Stack Overflow

Stack Overflow attack is an that take advantage of the overflow error that occurs when a program tries to use more memory space in the call stack than that has been stored to that stack. By using this we can reroute the return to the main to the hacked function instead which outputs the message ‘Hacked by Mayur Suresh!!!’ when the segmentation occurs.

To implement this attack, we are going to take advantage of the memory overflow. First we need to edit the lab2.c program to make changes to the hacked function where we set it to return ‘Hack by Mayur Suresh!!!’. Then we need to go into the return\_input function to change the array sizer to the last 2 digits of your student ID as shown in Image 2. We then trigger the stack overflow to test as shown in Image 1. By doing this we note and input more data than the array can handle and when this happens the data will be overflowed and move to the next register, when it moves to the next register, we will have the following register store the memory address of the hacked() function. Because the register now contains the beginning memory address of hacked() it will go there instead of returning to main() function which was initially stored in the same register.

To start this attack, we need to strip certain layers of protection when run (gcc). We need to strip the Address Space Layer Randomization (ASLR) (not needed for this lab), Data Execution Prevention (DEP), Stack Protector and Precision Independent Encoding (PIE) as shown in Image 3. After finding the magic number which is 105 you would need to put it in a file, along with this we need the starting memory address for hacked, which is found in its stack shown in Image 4 below which is 0x08049172. We need the ‘Magic Number’ and the memory address to specify when the hacked function can come in and activate, we do this by adding the memory address to the end of the Magic Number of characters in the hexadecimal form of the little endian(\x82\x91\x04\x07). The string will look like the following: A\*105+\x82\x91\x04\x07 like shown in Image 5. With this we can activate the hacked function in the lab2.c program by rerouting the return function from the main to the hacked function.

By running the exploit.py we indeed see that the Stack Overflow attack has been implemented in Image 6. By doing this we see how easily this attack can be implemented by just making use of the defined array size and redirect the return\_input function to the hacked function instead of the main function.

# Source Images:

A picture containing background pattern

Description automatically generated

Image 1

Text

Description automatically generated

Image 2

Text

Description automatically generated

Image 3

Text

Description automatically generatedText

Description automatically generated

Image 4 Image 5

Text

Description automatically generated

Image 6