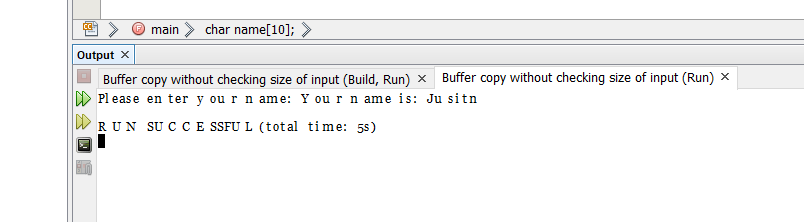
Buffer copy without checking size of input

Also known as buffer overflow. A buffer overflow condition occurs when the program attempts to put more data into a buffer than it can hold. Buffer copy without checking size of input is when the program copies an input buffer to an output buffer without checking to make sure the size of the input is smaller than the size of the output. The buffer simply cannot hold the amount of data.

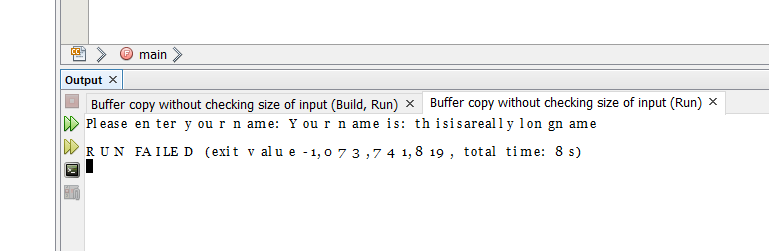
One of the simplest and most common is “classic buffer overflow.” This is usually when the programmer has not considered security in the development of their program. The program copies the buffer *without checking its length* at all. Buffer overflows can be used to execute arbitrary code and often result in crashes.

Vulnerability-

For our program we have an array buffer with the size of [10]. When we ask for the user’s name and they enter less than 10 character we get a successful execution of the program.



But let’s say that some people have a longer name than 10 characters. We get an overflow error.

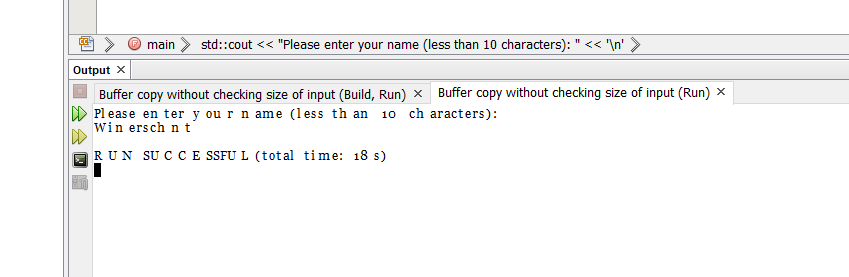


Mitigation-

For the mitigation I have limited the user’s input to 10 characters using the following line of code:

std::cin.getline(UserType, 11); // limit input to 10

When our application asks the user to type in a name, it now automatically cuts the user off at 10 characters. So now let’s try to type in a name that is more than 10 characters. Let’s do the same name from the previous example for constancy’s sake(Winerschntizelhause).



Once we reach the maximum length, the program moves forward. In this case it continues the code and prints the user input to the screen. W i n e r s c h n t

1 2 3 4 5 6 7 8 9 10

By checking the size of input, we have eliminated the possibility of buffer overflow.