CSEC-201 PRACTICAL FINAL EXAM	DUE: 12/12/2022 8:00 PM ET

QUESTIONS

1. What is the minimum amount of input to the INC command needed to provoke a crash?

The minimum amount of input I needed was 23

2. What is the amount of input to the INC command needed to overwrite the saved instruction pointer?

To completely overwrite the EIP register, we need an input of 32.
For Example:

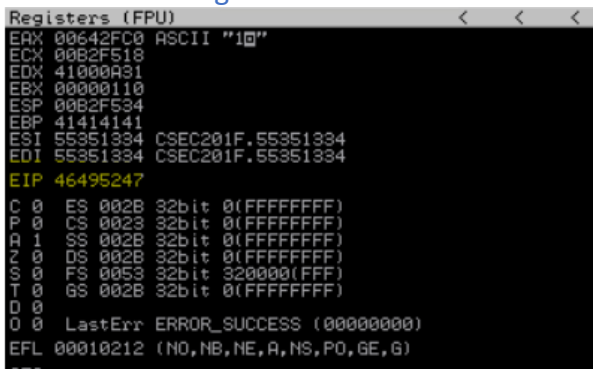
If our string is ”A”*28 + “BBBB” -> EIP = “42424242”

If our string is “A”*27 + “BBBB” -> EIP = “00424242”

So we need a garbage input of 28, and then we need 4 input of the EIP we want to overwrite.
Equaling a total of 32 length input to completely overwrite the EIP.

3. Provide a screenshot of Immunity Debugger demonstrating you were able to overflow the buffer using the INC command and overwrite the saved instruction pointer with the first four characters of your name.

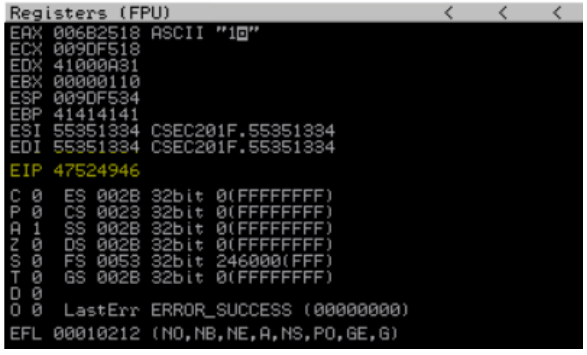
After Overwriting the EIP with “GRIF”:



46 = “F” 49 = “I” 52 = “R” 47 = “G”

(EIP overwrites backwards?)

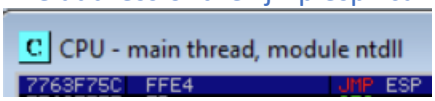
After Overwriting the EIP with “FIRG”:



47 = “G” 52 = “R” 49 = “I” 46 = “F”

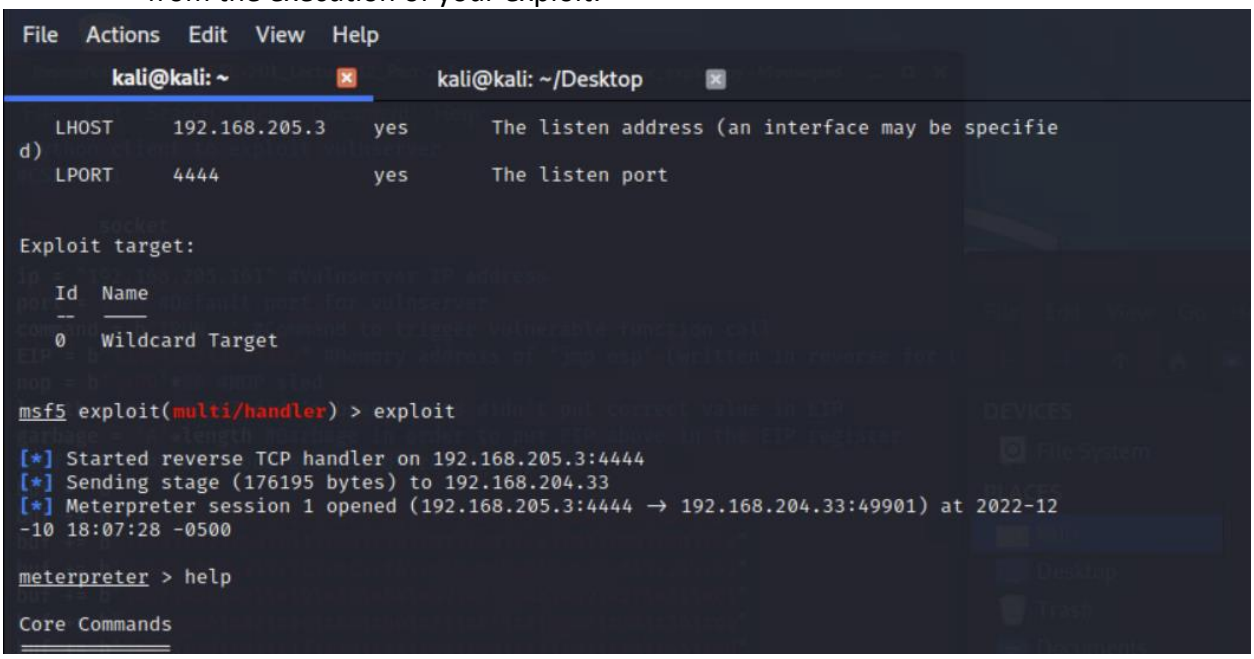
4. What is the address of the instruction which can be used to redirect execution to the stack?

The address of the “jmp esp” call is 7763F75C.



5. Provide a working exploit for the INC buffer overflow. (No response needed here. Include a Python script in your GitHub Classroom submission.)

6. Provide a screenshot demonstrating a successful Meterpreter connection resulting from the execution of your exploit.



7. Which line in the source code in the INC function is responsible for the overflow? How could that one line be rewritten to prevent the overflow?

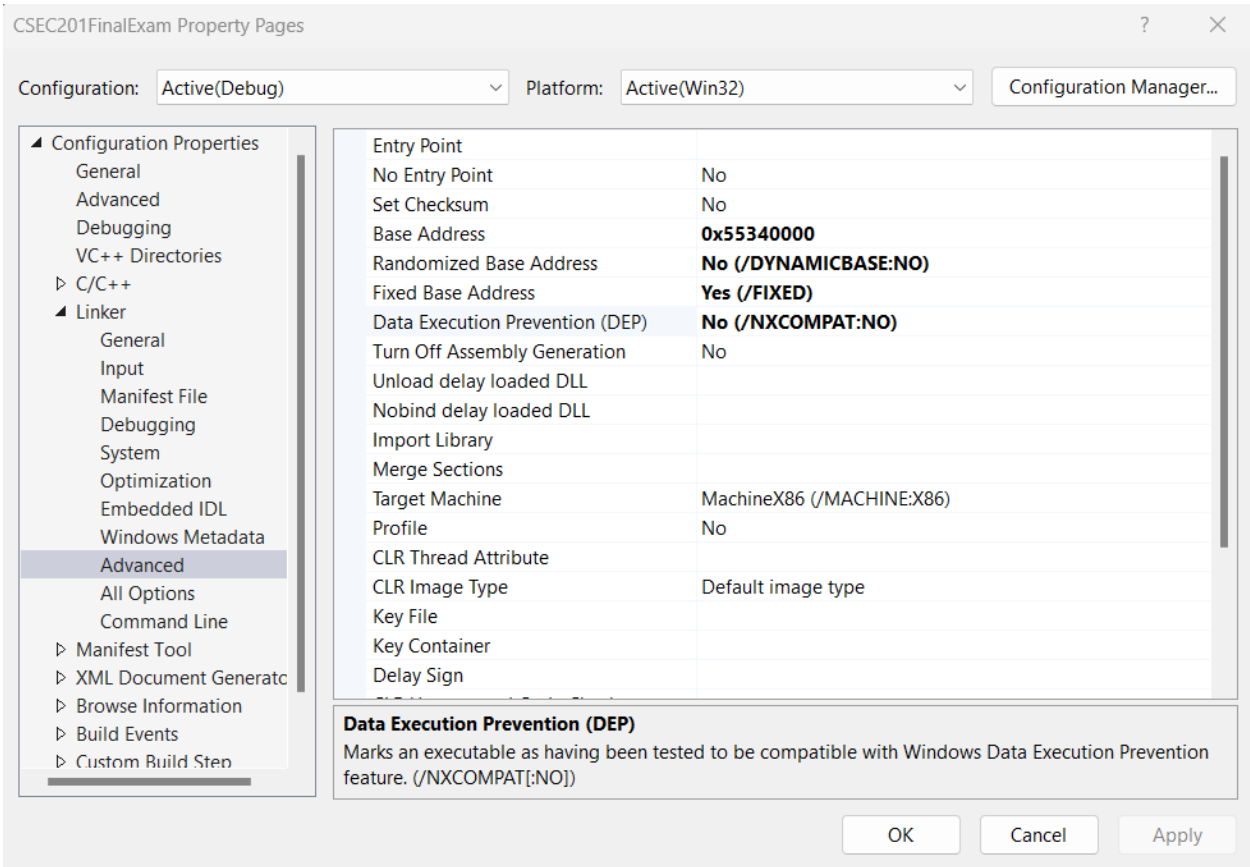
The line responsible for the overflow is line 117 inside the inc() function:

```
117     sprintf(newval, "%s", val);
```

This is where we put our val variable into a buffer that is too small.
To prevent this overflow, we can instead use snprintf(), which would allow us to limit the size of what we are storing into the buffer. When we limit the size, we will never encounter an overflow, no matter what the user inputs.

8. Search the Visual Studio project configuration settings. Determine if the binary is being compiled with DEP and ASLR enabled. If it is not, document what the current settings are with screenshots, and describe what the settings should be to enable them.

No, DEP and ASLR were not enabled.



To enable DEP, we would click the dropdown tab in the “Data Execution Prevention (DEP)” row, and we would select the “Yes (/NXCOMPAT)” option, which would turn on DEP.

To enable ASLR, we would click the dropdown tab in the “Randomized Base Address” row, and we would select the “Yes (/DYNAMICBASE)” option, which would turn on ASLR.