**Buffer Overflows**

James Porter

Department of Computer Science, Southern New Hampshire University

CS 405 Secure Coding

Professor Wasim Alim

September 11, 2022

**Less Than 20 Chars**

**Text

Description automatically generated**

**Just Greater Than 20 Chars**

**Text

Description automatically generated**

**Vastly Greater Than 20 Chars**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Summery**

This was a problem that I thought was going to be super easy to figure out due to some of my previous encounters with this issue, but I was wrong. To start, I tried using the cin.ignore() line to ignore all the characters after 20, but it created some unwanted side effects. Not only did it lead to the input not always being right, but I also ended up crashing the system a couple of times.

This led me to my next idea to write the user input to a temporary string that I would then write to the user\_input variable. This worked…somewhat. I was able to get input from the user and write it to the variable successfully, but in doing so, I ended up with extra characters at the end of my user\_input array. Like I said, this worked, but it is not practical if the user is meant to be able to interpret the data.

Finally, I went to the amazing world of Google to find a solution that might work. I stumbled across the amazing getline() function. In using this function, I was able to choose where the input was being recorded to, how many characters are read from the string, and even a delimiter. This was my magic ticket, and it gave me the solution that I needed to get this right. As a side note for future development, the 20 characters being read could be limited to 19 to leave an empty spot at the end of the array created at the start of the file.