GÖTEBORGS UNIVERSITET

The Board of the IT Faculty

DIT101, Language-based Security, 7.5 higher education credits

Second Cycle/A1F

This syllabus is the binding document.

1. Confirmation

The Board of the IT Faculty established the course plan at 2006-11-17. It has been revised 2009-09-18 to be valid from spring term 2010.

Field of education: Science

Department: Computer Science and Engineering

2. Position in the educational system

The course is a part of the Computer Science Master's programme and an elective course at the University of Gothenburg.

The level for the course in relation to degree requirements is Master´s degree, code A1F. The course has course/courses at second cycle level as entry requirements.

3. General prerequisites

The requirement for the course is to have successfully completed a first year studies within the subject Computer Science or equivalent. Knowledge of the material covered in the courses DIT230 Programming Languages and DIT641 Computer Security is also required. Previous knowledge of semantics, automata, and compiler construction is helpful (although not required as a prerequisite).

4. Course content

This course combines practical and cutting-edge research material. For the latter part, the course's particular emphasis is on the use of formal, or semantic, models of program behaviour for specifying and enforcing security properties. The course consists of lectures, group meetings and project presentations.

5. Learning outcomes

Modern attacks often succeed at circumventing standard security mechanisms. While operating-system security policies are low-level (such as access control policies, protecting particular files), many attacks are high-level, or application-level (such as email worms that pass by access controls pretending to be executed on behalf of a mailer application). Because applications are typically specified and implemented in programming languages, application-level security is a part of the more general area of language-based security. A direct benefit of language-based security is the ability to naturally express security policies and enforcement mechanisms using the techniques of the well-developed area of programming languages.

After the course the student should be able to apply practical knowledge of security for modern programming languages. This includes the ability to identify application- and language-level security threats, design and argue for application- and language-level security policies, and design and argue for the security, clarity, usability, and efficiency of solutions, as well as implement such solutions in expressive programming languages.

The student should also be able to demonstrate the critical knowledge of:

-principles behind application-level attacks (such as Trojan horses, worms, buffer overrun attacks, exploit attacks, covert channels, and malicious code) and

-language-based protection mechanisms (such as static security analysis, program transformation, and stack inspection).

6. Required reading

See separate literature list

7. Assessment

The course is examined by laborations and a written report.

A student who has failed a test twice has the right to change examiner, unless weighty argument can be adduced. The application shall be sent to the department and has to be in writing.

8. Grading scale

The course is graded with the following marks: Fail (U), Pass (G), Pass with Distinction (VG).

9. Course evaluation

The course is evaluated through meetings both during and after the course between teachers and student representatives. Further, an anonymous questionnaire can be used to ensure written information. The outcome of the evaluations serves to improve the course by indicating which parts could be added, improved, changed or removed.

10. Additional information

The course is held in English.