Program Analysis Project

The tool I decided on using for this project was Cppcheck. It is a static analysis tool that for C and C++ code. Cppcheck detects bugs and undefined behavior and dangerous coding constructs. Its goal is to limit as many false positives as possible. The tool is designed to analyze C/C++ code even when the code is not in its standard syntax.

The software I chose to run Cppcheck on was source C code files for BACnet Protocol Stack library, which provides an application layer, network layer, and MAC layer communication services for multiple platforms. The source code for this library is written in C and designed to be portable across many compilers and architectures, specifically designed for use in embedded BACnet devices. The specific library was version 0.8.6 which contains a total of 62 source C files. This experiment was executed on a Windows 10 Pro platform on an AMD Ryzen 7 2700X 3.70GHz processor and 16GB RAM. In the experiment, the entire source library and their libraries were scanned for errors, warnings, style warnings, portability, performance warnings, and information.

From the scans, it was found that there were 34 warnings and 8 errors that span across 11 of the 62 source C files. Of the 8 errors, 2 were from one file (bvlc.c) and the last 6 were present in another file (bacint.c). In the bvlc.c file, there were two instances were there were potential Array out of bounds errors, where an index table is being accessed through a for loop and that index passes the bounds of the table. In the bacint.c file, the 6 instances of errors are all signed integer overflow errors for an expression. The expression is the same for each error, a variable that’s pointer value is being accessed and changed. This test takes no longer than 11 seconds to execute. The static analysis software was not difficult to use at all.

For the array out of bounds error, I believe that it is being caused because the last value of the incremented index is being used by the analysis tool to assume that it will access this last index, where the last index is actually only used to exit the loop. That is my guess but, I cannot confirm it without dynamically running and monitoring the library itself. And given that it is a library, some code that uses said library must be developed that accesses the file. For the signed integer overflow error, I am not able to verify whether these are false positives or not. More analysis of the code itself would have to be researched before I can determine if it’s a true positive error.

A screenshot of a social media post

Description automatically generated