ctfshow baby杯 六一快乐 部分MISC WriteUp

原创

是Mumuzi 于 2021-06-02 19:39:21 发布 879 收藏 3

分类专栏： ctf ctfshow 文章标签： 信息安全

本文链接：https://blog.csdn.net/qq_42880719/article/details/117479024

ctf 同时被 2 个专栏收录

75 篇文章 28 订阅

订阅专栏

ctfshow

23 篇文章 8 订阅

订阅专栏

baby baby

babyLSB

首解：zsteg+reverse+猜测

```
b8,b,msb,XY,prime       ..  text: "nn~~~}~~"
b8,rgb,msb,XY,prime     ..  text: ".#9>CY^="
b1,g,msb,YX             ..  text: "LoaYuBeMnehSieW{wohsftc\n.htaeD ot nwonknU\n.sedalb dnasuo
ht a revo detaerc evah I\n.do"
```

输入文本信息：

LoaYuBeMnehSieW{wohsftc\n.htaeD ot nwonknU\n.sedalb dnasuoht a revo detaerc evah I\n.do

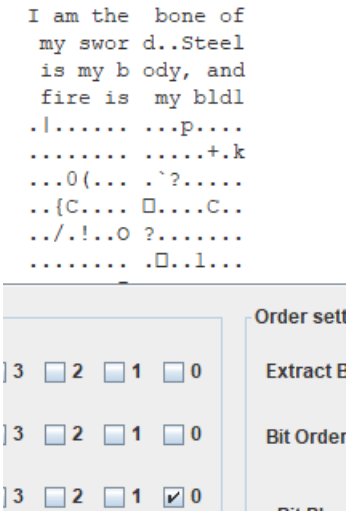
文本字符串的逆序：

od.n\I have created over a thousand blades.n\Unknown to Death.n\ctfshow{WeiShenMeBuYaoL

然后猜了个LSB

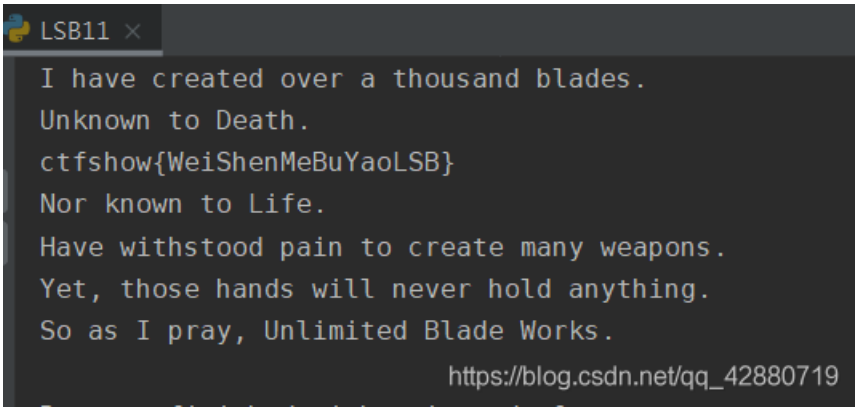
> ctfshow{WeiShenMeBuYaoLSB}

预期：
因为用stegsolve的时候g通道发现有内容



就怀疑是围了一圈，写个脚本：

```
from PIL import Image
img = Image.open("flag1.png")
w,h = img.size[0],img.size[1]
t1=t2=t3=t4=flag = ''
for i in range(w):
    p = img.getpixel((i,0))
    t1 += bin(p[1])[-1]
for i in range(h):
    p = img.getpixel((w-1,i))
    t2 += bin(p[1])[-1]
for i in range(w):
    p = img.getpixel((i,h-1))
    t3 += bin(p[1])[-1]
for i in range(h):
    p = img.getpixel((0,i))
    t4 += bin(p[1])[-1]
flag = t1[:-1]+t2[:-1]+t3[::-1][:-1]+t4[::-1][:-1]
s = ''
rflag = ''
for i in flag:
    s+=i
    if len(s)==8:
        rflag += chr(int(s,2))
        s=''
print(rflag)
```



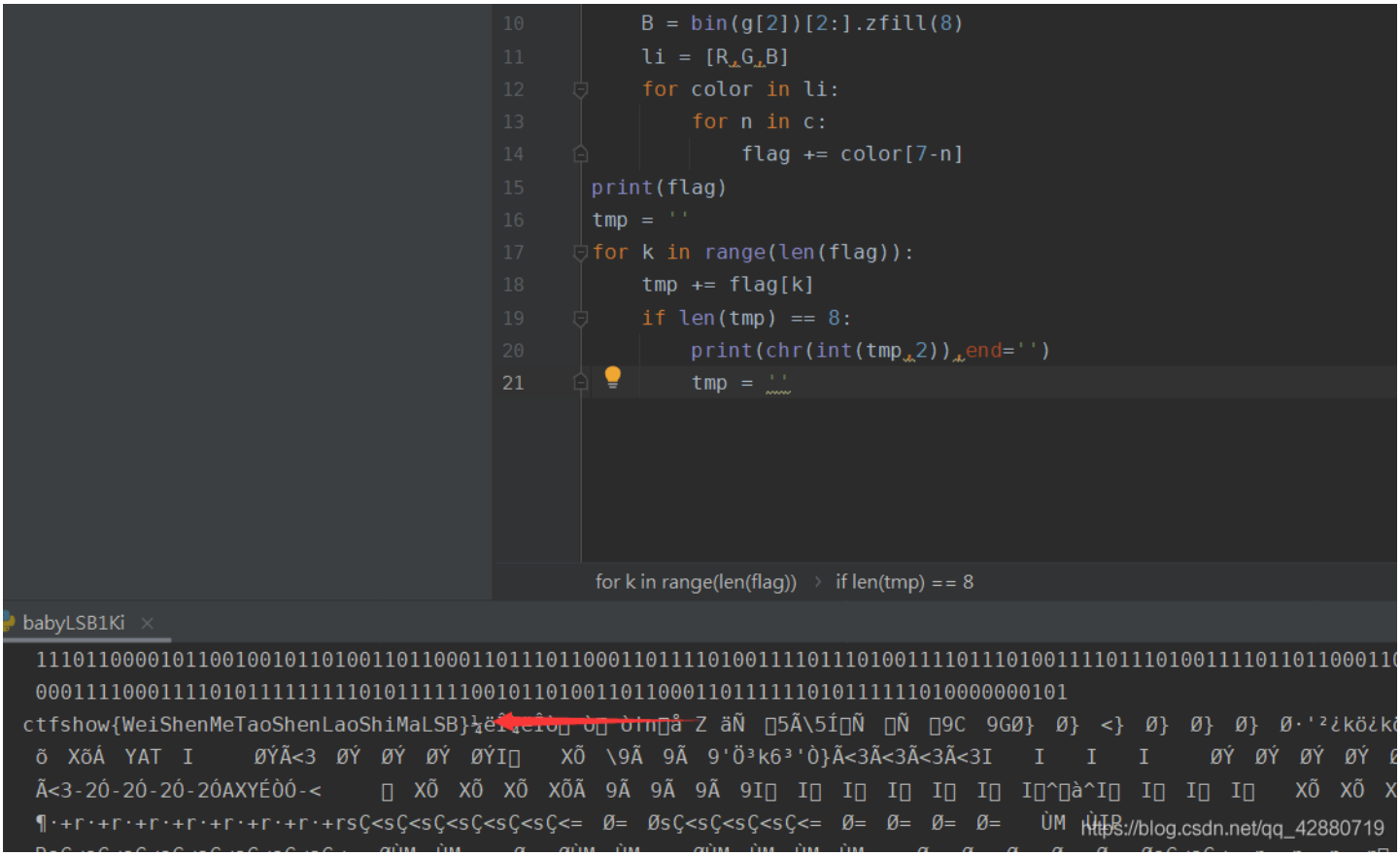
babyLSB1Ki

这里提示了1kb=1024b，然后题目说通道是RGB通道。

就需要写个脚本，按照1,0,2,4通道来读。

一开始一直以为是读bit位的RGB。最后才发现是读RGB的bit位…

```
from PIL import Image
pic = Image.open("flag2.png")
w,h = pic.size[0],pic.size[1]
flag = ''
c = [1,0,2,4]
for i in range(w):
    g = pic.getpixel((i,0))
    R = bin(g[0])[2:].zfill(8)
    G = bin(g[1])[2:].zfill(8)
    B = bin(g[2])[2:].zfill(8)
    li = [R,G,B]
    for color in li:
        for n in c:
            flag += color[7-n]
print(flag)
tmp = ''
for k in range(len(flag)):
    tmp += flag[k]
    if len(tmp) == 8:
        print(chr(int(tmp,2)),end='')
        tmp = ''
```



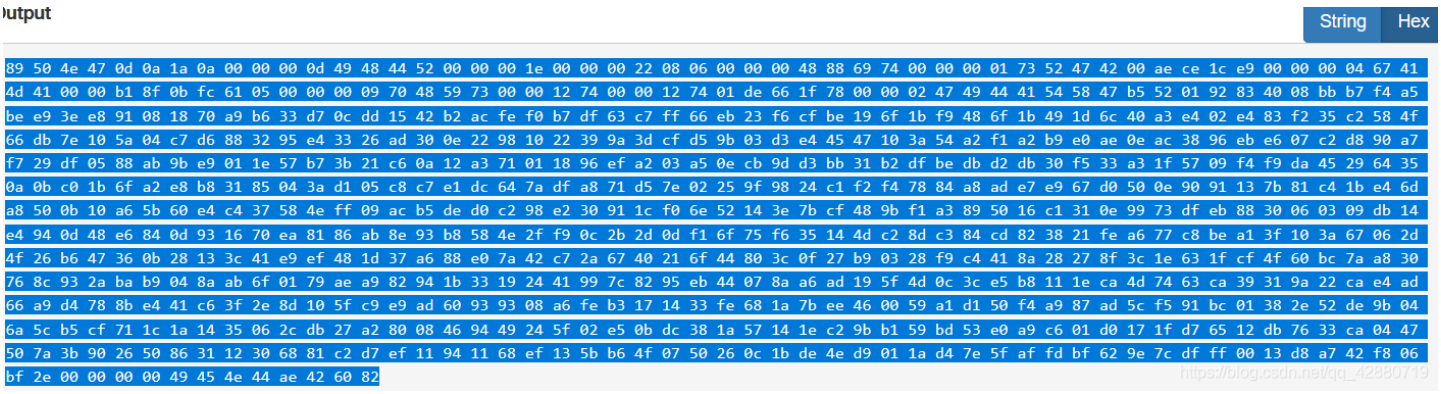
ctfshow{WeiShenMeTaoShenLaoShiMaLSB}

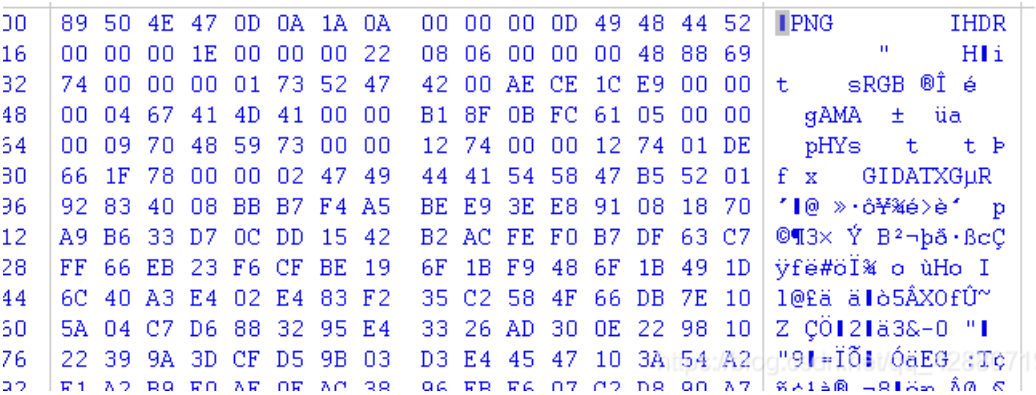
babyLSBwithHelicopter

这道题是靠8神给的hint：braincopter才出的

```
.\bftools.exe decode braincopter flag.png --output flag1.txt
```

https://gkucmierz.github.io/brainfuck-interpreter/





得到ffflag.png，是imagefuck，写脚本(脚本有注释)

```python
from PIL import Image
pic = Image.open("ffflag.png")
w,h = pic.size[0],pic.size[1]
print(w,h)
s = ['']*34*30
f=0
for i in range(h):
    for j in range(w):
        s[f] = pic.getpixel((j,i))
        f += 1
print(s)
count = {}
for item in s:
    count[item] = count.get(item, 0) + 1
print(count)#计算出每种RGB的出现的次数
#{(0, 0, 0, 0): 1, (0, 255, 0, 255): 427, (255, 255, 0, 255): 38, (0, 128, 0, 255): 211, (255, 0, 0, 255): 76, (
128, 0, 0, 255): 76, (128, 128, 0, 255): 38, (0, 255, 255, 255): 34, (0, 128, 128, 255): 32, (0, 0, 255, 255): 6
4, (178, 34, 34, 255): 23}
#其中[]和<>应该相等，去生成一个ctfshow{之后发现规律
print(s[2])
flag = ""
t = 0
for i in range(h):
    for j in range(w):
        if(i%2==0):#一排排扫过去发现不对，对比之后发现应该是S型，只需要判断高度为单双数即可
            s = pic.getpixel((j,i))
        else:
            s = pic.getpixel((29-j,i))
        if(s==(0, 255, 0, 255)):
            flag += '+'
            t += 1
        if (s == (255, 255, 0, 255)):
            flag += '['
            t += 1
        if (s == (0, 128, 0, 255)):
            flag += '-'
            t += 1
        if (s == (255, 0, 0, 255)):
            flag += '>'
            t += 1
        if (s == (128, 0, 0, 255)):
            flag += '<'
            t += 1
        if (s == (128, 128, 0, 255)):
            flag += ']'
            t += 1
        if (s == (0, 0, 255, 255)):
            flag += '.'
            t += 1
print(flag)
```

```
+++++++++[->+++++++++<]>++++++++++++++++++.<++++[->++++<]>+.<+++[->—<]>-----.<+++[->+++<]>++++.<+++[->—
<]>–.+++++++.++++++++.++++.<+++++++[->-------<]>---------.<++++++[->++++++<]>++++++.++++.<++++[->++++<]>+.<+++[->—<]>-----.-.
<+++[->+++<]>++++.<++++[->----<]>—.+++++.<+++[->+++<]>++.-.<+++++++++[->--------<]>-------.<+++++++++[->+++++++++
<]>++++++++++++++.<++++[->----<]>-----.<+++[->+++<]>++++++.----------…-.-----.<++++[->++++<]>++++.<+++[->—<]>-----.<++++[->++++
<]>+++.<+++++[->-----<]>.<+++[->—<]>–.<+++[->+++<]>++++.<+++[->+++<]>+++++.++++.<+++[->—<]>–.—.+++++++++.<+++[->—<]>-----
-.+++++++.<+++[->+++<]>++++++.<++++[->----<]>–.++++++++.++++++++.<++++[->----<]>----.+++++++++.+.<+++[->+++<]>+.<++++[->----
<]>----.+++.<++++[->++++<]>.<++++[->----<]>-.++++++++.+++++.<+++[->—<]>------.++++++.<++++[->++++<]>+.<++++[->----<]>-.<+++[-
>+++<]>++++.+++++++.<++++[->----<]>-----.—.<++++[->++++<]>++++++++.++++.<
```

```
1 +++++++++[->+++++++++<]>++++++++++++++++++.<++++[->++++<]>+.<+++[->---<]>-----.<+++[->+++<]>++++.<+++[->---
  <]>--.+++++++.++++++++.++++.<++++++++[->--------<]>----------.<++++++[->++++++<]>++++++.++++.<++++[->++++<]>+.<+++[->---<]>-----.-.
  <+++[->+++<]>++++.<++++[->-----<]>---.+++++.<+++[->+++<]>++.-.<+++++++++[->---------<]>--------.<+++++++++[->+++++++++<]>++++++++++++++.
  <++++[->----<]>-----.<+++[->+++<]>++++++.----------..-.-----.<++++[->++++<]>++++.<+++[->---<]>-----.<++++[->++++<]>+++.<+++++
  [->-----<]>.<+++[->---<]>--.<+++[->+++<]>++++.<+++[->+++<]>+++++.++++.<+++[->---<]>--.---.+++++++++.<+++[->---<]>------.+++++++.
  <+++[->+++<]>++++++.<++++[->-----<]>--.++++++++.++++++++.<++++[->-----<]>----.+++++++++.+.<+++[->+++<]>+.<++++[->-----<]>----.+++.
  <++++[->++++<]>.<++++[->-----<]>-.++++++++.+++++.<+++[->---<]>------.++++++.<++++[->++++<]>+.<++++[->-----<]>-.<+++[->+++
  <]>++++.+++++++.<++++[->-----<]>-----.---.<++++[->++++<]>++++++++.++++.<
```

Input

Clean Ru

Output

String He

```
ctfshow{A_ctfer_don't_need_sex_Taoshen_fucks_his_brain_everyday}
```


ctfshow{A_ctfer_don’t_need_sex_Taoshen_fucks_his_brain_everyday}

天书奇谭PLUS-misc·又一个签到题

7□yyds
用的就是初代脚本
https://7herightp4th.top/blog/CTF/Misc/CTFShow%E5%88%B7%E9%A2%98%E7%AC%94%E8%AE%B0/

```python
import cv2
import os
from tqdm import tqdm
import hashlib
dirpath = path#根目录
dirs_path = dirpath + r"\output" #字条单个字的图片所在文件夹
source = cv2.imread(dirpath + r"\demo.png")#碑文图片
li = []
def match(temp_file):
    template = cv2.imread(temp_file)
    result = cv2.matchTemplate(source, template, cv2.TM_SQDIFF_NORMED)
    cv2.normalize(result, result, 0, 1, cv2.NORM_MINMAX, -1)
    min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(result)
    li.append(min_loc)
dirs = os.listdir(dirs_path)
for k in tqdm(dirs):
    if k.endswith('png'):
        real_path = os.path.join(dirs_path, k)
        match(real_path)
    else:
        continue
md5str = ''
for i in li:
    md5str += (str(i[1] // 55 + 1) + str(i[0] // 71 + 1))
md5=hashlib.md5(md5str.encode())
print(md5.hexdigest())
```


但是需要修改一下，填上path,在第22行后加上print(li)

哦对，那几个字是“人美歌甜”，用https://www.qqxiuzi.cn/zh/jiudiezhuan/可以查字，第二个美，猜到第一个是人之后直接百度“人美”，后面就会出现“歌甜”

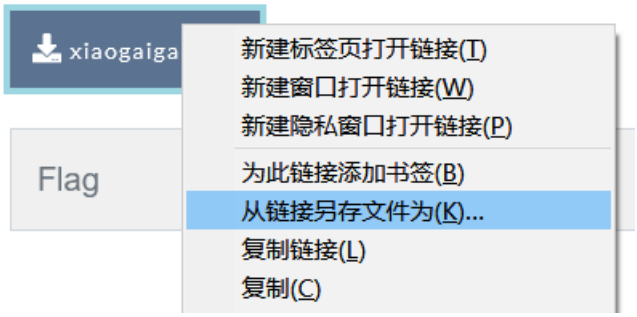
```python
import cv2
import os
from tqdm import tqdm
import hashlib
dirpath = "C:\\Users\\mumuzi\\Desktop"
dirs_path = dirpath + r"\output" #字条单个字的图片所在文件夹
source = cv2.imread(dirpath + r"\demo.png")#碑文图片
li = []
def match(temp_file):
    template = cv2.imread(temp_file)
    result = cv2.matchTemplate(source, template, cv2.TM_SQDIFF_NORMED)
    cv2.normalize(result, result, 0, 1, cv2.NORM_MINMAX, -1)
    min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(result)
    li.append(min_loc)
dirs = os.listdir(dirs_path)
for k in tqdm(dirs):
    if k.endswith('png'):
        real_path = os.path.join(dirs_path, k)
        match(real_path)
    else:
        continue
md5str = ''
print(li) #宽，高，并且开始是(0,0)，一定要注意是(0,0),而且这道题是反过来的
for i in li:
    md5str += (str(i[1] // 55 + 1) + str(i[0] // 71 + 1))
md5=hashlib.md5(md5str.encode())
print(md5.hexdigest())
```

```
[(600, 500), (200, 3600), (3599, 6599), (7200, 7200)]
```

注意脚本里面写的，这里是行列，而且是从(0,0)开始，这个图是100*100，所以得到的坐标是：(6,7),(37,3),(67,37),(73,73)

> ctfshow{人67_美373_歌6737_甜7373}

美丽的小姐姐



提示CRC错误，直接改高度

> flag{BF9FE48F92A9AB8948F5E266E7CE8EB4}
> baby 杯的唯二baby了

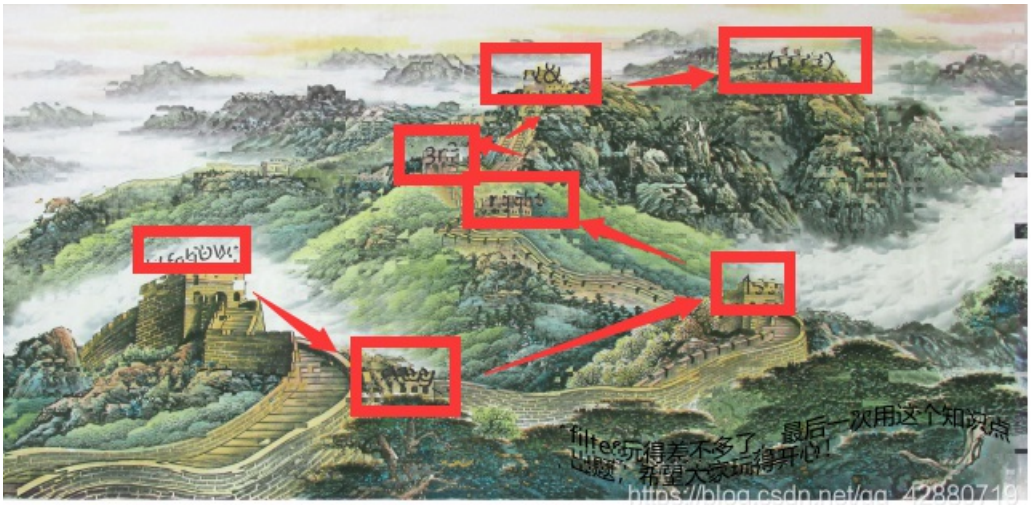
**万里长城 **
提示是filter，这里用filter然后拼图，太麻烦了。
这里考虑时间，因为出题人是一个个切的，切的时候估计是按照时间顺序切 的，所以如果对时间排序的话，可能可以成功还原。
但是我的脚本比较烂，是那种完全硬跑的烂脚本，这边师傅们做建议使用np和cv2来优化。或者只使用PIL模块的话可以在sort的时候进行重命名，这样大概十几秒就跑出来了
我跑了十分钟-.-，边跑顺便看别的题，我不慌的

```
import os
from PIL import Image
list1 = ['']*10000
list2 = ['']*10000
i = 0
path = 'C:\\Users\\mumuzi\\Desktop\\题目-拼图-长城\\random'
path1 = 'C:\\Users\\mumuzi\\Desktop\\题目-拼图-长城\\random\\'
for filename in os.listdir(path):
    list1[i] = filename
    i += 1
# print(list1)

for j in range(10000):
    list2[j] = os.path.getmtime(path+'\\'+list1[j])
# print(list2)

list3 = list2.copy()
list3.sort()
print(list2[:20])
print(list3[:20])
pic = Image.new('RGB',(10000,10000),(255,255,255))
for i in range(10000):
    s = list3[i]
    for j in range(10000):
        if(s == list2[j]):
            f = Image.open(path1+list1[j])
            pic.paste(f,((int(i%100))*99,(int(i/100))*40))
            break
        else:
            print(i)
pic.show()
pic.save("flag.png")
```

跟着塔走就能拼起来
在这里插入图片描述



不是我不想放大图，是写完发现图片违规???

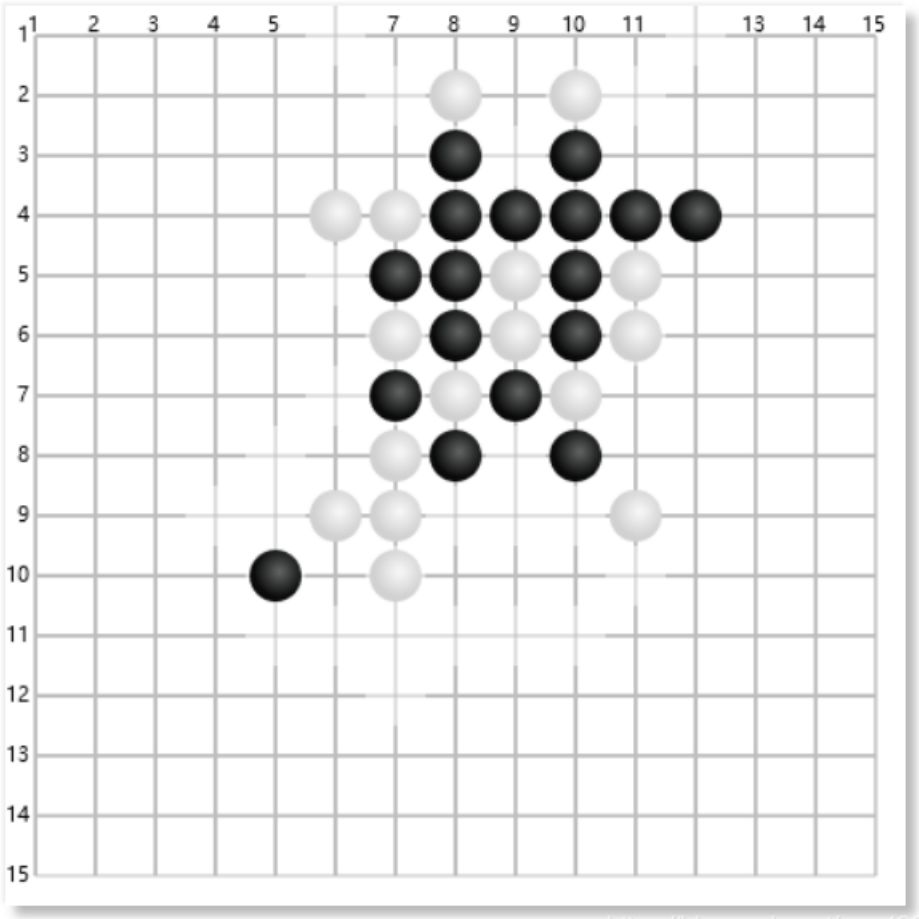
多试几次，提交

> ctfshow{flag_1s_r1ght_h3r3_&&_th3r3}

五子棋

就下棋呀，不难的

你赢了！！！ flag{Wu_J1n_



当然师傅们也厉害，有用app的，有B站看视频的，还有开两个窗口让这个机器互下的(有一方黑落1,1即可)

不问天

下载下来，foremost

zip加了密，看那个音频，文件尾有一串base64

6L+R6YK75rOV，得到近邻法，用PS即可，记得先选近邻

当然，近邻法有加密脚本，如下：

```python
import sys
from PIL import Image

#将small_img中的像素用近邻法嵌入到big_img中
def my_nearest_resize(big_img, small_img):

    big_w, big_h = big_img.size
    small_w, small_h = small_img.size

    dst_im = big_img.copy()

    stepx = big_w/small_w
    stepy = big_h/small_h

    for i in range(0, small_w):
        for j in range(0, small_h):
            map_x = int( i*stepx + stepx*0.5 )
            map_y = int( j*stepy + stepy*0.5 )

            if map_x < big_w and map_y < big_h :
                dst_im.putpixel( (map_x, map_y), small_img.getpixel( (i, j) ) )

    return dst_im



if __name__ == '__main__':
    big_img=Image.open(sys.argv[1])     # 大图
    small_img=Image.open(sys.argv[2])   # 小图

    dst_im = my_nearest_resize(big_img, small_img)
    dst_im.save(sys.argv[3])            # 嵌入小图像素的大图
```

密码是BVnumber，平台上写了

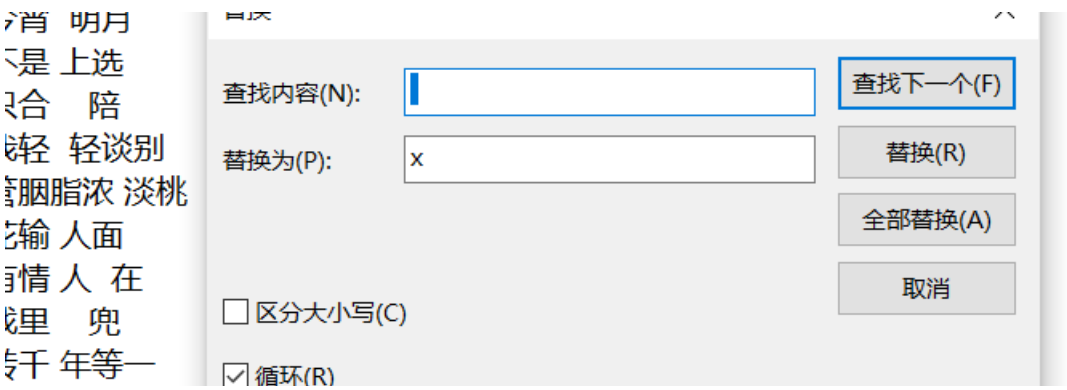
100

感谢@ErrOr师傅供题

原曲：https://www.bilibili.com/festival/2021bnj?bvid=BV1sv4y1f7Q2

附件地址：https://ctfshow.ianzoui.com/iRQD

解开之后得到一个文本，观察发现，空格有长有短，替换一下



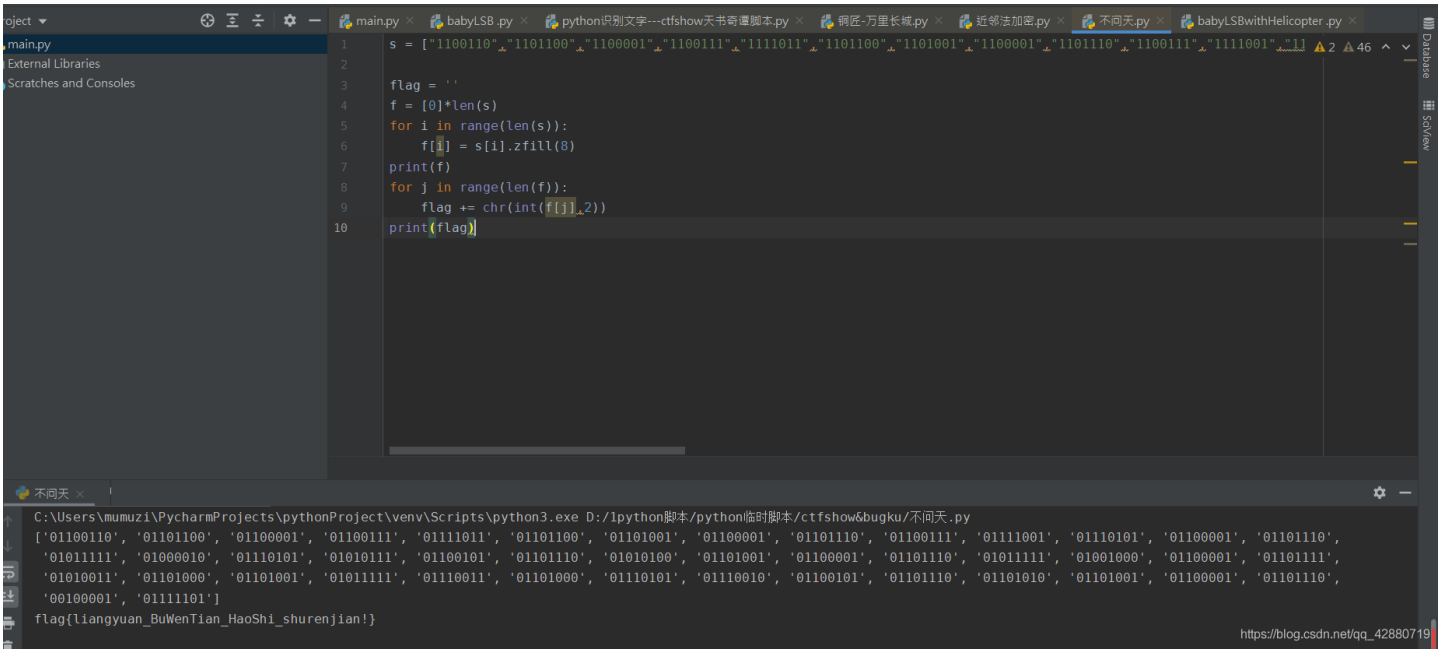
声喊 冲天和

今宵xx明月x
不是x上选xx
只合xxxx陪
我轻xx轻谈别
管胭脂浓x淡桃
花输x人面xx
有情x人xx在
戏里xxxx兜
转千x年等一x
声喊xx冲天和
地拜上一xx拜
才算圆x满x花
要向xxxx枝
头簪x才不负x
人x间西厢外那
一xxxx眼x
比梦勇x敢x而
我x不x必独自
寻遍xx全x闲
院就x遇见余x
生x听x琴xx
的少x年xx欢
喜不xxxx问
天十x万春风x
心x尖儿看管他
江xx南xxx
岸还xxxx是
杨柳x玉门关天
山x千xx里雪
伸手x拦xxx
一拦x花xx雕
敬x一盏请为我
开颜风xx流不
问天x白xxx
云遮住x长x生

发现每行长度都是7，将中文换成1，空格换成0，然后处理前两个，发现正好是fl，撸下来，用脚本跑：

```python
s = ["1100110","1101100","1100001","1100111","1111011","1101100","1101001","1100001","1101110","1100111","1111001","1110101","1100001","1101110","1011111","1000010","1110101","1010111","1100101","1101110","1010100","1101001","1100001","1101110","1011111","1001000","1100001","1101111","1010011","1101000","1101001","1011111","1110011","1101000","1110101","1110010","1100101","1101110","1101010","1101001","1100001","1101110","100001","1111101"]

flag = ''
f = [0]*len(s)
for i in range(len(s)):
    f[i] = s[i].zfill(8)
print(f)
for j in range(len(f)):
    flag += chr(int(f[j],2))
print(flag)
```



> flag{liangyuan_BuWenTian_HaoShi_shurenjian!}

baby_gay

签到

IDA打开找到密文

```
unsigned __int64 v5; // [rsp+48h] [rbp-8h]

v5 = __readfsqword(0x28u);
puts("please input your password:");
while ( 1 )
{
  scanf("%s", &v4);
  if ( verifyPwd(&v4) )
    break;
  puts("wrong password, please input again:");
}
printf(
  "wow!Congratulation!you find me,here you are,some important information:Gif4BdadxXkMLA6CXdipU3dnesRGYMzuio/D48HJ+rQ=");
return 0;
}
```


verifypwd里面找到秘钥

```
return strcmp("ZGFuaXU=", a1) == 0;
```

然后试呗，RC4成功解开

> ctfshow{76720c5adee75ce9c7779500893fb648}