CPP201 - DEFENDING C AND C++

Course Learning Objectives

Discover how C and C++ vulnerabilities occur in software applications. Describe the dangers of poor memory management, buffer overflows, pointers, and format string exploits. In addition, learn about the common pitfalls of programming in C/C++ by exploring SEI CERT secure coding standards.

Description

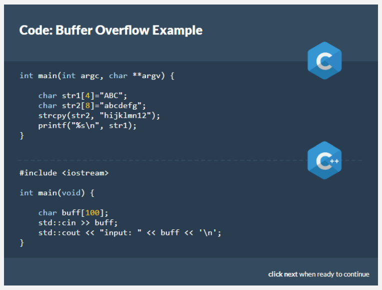
Software vulnerabilities often occur in C/C++ languages because they do not have strong protection mechanisms. Students will learn about how the inherent characteristics of these languages can be exploited to cause a range of vulnerabilities. This course also takes a look at some of the coding standards widely used by the Software Engineering Institute.

Audience

C and C++ developers

Time Required

Tailored learning - 60 minutes total (approx.)

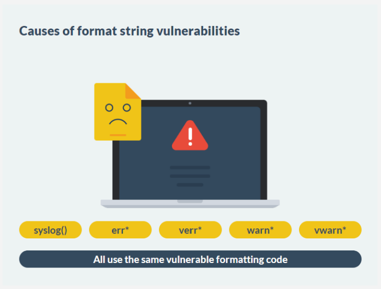


Code: Buffer Overflow Example

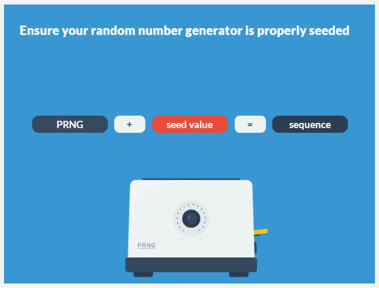
```
int main(int argc, char **argv) {

    char str1[4]="ABC";
    char str2[8]="abcdefg";
    strcpy(str2, "hijklmn12");
    printf("%s\n", str1);
}
```

```
#include <iostream>

int main(void) {

    char buff[100];
    std::cin >> buff;
    std::cout << "input: " << buff << '\n';
}
```

click next when ready to continue



Causes of format string vulnerabilities



Ensure your random number generator is properly seeded

CPP201 - DEFENDING C AND C++

Course Outline

1. Memory Organization

- About C/C++
- Trust the developers
- Vulnerabilities
- Memory
- Memory space layout
- Environment data and pointers
- Argument strings and pointers and argument count
- The stack
- The heap
- .bss / .data / .text
- Pointers
- Pointer arithmetic
- Bad pointer arithmetic examples
- Prevent pointer arithmetic vulnerabilities

2. Buffer Overflow

- What is buffer overflow?
- Buffer overflow example
- Vulnerability: Admin access
- Vulnerability: Running arbitrary code
- Before you defend
- Unsafe APIs
- Strlen() and extraction operator
- API quirks
- Avoid unsafe APIs
- Avoid dangerous functions
- Terminate variables and review code

3. Format String Attacks

- Format string attacks
- Format string attacks examples
- Conversion specifiers
- Conversion specifier %n
- Washington University's example
- The impact of format string attacks
- Causes of format string vulnerabilities

4. SEI CERT C Coding Standards

- Coding standards
- SEI CERT C/C++ coding standards
- Expressions
- Arrays
- Characters and strings
- Memory
- Input and output
- Environment

5. SEI CERT C++ Coding Standards

- Declarations and initializations
- Containers
- Memory
- Exceptions and error handling
- Miscellaneous

SecurityCompass