CS 395: Binary Exploitation in Linux
Spring 2021

Contact Information

Facilitators:
Nihaal Prasad
Samuel Goodwin

Faculty Advisor:
Elizabeth White (white@cs.gmu.edu)

Course Description

CS 395: Student Initiated Special Topics are 1-credit courses that cover special and emerging topics of interest to computer science undergraduates. Lectures are guided by student facilitators under faculty advisement.

This class will introduce students to the basic concepts for developing exploits for vulnerable Linux programs. Students will learn how to identify vulnerabilities in software and produce more secure code. The class will include readings, assignments, and a project. Discussion topics include:

- Ethical hacking
- Stack/buffer overflows
- Integer overflows/underflows
- String format vulnerabilities
- Exploit scripting
- Linux shellcoding
- Exploit mitigation technologies (PIE, ASLR, NX, Stack Canaries)
- Return-oriented Programming (ROP)
- Fuzzing

This is a 10 week course.

Course Outcomes

Upon completion of this course, students should be able to do the following:

- Understand how to reverse engineer programs legally and responsibly.
- Identify and exploit common vulnerabilities in C programs.
- Write powerful scripts to attack vulnerabilities.
- Write safer C programs and avoid vulnerable functions.

Students will demonstrate their understanding of the course material in a final project.

Prerequisites

Grade of C or better in CS 367, or permission from the instructor.

Required Materials

- Vulnerable Kali Linux virtual machine with exercises (Provided by instructors).
- VirtualBox.
- Computer capable of smoothly running a virtual machine.

Grading Policy

Students will be given assignments and readings. There will be a final project due at the end of the course where students will need to use everything that they have learned from the class. The assignments are worth 40% of the final grade, and the final project is worth 60% of the final grade. Regular grading scale A-F will be used.

Capture-the-Flag (CTF) competitions are competitive environments where students can practice what they have learned in class. CTFtime keeps a list of ongoing CTFs, and Mason Competitive Cyber also sends students to CTF competitions. If a student participates in a CTF and creates a write-up for a binary exploitation problem, they may receive extra credit for the class.

Contesting of Grades on any/all submissions must be requested within one week of the item's return. No grade changes will be considered subsequent to that deadline, or after the final project due date.

Honor Code

All students are expected to abide by the GMU Honor Code and the CS Department Honor Code. This policy is rigorously enforced. All class-related assignments are considered individual efforts unless explicitly expressed otherwise (in writing). Review the university honor code and present any questions regarding the policies to the instructors. Cheating on any assignment will be prosecuted and result in a notification of the Honor Committee as outlined in the GMU Honor Code.

Disability Accommodations

Students with a learning disability or other condition (documented with GMU Office of Disability Services) that may impact academic performance should speak with the faculty advisor ASAP to discuss accommodations.

Tentative Schedule

Week	Topics	Assignments
1	● Introduction to ethical hacking. ● Basic integer overflow/underflow. ● Overflow to RIP to call another function.	Review the basics of Python, C, and x86-64 assembly on your own.
2	● Basic Linux shellcoding and testing. ● Developing shellcode for calling /bin/sh. ● Bad characters.	Read "Smashing the Stack For Fun And Profit."
3	● Execute shellcode using stack overflow. ● Off-by-one buffer overflow.	Assignment 0: Stack Overflow.
4	● Introduction to Ghidra ● Format string vulnerabilities. ● Leaking memory.	Assignment 1: Format String Vulnerability.
5	● Introduction to MSFvenom. ● Introduction to Pwntools. ● Fuzzing Basics	Assignment 2: Scripting Exploits.
6	● Exploit mitigation technologies (PIE, ASLR, NX, Stack Canaries)	Assignment 3: Dealing with ASLR.
7	● ROP gadgets and ROP chains.	Assignment 4: ROP chains.
8	● Overwriting PLT and GOT. ● Return-to-libc (ret2libc) Attack.	Final Project Assigned.
9	● Introduction to Z3 Theorem Prover and Angr.	Continue Final Project.
10	● Introduction to heap allocators. ● Basic heap overflow.	Continue Final Project.
11	No Lecture.	Final Project Due