Syllabus - ECE 4833/CS 4xxx/ECE 8803/CS 8803 - Advanced Topics in Malware Analysis

Course Summary:
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.

Grading:
Mini Projects (6): 60%
Semester Project: 30%
Paper Presentation & Class Participation: 10%
Total: 100%

Grading Scale (Graduate Students):
90% - 100% A
80% - 89% B
70% - 79% C
60% - 69% D
< 60% F

Grading Scale (Undergraduate Students):
85% - 100% A
70% - 84% B
55% - 69% C
40% - 54% D
< 40% F

Topics Outline:
1. Intro to the Course (Binary Program Analysis, Security, and Forensics)
2. Binary Analysis Principles
    2.1. Static Analysis
        2.1.1. Static Binary Code Analysis Techniques/Tools
        2.1.2. Reverse Engineering
            2.1.2.1. Intro to Malware Classification and Triage
        2.1.3. Program Representations
        2.1.4. Pointer Analysis and Points-To
        2.1.5. Binary Code Control Flow Analysis
            2.1.5.1. Intro to Control Flow Integrity
    2.2. Dynamic Analysis
        2.2.1. Dynamic Program Tracing Techniques/Tools
        2.2.2. Program Profiling
        2.2.3. Dynamic Slicing
        2.2.4. Data Flow Tracking

2.2.4.1. Practical Data Flow Integrity (e.g., libdtf)
2.3. Symbolic Execution
2.3.1. Deep Software Vulnerabilities
2.3.2. Trigger Input Generation
2.3.3. Automated Exploit Generation
3. Binary Software Security
3.1. Introduction to Software Security and Access Control
3.2. Software Vulnerabilities
3.2.1. Static Protection through Software Bug Finding
3.2.2. Dynamic Vulnerability Discovery
3.3. Malware Analysis
3.3.1. Return of Malware Classification and Triage
3.4. Android/iOS Malware
3.5. Input Generator for Malware Triggering
3.6. Software Defense
3.6.1. Dynamic Defense Mechanisms
3.6.2. Detecting Malicious Logic in Binaries
3.6.3. Large-Scale Software Vetting
3.6.4. Binary Program Hardening
3.6.4.1. Return of Control Flow Integrity
4. Software Forensics and Incident Response
4.1. Memory Forensics
4.1.1. Data Structure Reverse Engineering
4.1.1.1. Value-Invariant Discovery
4.1.1.2. Structural-Invariant Discovery
4.1.2. Program-Analysis-Driven Evidence Recovery
4.2. Execution Recreation
4.2.1. Postmortem Execution Analysis
4.2.2. Relationships to Debugging

The mini project assignments will cover the following topics:
1) Software Disassembly
2) Static Malware Reverse Engineering
3) Automated Static Malware Analysis
4) Static Data Dependence Detection
5) Dynamic Control Flow Analysis
6) Dynamic Control Dependence Detection

Accommodations for undergraduates in this course:
The required prerequisite for undergraduates seeking to enroll in this course is Dr. Saltaformaggio's ECE 4894 A - Intro. To Malware Reverse Engineering. ECE 4894 exposes students to an immersive, hands-on experience with practical techniques for static malware analysis. This foundation puts them at an advantage to graduate students who will have to learn these techniques anew in the first weeks of this course.

The learning objectives for undergraduate teams' semester projects will also differ from graduate student teams. The focus for undergraduates will be on practical extensions to existing tools, rather than novel research developments. An example of such a project would be to create a plugin for the IDA Pro tool (used in mini projects 1 – 4) which overcomes a known limitation of or provides an enhancement to IDA Pro. Successful projects of this nature may even be submitted to the annual IDA Pro plugin contest for a cash prize and international recognition.

Additionally, mini projects 4 and 6 (the most complex of the course) are constructed from clearly delineated components, the most advanced of which can be marked as "extra credit" for undergraduate teams. Lastly, undergraduates will have a more lenient grading scale (shown above).