(In)Secure C++

4-day training

Are you a C++ programmer who wants to learn how to write more secure and robust code? Do you want to understand how hackers exploit common vulnerabilities in native applications? Do you want to get a firmer understanding of the domain and language of native exploitation? Do you want to use the best tools and practices to find, prevent and fix security bugs?

If you answered yes to any of these questions, then you should enroll in the ***"(In)Secure C++"*** training by TurtleSec. This ***hands-on*** training is designed to provide you with a solid foundation in security for native applications, covering topics such as:

- **Fuzzing and Sanitizers**: How to use tools like Address Sanitizer and fuzzers like AFL/libFuzzer to find and fix security vulnerabilities. Here you will use fuzzing to find the Heartbleed vulnerability in OpenSSL and fix it.
- **Exploiting Buffer Overflows with Custom Exploit Shellcode**: How to exploit buffer overflows and execute arbitrary code, and the mitigations that can help prevent it from happening. Here you will exploit a program with your own custom shellcode.
- **Return Oriented Programming and Format Strings**: How to bypass stack protection mechanisms using Return Oriented Programming (ROP) and looking at format string vulnerabilities as an example of a completely different way of exploiting applications. Here you will use tools to generate ROP chains.
- **Memory Managers and Heap Exploitation**: How to understand and manipulate the memory layout and exploit heap-based vulnerabilities.
- **Reverse Engineering and Sandboxing**: How to analyze binary code and use sandboxing techniques to isolate untrusted code.
- **Secure Coding Practices**: How to write secure C++ code and avoid common pitfalls.

The training can be done either remotely or on-site. You will learn from experienced instructors who will guide you through theory, exercises, and discussions. You will also get access to a personal cloud VM where you can practice safely, and a Cyber Dojo cloud instance for group exercises.

Don’t miss this opportunity to improve your skills and knowledge in C++ security. Register now for the next session of ***"(In)Secure C++"*** by TurtleSec!