Poznan University of Technology
Faculty of Computing and Telecommunications

michal.apolinarski[at]put.poznan.pl

Course: Application Security – laboratories

Lecturer: Michał Apolinarski, Ph.D.

Topic: User registration process

Duration (on site): 240 min.

Participants: groups of max. 2 persons

PREREQUISITES:

General knowledge of computer networks, operating systems, and databases. Basic programming skills in any language. Familiarity with forms, hashing, tokens, database design, and UML modeling[1].

GOALS:

The purpose of this laboratory is to design and implement a secure user registration module[2], including:

- a user registration form (optionally with local input validation),
- server-side input validation and error handling,
- secure password handling,
- storing user data in a database,
- generating and processing account activation tokens
- creating software documentation (UML and architecture),
- presenting and demonstrating a working prototype.

Optional registration component features (for extra grade):

- password strength meter / advanced password policy,
- email verification via SMTP server / DNS MX lookup,

[1] https://www.visual-paradigm.com/guide/

[2] It is recommended to implement the module as a web application, but desktop or mobile apps are also acceptable. The business domain of the application is irrelevant – the group should focus on the authentication component, not the full application.

- domain restrictions for e-mail addresses (whitelist / blacklist),
- invites tokens,
- rate limiting / CAPTCHA,
- improved password hashing,
- activation-token hardening,
- security event logging,
- enforcing HTTPS and basic transport security,
- … your idea

PLEASE NOTE: This laboratory forms the foundation for the next topics: "Login and session management" and "Password reset feature".

INSTRUCIONS (tasks for a group of max. 2 persons)

PART A (design):

1. Prepare draft documentation[3] describing your planned registration module. The document must include:
    - full details of the student group, course, and exercise,
    - a short description of the component (its purpose, what data it collects, security assumptions),
    - component requirements (functional and non-functional[4]),
    - component architecture (simple diagrams, technology stack),
    - database structure (tables, fields, constraints),
    - UML sequence diagrams for registration and activation (including validations and alternative paths).
2. Send your draft[5] documentation to the lecturer for review.
3. Present and discuss your documentation with the lecturer.

PART B (implementation):

1. Build designed registration components.
2. Prepare and send to lecturer the improved, final[6] documentation, add:

[3] For diagrams it's recommended to use: Draw.io, https://app.diagrams.net
[4] All communication between the client and the server that involves credentials (passwords, activation tokens) must be protected using HTTPS in a real deployment. In this laboratory environment, HTTPS configuration is not strictly required, but the design and documentation must clearly assume the use of HTTPS in production.
[5] Include suffix "_draft" in report filename.
[6] Include suffix "_final" in report filename.

- screenshots,
- explanations of key implementation choices,
- description of security mechanisms,
- conclusions

3. Demonstrate the working functionality (show a complete registration and activation flow, explain your security-related decisions).

REPORT:

- Include a title page with full details of the student's group, course and exercise.
- Should be carefully edited and provide evidence of the completion of all exercises (screenshots, answers, and conclusions).
- **A complete reports must be submitted to the lecturer at least two days before the next class in which it will be presented.**