II.3524 – Software security

General information

Module Title: Software Security
Module ID: II.3524
Module leader: Saad EL JAOUHARI
ECTS: 5 credits
Average amount of work per student: from 100 to 150 hours, 42 of which are supervised
Teamwork: yes
Keywords: Software security vulnerabilities, OWASP, vulnerability testing, security by design, change management, low-level vulnerabilities, DevSecOps

Presentation

Designing and developing software, a website or a mobile application without considering the safety aspect is like designing a car without a bumper, windshield or door closing system.

This module aims to raise awareness and teach security best practices in software development in general and web and mobile application development in particular. It is based on the recommendations of the *ANSSI* (French National Agency for the Security of Information Systems) and on the findings of OWASP on the 10 most widespread software security flaws (*https://owasp.org/www-project-top-ten/*).

In this module, students will be able to participate in security-focused coding workshops that will allow them, among other things, to anticipate security flaws during the design and implementation stages of software, to detect security flaws, to secure existing code or to overcome an intrusion problem.

Educational objectives

- Understand security requirements throughout the software lifecycle
- Detect vulnerabilities in in-use applications
- Manage changes: update/upgrade of existing software and/or replacement of one system with another

Prerequisite

- Good knowledge of software development methods (V-shaped, agile, etc.)
- Be comfortable in software development: programming
- Linux Basics

Content/Program

Concepts

- Confidentiality, Authenticity, Integrity, Availability, Traceability, Non-Repudiation
- Authentication and access control
- Identity and access management
- Security in the Software Development Lifecycle (SDLC)
- Security by design
- Web vulnerabilities
- Exploitation of low-level vulnerabilities *(buffer overflow, string format, ...)*
- Static and dynamic application testing
- Malicious code and attacks on applications: *viruses, Trojans, worms, logic bombs, backdoors, etc.*
- DevOps and DevSecOps

Tools used

- Kali linux
- Detecting Web Application Security Vulnerabilities with OWASP Zed Attack Proxy (ZAP)

Pedagogical methods

Learning methods

- Presentation of the fundamentals in class with exercises and case studies.
- Implementation in the form of practical work initially, then in the form of a mini project.
- The practical work and the project will be carried out in groups of two.

Evaluation methods

- Individual: Tabletop examination (40%)
- Collective: Submission of practical work (30%) and a mini-project (30%)

Language of work

- Module delivered entirely in English
- Students' productions in French or English of your choice

Bibliography, Webography, Other sources

- https://www.ssi.gouv.fr/guide/regles-de-programmation-pour-le-developpement-securise-de-logiciels-en-langage-c/
- https://www.ssi.gouv.fr/agence/publication/securite-et-langage-java/
- https://www.ssi.gouv.fr/particulier/logiciels-preconises-par-lanssi-2/
- Mike Chapple, James Michael, Darill Gibson. (ISC)2 CISSP Certified Information Systems Security Professional Official Study Guide, 8th Edition. Chapter 21. Malicious Code and Application Attacks. Copyright © 2018 by John Wiley & Sons, Inc., Indianapolis, Indiana. ISBN: 978-1-119-47593-4