CS 647: Counter-Hacking Techniques

New Jersey Institute of Technology

Ying Wu College of Computing

Department of Computer Science

Instructor Contact Information

Instructor: Michael Martin
Email: mjm222@njit.edu
Course website: https://canvas.njit.edu/
Online Office Hours:

Wednesdays	17:30 – 18:30
Thursdays	17:30 – 18:30

Statement of Purpose

This course covers techniques that can be used for offensive or defensive goals in computer networks, systems, and applications and follows a strict "learn-by-doing" approach. It seeks to familiarize students with cyber-attacks on both Windows and Linux systems. The course relies heavily on the use of virtual machines, as these provide a uniform, controlled, and safe environment for learning offensive cyber techniques.

The focus of the course is to learn how to defend against cyberattacks by learning how they work in the first place. It is also to encourage the students to think creatively and get into an adversarial mindset when creating or securing systems, networks, and applications. The course forces students to get out of their comfort zone and learn how to quickly pick up new topics to solve the cyber problems at hand. Furthermore, and maybe one of the most important skills developed, is the ability to communicate technical information effectively to others.

The general topics covered are:

- Basic Computer Architecture
- Programming in x86 Assembly
- Buffer Overflow Exploits and Protections
- Format String Exploits

- Network-based Attacks
- Code Hardening
- Ethical Hacking
- Cybersecurity Ethics
- Malware Analysis and Reverse Engineering
- Hardware Side Channel Attack (Spectre & Meltdown)

Prerequisites

Courses:

- CS 645
- CS 656
- CS 608 (Recommended)

Topics:

- All students should have a basic understanding of computer architecture.
- Prior experience working with assembly language will be helpful (x86 specifically) but is not required.
- All students should have some prior experience with the C programming language or the ability to self-learn.
- All students should have prior experience using a Linux Operating system and with basic Linux commands.
- Prior experience using and configuring virtual machines will be helpful but is not required.
- All students should understand basic cryptographic principles such as hashing and public key infrastructure.

Required Materials

This course **does not require any text**; however, it will reference information and material from the following sources:

- "Hacking: The Art of Exploitation" 2nd Edition, *by Jon Erickson*,
  ISBN-13: 978-1593271442
  ISBN-10: 1593271441
- "Introduction to Computer Security", *by M. Goodrich and R. Tamassia*,
  Addison Wesley, 2010,

ISBN: 0321512944

- "Computer Systems: A Programmer's Perspective" 3rd Edition, *by Randal Bryant and David O'Hallaron*
  ISBN-13: 978-0134092669
  ISBN-10: 013409266X
- "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software" 1st Edition, *by Michael Sikorski and Andrew Honig*
  ISBN-13: 978-1593272906
  ISBN-10: 1593272901

All students must own, or have access to, a computer with virtualization capabilities. It is recommended to have at least 50-60GB of available storage space for the virtual machines and at least 4 GB of memory. Many host operating systems are supported but Windows 10 is recommended. If the student cannot run the course VM on their machine, they can run it in Amazon Web Services. Costs for the semester are likely to be low if the student is diligent about managing their VM uptime.

A webcam is required to take the proctored exams via Respondus Lockdown Browser and Respondus Monitor.

Course Learning Outcomes

After completion of this course, students will be able to:

- Test and discover vulnerabilities in applications
- Develop exploits for vulnerable local/remote Linux applications
- Develop exploits for vulnerable local/remote Windows applications
- Recommend appropriate mitigation countermeasures
- Demonstrate the ability to read and write x86 assembly code
- Describe the virtual memory layout of an application
- Describe modern buffer overflow defensive mechanisms
- Identify buffer overflow vulnerabilities in programs
- Perform buffer overflow attacks on vulnerable programs
- Write or generate custom shellcode
- Recommend mitigations for buffer overflow attacks
- Identify a heap overflow in a program

- Perform a heap overflow on a vulnerable program
- Exploit a format string vulnerability to leak virtual memory
- Chain exploits together to take command and control of a system
- Explain the importance of input validation
- Describe what a penetration test is
- Explain the general attack methodology for a hacker
- Describe the goals and importance of malware analysis
- Create a safe environment to perform malware analysis
- Identify host-based and network-based indicators of compromise
- Describe and recognize common malware anti-debugging techniques
- Recommend malware inoculation procedures
- Effectively communicate a summary of vulnerabilities and appropriate mitigations to a client or colleague

Course Structure and Components

Announcements

Each week, typically when a new Module is released, a "Housekeeping" announcement that contains crucial information will be posted. This part of the course is by design and students must read these announcements carefully. **Please be on the lookout for these, and other ad hoc announcements, and make sure to read them carefully; it is very important!**

Lectures

Lecture slides, supplemental materials, and lecture recordings are made available weekly through Canvas modules. The lectures aggregate material from personal experience and training as well as the text references mentioned in the Required Materials section.

The lecture recordings are mostly broken down into short videos with descriptive titles. This should make it easier to go through at your own pace and reference back to specific sections of the lecture.

You will find that the majority of the lectures include hands-on challenges, administered through the use of virtual machines. Attempting the challenges is not optional; it is a structured part of the course. Many of the topics presented are low-level, semi-advanced topics that require hands-on practice. The lectures follow a learn-by-doing approach where the students are presented with a concept/technique and then challenged to implement it themselves in a controlled environment. The video following the challenge will include a walk-through solution to explain it in detail.

Weekly Assignments

The weekly assignments are cyber vulnerability assessments of given programs. These are specifically designed to promote an adversarial mindset as well as student interaction. Most importantly, they require the student to summarize and communicate the risks posed by the identified vulnerabilities and mitigation strategies to a would-be client. Most of them are directly related to the lecture material, however, they all require the student to figure out how to apply learned techniques to unique problems. Depending on each student's technical background and experience, some students may need to do their own external research. All the assessment materials exist in a virtual machine built for this course.

Students must adhere to the course writing guide for the assessment reports. The writing guide is provided on Canvas and covers writing style as well as other requirements for the reports. A rubric is provided for the reports to allow students to have a more granular view of their grades.

Once the assessment is complete and the student has crafted a successful exploit, they are to submit their write-up according to the course's Report Writing Guide. All submissions MUST include a text-based version of the content of the challenges' flag files. Students MUST also include in their submission a screenshot that contains the output of the flag file printed to the terminal.

Every student must complete an anonymous peer review of one other student's report. Assessment reports are due on Sunday nights at midnight, at which time peer reviews are assigned. Students should complete the peer review as soon as possible, but within a week once assigned. Students should use the provided rubric to review their classmate's work. Failure to complete peer reviews, or if done carelessly, may result in a reduced grade on the assignment.

Projects

The course consists of two group projects. Collaboration between groups is prohibited. The first project will demonstrate the students' ability to write a working program in x86 assembly language. The second project will demonstrate the student's ability to think creatively and craft an attack that exploits a buffer overflow vulnerability to gain arbitrary code execution on a machine. The exploit must bypass all the modern buffer overflow defenses. Students will also need to create a well-written technical report to describe their technique and procedures and provide a live demonstration via Webex.

Exams

The midterm and final exams are closed-book exams. Many problems are open-ended questions that probe your knowledge of core concepts taught throughout the course. These exams will be held online with Respondus LockDown Browser and Monitor, and you will need a webcam to be proctored. For more information, please see the "Online Exams and Proctoring" page at the top of Module 0x0 in Canvas.

Students may instead opt to take the exams on campus using only Respondus LockDown Browser without the Monitor webcam component. This is the recommended mode to take the exam. A time and location will be announced, and you can email the instructor that you'll be attending in person.

Schedule of Assignments

Important Dates

Spring 2023 Academic Calendar

Sept	4	Labor Day. University Closed
Sept	5	First Day of Classes
Sept	11	Last Day to Add/Drop a Class
Sept	11	Last Day for 100% Refund, Full or Partial Withdrawal
Sept	12	W Grades Posted for Course Withdrawals
Sept	18	Last Day for 90% Refund, Full or Partial Withdrawal - No Refund for Partial Withdrawal after this date
Oct	2	Last Day for 50% Refund, Full Withdrawal

Oct	23	Last Day for 25% Refund, Full Withdrawal
Nov	13	Last Day to Withdraw from Classes
Nov	21	Thursday Classes Meet
Nov	22	Friday Classes Meet
Nov	23	Thanksgiving Recess Begins. No Classes
Nov	26	Thanksgiving Recess Ends
Dec	13	Last Day of Classes
Dec	14	Reading Day 1
Dec	15	Reading Day 2
Dec	16	Saturday Classes Meet
Dec	17	Final Exams Begin
Dec	23	Final Exams End
Dec	25	Final Grades Due

Weekly Outline

Week Number	Dates	Topic	Assignments Due	
0	3 Sep – 10 Sep	Course Introduction		
1	11 Sep – 17 Sep	Basic Computer Architecture	**10 Sep**	Syllabus Acknowledgment and Introduction Video
2	18 Sep – 24 Sep	x86 Assembly Part 1 x86 Assembly Part 2 ***Project 1 Assigned***	**17 Sep**	Pippin Assessment Report
3	25 Sep – 1 Oct	***Work on Project 1***	**1 Oct**	Project 1 Due
4	2 Oct – 8 Oct	Exploitation Buffer Overflows Part 1	**8 Oct**	Merry Assessment report
5	9 Oct – 15 Oct	Buffer Overflows Part 2	**15 Oct**	Sam Assessment Report
6	16 Oct – 22 Oct	Midterm Exam ***DATE: Saturday, October 21st, 2023*** ***The time and location will be announced on Canvas.***		
7	23 Oct – 29 Oct	Buffer Overflows Part 3 Buffer Overflows Part 4 Format String Vulnerabilities ***Project 2 Assigned***		

8	30 Oct – 5 Nov	Work on modules 0x8 and project 2		
9	6 Nov –12 Nov	Work on project 2 and demos	**12 Nov**	Live demos must be scheduled
10	13 Nov – 19 Nov	Introduction to Cybersecurity Ethics	**6 Nov**	Demos for Project 2 scheduled
11	21 Nov – 27 Nov	Network-Based Attacks and Code Hardening	**20 Nov**	Legolas Assessment Report (Group Assignment)
12	27 Nov – 3 Dec	Introduction to Ethical Hacking		
13	4 Dec – 10 Dec	Malware Analysis and Reverse Engineering		
14	11 Dec – 17 Dec	Hardware Side Channel Attack Lab (Spectre & Meltdown)		
15	17 Dec – 23 Dec	Final Exam ***Time and location TBD by Registrar***	**17 Dec**	Frodo Assessment Report

Grading

Letter Grade	Significance	Calculation
A	Excellent	90% and above
B+	Good	85% - 89%
B	Acceptable	80% - 84%
C+	Marginal Performance	75% - 79%
C	Minimum Performance	65% - 74%
F	Failure	Below 65%

Grading based on assignments:

- Weekly Assignments 20%
    - Introduction Video
    - Pippin Assessment
    - Merry Assessment
    - Sam Assessment
    - Frodo Assessment
- Legolas Assessment (Project 2) 10%
- Programming Project (Project 1) 10%
- Midterm Exam 30%
- Final Exam 30%

To earn full points on the vulnerability assessments, students must:

- Craft a working exploit to obtain the flag file.
- Write a vulnerability assessment report which includes all the required sections and follows the course's writing format guide.
- Complete one anonymous peer review of another student's assessment.
- Upon the instructor's request, demo the exploit in Webex.

Assignments have an 8-hour period after the due date for which they can still be submitted with a 50% deduction in score. Do not wait until just before the deadline to submit. After the eight hours pass, assignments will no longer be accepted without prior approval.

Submissions of all assignments MUST be on Canvas for accountability. I will not accept submissions through email or any other form.

Responsiveness and Availability

Under normal conditions, emails will be responded to within 24 hours. If you have not heard back from me within 24 hours feel free to send another email. Assignments will be graded weekly. Projects and exams will be graded within two weeks of their due date.

IMPORTANT: Start the subject line of all emails with **CS647:** <Your subject here>
This will allow me to filter out course-related emails so I can respond more promptly.

Students are encouraged to use the discussion thread in Canvas to ask general, course-related questions. Every student on the course comes from various backgrounds and experiences. Asking questions is great and can benefit the entire class. I ask that all students ask questions via email for me to be able to answer but check the dedicated discussion first to see if their question has already been answered.

Feedback

Feedback for the course assignments will be delivered using the comments feature in Canvas. Any details that require further discussion will take place over email, in an online chat, or over the phone.

Academic Integrity

The below statement also goes for copying code. Students will have individual assignments and group assignments. The source code is not to be shared. Information on vulnerability assessments is not to be shared. You can read more detailed information about NJIT's academic integrity policy here.

Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree to which you are working. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the academic code of integrity policy which is found here.

Please note that it is my professional obligation and responsibility to report any academic misconduct to the Dean of Students Office. *Any student found in violation of the code by cheating, plagiarizing, or using any online software inappropriately will result in disciplinary action. This may include a failing grade of F, and/or suspension or dismissal from the university.*

There will be no warnings or second chances about cheating. It is your responsibility to understand specifically what constitutes academic dishonesty. Ignorance is not an excuse or a defense. It is also your responsibility to understand the rules for properly citing the work of others in the submission of class work. Improper citation with a simple “copy/paste” from online sources may be grounds for failure of the assignment and/or the course. If you have any questions about the Code of Academic Integrity, please contact the Dean of Students Office at dos@njit.edu.

Canvas Accessibility Statement

Please note: If you need an accessible version of the course PowerPoint slides, open the files with Microsoft Office 365.

Student Support

Canvas Orientation

For students who are unfamiliar with Canvas, this is a link to self-paced student orientation.

Resources for NJIT Online Students

Use the following link to visit NJIT Online Students Resources.

Last updated: September 2023