**Vulnerability Analysis**

Kyle D. Musegades, Jr.

Colorado State University Global

CSC450-1 Programming III

Professor Reginald Haseltine

October 22, 2023

**Vulnerability Analysis**

When manipulating strings in C++, especially in string concatenation, there are many possible vulnerabilities that can occur if improper practice is utilized. Since string concatenation is a form of string manipulation, avoidance of creating bounds through an array is important as it prevents the changes of buffer overflow (Ballman, 2017). Another valuable aspect of not using bounds is that string truncation will not occur with this method. The provided code is also refraining from using many string operations, which keeps it simple and prevents many other errors that occur when using them incorrectly.

**Potential Flaws**

One major issue with not setting any bounds with strings is the issue in which users are able to enter strings of any length. While it is not a huge issue as this program is relatively small, larger programs can run into memory issues if users decided to enter incredibly long string characters. To combat this it would be possible to construct a string with a character limit, however, this may lead to more errors down the line as swapping data types leads the program to be more vulnerable.

**References**

Ballman, A. (2017). *SEI CERT C++ Coding Standard* (2016th ed.). Carnegie Mellon University. October 10, 2023, https://resources.sei.cmu.edu/downloads/secure-coding/assets/sei-cert-cpp-coding-standard-2016-v01.pdf