CS 305 Module Two Written Assignment Guidelines and Rubric

Overview

Writing code is difficult. Writing secure code can be even more challenging. As the developer, you are responsible for writing secure code. You'll know your code is secure when you manually search for and identify possible security vulnerabilities. Developing this skill is important because it becomes more challenging as the number of lines and complexity of your code increase.

As you learned in this module, you can follow a workflow. You can also use tools widely accepted in software security and vulnerability assessments. You can focus your manual code inspection and narrow your search for possible security vulnerabilities within your code by following the vulnerability assessment process flow diagram.

Specifically, in this assignment, you will complete the following actions:

- Determine relevant areas of security for a software application.
- Identify software security vulnerabilities by manually reviewing source code.
- Identify potential mitigation techniques that have been used to mitigate vulnerabilities associated with known exploits.

Scenario

You are a senior software developer on a team of software developers. You are responsible for a complex web application that uses the Spring framework. The team has been tasked with implementing an expressive command input function for the application. The team is using version 2.6.5 of the spring-data-rest-webmvc in the Spring framework. You also want to use the Spring Expression Language to accomplish the task.

Review the resources in this module's Resources section to learn about the Spring framework.

Directions

As the lead person on this application, you are responsible for making certain that the code is secure. You will need to assess potential vulnerabilities in the code and create a mitigation plan for any existing vulnerabilities that the software development team must address.

To begin, see the vulnerability assessment process flow diagram linked in the Supporting Materials section to help guide your code review and mitigation plan.

Specifically, you must address the following rubric criteria:

1. **Areas of Security:** Review the scenario and use what you know about the architecture of the web application to identify relevant areas of security that are applicable for a software application:
   A. Provide sufficient detail to address which of the seven areas of security are relevant to assess from the first level of the vulnerability assessment process flow diagram.
   B. Document your findings for the software development team in the Module Two Written Assignment Template linked in the What to Submit section.

2. **Areas of Security Justification:** Provide a justification and rationale for why each area of security is relevant to the software application.
3. **Code Review Summary:** Once you have identified the relevant areas of security to review from the first level of the vulnerability assessment process flow diagram, work through the second level. At this stage, you should complete the following actions:
   A. Manually inspect the code base provided to identify which vulnerabilities exist. To do this, upload the Module Two Written Assignment Code Base linked in the Supporting Materials section as a new project into Eclipse.
   B. Refer to the Uploading Files to Eclipse Desktop Version Tutorial linked in the Supporting Materials section to learn how to open the code base for review.
   C. Document your findings in detail for the software development team in the Module Two Written Assignment Template.

4. **Mitigation Plan:** Once you have manually inspected the code and identified the security vulnerabilities, complete the following actions:
   A. Describe potential mitigation techniques. For example, describe secure software designs that you could use to address the software security vulnerabilities you identified.
   B. Refer to the Module Two Resources section for help with this response.
   C. Document your findings for the software development team in the Module Two Written Assignment Template. The software development team will use this plan to address all vulnerabilities in the code.

What to Submit

Submit a completed Module Two Written Assignment Template as a 1- to 2-page Microsoft Word document.

Supporting Materials

The following resources support your work on this assignment:

Diagram: Vulnerability Assessment Process Flow Diagram

This diagram illustrates the process flow for conducting an architecture review and code review to identify security vulnerabilities in code. Reference this diagram as you complete the assignments for this module.

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

Code Base: Module Two Written Assignment Code Base

This resource provides the code base needed for the Module Two written assignment.

Tutorial: Uploading Files to Eclipse Desktop Version Tutorial

This tutorial highlights how to upload files to Eclipse.

Module Two Written Assignment Rubric

Criteria	Meets Expectations (100%)	Partially Meets Expectations (70%)	Does Not Meet Expectations (0%)	Value
Areas of Security	Identifies relevant areas of security that are applicable for a software application	Shows progress toward meeting expectations, but with errors or omissions	Does not attempt criterion	20
Areas of Security Justification	Provides justification and rationale for why each area of security is relevant to the software application	Shows progress toward meeting expectations, but with errors or omissions	Does not attempt criterion	15
Code Review Summary	Manually inspects the code base provided to identify which vulnerabilities exist and documents findings for the software development team	Shows progress toward meeting expectations, but with errors or omissions	Does not attempt criterion	30
Mitigation Plan	Describes potential mitigation techniques such as secure software designs that could be used to address the identified software security vulnerabilities	Shows progress toward meeting expectations, but with errors or omissions	Does not attempt criterion	30
Clear Communication	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%