Secure Coding in C and C++

Producing secure programs requires secure designs. However, even the best designs can lead to insecure programs if developers are unaware of the many security pitfalls inherent in C and C++ programming. This four-day course provides a detailed explanation of common programming errors in C and C++ and describes how these errors can lead to code that is vulnerable to exploitation. The course concentrates on security issues intrinsic to the C and C++ programming languages and associated libraries. The intent is for this course to be useful to anyone involved in developing secure C and C++ programs regardless of the specific application.

Please note: you must bring a laptop computer equipped with the latest version of Adobe Reader and VMware Player. See the Prerequisites section for download information.

The course assumes basic C and C++ programming skills but does not assume an in-depth knowledge of software security. The ideas presented apply to various development environments, but the examples are specific to Microsoft Visual Studio and Linux/GCC and the 32-bit Intel Architecture (IA-32). Material in this presentation was derived from the Addison-Wesley books *Secure Coding in C and C++* and *The CERT C Secure Coding Standard*.

Who should attend?

This course is designed for C and C++ developers.

Topics

- string management
- dynamic memory management
- integral security
- formatted output
- file I/O

Subjects covered in the first two days are general, but examples are taken from both the Microsoft Visual Studio and GCC compilers on Windows and Linux platforms. Course material on integers uses examples from the IA-32 architecture.

The third and fourth days of the course focus on POSIX platforms. Doug Lea's malloc (dlmalloc) is used to demonstrate exploits in the Linux environment, while the file I/O sections focus on UNIX and the UNIX file system (UFS).

An online demonstration version of the course can be accessed at http://oli.web.cmu.edu. Enter the course key: sc-demo-4.

Objectives

Participants should come away from this course with a working knowledge of common programming errors that lead to software vulnerabilities, how these errors can be exploited, and effective mitigation strategies for preventing the introduction of these errors. In particular, participants will learn how to

- improve the overall security of any C or C++ application
- thwart buffer overflows and stack-smashing attacks that exploit insecure string manipulation logic
- avoid vulnerabilities and security flaws resulting from the incorrect use of dynamic memory management functions
- eliminate integer-related problems: integer overflows, sign errors, and truncation errors
- correctly use formatted output functions without introducing format-string vulnerabilities
- avoid I/O vulnerabilities, including race conditions

Moreover, this course encourages programmers to adopt security best practices and develop a security mindset that can help protect software from tomorrow's attacks, not just today's.

Prerequisites

It is recommended that participants have a basic to intermediate understanding of the C and C++ programming languages. Software security knowledge or experience is not required.

Required Equipment

Students must bring a personal computer equipped with

- 3GB or greater of free hard disk space
- C and C++ programming language development environments (compiler, editor, etc.)
- CD-ROM or memory stick
- the latest version of Adobe Reader (this can be downloaded from http://www.adobe.com/products/acrobat/readstep2.html)
- the latest version of VMware Player (this can be downloaded from http://www.vmware.com/download/player/)