

Course 644: Secure Programming Techniques in C (4 days)

Course Description...

This course works thru the problems and mitigation techniques of doing secure programming in C. The course starts by covering the problems of the language definition and gives recommendations for a subset of the language that is best for secure programming. It also covers compiler and linkage issues to include tests to run on the compiler. The general principles of secure code are covered with example code illustrating the correct and the wrong way to implement the principles. There follows several sections on specific problem areas and mitigation techniques. (Please see course outline.) The last section of the course cover changes needed in the software lifecycle development to support secure programming.

Learning Objectives...

- Know the principles of secure coding
- List areas of the C programming language which are harmful to secure coding
- Be able to implement methods to determine how the compiler reacts at boundary conditions
- Be able to identify specific programming vulnerabilities and know the possible mitigations against them
- Know the changes necessary in the software development lifecycle to support secure coding

Who should attend...

Programmers who want to learn principles, approaches and techniques of secure programming in C.

Prerequisites...

Without a strong background in C, this course will be of little use to the attendee. The course is taught on a Linux platform using gcc 4.1.1. A slight background in Linux, gcc, and make would be useful.

See next page for a detailed course outline...



Course Outline...

Security Review

- What is Security Programming?
- What Causes Security Problems
- Goals of Secure Programming

Language Problems

- There is a Standard
- Unspecified Behavior
- Undefined Behavior
- Locale Specific Behavior
- Language Feature Restrictions

Compiler and Linkage Problems

- Options Review
- Standard not implemented
- Standards implemented Differently
- Conversions and operations
- Libraries and Linkage Problems
- Influence of the Operating System

Software Development Principles

- Economy of Mechanism
- Fail-Safe Defaults
- Complete Mediation
- Open Design
- Separation of Privilege
- Least Privilege
- Least Common Mechanism
- Psychological Acceptability

String Manipulation Techniques

- Common String Manipulation Error
- Input Data
- Buffer Overflows
- Stack Smashing
- Code Injection
- Arc Injection

Process Techniques

- Environment Variables
- Sandboxes and Jails
- The Special Problems of SUID Programs
- Shared Memory Problems



Pointer Techniques

- Common Pointer Error
- Function Pointers
- atexit() and on_exit()
- longjmp() Function
- Exception Handling

Memory Management Techniques

- Common Memory Management Errors
- Leaving Things in Memory
- Double-Free Vulnerabilities
- Writing to Freed Memory
- Heap Problems
- Swap File Problems

Math Techniques

- Integer Overflows
- Sign Errors
- Truncation Errors
- Safe operations
- Testing Techniques

File I/O Techniques

- Common I/O Errors
- Input Variable Problems
- Race Conditions
- Mutual Exclusion and Deadlock
- Permission Problems
- Symbolic Link Exploits
- Temporary File Exploits
- Deleting Files

Software Engineering

- Static Analysis
- Penetration Testing
- Security Review

Please contact your ROI representative to discuss course tailoring!!!