2-3 Activity: Buffer Overflow Coding

**Screenshot With Exploit**

**A screenshot of a computer

Description automatically generated**

**Screenshot Without Exploit**

**A screenshot of a computer

Description automatically generated**

**Summary**

In this assignment, we are supplied with code that is susceptible to buffer overflow exploits. Specifically, users can enter input beyond 20 characters and overflow the buffer and the account\_number variable can potentially be overwritten. To address this, I implemented a form of input validation that limits the maximum number of characters that users can input. I also added a null terminator to ensure that the user\_input string is properly terminated. All of the changes that were made to the program reside within the main() function.

Starting with the first change, the user\_input variable was changed to hold 21 characters to account for the null terminator added to the string. Since the null terminator occupies an index within the array for user\_input, this was changed to 21 to account for that. Next, user input is read into the user\_input variable with the setw(20) function to limit the input width to 20 characters. This change is to address the issue with buffer overflow within the program with input beyond 20 characters. The last change made was adding the null terminator to the user\_input array to properly terminate the string.