
Course: Application Security – laboratories (GENERAL INFORMATION)

PREREQUISITES:

Knowledge of computer networks, operating systems, cryptography. Knowledge of structured and object-oriented programming and basic knowledge of database design.

GOALS:

- The aim of the labs is to familiarize students with the design, development and maintenance of secure web/mobile/desktop applications.
- Design and implementation own secure web/mobile/desktop application. Development of system design documentation containing: functional and non-functional requirements of the application, UML diagrams, database schema, OWASP security audit (self-tests). Taking into account the latest technologies and trends in the design.
- Application should have public side (incl. functions restricted for unregistered users) and managements side (admin dashboard) accessible only for trusted users.
- Final web application should be deployed in public domain or virtual environment (Docker / VritualBox etc.)

EXAMPLE PROJECT TOPICS (WEB APPS):

- “*themactic blog*”,
- “*meme website*”,
- “*simple e-commerce solution*”,
- “*company website with simple CMS*”
- “...” - the team's own proposal.

¹ <https://www.michal.apolinarski.pracownik.put.poznan.pl> | e-mail: michal.apolinarski[at]put.poznan.pl

PLAN:

- Organizational classes - discussion of the form of the course and conditions of passing the course. Creation of groups of 2 persons.
- **topic #1: Websites-recon [180 min.]**
- **topic #2: Open-source-web-application-basic-analysis [180 min.]**
- **topic #3: User-registration-and-login-process [360 min.]**
- **topic #4: Forgotten-password-feature [180 min.]**
- **topic #5: Web-app-development [560 min.]**
- **topic #6: Security-audit-#1-black-box-approach [180 min.]**
- **topic #7: Security-audit-#2-white-box-approach [180 min.]**

PASS REQUIREMENTS:

- Active participation in classes – completing all the tasks confirmed by reports.
- Systematic work on the projects.
- Positive mark for a project completed as specified and required.

CONTENT OF PROJECT DOCUMENTATION (minimum requirements):

- Title page - including details of the subject and authors.
- Table of contents.
- General characteristics of the project.
- Requirements (incl. actors).
 - Functional.
 - Non-functional.
- System architecture, tools (including team support tools), environment, technologies, possible libraries and frameworks, etc.
- UML diagrams (min. 1 of each type):
 - Use case diagram.
 - Sequence diagram.
 - ~~State diagram.~~
 - Class diagram.
- Database schema:

- ~~Entity relationship model.~~
- Relational model.
- Graphical interface designs.
- The most important methods and code fragments of the application.
- Security self-analysis.
- Summary:
 - Division of work.
 - Objectives met, objectives not met, problems encountered.
 - Development perspective.

GENERAL NOTES:

- The project documentation should be neatly edited i.e. include a title page, table of contents, division into chapters and subchapters, drawings and tables should have captions.
- For a front-end, students can use free templates for the public site and admin dashboard e.g. <https://colorlib.com/wp/free-html5-admin-dashboard-templates/>
- More ambitious, complex and elaborate projects will be evaluated higher than projects containing the necessary minimum.
- **Students are required to send a lab report or an updated version of the project documentation no later than two days before each class.**