James Soto

7-15-2023

CS-405 Secure Coding

5-3 Activity: Static Code Analysis

This assignment supplied a file with questionable source code that required that we complete an analysis on the code to identify issues by using the built in analysis tools in Clion (supposed to be Visual Studio but I have a MacBook) and CppCheck. To start I compiled and ran the code to see what errors it showed. Immediately the program failed on line 129 Assertion failed: (my\_function() == 3), function main, file main.cpp, line 129, which was caused by the function my\_function() returns a bool value while it is being are compared with an int value. Commenting out line 129 and uncommenting line 123 let the program run and display the following:

Welcome to the Questionable Code Test!

I summed up 499500 as the answer.

x + y + z = 15

myobject.MyVal = 1

Like I mentioned earlier I used Clion on my MacBook to run the static analysis code. I installed the necessary plugins to run a static analysis using Clion’s built in tools. Clion displayed 26 errors and 8 minor warnings. I attached a picture of the Clion errors and the inspection tool I used. The first is C:: =is\_type() which is picked up by Clion and not CppCheck, it contains an endless recursion and it is a risk due to the bug causing the program to crash. The next error is A is a non-copyable class of variable x which is picked up by Clion and CppCheck, it could lead to undefined behavior if the copy constructor of class A does not initialize the member variable of x. MySpecialType::DontThrow() throws an exception, which leads to unexpected behavior if the exception is not handled, it is a risk and it is picked up by both Clion and CppCheck. Foo() passes a nullptr pointer to \*a, that could lead to a crash or unexpected behavior which is a risk and it is picked up by both Clion and CppCheck. Work\_with\_arrays() attempts to write an out-of-bounds array element that could lead to a crash or unexpected behavior, it is a risk, and it was picked up by both Clion and CppCheck. The function vector\_test potentially invalidates an iterator when it erases an element from the vector items, leading to undefined behavior, which is a risk and it is picked up by both Clion and CppCheck. Assert(z = 2) is an invalid assertion, it does not pose a risk but it is a bug, it is picked up by both Clion and CppCheck. Assert(my\_function() == 3) fails assertion because the function my\_function returns a bool value when its compared to an int value, it is not a risk but it fails to produce the correct output when the program is compiled and ran. While for this assignment I was not able to use Visual Studio, Clion did a really good job of finding the vulnerabilities in the program and displaying them to the same capacity as CppCheck.