Southern New Hampshire University

CS 405 Secure Coding

Module 5 Static Code Analysis

George Harrison

**Warnings and Messages From the Visual Studio Built-in Analysis Tool**

Text

Description automatically generated

**Warnings and errors from CppCheck**

Text

Description automatically generated with medium confidence

When comparing the built-in Analysis Tool provide within Visual Studio to the analysis provided by CppCheck, there are seem to substantial differences in the output of the two tools. While analyzing the output the first thing notable is the amount information provided by the two. The built-in tool compiled with a few messages to include warnings for function usages, exceptions, and variables declarations being initialized. Although the code compiles, when executed it throws an error.

The first warning brings to light the use of DontThrow functions. This function is used but the says that in line 52 of the code is throw and exception which violates best practice. One remedy for this issue would be the use of a try/catch statement. In line 129 the analysis indicated there is an unsafe usage of the “==” logical operation when using a bool type. In line 42 there is a warning pertaining to and uninitialized variable. This can be remedies simply by initializing the variable prior to that line of code executing. In line 66 the analysis indicates a buffer overrun while writing to the variable ‘buf’. The remedy here would be to write code to detect and or prevent buffer overflows from occurring. This is also mentioned in the CppCheck tool. I looks like the count variable conditions needs to be adjusted to fit inside the parameters of the size of the buf[10] array.

The CppCheck Tool seems to be much more in-depth in its analysis of the code. Although many of the same issue represented in the built-in tool, there are a few errors that jump out when going through the CppCheck output. For example, in line 52 instead of a warning, CppCheck indicates an actual error with the “DontThrow” functions. There is also an error with the code in line 87 with an invalid iterator which isn’t in the built-in analysis. It is also nice that CppCheck gives a much better explanation of the issues it found as well as a possible solution to fix the issue in accordance with coding standards.