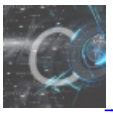
2020 DASCTF&BJD MISC WriteUp

原创

七月7yue 于 2020-12-28 21:06:39 发布 622 收藏 3

分类专栏： DASCTF MISC CTF

本文链接：https://blog.csdn.net/qyCraner/article/details/111872637


DASCTF 同时被 3 个专栏收录

1 篇文章 0 订阅

订阅专栏


MISC

4 篇文章 0 订阅

订阅专栏


CTF

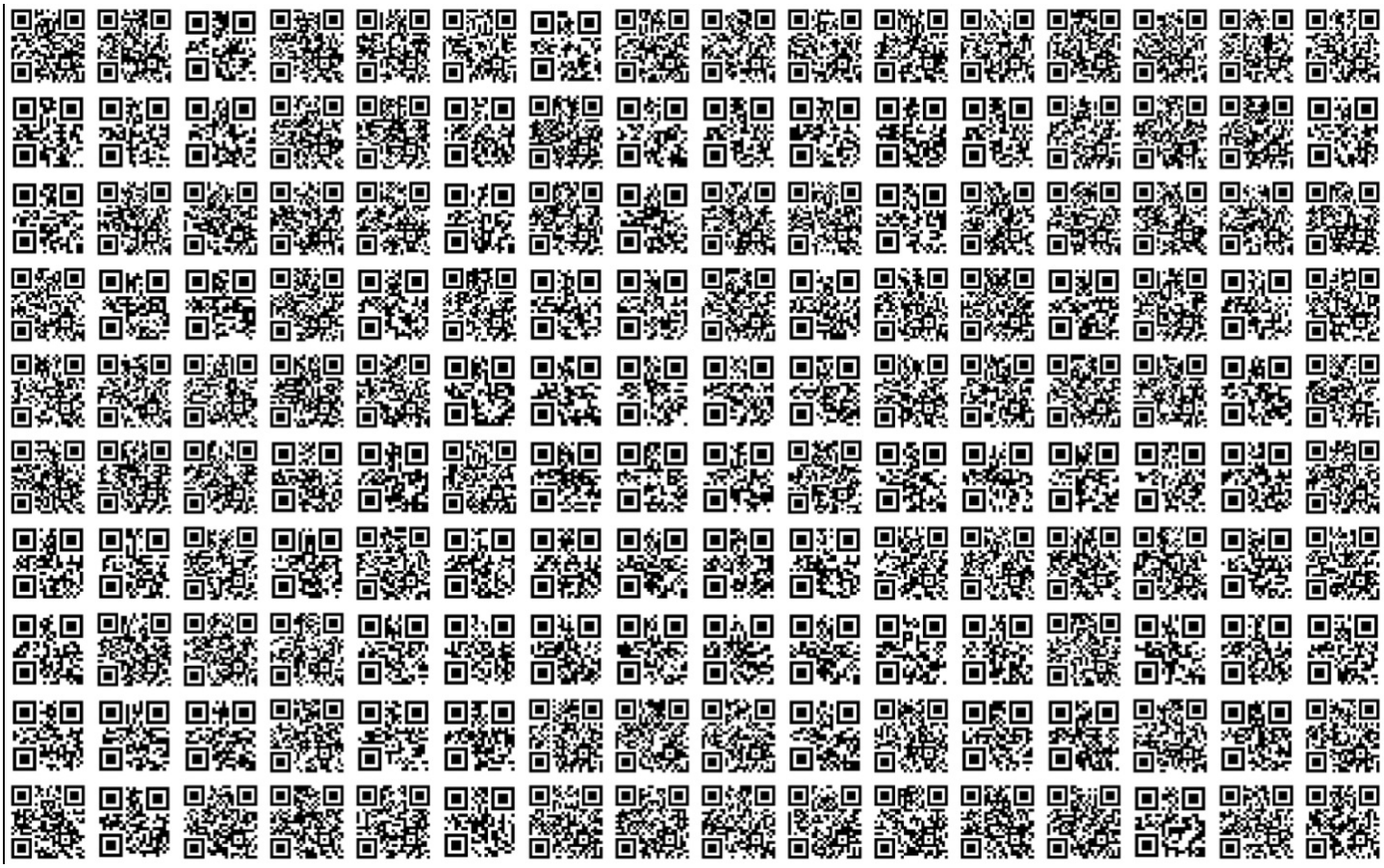
5 篇文章 0 订阅

订阅专栏

这次比赛队里的大师傅们太强惹，只会做2道MISC，被大佬队友们带到了第三名。这篇文章就记录一下这次比赛MISC的题解。

马老师的秘籍

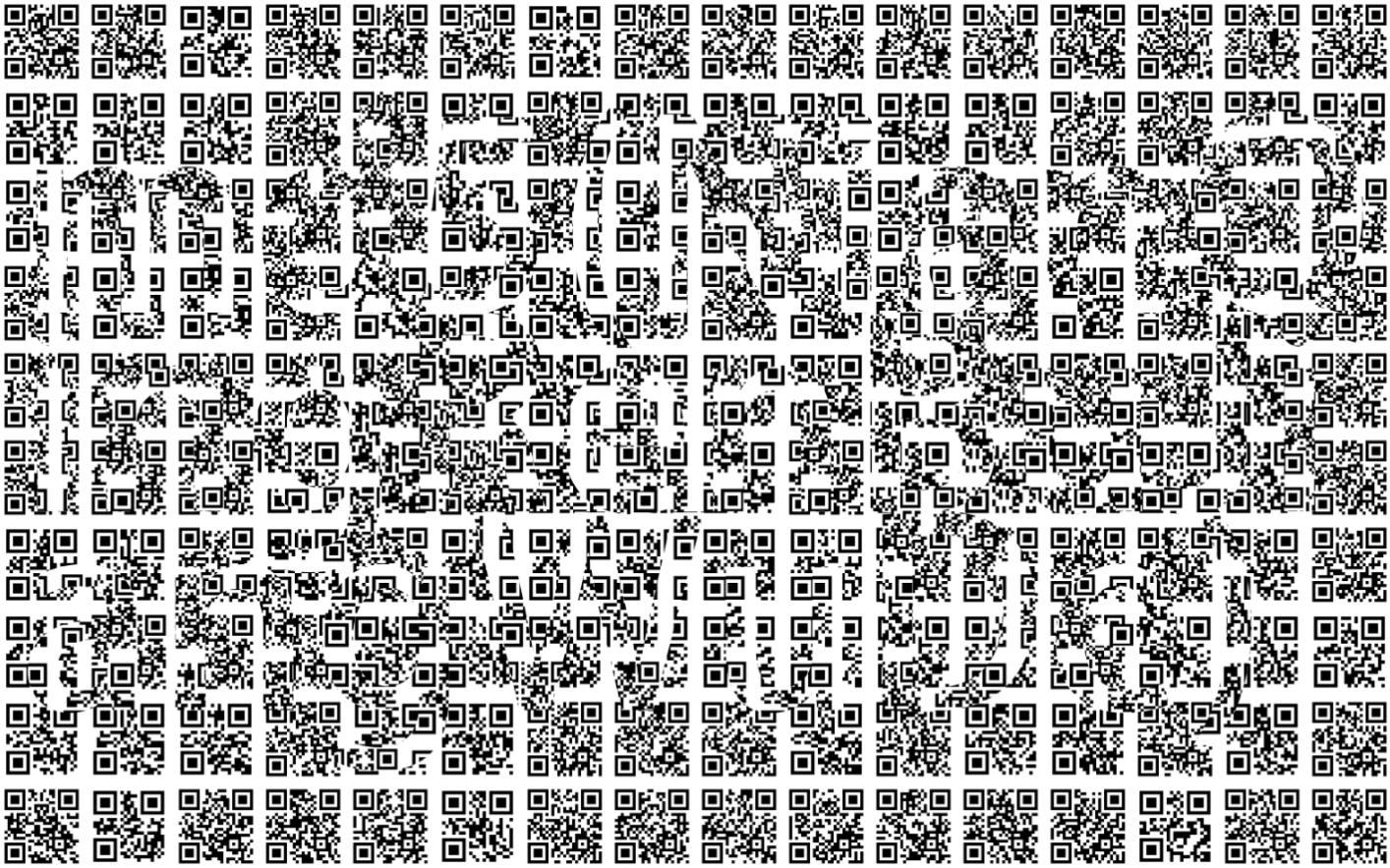
打开得到一张全是二维码的png图片，下载下来



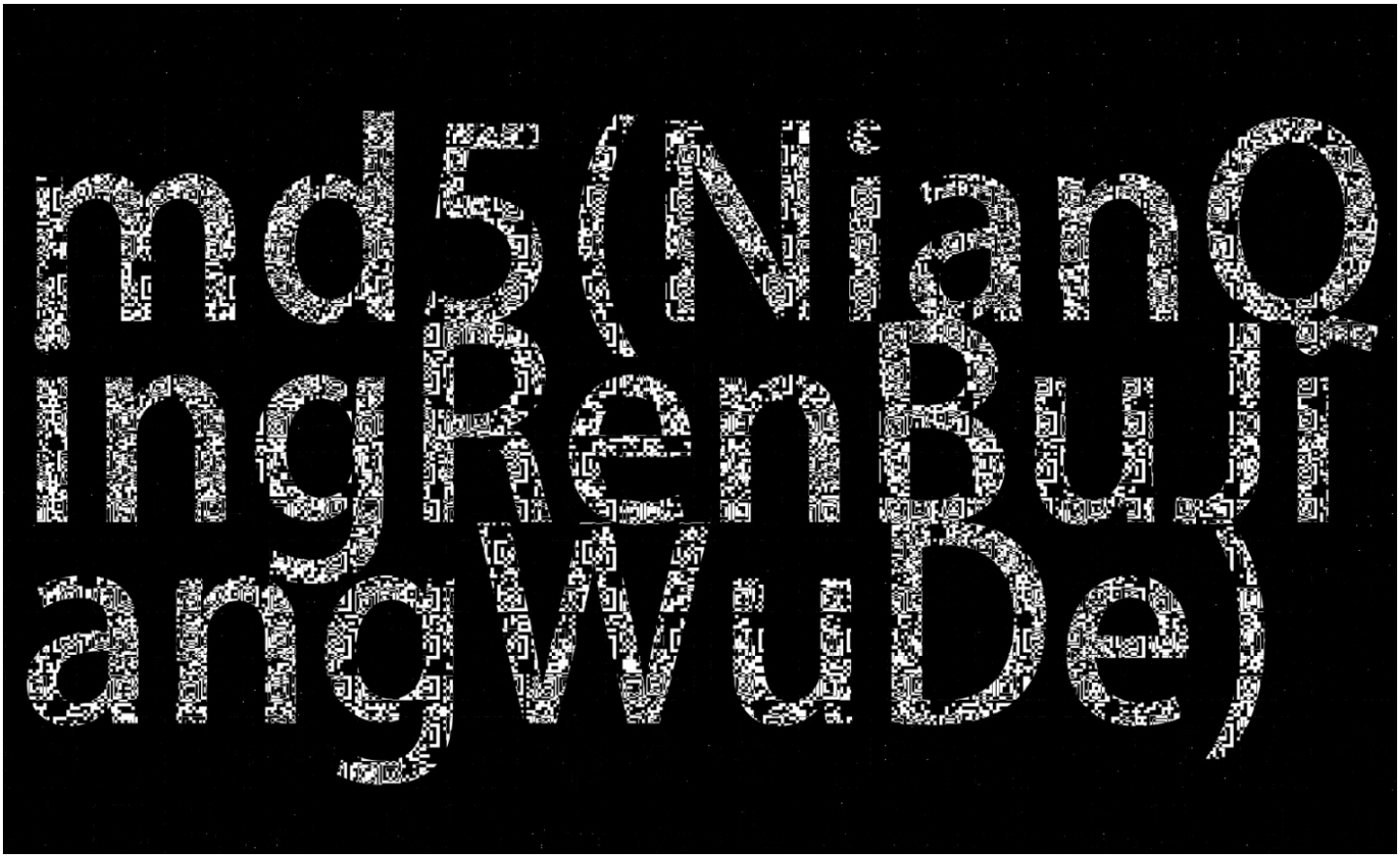
按顺序扫码后得到：

呃朋友们好啊我是某某门掌髡亻莠コ閾ヤ某某刚才有个譜句暑髡ヨ我马老师发生什么事了我说諤惹ケ亥屓事给我发了几张截图我一看嗷原来是昨天譜我ク、荁ェ蟷工霓サ莠コ三十多岁一个体重九十多公斤一个体重八十多公譁、莉紋サヶ隸工蜇画怏一个说是我在健身房练功颈椎练坏了马老师你能不能教教我浑元功法遏ヨ蟶ヨ蚜ッ治疗一下我的鬚域、守羅謂題ッエ蜿ッ莉·謂題ッエ你在健身房练豁サ蚜亻蜆ソ不好用他不服气我说小朋友ä¼ ä¸¤ä¸ª手来找我一个謇区欠螟工他折不蚜亻莉冶ッエ你这也没用我说我这个有用这是化劲儿传扈溷粥螟ォ是讲化劲儿的蝗帑ク、謚亻千金二百多斤的英国大力士都握我不动我这一个謇区欠螟工蝠贗サ夜撼要和我试试我隸工蜿ッ莉·诶我一隸工莉門分就站起来了很蠢ォ蝠颜ㄒ后上来就是一个左正蹬一个右鞭腿一个左刺拳我全部防出去了蝠企亟蜃コ去以后自然是传统功夫以点蛻-荁コ豁「右拳放到他鼻子上没打他我笑一下准备收拳因为需呎慮髡工欸传统功夫的轤ケ蛻-荁コ止他已经输了如果这一拳发力一拳就把他鼻子打骨折了放在鼻子上没有打他他也承认我先打到他面部他不知道拳放在他鼻子上他也承认我先打到他面部啊我收拳的时髡工荁肴遠了他突辟力陲ュ蜃サ左刺拳来打我脸啊我大意了啊没有闪矮他的左拳给我眼蝠雁承逵シ蹭了一下但没关系啊莉紋ケ溯ッエ蝠贗サ匆穭蝗ヒ惹溯ッエ了两分螟复帖莉·蜷主ス捺虙流眼泪了捂着逵シ謂题ッエ

实际并没有什么用，binwalk一下，得到两个txt文件，还有一张图片文件，但是打不开
foremost一下，获得不了两个txt文件，但是能把隐藏的jpg图片提取出来



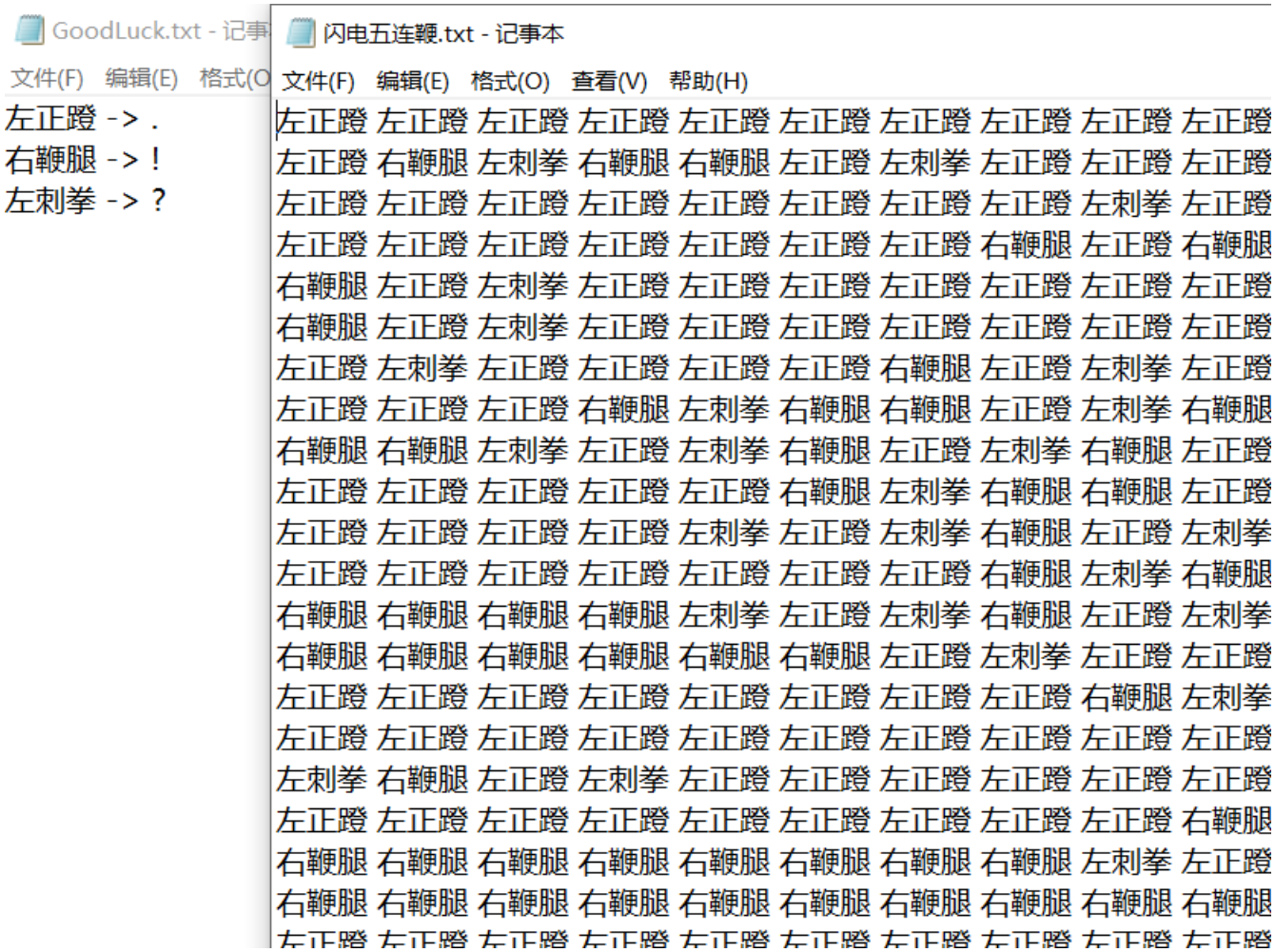
两张图片异或一下，得到：



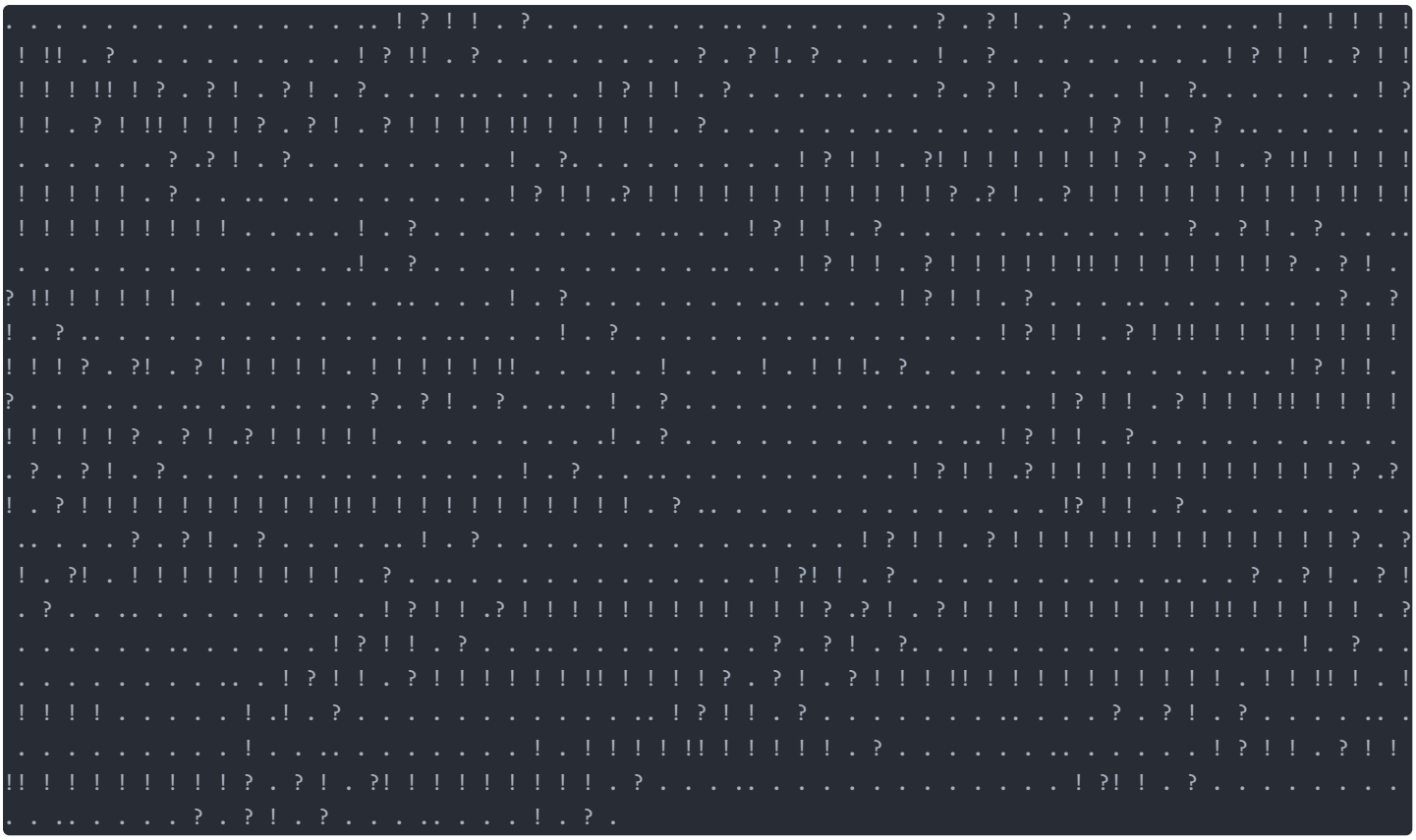
NianQingRenBuJiangWuDe

md5加密一下，得到压缩包密码：c57988283c92f759585a0c1aebfdd743

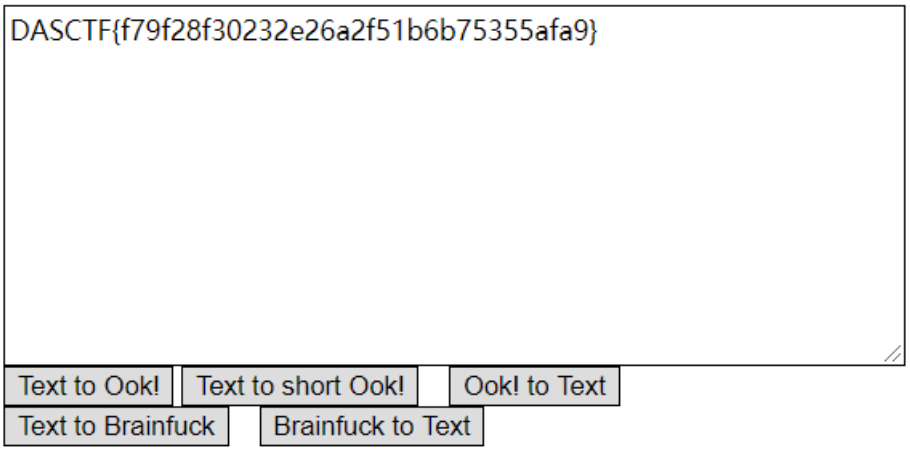
压缩包内有两个记事本文件：



根据Goodluck中的提示，将闪电五连鞭.txt中的数据替换后得到：



Ook!解码一下，得到flag：



FakePic

下载附件得到一个加密的压缩包，提示password：1???小写

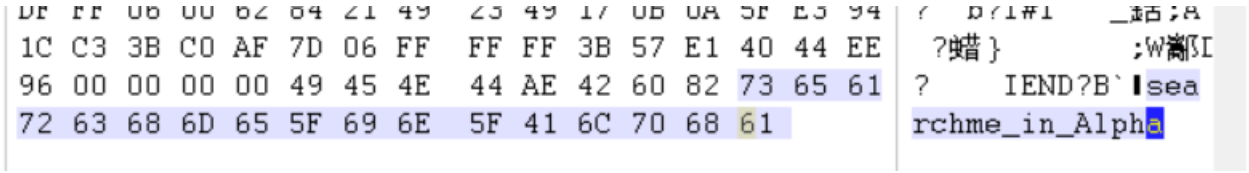
爆破一下得到密码：1cpp

得到一张图片和一个hint.txt

hint.txt内容：

```
10132430
取最前面500位就行
```

用winhex打开图片，在最后能发现提示在alpha通道内有数据



利用stegsolve也能发现alpha通道内图片坐标有特殊数据：

写一个脚本提取数据，提示了提取前500个就好，经过测试只需要提取前320个即可

```
from PIL import Image
pic = Image.open("flag.png")
red, green, blue, alpha = pic.split()
a,b = alpha.size
fp = open("1.txt","w")
for y in range(0,320):
    fp.write(str(alpha.getpixel((0,y))))
    fp.write('\n')
fp.close()
```

发现只有 1 2 4 8 16 32六种数据，且开头大多为4 1 2，少部分其他的，但也类似。
一共320个字符，八位为一组转化成ASC码，40长度，DASCTF{md5}正好也是40长度，可知可能正好为8位二进制替换asc码为一组。

其中1 2 4 8 16 32转换成2的次方，即1 -> 0， 2-> 1， 4-> 2，8-> 3,16 -> 4， 32-> 5

然后与hint中的10132430相减，发现只有01字符。

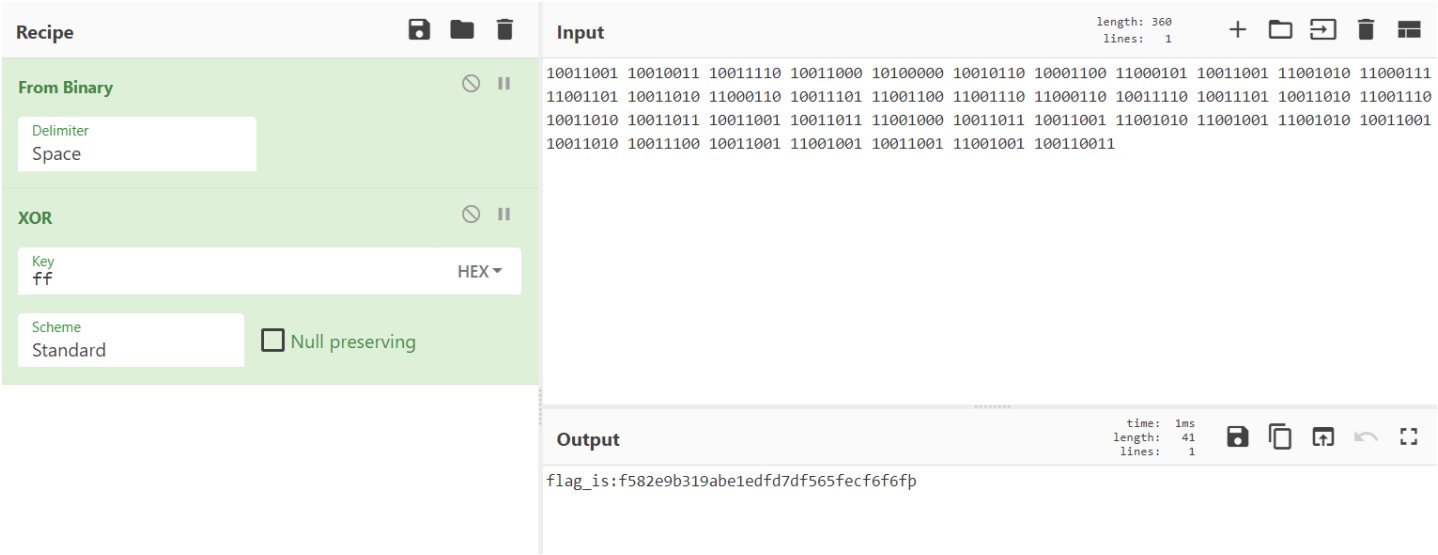
写一个脚本：

```
from PIL import Image
import gmpy2
pic = Image.open("flag.png")
red, green, blue, alpha = pic.split()
i = 0
st = ''
flag = ''
for y in range(320):
    s = alpha.getpixel((0,y))
    st += str(int(gmpy2.log2(s)))
    i += 1
    if i == 8:
        i = 0
        num = int(st) - 10132430
        st = ''
        flag += str(num)
        flag += ' '
print(flag)
```

得到：

```
10011001 10010011 10011110 10011000 10100000 10010110 10001100 11000101 10011001 11001010 11000111 11001101 1001
1010 11000110 10011101 11001100 11001110 11000110 10011110 10011101 10011010 11001110 10011010 10011011 10011001
 10011011 11001000 10011011 10011001 11001010 11001001 11001010 10011001 10011010 10011100 10011001 11001001 100
11001 11001001 10011001
```

asc码的第一位不可能为1，所以1肯定是有问题的，将二进制转码之后对FF异或，即可得到flag，发现其实并不是DASCTF开头的，但是正好也是40位，就因为我一直以为开头是DASCTF，导致卡了好几个小时，最终好像是一个二血吧。



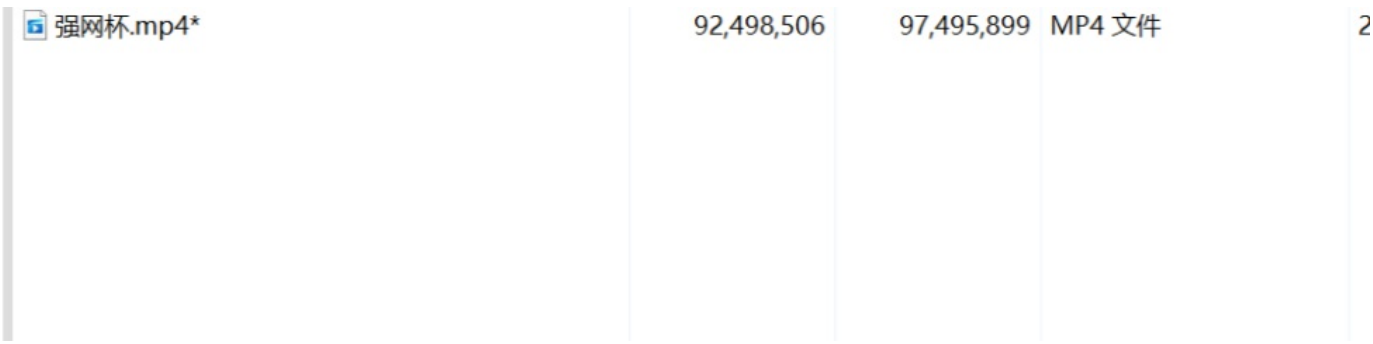
FakePixel

这题比赛的时候没有解出来，当时好像是0解，赛后复现一下。
下载附件得到一张794MB的图片。然后还有一个加密脚本，加密脚本比较简单，主要就是跟0xFF异或。
写一个脚本提取数据，当时比赛的时候采用的是遍历读写的方法，要跑很久。赛后大佬队友写了个只需要三秒就能跑完的脚本，膜拜host。

```
from PIL import Image
import numpy as np
import time

Image.MAX_IMAGE_PIXELS = None
t=time.time()
print(t)

with open("secret2.txt", 'wb') as f:
    im = Image.open('./FakePicture.bmp')
    img = np.array(im)
    for i in img:
        f.write(np.delete(i, [0, 1], axis=1).flatten().tobytes())

print(time.time()-t)
```

能发现末尾是 **04034b50**，显然是一个压缩包倒序了，有很多空格，删掉后，将数据倒序即可。

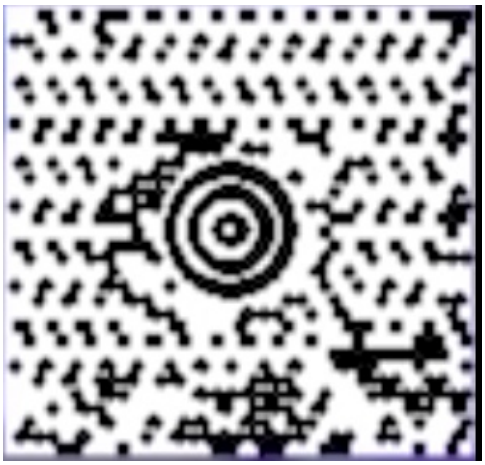
得到一个压缩包，里面又有Ook编码，但是得倒序一下才是正常的Ook



```
..... ..... ..... ..... !?!!. ?.... ..... ..... ..... .?.?! .?!.! !!!!!
!.?.. ..... ..!?! !.?.. ..... .?.?! .?... .!.?. ..... ...!? !!.?! !!!!!
!!?.? !.?!. ?.... ..... !?!!. ?.... ....? .?!.? ..!.? ..... ..!?! !.?!!
!!!!? .?!.? !!!!! !!!!! !.!!! !!!!! !!!!! !!.?. ..... ..... ..!?! !.?!!
!!!!! !!!!! ?.?!. ?!!!! !!!!! !!!!! !!!!! !!.?. ..... ..... ..... .!?!!
.?... ..... ..... ...?. ?!.?. ...!. ?.... ..... !?!!. ?!!!! !!!!? .?!.?
!!!!! !!!!. ?.... ..... !?!!. ?.... ....? .?!.? ..... ...!. ?.... .....
..... ...!? !!.?! !!!!! !!!!! !!!!! ?.?!. ?!!!! !!!.? ..... ..... ...!?
!!.?. ..... ..... .?.?! .?... ..... ..... ..... ....! ..... ..... .....
!.?.. ..... !?!!. ?.... ..?.? !.?.. ..... ..... !.!!! !!!!! !!!!! !!.!.
?.... ...!? !!.?. ..... ?.?!. ?.... !.?.
```


Ook解密得：**dasctf_1s_s0_funny**

得到一个视频，视频的33秒左右的右下角显示了一个码



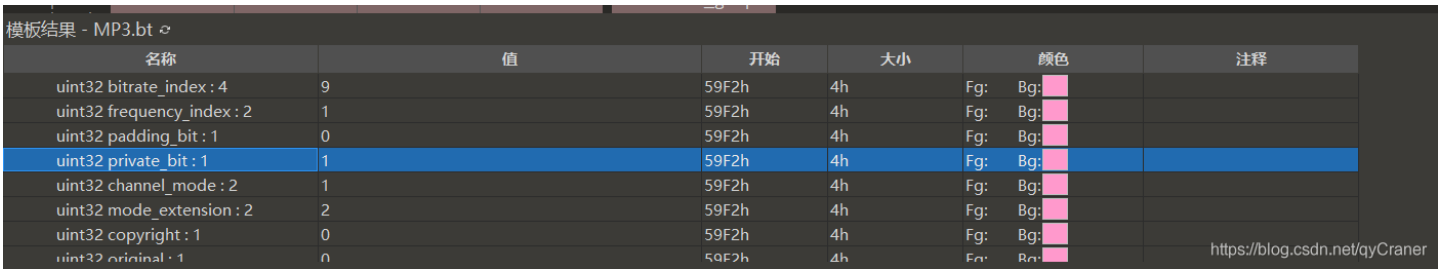
这是个MaxiCode码，网上应该有在线扫码工具，我用的是手机上的一个扫码软件。

扫码得到：**Citrix CTX1**，这是一种加密，在CrybeChef上就能解密，不需要密钥，但是目前还没有密文。

然后重新看一下视频，发现有一个音频，搜索一下mp3的头文件：**49443303**，发现存在一个mp3文件，从49443303到最后均为mp3文件的数据，手动提取出来。但是尝试了各种以前常用的加密都没用。

google了一下发现了一种MP3的隐写方式 **private_bit**，这在之前的De1CTF上也出现过

用010editor打开mp3文件，发现private_bit这位既有0又有1，猜测可能隐写了二进制数据。



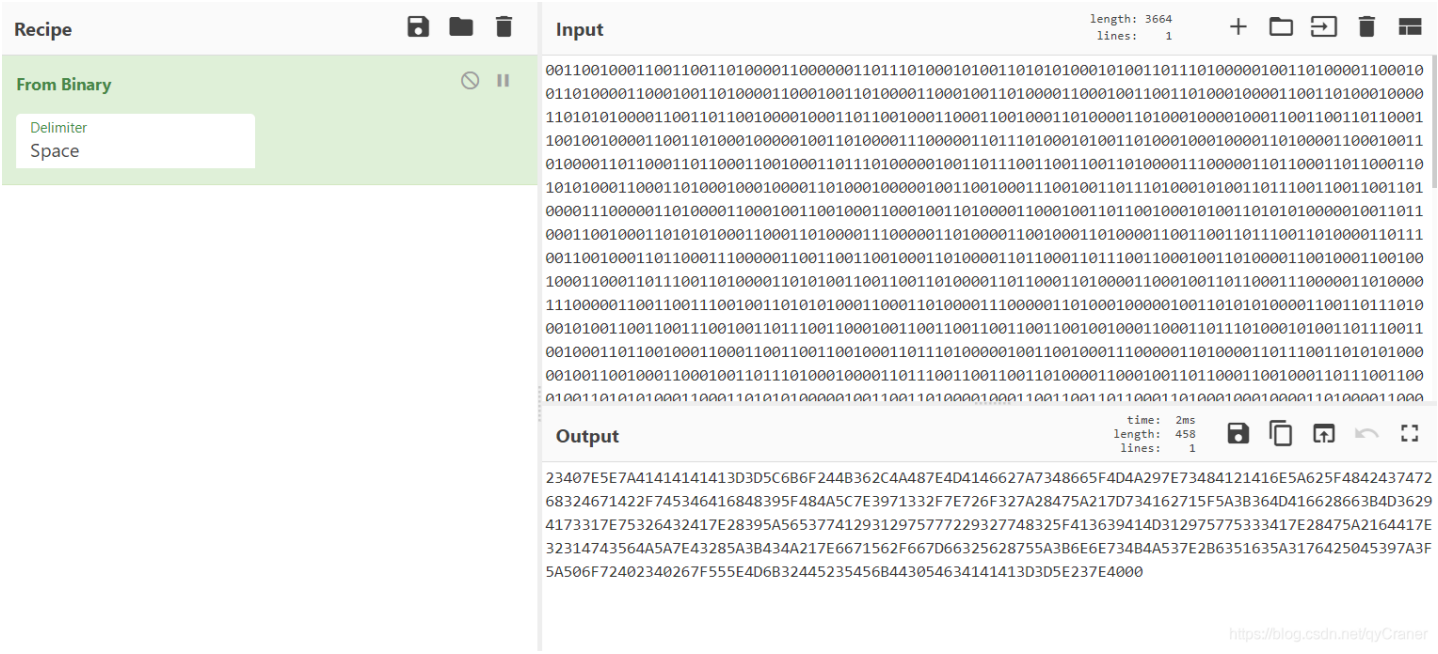
详细资料可见下方链接：

https://blog.csdn.net/jeffchenbiao/article/details/7332863?>

直接上脚本

```
n = 21492
flag = ''
fp = open('1.mp3','rb')
while n < 2543729 :
    fp.seek(n,0)
    n += 384
    read_result = fp.read(1)
    flag += bin(ord(read_result))[-1]
print(flag[::-1]) #经过测试可知最后需要倒序输出
```

将得到的二进制中前面那些很长的00000字符串忽略掉，然后转码一下得到：

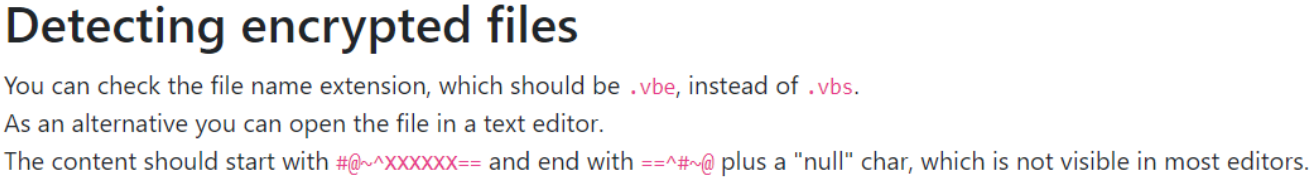



```
23407E5E7A41414141413D3D5C6B6F244B362C4A487E4D4146627A7348665F4D4A297E73484121416E5A625F484243747268324671422F74
5346416848395F484A5C7E3971332F7E726F327A28475A217D734162715F5A3B364D416628663B4D36294173317E75326432417E28395A56
5377412931297577722932774832 5F413639414D312975775333417E28475A2164417E32314743564A5A7E43285A3B434A217E6671562F66
7D66325628755A3B6E6E734B4A537E2B6351635A3176425045397A3F5A506F72402340267F555E4D6B32445235456B44305463414141413D3D
5E237E4000
```

十六进制解码一下，得到：

```
#@~^zAAAAA==\ko$K6,JH~MAFbzsHf_MJ)~sHA!AnZb_HBCtrh2FqB/tSFAhH9_HJ\~9q3/~ro2z(GZ!}sAbq_Z;6MAf(f;M6)As1~u2d2A~(9ZV
SwA)1)uwr)2wH2_A69AM1)uwS3A~(GZ!dA~21GCVJZ~C(Z;CJ!~fqV/f}f2V(uZ;nnsKJS~+cQcZ1vBPE9z?ZPor@#@&.U^Mk2DR5EkD0TcAAA==
^#~@.
```

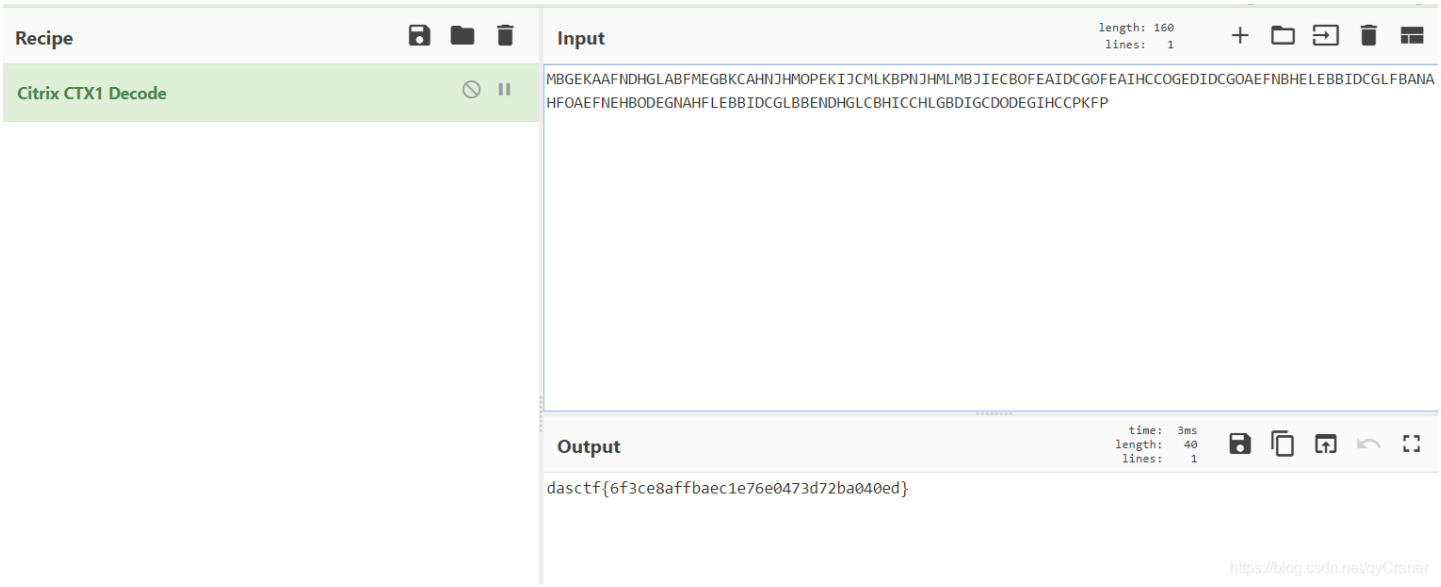
这在西湖论剑的线下赛里出现过，利用google搜索前几个字符也能直接搜到工具，直接放链接：
https://master.ayra.ch/vbs/vbs.aspx



将字符串复制进一个txt中，按照网站要求的格式，修改后缀名为**vbe**，解码后得到一个**vbs**文件。
得到CXT1字符串：

```
MBGEKAAFNDHGLABFMEGBKCAHNJHMOPEKIJCMLKBPNJHMLMBJIECBOFEAIDCGOFEAIHCCOGEDIDCGOAEFNBHELEBBIDCGLFBANAHFOAEFNEHBODEG
NAHFLEBBIDCGLBBENDHGLCBHICCHLGBDIGCDODEGIHCCPKFP
```

最后Citrix CTX1解码一下：



得到flag：

```
dasctf{6f3ce8affbaec1e76e0473d72ba040ed}
```