UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Software Security and Testing

2122-3-E3101Q123

Aims

This course aims to provide knowledge of the problems of software security and software validation, and competencies on techniques to address these problems. In particular, by the end of this course, the perspective students will be capable of analyzing risks and critical points of an information system, administering tools for defending information systems, understanding the phases of an attack, and analyzing and designing solutions in application contexts that require software validation.

Contents

The problem of information security. Heterogeneity of expertise required in the field of security. Attack and defense roles in information systems. The problem of software validation. Methods of software testing: functional testing, structural testing, model-based testing. Infrastructures for test execution. Static and dynamic analysis of programs.

Detailed program

1 "Risks when using an information system, and related roles and expertise"

2 "Techniques and protocols for security

- Criptography, implementation errors and attacks

- Security in operating systems and networks"

3 "Secure programming

- Security errors in applications

- Analysis of known programs that exhibit vulnerabilities"

4 Malicious software: troians, back-doors, locgical bombs, viruses, worms

5 Defenses: Intrusion Detection System, verification attacks, firewall

6 Software validation: dimensions of the problem.

7 " Testing:

- Functional testing: sources of information to design tests, functional (black-box) testing, pros and cons in comparison to the random approach

- Combinatorial testing: category partition, pairwise combinations.

- Structural testing: coverage of statements, branches, conditions and paths.

- Model based testing: finite state machines, decision structures and flow graphs.

- Test infrastructures: driver, stub and oracles"

8 "Program analysis:

- The difference between static and dynamic analysis

- Dynamic analysis: memory analysis and lockset analysis

- Static analysis: techniques based on data-flow models, symbolic analysis"

Prerequisites

No essential prerequisite. It can be useful to master the base concepts from the following courses: Fondamenti dell'informatica, Programmazione 1, Programmazione 2, Reti e Sistemi Operativi, Analisi e Progettazione del Software

Teaching form

Lessons in class. Laboratory work in class. Lessons will be given in Italian.

During the Covid-19 emergency the course will be given in blended mode: partially in class, partially with asynchronous video-lessons, partially with synchronous teleconferences.

Textbook and teaching resource

Textbooks:

Ross Anderson. Security Engineering. 2 ed, Wiley 2008

Mauro Pezzè, Michal Young. Software Testing and Analysis: Process, Principles and Techniques. John Wiley, 2008

Semester

Second semester

Assessment method

The assessment method consists of a written exam, and possibly of an oral exam.

The written exam consists of exercises that require calculations, exercises that require to solve an assigned problem, questions on the presented notions, and questions that require reasoning and deductions. All questions are open questions. The exercises aim to control the problem solving competencies acquired as a result of the course, and the open questions allow for in-depth control of the theoretical topics and on the capability of the students to autonomously and critically reflect on the critical points of the course programme. Usually the written exam includes 3 exercises and 5 open questions. The exercises are evaluates based on the correctness of the proposed solution, and the open questions based on the completeness and precision of the corresponding answers.

The oral exam is options and, if requested by a student, consists on a discussion on all points of the course programme. The result of the oral exam complements, either positively or negatively, the grade that derives from the evaluation of the written exam.

Furthermore the assessment can be accomplished with two separate partial exams, which are both written exams and evaluated with in the same way as the entire written exam explained above. Each partial exam addresses half of the course programme. The first partial exam focuses and the programme on software validation, test and analysis. The second partial exam focuses on the programme on software security. The final evaluation is the average of the evaluations of the partial exams, which must necessarily be both sufficient.

Office hours

By appointment