CUNY GC CSc 81300
Special Topics: Software Security

Prof. Sven Dietrich

Fall 2020
Tuesdays: 11:45am-1:45pm
Location: Virtual
Textbook: none (research papers only)

Rationale

Software security is a challenging subject: making systems work fast, reliably, and securely often leads to one of these aspects being sacrificed. Most likely that sacrifice will be security. This course gives a thorough grounding in the methodologies, technologies, and algorithms currently needed by people who do research in software security, secure software engineering, and malware.

Description

Software security is an advanced course for individuals interested in the theory and practice of software security. This course will study approaches, mechanisms, and tools used to make software systems more secure against attacks, abuse, andmalware. Instead of a textbook, reading will be a survey of recent and seminal research papers, including from top-tier security and software engineering conferences. We will motivate the study by discussing common software security dangers, software and malware analysis tools.

Topic list

The course may cover topics related to two main modules:

- Architectural approaches to building secure software and systems;
- Software analysis for finding software flaws as well as analyzing malicious code

Topics may include but are not limited to:

- Building Secure Systems
  - Confinement
  - Virtual Machines
  - Trusted Computing and Attestation

  - Control Flow Integrity

- Software Security & Analysis
  - Formal methods (e.g. model checking)
  - Static analysis and testing
  - Dynamic analysis
  - Taint analysis
  - Code cloning detection (binary and source code)
  - Control flow graph extraction, data dependency
  - Buffer overflow attacks
  - Control-hijacking
  - Vulnerability discovery (e.g. fuzzing)
  - Decompilation/Disassembly approaches and challenges
  - Vulnerability identification propagation
  - Compiler provenance
  - Implications for open-source software

Learning Objectives

- Understand the state of the art in control-hijacking and associated defenses in software systems.
- Understand the specific security architectures for system isolation and analysis.
- Understand the strengths and limitations of various methods of software analysis, and their application of vulnerability discovery and verification of security properties of software, as also applied to malware analysis.

Assessment

Students will be evaluated based on small projects (30%), one to two research papers (60%), and class participation (10%). As this is a systems-oriented course, the projects will provide students with practical experience with the tools and mechanisms studied in class. Students will work on the projects either individually or in groups, and the projects will be evenly spaced over the course of the semester.

The students are expected to have basic operating system (Unix, Windows, Mac) and computer security knowledge, as well as familiarity with C/C++ and/or python.

Class notes

We will be using Blackboard as a Learning Management System. Announcements, assignments, and class notes will be located there.