CS 410 Project Two Guidelines and Rubric

Competency

In this project, you will demonstrate your mastery of the following competency:

- Review common security issues that may be identified through reverse engineering

Scenario

In Project One, you applied for a Software Engineer position at a large investment company called SNHU Investments. Recently, they have been trying to move their legacy code into a cloud-based application. You were selected to participate in a proficiency test as part of the interview process. In the proficiency test, you demonstrated your ability to reverse engineer code.

For the next part of the proficiency test, you have been asked to identify and explain security vulnerabilities within the client management application. You will identify multiple security vulnerabilities within blocks of assembly code, explain the vulnerabilities, and describe recommendations to fix the vulnerabilities.

Directions

Using the C++ (CPP file) that you created in Project One, you will identify multiple security vulnerabilities, explain the vulnerabilities, and describe recommendations to fix the security vulnerabilities.

1. Identify where multiple security vulnerabilities are present within the blocks of C++ code.
    - Use the first section in the Project Two Security Report Template, located in the Supporting Materials section, to map each security vulnerability to the block of C++ code.
2. Comment within the C++ code (CPP file) to indicate where the security vulnerabilities are identified.
3. Explain the **security vulnerabilities** that are found in the blocks of C++ code.
    - Use the second section of the Project Two Security Report Template to explain in detail how and why these are security vulnerabilities.
4. Describe **recommendations** for how the security vulnerabilities can be fixed.
    - Use the third section of the Project Two Security Report Template to complete this step.
5. Fix some of the security vulnerabilities within the "main" function.
    - Correct the C++ code to fix the security vulnerabilities. There will be security vulnerabilities which you cannot correct by adjusting the C++ code. Determine which vulnerabilities you are able to fix by adjusting the C++ code, and fix them. You are not required to fix the others.
6. Comment within the C++ code to indicate how the security vulnerabilities were fixed.
7. Convert the CPP file to a binary file (O file).

What to Submit

To complete this project, you must submit the following:

Project Two Security Template

This should be a Word Document (DOCX) identifying where the security vulnerabilities are, explaining how these are security vulnerabilities, and including recommendations to fix them. Use the template provided in the Supporting Materials section.

C++ File (CPP file)

This file includes your comments on the identified security vulnerabilities, the fixes, and the comments about how the security vulnerabilities were fixed.

Binary (O file)

This file includes some of the repairs of the security vulnerabilities.

Supporting Materials

The following resource(s) may help support your work on the project:

Project Two Security Report Template Word Document

Use this Word document template to complete steps one, three, and four in the directions.

Guide to Software Reverse Engineering

This guide provides information on the process of software reverse engineering, assembly language, binary, C++, and more.

Codio Guide

This guide provides information on how to use the Codio platform to complete aspects of this project.

Project Two Rubric

Criteria	Exemplary	Proficient	Needs Improvement	Not Evident	Value
Security Vulnerabilities: Identified	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Identifies where multiple security vulnerabilities are within the blocks of C++ code (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include identifying more than a few security vulnerabilities within the blocks of C++ code (55%)	Does not attempt criterion (0%)	26.67
Security Vulnerabilities: Commented	N/A	Comments within the C++ code (CPP file) where the security vulnerabilities are identified (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include adding comments within the C++ code (CPP file) where the security vulnerabilities are identified (55%)	Does not attempt criterion (0%)	6.67
Security Vulnerabilities: Explained	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Explains the security vulnerabilities that are found in the blocks of C++ code (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include richer detail as to how and why there are vulnerabilities within the C++ code (55%)	Does not attempt criterion (0%)	16.67
Security Vulnerabilities: Recommendations	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Describes recommendations for how the security vulnerabilities can be fixed (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include richer detail to the described recommendations (55%)	Does not attempt criterion (0%)	13.33
Security Fixes	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Fixes security vulnerabilities within the “main” function (85%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include fixing security vulnerabilities within the “main” function (55%)	Does not attempt criterion (0%)	26.67
Security Fixes: Commented	N/A	Comments within the C++ code (CPP file) about how the security vulnerabilities were fixed (100%)	Shows progress toward proficiency, but with errors or omissions; areas for improvement may include adding comments within the C++ code (CPP file) about how the security vulnerabilities were fixed (55%)	Does not attempt criterion (0%)	6.67
Articulation of Response	Exceeds proficiency in an exceptionally clear, insightful, sophisticated, or creative manner (100%)	Clearly conveys meaning with correct grammar, sentence structure, and spelling, demonstrating an understanding of audience and purpose (85%)	Shows progress toward proficiency, but with errors in grammar, sentence structure, and spelling, negatively impacting readability (55%)	Submission has critical errors in grammar, sentence structure, and spelling, preventing understanding of ideas (0%)	3.32

Total:	100%