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: Custom web application development

Duration (on site): 240 min.

PREREQUISITES:

Completion of previous laboratories covering user registration, login, session management, and password reset. Basic knowledge of web applications, databases, HTTP, and common web security issues.

GOALS:

The goal of this laboratory is to extend an existing authentication system into a small content-based web service (e.g. Meme Service or a similar idea such as a microblog) and to apply practical application security mechanisms in a realistic scenario. Students will design and implement:

- role-based access control,
- secure handling of user-generated content (posts, comments, search),
- safe handling of file uploads,
- protection against common access control and injection vulnerabilities,
- documentation of the design and security decisions.

The system is based on distinct types of actors with different privileges and responsibilities:

- **Guest users (unauthenticated)** – can browse public content and use keyword-based search.
- **Registered users (authenticated) –** can add new content (including file uploads), comment / rate posts, and delete only their own content.
- **Administrators (authenticated)** – can moderate the platform by deleting or managing any content item, comment or user account.

Optional components features (for extra grade):

- CSRF protection,
- rate limiting for comments, ratings, uploads or search,
- content reporting and moderation queue (admin approval),
- security event or audit logging,
- advanced file upload hardening (image re-encoding, metadata stripping),
- security headers (CSP, X-Frame-Options, HSTS if HTTPS is enabled),
- soft-delete and restore functionality for administrators,
- optional features from previous laboratories,
- any other security-related feature proposed by the student.

INSTRUCTION (tasks for a group of max 2 persons):

PART A – design (draft documentation)

1. Prepare draft documentation for a content platform integrated with your authentication module. The document must include:
   - full details of the student group, course, and exercise,
   - short description of the service and its actors,
   - functional and non-functional requirements (including security requirements),
   - component architecture (simple diagram, technology stack, storage),
   - database structure (tables, relations, constraints, triggers),
   - UML sequence diagrams (at least one new major process, including alternative paths).
2. Send your draft[1] documentation to the lecturer for review.
3. Present and discuss your documentation with the lecturer.

PART B – Implementation (final documentation)

1. After feedback, implement the required functionality and update your documentation.
2. Prepare and send to lecturer the improved, final[2] documentation, add:
   - screenshots,
   - explanations of key implementation choices,
   - description of security mechanisms,
   - conclusions.
3. Demonstrate the working app.

[1] Include suffix "_draft" in report filename.
[2] Include suffix "_final" in report filename.

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.