CS 305 Project One Guidelines and Rubric

Competency

In this project, you will demonstrate your expectations in the following competency:

- Analyze how advanced security concepts are applied to develop secure code

Scenario

You work as a developer for a software engineering company called Global Rain. Global Rain specializes in custom software design and development. The software is for entrepreneurs, businesses, and government agencies around the world. Part of the company's mission is "Security is everyone's responsibility." Global Rain has promoted you to its new agile scrum team.

At Global Rain, you work with a client, Artemis Financial. Artemis Financial is a consulting company that develops individualized financial plans for its customers. The financial plans include savings, retirement, investments, and insurance.

Artemis Financial wants to modernize its operations. As a crucial part of the success of its custom software, the company also wants to use the latest and most effective software security. Artemis Financial has a RESTful web application programming interface (API). The company is seeking Global Rain's expertise about how to protect the organization from external threats.

As part of the team, you must examine Artemis Financial's web-based software application to identify any security vulnerabilities. You'll document what you learn in a vulnerability assessment report used to mitigate the security vulnerabilities you find.

Directions

You must conduct a vulnerability assessment. In the assessment, you will examine Artemis Financial's web-based software application. Use what you have learned so far, the resources in the Supporting Materials section, and the resources in the Resources section of this module to help you. Review and analyze the security vulnerabilities specific to Artemis Financial's web-based software application. Use the Project One Template linked in the What to Submit section to document the following items for your vulnerability assessment report:

1. **Interpreting Client Needs**: Review the scenario to determine your client's needs and potential threats and attacks associated with its application and software security requirements. Document your findings in your vulnerability assessment report. Consider the scenario information and the following questions regarding how companies protect against external threats:
    A. What is the value of secure communications to the company?
    B. Does the company make any international transactions?
    C. Are there governmental restrictions about secure communications to consider?
    D. What external threats might be present now and in the immediate future?
    E. What are the modernization requirements that you must consider? For example:
        i. The role of open-source libraries
        ii. Evolving web application technologies

2. **Areas of Security**: Use what you have learned in step 1 and refer to the vulnerability assessment process flow diagram provided in the Supporting Materials section. Think about the functionality of the software application to identify which areas of security apply to Artemis Financial's web application. Document your findings in your vulnerability assessment report and justify why each area is relevant to the software application.

    Note: Not all seven areas of security in the vulnerability assessment process flow diagram apply to the company's software application.

3. **Manual Review**: Refer to the seven security areas outlined in the vulnerability assessment process flow diagram. Use what you've learned in steps 1 and 2 to guide your manual review. Identify all vulnerabilities in the Project One Code Base linked in the Supporting Materials section by manually inspecting the code. Document at least 7 to 10 findings in your vulnerability assessment report. Include a description that identifies where the vulnerabilities are found. Provide the specific class file, if applicable.
4. **Static Testing:** Integrate the dependency-check plug-in into Maven by following the instructions in the Integrating the Maven Dependency-Check Plug-in tutorial provided in the Supporting Materials section. Run a dependency check on Artemis Financial's software application to identify all security vulnerabilities in the code. Specifically, identify all vulnerabilities in the code base by analyzing results from running the code through a static test. Include the following items from the dependency-check report in your vulnerability assessment report:
    A. The names or vulnerability codes of the known vulnerabilities
    B. A brief description and recommended solutions that are found in the dependency-check report
    C. Any attribution that documents how this vulnerability has been identified or how it was documented in the past

5. **Mitigation Plan:** Interpret the results from the manual review and static testing report. Identify steps to mitigate the identified security vulnerabilities by creating an action list that documents how to fix each vulnerability in your vulnerability assessment report.

   Note: You do not need to fix these vulnerabilities in this project.

What to Submit

To complete this project, you must submit the following:

Vulnerability Assessment Report

Use the Project One Template to complete your vulnerability assessment report.

Supporting Materials

The following resources support your work on the project:

Diagram: Vulnerability Assessment Process Flow Diagram

Reference this process flow diagram during the project to determine which of the seven areas of security to assess for Artemis Financial's software application.

A text-only version is available: Vulnerability Assessment Process Flow Diagram Text-Only Version.

Code Base: Project One Code Base

Open a new Java project in Eclipse and upload this zipped file folder. This folder contains the code for the web application from Artemis Financial. The folder also contains security vulnerabilities for you to identify using the guidelines provided.

Tutorial: Integrating the Maven Dependency-Check Plug-In Tutorial

Follow the instructions in this tutorial to learn how to integrate the dependency-check plug-in into Maven. You'll need to edit the pom.xml file to add the dependency-check plug-in to Artemis Financial's software application. Eclipse will run the Maven plug-in when you compile your code.

Project One Rubric

Criteria	Exceeds Expectations (100%)	Meets Expectations (85%)	Partially Meets Expectations (55%)	Does Not Meet Expectations (0%)	Value
Interpreting Client Needs	Exceeds expectations in an exceptionally clear, insightful, sophisticated, or creative manner	Determines client's needs and potential threats and attacks associated with its application and software security requirements	Shows progress toward meeting expectations, but with errors or omissions	Does not attempt criterion	15
Areas of Security	Exceeds expectations in an exceptionally clear, insightful, sophisticated, or creative manner	Identifies which areas of security apply to Artemis Financial's web application, and justifies why each area is relevant to the software application	Shows progress toward meeting expectations, but with errors or omissions	Does not attempt criterion	15
Manual Review	Exceeds expectations in an exceptionally clear, insightful, sophisticated, or creative manner	Identifies all vulnerabilities in the code base by manually inspecting the code	Shows progress toward meeting expectations, but with errors or omissions	Does not attempt criterion	20
Static Testing	Exceeds expectations in an exceptionally clear, insightful, sophisticated, or creative manner	Identifies all vulnerabilities in the code base by analyzing results from running the code through a static test	Shows progress toward meeting expectations, but with errors or omissions	Does not attempt criterion	20
Mitigation Plan	Exceeds expectations in an exceptionally clear, insightful, sophisticated, or creative manner	Identifies steps to mitigate the identified security vulnerabilities by creating an action list that documents how to fix each vulnerability	Shows progress toward meeting expectations, but with errors or omissions	Does not attempt criterion	25
Clear Communication	Exceeds expectations with an intentional use of language that promotes a thorough understanding	Consistently and effectively communicates in an organized way to a specific audience	Shows progress toward meeting expectations, but communication is inconsistent or ineffective in a way that negatively impacts understanding	Shows no evidence of consistent, effective, or organized communication	5
				Total:	100%