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: Security audit – white-box approach (code review)

Duration (on site): 240 min.

PREREQUISITES:

Completion of previous laboratories involving the design and implementation of a web-based content service, as well as the black-box security audit laboratory. Basic knowledge of web application architecture, programming languages used in the project, and common web security vulnerabilities.

GOALS:

The goal of this laboratory is to perform a security audit using a white-box approach, focusing on source code analysis and design review. Students will:

- analyze application source code and internal components,
- identify security vulnerabilities, logic flaws, and insecure design decisions,
- verify and explain vulnerabilities discovered during black-box testing,
- understand the root causes of vulnerabilities at the code and architecture level,
- document findings in a structured security audit report.

GENERAL NOTES

The white-box approach assumes full access to the application source code and configuration. Testing must be conducted in a controlled environment approved by the lecturer. The goal is analysis and understanding, not exploitation or system disruption. Ethical behavior and responsible handling of discovered vulnerabilities are required.

INSTRUCIONS (tasks for a group of max 2 persons)

PART A – Target selection

1. As in the previous laboratory, select one of the following as the target of the security audit:
    - an application developed by another student group during this course[1],
    - ~~an open-source application used in previous laboratories,~~
    - ~~your own application (only if none of the above options are available).~~
2. The selected application must be approved by the lecturer.

PART B – testing (code review and analysis)

1. Perform a security audit using a white-box approach, with full access to the application's source code, configuration files, and documentation.
2. The audit must include manual code review and may be supported by automated analysis tools.
3. **Review the application source** code with particular attention to:
    - authentication and authorization logic,
    - role-based access control and ownership checks,
    - session management and token handling,
    - input validation and output encoding,
    - file upload handling and content processing,
    - error handling and logging,
    - configuration and secret management.
4. **Manually inspect**:
    - critical code paths (login, content creation, deletion, admin actions),
    - access control checks and missing validations,
    - trust boundaries between components,
    - assumptions made by developers that may lead to vulnerabilities.
5. Students should explicitly link identified issues to:
    - broken access control,
    - injection vulnerabilities,
    - insecure design or business logic flaws.
6. **Automated tools (optional, supportive)** may be used to support the review, such as:

[1] Mutual testing between groups is prohibited.

- static analysis tools (e.g. SonarQube, Semgrep),
- dependency and vulnerability scanners,
- security linters or SAST tools.

7. Automated results must be manually reviewed. False positives should be clearly identified and explained in the report.
8. Whenever possible, students should explain correlation with black-box results such as:
   - map vulnerabilities found during black-box testing to their root causes in the source code,
   - explain why the vulnerability was possible and how it could be fixed,
   - identify issues that were not visible in black-box testing.

REPORT:

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