0. Overview

Prof. Dr. Marc Rennhard
Institut für angewandte Informationstechnologie InIT
ZHAW School of Engineering
rema@zhaw.ch

Software Security (SSI)

- Lecturer
  - Prof. Dr. Marc Rennhard, rema@zhaw.ch, office TG 210, 058 934 7245

  - I'm leading the Informa[illegible]e Institute of applied information technology [illegible]haw.ch)
  - If you are interested in doing further work in information security, don't hesitate to contact me
    - E.g. project or bachelor theses, MSE positions, research positions ..

Sabbatical

Software Security (SSI)

- Lecturers
  - Dr. Bernhard Tellenbach (lecturer)
    - tebe@zhaw.ch, TG 205
  - Deputy lead Information Security group at the InIT

  - Dr. Stephan Neuhaus (lecturer)
    - tebe@zhaw.ch, TG 203

  - If you are interested in doing further work in information security, don't hesitate to contact us
    - E.g. project or bachelor theses, MSE positions, research positions...

- Teaching platform: OLAT, http://olat.zhaw.ch
  - Primary source of information, timetable, material

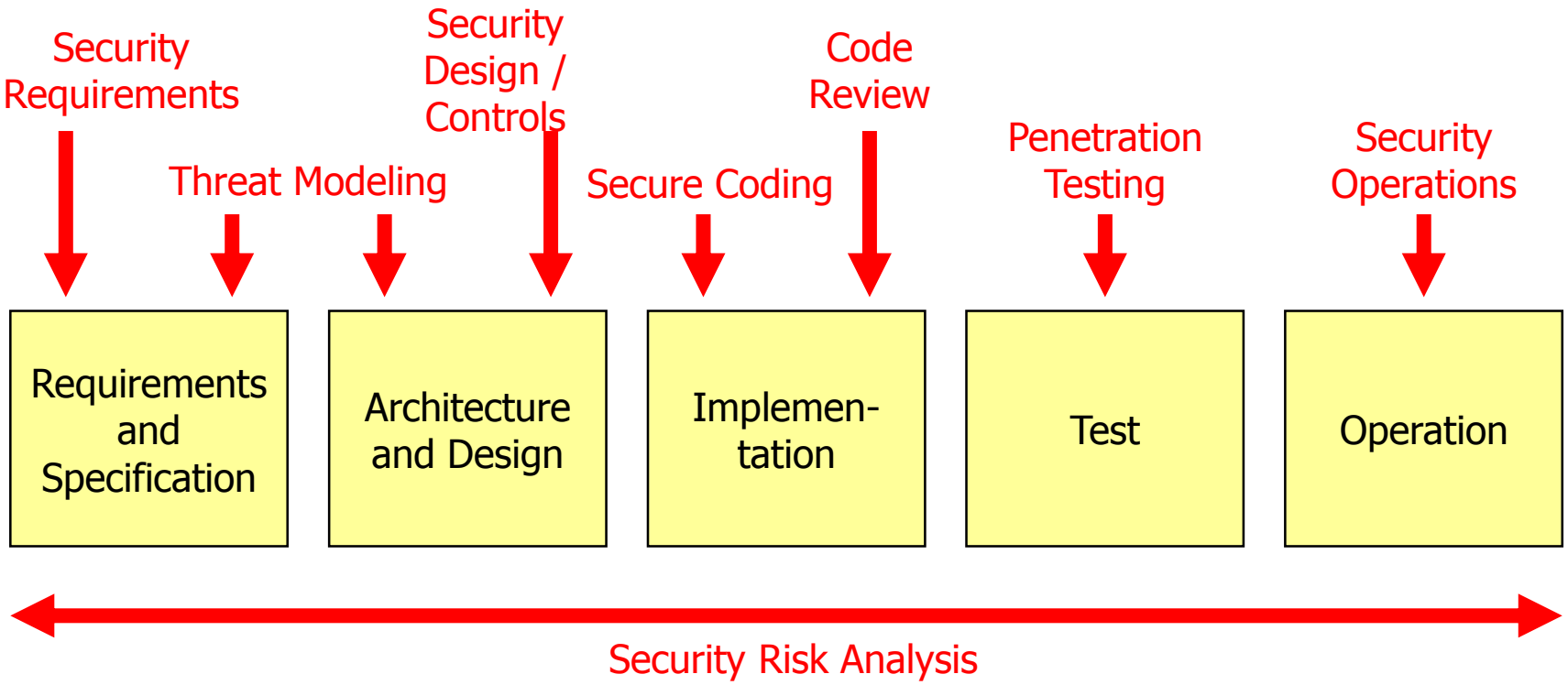
Goals

The students get a profound introduction to software security. The focus lies on "secure software development process", "security testing of software and systems", and "secure software development with Java". In particular, the students will learn the following competencies:

- You know and understand what must be considered during secure software development.
- You can apply the principles of secure software development to an arbitrary software development process to turn it into a secure software development process.
- Using appropriate methods and tools, you can test applications and systems with respect to security and exploit uncovered vulnerabilities.
- You know typical, security-critical programming errors that are often made and know how you can prevent them in your own programs.
- You are capable of developing secure Java applications (with a focus on Java EE web applications) by appropriately using the security features provided by Java and security libraries.

Course Overview

The course will discuss the various activities that should be taken into account during a secure software development lifecycle



Lecture Topics (1)

1. Introduction to Software Security
   - Motivation, examples, terminology
2. Introduction to SDL
   - Activities during a secure software development lifecycle
3. Typical Programming Errors
   - Overview of coding errors and discussion of some typical examples (buffer overflows, dangerous functions, race conditions)
4. Java Security
   - Components of the Java library to implement cryptographic operations and secure communication in Java programs (JCA, JSSE)
5. Security Controls
   - A summary of various security controls that are important for secure software development, mostly repetition from course ISI

Lecture Topics (2)

6. Secure Design Principles
   - Fundamental security principles you should always keep in mind
7. Finding and Exploiting Vulnerabilities in Web Applications
   - Security testing of web applications
8. Java Web Application Security
   - How to develop secure Servlet/JSP-based Java EE web applications
9. Security Requirements Engineering and Threat Modeling
   - Methods to define the right security requirements and to uncover flaws in a security architecture / design
10. Security Risk Analysis
    - Methods to rate the risk (severity) of vulnerabilities
11. Penetration Testing
    - Activities performed during a complete penetration test (footprinting, scanning etc.)

Lab Topics (1)

1. Analysis of the MELANI semi annual report
   - Get an overview of current cyberattacks and analyse and rate them
1. Simple Web Server
   - Experiment with and fix a simple but heavily flawed web server program to see how simple bugs can have security implications
1. Buffer Overflow Attacks
   - Find and exploit buffer-overflow vulnerabilities in a simple program
1. Cryptography in Java
   - Develop a program to integrity-protect and encrypt files
1. TLS in Java
   - Develop a program that can test any TLS server to make statements about supported cipher suites and used certificates

Lab Topics (2)

6. Finding and Exploiting Vulnerabilities in a Web Applications
   - Using an e-shop application developed by security-unaware students

7. Security Testing Tools
   - Experiment with static code analysis tools and a vulnerability scanner to lean about the possibilities and limitations of automated testing tools

8. Java Web Application Security
   - Extend a Java EE application discussed in the lecture with additional functions and implement the right security measures

9. Security Requirements Engineering and Threat Modeling
   - Analyse a given scenario for vulnerabilities and propose appropriate security requirements