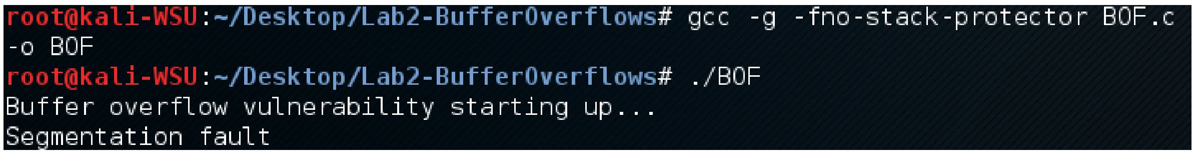
Lab2 Answer

11911409 孙永康

a

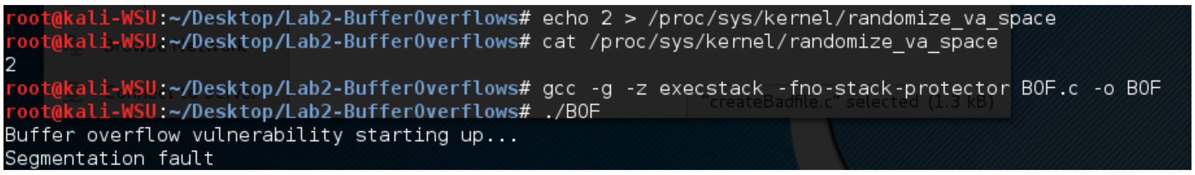
if I run without "-z execstack", it will report a **Segmentation Fault** :



```
root@kali-WSU:~/Desktop/Lab2-BufferOverflows# gcc -g -fno-stack-protector BOF.c
-o BOF
root@kali-WSU:~/Desktop/Lab2-BufferOverflows# ./BOF
Buffer overflow vulnerability starting up...
Segmentation fault
```

b

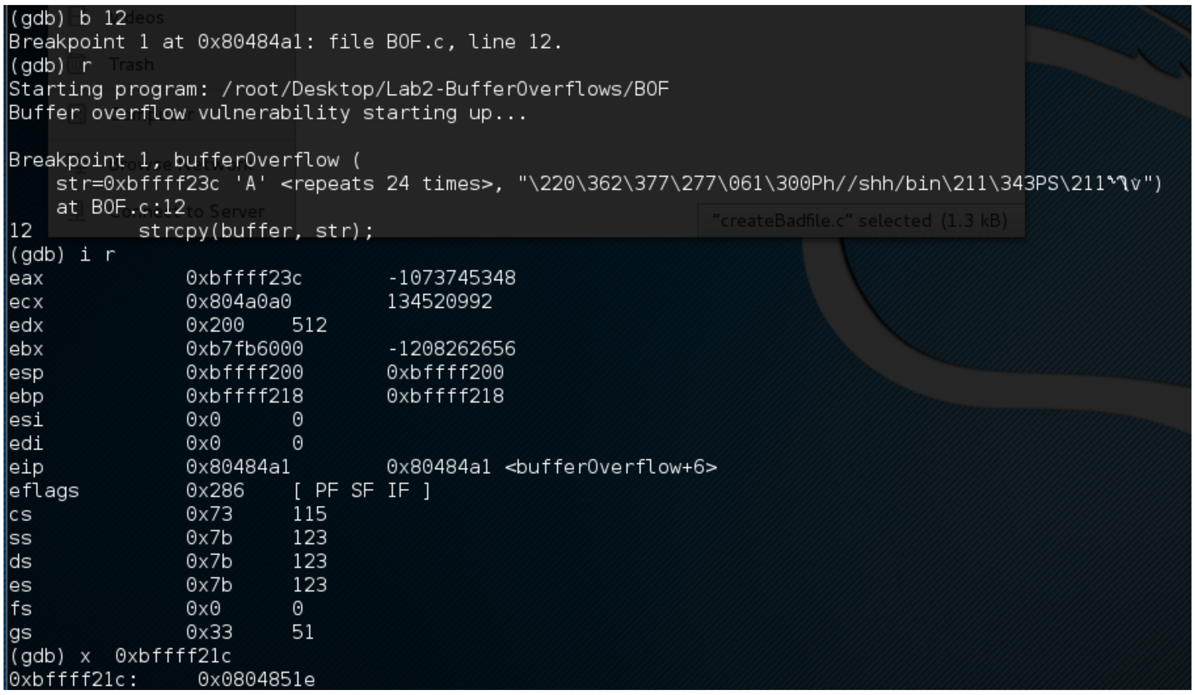
Lets first close the ASLR. and run our code. Then we will receive a **Segmentation Fault** :



```
root@kali-WSU:~/Desktop/Lab2-BufferOverflows# echo 2 > /proc/sys/kernel/randomize_va_space
root@kali-WSU:~/Desktop/Lab2-BufferOverflows# cat /proc/sys/kernel/randomize_va_space
2
root@kali-WSU:~/Desktop/Lab2-BufferOverflows# gcc -g -z execstack -fno-stack-protector BOF.c -o BOF
root@kali-WSU:~/Desktop/Lab2-BufferOverflows# ./BOF
Buffer overflow vulnerability starting up...
Segmentation fault
```

Then lets check the return address in **GDB** :

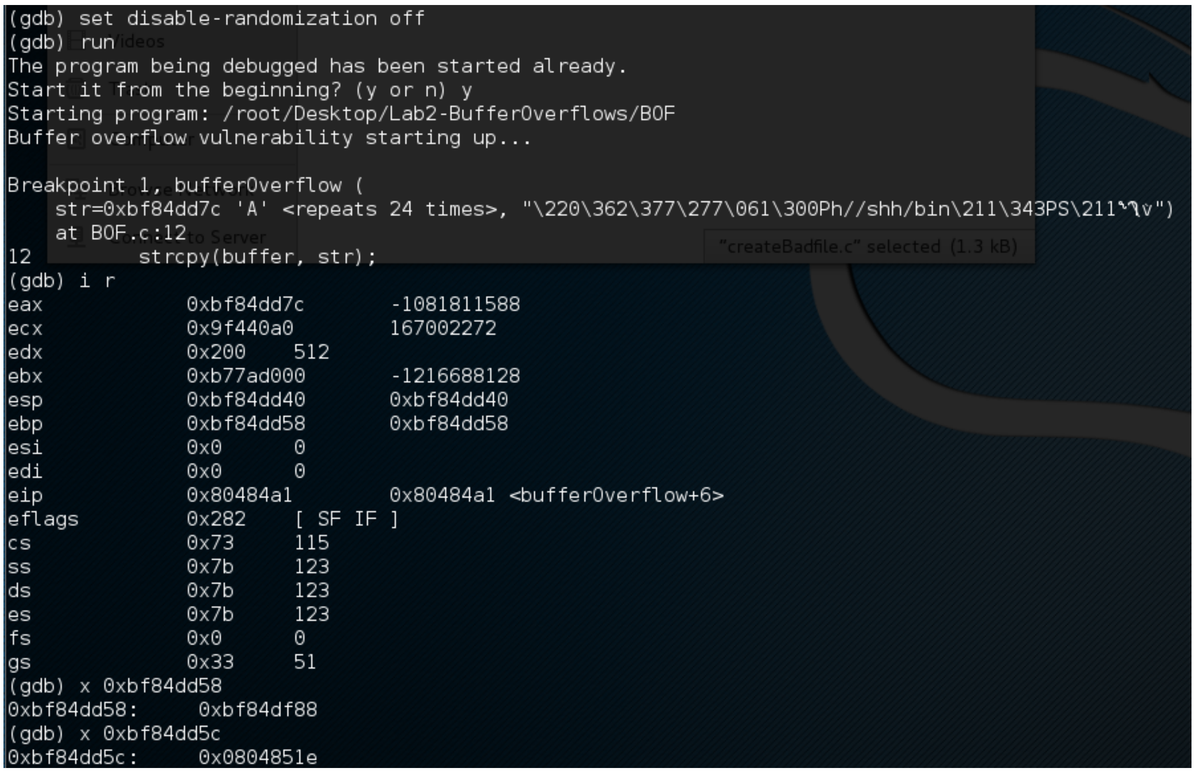
The **GDB** 's ASLR is turn off at first. As we can see that the return address is in **($ebp + 4)**, which is 0x0804851e :



```
(gdb) b 12
Breakpoint 1 at 0x80484a1: file BOF.c, line 12.
(gdb) r
Starting program: /root/Desktop/Lab2-BufferOverflows/BOF
Buffer overflow vulnerability starting up...

Breakpoint 1, bufferOverflow (
    str=0xbffff23c 'A' <repeats 24 times>, "\220\362\377\277\061\300Ph//shh/bin\211\343PS\211[illegible]\v")
    at BOF.c:12
12          strcpy(buffer, str);
(gdb) i r
eax            0xbffff23c       -1073745348
ecx            0x804a0a0        134520992
edx            0x200    512
ebx            0xb7fb6000       -1208262656
esp            0xbffff200       0xbffff200
ebp            0xbffff218       0xbffff218
esi            0x0      0
edi            0x0      0
eip            0x80484a1        0x80484a1 <bufferOverflow+6>
eflags         0x286    [ PF SF IF ]
cs             0x73     115
ss             0x7b     123
ds             0x7b     123
es             0x7b     123
fs             0x0      0
gs             0x33     51
(gdb) x  0xbffff21c
0xbffff21c:     0x0804851e
```

Then we turn on **GDB** 's ASLR. Then we can see that the return address is still the same:

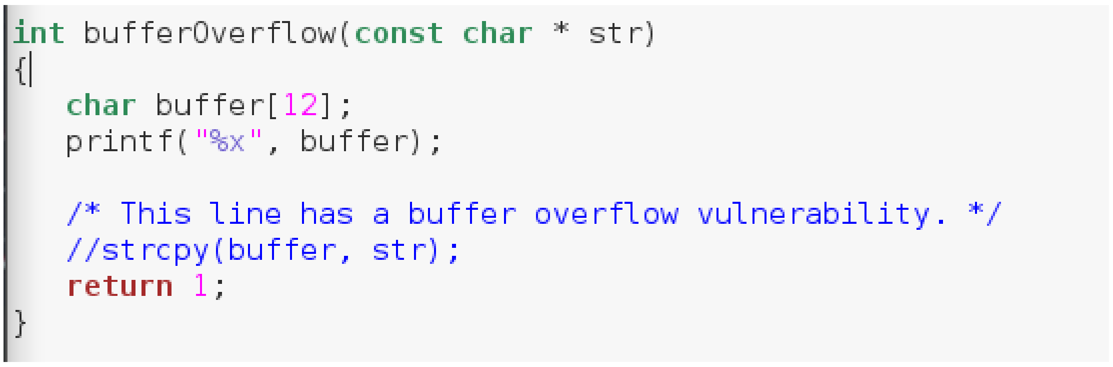

```
(gdb) set disable-randomization off
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /root/Desktop/Lab2-BufferOverflows/BOF
Buffer overflow vulnerability starting up...

Breakpoint 1, bufferOverflow (
    str=0xbf84dd7c 'A' <repeats 24 times>, "\220\362\377\277\061\300Ph//shh/bin\211\343PS\211[illegible]\v")
    at BOF.c:12
12	      strcpy(buffer, str);
(gdb) i r
eax            0xbf84dd7c       -1081811588
ecx            0x9f440a0        167002272
edx            0x200    512
ebx            0xb77ad000       -1216688128
esp            0xbf84dd40       0xbf84dd40
ebp            0xbf84dd58       0xbf84dd58
esi            0x0      0
edi            0x0      0
eip            0x80484a1        0x80484a1 <bufferOverflow+6>
eflags         0x282    [ SF IF ]
cs             0x73     115
ss             0x7b     123
ds             0x7b     123
es             0x7b     123
fs             0x0      0
gs             0x33     51
(gdb) x 0xbf84dd58
0xbf84dd58:     0xbf84df88
(gdb) x 0xbf84dd5c
0xbf84dd5c:     0x0804851e
```

C

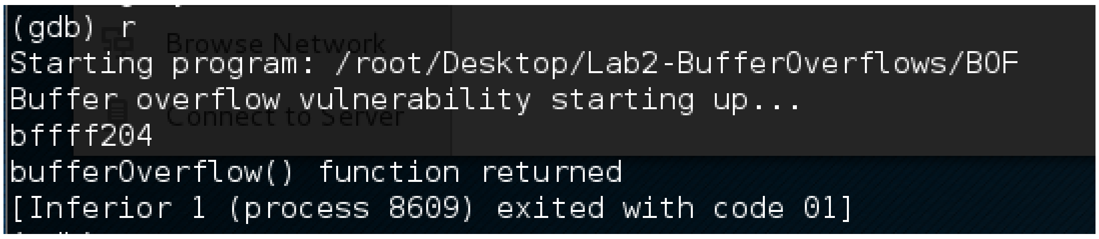
Yes, It will change. Because when I finish my code in GDB, I found it become wrong running through **./BOF** :

So, Let us do an experiments. first we shall change some codes in **BOF.c** in order to print out the buffer's address :

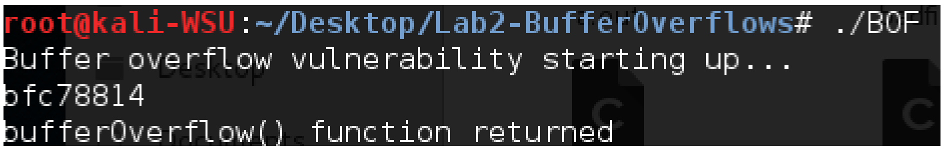

```c
int bufferOverflow(const char * str)
{
    char buffer[12];
    printf("%x", buffer);

    /* This line has a buffer overflow vulnerability. */
    //strcpy(buffer, str);
    return 1;
}
```

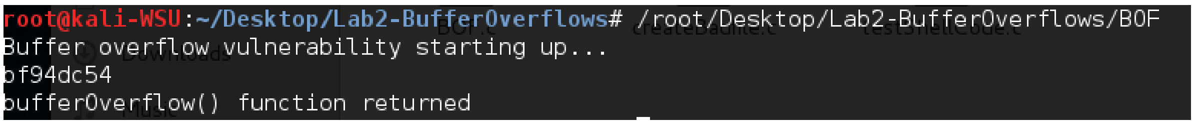
Then run it in **GDB**, we got 0xbffff204 :


```
(gdb) r
Starting program: /root/Desktop/Lab2-BufferOverflows/BOF
Buffer overflow vulnerability starting up...
bffff204
bufferOverflow() function returned
[Inferior 1 (process 8609) exited with code 01]
```

Then run it through **./BOF**, it is changed to 0xbfc78814 :


```
root@kali-WSU:~/Desktop/Lab2-BufferOverflows# ./BOF
Buffer overflow vulnerability starting up...
bfc78814
bufferOverflow() function returned
```

Finally run it through **/home/root/Desktop/Lab2-BufferOverflows/BOF**, changed again :



```
root@kali-WSU:~/Desktop/Lab2-BufferOverflows# /root/Desktop/Lab2-BufferOverflows/BOF
Buffer overflow vulnerability starting up...
bf94dc54
bufferOverflow() function returned
```