LOG8371E: Software Quality Engineering (Fall 2023)

TP3 (10% weight)

Security

Notes:

Practical work is an important part of the course and is intended to motivate you to understand software quality assurance practices, design software quality assurance plans, develop verification strategies, and use the various tools available to evaluate the quality characteristics of software. It is recommended that you take this work seriously and use your creativity and critical thinking to make it a success. Collaboration with your colleagues is allowed during and outside the laboratory sessions. However, plagiarism regulations still apply at all times.

I) Objectives:

The objectives of this third practical work are to get familiar with:

- Software security objectives and security assurance process.
- Identification of security vulnerabilities through static analysis of the source code.
- Identification of security vulnerabilities through penetration testing.

II) Specifications:

Q1) Static analysis (45 points)

Perform a static analysis of the source code of the PetClinic system using SonarCloud or SonarQube. You should focus on your chosen feature (REST controller). Prepare a report of the results of the static analysis, including a) a summary of the results, b) comments for eight vulnerabilities or security hotspots from at least three different types (focus on Blocker/Critical/Major issues). Each comment should include a short description of the vulnerability and its potential risk (you can use an example of an attack), the file of the vulnerability, the severity level, the type of the vulnerability according to OWASP, SANS, or CWE, and a recommendation for solving the problem. In addition, provide a small manual of how you perform the static analysis. Discuss the challenges that you encountered and how you solved them.

Tips:

- You can use either SonarCloud or SonarQube to perform the static analysis.
- A quick view of SonarCloud: https://docs.sonarsource.com/sonarcloud
- A quick view of SonarQube: https://docs.sonarsource.com/sonarqube

- Choosing SonarCloud or SonarQube? https://community.sonarsource.com/t/sonarcloud-vs-sonarqube/9557/2
- There is tutorial on Moodle that demonstrates how to use SonarCloud to perform static analysis on DVWA demo application.

Required tools:

- Alternative 1: SonarCloud + SonarScanner (available independently or as plugins for mainstream CI tools such as Maven)
- Alternative 2: SonarQube + SonarScanner (available independently or as plugins for mainstream CI tools such as Maven)

Q2) Penetration testing (45 points)

Perform penetration testing on the PetClinic system using the OWASP ZAP tool. You can either consider the entire software or focus on your chosen feature. Prepare a report of the results of the penetration testing, including a) a summary of the results, b) comments for eight alerts (potential vulnerabilities) from at least three different types (focus on the vulnerabilities from the highest to the lowest severity level and highlight the OWASP top 10 vulnerabilities). Each comment should include a short description of the vulnerability and its potential risk (you can use an example of an attack), the URL of the vulnerability, the severity level, the type of the vulnerability according to OWASP or CWE, and a recommendation for solving the problem. In addition, provide a small manual of how you perform the penetration testing. Discuss the challenges that you encountered and how you solved them.

Tips:

- A quick view of ZAP: https://www.zaproxy.org/getting-started/.
- There is a tutorial on Moodle that demonstrates how to use ZAP to perform penetration testing on the DVWA demo application. You can follow a similar approach.
- Combine manual walkthroughs of the webpages and automatic scans to achieve maximum coverage.
- You should avoid port conflicts between PetClinic and ZAP (don't use localhost:8080 as the PetClinic port).

Required tools:

- OWASAP ZAP

Q3) Compare static analysis and penetration testing results (10 points)

Compare the results of the static analysis (from Q1) with the results of the penetration testing (from Q2). Discuss the differences between the results of the two approaches in terms of the numbers and the CWE categories of the detected vulnerabilities (note: both the

SonarCloud/SonarQube and ZAP tools report the CWE categories of each vulnerability). Discuss your reasoning for the differences (why are some categories of vulnerabilities found only by one of the approaches?)

Tips:

- CWE (Common Weakness Enumeration) is a categorization of vulnerabilities and weaknesses. Link: https://cwe.mitre.org/.
- Understanding Vulnerabilities, CWE, and CVE: https://www.codiga.io/blog/cve-vs-cwe/

III) Presentation:

Each team should submit a report no longer than 15 pages. The report must contain: 1) a table of contents; 2) an abstract that provides a clear overview; 3) an introduction that describes the subject software, the importance of its quality, the scope of the work (the chosen feature or the entire software), the considered quality characteristics, and an overview of the approaches (you can reuse part of the introduction from TP1); 4) the static analysis for detecting security vulnerabilities (process, results, and discussion of challenges and solutions); 5) the penetration tests for detecting security vulnerabilities (process, results, and discussion of challenges and solutions); 6) comparison of the static analysis and penetration testing results; and 7) a conclusion that summarizes the key points of the report. Diagrams and tables should be clear and have captions; diagrams and tables are self-explanatory or explained in the text. References should be used appropriately when necessary. The report shall be of high quality and be treated professionally (suppose you will submit it to the managers of a company).

IV) Evaluation:

Your report will be evaluated by both the quality of work and the quality of presentation:

Team Score = Quality of work * sqrt (Quality of presentation) * 10 (Weight of TP3)

For example, if you get 90% for the quality of work and 90% for the quality of presentation, you score will be: 0.9 * sqrt (0.9) * 10 = 8.5

Individual Score = Team Score * Contribution Factor

Contribution Factor (CF): CF = 1 for satisfactory contribution; CF = 0 for no contribution; 0 < CF < 1 for unsatisfactory contribution. CF is determined by optional peer reviews.

Optional peer reviews: Each team member may optionally email a confidential statement of work to the lab instructor within 24 hours after the due date of the assignment. A statement of work first lists in point form the parts of the assignment to which each team member contributed. In addition, the statement of work also describes whether the work load was distributed fairly evenly among the team members (you may describe the percentages of contribution by each team member). A statement of work will be used to determine the contribution factor and the score of a team member who is not contributing sufficiently to the assignment, subject to further investigations. It is not necessary to send a statement of work, if a team distributed the work for the assignment fairly evenly and each team member had satisfactory contribution.

I strongly encourage each team to use a **private GitHub repository** for the collaboration. The repository can not only help the team manage and track the work, but also provide evidence for each individual's contribution in case an investigation is needed.

V) Submission:

The work must be done in a team (same team as TP 1 and TP2) and must be submitted via Moodle no later than:

➢ **November 30th before 11:59 pm** (one submission is needed for each team).

Please submit a PDF report that includes the names and student numbers of the team members. The file will have the name:

➢ **log8371E_TP3_[TeamName].pdf**

Late work will be penalized by 10% per day of delay. No work will be accepted after 4 days of delay.