For this activity I started by defining the max number of characters allowed. I then changed the user\_input variable to use num\_of\_chars. To prevent buffer overflow, I used information found on (IncludeHelp, 2023) to use the cin.getline() function so that only a specified number of characters will be written to the variable given. Again, I used num\_of\_chars to specify the maximum number of characters allowed. I then used the cin.fails() function to determine if more than 20 characters are being passed to user\_input, if so, it will display a message about the possibility of buffer overflow. The screenshot below shows my manual input and both python scripts being run, for the script that exploits buffer overflow it can be seen the logic built into the application prevents the overflow.

|  |
| --- |
| Text  Description automatically generated |

# Bibliography

IncludeHelp. (2023, Mar 01). *C++ program to read string using cin.getline()* . Retrieved from includehelp.com: https://www.includehelp.com/cpp-programs/cpp-program-to-read-string-using-cin-getline.aspx