**Module 2: Buffer Overflow**

Zane Milo Deso

Southern New Hampshire University

CS-405: Secure Coding

**Buffer Overflow Post-Mortem**

**Introduction**

As a Computer Science undergraduate attending Southern New Hampshire University (SNHU), I have been tasked with summarizing the completion of this assignment. In this paper, I will briefly outline buffer overflow, elaborate on my approach to the prevention of buffer overflow, why it works, problems encountered, and how I ameliorated the issues.

**Buffer Overflow**

Buffer overflow occurs when the range of the expected input falls out of the range of the container. More specifically, the input data exceeds the allocated memory for the container and erroneously overwrites into an adjacent memory address. This can lead to unexpected behavior, crashes, errors, etc.

**Approach to Detect and Prevent Buffer Overflow**

At first, I considered a few different angles that a buffer overflow may be prevented and detected. I opted to check the input length in a while loop, only allowing the correct amount of data to escape the while loop, continuing into the rest of the program.

**Why does the Approach work?**

The approach is actionable since the range of acceptable inputs has now been limited to the acceptable range of the allocated memory. This limited range disallows any input that can cause a buffer overflow.

**Problems Encountered & Overcoming the Obstacle**

The only problem that I encountered was when I attempted to use .size() on the input variable. This caused an error to be thrown, which was due to the variable not being a class object. This makes sense, so I fixed the problem by instead setting a new variable to capture the input length, accomplished via strlen().

**Conclusion**

In conclusion, Buffer Overflow is a major threat that if left unchecked may alter the behavior of programs and even alter records in databases if allowed to pass through undetected. With input validation in place and a careful plan to ensure memory allocations are not exceeded, buffer overflow can be avoided, neutralizing its potential damage.