ECE 8803-A / CS 8803-SVS / ECE 4833-A: Advanced Topics in Malware Analysis

Course Overview

This course covers advanced approaches for detecting the presence of vulnerabilities in binary software, the analysis of malicious software, and explores recent research and unsolved problems in software protection and forensics.

The goal of this course is to engage in critical discussion around key research topics in software security and forensics. This course will cover: Binary Program Analysis Principles, Binary Software Security, Software Forensics and Cyber Attack Response. Students will be required to study published research papers from the top-tier academic venues in computer security and cyber forensics.

Students will conduct a half-semester long research project which will focus on solving open-ended research problems *and can lead to publications in reputed conferences/journals/workshops*. Projects aligned with a student's own graduate research (MS/PhD) will be encouraged if they have an interesting security/forensics component.

Why take this course?: You are interested in learning the fundamental principles of dissecting malware, vulnerability finding/defense, and cyber attack triage. You want to read cutting-edge research publications on these topics. *There is ample scope to publish in this area*: If the results from your course project look promising, we can write a paper on it and I will fund your travel to go present it.

Materials

There is **no** required textbook for this course. Instead we will study published research papers from the top-tier academic venues in computer security and cyber forensics.

The following books are recommended for additional background or more in-depth understanding of the topics discussed in class. Read these books *only* if you want to learn more! *They will not be covered in lectures or on exams!*

- Practical Tools/Techniques For Malware Reverse Engineering:
  Michael Sikorski, Andrew Honig. Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software. No Starch Press, 2012. ISBN: 978-1593272906
- Practical Tools/Techniques For Memory Forensics:
  Michael Hale Ligh, Andrew Case, Jamie Levy, AAron Walters. The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory. Wiley, 2014. ISBN: 978-1118825099
- Background On Low-Level Computer Systems Programming:
  Randal E. Bryant, David R. O'Hallaron. Computer Systems: A Programmer's Perspective. Pearson (3rd Edition), 2015. Online: http://csapp.cs.cmu.edu/. ISBN: 978-0134092669
- A Great Reference For The IDA Pro Projects:
  Chris Eagle. The IDA Pro Book. No Starch Press (2nd Edition), 2011. ISBN: 978-1593272890
  You may be able to find this one online.

You may also need a copy of the Intel Developer’s manuals. These are free and available via this link:
http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html
It's large, but the best PDF to get is the combined set, downloadable via the first link on that page. If you have an iPad or other tablet, drop this PDF on it and read it whenever you have spare time.

Course Outline

1. Binary Analysis Principles
    1.1. Static Analysis
        1.1.1. Static Binary Code Analysis Techniques/Tools
        1.1.2. Reverse Engineering
            1.1.2.1. Intro to Malware Classification and Triage
        1.1.3. Program Representations
        1.1.4. Pointer Analysis and Points-To
        1.1.5. Binary Code Control Flow Analysis
            1.1.5.1. Intro to Control Flow Integrity
    1.2. Dynamic Analysis
        1.2.1. Dynamic Program Tracing Techniques/Tools
        1.2.2. Program Profiling
        1.2.3. Dynamic Slicing
        1.2.4. Data Flow Tracking
            1.2.4.1. Practical Data Flow Integrity (e.g., libdtf)
    1.3. Symbolic Execution
        1.3.1. Deep Software Vulnerabilities
        1.3.2. Trigger Input Generation
        1.3.3. Automated Exploit Generation
2. Binary Software Security
    2.1. Introduction to Software Security and Access Control
    2.2. Software Vulnerabilities
        2.2.1. Static Protection through Software Bug Finding
        2.2.2. Dynamic Vulnerability Discovery
    2.3. Malware Analysis
        2.3.1. Return of Malware Classification and Triage
    2.4. Android/iOS Malware
    2.5. Input Generator for Malware Triggering
    2.6. Software Defense
        2.6.1. Dynamic Defense Mechanisms
        2.6.2. Detecting Malicious Logic in Binaries
        2.6.3. Large-Scale Software Vetting
        2.6.4. Binary Program Hardening
            2.6.4.1. Return of Control Flow Integrity
3. Software Forensics and Incident Response
    3.1. Memory Forensics
        3.1.1. Data Structure Reverse Engineering

3.1.1.1. Value-Invariant Discovery

3.1.1.2. Structural-Invariant Discovery

3.1.2. Program-Analysis-Driven Evidence Recovery

3.2. Execution Recreation

3.2.1. Postmortem Execution Analysis

3.2.2. Relationships to Debugging

Assigments & Grading

There will be 6 mini-projects during the Binary Analysis Principles portion of the class. 3 of the projects will be static analysis with IDA Pro and 3 will be dynamic analysis with Pin. Each project will require careful time allocation to complete on time (1 or 2 week deadlines). Grades will be based on the results produced by your tool. For some mini-projects, we will schedule demos during office hours.

The mini-projects will cover the following topics:

1. Intro. to Software Disassembly
2. Manual Static Malware Reverse Engineering
3. Automated Static Malware Analysis
4. Static Data Dependence Detection
5. Dynamic Control Flow Analysis
6. Dynamic Control Dependence Detection

For the remainder of the course, students will conduct a large research project (team projects are fine) which will focus on solving open problems in the areas of software security and cyber forensics. I expect that the best among these will lead to publications in reputable conferences & journals. Grading for these research projects will be based on each team's **understanding of the problem** and the success of their final prototype.

Grading Breakdown:

- 60% for the 6 mini-projects
- 30% for the large research project
- 10% for paper presentations & class participation
- In lieu of a final exam, each student will need to prepare a final presentation and report on the outcomes of their research project to be due during the final instruction days of class (Nov. 27 - Dec. 4)

Grading Scheme for ECE 8803-A & CS 8803-SVS:

- A 100% to 90%
- B 90% to 80%
- C 80% to 70%
- D 70% to 60%
- F below 60%

Grading Scheme for ECE 4833-A:

- A 100% to 85%

- B 84% to 70%
- C 69% to 55%
- D 54% to 40%
- F below 40%

Educational Objectives

- Learn and apply the fundamental principles of dissecting malware, vulnerability finding/defense, and cyber attack triage
- Become aware of limitations of existing defense mechanisms and how to avoid them
- Study cutting-edge research publications on these topics
- Engage in critical discussion around key research topics in software security and forensics
- Propose solutions to open-ended research problems and implement novel prototype solutions

Educational Outcomes

After successfully completing this course, students should be able to:

- Statically reverse engineer malware samples in a disassembler
- Build static analysis tools to automate control flow recovery and identify intractable indirect jumps
- Design and implement static analysis routines to perform automated data dependency tracking
- Instrument binary programs and malware to collect dynamic instruction traces
- Implement dynamic analysis tools to perform online control dependence tracking
- Read and present cutting-edge research publications relating to malware analysis, vulnerability finding/defense, and cyber attack triage

Honor Code

Students are expected to abide by the Georgia Tech Academic Honor Code. Honest and ethical behavior is expected at all times. All incidents of suspected dishonesty will be reported to and handled by the Office of Student Integrity. You will have to do all assignments individually unless explicitly told otherwise. You may discuss with classmates but you may not copy any solution (or any part of a solution).

Learning Accommodations

Whenever needed, the instructor will make accommodations for students with documented disabilities. These accommodations must be arranged in advance and in accordance with the Office of Disability Services.

Class Attendance

Class attendance is mandatory, and past offerings have shown that students who are actively involved in class discussions have the best experience conquering this challenging subject matter. Course deadlines and assignments can be modified for students with documented absences. These accommodations must be arranged in advance and in accordance with the Georgia Tech Attendance Policy.