Didactic Planning

1 Course Description

Title name:	Secure Software Programming
ECTS:	3 ECTS
Lessons:	24 hours, divided into theory lessons and laboratory lessons
Instructor name:	Ricardo J. Rodríguez (visiting professor at DFM)
Teaching period:	First semester (in October)

Summary

This course covers the basic concepts to teach students about how they must develop software in a *secure way*. In this regard, we will first recap some knowledge about C programming language, since the course is mainly oriented to work with it. Then we will show the main memory error vulnerabilities that software may suffer, such as buffer overflows, integer overflows, or format string, among others. Standard guidelines for secure programming provided by national organizations as OWASP, CWE, and SEI CERT (from Carnegie Mellon University) will also be introduced in the course. During laboratory lessons, the students will learn to work with debugging tools, to identify vulnerabilities in source code, and to fix potential bugs that may lead to software errors. Furthermore, the course will also show how to use automated tools to audit the source code.

2 Course Schedule (tentative)

Session 1. (Oct 5, 2 hours) *Introduction to C programming*

Session 2. (Oct 8, 2 hours) *Basics on Computer Architecture*

Session 3. (Oct 9, 3 hours) Laboratory: *Programming. Use of a debugger*

Session 4. (Oct 16, 2 hours) *Common memory errors I*

Session 5. (Oct 18, 2 hours) *Common memory errors II*

Session 6. (Oct 19, 3 hours) Laboratory: *Vulnerability analysis and exploiting*

Session 7. (Oct 23, 2 hours) *Guidelines for secure programming I*

Session 8. (Oct 25, 2 hours) *Guidelines for secure programming II*

Session 9. (Oct 26, 2 hours) *Source code auditing*

Session 10. (Oct 29, 3 hours) Laboratory: *Automated tools for source code auditing*

Session 11. (Oct 30, 2 hours) **Examination**