

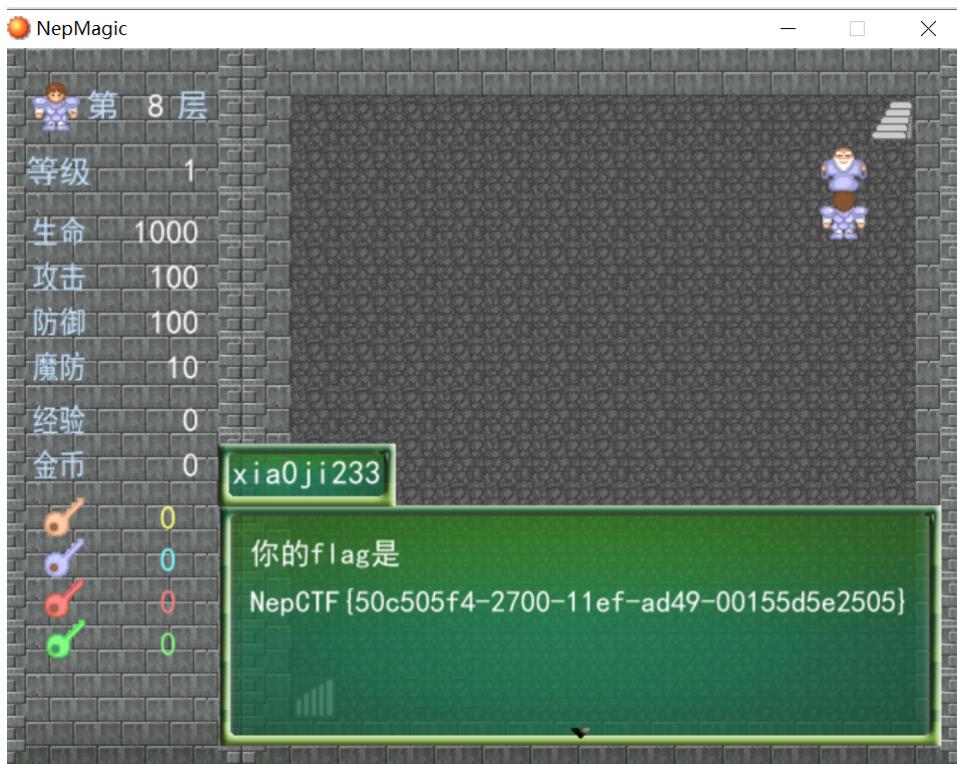
# NepCTF2024 St4rr Writeup

又双叒叕是差随便一题的分数就能得奖了(TOT)，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("_")))[1] != [6, 12, 14, 7, 5] and list(map(len,flag.split("&")))[1] != [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)

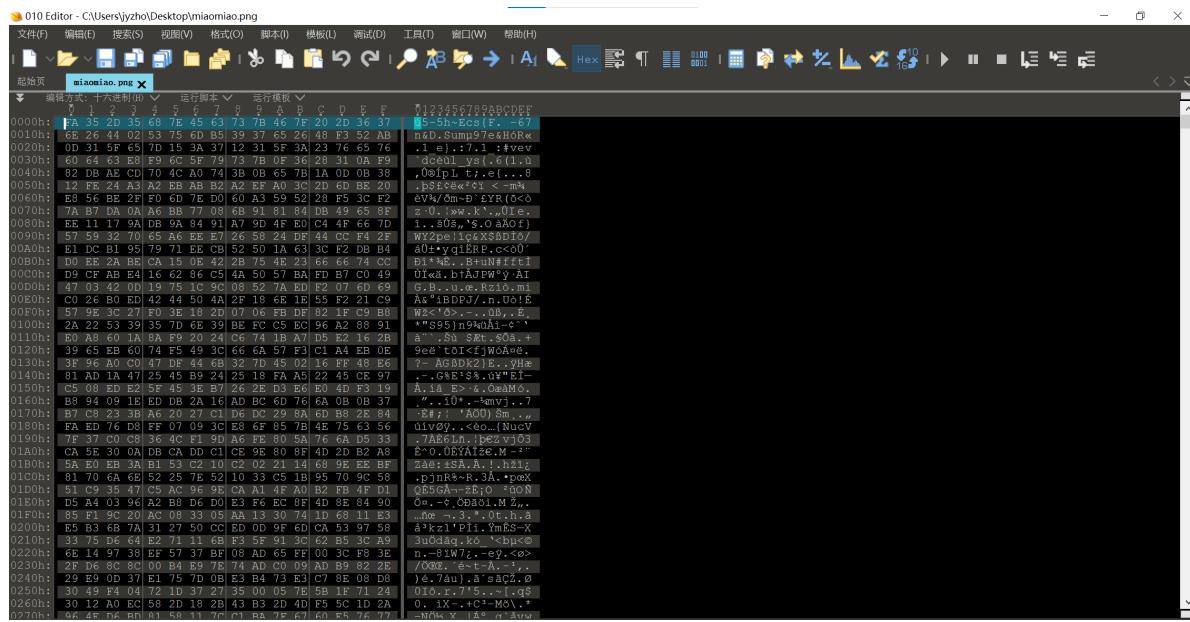
if "SunR" in flag and "eren" in flag:
    print("好像对了! 可以先去试试! ")
    exit(1)

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miao.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

The screenshot shows the CyberChef interface with the following details:

- Operations:** XOR
- Input:** Name: miao.png, File icon, Type: image/png, Length: 1,220,964 bytes, Loaded: 100%
- Output:** Length: 1,220,964 bytes, Lines: 45699
- Recipe:** XOR, Scheme: Standard, Null preserving
- Key:** rieren&C\_SunR15e&Himmel\_1eternal\_10v...
- Options:** Options, About / Support

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

	编辑方式: 十六进制(H)								运行脚本								运行模板: PNG, bt																	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
0000h:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	‰PNG	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	
0010h:	00	00	07	5D	00	00	04	1D	08	02	00	00	00	9A	3F	C6	‰IHDR	...]	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0020h:	68	00	00	00	09	70	48	59	73	00	00	0B	13	00	00	0B	h	....	pHys	....	.	.	.	.	.	.	.	.	.	.	.	.	.	
0030h:	13	01	00	9A	9C	18	00	10	00	00	49	44	41	54	78	9C	....Se,	....	IDAT	xœ	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0040h:	EC	FD	ED	92	23	39	CE	26	0A	3E	00	5D	52	64	66	55	íýí' #9Í&.>.]RdfU	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0050h:	77	CF	7B	C6	D6	8E	D9	DC	C3	DE	FF	0D	1D	1B	DB	5D	wï{ÆÖŽÜÜÄþý...Û]	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0060h:	9B	33	DD	5D	95	19	21	B9	13	D8	1F	20	41	90	4E	97	>3Ý]•..!^ø. A.N-	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0070h:	14	91	99	55	F5	CE	19	5A	5A	A4	E4	A2	93	20	08	E2	.`HÝ~ÝÖÝÈ~. Ñô9..	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0080h:	8B	20	48	FF	AF	FF	F6	FF	C6	AC	10	D1	F4	39	03	00	\$<Q..Ø±ŽU#b--@†J	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0090h:	24	3C	51	02	00	D2	B1	8E	55	23	62	AD	2D	A9	86	4A	.úðÈ*.€™ce.Et>Ù	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
00A0h:	8F	FA	F2	CA	2A	04	80	99	63	65	7F	45	74	9B	B6	D9	µßuÙ³4p ,JD..V..±	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
00B0h:	B5	DF	75	DB	BE	70	7C	2C	4A	44	11	12	56	10	11	B1	„aÈ-s.Ù¬9+.È"Ø²,	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
00C0h:	AA	AA	C8	96	73	16	D9	AC	39	2B	11	C8	94	D2	B2	2C	)%.RJ.rî9g.È°n..	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
00D0h:	29	25	01	52	4A	00	72	CE	39	67	1F	CB	BA	6E	00	04	...™? .þœRJ\$ED-å"sV	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
00E0h:	A5	17	EF	88	36	21	22	24	4E	29	31	2F	65	84	44	B4	¥.í^6!"\$N)1/e,D'	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
00F0h:	24	FB	5F	55	95	4A	47	44	74	7D	BD	AD	EB	7A	BB	DD	\$Ù_U•JGDt}½-ëz»Ý	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0100h:	44	04	10	66	66	08	00	6B	8F	C9	A0	CA	DE	CB	E5	FC	D..ff..k.É ÈPEåü	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0110h:	85	99	3F	7F	FE	9C	52	4A	A7	45	44	96	E5	94	73	56	...™? .þœRJ\$ED-å"sV	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0120h:	4A	00	88	12	11	81	16	55	15	11	11	81	A8	C1	99	6B	J.^....U...."Å™k	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0130h:	51	B0	E3	9F	14	AA	2A	39	03	48	20	24	5E	96	25	8B	Q°ãÝ. ^*9.H \$^-%<	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0140h:	E4	9C	45	22	51	20	CB	4A	44	29	A5	94	12	33	AB	EA	äœE "Q ÈJD) ¥". 3«ê	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0150h:	B6	6D	8E	90	3A	31	61	DE	55	55	95	94	89	28	81	7C	Ímž.:1aÈUU•"‰(.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0160h:	D6	B2	4A	41	BE	AE	44	44	9C	89	08	50	22	62	66	5A	Ö²JA¾®DDœ‰. P"bfZ	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0170h:	D2	F9	7C	5E	D2	45	55	AF	B7	ED	76	BB	5D	CE	4B	F9	Ðù ^ÐEU^-iv»] îKù	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0180h:	89	88	15	AA	9A	73	56	55	9B	14	C3	09	27	10	11	33	%^..^sSVU>.Ã.'..3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0190h:	11	11	83	97	65	39	9F	CF	97	CB	E5	7C	3E	03	B8	5E	..f-e9ÝI-Èå >..^	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
01A0h:	AF	6F	6F	6F	AF	AF	AF	AF	AF	AF	DF	BE	7D	5B	D7	D5	-ooo-----ß³4} [×Ó	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.



# 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



## 扫描结果

扫描内容 NepCTF{6e766b59-23d1-395c26d70  
8a4}

码制 HANXIN

### 条码知识

汉信码是由中国物品编码中心研制开发，是我国第一个制定了国家标准的自主知识产权的二维码，具有知识产权免费、汉字编码能力强、抗污损、抗畸变、信息容量大等特点。2007年8月23日，国家标准化管理委员会发布了GB/T 21049《汉信码》国家标准。和其他二维码相比，汉信码更适合汉字信息的表示，其支持GB 18030中规定的160万个汉字信息字符，具有高度的汉字表达能力和汉字压缩效率；具有很强的纠错能力、抗污损和畸变能力，支持加密技术。

## 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

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000 1.6.8	host	USB	URB_ISOCCHRONOUS in	377272	
2	0.015999 1.6.8	host	USB	URB_ISOCCHRONOUS in	278911	
3	0.032001 1.6.8	host	USB	URB_ISOCCHRONOUS in	171391	
4	0.048002 1.6.8	host	USB	URB_ISOCCHRONOUS in	26151	
5	0.064000 1.6.8	host	USB	URB_ISOCCHRONOUS in	206745	
6	0.080044 1.6.8	host	USB	URB_ISOCCHRONOUS in	1575	
7	0.096048 1.6.8	host	USB	URB_ISOCCHRONOUS in	226164	
8	0.112000 1.6.8	host	USB	URB_ISOCCHRONOUS in	1575	
9	0.128001 1.6.8	host	USB	URB_ISOCCHRONOUS in	216984	
10	0.144005 1.6.8	host	USB	URB_ISOCCHRONOUS in	1575	
11	0.160002 1.6.8	host	USB	URB_ISOCCHRONOUS in	211283	
12	0.176003 1.6.8	host	USB	URB_ISOCCHRONOUS in	1575	
13	0.192000 1.6.8	host	USB	URB_ISOCCHRONOUS in	208914	
14	0.208000 1.6.8	host	USB	URB_ISOCCHRONOUS in	1575	
15	0.223998 1.6.8	host	USB	URB_ISOCCHRONOUS in	206134	
16	0.240003 1.6.8	host	USB	URB_ISOCCHRONOUS in	1575	
17	0.256004 1.6.8	host	USB	URB_ISOCCHRONOUS in	206134	
18	0.272003 1.6.8	host	USB	URB_ISOCCHRONOUS in	1575	
19	0.288001 1.6.8	host	USB	URB_ISOCCHRONOUS in	179751	
20	0.304000 1.6.8	host	USB	URB_ISOCCHRONOUS in	27701	

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]))
            ff2.write(bytes.fromhex(tmp[j:j+2]))

```

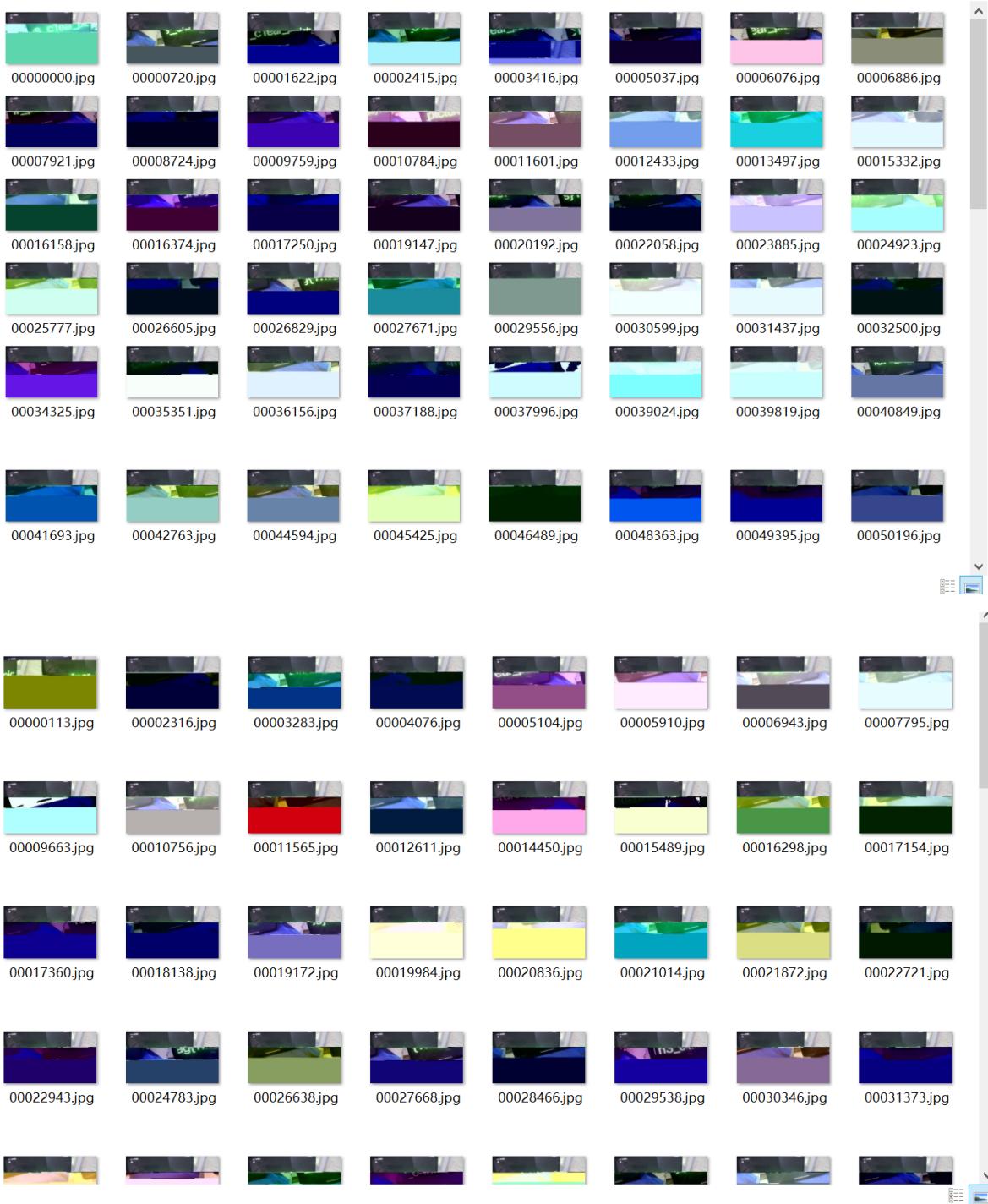
```

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}

