Dominic Drury

06/12/2024

CS-410

6-1 Journal

Define: A security vulnerability in the software development sense is an area of code that can allow a hacker to perform actions that are unintended by the developer. These kinds of actions can be allowing a hacker to take down the program by overloading it, gaining access to privileged information, learning proprietary information, etc.

Identify: Vulnerabilities that can be identifiable in C++ code are numerous. Some examples are over and underflow vulnerabilities that allow someone to go outside of the bounds of a variable and cause unintended functionality, invalid inputs that allow an attacker to sneak in commands to cause unintended functionality or crashes, unhandled error catching that allow an attacker to cause an abort of the system, and unhandled memory deallocation that can cause crashes, just to name a few.

Purpose: Looking for vulnerabilities during legacy to C++ conversion rather than during testing is useful for two reasons. The first is that you are at that time you are going through the existing code nearly line by line and functionality nearly step by step so vulnerabilities can be easier to notice and the second is that coding securely should be done at every step of the program, and not saved for any specific part of the software development life cycle.

Solutions: Finding out the appropriate fix for a security vulnerability can be a challenging process. You need to be able to prevent the vulnerability without causing new ones, slowing down the system, breaking other functionality, etc. It can be a challenge, but I think that for most vulnerabilities they can be fixed by following best practices, SEI guidelines, and the company's security guidelines.