NEU CY 5770 Software Vulnerabilities and Security

Instructor: Dr. Ziming Zhao

First off, Logistics!

Classes are recorded and released publicly on YouTube
But you have to attend the class in-person

Have a notebook in front of you
Bring your own laptop

https://zzm7000.github.io/teaching/2025springcy5770/index.html

We have an online CTF platform for this class.

Feel free to interrupt me and ask questions.

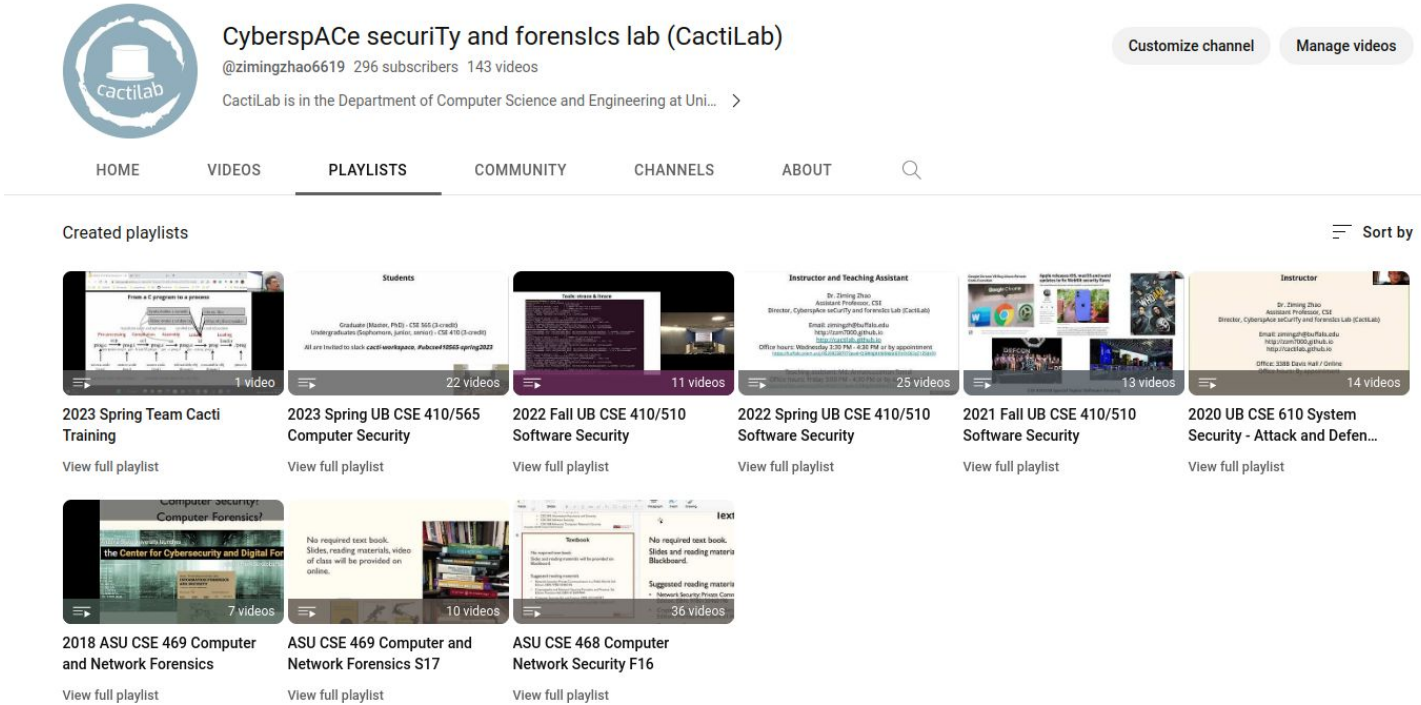
Instructor and Teaching Assistant

Dr. Ziming Zhao
Associate Professor, Khoury College of Computer Sciences
Director, CyberspAce seCuriTy and forensIcs Lab (CactiLab)

Email: z.zhao@northeastern.edu
http://zzm7000.github.io
http://cactilab.github.io
W 6:00 pm - 9:20 pm or by appointment
Office hours will be W 4:30 pm - 5:30 pm or by appointment

https://northeastern.zoom.us/j/99475115019?pwd=4dMw5mmuLNHHh0LS9CCll93xapoB4o.1

YouTube Channel



https://www.youtube.com/channel/UCkSeVUu-AxytXqalx66j7Eg/playlist

About CactiLab

Research areas:

- Systems and software security (Arm Cortex-M, Cortex-A, RISC-V, FPGA, GPU, etc.)
- Security in/with ML/DL/LLM
- Autonomous driving security
- Formally verify the security properties of crypto protocols and system code
- Hacking/CTF platforms

We need students at all levels for funded research, volunteer work, independent study, undergraduate research experience, etc.

Students

Graduate (Master, PhD) - CY 5770 (4-credit)
Undergraduates (junior, senior) - CY 5770 (4-credit)

All are invited to slack ***cacti-workspace***, ***#neucy5770-spring2025***

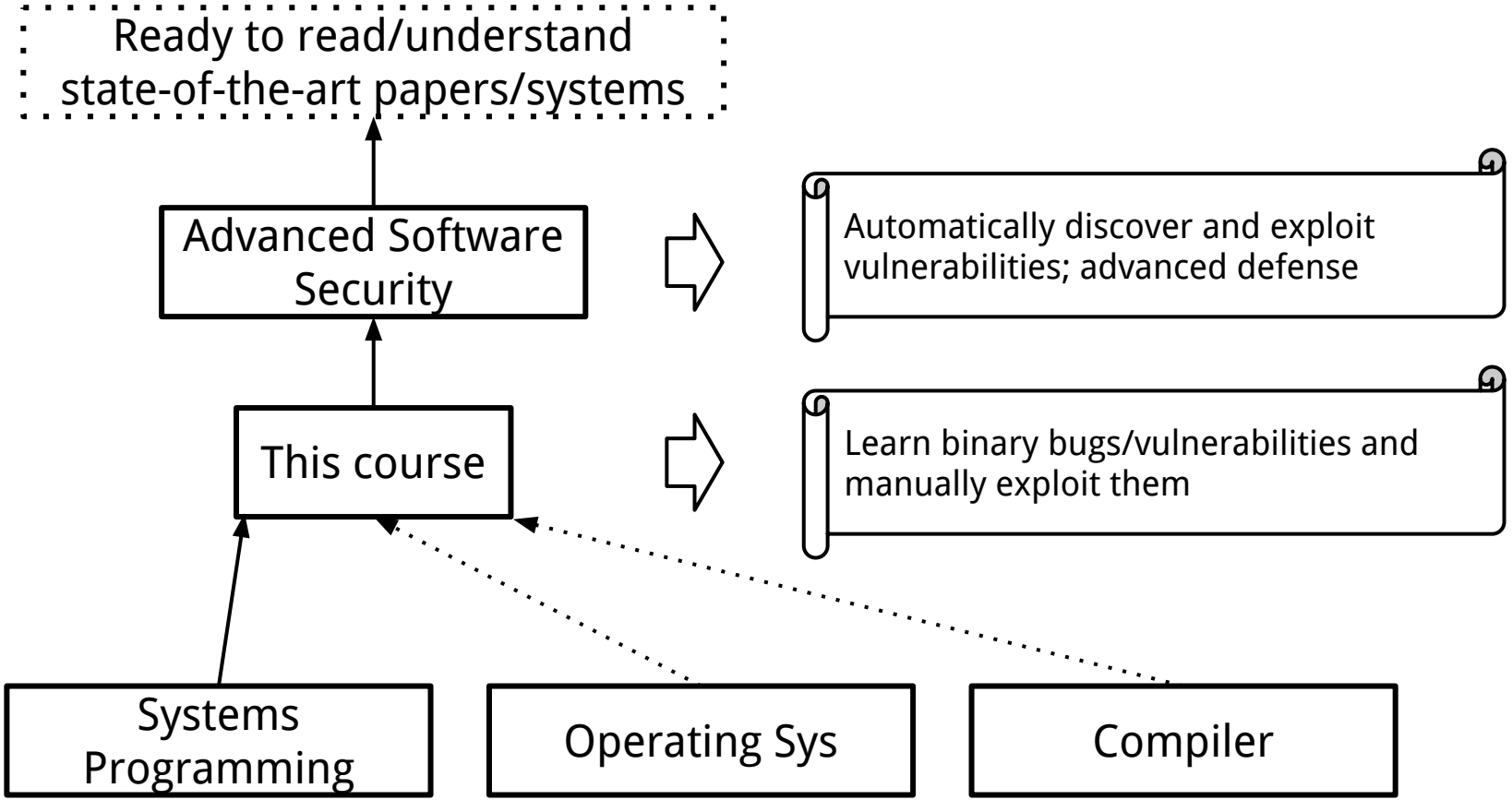
Course Goals

To provide you with good understanding of the **theories**, **principles**, **techniques** and **tools** used for binary software and system hacking and defense.

By software and system, I mean native software, binary, most likely developed in C/C++. The security of web software, Java, Python are out of the scope.

You will study, in-depth, binary reverse engineering, vulnerability classes, vulnerability analysis, exploit/shellcode development, defensive solutions, etc., to understand how to crack and protect **native** software. You will get your hands dirty.

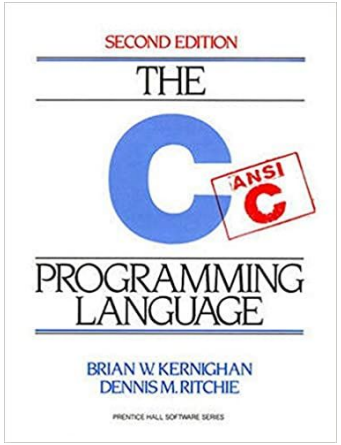
If you want to be a systems/software security guy …



First week's Agenda

1. Class overview and logistics
2. Background knowledge
    a. Compiler, linker, loader
    b. x86 and x86-64 architectures and ISA
    c. Linux fundamentals
        i. Linux file permissions
        ii. Set-UID programs
        iii. Memory map of a Linux process
        iv. System calls
        v. Environment and Shell variables
    d. Basic reverse engineering

Prerequisites



The real prerequisite:
The C Programming Language

Classes that will help you understand this class:
Systems Programming
Operating Systems

Other skills:
Reverse engineering (Using objdump, IDA Pro, Ghidra, etc.)
Debugging (GDB, pwngdb)
Google, reading, self-learning, getting hands dirty

Topics

Binary attack and defense using x86 and x86-64 as examples. Discover **vulnerabilities**. Develop **exploits**. Memory corruption attacks.

1. Stack-based buffer overflow
2. Defenses against stack-based buffer overflow
3. Shellcode development
4. Format string vulnerabilities
5. Heap-based buffer overflow
6. Integer overflow
7. Return-oriented programming
8. …

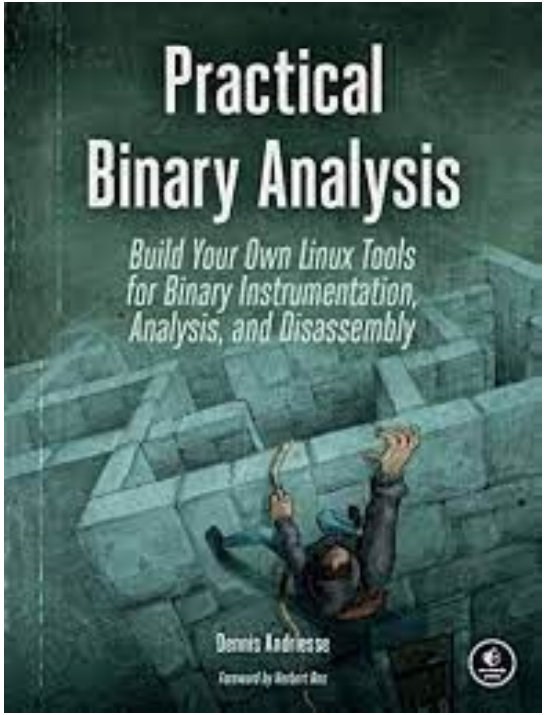
Related Books and Papers

SoK: Eternal War in Memory. IEEE S&P 2013

SoK: (State of) The Art of War: Offensive Techniques in Binary Analysis. IEEE S&P 2016

SoK: Shining Light on Shadow Stacks. IEEE S&P 2019

Practical Binary Analysis: Build Your Own Linux Tools for Binary Instrumentation, Analysis, and Disassembly



Related Books and Papers

SoK: Eternal War in Memory. IEEE S&P 2013

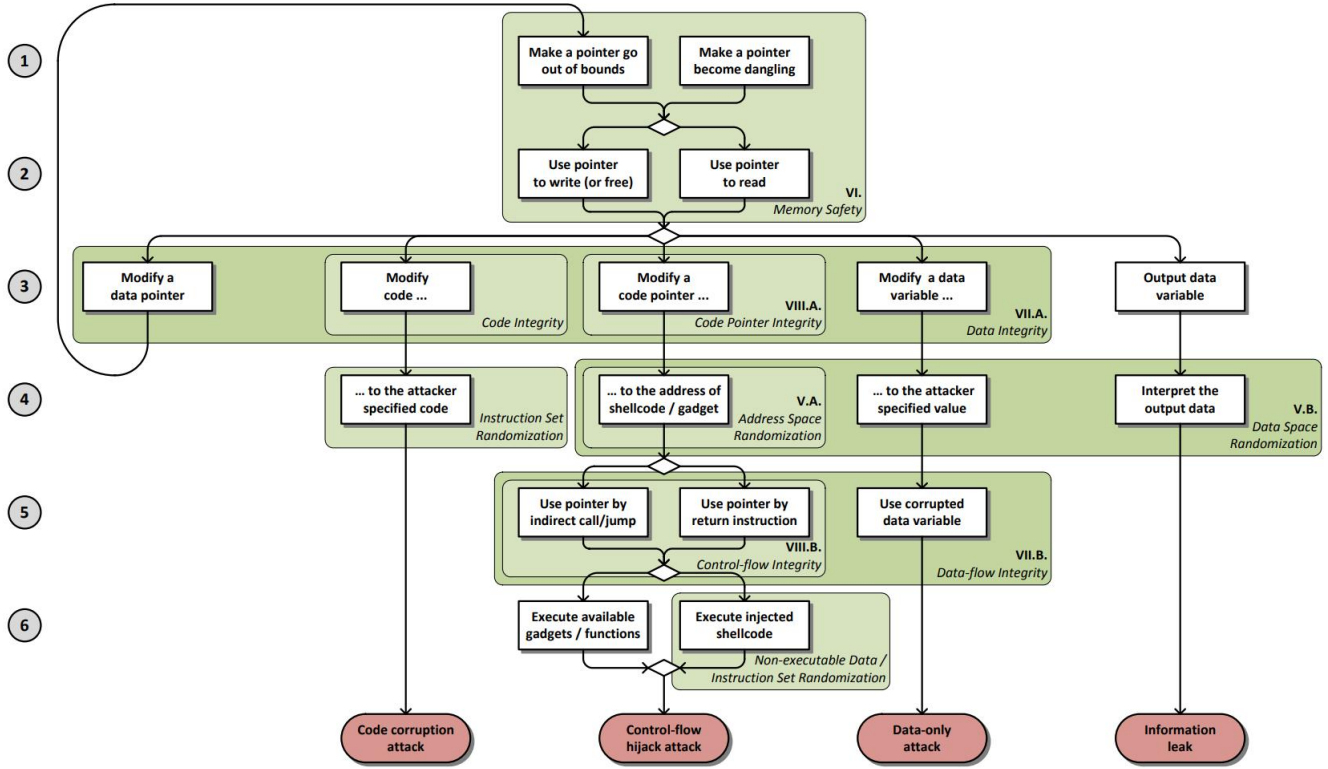


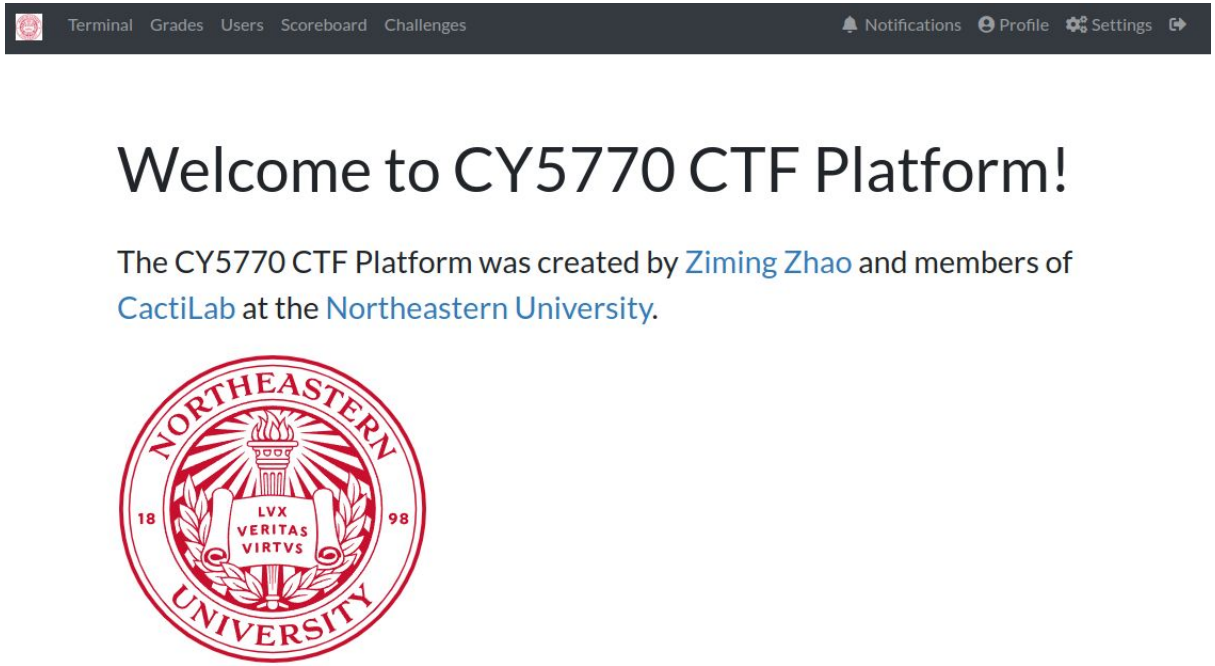
Figure 1. Attack model demonstrating four exploit types and policies mitigating the attacks in different stages

The Hacking Environment

http://cy5770-cacti.khoury.northeastern.edu/

Only NEU students can access this website. If you are off-campus, you need to VPN to connect to NEU network to access

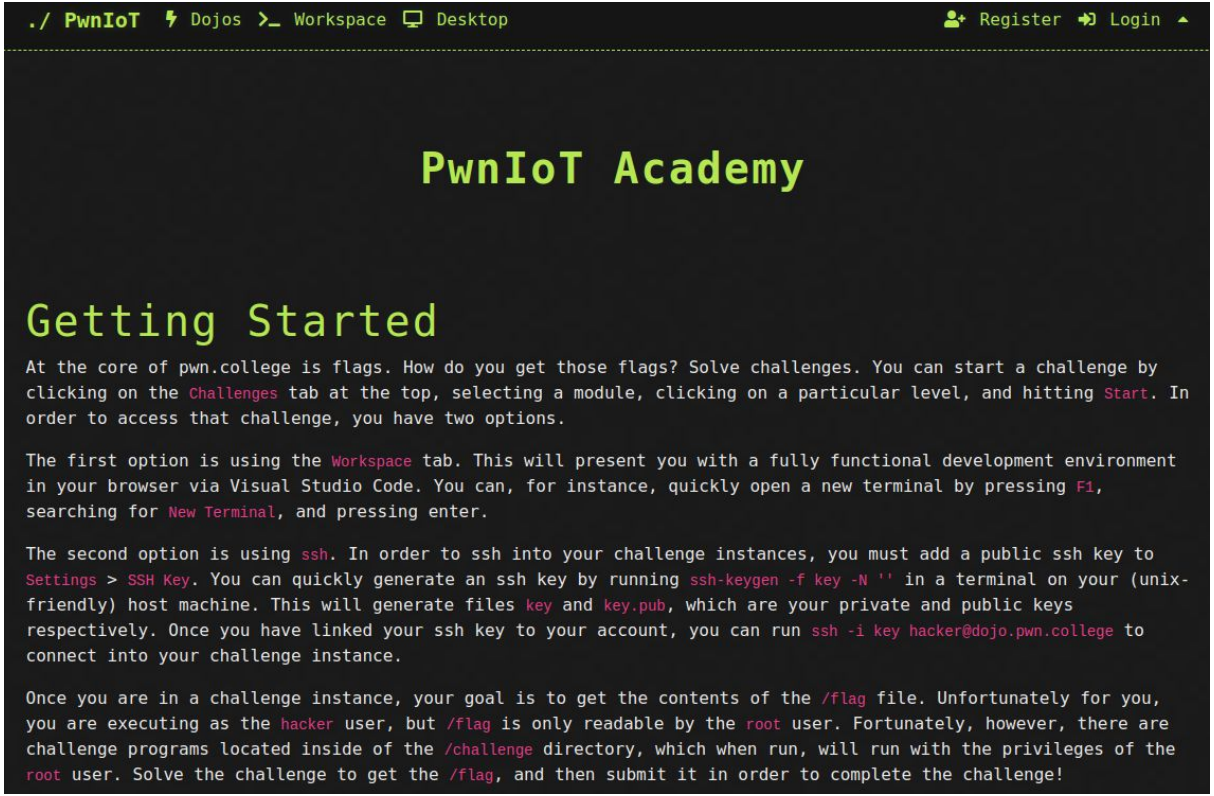
Register an account with your NEU username and email address, so we know who you are.



New Environment Under Construction

Only NEU students can access this website. If you are off-campus, you need to VPN to connect to NEU network to access

Register an account with your NEU username and email address.



The Hacking Environment

Intel x86
x86-64, a.k.a amd64
ARM Cortex-A, Cortex-M
Linux (Ubuntu)

Pwngdb
Pwntools
GDB peda
NSA Ghidra
Binary Ninja

Homework

Reading: book chapter, whitepaper, paper, blog, etc.
Hands-on: hacking, debugging, etc.

Submit before a class on Canvas. We may discuss homework at the beginning of each class.

30% penalty if you submit within 10 mins after class starts. 0 points after 10 mins.

0 points for homework if plagiarising is found. No exceptions.

Disability Access Services

If you need DAS, please inform me in the first two weeks.

Hacking Assignment Rules

- For each hacking assignment, you will submit your exploit, a simple write-up, and screenshots to show it works
  - Simple write-up:
    - Briefly describe how you solve the challenge
    - Mention who you worked with if any in the write-up

- Discussion is encouraged. But, you cannot share your code, exploits, write-ups to your classmates or post them online.

Exams, a.k.a, Capture-the-Flag (CTF) Hacking

Midterm CTF: 3 hours and 20 minutes
Final CTF: 3 hours and 20 minutes

Grades

Students will be evaluated on their performance on the homework and CTFs. Attendance check will be performed in each class. Table 1 shows the grade breakdown.

Area	No. Items	Points per Item	Points for Area
Homework	14	45	630
Exams (CTFs)	2		360
Midterm Exam (CTF)	1	160	
Final Exam (CTF)	1	200	
Attendance	10	1	10
Anonymous Course Evaluation Bonus	2	12	24
Total			1024

5770 (Undergraduate)		5770 (Graduate)	
Points	Grade	Points	Grade
874 -	A	924 -	A
850 - 874	A-	900 - 924	A-
820 - 850	B+	870 - 900	B+
780 - 820	B	830 - 870	B
750 - 780	B-	800 - 830	B-
720 - 750	C+	770 - 800	C+
650 - 720	C	700 - 770	C
550 - 650	D	600 - 700	D
0 - 550	F	0 - 600	F

Academic Integrity

Your first assignment is to to read the NEU academic integrity policies

Here are examples for your consideration

- you work on your laptop at a library with friends and step away from your computer without locking it
- you look at your neighbors' screen/papers during an exam, but don't copy their answers
- you take a piece of code from some website and give a link to the website at the end of the homework
- you work on a homework problem with friends, type the solution at home, but it's exactly the same as that of your friends

Academic Integrity

- Discussion is encourage. But, you cannot share your code, exploits to your classmates or post them online.

- The university, college, and department policies against academic dishonesty will be strictly enforced. To understand your responsibilities as a student read: UB Student Code of Conduct.

- Plagiarism or any form of cheating in homework, assignments, labs, or exams is subject to serious academic penalty.

- Any violation of the academic integrity policy will result in a 0 on the homework, lab or assignment, and even an **F** or **>F<** on the final grade. And, the violation will be reported to the Dean's office.

ChatGPT/LLM Policy

- ChatGPT/LLM is forbidden in the midterm and final CTFs

Ethical Hacking

- Do not attempt to violate the law.
- If you discover real-world vulnerabilities using the knowledge you learn from this class, report the vulnerabilities responsibly. Companies may reward you for that.