

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

Stephen McCamant  
University of Minnesota, Computer Science & Engineering

## Outline

Big-Picture Introduction

Roll Call

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 mandatory access control
- 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

- Big-Picture Introduction
- Roll Call
- Course Logistics

## Outline

- Big-Picture Introduction
- Roll Call
- Course Logistics

## Instructor information

- Stephen McCamant
- Office: 4-225E Keller
- Office hours: Monday 10-11am, Tuesday 3-4pm, or by appointment
- Email: [mccamant@cs.umn.edu](mailto:mccamant@cs.umn.edu)

## Teaching assistant

- Se Eun Oh
- Email: [seoh@umn.edu](mailto:seoh@umn.edu)
- Office hours Tuesday 1-2pm, Thursday 10-11am, in 2-209 Keller

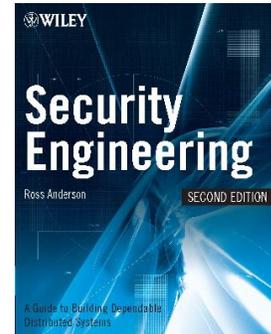
## Prerequisites

- 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



## 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 2 or 3
- ▣ Mostly thinking and writing, not much programming
- ▣ Submit one set per group, plain text or PDF, via Moodle

## Hands-on assignments

- ▣ Two assignments, by large topic divisions
- ▣ Do individually or in groups of 2 or 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 16th in class
- ▣ Final: Monday December 18th 8:00-10:00am
- ▣ Mark your calendars!

## Group research project

- Single most important and time-consuming part of course
- Groups of 3-6, preferably 4 or 5
- Engage with a recent research paper
  - Reproduce and extend, or
  - Reproduce and attack

## Project milestones

- Pre-proposal (due Sept. 20)
- Progress meetings and reports (monthly)
- Short in-class presentation (last two weeks)
- Paper-style final report (due Dec. 13)

## Pre-proposal (Sept. 20)

- 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 11:55pm 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

## Moodle

- Assignment submissions
- Discussion forums
  - Including: group formation

## Challenging course aspects

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

## Hands-on Assignment 1

- ▣ Weekly attacks 9/15-10/13
- ▣ Attack a badly coded text editor (BCVI)
- ▣ Test your attacks using Linux virtual machines (coming soon)

## Exploiting BCVI

- ▣ BCVI 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 Monday

- ▣ Registration reminders
- ▣ Readings, projects, exercise 1
- ▣ See you on Monday!