2808ICT/7623ICT
Secure Development Operations
Assignment 2

Code Vulnerability Review

Due Date: 23:59 on 13th October **Group Assignment**
Total Marks: 100 **Weighting: 50%**

Rationale:
The aim of this assignment is to demonstrate your understanding of the theory presented in lectures and the practical exercises conducted in workshops about secure development, common vulnerabilities, software testing and code review.

Teamwork:
This assignment is a group assessment. You can work in the same group as Assignment 1 or change group. Each group should have *2 or 3* students. A group *cannot* mix undergrads and postgrads.

Each group member is responsible for the quality of each task. The workload should be evenly distributed, and by default, each member receives the same mark. If there is a concern about work contribution, please contact the teaching team for mark adjustment.

Tasks:
This assignment has three main tasks. The first task is about automated security testing and vulnerability analysis of the Web application project from Assignment 1. The second task is to find a different (full-stack) project on GitHub and containerize it. The third task requires you to perform automated security testing and vulnerability analysis on the project from the second task.

For Tasks 2 and 3, the requirements of project selection are exactly the same as in Assignment 1:

- The project must include a front-end Web application and a back-end database, and allow you to run the three containers (including a Nginx proxy) described in Task 2.
- The project must run correctly and allow users to use the front-end Web application to add, update, and delete data in the database, which is reflected in the front-end application.
- The GitHub project itself does not provide any Dockerfile nor yaml files for containerizing the project.
- It must be a public GitHub project, and you need to provide the link to it.
- The version of the project cannot be later than July 2023.

For 7623ICT students
You will need to do some research and write one additional section on the security considerations in DevOps and technologies for secure software development. In particular, you need to

- briefly describe the technologies you used in the assignment,
- discuss how they fit into the bigger picture of DevOps, e.g., who will use them at what stages of the software development and deployment,
- what are the major benefits of these technologies compared to more traditional ways of software development.

Submission:

1) *Each team* should submit *one* report to describe the process and your results. In particular, the report must include
    - the student numbers and names of team members;
    - the content of all the files you created, including the Dockerfiles and yaml files;
    - sufficient screenshots showing what you have done and the results;
    - a section with details of the self-learning resources you have used, such as online forums, articles, videos, and ChatGPT, including examples of questions/ keywords you used
    - a section named Revision History (using the table below); and

Date of Change	Contributor	Summary of Change

    - a list of references with proper formatting.

2) *Each team* should submit *one* zipped file containing all the source files you created or adapted. Note that you don't need to submit the files that can be downloaded from GitHub.
3) *Each team* should submit *one* video (no longer than 20 minutes, compressed to max 100 MB) where each team member takes turns to describe what they did in the assignment and demonstrate the results.
    - Each member describes their name and contributions (less than 1 minute), demonstrates the projects are containerized and running correctly (around 2 minutes), and briefly explains how you detect and verify the vulnerabilities (around 4 minutes).
    - Make sure the video and sound can play, otherwise, it will be considered unsubmitted.
    - The goal of the video is to demonstrate your understanding and the quality of your work to help the evaluation of your work.

Task 1. Security Testing and Vulnerability Analysis of the Assignment 1 Project (30 Marks)

For this task, you should use the ZAP project to conduct automated and manual security testing of the project. Describe in detail the steps of your testing and discuss the outcomes of the testing results. You may discuss whether a testing approach is more effective and what you have done to enhance the number and/or the accuracy of the detected vulnerabilities.

You are required to analyze **3** potential security vulnerabilities. The 3 vulnerabilities you identify should be of **3 different classes**, and you should classify them using 3 distinct CWE codes. Higher marks will be given to vulnerabilities that cover a broader range or are very different in nature.

For each of the vulnerabilities, you should write a short report discussing the following:

- How the vulnerability was detected, e.g., through one method or a combination of multiple methods/techniques.
- A description of the vulnerability, including
    - typical example(s) of the vulnerability with explanations,
    - how the vulnerability may be exploited, e.g., example inputs from malicious users that can trigger the attack, and
    - consequences of the vulnerability being exploited.
- A **justification** of why the vulnerability you identified in the project is indeed a security vulnerability corresponding to the CWE classification. You may achieve this by one or more of the following:

    - Verify the vulnerability by trying to exploit it and describe in detail what action you performed.
    - Perform a code review to identify the source code instance, describe where it appears in the project, and provide the code snippets.
    - Conduct some research and cite references to support your claims.
- A recommendation of remediation that would fix the vulnerability or a discussion on why remediation is challenging. Ideally, a fix is provided specific to the project, e.g., by revising the source code.

You should not write more than 2 pages for each vulnerability and use an appendix if necessary. The report format for each vulnerability should contain the following headings.

- Name:
- CWE Code:
- Detection:
- Description:
    - Examples:
    - Exploitation:
    - Impact:
- Justification:
    - Verification (optional):
    - Code Source (optional):
    - Supporting Data (optional):
- Remediation:

Task 2. Containerize and Run a Different Project (30 Marks)

For the second task, you need to find a different project on GitHub and run the project using Docker containers, including at least 3 containers:

- one that runs the front-end Web application,
- one that runs the back-end database, and
- one that runs a Nginx proxy to allow secure HTTPS connections for the project.

Note that the project you find online must have a front-end and a back-end but may not have the reverse proxy part --- you can add the proxy container yourself.

You can run each container individually, or use docker-compose, or Kubernetes. You should demonstrate that your containers are working together in the same way as Assignment 1.

Task 3. Security Testing and Vulnerability Analysis of the Task 2 Project (30 Marks)

For the third task, you should use the ZAP to conduct automated and manual security testing of the project from Task 2 and analyse **3** potential security vulnerabilities belonging to **3 different classes** in the same way as Task 1.

It is not required that these 3 vulnerabilities belong to 3 classes that are completely different from those in Task 1, but higher marks will be given to vulnerabilities that are so.