

# **CSE 410/565: Computer Security**

Instructor: Dr. Ziming Zhao

# Academic Integrity

Your first assignment is to read the CSE and UB 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' 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 the same as that of your friends

# Academic Integrity

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

# Logistics

Classes are recorded and released publicly on YouTube  
But you have to attend the class in-person  
There will be one attendance check each week

***<https://zzm7000.github.io/teaching/2023springcse410565/index.html>***

Feel free to interrupt me and ask questions.

# Instructor and Teaching Assistant


Dr. Ziming Zhao  
Assistant Professor, CSE  
Director, CyberspAce seCuriTy and forensIcs Lab (CactiLab)

Email: [zimingzh@buffalo.edu](mailto:zimingzh@buffalo.edu)  
<http://zzm7000.github.io>  
<http://cactilab.github.io>

Office hours: Monday 2:30 PM - 3:30 PM or by appointment  
338B or <https://buffalo.zoom.us/j/95299258797?pwd=QlBhbjlIUlM5WmlETmFtOE5qT1Z5dz09>


Teaching assistant: Md. Armanuzzaman Tomal  
Office hours: Wednesday 2:30 PM - 2:30 PM or by appointment  
<https://buffalo.zoom.us/j/95299258797?pwd=QlBhbjlIUlM5WmlETmFtOE5qT1Z5dz09>

# YouTube Channel




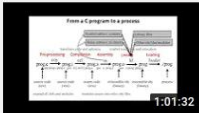
**Ziming Zhao**  
156 subscribers

[CUSTOMIZE CHANNEL](#)[MANAGE VIDEOS](#)

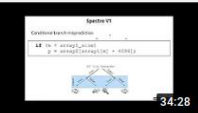
[HOME](#)[VIDEOS](#)[PLAYLISTS](#)[CHANNELS](#)[ABOUT](#)

Uploads


 SORT BY




**W15L1 Beyond 410**  
25 views • 3 months ago



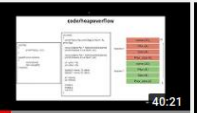
**W14L2 Spectre**  
39 views • 3 months ago




**W14L1 Meltdown**  
37 views • 3 months ago




**W13L2 Cache side-channel**  
72 views • 3 months ago




**W13L1 Heap Exploitation 2**  
105 views • 3 months ago




**W12L2 How Heap works**  
124 views • 4 months ago



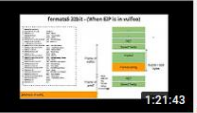
**W12L1 Return-oriented Programming 3**  
122 views • 4 months ago




**W11L2 Return-oriented Programming 2**  
151 views • 4 months ago




**W11L1 Return-oriented Programming 1**  
172 views • 4 months ago




**W10L1 Format String Vulnerability 2**  
156 views • 4 months ago



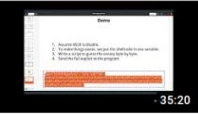
**W10L1 Format String Vulnerability 1**  
176 views • 4 months ago




**W9L2 Shellcode Development**  
132 views • 4 months ago




**W9L1 ASLR and Seccomp**  
102 views • 4 months ago




**W6L2 Bypass Canary**  
196 views • 5 months ago




**W6L1 Shadow Stack and Canary**  
77 views • 5 months ago




**W5L2 Buffer overflow**  
191 views • 5 months ago




**W5L1 Buffer Overflow (Frame Pointer)**  
229 views • 5 months ago




**W4L2 Buffer Overflow**  
278 views • 6 months ago




**W3L2 Buffer Overflow**  
278 views • 6 months ago




**W2L2 Buffer Overflow**  
278 views • 6 months ago



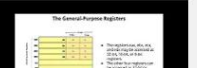
**W1L2 Buffer Overflow**  
278 views • 6 months ago



**W0L2 Buffer Overflow**  
278 views • 6 months ago



**W0L1 Buffer Overflow**  
278 views • 6 months ago



**W0L0 Buffer Overflow**  
278 views • 6 months ago

<https://www.youtube.com/channel/UCkSeVUu-AxytXqalx66j7Eg/videos>

# About CactiLab

## Research areas:

- Embedded system and software security (Arm Cortex-M, Cortex-A, RISC-V, FPGA, etc.)
- Security in/with machine learning/deep learning
- Autonomous driving security
- Formally verify the security properties of crypto protocols and system code
- Blockchain security
- IoT hacking/CTF platforms (Roblox for hacking)

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

# Students

Graduate (Master, PhD) - CSE 565 (3-credit)

Undergraduates (Sophomore, junior, senior) - CSE 410 (3-credit)

All are invited to slack ***cacti-workspace, #ubcse410565-spring2023***



# Course Goals

The objectives of this course consist of developing a solid understanding of **fundamental principles** of the **security** field and building knowledge of tools and mechanisms to safeguard a wide range of software and computing systems.

Topics:

- Cryptographic background and tools;
- Access control; authentication;
- Software security, malware;
- Internet security protocols and standards (SSL/TLS, IPsec, secure email);
- Intrusion detection and intrusion prevention systems (firewalls);
- Database security;
- Privacy; identity management;
- Security management and risk assessment;
- Legal and ethical aspects (cybercrime, intellectual property)

# What is Computer Security as a field?

Computer security is very broad as a field

It covers many areas:

- Network security
- Software security
- System security
- Web security
- Safety in programming language
- Database security
- Usable security
- Access control
- Privacy
- Cybercrime
- ...

# If you want to be a security researcher ...

Ready to read/understand state-of-the-art  
software security papers/systems

Advanced Software  
Security

Web Security

Network Security

CSE 410/510  
Software Security

CSE 365 Intro to  
Computer Security

This course

CSE 220 Systems  
Programming

Operating  
Sys

Compiler

# CSE 410/510 Software Security - 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. ...

# CSE 410/510 Software Security - The Hacking Environment

<http://cse410.cacti.academy/>

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

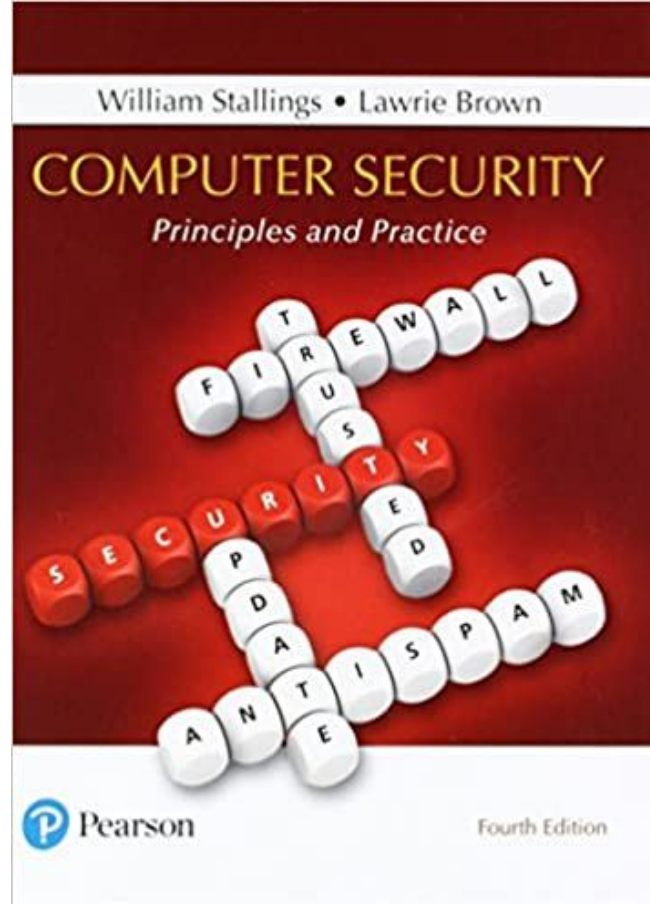
Register an account with your UB username and email address.

## Welcome to CSE410/510 CTF Platform!

CSE410/510 CTF Platform for course practice problems on 0x86 architecture. The platform was created by [Ziming Zhao](#) and members of [CactiLab](#) at the [University at Buffalo](#).



# Textbooks



# Homework

Reading: book chapters

Hands-on: programming

- All assignments must be done **individually** unless announced otherwise; no collaboration on solving or writing assignments
- Searching for homework answers online is not permitted
- The use of any external resources should be properly documented when answering the question (i.e., not at the end of the homework) and the nature of the help the document provided (i.e., what information it contained and how it helped with answering the question)
- Homeworks will be submitted via UBlearns; they must be typed (diagrams can be hand-drawn) and normally would need to be submitted as a PDF.
- Late submissions in 48 hours will be accepted with a 15% (24 hours late) and 30% (48 hours late) penalty, respectively. Late submissions beyond 48 hours will not be accepted.
- 0 points for homework if plagiarising is found. No exceptions.

# Regrading requests

- Homework or exam regrade requests need to be submitted within two weeks of releasing the graded material to the class
- The request needs to be in writing clearly describing the error in grading



# **Disability Accommodations**

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

# Exams

Midterm: 1 hour 20 minutes

Final exam: 3 hours

# Grades

Area	No. Items	Points per Item	Points for Area
Homework	6	60	360
Exams	2		630
Midterm	1	250	
Final	1	380	
Attendance	14	1	14
Anonymous Course Evaluation Bonus	2	10	20
Total			1024

Points	Grade
930 -	A
900 - 930	A-
870 - 900	B+
830 - 870	B
800 - 830	B-
770 - 800	C+
700 - 770	C
670 - 700	D+
600 - 670	D
0 - 600	F

# **Basic Computer Security Concepts**



Jonathan Greig  
January 6, 2023

Cybercrime Malware News



## SickKids: 80% of hospital priority systems back online after LockBit ransomware attack

Toronto's Hospital for Sick Children, Canada's largest pediatric health center, said it has restored 80% of its systems that have a direct impact on hospital operations following a ransomware attack.

A spokesperson for the hospital told The Record that patients and families dealt with diagnostic and treatment delays because clinical teams struggled to receive lab reports and imaging results after the LockBit ransomware group launched an attack on the night of December 18.

While the hospital's electronic medical record system was not affected by the attack, several other tools connected to the system were damaged – including dictation services, pharmacy systems and the ability to view diagnostic imaging results. The hospital's internal timekeeping system for staff and intranet were also affected by the ransomware attack.

## JD Sports says 10 million customers hit by cyber-attack

6 hours ago



By Michael Race  
Business reporter, BBC News

Sportswear chain JD Sports has said stored data relating to 10 million customers might be at risk after it was hit by a cyber-attack.

The company said information that "may have been accessed" by hackers included names, addresses, email accounts, phone numbers, order details and the final four digits of bank cards.

# What is Computer Security?

The NIST Internal/Interagency Report NISTIR 7298:

Computer Security: Measures and controls that ensure **confidentiality**, **integrity**, and **availability** of information system **assets** including hardware, software, firmware, and information being processed, stored, and communicated.

# Security Objectives

Confidentiality

Integrity

Availability

Authenticity

Non-repudiation

# Security Objectives (CIA Triad)

**Confidentiality:** Prevent/detect/deter improper *disclosure* of information.

- data confidentiality: sensitive information is available to authorized parties only
- privacy: individuals can control what information about them can be collected and stored and to whom it is made available
- ...



# Security Objectives (CIA Triad)

**Integrity:** Prevent/detect/deter improper *modification* of information

- data integrity: information and software can be modified only in a predetermined and authorized manner
- system integrity: a system performs its intended functions in an expected manner and has not be manipulated in an unauthorized way
- ...

# Security Objectives (CIA Triad)

**Availability:** Prevent/detect/deter improper *denial of access to services* provided by the system

# Examples

- **Confidentiality:** You should not come to know the scores of your classmates in this class
- **Integrity:** You should not be able to change your or others' scores in this class
- **Availability:** Your scores should always be available on UBLearns

## In Addition to CIA Triad

- **Authenticity:** The assurance that a message, transaction, or other exchange of information is from the **source** it claims to be from.
- **Non-repudiation:** The assurance that someone cannot deny something, such as the receipt of a message or the authenticity of a statement or contract.

# Examples

- **Authenticity:** You should not pretend as the TA to send an email to your classmates
- **Non-repudiation:** The TA can not pretend he did not send out the message

# How to we achieve the objectives? What are the possible measures and controls?

The means of achieving these objectives greatly differ

- cryptographic techniques
- access control policies
- software checking tools
- virus scanners
- firewalls
- spam filters, etc.

Each system must be evaluated uniquely in terms of its requirements

- security mechanisms must be adequately chosen in accordance with those requirements

# Why is security hard?

- Identifying security requirements of a system is non-trivial
  - must take into account services, environment, etc.
- Finding adequate (often complex) solutions is not easier
  - the decision must take into account known and unknown attacks and threats
  - security mechanisms must be logically placed
- Securing a system is not a one-time task
  - the system must be constantly monitored in face of changing threats
  - security mechanisms need to be re-evaluated

# Why is security hard?

- Managers do not perceive value in security investment (until a security failure occurs)
  - system administrators might not influence decisions or not make good decisions
- Users view security measures as an obstacle on the way of getting their work done
  - we would like security mechanisms to be as intuitive and robust as possible
- Adding security to an existing system might not be pretty
  - ideally, security is an integral part of the design



# Takeaways

- Security is not absolute
  - assets can have different security grades depending on the impact of a security breach that can range from low to high
  - by building more secure systems, we make it harder for an attacker to breach security
  - the more resources we can invest in a system, the more secure we can make it
  - there is a trade-off between security and resources (money, equipment, personnel, training)
  - training must cover all users, as security can often be easiest breached by exploiting human error

# More Concepts

- Security policy – a set of rules or practices that specify how a system or organization is prescribed to protect its assets
- Vulnerability – a flaw or weakness in system's design, implementation, or operation that could be exploited to violate the security policy
- Threat – a potential for violation of security, i.e., a possible danger that might exploit a vulnerability

# More Concepts

- Attack – a deliberate and intelligent attempt to violate the security policy of a system or get around security services
- Adversary or attacker – an entity that attacks a system or is a threat to it
- Countermeasure – an action, procedure, or technique that reduces a threat or vulnerability, prevents or mitigates an attack
- Risk – an expectation that a particular threat will exploit a particular vulnerability with a particular harmful result

# Type of Adversaries

- passive: observes information without intervention • e.g., passively monitoring a communication link
- active: alters system resources or affects their operation • e.g., changing messages, replaying old messages on the network, corrupting users, etc.
- insider: is legitimately a part of the system with access to internal data or is inside the security perimeter
- outsider: is outside of the security perimeter or is not a legitimate user

# Security Design Principles

- economy of mechanism
- open design, modularity
- layering
- complete mediation
- fail-safe defaults
- separation of privilege, least privilege
- least common mechanism
- psychological acceptability, least astonishment
- isolation, encapsulation

Academic integrity quiz is assigned. The academic integrity quiz must be answered correctly within the two weeks of classes to pass the course.