**Module Two – Buffer Overflow**

**Justin Hancock**

**CS-405 Secure Coding**

**09.14.2024**

In this assignment, we were tasked with identifying and solving a buffer overflow problem. The issue occurs when users were able to enter more characters than the program could safely store. This could lead to potential overwriting of important data like the account number in this problem. The goal is to prevent overflow, handle situations where it might occur, and make sure the program stays secure and functional. I first detected the buffer overflow problem by reviewing how the program handled user input. The program used a char array to store the input but did not limit the number of characters a user could input. This created an overflow vulnerability that could alter nearby data in memory. I implemented a safer method for accepting user input by changing the way that the input is read. I restricted the input to 19 characters to make sure that nothing could spill over into unintended areas of memory. This change stops the overflow before it can happen. When the user tries to enter more than 19 characters, the program sends a warning message to let the user know their input is too long and to try again. To prevent misuse of this tool, the user is given 3 opportunities to attempt a valid input before exiting the program. By limiting the input size and giving users a fixed number of attempts, I made sure the program was protected from harmful overflow. It protects both the program’s memory and important data from being altered by excessive input. The retry system provides good balance between security and usability by allowing the user to correct mistakes while preventing potential abuse of the input system.

Screenshots:

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated