* **Extra credit quiz**

const long MAX\_LEN = 20,000;

const buf[MAX\_LEN];

if (len < MAX\_LEN) {

strcpy(buf, input);

}

Can a buffer overflow attack occur? If so, how long does input need to be?

**To cause a buffer overflow, the length of input needs to be larger than the maximum value that can be stored in a short int (ranger of -32768 to 32767); therefore, the length needs to be at least 32768. However, not any value that overflows the short int will work, as the resulting value also needs to be less than 20,000 (MAX\_LEN) when converted to a short int.**

***Lower bound: 32768 (one more than the maximum value for a short int)***

strlen of 32768 to short int

32768 in binary -> 1000 0000 0000 0000

Since the first digit is a 1, the resulting number will be negative.

2’s compliment -> 0111 1111 1111 1111

+1

->1000 0000 0000 0000

This translates to 32,768 and with the original first byte negative, a short int value of -32,768

-32,768 is less than MAX\_LEN, so this will overflow the buffer.

**Upper bound: 85535 (19,999 plus the ranger of the maximum value for a short int)**

strlen of 85535 to short int:

85535 in binary -> 0001 0100 1110 0001 1111

Since it is too large to fit in a short int, reject the high byte

-> 0100 1100 0001 1111

This translates to 19,999. Since this is less than MAX\_LEN (20,000), the buffer will overflow.

**Hence the range of values for the length of input that will cause a buffer overflow are 32768 to 85535.**