CS-510

Topics in software security

Payer Mathias

Cursus	Sem.	Type
Computer science	MA1, MA3	Opt.
Cyber security minor	H	Opt.
Cybersecurity	MA1, MA3	Opt.
SC master EPFL	MA1, MA3	Opt.

Language of teaching	English
Credits	3
Session	Winter
Semester	Fall
Exam	During the semester
Workload	90h
Weeks	14
Hours	**2 weekly**
Courses	1 weekly
Exercises	1 weekly
Number of positions	

Summary

Memory corruption and type safety flaws dominate the threat landscape. We will approach current research from three dimensions: sanitization (finding flaws through runtime monitors); fuzzing (testing software automatically); and mitigation (protecting software at runtime).

Content

Unsafe languages like C/C++ are widely used for their great promise of performance. Unfortunately, these languages are prone to a large set of different types of memory and type errors that allow the exploitation of several attack vectors such as code reuse, privilege escalation, or information leaks.
On a high level memory and type safety (and type safety) would solve all these problems. Safe languages can (somewhat) cheaply enforce these properties.
Unfortunately, these guarantees come at a high cost if retrofitted onto existing languages.
When working with unsafe languages, three fundamental approaches exist to protect against software flaws: formal verification (proving the absence of bugs), software testing (finding bugs), and mitigation (protecting against the exploitation of bugs). In this seminar, we will primarily focus on the latter two approaches. Formal verification, while giving strong guarantees, struggles to scale to large software.
This seminar explores three areas: the understanding of attack vectors,approaches to software testing, and mitigation strategies. First you need to understand what kind of software flaws exist in low level software and how those flaws can be exploited.

Learning Prerequisites

Required courses

A security course like COM-301
An operating/systems course like CS-323

Recommended courses

COM-402 Information security and privacy
CS-412 Software security

Learning Outcomes

By the end of the course, the student must be able to:

- Investigate select advanced concepts in software security
- Promote their programming and systems skills in core security topics
- Assess / Evaluate the contributions of a software security research paper

- Investigate software security research papers
- Present a research paper and lead the resulting discussion

Teaching methods

In this seminar course, students will read, prepare, and present recent research papers in the filed of software security. The papers will be discussed in class. The presenter will organize the discussion among their peers and prepare a set of discussion points.

Expected student activities

The students are expected to

- Prepare and hold the presentation of their assigned research paper
- Summarize the paper along with the class discussion after their presentation
- Participate in the presentations and discussions of the other students

Assessment methods

- Presentation : 40%
- Summary/review : 50%
- Class participation : 10%

Resources

Websites

- https://go.epfl.ch/cs510

Moodle Link

- https://go.epfl.ch/CS-510