Edwin Jones

CS-405: Secure Coding

May 18, 2025

Process Summary

To detect and prevent a buffer overflow for this code, I used std::cin.getline(user\_input, sizeof(user\_input)) to limit the input to the size of the user\_input buffer. This ensures that no more than nineteen characters (plus the null terminator) are stored, preventing overflow.

Next, I implemented a check using std::cin.fail() to detect if the user entered more characters than the buffer can hold. If a failure is detected, I called std::cin.clear() to reset the input stream’s error flags and then used std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n') to flush the remaining characters from the input buffer.

If an overflow is detected, the program prints an error message to std::cerr and exits with a non-zero status code (return 1), preventing any further processing of invalid input.

