

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](#)

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/qq_42880719/article/details/117479024

版权



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

输入文本信息:

LoaYuBeMnehSieW{wohsftc\n.n.htaeD ot nwonknU\n.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

https://blog.csdn.net/qq_42880719

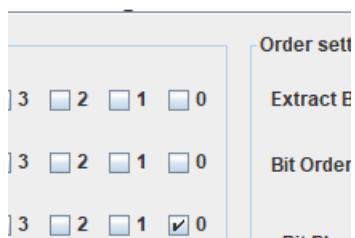
然后猜了个LSB

ctfshow{WeiShenMeBuYaoLSB}

预期：

因为用stegsolve的时候g通道发现有内容

```
I am the bone of
my swor d..Steel
is my b ody, and
fire is my bldl
.|..... .p....
..... .+..k
..0(.... .`?.....
..{C.... □....C..
../.!..O ?.....
..... □..1...
```



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]+t4[:-1]
s = ''
rflag = ''
for i in flag:
    s+=i
    if len(s)==8:
        rflag += chr(int(s,2))
        s=''
print(rflag)
```

```
LSB11 ×
I have created over a thousand blades .
Unknown to Death .
ctfshow{WeiShenMeBuYaoLSB}
Nor known to Life .
Have withstood pain to create many weapons .
Yet , those hands will never hold anything .
So as I pray , Unlimited Blade Works .
https://blog.csdn.net/qq_42880719
```

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 = ''
```


babyLSBwithHelicopter

这道题是靠8神给的hint: braincopter才出的

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



The screenshot shows a Windows Notepad window with the title "flag1.txt - 记事本". The window contains a massive amount of Brainfuck code, which is highly compressed and non-readable. The code is composed of sequences of characters such as '>', '<', '+', and '-'. The Notepad interface includes standard menu options like "文件(F)", "编辑(E)", "格式(O)", "查看(V)", and "帮助(H)". At the bottom, there are status bars showing "第 1 行, 第 1 列", "100%", "Windows (CRLF).csf", and "UTF-8".

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

Output

String Hex

89 50 4e 47 0d 0a 1a 0a 00 00 00 0d 49 48 44 52 00 00 00 1e 00 00 00 22 08 06 00 00 00 48 88 69 74 00 00 00 01 73 52 47 42 00 ae ce 1c e9 00 00 00 04 67 41
4d 41 00 00 b1 8f 0b fc 61 05 00 00 00 09 70 48 59 73 00 00 12 74 00 00 12 74 01 de 66 1f 78 00 00 02 47 49 44 41 54 58 47 b5 52 01 92 83 40 08 bb b7 f4 a5
be e9 3e e8 91 08 18 70 a9 b6 33 d7 0c dd 15 42 b2 ac fe 0b df 63 c7 ff 66 eb 23 f6 cf be 19 6f 1b f9 48 6f 1b 49 1d 6c 40 a3 e4 02 e4 83 f2 35 c2 58 4f
66 db 7e 10 5a 04 c7 d6 88 32 95 e4 33 26 ad 30 0e 22 98 10 22 39 9a 3d cf df 5d 9b 03 d3 e4 45 47 10 3a 54 a2 f1 a2 b9 e0 ae 0e ac 38 96 eb e6 07 c2 d8 90 a7
f7 29 df 05 88 ab b9 e9 01 1e 57 b7 2b 21 c6 0a 12 a3 71 01 18 96 ef a2 03 a5 0e cb 9d 03 b3 31 b2 df be db dd 2b 30 f5 33 a3 1f 57 09 f4 f9 da 45 24 64 35
0a 0b c0 1b 6f a2 e8 b8 31 85 04 3a d1 05 c8 c7 e1 dc 64 7a df a8 71 d5 7e 02 25 9f 98 24 c1 f2 f4 78 84 a8 ad e7 e9 67 d0 50 0e 90 91 13 7b 81 c4 1b e4 6d
a8 50 0b 10 a6 5b 60 e4 c4 37 58 4e ff 09 ac b5 de d0 c2 98 e2 30 91 1c f0 6e 52 14 3e 7b cf 48 9b f1 a3 89 50 16 c1 31 0e 99 73 df eb 88 30 06 03 09 db 14
e4 94 0d 48 e6 84 0d 93 16 70 ea 81 86 ab 8e 93 b8 58 4e 2f f9 0c 2b 2d 0d f1 6f 75 f6 35 14 4d c2 8d c3 84 cd 82 38 21 fe a6 77 c8 be a1 3f 10 3a 67 06 2d
4f 26 b6 47 36 0b 28 13 3c 41 e9 ef 48 1d 37 a6 88 e0 7a 42 c7 2a 67 40 21 6f 44 80 3c 0f 27 b9 03 28 f9 c4 41 8a 28 27 8f 3c 1e 63 1f cf 4f 60 bc 7a a8 30
76 8c 93 2a ba b9 04 8a ab 6f 01 79 ae a9 82 94 1b 33 19 24 41 99 7c 82 95 eb 44 07 8a a6 ad 19 5f 4d 0c 3c e5 b8 11 1e ca 4d 74 63 ca 39 31 9a 22 ca e4 ad
66 a9 d4 78 8b e4 41 c6 3f 2e 8d 10 5f c9 e9 ad 60 93 93 08 a6 fe b3 17 14 33 fe 68 1a 7b ee 46 00 59 a1 d1 50 f4 a9 87 ad 5c f5 91 bc 01 38 2e 52 de 9b 04
6a 5c b5 cf 71 1c 1a 14 35 06 2c db 27 a2 80 08 46 94 49 24 5f 02 e5 0b dc 38 1a 57 14 1e c2 9b b1 59 bd 53 e0 a9 c6 01 d0 17 1f d7 65 12 db 76 33 ca 04 47
50 7a 3b 26 50 86 31 12 30 68 81 c2 d7 ef 11 94 11 68 ef 13 5b b6 4f 07 50 26 0c 1b de 4e d9 01 1a 4d 7e 5f af fd bf 62 9e 7c df ff 00 13 d8 a7 r2 48 f2 06
bf 2e 00 00 00 00 49 45 4e 44 ae 42 60 82 https://ilog.csdrone-rlkit.com:20200/join

JO	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52	IPNG	IHDR
16	00 00 00 1E 00 00 00 22 08 06 00 00 00 48 88 69	"	Hl
32	74 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00	t sRGB @I é	
48	00 04 67 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00	gAMA ± üa	
54	00 09 70 48 59 73 00 00 12 74 00 00 12 74 01 DE	pHYs t t b	
30	66 1F 78 00 00 02 47 49 44 41 54 58 47 B5 52 01	f x GIDATXGpR	
36	92 83 40 08 BB B7 F4 A5 BE E9 3E E8 91 08 18 70	'@ » .óYé>` p	
12	A9 B6 33 D7 0C DD 15 42 B2 AC FE F0 B7 DF 63 C7	©¶3x Ÿ B²-þþ·Bcç	
28	FF 66 EB 23 F6 CF BE 19 6F 1B F9 48 6F 1B 49 1D	ýfe#öI% o ùHo I	
44	6C 40 A3 E4 02 E4 83 F2 35 C2 58 4F 66 DB 7E 10	1@fæ àíð5ÁXOFU~	
30	5A 04 C7 D6 88 32 95 E4 33 26 AD 30 0E 22 98 10	Z ÖÖ121à3&-0 "I	
76	22 39 9A 3D CF D5 9B 03 D3 E4 45 47 10 3A 54 A2	"91=ÍÖI óáEG@Tç	
32	E1 A2 B9 F0 AE DE AC 38 96 FB F6 07 C2 D8 90 A7	831à@ -816n ÁQ S	

得到fflflag.png，是imagefuck，写脚本(脚本有注释)

```
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): 64, (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	Run
Output	String	Hex

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

https://blog.csdn.net/cq_42880719

```
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://7heright4th.top/blog/CTF/Misc/CTFShow%E5%88%B7%E9%A2%98%E7%AC%94%E8%AE%B0/>

```
1 import cv2
2 import os
3 from tqdm import tqdm
4 import hashlib
5 dirpath = path#根目录
6 dirs_path = dirpath + r"\output" #字条单个字的图片所在文件夹
7 source = cv2.imread(dirpath + r"\demo.png")#碑文图片
8 li = []
9 def match(temp_file):
10     template = cv2.imread(temp_file)
11     result = cv2.matchTemplate(source, template, cv2.TM_SQDIFF_NORMED)
12     cv2.normalize(result, result, 0, 1, cv2.NORM_MINMAX, -1)
13     min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(result)
14     li.append(min_loc)
15 dirs = os.listdir(dirs_path)
16 for k in tqdm(dirs):
17     if k.endswith('png'):
18         real_path = os.path.join(dirs_path, k)
19         match(real_path)
20     else:
21         continue
22 md5str = ''
23 for i in li:
24     md5str += (str(i[1] // 55 + 1) + str(i[0] // 71 + 1))
25 md5=hashlib.md5(md5str.encode())
26 print(md5.hexdigest())
```

https://blog.csdn.net/cjq_42880719

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”

```

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())

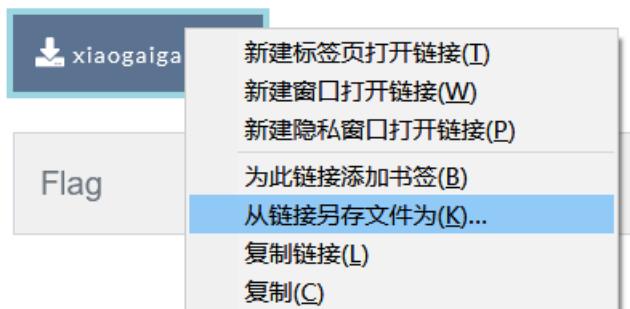
```

[(600, 500), (200, 3600), (3599, 6599), (7200, 7200)]
02e72d008ba1ce0c40aa6dcffcc1b6ad

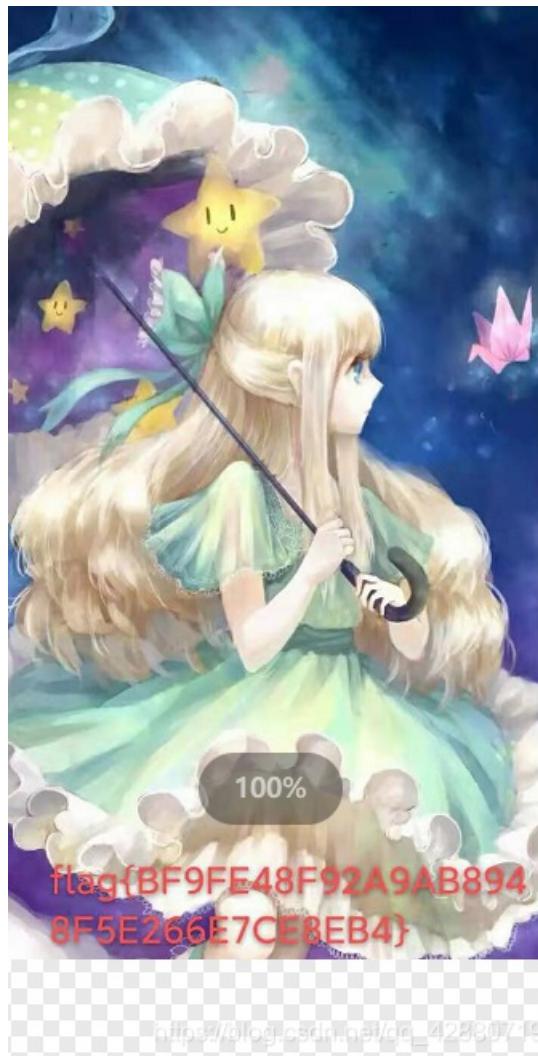
注意脚本里面写的, 这里是行列, 而且是从(0,0)开始, 这个图是100*100
, 所以得到的坐标是: (6,7),(37,3),(67,37),(73,73)

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



<https://blog.csdn.net/coder4280719>

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)

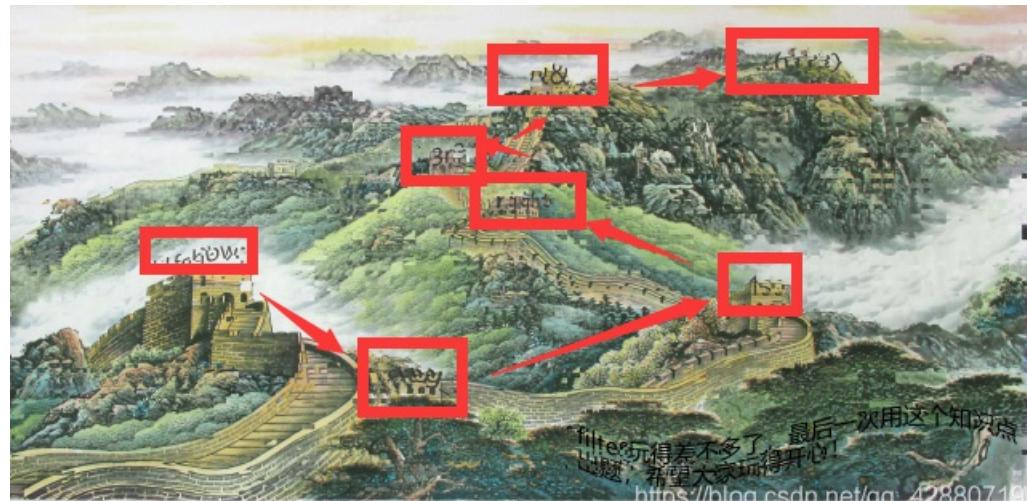
for j in range(10000):
    list2[j] = os.path.getmtime(path+'\\\\'+list1[j])
# print(list2)

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???

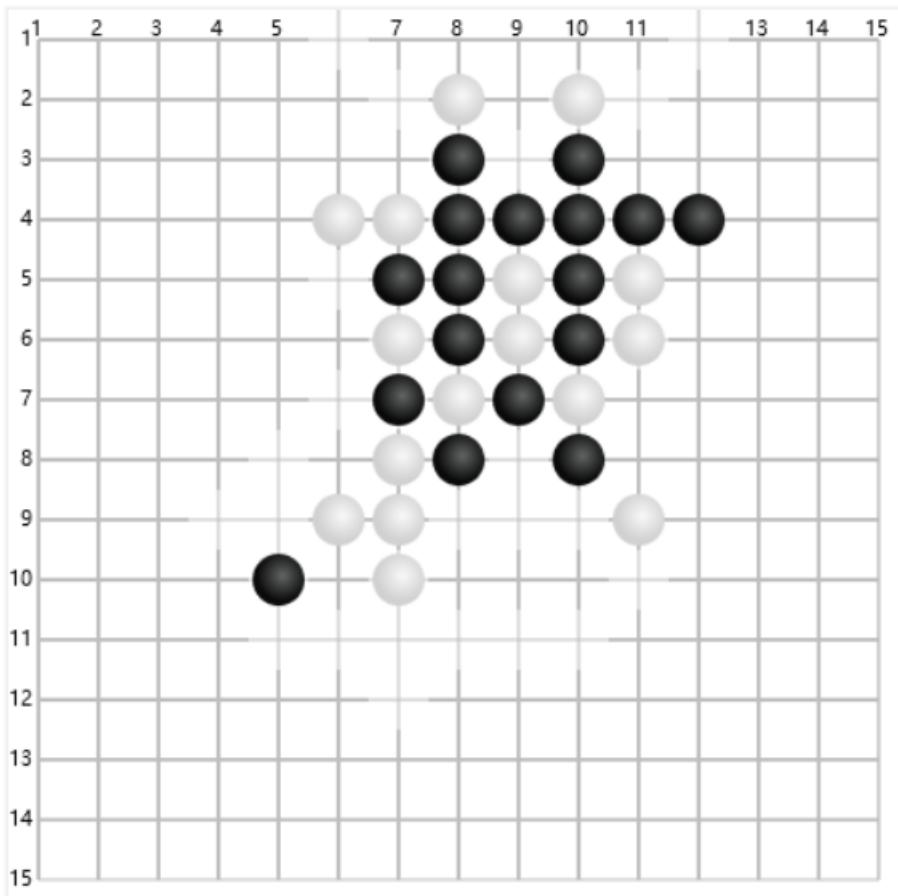
多试几次，提交

```
ctfshow{flag_1s_r1ght_h3r3_&&_th3r3}
```

五子棋

就下棋呀，不难的

你赢了！！！ flag{Wu_J1n_5



https://blog.csdn.net/qq_42880719

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：

```

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

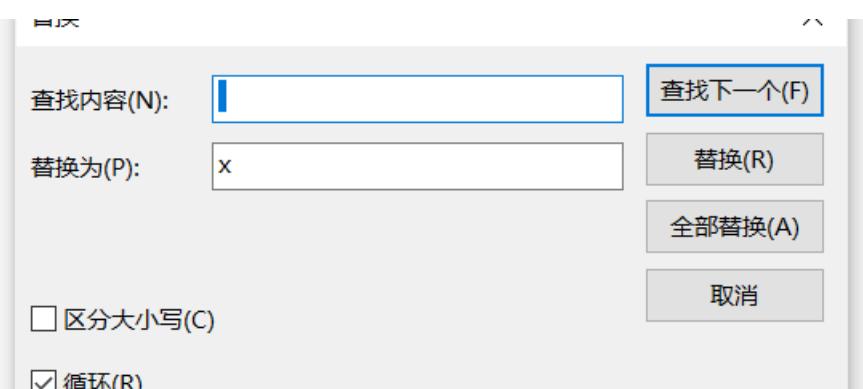
感谢@ErrOr师傅供题

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

附件地址：<https://ctfshow.lanzoui.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

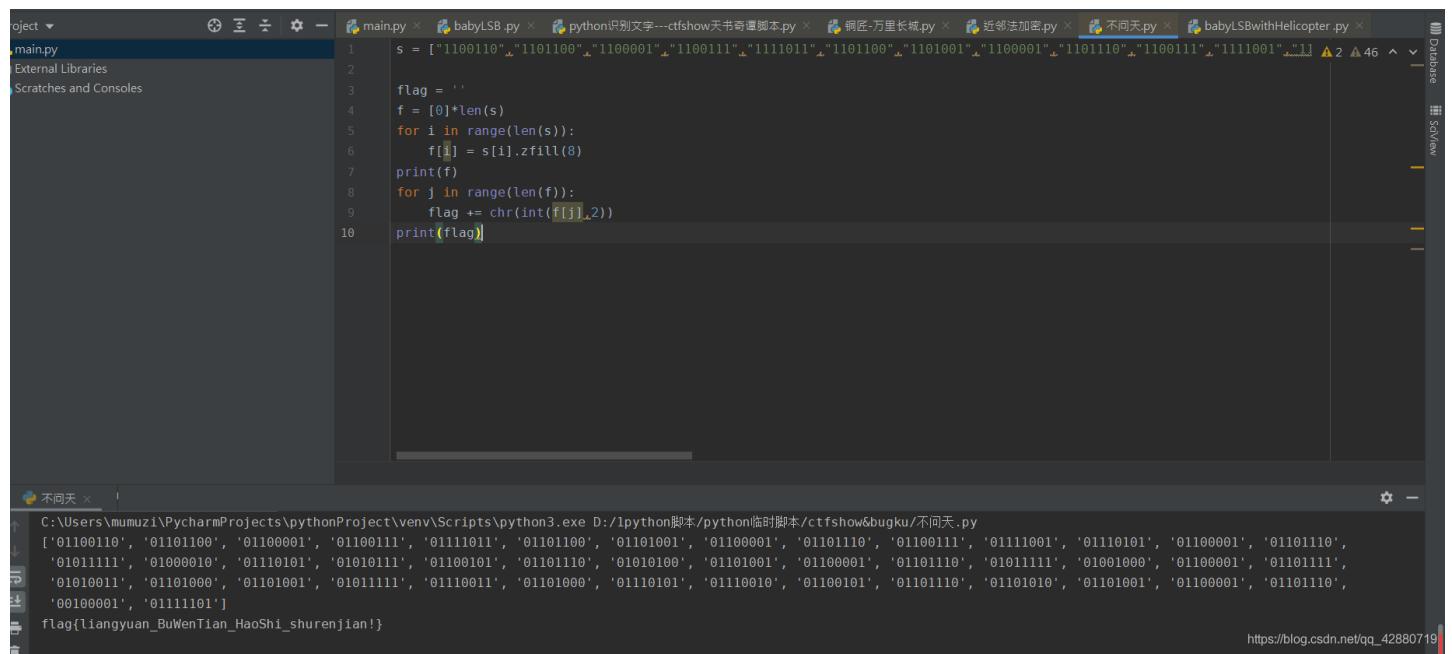
肖明月
是上选
合陪
轻轻谈别
胭脂浓淡桃
输入人面
情人在
里兜
千年等一



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—x
声喊xx冲天和
地拜上—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
—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—蓋请为我
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：

```
s = ["1100110", "1101100", "1100001", "1100111", "1111011", "1101100", "1101001", "1101110", "1100111", "1111001", "1110101", "1100001", "1101110", "1011111", "1000010", "11010101", "1010111", "1100101", "1101110", "1010100", "1101001", "1101111", "1010001", "1101110", "1011111", "1001000", "1100001", "1101111", "1010011", "1101001", "1101111", "1110011", "1101000", "11010101", "1110010", "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)
```



The screenshot shows the PyCharm IDE interface. On the left, the project tree displays files like main.py, babyLSB.py, and others. The main editor window contains the provided Python script. Below the editor, the terminal window shows the execution of the script and its output:

```
C:\Users\mumuzi\PycharmProjects\pythonProject\venv\Scripts\python3.exe D:/1python脚本/python临时脚本/ctfshow&bugku/不问天.py  
['01100110', '01101100', '01100001', '01100111', '01110111', '01101100', '01101001', '01101110', '01100111', '01111001', '01110101', '01100001', '01101110', '01000010', '01110101', '01010111', '01100101', '01101110', '01010100', '01101001', '01100001', '01101111', '01010001', '01101000', '01100001', '01101111', '01010011', '01110011', '01101000', '01110101', '01110010', '011010101', '01101110', '01101010', '01101001', '01101110', '00100001', '01111101']  
flag{liangyuan_BuWenTian_HaoShi_shurenjian!}
```

flag{liangyuan_BuWenTian_HaoShi_shurenjian!}

baby_gay

签到

IDA打开找到密文

```
unsigned __int32 v3, v4, v5;
char v6[100];
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;
}
```

https://blog.csdn.net/qq_42880719

verifyPwd里面找到秘钥

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

然后试呗，RC4成功解开

加密密码: 选择字符集:

待加密、解密的文本:

↑ 将你电脑文件直接拖入试试^-^

RC4加密、解密转换结果(base64了):

76720c5addee75ce9c7779500893fb648

https://blog.csdn.net/qq_42880719

ctfshow{76720c5addee75ce9c7779500893fb648}