

CSci 5271  
Introduction to Computer Security  
Day 1: Introduction and Logistics

Stephen McCamant  
University of Minnesota, Computer Science & Engineering

## Outline

COVID-19-related Logistics

Big-Picture Introduction

Course Logistics

## Masks required

- You should be wearing a mask, cloth face covering, etc., covering your nose and mouth
  - So, don't plan on eating lunch in class

## Vaccination

- Getting a vaccine is the single most important thing you can do to reduce your risk of serious illness from COVID-19
  - And reduces risk of spreading to others, too
- I request that you cooperate promptly with the U's COVID-19 vaccine requirement
  - Though I am not otherwise involved

## Viewing lectures online

- Lectures will be live-streamed and recorded by the UNITE program
  - Unlike previous semesters, available right away to on-campus section
- Can be an alternative if it feels safer than coming to lecture in person
- If you're watching online today, email me to confirm your interest in the course

## Other things you can do

- Be more careful than usual to stay home if you're sick
- Sit in a consistent place in the classroom
- Exchange contact info with the students near you, or install a contact-tracing app on your phone

## Group work is up to you

- 5271 has optional and required group work, in groups of your choice
- Group work can be entirely online if your group prefers
  - Including meetings with me and presentations over Zoom

## Exams are still in person

- Exams for the on-campus section are scheduled in-person, in this room
  - UNITE section has separate procedures, still synchronous

## My precautions in class

- For subsequent lectures by default choice will be a cloth face mask
- Next lecture, I will experiment with a clear face shield
  - Might make me easier to understand, but will only keep if its a lot better

## Be prepared for change

- I'm optimistic the pandemic will improve before the end of the semester
- If it gets too much worse, we may shift to all-online
  - Likely based on campus-wide decisions
  - If we switch to Zoom, links will on the course Canvas page

## Outline

COVID-19-related Logistics

Big-Picture Introduction

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

## Classic security goals

- Confidentiality
- Integrity
- Authenticity
- Availability

## Managing risk

- Threat model, likely adversary goals
- Expected damage
- Expected attack rate

## Course areas

- Software security
- OS security
- Cryptography
- Network application security
- Other topics

## Software security

- Security bugs aka *vulnerabilities*
  - Some specific to low-level languages like C, others not
- Arms race
  - Attack techniques
  - Defenses against unknown bugs
  - Countermeasures against defenses
- Defensive programming and design

## OS security

- Classic area for secure design and security policies
  - Some specific examples from Unix/Linux
- Access control and capabilities
- Multi-level security and information flow
- Assurance and trust

## Cryptography

- Mathematical techniques for protecting information
- Symmetric-key techniques (e.g. AES)
- Public-key techniques (e.g. RSA)
- Cryptographic protocols
- What can go wrong (lots!)

## Security and the network

- Network protocols, basic and “S”
- Firewalls, NATs, intrusion detectors
- Web servers and web clients
- Network malware and network DoS

## Short topics

- Privacy-enhancing network overlays
- Security and usability
- Electronic voting
- Electronic cash (e.g., Bitcoin)

## Learning goals

- Think like your adversary
- Recognize and eliminate vulnerabilities
- Design and build systems securely
- Apply security principles to research problems

## Outline

COVID-19-related Logistics

Big-Picture Introduction

Course Logistics

## Instructor information

- Stephen McCamant
- Office: 4-225E Keller
- Office hours: Monday 1-2pm in person, Wednesday 11:30-12:30 on Zoom, or by appointment
- Email: [mccamant@cs.umn.edu](mailto:mccamant@cs.umn.edu)

## Teaching assistants

- Andrew (Dong) Bao
  - Office hours Wednesday 4-6pm in person, location TBA
- Qiuchen Yan
  - Office hours Thursday 3-4pm on Zoom (see Canvas)

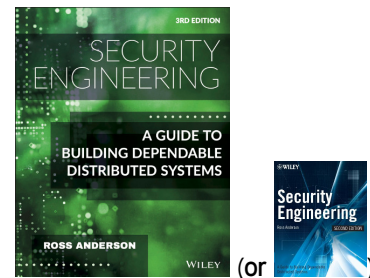
## Prerequisites

- Undergraduate-level OS, e.g. 4061
- Machine code and compilation
  - E.g. 2021, transitive for 4061
- Useful: networks (421f)
- Graduate level maturity and resourcefulness
- C, Unix, (Perl | Python | Ruby | ...)

## Reading materials

- Posted on the course web site
- Download, perhaps with library proxy
- Read before corresponding lecture
- Readings and lecture may not match
  - Both may appear on exams

## Textbook



(or

## Evaluation components

- 10% Written exercise sets (5)
- 15% Hands-on assignments (2)
- 20% Midterm exam
- 25% Final exam
- 30% Group research project

## Exercises

- Five sets, roughly by topic areas
- Do individually or in groups of up to 3
- Mostly thinking and writing, not much programming
- Submit one set per group in PDF, via Canvas/Gradescope

## Hands-on assignments

- Two assignments, by large topic divisions
- Do individually or in groups of up to 3
- Mostly programming and attacking
- Draws heavily on your C and Unix skills
- First up: penetrate-and-patch HA1

## Exams

- Open book, open notes, no laptops/calculators/phones
- Mix of multiple-choice/true-false and short-answer
- Midterm: Monday October 25th in class
- Final: Saturday December 18th 1:30-3:30pm
- Mark your calendars!

## Group research project

- Single most important and time-consuming part of course
- Groups of 4-6, preferably 5 or 6
- Engage with a recent research paper
  - Reproduce and extend, or
  - Reproduce and attack

## Project milestones

- Pre-proposal (due Sep. 22)
- Progress meetings and reports (monthly)
- Short in-class presentation (last two weeks)
- Paper-style final report (due Dec. 15)

## Pre-proposal (Sep. 22)

- Who: group members
- What: paper you're engaging with
- Why: are you suited for this project
- How: preliminary action plan
- When: available times for progress meetings

## Project evaluation

- 15% Originality
- 15% Scholarship
- 30% Strength of evaluation
- 40% Individual contribution

## Late assignments

- Due dates usually 11:59pm Central Time
- 1 sec late - 23:59:59 late: 75%
- 24 hrs - 47:59:59 late: 50%
- 48 hrs - 71:59:59 late: 25%
- After that: 0

## Collaboration, within groups

- Main kind of collaboration expected in class
- Think about how you structure your collaboration
- For best results, but also to learn from teammates

## Collaboration, between groups

- Be careful: "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**

## 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 under `csci5271`
- Also linked from my home page `~mccamant`
- These slides will be posted after class

## Canvas

- Assignment submissions (going to try Gradescope this semester)
- Gradebook viewing

## 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

## Challenging course aspects

- Stressing C, low-level, and Unix skills
- Thinking like an attacker
- Thinking like a researcher
- Time management

## 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

## Hands-on Assignment 1

- Weekly attacks 9/24-10/22
- Attack a badly coded extensible text editor (BCEMACS)
- Test your attacks using Linux virtual machines

## Exploiting BCEMACS

- BCEMACS can run as super-user ("root")
- Bugs allow a regular user to gain root privileges (shell)
- Challenge: many steps from bug to working exploit
- Challenge: bugs fixed over time

## Detailed material starts next week

- Readings, projects, exercise set 1
- See you on Monday!