Course Outline

Writing Secure Code

Course Summary

Description

When writing software, we're faced with many different requirements: It should work, it should have good performance, and it should be easy to update and change. Most of the requirements we're facing have one thing in common: If mistakes are made, they are usually obvious. With security, things are different. It can take moments to write insecure code, but months of hard work to find and exploit it.

In this course, we'll learn the patterns that make our code insecure, as well as the concepts behind them. We'll explore famous hacks from-the-past, and understand how to mitigate security risks.

This course is t focused on the C++ Programming. We reference the current CERT C++ Coding standard as another reference source for the course.

Objectives

After taking this course, students will be able to:

- Identify and explicate the relevant portions of the NIST Special Publication 800-53; and the CERT SEI Secure C++ Coding Standards that are relevant to the topics selected for the course. (https://wiki.sei.cmu.edu/confluence/pages/viewpage.action?pageId=88046682)
- Provide specific C++ coding practice and examples that implement the identified points from the standards including related best practices used in the secure C++ coding standards community.
- Explore how the secure programming standards interact with and impact the other aspects of managing a secure closed system including but not limited to network hardening, encryption, intrusion detection and standard preventative measures in both design and operations

Topics

- C++ Secure Coding Environment and Practices
- C++ Coding practices to CERT and NIST standards.
- Secure Networking

Audience

This course is designed for software developers.

Prerequisites

To take this course, you must have practical software development experience, and feel comfortable with jumping into new concepts and programming languages. Please note that this course can be customized to any programming language of the client's choice

Duration

Two days

Course Outline

Writing Secure Code

Course Outline

I. C++ Secure Coding Environment and Practices

A. Primarily this section would refer to the standards and best practices that are used for secure configuration management of C++ code, coding practices, documentation standards and code handling practices necessary for asset control, review, auditing and forensic analysis.
B. The other main topic in this section would be process based covering build, release and test practices for C++ applications in a secure environment.
C. Given the scope of the material, I would expect this to be a very high level and brief overview unless otherwise directed. This module can be omitted without impacting the rest of the material.

II. C++ Coding practices to CERT and NIST standards.

The main topics identified in C++ Coding standards would be covered with specific examples of how to implement those in C++ code as well as to identify and correct code that may fail one these points. Suggested points include but are not limited to the following C++ CERT Secure Coding Standards:

A. Declarations and Initialization
B. Expressions
C. Integers
D. Containers
E. Characters and Strings
F. Memory Management
G. Input Output
H. Exceptions and Error Handling
I. Object Oriented Programming
J. Concurrency
K. Miscellaneous

The instructor expects this module would take up the bulk of the course and would break down into about 5 teachable sub-presentation modules.

III. Secure Networking

This module will reference the relevant portions of the NIST Cybersecurity framework https://www.nist.gov/cyberframework but only within the context of the portions that are relevant to secure coding. Suggested topics here are:

A. Best practices for managing network connections in C++
B. Hardening connections.
C. Encryption, validation, and verification protocols.
D. Survey of network intrusion detection tools and C++ libraries.
E. Survey of network hardening tools