Introduction
TDDC90 – Software Security

Ulf Kargén

Division for Database and Information Techniques (ADIT) at the Department of Computer and Information Science (IDA)
Agenda

- Why study software security?
- Organization of the course
  - Course contents
  - Prerequisites
  - Lectures overview
  - Labs
  - Reading material
  - Course evaluation
  - Notes about distance mode

Examiner
Ulf Kargén

Lecturer
Ahmed Rezine

Lecturer
Kristian Sandahl

Lab assistant
Alireza Mohammadinodooshan
Why study software security?

1. What kind of software is security critical?
2. Why do people try to hack software?

**15 years ago**

1) Mostly server software
2) Fun

**Today**

1) Pretty much all software
2) Profit

- Hackers increasingly target end-user equipment
- Break-ins happen increasingly by exploiting client-side software (document viewers, media players, browsers), not by attacking central infrastructure
- “Everything” is connected to the internet
Why study software security?

Software is everywhere …
Developing secure software requires...

- Security-aware developers
  - Know about common vulnerability types
  - Know common attacks
  - “Think like a hacker”
  - The devil is in the details...

- Adequate software engineering processes
  - Methods for eliciting security requirements
  - Security in the specification, architecture and design
  - Secure coding guidelines and patterns

- Software security assurance methods and tools
  - Many methods:
    Code reviews, formal methods, static analysis, fuzzing, etc.
Organization of the course
Organization

- 9 lectures
- 3 mandatory labs
  - Pong – the insecure ping
  - Static analysis
  - Web security
- Examination:
  - Written exam (3 hp)
  - Labs (3 hp)
Prerequisites

- **Required:**
  - Basic computer security course
  - Programming experience
  - Course in software engineering

- **Recommended:**
  - Operating systems and assembly programming basics
  - Some prior experience with C-programming
  - Basic course in logic
  - Basic web programming
    (HTML, JavaScript, some server-side language)
For those unfamiliar with C

Google these things (in this order):

✓ C pointers
✓ Pointer arithmetic
✓ Pointers and arrays
✓ C dynamic memory allocation
✓ C sizeof operator
  ▪ Pay special attention to the difference between sizeof on pointers and arrays!
Lectures

- Secure software development (1 lecture)
  Given by Ulf Kargén
  - Secure software development processes
  - Secure design patterns
  - Modeling and risk analysis

- Vulnerabilities and exploits (2 lectures)
  Given by Ulf Kargén
  - Common vulnerabilities in C/C++ programs
  - Known attack techniques
  - OS and compiler mitigations
Lectures (continued)

• Web security (1 lecture)
  Given by Ulf Kargén
  • Common vulnerabilities in web applications
  • Attack techniques and protections

• Static analysis (2 lectures)
  Given by Ahmed Rezine
  • Introduction to static analysis
    • Abstract interpretation
    • Symbolic execution
Lectures (continued)

- Code reviews (1 lecture)
  Given by Kristian Sandahl
    - Software inspections and other techniques

- Security testing and course wrap-up (1 lecture)
  Given by Ulf Kargén
    - Fuzzing, concolic testing
    - Course wrap-up
Labs

• **Pong – the insecure ping**
  • Perform a code review to find vulnerabilities
  • Exploit a buffer overflow to gain root
  • Fix all vulnerabilities
  • **Requires considerable time and effort, especially if you don’t posses all recommended prerequisite knowledge**

• **Websec**
  • Deliberately vulnerable web app
  • Study common weaknesses and understand attack techniques
  • Typical time needed: 1-2 lab sessions

• **Static**
  • Study common static analysis techniques described in the lectures
  • Typical time needed: ~1 lab session
Labs

• Different assistants for some labs – see lab page on course web

• Webreg signup deadline **November 11**\(^{th}\)
  • Unregistered students not allowed to sign up!

• Students are required to work in pairs
  • If you sign up alone, we may randomly group you with another student. (We may be a little more lenient in this regard due to the current coronavirus situation.)

• **Hard** deadline for handing in solutions is **December 18**\(^{th}\)
  • Complete all labs **at least one week before this** to allow time for corrections and re-submission
  • Hand in solutions continuously during the study period – don’t save everything for the last week!
  • Start with labs **as early as possible**, especially Pong!
Reading material

• No course book (no one book covers all topics in the course)
• Mandatory reading:
  • Papers/articles, web resources, and lecture slides
  • Lectures don’t cover all articles, and vice versa
• Also a list of extra reading for interested students
  • Not needed for exam
Previous year’s course evaluation

• Overall score last year was 4.5 (of 5)
• Scores of all evaluation items available at:
  https://admin.evaluate.liu.se/search?lang=en

Comments from students:

• Some found the static analysis lectures hard to follow, or wanted more examples in the slides
  • **Action**: We plan to overhaul the lectures and lab. Unfortunately, insufficient time for making large changes this year due to Covid-19. Will happen 2021.

• Some found the code review lecture to be redundant.
  • **Comment**: Quite heterogenous intake to course – some have already taken software engineering courses (e.g., given by Kristian Sandahl) that covers similar material, while others have not.
  • **Action**: No action as of now.
Distance-mode changes

- Lectures given via Zoom
  - Need to authenticate with LiU-ID to join
- Lab supervision also via Zoom.
  - Exact procedure will be communicated soon
- All labs are being prepared/modified for distance mode
  - Still work-in-progress – we expect to have the Pong lab up within a few days, Websec lab will follow.
  - Static analysis lab will be published after Ahmed’s lectures
Questions?