CMSC389J: Introduciton to Reverse Engineering

Christopher Brown, Drake Petersen

Prequisites and Description

Prerequisites: C- or better in CMSC250 and CMSC216
Credits: 1
Faculty Advisor: Jonathan Katz
Course Instructors: Christopher Brown, Drake Petersen (contact below)

The focus of Introduction to Reverse Engineering is for students to gain experience in a high demand practice of cybersecurity though weekly reversing assignments. In this course students will be challenged to think outside the box in order to solve reversing challenges. Assignments will be challenges focused in the analysis of Linux binaries including various malware. Various tools for reversing will be demonstrated in class such as Binaryninja/IDA, gdb, strace, objdump, readelf, pev, etc. The goal of this class is to have students go from a beginner to an intermediate level reverse engineer. Students will be expected to have some assembly experience (calling conventions, stack/heap, registers), but we will refresh and introduce students to the x86 assembly language.

Time and Location

Day/Time: Friday 12:00pm-12:50pm
Location: CSI 3118

Textbooks & Required materials

No textbooks required. Readings will be recommended as the semester goes on. A laptop with VMware or VirtualBox. **Class VM will be provided.**

Topics Covered

- Static binary analysis
- Dynamic binary analysis
- PE32/+ vs ELF32/64
- API Hooking
- Legacy code analysis
- Malware
- Fuzzing/Symbolic Execution

Software used

Virtualization software	**Static Analysis Tools**
VMware Virtual Box Linux VM (Kali reccommened)	Binary Ninja (Free version) Ida 7 Free Pev/Readelf/Objdump Shellen
Symbolic Execution	**Dynamic Analysis Tools**
Valgrind	GDB/EDB

Homeworks

Homeworks will be 1-2 page write ups on the analysis and findings of the provided files. Some will be question driven, while others are open ended. The point to the write ups is to explain and document the reversing process, not to have everyone get the same answer. This is to benefit those who attempt the homeworks, the main focus of these writing assignments is to gauge the students' thinking and methodology. Correct answers will factor into this but effort and methodology will hold the most weight.

HW #0: What is your RE experience? Ethics
HW #1: Bufferoverflow Attack
HW #2: Static – Analyze a binary
HW #3: Static – Get a valid key
HW #4: Dynamic – Get a key (Packed)
HW #5: Dynamic – Get a key (Obf.)
HW #6: API Hooking – write an LD-PRELOAD
HW #7: Crypto – Break the scheme
HW #8: AP RE – Whats the damage? (malware)
HW #9: AP RE – Update old database
HW #10: AP RE – Swap out Legacy Library
HW #11: Fuzzing/Sym. Exec. – Exploit me

Quizzes

Three 10-15 minute quizzes, spread out on important over arching topics that we want to emphasize. Quizzes will be higher level, not as technical as what may be found as homework questions. Quizzes will be themed: Static Analysis, Dynamic Analysis, and Forensics.

Grading

Homeworks and quizzes will be scanned and graded using Gradescope. Students will have 2 weeks to request a regrade from the time of receiving their grade. Grades will be posted on the CS Grade Server.

You are responsible for all materials discussed in lecture and posted on the class repository, including announcements, deadlines, policies, etc.

Quizzes	27%
Homeworks	73%

Schedule

Week #	Lecture Topic	Assignment
0 (2/1)	What is RE? Ethics, C review, Assembly review (x86/64), Calling conventions	
1 (2/8)	Advanced Assembly: Linking, Bytecode, Instruction format	HW #0 (due)
2 (2/15)	Static Analysis I: Linux file headers, file types, Dynamic libraries	HW #1 (due)
3 (2/22)	Static Analysis II: Disassembly, Libc Basic block analysis, patching	HW #2 (due)
4 (3/1)	Quiz #1, Dynamic Analysis I: gdb/edb, packers	HW #3 (due) Quiz #1: Static Analysis
5 (3/8)	Dynamic Analysis II: Obfuscation, self modifying code	HW #4 (due)
6 (3/15)	API hooking, LD-PRELOAD	HW #5 (due)
8 (3/22)	**Spring Break**	**Spring Break**
7 (3/29)	Quiz #2, Applied RE: Breaking Crypto	HW #6 (due) Quiz #2: Dynamic Analysis
9 (4/5)	Applied RE: Malware triage and analysis	HW #7 (due)
10 (4/12)	Applied RE: Legacy code base reversing I	HW #8 (due)
11 (4/19)	Applied RE: Legacy code base reversing II	HW #9 (due)
13 (4/26)	Quiz #3: Symbolic Execution/Fuzzing, Side channel	HW #10 (due) Quiz #3: Applied RE
14 (5/3)	Malware & Detection: Trojans, Protocols, Snort	HW #11 (due)
15 (5/10)	Special Topic: TBD	No Homework

Out of class communication with course Staff

We will interact with students outside of class in primarily two ways: in-person during office hours and piazza. Email should only be used for emergencies and not class related questions (e.g., homework).

Faculty Advisor:

- Jonathan Katz - **jkatz AT cs.umd.edu**

Course factors:

- Christopher Brown - **chris03 AT terpmail.umd.edu**
  - Office Hours: TBD

- Drake Petersen - **drakemp AT terpmail.umd.edu**
  - Office Hours: TBD

Excused Absence and Academic Accommodations

See the section titled "Attendance, Absences, or Missed Assignments" available at Course Related Policies.

Disability Support Accommodations

See the section titled "Accessibility" available at Course Related Policies.

Academic Integrity

Note that academic dishonesty includes not only cheating, fabrication, and plagiarism, but also includes helping other students commit acts of academic dishonesty by allowing them to obtain copies of your work. In short, all submitted work must be your own. Cases of academic dishonesty will be pursued to the fullest extent possible as stipulated by the Office of Student Conduct.

It is very important for you to be aware of the consequences of cheating, fabrication, facilitation, and plagiarism. For more information on the Code of Academic Integrity or the Student Honor Council, please visit http://www.shc.umd.edu.

Course Evaluations

If you have a suggestion for improving this class, don't hesitate to tell the instructor or TAs during the semester. At the end of the semester, please don't forget to provide your feedback using the campus-wide CourseEvalUM system. Your comments will help make this class better.

Thanks to https://github.com/UMD-CS-STICs/389Cfall18 for the syllabus formatting