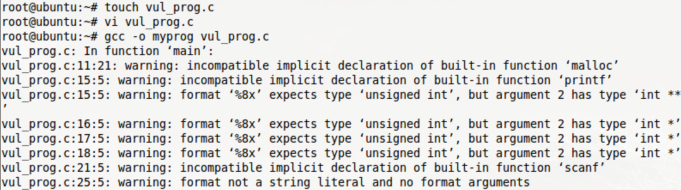
Lab 2 Format String Vulnerability Lab

Yukui Ye

SUID: 439644268

**Task 1: Exploit the vulnerability**

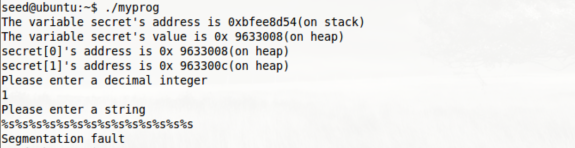
creat a file named vul\_prog.c, compile it, we can ignore the warning parts.



and make the program Set\_UID program



*a: Crash the program*

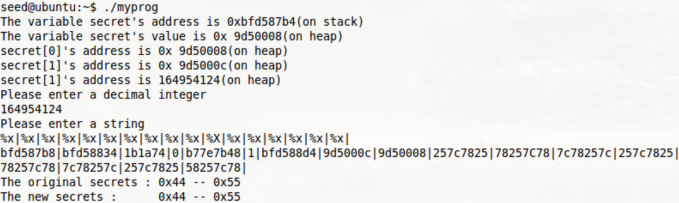
To crash the program , we need to make pointer points to an invalid place. Thus we input enough ”%s” till it reaches “\0”. 

*b: Print out the secret[1]value*

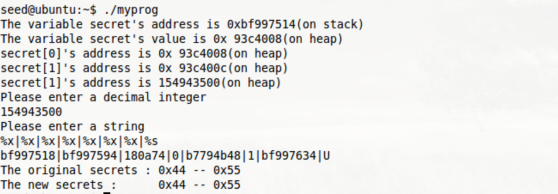
To read the secret[1]value, we need to put the address of secret[1] into stack since the original value of secret[1] is stored in heap. Therefore, we need to add one line code in the original code file to print out the address of secret[1] which is in decimal formal. Then we use int\_input to store the address of secret[1] into the stack. After all the prepared work done, we need to use “%x” to move the pointer to that the place where store the address of secret[1]; and print it out.

Add one line to get the address of secret[1] in decimal formal



put the address of secret[1] on the stack by using int\_input ; then calculate the secret[1]address is stored in the 8th %x place. 

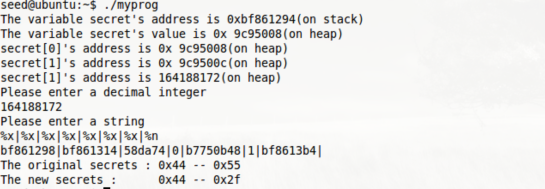
read the value of secret[1] by put %s after the 7th %x, The 8th block store the address of secret[1], by using %s, the pointer exactly point to the address of secret[1] to get the value.



The result is “U” which is 0x55 in ASCII.

*c: Modify the secret[1] value.*

Depend on the previous task, we already known the 8th %x store the address of secret[1], all we need to do is change the “%s” to “%n”. “%n” will write the number of the string into a pointer. Then we could set the value of secret[1] as the number of the string that has been written.



we can easily tell that the value of secret[1] has been changed from 0x55 to 0x2f.

*d: Modify the secret[1] value to a pre-determined value*