CS 390R - Reverse Engineering and Vulnerability Analysis

Instructors: Brian Levine, Steven Rossi, and Gilbert Hoermann

Important Details:

Credits: 3
Location: CS Building - Room 140
Meeting times: Tuesday/Thursday 2:30-3:45
Required Materials: No Textbook
Prereqs: CS230 OR ECE322

Contact:

Name: Brian Levine
Email: brian@cs.umass.edu

Name: Steven Rossi
Email: srossi@umass.edu
Office: LGRT 212

Name: Gilbert Hoermann
Email: gilberthoerm@umass.edu
Office: LGRT 212

Course Description:

Software is everywhere and many developers aren't aware of how to properly write secure code. We will cover practical skills in reverse engineering and binary exploitation, and delve into the techniques used by hackers for some of the largest security incidents of the century. With a strong understanding of attack patterns, students will be able to implement more secure coding practices into their own code.

This course will start by covering Intel-based assembly, reverse engineering, vulnerability analysis, and various forms of Linux-focused binary exploitation. Throughout this course, we will cover stack, heap, and Linux kernel-based exploitation and dive into common defensive mitigations such as ASLR, NX, and Stack Cookies alongside techniques to bypass each of them. This course will be focused on low-level software written in C, which is why an understanding of the topics covered in CS230 is required.

Learning Objectives:

- Gain a deeper understanding of operating systems and memory management
- Learn how to reverse engineer closed source programs
- Analyze programs for potential security vulnerabilities and learn more secure coding practices

Grading:

Grading will be weighed as follows:

Presentation	-	10%
Labs	-	10%
Homework	-	10%
Projects	-	40%
Midterm & final	-	15/15%

The exact grading scheme may be adjusted during the course. However, a typical breakdown of percentages and final grades for this course are A (93-100), A- (90-92), B+ (87-89), B (83-86), B- (80-82), C+ (77-79), C (73-76), C- (70-72), D+ (67-69), D (60-66), F (0-59).

Late work will generally not be accepted, however, we do adhere to the campus policy around illnesses and other events.

Out of the 18 given questions distributed throughout the 6 projects, we will be dropping the 3 that have the worst impact on your grade.

Schedule

Tue 01/25: *Lecture:* Course Introduction + review of CS230 concepts (gdb, x86 asm, OS)
Thu 01/27: *Lecture:* Continued review of CS230 concepts (gdb, x86 asm, OS)
Tue 02/01: *Lecture:* Reverse Engineering w/ Ghidra - Project 1 Assigned - [Guest Lecturer]
Thu 02/03: *Lecture:* Rev w/ Ghidra+ & pwntools introduction - [Guest Lecturer]
Tue 02/08: *Lab 1:* Practical Reverse Engineering Lab
Thu 02/10: *Lecture:* Common Vulnerability Patterns and Code Auditing Techniques
Tue 02/15: *Lecture:* Exploiting Buffer Overflows - ret2win - Project 2 Assigned (1 due)
Thu 02/17: *Lecture:* Exploiting Buffer Overflows - shellcode
Tue 02/22: *No class* - Monday Schedule
Thu 02/24: *Lab 2:* Walkthrough Buffer Overflow Example
Tue 03/01: *Lecture:* Return Oriented Programming - Project 3 Assigned (2 due)
Thu 03/03: *Lecture:* Format String Exploits
Tue 03/08: *Lecture:* Exploit Mitigations
Thu 03/10: *Lab 3:* Walkthrough ROP
Tue 03/15: *Spring Break*
Thu 03/17: *Spring Break* (3 due)
Tue 03/22: *Lecture:* Advanced ROP
Thu 03/24: *Lecture:* Midterm Review - 48 Hour Take-home Midterm
Tue 03/29: *Lecture:* Glibc Heap Allocator Internals - Project 4 Assigned
Thu 03/31: *Lecture:* Glibc Heap Based Vulnerabilities
Tue 04/05: *Lecture: Important Heap Exploitation Techniques - 1*
Tue 04/07: *Lab 4:* Heap Exploit Walkthrough - Project 5 Assigned (4 due)
Thu 04/12: *Lecture: Important Heap Exploitation Techniques - 2*
Thu 04/14: *Lecture:* Kernel & driver development
Tue 04/19: *Lab 5:* Writing a kernel driver - Project 6 Assigned (5 due)
Thu 04/21: *Lecture:* Kernel Buffer Overflow w/ Shellcode
Tue 04/26: *Presentations*
Thu 04/28: *Presentations*
Tue 05/03: *Presentations/Midterm review* - (6 due)
FINALS: 72 Hour Take-home Final

Class Structure:

We will be meeting twice weekly. Some of these meetings will be lecture-driven, others will have more of a lab structure in which you will attempt problems in a guided fashion. The distribution of these is noted in the schedule section. Attendance to both lecture and lab sections is mandatory.

Labs:

In these in-person labs, we will be walking you through guided examples that are meant to prepare you for the relevant projects. Grading in these discussions will be entirely based on attendance & participation.

Presentations:

Every student will be expected to hold a presentation on a topic related to the class (but not taught in the course). These will be due at the end of the year. These should be on topics relevant to the course, however, we will be giving students a good amount of leeway in terms of what they wish to present. These presentations will be held in groups of two.

Projects:

These take-home assignments make up a big part of your grade. They will be released approximately biweekly, and consist of challenges related to course content. A rubric will be used so that students can receive partial credit based on progress. Each of these projects has 3 sections.

Midterm & Final:

These will be very similar to the projects. They will consist of take-home challenges that you are expected to complete within 48/72 hours. Additionally, you will be expected to create a write-up on these challenges.

Homework:

Homework will be given out weekly and aims to test your knowledge on topics that we covered in class in the preceding week. These will have a stronger focus on short answers instead of programming.

How to Succeed in this Course:

This class will be quite different than most of the classes you have taken in your CS/ECE career thus far. We will focus less on coding and more on understanding the internals of binaries and operating systems to learn how to exploit and defend against various vulnerability categories. That being said, we still want everyone in this class to succeed and do well. We will be holding frequent office hours during which you can come by and get help for the course.

Equality Statement:

The instructors are dedicated to establishing a learning environment that promotes diversity of the students including (but not limited to) race, class, culture, religion, gender, sexual identity, and physical ability. It is important that this is a safe classroom environment. We will practice being generous and respectful members of our class and the computer science community. Please let us know immediately if you notice discriminatory behavior in this class, or feel discriminated against.

Accommodations:

The University of Massachusetts Amherst is committed to making reasonable, effective and appropriate accommodations to meet the needs of students with disabilities and help create a barrier-free campus. If you have a documented disability on file with Disability Services (www.umass.edu/disability), you may be eligible for reasonable accommodations in this course. If your disability requires an accommodation, please notify your instructors as early as possible in the course so that we may make arrangements in a timely manner.

Students who believe they are eligible for accommodations but who have not yet obtained approval through Disability Services should contact the office immediately at (413) 545-0892. If you are a student with a documented disability and are registered with Disability Services, please contact us immediately to facilitate arranging academic accommodations. Reasonable arrangements will be made in accordance with your accommodations provided by Disability Services in the context of this course.

Communication:

We will be using campuswire as the main communication platform for this course. This discussion forum should be your first choice for asking questions as others most certainly have the same question. You should check campuswire before asking your question to see if the same question has already been posted. Think before you post. We expect you to do a reasonable amount of thinking to try to solve your problems before posting for help. Make sure you understand the rules and try to be articulate and clear with your post (again, think before you post). You should post questions related to assignments early rather than waiting until the last minute. If you post a question too close to an assignment deadline, you may not receive an answer before that deadline. When creating public campuswire posts, make sure that you do not reveal crucial information about the assignments to the rest of the class. When in doubt make a private post.

Honesty Policy:

The University of Massachusetts Amherst is committed to providing an equal educational opportunity for all students. If you have a documented physical, psychological, or learning disability on file with Disability Services (DS), you may be eligible for reasonable academic accommodations to help you succeed in this course. If you have a documented disability that requires an accommodation, please notify me within the first two weeks of the semester so that we may make appropriate arrangements.

Since the integrity of the academic enterprise of any institution of higher education requires honesty in scholarship and research, academic honesty is required of all students at the University of Massachusetts Amherst. Academic dishonesty is prohibited in all programs of the University. Academic dishonesty includes but is not limited to: cheating, fabrication, plagiarism, and facilitating dishonesty. Appropriate sanctions may be imposed on any student who has committed an act of academic dishonesty. Instructors should take reasonable steps to address academic misconduct. Any person who has reason to believe that a student has committed academic dishonesty should bring such information to the attention of the appropriate course instructor as soon as possible. Instances of academic dishonesty not related to a specific course should be brought to the attention of the appropriate department Head or Chair. Since students are expected to be familiar with this policy and the commonly accepted standards of academic integrity, ignorance of such standards is not normally sufficient evidence of lack of intent. http://www.umass.edu/dean_students/codeofconduct/acadhonesty/