Software Reverse Engineering and Malware Analysis

CAP 5137, Fall 2022
Department of Computer Science, Florida State University

Class time and location

Monday and Wednesday, 06:35-07:50pm, BEL (Bellamy Building) 0180
(or FLEX mode via zoom meeting 179 200 917 (with password: 518581) (zoom link: https://fsu.zoom.us/j/179200917?pwd=Z3p3M0JUaGs4a0R6NSsyMXlkakZtdz09).

Instructor

- Instructor: Xiuwen Liu
  - Email: liux@cs.fsu.edu
  - Home page: http://www.cs.fsu.edu/~liux
- Teaching assistant: Sajib Biswas
  - Email: sb19t@my.fsu.edu
- Office
  - Xiuwen Liu, 259 Love Building (LOV); Phone: (850) 644-0050
- Office Hours:
  - Xiuwen Liu, Monday and Wednesday, 5:15 - 6:15pm via zoom meeting 850 644 0050 (link https://fsu.zoom.us/j/8506440050) and by appointments for in-person and additional zoom meetings.
  - Sajib Biswas, Tuesday, 4:00 – 5:00pm, via zoom meeting (Meeting ID: 884 0076 9148 Passcode: 3sLV59 (link: https://us02web.zoom.us/j/88400769148?pwd=NlFaZlJ1K3RMaHJ4UDJtaFhZaElWZz09)

Class Home Page

http://www.cs.fsu.edu/~liux/courses/reversing/index.html. This web site contains the up-to-date information related to this class such as news, announcements, assignments, lecture notes, and links to resources that are helpful to this class. Besides the web pages, Canvas will be used to communicate changes and updates and post grades for this class. In particular, we will send emails using email addresses in the Canvas system; so please make sure that your email address on record is current.

Rationale

Computers and communication technologies have been incorporated into many applications and have fundamentally changed many aspects of human activities. Unfortunately, the changes have also created new problems, from spyware that steal data, computer viruses and worms that destroy data, to network enabled weapons, to cyber wars that can disable companies and even countries (such as Stuxnet). All these problems are related to computer security. Due to its paramount importance, computer security is not just one academic research area. Many security products are installed on typical computers; in the United States, there are multiple federal agencies dedicated to computer security; the computer security is a multibillion industry that is estimated to grow steadily. Computer security related issues have been widely recognized in software development companies. As computer security techniques evolve continuously along with product improvements and new service opportunities, computer security is and will remain to be an important and valuable area in the perceivable future with new career opportunities. As all computers (including communication devices) execute instructions, a fundamental requirement to achieving security is to be able to analyze binary programs as source code is not available in many situations and security often relies on implementation details not present in source code. This course is designed to cover the basic principles and techniques for software reverse engineering so that you can audit binary programs and analyze firmware samples and other stripped binaries.

Course Description

This course covers fundamental problems, principles, and techniques in software reverse engineering of binaries including static analysis techniques, disassembly algorithms, dynamic analysis techniques, automated static and dynamic analysis techniques, malware analysis techniques, anti-analysis techniques, and malware obfuscation and packing techniques; many of the techniques will be demonstrated and practiced using IDA. It also involves research opportunities to analyze new malware samples and firmware samples, and develop new analysis tools.

Prerequisites

CDA 3100 – Computer Organization I; having a good understanding of instruction set architectures (registers, instruction encoding and decoding, and memory organization) and basic data types, data structures, function calls (calling conventions), and memory layout of programs; be able to understand x86 and other assembly (assuming that instruction reference manuals are available); having a general understanding of computer security.

Course Objectives

Upon successful completion of this course of study, the student will be able to:

- Recognize commonly used file formats
- Extract information from files in PE format and ELF format
- Dissemble code segments using the linear sweep and recursive descent disassembly algorithms
- Recognize the commonly used function calling conventions in disassembled files
- Construct basic blocks and calling graphs
- Identify conditional execution constructs in disassembled files
- Identify loop constructs in disassembled files
- Identify switch statements using a jump table in disassembled files
- Perform data flow analysis
- Recognize commonly used anti-disassembly techniques and overcome them in GHIDRA/IDA
- Execute malware samples safely in a virtual machine
- Use a debugger to monitor program execution
- Analyze an executable file in GHIDRA/IDA
- Recognize commonly used anti-debugging techniques and overcome them in GHIDRA/IDA
- Recognize commonly used anti-virtual machine techniques and overcome them in GHIDRA/IDA
- Recognize buffer overflow vulnerabilities in disassembled files
- Recognize and decode commonly used encoding algorithms in disassembled files
- Recognize and overcome commonly used obfuscation techniques in disassembled files
- Recognize commonly used packing methods in disassembled files
- Analyze malware samples packed using common packing techniques in GHIDRA/IDA
- Recognize commonly used malware mechanisms in disassembled files
- Incorporate new methods documented in research papers in reversing binaries
- Incorporate new malware analysis techniques documented in research papers in malware analysis
- Use scripts in GHIDRA/IDA

- Analyze firmware in GHIDRA/IDA

Textbook and Course Materials

Required textbook: “Practical Malware Analysis: The Hands-on Guide to Dissecting Malicious Software” by Michael Sikorski and Andrew Honig (published by No Starch Press, 2012).

Recommended readings: “Hacking: The Art of Exploitation, 2nd Edition” by Jon Erickson**:** this is a book with accurate and detailed descriptions and commands of common vulnerabilities and corresponding exploits. **“The IDA PRO Book: The Unofficial Guide to the World’s Most Popular Disassembler, 2nd Edition”** by Chris Eagle (published by No Starch Press, 2011); this book provides a comprehensive coverage about IDA Pro, one of the most commonly used reverse engineering tools. “**The GHIDRA Book: The Definitive Guide**” by Chris Eagle and Kara Nance (Penguin Random House Publisher Services, 2020) ; this book provides a systematic coverage of Ghidra.

In addition to the textbooks, papers and notes from the literature will be distributed along with the lectures.

Student Responsibilities

Attendance is required for this class. In case that it is necessary to skip a class, a student is required to notify the instructor beforehand; the absence is excused if it is allowed by the University Attendance Policy (see below). The penalty for each unexcused absence is 10% reduction of attendance points (see the Grading Policy below); a student will receive 0 for attendance points if he or she has ten or more unexcused absences through the semester. In both excused and unexcused cases, the students are responsible for making up missed materials. Participation in in-class discussions and activities is also required. All submitted assignments and projects must be done by the author(s). It is a violation of the Academic Honor Code to submit other’s work and the instructor of this course takes the violations very seriously.

University Attendance Policy - Excused absences include documented illness, deaths in the family and other documented crises, call to active military duty or jury duty, religious holy days, and official University activities. These absences will be accommodated in a way that does not arbitrarily penalize students who have a valid excuse. Consideration will also be given to students whose dependent children experience serious illness.

This course will cover certain techniques to exploit and break down known systems in order to demonstrate their vulnerabilities. It is **illegal**, however, to practice these techniques on others' systems. The students will be **liable** for their behaviors and therefore consequences.

Assignments and Projects

About six homework assignments (most of them will involve using GHIDRA/IDA Pro) will be given along with the lectures and they need to be done individually and turned in. There will be a hands-on project, where a malware or firmware sample will be analyzed using the techniques studied in this course. There will also be a term project, where a student proposes a project that uses the techniques and skills learned in this course. In addition, there will be a research paper assignment, where a student chooses a research paper (from an approved list or one approved by the instructor) and writes a report on how the techniques in the paper can be used to analyze binary programs and malware. There will be a midterm exam and a final exam.

Grading Policy

Grades will be determined as follows:

Assignment	Points	Assignment	Points
Class Attendance	5 %	Final Exam (cumulative)	15 %
In-class Participation	5 %	Midterm Exam	15 %
Homework Assignments	35 %	Hands-on Project	10 %
Research Paper Assignment	5 %	Term Project	10 %

Grading will be based on the weighted average as specified above and the following scale will be used (S is the weighted average on a 100-point scale):

Score	Grade	Score	Grade	Score	Grade
$93 \leq S$	A	$80 \leq S < 83$	B-	$67 \leq S < 70$	D+
$90 \leq S < 93$	A-	$77 \leq S < 80$	C+	$63 \leq S < 67$	D
$87 \leq S < 90$	B+	$73 \leq S < 77$	C	$60 \leq S < 63$	D-
$83 \leq S < 87$	B	$70 \leq S < 73$	C-	$S < 60$	F

Late Penalties

Assignments are due at the beginning of the class on the due date. Assignments turned in late, but before the beginning of the next scheduled class will be penalized by 10 %. Assignments that are more than one class period late will **NOT** be accepted.

Submission and Return Policy

All tests/homework assignments/projects will be returned as soon as possible after grading but no later than two weeks from the due date.

Tentative Schedule

Here **Mal** refers to the “Practical Malware Analysis: The Hands-on Guide to Dissecting Malicious Software”.

- Week 1: Introduction (**Mal**: Chapters 0, 1, and 5)
  - **Fundamentals**
    - General introduction to software reverse engineering (static analysis, dynamic analysis, symbolic execution approaches) and applications
    - Steps in software reverse engineering
  - **Practice**
    - Overview of GHIDRA (mainly) and IDA (briefly)
- Week 2: Instruction set architectures and file formats; disassembly algorithms (**Mal**: Chapter 4)

    - **Fundamentals**
        - Instruction set architectures
        - Common file formats and file-level reverse engineering tools
        - Fundamental principles and techniques to disassembly (linear sweep disassembly and recursive descent disassembly)
    - **Practice**
        - GHIDRA Components, data displays, and disassembly navigation
- Week 3: Binary program analysis - I (**Mal**: Chapter 6)
    - **Fundamentals**
        - Recognizing high-level constructs (if-then-else, loop structures, and switch statements)
    - **Practice**
        - GHIDRA disassembly manipulation, datatypes, and data structures
- Week 4: Binary program analysis – II
    - **Fundamentals**
        - Control flow and data flow analysis
        - Calling conventions
        - Function recognition
        - Windows internals (Windows libraries, Windows data structures and prototypes and Windows organizations)
    - **Practice**
        - GHIDRA cross-referencing and graphing
- Week 5: Decompilation (Papers from the literature)
    - **Fundamentals**
        - Decompilation challenges
        - Introduction to decompilation techniques
    - **Practice**
        - GHIDRA decompiler
- Week 6: Anti-disassembly techniques (**Mal**: Chapter 15)
    - **Fundamentals**
        - Challenges of disassembly techniques
        - Common anti-disassembly techniques
    - **Practice**
        - GHIDRA advanced features
- Week 7: Dynamic analysis techniques – I (**Mal**: Chapter 3)
    - **Fundamentals**
        - Limitations of static analysis techniques
        - Basic dynamic analysis techniques
    - **Practice**
        - Options for GHIDRA debugging
- Week 8: Dynamic analysis techniques – II (**Mal**: Chapters 8-10)
    - **Fundamentals**
        - Dynamic instrumentation
        - Automatic malware classification based on dynamic analysis
    - **Practice**
        - Disassembler/debugger integration in GHIDRA
        - Ollydbg
        - Kernel debugging with Windbg
- Week 9: Automated analysis techniques (Papers from the literature)
    - **Fundamentals**
        - Integrated static and dynamic analysis

        - Symbolic execution
        - Reverse engineering of network protocols
        - Midterm exam review
    - **Practice**
        - GHIDRA scripting
        - IDA features
- Week 10: Midterm exam (October 27th, 2021; it will be a take-home exam)
- Week 11: Anti-analysis techniques (**Mal**: Chapters 16, 17, and 21 (anti-dynamic analysis techniques))
    - **Fundamentals**
        - Anti-debugging techniques
        - Anti-virtual machine techniques
    - **Practice**
        - IDA scripting using IDAPython
- Week 12: Software vulnerabilities (Papers from the literature)
    - **Fundamentals**
        - Common software vulnerabilities (buffer overflow, string format, and type conversion vulnerabilities) and exploits
        - Vulnerability analysis
        - Cross-channel vulnerability
    - **Practice**
        - Vulnerability analysis using GHIDRA
- Week 13: Malware mechanisms and behavior (**Mal**: Chapters 11 and 12)
    - **Fundamentals**
        - Malware mechanisms
        - Malware behavior
    - **Practice**
        - Real-world GHIDRA and IDA plug-ins
- Week 14: Malware detection and anti-signature techniques (**Mal**: Chapters 13, 18, and 19)
    - **Fundamentals**
        - Malware detection techniques
        - Anti-signature techniques: Obfuscation and packing
    - **Practice**
        - GHIDRA extension and IDA software development kit and plug-in architecture
- Week 15: Firmware analysis (Papers from the literature)
    - **Fundamentals**
        - Firmware analysis challenges
        - Firmware analysis techniques
    - **Practice**
        - GHIDRA and IDA binary loader modules
- Week 16: Final Exam Week
        - Final exam (cumulative), Tuesday, December 7th, 2020, 08:00pm–10:00pm (it will be a take-home exam)
        - Term project and hands-on project due, December 10th, 2021 at 5:00pm

Academic Honor Code

The Florida State University Academic Honor Policy outlines the University's expectations for the integrity of students' academic work, the procedures for resolving alleged violations of those expectations, and the rights and responsibilities of students and faculty members throughout the process. Students are responsible for reading the Academic Honor Policy and for living up to their pledge to "...be honest and truthful and...[to] strive for personal and institutional integrity at Florida State University." (Florida State University Academic Honor Policy, found at http://fda.fsu.edu/academic-resources/academic-integrityand-grievances/academic-honor-policy.)

Assignments/projects/exams are to be done individually, unless specified otherwise. It is a violation of the Academic Honor Code to take credit for the work done by other people. It is also a violation to assist another person in violating the Code (See the FSU Student Handbook for penalties for violations of the Honor Code). The judgment for the violation of the Academic Honor Code will be done by the instructor and a third party member (another faculty member in the Computer Science Department not involved in this course). Once the judgment is made, the case is closed and no arguments from the involved parties will be heard. Examples of cheating behaviors include:

- Discussing the solution for a homework question.
- Copying programs for programming assignments.
- Using and submit existing programs/reports on the World Wide Web as written assignments.
- Submitting programs/reports/assignments done by a third party, including hired and contracted.
- Plagiarizing sentences/paragraphs from others without giving the appropriate references. Plagiarism is a serious intellectual crime and the consequences can be very substantial.

Penalty for violating the Academic Honor Code: A 0 grade for the particular assignment /exam and a reduction of one letter grade in the final grade for all parties involved for each occurrence. A report will be sent to the department chairman for further administrative actions.

Americans With Disabilities Act

Students with disabilities needing academic accommodation should: (1) register with and provide documentation to the Student Disability Resource Center; and (2) bring a letter to the instructor indicating the need for accommodation and what type. Please note that instructors are not allowed to provide classroom accommodation to a student until appropriate verification from the Student Disability Resource Center has been provided. This syllabus and other class materials are available in alternative format upon request.

For more information about services available to FSU students with disabilities, contact the:

Student Disability Resource Center
874 Traditions Way
108 Student Services Building
Florida State University
Tallahassee, FL 32306-4167
(850) 644-9566 (voice) (850)
644-8504 (TDD)
sdrc@admin.fsu.edu
http://www.disabilitycenter.fsu.edu/

Additional Information

Free Tutoring from FSU - On-campus tutoring and writing assistance is available for many courses at Florida State University. For more information, visit the Academic Center for Excellence (ACE) Tutoring Services' comprehensive list of on-campus tutoring options at http://ace.fsu.edu/tutoring or contact tutor@fsu.edu. High-quality tutoring is available by appointment and on a walk-in basis. These services are offered by tutors trained to encourage the highest level of individual academic success while upholding personal academic integrity.

Syllabus Change Policy: Except for changes that substantially affect implementation of the evaluation (grading) statement, this syllabus is a guide for the course and is subject to change with advance notice.