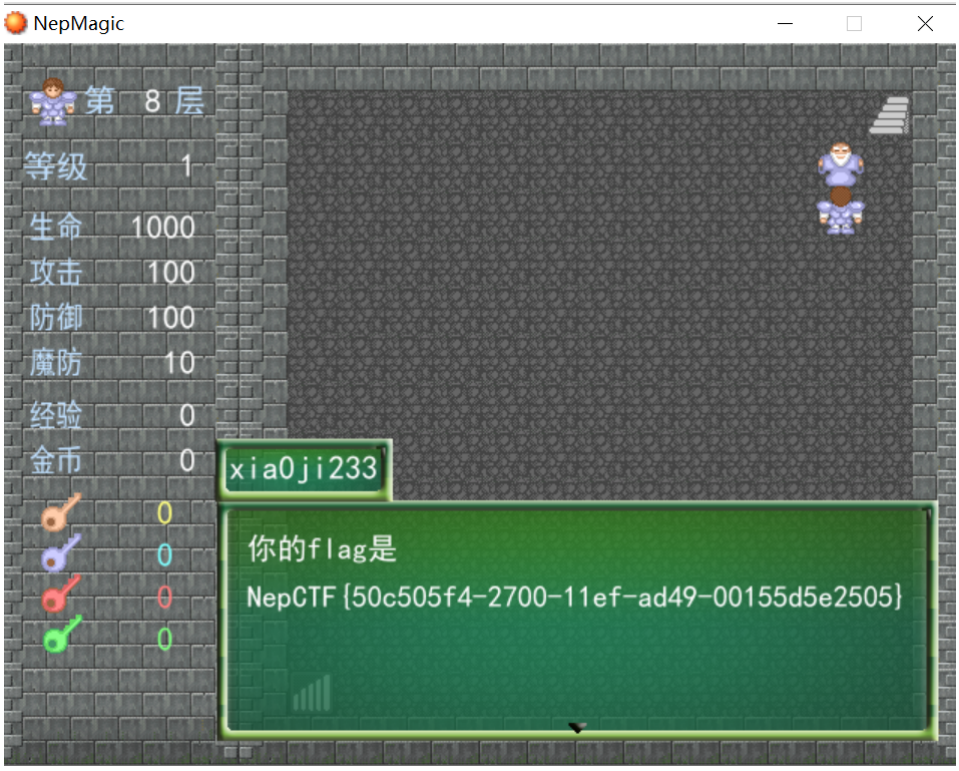
NepCTF2024 St4rr Writeup

> 又双叒叕是差随便一题的分数就能得奖了(T0T)，可惜我是一个菜鸡，只会玩玩misc。。。还有几道题比如DCTris和区块链（这两题花了我一整天都没出）有思路但是没能力实现，赛后看官方wp学习学习吧。

Misc

NepMagic——CheckIn

玩通关就行



Nemophila

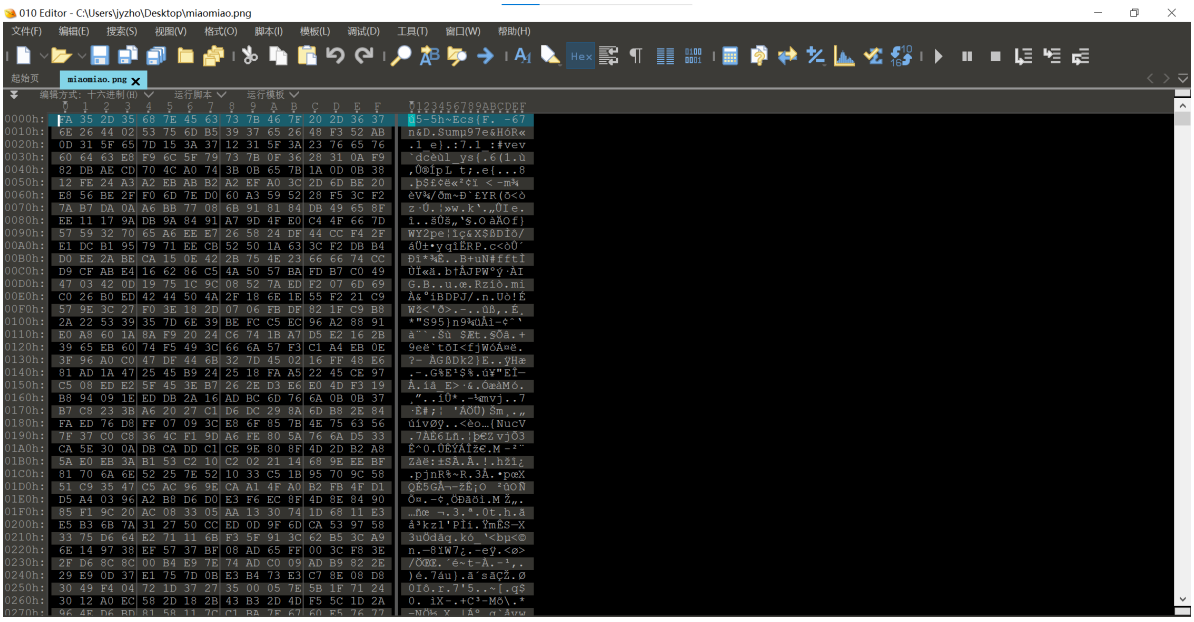
题目mimi.py

```
import base64

print("这里有一个藏宝室，镇守着一个宝箱怪，当你说出正确的口令时，你也就快获得了这个屋子里最至高无上的宝物。")
print("提示：宝箱怪只会提示你口令正确与否，请你试试吧！")
flag = input('Turn in your guess: ')

if len(flag) !=48:
    print("长度不对！")
    exit(1)

if ord(flag.capitalize()[0]) != 83 or not flag[0].islower():
    print("Please try again!")
    exit(1)

if flag[-3:] != "ve}":
```

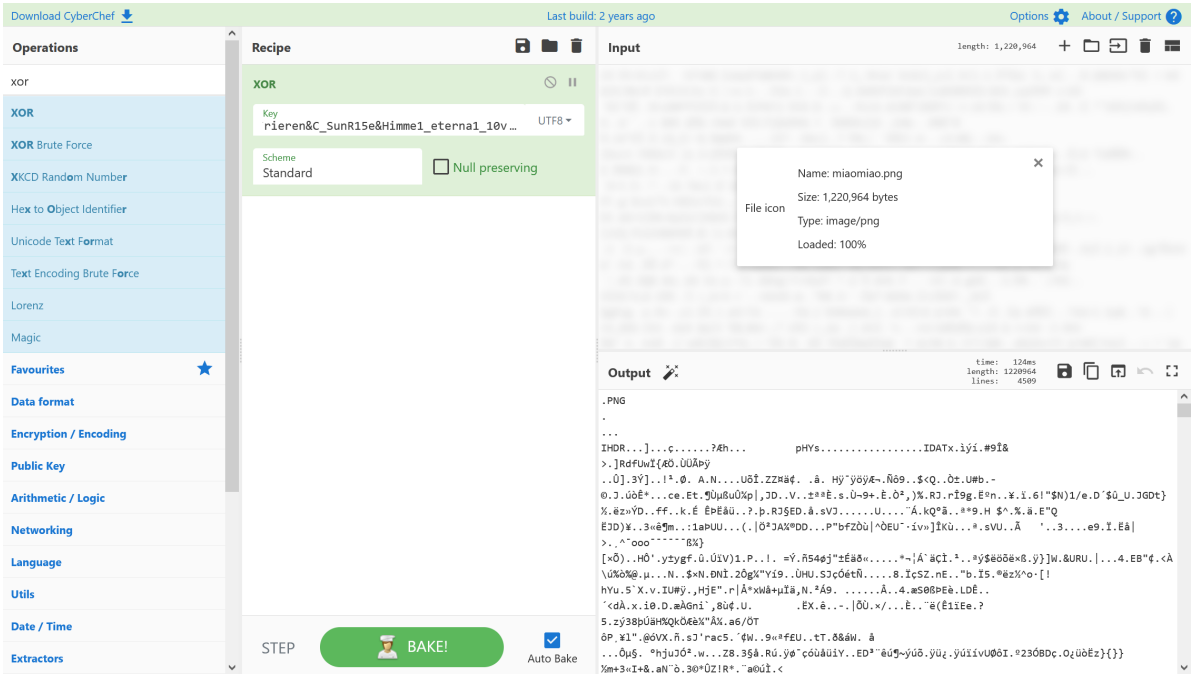
```
    print("Please try again!")
    exit(1)

if flag.count(chr(95)) != 4:
    print("Please try again!")
    exit(1)

if base64.b64encode((flag[10:13]+flag[28:31]).encode('utf-8')).decode() !=
'RnJpSGlt':
    print("Please try again!")
    exit(1)

if int(flag[24:26]) > 10 and int(flag[24:26]) < 20 and pow(int(flag[24:26]),2,5)
!= 0:
    print("好像有点不对！")
    exit(1)

number = flag[33] + flag[41] + flag[43:45]
if int(number) * 9_27 != 1028970 and not number.isnumeric():
    print("还是不对呢！")
    exit(1)

if flag[35:41].replace("e", "1") != "1t1rna":
    print("Please try again!")
    exit(1)

if flag[31:33].swapcase() != "ME":
    print("这不是我!")
    exit(1)

if list(map(len,flag.split("_"))) != [6, 12, 14, 7, 5] and
list(map(len,flag.split("&"))) != [17, 9, 20]:
    print("换个顺序！")
    exit(1)

if ord(min(flag[:2].swapcase())) != 69:
    print("Please try again!")
    exit(1)

if flag[2] + flag[4:6] != "cet4"[:3]:
    print("我不想考四级！")
    exit(1)

new=""
for i in flag[7:10] + flag[18] + flag[26]: new += chr(ord(i) + 1)
if new != "jt|Df":
    print("Please try again!")
    exit(1)

if "SunR" in flag and "eren" in flag:
    print("好像对了！可以先去试试！")
    exit(1)

print("恭喜你~发现了上个世纪的秘密~快去向冒险家协会索要报酬吧！")
```

一条一条对照着解就行，得到secret_is{Frieren&C_SunR15e&Himme1_eterna1_10ve}

这就是压缩包密码，解开压缩包，拿到一个很奇怪的文件miaomiao.png，放进010看一看，没有任何特征



猜测是异或，拿这个文件的开头与png的开头89504e47异或一下可以发现，这个文件应该是被整体疑惑了之前解出来的那个压缩包密码，异或回去得到正常的png文件



放进010看到得到的png存在CRC校验错误，得知需要修改宽高

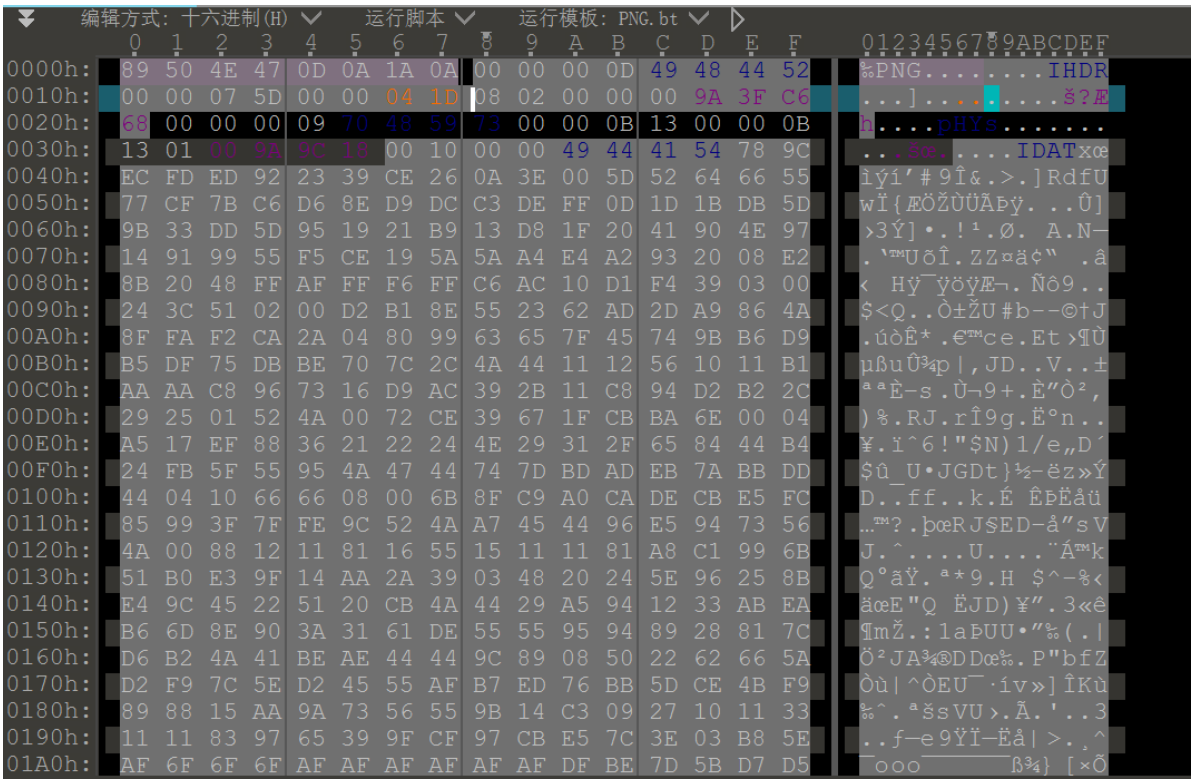
脚本：

```
import binascii
import struct
crcbp = open("download.png", "rb").read()     #填入图片名
crc32frombp = int(crcbp[29:33].hex(),16)
print(crc32frombp)

for i in range(10000):
    for j in range(10000):
        data = crcbp[12:16] + \
            struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
```

```
        # print(crc32)
        if(crc32 == crc32frombp):
            print(i, j)
            print('hex:', hex(i), hex(j))
            exit(0)
```

```
2587870824
1885 1053
hex: 0x75d 0x41d
```

照着改回去即可看到flag



3DNep

不知道什么东西，用trid识别一下

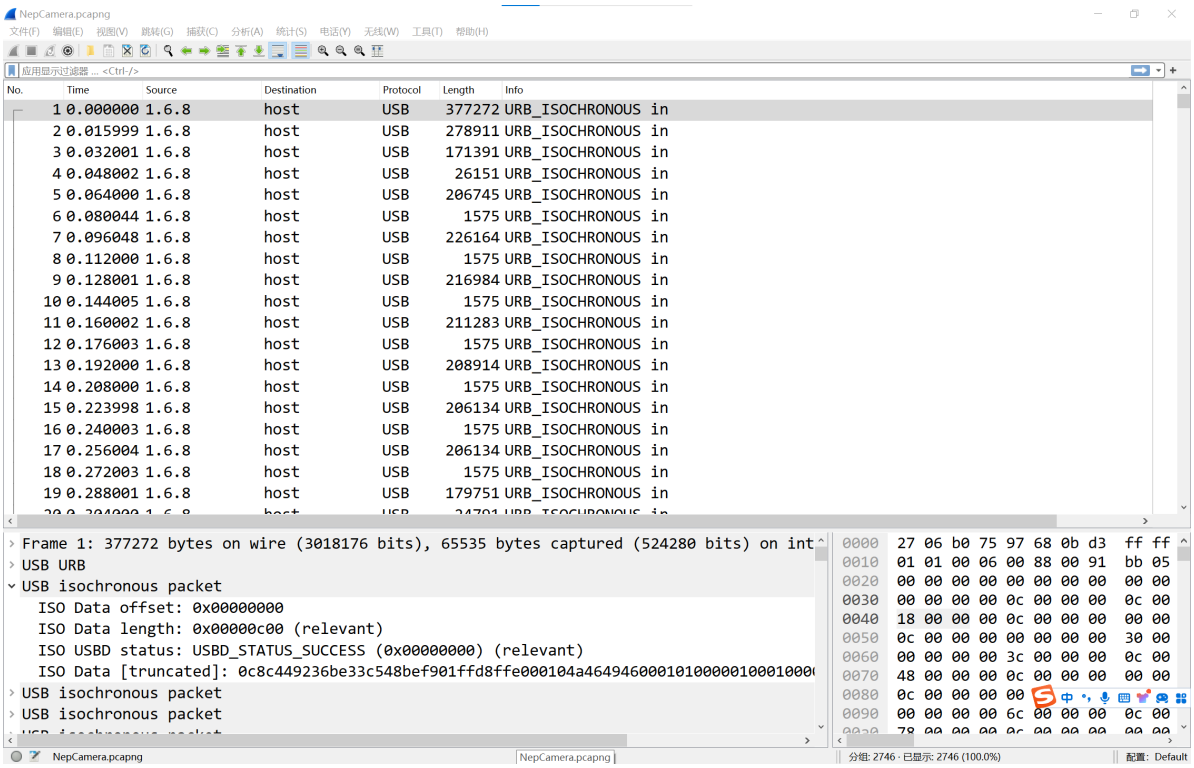
```
C:\Users\jyzho\Desktop\h4ck3r_t0015\trid>trid nepctf

TrID/32 - File Identifier v2.24 - (C) 2003-16 By M.Pontello
Definitions found:  17100
Analyzing...

Collecting data from file: nepctf
100.0% (.GLB) GL Transmission Format Binary (v2.0) (8000/1)
```

发现是个100%纯正的GLB文件，改后缀名为.glb后直接打开就行

看这四个角上的定位点一眼汉信码，扫描就有flag

NepCamera

打开流量包，全都是usb的isochronous流量，在isodata中我们可以看到熟悉的ffd8ffe0，这是jpg的文件头



用tshark把iso data全都提取出来

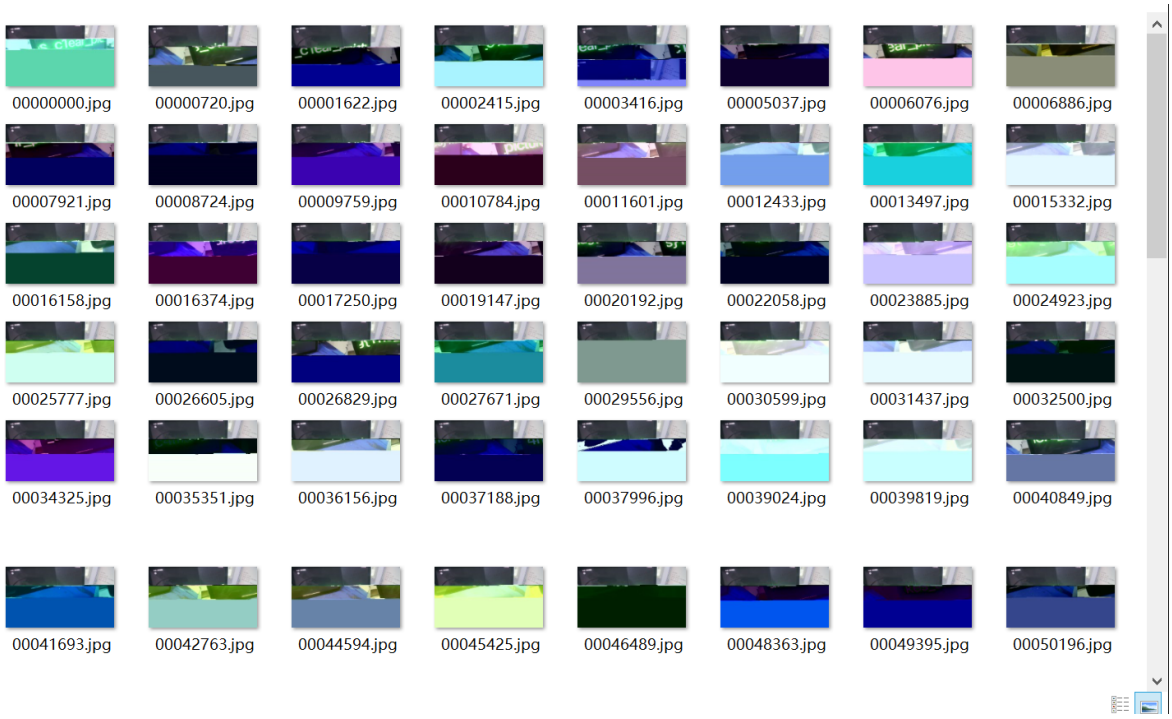
```
tshark -r NepCamera.pcapng -T fields -e usb.iso.data > 1.txt
```

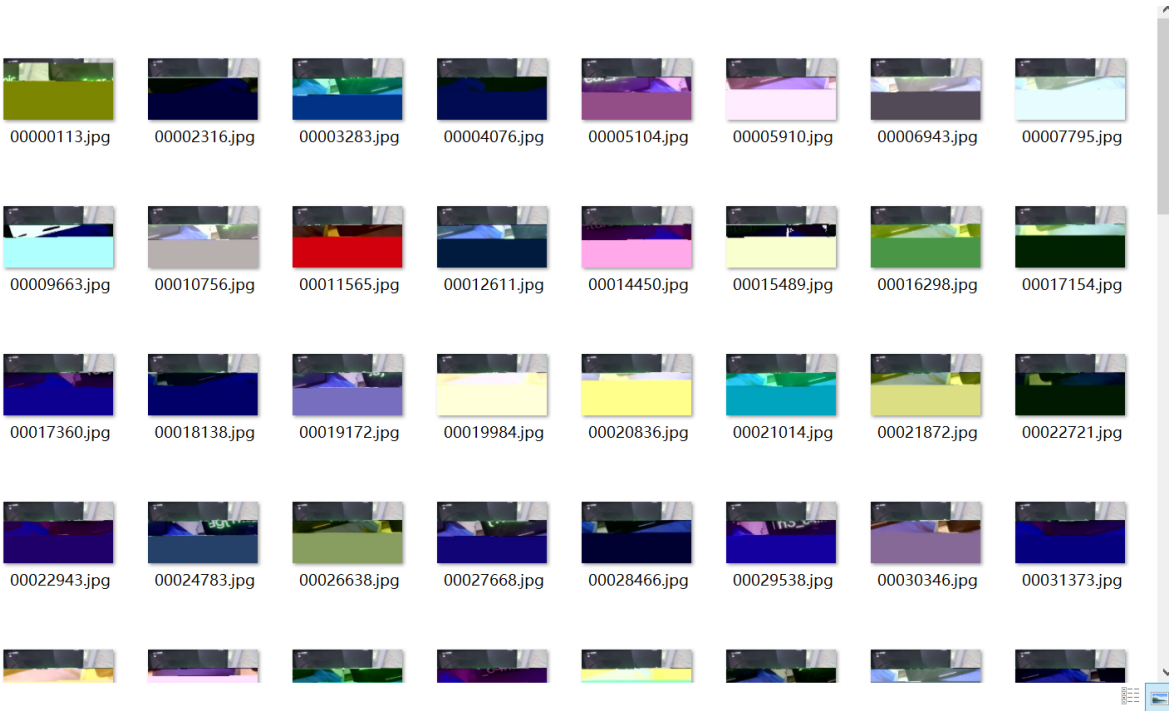
仔细观察后可以发现，iso data总共有四种开头：0c8c、0c8d、0c8e、0c8f，其中0c8e、0c8f开头的都是以ffd9结尾，说明这些是jpg的文件尾。这里是将0c8e的接在0c8c的后面，将0c8f的接在0c8c的后面。写个脚本完成数据处理：

```
f=open('1.txt','r')
ff1=open('output1.jpg','wb')
ff2=open('output2.jpg','wb')
def remove_empty_lines(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        lines = file.readlines()
    non_empty_lines = [line for line in lines if line.strip()]
    with open(file_path, 'w', encoding='utf-8') as file:
        file.writelines(non_empty_lines)
file_path = '1.txt'
remove_empty_lines(file_path)
a=f.readline().strip()
def find_comma_positions(s):
    start = 0
    positions = []
    while True:
        position = s.find(',', start)
        if position == -1:
            break
        positions.append(position)
        start = position + 1
    return positions
while a:
    if a.startswith('0c8c' or '0c8e'):
        pos=find_comma_positions(a)
        if pos==[]:
            tmp=a[24:]
            for j in range(0,len(tmp),2):
                ff1.write(bytes.fromhex(tmp[j:j+2]))
            a=f.readline().strip()
        else:
            l=0
            r=pos[0]
            for i in range(len(pos)-1):
                tmp=a[l:r][24:]
                for j in range(0,len(tmp),2):
                    ff1.write(bytes.fromhex(tmp[j:j+2]))
                l=pos[i]+1
                r=pos[i+1]
            tmp=a[pos[-1]+1:][24:]
            for j in range(0,len(tmp),2):
                ff1.write(bytes.fromhex(tmp[j:j+2]))
            a=f.readline().strip()
    else:
        pos=find_comma_positions(a)
        if pos==[]:
            tmp=a[24:]
            for j in range(0,len(tmp),2):
                ff2.write(bytes.fromhex(tmp[j:j+2]))
            a=f.readline().strip()
        else:
            l=0
            r=pos[0]
            for i in range(len(pos)-1):
                tmp=a[l:r][24:]
                for j in range(0,len(tmp),2):
                    ff2.write(bytes.fromhex(tmp[j:j+2]))
```

```
                l=pos[i]+1
                r=pos[i+1]
            tmp=a[pos[-1]+1:][24:]
            for j in range(0,len(tmp),2):
                ff2.write(bytes.fromhex(tmp[j:j+2]))
            a=f.readline().strip()
f.close()
ff1.close()
ff2.close()
```

对输出的output1.jpg和output2.jpg进行foremost分解，可以得到两组图片





虽然仍然都不是正常的图片，但是已经可以辨认出图片中pad上播放的文字:

flag{Th3_c4mer4_takes_c1ear_pictures}