

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

Examiner

Ulf Kargén



Lecturer

Ahmed Rezine



Lecturer

Kristian Sandahl



Why study software security?

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

20 years ago

- 1) Mostly server software
- 2) Fun

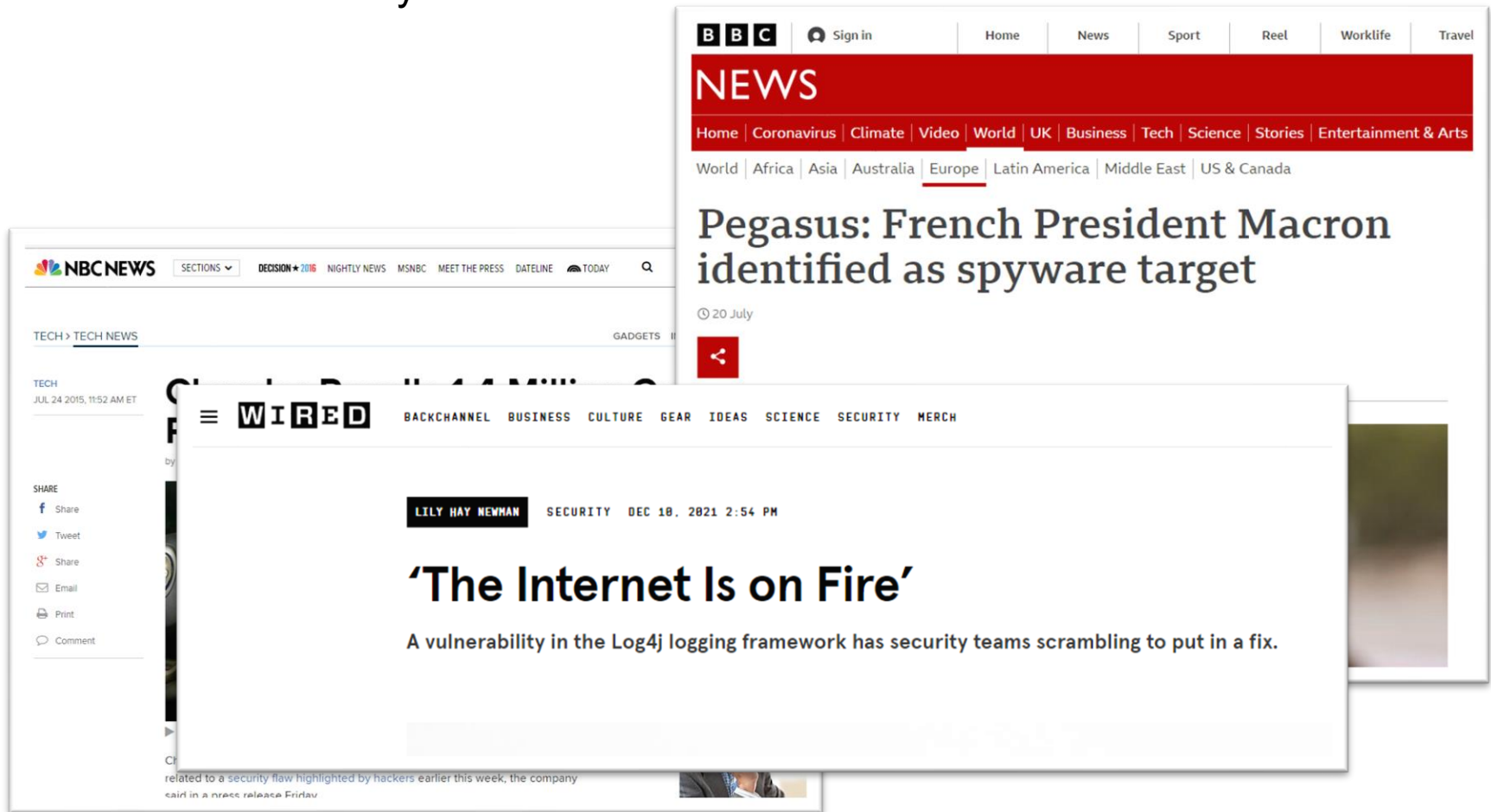
Today

- 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 – attacks against poorly secured IoT targets are very common

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
 - Web security
 - Static analysis
- Examination:
 - Written exam (3 hp)
 - Labs (3 hp)

Detailed information on course organization, lecture slides, lab instructions, etc., is available on the course web site:

<https://www.ida.liu.se/~TDDC90/index.en.shtml>

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

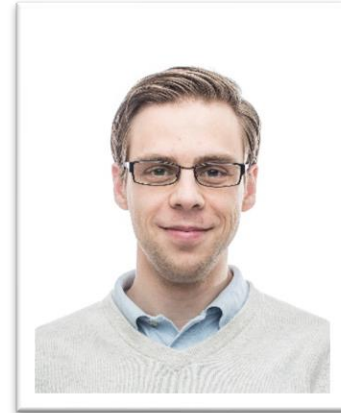
Google these things (in this order):

- ✓ 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



- **Memory safety vulnerabilities (2 lectures)**
Given by Ulf Kargén

- Common vulnerabilities in C/C++ programs
- Known attack techniques
- OS and compiler mitigations



Lectures (continued)

- **Static analysis (2 lectures)**
Given by Ahmed Rezine

- Introduction to static analysis
 - Abstract interpretation
 - Symbolic execution



- **Code reviews (1 lecture)**
Given by Kristian Sandahl

- Software inspections and other techniques



Lectures (continued)

- Web security (1 lecture)

Given by Ulf Kargén

- Common vulnerabilities in web applications
- Attack techniques and protections



- 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 possess all recommended prerequisite knowledge**
- Static
 - Study common static analysis techniques described in the lectures
 - Typical time needed: 1-2 lab sessions
 - **Note 1:** Requires demoing for Ahmed or Ulf. The other labs do **not** require demos.
 - **Note 2:** Ahmed is only available for demos during **two** specific lab sessions. Check the lab schedule on the web!
- Websec
 - Deliberately vulnerable web app
 - Study common weaknesses and understand attack techniques
 - Typical time needed: 1-2 lab sessions

Labs

- Different assistants for some labs – see lab page on course web
- Webreg signup deadline **November 12th**
 - Unregistered students not allowed to sign up!
- Labs are meant to be done in pairs
 - *Might* be possible to do labs alone if you have a good motivation, **however:**
 - If too many sign up alone, we may randomly group lone students.
- **Hard** deadline for handing in solutions is **December 17th**
 - 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!
 - OK (and recommended) to hand in individual parts of Pong separately, but please **use the same email thread for all parts**
 - 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>

Suggestions on improvements from students:

- “The static analysis lab was a bit underwhelming. Maybe focus a bit more in one area rather than a little bit of several areas”
 - **Comment:** Static analysis is a complex topic, which would require its own course to cover fully (see TDDE34). The purpose of the lectures and the labs is to get the “gist” of the methods, without going into much detail.

Previous year's course evaluation

Positive remarks:

- “I liked the labs a lot :)”
- “The pong lab was really fun and I learned a lot.”
- “Fun labs!”
- “The slides and lectures where good even if they where on zoom”

Questions?