System Security - Attack and Defense for Binaries

CS 4390/5390, Spring 2026

Instructor: MD Armanuzzaman (***Arman***)

Getting to know ourselves

- Name
- Undergraduate
    - What excites you in CS/Cybersecurity/research direction?
- Graduate
    - What is your research direction?
    - Advisor?
- What do you expect to learn from this course?
    - High-level is fine
    - If you just found the term hacking exciting, it's also fine

Logistics

- Class attendance
- Keep a notebook
- Always bring your laptop
- We will hack in a class CTF platform
- During the lectures feel free to interrupt and ask questions
- Class website : slides will be posted here
- Homework submissions on blackboard

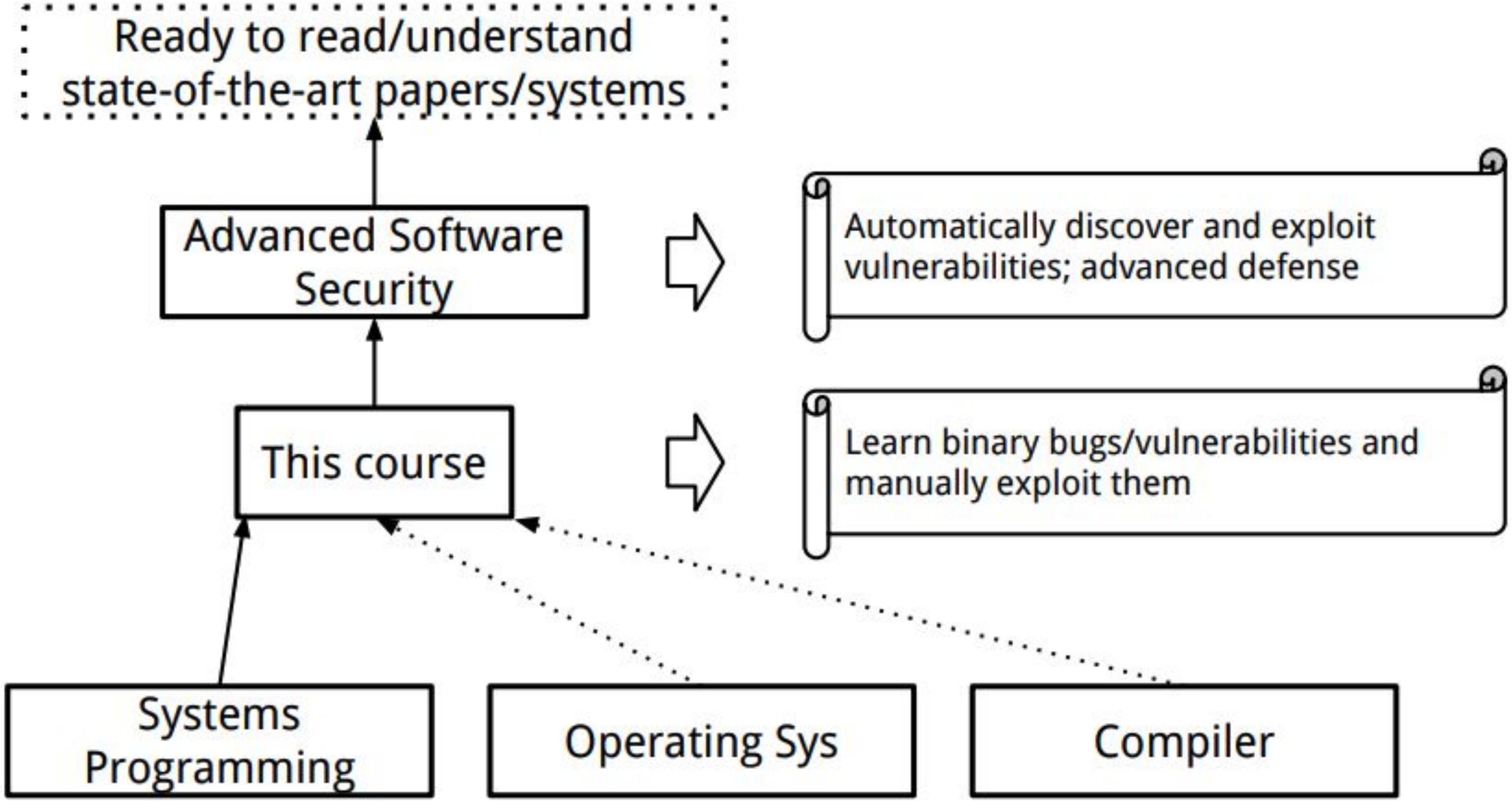
Instructor

- Dr. MD Armanuzzaman, Assistant Professor at UTEP
  - Just call me **Arman**
- Email: marmanuzzaman@utep.edu
- Website: https://tomal-kuet.github.io/armanuzzaman/
- Office hours on TR: 1.30PM - 3.00PM
  - In person at CCSB: 3.1008 or Teams (course website)
- Research area: Systems and Software Security
  - Embedded and IoT systems, Cloud systems, FPGAs, Fuzzing, etc.
- Looking for team members:
  - I am currently looking BS/MS/PhD students to join my research team.

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 …



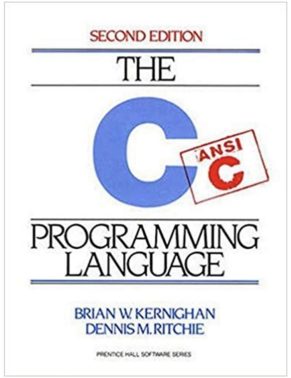
Ready to read/understand state-of-the-art papers/systems
Advanced Software Security
Automatically discover and exploit vulnerabilities; advanced defense
This course
Learn binary bugs/vulnerabilities and manually exploit them
Systems Programming
Operating Sys
Compiler

First week’s Agenda

- Class overview and logistics
- Background knowledge
    - Compiler, linker, loader
    - x86 and x86-64 architectures and ISA
    - Linux fundamentals
        - Linux file permissions
        - Set-UID programs
        - Memory map of a Linux process
        - System calls
        - Environment and Shell variables
    - 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.

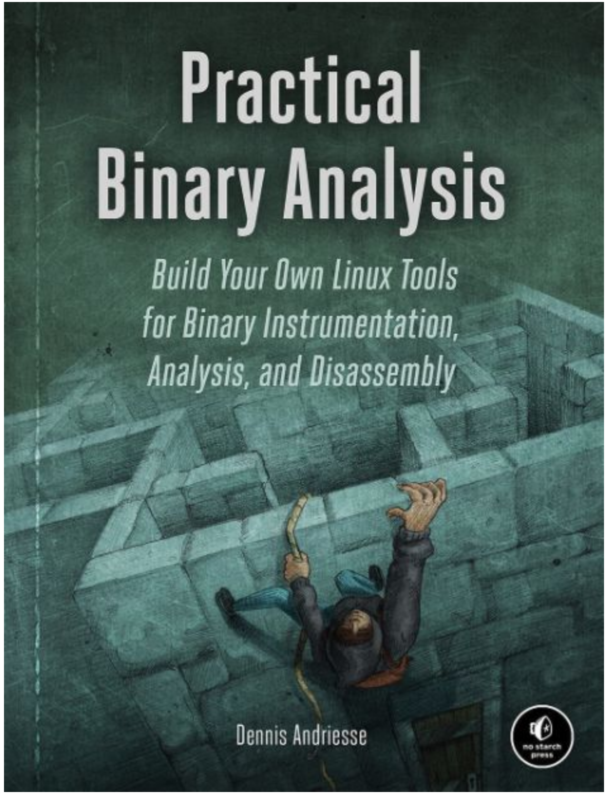
Discover **vulnerabilities**. Develop **exploits**. Memory corruption attacks.

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

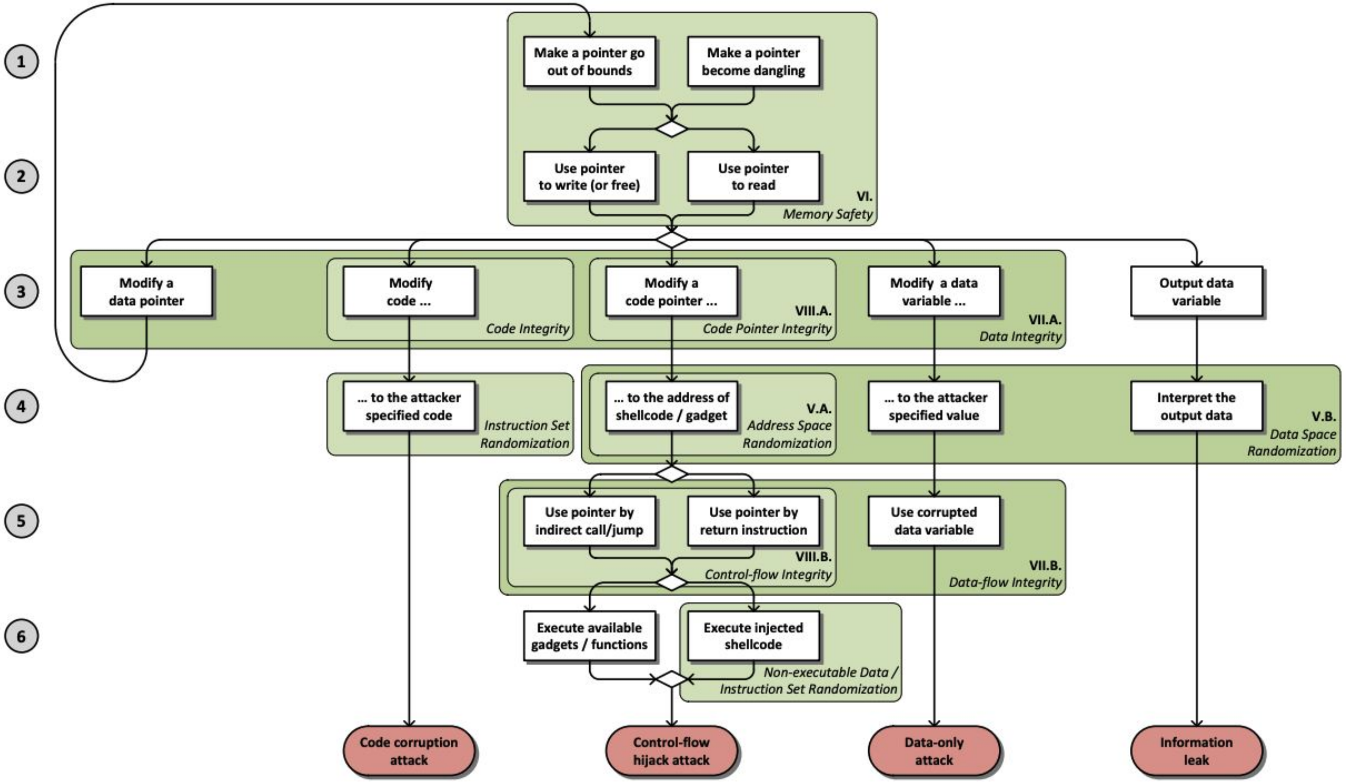
- SoK: Eternal War in Memory. IEEE S&P 2013
- SoK: (State of) The Art of War: Offensive Techniques in Binary Analysis. IEEE S&P 2016
- SoK: Shining Light on Shadow Stacks. IEEE S&P 2019

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


Practical
Binary Analysis
Build Your Own Linux Tools
for Binary Instrumentation,
Analysis, and Disassembly
Dennis Andriesse
no starch press

Related Books and Papers

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

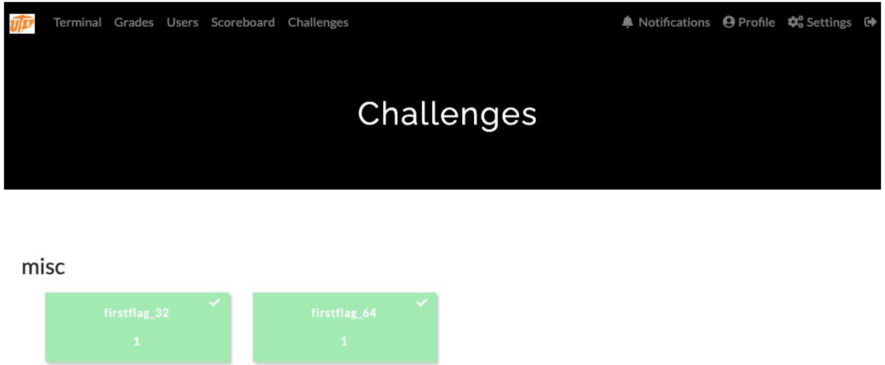

1
2
3
4
5
6
Make a pointer go out of bounds
Make a pointer become dangling
Use pointer to write (or free)
Use pointer to read
VI.
Memory Safety
Modify a data pointer
Modify code ...
Code Integrity
Modify a code pointer ...
VIII.A.
Code Pointer Integrity
Modify a data variable ...
VII.A.
Data Integrity
Output data variable
... to the attacker specified code
Instruction Set Randomization
... to the address of shellcode / gadget
V.A.
Address Space Randomization
... to the attacker specified value
Interpret the output data
V.B.
Data Space Randomization
Use pointer by indirect call/jump
Use pointer by return instruction
VIII.B.
Control-flow Integrity
Use corrupted data variable
VII.B.
Data-flow Integrity
Execute available gadgets / functions
Execute injected shellcode
Non-executable Data / Instruction Set Randomization
Code corruption attack
Control-flow hijack attack
Data-only attack
Information leak

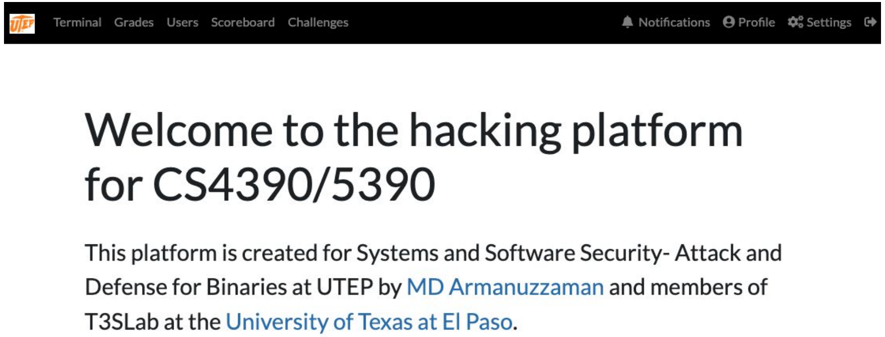
The Hacking Environment

http://yeast.utep.edu:2223/

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

Register and account with your **UTEP username and email address**, so I know who you are





Welcome to the hacking platform for CS4390/5390

This platform is created for Systems and Software Security- Attack and Defense for Binaries at UTEP by MD Armanuzzaman and members of T3SLab at the University of Texas at El Paso.

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 blackboard. 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.

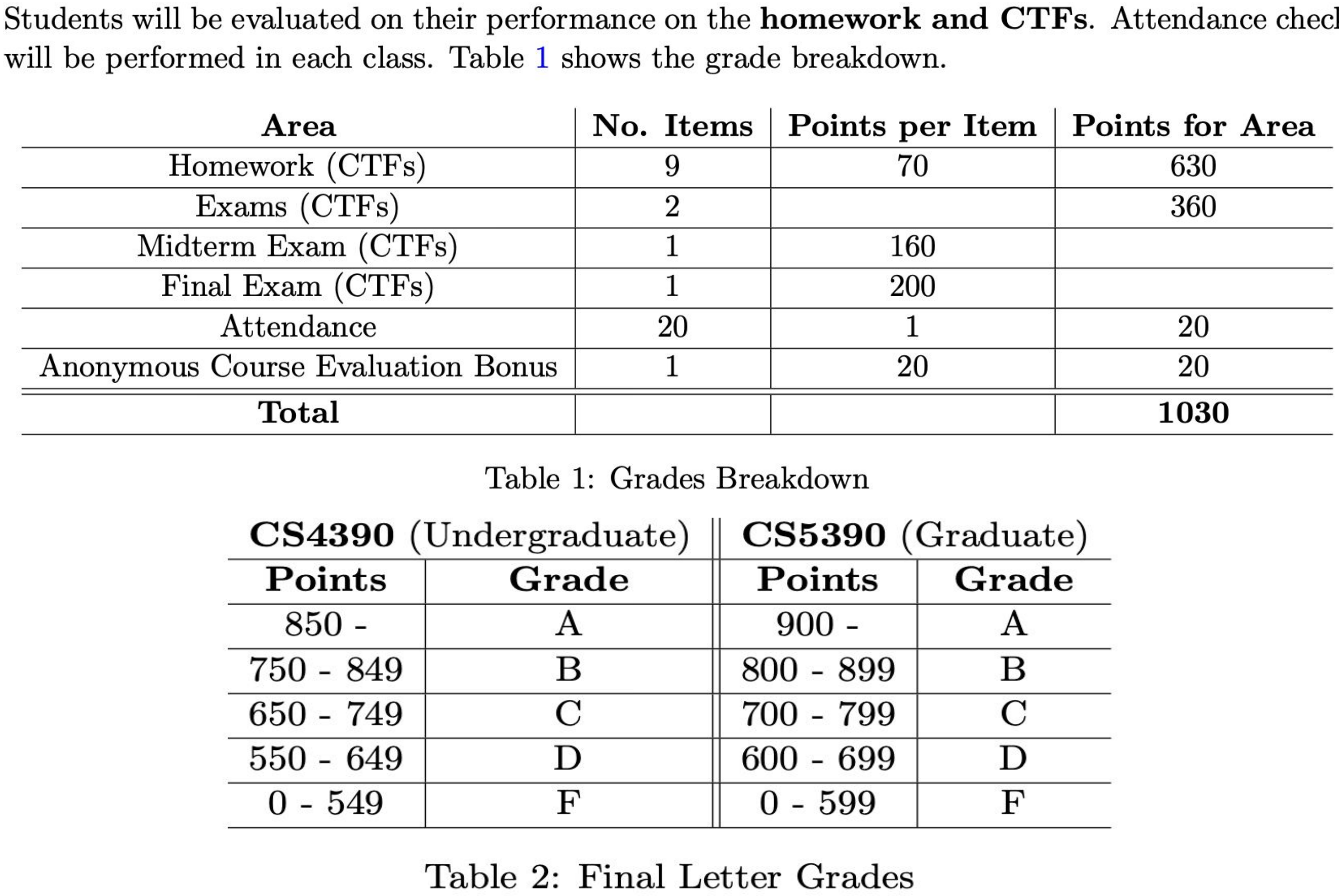
Hacking Assignment Rules

- 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 checl will be performed in each class. Table 1 shows the grade breakdown.

Area	**No. Items**	**Points per Item**	**Points for Area**
Homework (CTFs)	9	70	630
Exams (CTFs)	2		360
Midterm Exam (CTFs)	1	160	
Final Exam (CTFs)	1	200	
Attendance	20	1	20
Anonymous Course Evaluation Bonus	1	20	20
Total			**1030**

Table 1: Grades Breakdown

CS4390 (Undergraduate)		**CS5390** (Graduate)	
Points	**Grade**	**Points**	**Grade**
850 -	A	900 -	A
750 - 849	B	800 - 899	B
650 - 749	C	700 - 799	C
550 - 649	D	600 - 699	D
0 - 549	F	0 - 599	F

Table 2: Final Letter Grades

Academic Integrity

Your first assignment is to to read the UTEP 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: UTEP 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.

Thank you