NYU TANDON SCHOOL OF ENGINEERING

Course Syllabus

Computer Science and Engineering

Introduction to Offensive Security

Course Information

Course Prerequisites

None.

Course Description

This course aims to teach offensive security in the context of Capture the Flag (CTF) competitions. We will cover common flaws in websites, techniques and methods to reverse x86 assembly, exploitation strategies for binaries, and basic cryptographic flaws.

Course Objectives

Learn technical skills focused on offense in:

- Recognizing web-based vulnerabilities and developing relevant attacks
- Assessing the logic of source-less code via binary reverse engineering and solving programming challenges based on the functionality of the binary
- Employing memory corruption tactics and strategies to exploit binaries
- Assessing cryptographic implementations to identify vulnerabilities and write related exploits to demonstrate understanding

- Understand how the CTF examples can translate to real-world software
- Generally become familiar with CTF competitions as a competitor

---

Course Structure

Throughout the course, we will be running a CTF which is available any time at https://class.osiris.cyber.nyu.edu.

Each week, a set of challenges related to that week's material will be released and marked as "hot" which indicates that they count towards that week's homework. Additionally, you will be required to compete in at least one CTFTime-ranked CTF, and provide a writeup about at least one non-trivial problem that your team worked on. We encourage you to form teams with classmates and submit a group writeup. Please send an email to the teachers with your team name and write up within one week of competing, no later than final lecture.

Lectures will take place each week. They will begin with discussing the homework assignment that was just completed (to include solving it live if time provides), as well as discuss the content for the upcoming challenges. There will be slides distributed at the start of the class.

The midterm and final will have the same structure as other weeks, but will be more complex and cumulative. For instance, the midterm will encompass all previous sections, include more challenges and more points, as well as more time to complete. In past classes, this means that there are 4 challenges for a total of 1000 points (600 points required for passing), and 2 weeks to accomplish.

The final will be similar: it will be one very large challenge that will require multiple exploits to accomplish. The final will be considered optional and will replace one other weeks homework if accomplished.

Additional extra credit opportunities (replace a weeks grade) may occur at the discretion of the professor. In past classes, this is typically limited to the final and one other opportunity, both of which are quite complex. (Read: it makes more sense to not need to use the extra credit and just accomplish each as it comes).

Weekly Structure

Part I: Introduction to CTF

Introduction

- What is CTF
- Syllabus overview
- Environment/Tooling Setup
- First simple warmup challenges (basic programming warmups) assigned

Part II: Web-Based Vulnerabilities (WEBVULN)

Intro to Web-Based Vulnerabilities

- SQL Injection
- XML Entity Injection
- XSS
- Previous week CTF challenges due ; next week assigned

Additional Web-Based Vulnerabilities

- Command Injection
- File Inclusion
- Serialization
- Previous week CTF challenges due ; next week assigned

Part III: Reverse Engineering (RE)

Intro to Reverse Engineering

- Basics of assembly
- x86 semantics
- Techniques / Strategies

- Debugging
- Previous week CTF challenges due ; next week assigned

Further Reverse Engineering

- Structs
- Symbolic Execution
- Previous week CTF challenges due ; next week assigned

Part IV: Exploitation (RCE)

Intro to Exploitation

- Control Flow
- Stack Overflow
- Previous week CTF challenges due ; next week assigned

Further Exploitation

- Structs
- Symbolic Execution
- Previous week CTF challenges due ; midterm assigned

Defeating Exploit Mitigations

- Binary layout
- Mitigations and bypasses
- Return-Oriented Programming
- Midterm due ; next week assigned

31MAR Introduction to Heap Exploitation

- Heap basics
- Previous week CTF challenges due ; next week assigned

07APR Further Heap Exploitation

- More complex heap primitives and exploitation
- Previous week CTF challenges due ; next week assigned

Part IV: Cryptography (CRYPTO)

Introduction to Cryptanalysis

- Frequency analysis
- XOR
- Previous week CTF challenges due ; next week assigned

Further Cryptanalysis

- Block ciphers
- Common RSA attacks/mistakes
- Padding oracle attacks
- Hash-length extension
- Previous week CTF challenges due ; next week assigned

Special Topic

- TBD based on class voting
- Final exam assigned

Final Exam Topic Discussion

- CTF Participation and Write-up Deadline
- **Final Assignment Due**

Learning Time Rubric

Please modify the below table to represent the breakdown of learning time in each week of your course.

Learning Time Element	Asynchronous* / Synchronous**	Time on Task for Students (weekly)	Notes
Reading Assignments / Recorded Lecture	Asynchronous	2.5 hours	Video format. Expect quizzes throughout the module or weekly chapter readings

Weekly Discussion Board	Asynchronous	1.5 hours	Students are expected to post initial response to weekly topic questions. See Interaction Policy.
Assessment (Labs and Programming assignments)	Asynchronous	2 hours	Students submit their assignment by [the end of the week]
Reading Assignment	Asynchronous	2 hours	Reading assigned textbook chapters and journal articles.
Live webinars	Synchronous	2 hours	Group discussion in class, live, overly weekly chapter

Course Communication

Grade Calculation

Weekly grading will be based on the number of points you score in the CTF each week. If you score at least 300 CTF points in the week, you will receive credit for that week's homework. Points will be tallied for hot challenges at the beginning of class one week after it is assigned. Past challenges will continue to be available for the entire semester, and we recommend that you solve as many of them as you can.

The final grade will be calculated as follows:

- Homework will be worth 90% of your final grade.
- CTF participation & writeup will be worth 10% of your final grade, however this is **required**. You will not pass the course if you do not compete in a CTFTime CTF.

Announcements

Announcements will be posted on NYU Classes on a regular basis. You can locate all class announcements under the *Announcements* tab of our class. Be sure to check the class announcements regularly as they will contain important information about class assignments and other class matters.

Email

You are encouraged to post your questions about the course in the Forums discussions on NYU Classes. This is an open forum in which you and your classmates are encouraged to answer each other's questions. But, if you need to contact me directly, please email me. All homework, labs or programming assignments related questions must be researched first on own time, then posted on forums, then discussed with TAs during weekly reviews, and then can be forwarded to me. Typically, you can expect a response within 48 hours.

Readings

There is no textbook. Supplemental readings will be provided each week that will provide further discussion about the topics covered. These readings will typically be blog posts or technical articles.

Assignments and Exams

Exams Administered and Proctored Online

Exams in this course are administered through NYU Classes. You are required to arrange an online proctor for your exams via ProctorU. More information on ProctorU and scheduling proctoring sessions can be found on Tandon Online's website.

Exams Administered On Paper and Proctored Remotely

Exams in this course are administered via paper and pencil. If you are not able to attend an exam session on-campus, you are required to secure in-person proctoring arrangements near your location. Tandon Online's website.

University Policies

Moses Center Statement of Disability

Academic accommodations are available for students with disabilities. Please contact the Moses Center for Students with Disabilities (212-998-4980 or mosescsd@nyu.edu) for further information. Students who are requesting academic accommodations are advised to reach out to the Moses Center as early as possible in the semester for assistance.

NYU Tandon School of Engineering Policies and Procedures on Academic Misconduct[1]

A. Introduction: The School of Engineering encourages academic excellence in an environment that promotes honesty, integrity, and fairness, and students at the School of Engineering are expected to exhibit those qualities in their academic work. It is through the process of submitting their own work and receiving honest feedback on that work that students may progress academically. Any act of academic dishonesty is seen as an attack upon the School and will not be tolerated. Furthermore, those who breach the School's rules on

[1] Excerpted from the Tandon School of Engineering Student Code of Conduct

academic integrity will be sanctioned under this Policy. Students are responsible for familiarizing themselves with the School's Policy on Academic Misconduct.

B. Definition: Academic dishonesty may include misrepresentation, deception, dishonesty, or any act of falsification committed by a student to influence a grade or other academic evaluation. Academic dishonesty also includes intentionally damaging the academic work of others or assisting other students in acts of dishonesty. Common examples of academically dishonest behavior include, but are not limited to, the following:
    a. Cheating: intentionally using or attempting to use unauthorized notes, books, electronic media, or electronic communications in an exam; talking with fellow students or looking at another person's work during an exam; submitting work prepared in advance for an in-class examination; having someone take an exam for you or taking an exam for someone else; violating other rules governing the administration of examinations.
    b. Fabrication: including but not limited to, falsifying experimental data and/or citations.
    c. Plagiarism: intentionally or knowingly representing the words or ideas of another as one's own in any academic exercise; failure to attribute direct quotations, paraphrases, or borrowed facts or information.
    d. Unauthorized collaboration: working together on work that was meant to be done individually.
    e. Duplicating work: presenting for grading the same work for more than one project or in more than one class, unless express and prior permission has been received from the course instructor(s) or research adviser involved.
    f. Forgery: altering any academic document, including, but not limited to, academic records, admissions materials, or medical excuses.