, Tuesday February 20th and Tuesday April 9th
- 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
- Tentative due date: Friday, February 23rd
Project 1: BCImgView

- 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 11:15am-12:30pm 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
- Monday morning/noon in 1-262 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

No lab this week

- Material on online interaction will be on the course web site
- Vole and SSH access to CSE Labs (review)
- 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
<table>
<thead>
<tr>
<th>Challenging course aspects</th>
<th>Detailed material starts Thursday</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Stressing C, low-level, and Unix skills</td>
<td>- Most other lectures won’t end early</td>
</tr>
<tr>
<td>- Thinking like an attacker</td>
<td>- I’ll see in you here again Thursday</td>
</tr>
<tr>
<td>- Time/project management</td>
<td></td>
</tr>
</tbody>
</table>