

CS 489 / 698

Software and Systems Security

Module 1
Introduction to Software and Systems Security

Winter 2024

Instructors

Yousra Aafer

- yousra.aafer@uwaterloo.ca
- <https://cs.uwaterloo.ca/~yaafer/>

Meng Xu

- meng.xu.cs@uwaterloo.ca
- <https://cs.uwaterloo.ca/~m285xu/>

CS 489 / 698

- This is the **second** offering of this course
- The pilot edition was offered last Spring
- This course replaces CS458/658 (Computer Security and Privacy)
- Expands on the Security Aspect
- The course includes (pilot) experimental topics

- We seek your feedback on the course content, delivery and topics covered

CS 489 / 698 (Prerequisites)

- CS 350 (Operating Systems)
- Familiarity with C

Instructor Office Hours

- Weekly Office hours: Tuesdays 3:00 – 4:00 pm virtual
Link: <https://bbb.crysp.org/rooms/men-slj-j7n-0lq/join>
Access code: 1xszia
- In-person office hours by appointment only
- Instructor office hours are meant to answer questions related to module content, course policies, syllabus matters, and special situations

Teaching Assistants

- Liyi Zhang
- Parjanya Vyas

Office hours on **Thursdays 2:30pm to 3:30pm;**

Link: <https://bbb.crysp.org/rooms/i01-wb5-pvq-hc9/join>

Access code: 6gx9wq

Course Mechanics

- Campus and CS VPNs: remote working
- student.cs account: code submission
 - If you don't have a student.cs account for some reason, ask cscfhelp@uwaterloo.ca for help
- LEARN: assignment and grade distribution

Communication Channel

- Important course announcements will be made on Piazza.
 - Please keep up with the information there.
- Use discussion forums in Piazza for all communication
 - Use a **private** question for questions not of general interest
- Use email only as a last resort and then it must be from your uwaterloo.ca email address
- Some communication might be sent to your uWaterloo email address
 - Check your uWaterloo email account regularly or have email forwarded to your regular account

Course Mechanics

- [Piazza](#): Q&A, general discussions
- [Logistics](#), office hours links, assignment due dates, etc
- [Module Discussions](#) – the place to ask questions about that module's content
- [Assignment Discussions](#) – the place to ask questions about assignments
- ...

Course Mechanics

- [Course website](#): syllabus, slides, public materials

Course Syllabus

- <https://cs.uwaterloo.ca/~m285xu/courses/cs489-w24/syllabus/>
- You are expected to be familiar with the contents of the course syllabus
- If you haven't read it, read it after this lecture

Course Website

- <https://cs.uwaterloo.ca/~m285xu/courses/cs489-w24/modules/>
- Contains the lecture slides (and corresponding readings)
- A draft of the lecture slides for each module will be made available before the module begins.
- **The final version of the lecture slides will be made available after the module is completed**

Course Calendar

- Piazza & course website:

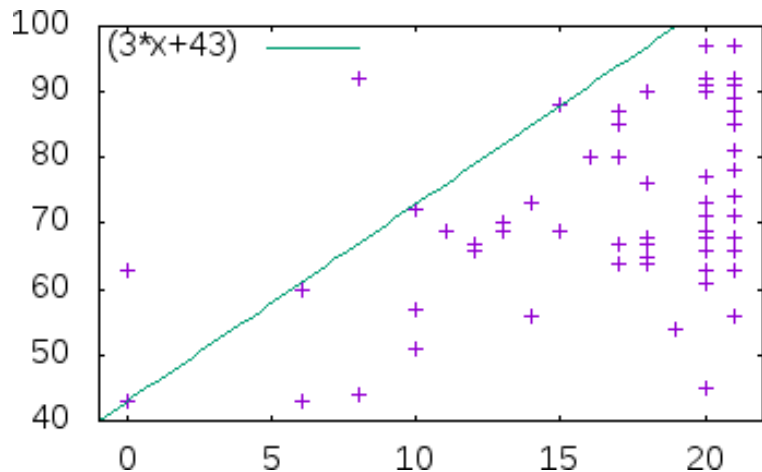
<https://cs.uwaterloo.ca/~m285xu/courses/cs489-w24/schedule/>

- Assignment (and milestones, if any) due dates
- Research project dates (applies to CS698 only)
- **Make sure to check regularly**

Per-student information

- Per-student information will be distributed using LEARN:
 - [Link to be provided in Assignment Handouts](#)
- Assignment marks and comments
- Details regarding Login accounts for assignment machines will be provided in Assignment Handouts

Attend the Lectures!



Grading Scheme

- 4 Assignments (each worth 25%)
 - Might contain either written and programming portions, or both
 - Work alone
- No Final Assessment
- For CS698: an additional research project is required
 - See syllabus for more details
 - Grading scheme = 4 x 20% (Assignments) + 20% (Research Project)
- See syllabus for late and reappraisal policies, academic integrity policy, and other details

Assignments

- Assignments will be due **at the end of the day (i.e., 11:59)** Waterloo time.
 - CS698: Same applies to the research project
 - Assignments should be submitted electronically
 - with the submit command
 - See [CSCF submit page](#) for more details.
- **Important Notes:**
 - Only assignments submitted with the official submission system will be accepted

Late Policy

- No assignments will be accepted after the due date (unless you have a Verification of Illness Form and a doctor's note)
- Recommendation: Submit early and submit often!
- You must notify your instructor **well before the due date (at least 1 week)** of any severe, long-lasting problem preventing you from completing an assignment on time
- **No lates** are accepted for the CS698 research project

Re-appraisal

- You can request a re-appraisal for graded assignments
- You need to provide a clear justification of why you think the assignment should be regraded
- We will allow re-appraisal requests within one week of grade release
- Submit requests on the course's Piazza

Plagiarism and Academic Offenses

- We take academic offenses very seriously
 - Even (especially?) in fourth year
- Nice explanation of plagiarism online
 - <https://uwaterloo.ca/math/academic-matters/academic-integrity>
- Read this and understand it
 - Ignorance is no excuse!
 - Questions should be brought to instructor
- Plagiarism applies to both text and code.
- You are free (even encouraged) to exchange ideas, but **no sharing code or text.**
- We may run submissions through MOSS to detect code similarity

Plagiarism (2)

- Common mistakes
 - Excess collaboration with other students
 - Share ideas, but no design or code!
 - Using solutions from other sources
 - Asking public questions containing (partial) solutions
 - Posting (partial) solutions to websites (e.g., github)
- Possible penalties
 - First offense (for assignments; exams are harsher)
 - 0% for that assignment, -5% on final grade
 - Second offense
 - More severe penalties, including suspension
- Penalties for graduate students are more severe
- More information linked to from course syllabus

A Note on Security

- In this course, you will be exposed to information about security problems and vulnerabilities with Software and Systems.
- To be clear, **you are not to use this or any other similar information to test the security of, break into, compromise, or otherwise attack, any software or system** without the express consent of the owner
- In particular, you will comply with all applicable laws and University policies.
- See syllabus for more details.

Recommended Textbooks

- **Computer Security and the Internet: Tools and Jewels from Malware to Bitcoin (2nd Edition)**, Paul van Oorschot, Springer, 2021.
- **Security in Computing**, 5th edition, Charles P. Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, Prentice-Hall, 2015.
- Digital copies are available via the library website (linked from [Course Syllabus](#))
- You are expected to know all the material presented in class, even if it's not in the textbooks.

Other readings

- From time to time, there will be additional assigned readings
- Links will be provided in the course website

<https://cs.uwaterloo.ca/~m285xu/courses/cs489-w24/modules/>

- There will be both mandatory and optional readings
- You must read the mandatory ones **before** the class in which we will discuss them.
 - There is such a reading for the next lecture

Course Modules

- 1) Introduction to Software and Systems Security
- 2) Program Security
- 3) Operating System Security
- 4) Mobile Security
- 5) Common Bugs and Vulnerabilities
- 6) Bug Finding Tools and Practices
- 7) Defenses against Common Vulnerabilities
- 8) Hardware Security
- 9) Non -Technical Aspects in Security

Module outline

- ① What is our goal in this course?
- ② What is security?
- ③ How is it different from privacy?
- ④ Who are the adversaries?
- ⑤ Assets, vulnerabilities, threats, attacks, and defences
- ⑥ Methods of defence

What is our goal in this course?

- Our primary goal is to be able to **identify security issues** in various aspects of modern computing environments including:
 - Programs
 - Operating systems
 - Mobile systems
- Second, deploy state-of-the-art detection and defence mechanisms
 - To be able to use this ability to **design systems that are more protective of security**.

What is security?

- In the context of computers, **security** generally means three things:
 - **Confidentiality**
 - Access to systems or data is limited to authorized parties
 - **Integrity**
 - When you receive data, you get the “right” data
 - **Availability**
 - The system or data is there when you want it
- A computing system is said to be secure if it has all three properties
 - Well, usually

Security and reliability

- Security has a lot to do with “reliability”
- A secure system is one you can rely on to (for example):
 - ① Keep your personal data confidential
 - ② Allow only authorized access or modifications to resources
 - ③ Ensure that any produced results are correct
 - ④ Give you correct and meaningful results **whenever you want them**

Security vs. privacy

- Sometimes people place security and privacy as if they're opposing forces.
- Are they really? Do we have to give up one to get the other?

How is Privacy different from Security?

- There are many definitions of privacy
- A useful one: “**informational self-determination**”
 - This means that **you** get to **control** information **about you**
 - “**Control**” means many things:
 - Who gets to see it
 - Who gets to use it
 - What they can use it for
 - Who they can give it to
 - etc.

Example: PIPEDA

- PIPEDA (Personal Information Protection and Electronic Documents Act) is Canada's private-sector privacy legislation
- Lists ten Fair Information Principles companies need to abide by:
 - ① Identify the purpose of data collection
 - ② Obtain consent
 - ③ Limit collection
 - ④ Limit use, disclosure and retention
 - ⑤ Use appropriate safeguards
 - ⑥ Give individuals access
 - ⑦ Be accurate
 - ⑧ Be open
 - ⑨ Be accountable
 - ⑩ Provide recourse

(Read more: https://www.priv.gc.ca/leg_c/p_principle_e.asp)

Consumer Privacy Protection Act

- Forthcoming legislation to regulate private sector use of personal information.
- Modernizing protection: meaningful consent, right to erasure, etc.
- Stronger provisions for enforcement.
- Private right of action.

Who are the adversaries?

- Who's trying to mess with us?
- Various groups:
 - Murphy Amateurs
 - “Script kiddies”
 - Crackers or Hackers
 - Organised crime
 - Government “cyberwarriors”
 - Terrorists
 -
- Which of these is the most serious threat today?

Some terminology

- **Assets**
 - Things we might want to protect, such as:
 - Hardware
 - Software
 - Data
- **Vulnerabilities**
 - Weaknesses in a system that may be able to be **exploited** in order to cause loss or harm
 - e.g., a file server that doesn't authenticate its users
 - e.g., an API that allows accessing gps coordinates without authenticating apps

Some terminology

- **Threats**
 - A loss or harm that might befall a system
 - e.g., users' personal files may be revealed to the public

 - There are four major categories of threats:
 - ① Interception
 - ② Interruption
 - ③ Modification
 - ④ Fabrication

 - When designing a system, we need to state the **threat model**
 - Set of threats we are undertaking to defend against
 - **Whom** do we want to prevent from doing **what**?

Some terminology

- **Attack**
 - An action which **exploits** a **vulnerability** to **execute** a **threat**
 - e.g., telling the file server you are a different user in an attempt to read or modify their files
- **Control/Defence**
 - Removing or reducing a vulnerability
 - You **control** a **vulnerability** to prevent an **attack** and defend against a **threat**.
 - How would you control the file server vulnerability?
 - Our goal: control vulnerabilities

Methods of defence

- How can we defend against a threat?
 - **Prevent it:** prevent the attack
 - **Deter it:** make the attack harder or more expensive
 - **Deflect it:** make yourself less attractive to attacker
 - **Detect it:** notice that attack is occurring (or has occurred)
 - **Recover from it:** mitigate the effects of the attack
- Often, we'll want to do many things to defend against the same threat
 - “**Defence in depth**”
- How to defend against the following threat?
your car may get stolen

Example of defence

- Threat: your car may get stolen
- How to defend?
 - Prevent: Immobilizer? Is it possible to absolutely prevent?
 - Deter: Store your car in a secure parking facility
 - Deflect: Have sticker mentioning car alarm, keep valuables out of sight
 - Detect: Car alarms,
 - Recover: Insurance

How secure should we make it?

- Principle of Easiest Penetration
 - “A system is only as strong as its weakest link”
 - The attacker will go after whatever part of the system is easiest for them, not most convenient for you.
 - In order to build secure systems, we need to **learn how to think like an attacker!**
 - How would you get private information from the US Social Security Administration database?
- Principle of Adequate Protection
 - “Security is economics”
 - Don't spend \$100,000 to protect a system that can only cause \$1,000 in damage

Weakest link



Defend like an attacker... too



Captured from [Google Map Street View](#)

Defence of computer systems

- Remember we may want to protect any of our **assets**
 - Hardware, software, data
- Many ways to do this
 - Cryptography
 - Software Controls
 - Hardware Controls
 - Physical Controls
 - Policies and Procedures

Defence of computer systems

- Cryptography
 - Protecting data by making it unreadable to an attacker
 - Authenticating users with digital signatures
 - Authenticating transactions with cryptographic protocols
 - Ensuring the integrity of stored data
 - Aid customers' privacy by having their personal information automatically become unreadable after a certain length of time

Defence of computer systems

- Software controls
 - Passwords and other forms of access control
 - Operating systems separate users' actions from each other
 - Virus scanners watch for some kinds of malware
 - Development controls enforce quality measures on the original source code
 - Personal firewalls that run on your desktop

Defence of computer systems

- Hardware controls
 - Not usually protection of the hardware itself, but rather using separate hardware to protect the system as a whole
 - Fingerprint readers
 - Smart tokens
 - Firewalls, intrusion detection systems
 - Trusted Execution Environments (TEEs)

Defence of computer systems

- Physical controls
 - Protection of the hardware itself, as well as physical access to the console, storage media, etc.
 - Locks
 - Guards
 - Off-site backups
 - Don't put your data centre on a fault line in California
 - Don't put your nuclear power plant in a tsunami zone

Defence of computer systems

- Policies and procedures
 - Non-technical means can be used to protect against some classes of attack
 - If an employee connects their own Wi-Fi access point to the internal company network, that can accidentally open the network to outside attack
 - So don't allow the employee to do that!
 - Rules about choosing passwords
 - Training in best security practices

Recap

- What is our goal in this course?
 - Identify security and privacy issues
 - Design systems that are more protective of security and privacy
- What is security?
 - Confidentiality, Integrity, Availability
- What is privacy?
 - Informational self-determination

Recap

- Who are the adversaries?
 - Learn to think like an attacker
- Assets, vulnerabilities, threats, attacks and controls
 - You **control** a **vulnerability** to prevent an **attack** and block a **threat**
- Methods of defence
 - Cryptography, software controls, hardware controls, physical controls, policies and procedures