ctf.show misc入门 21~40

原创

ThnPkm 于 2022-01-23 23:42:20 发布 359 收藏
分类专栏： 刷题 wp 文章标签： ctf misc 安全

目录

---

misc21

Hint：flag在序号里。

Exif版本	0232
ComponentsConfiguration	Y, Cb, Cr, -
SecurityClassification	Top Secret
Flashpix版本	0100
色彩空间	Uncalibrated
序列号	686578285826597329

得到信息hex（X&Ys），hex（）是转16进制

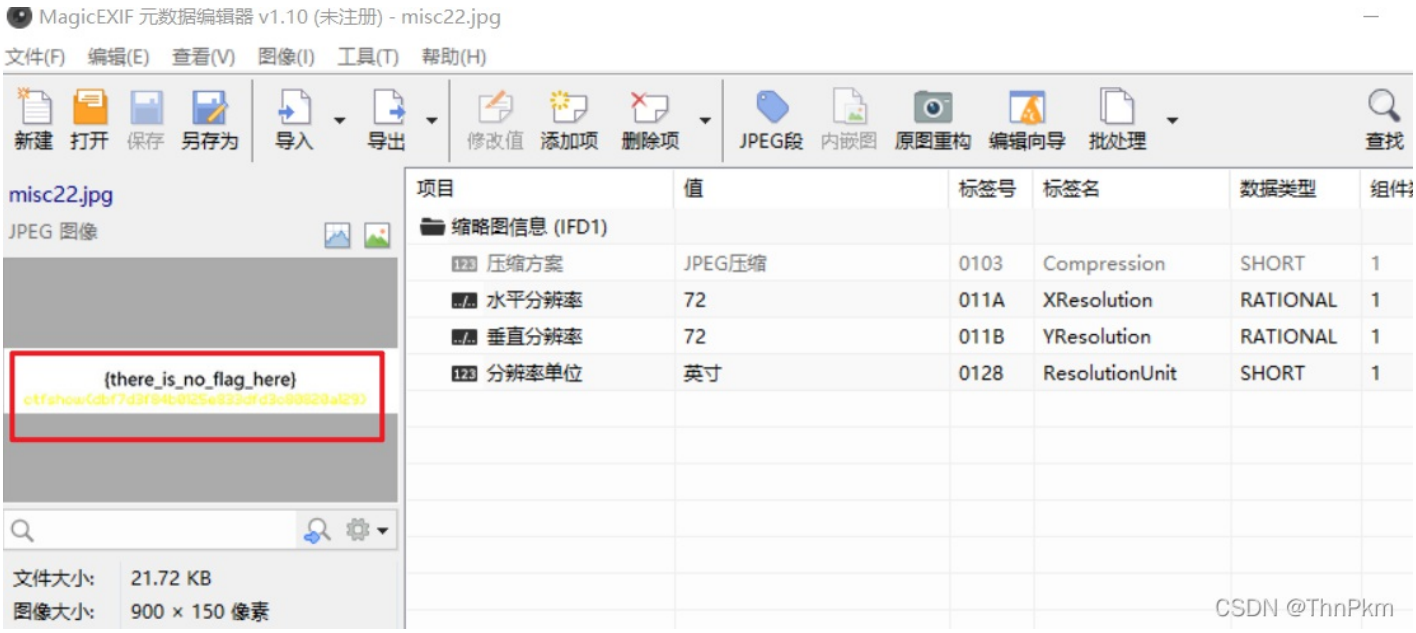
X分辨率	3902939465
Y分辨率	2371618619
PageName	https://ctf.show/
X定位	1082452817
Y定位	2980145261
目标Printer	ctfshow{}

上面四段分别10进制转16进制 合起来 就是flag

misc22

Hint:flag在图片

涉及到我没用过的工具magicexif



直接打开就是 不过黄色的看起来很糊

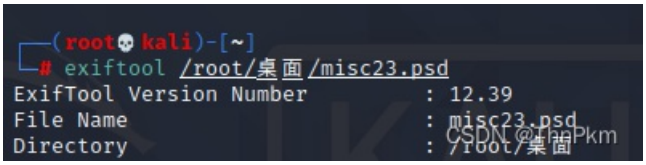
ctfshow{dbf7d3f84b0125e833dfd3c80820a129}

misc23

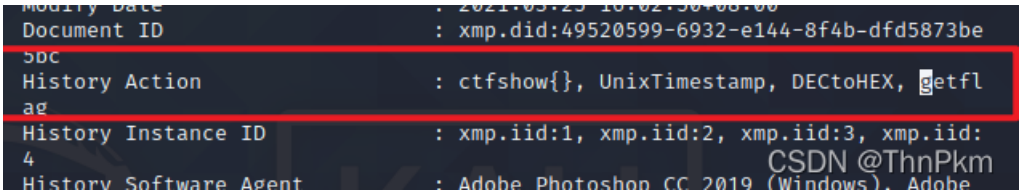
Hint:flag在时间里。

又是新的知识点

使用kali中的exiftoor

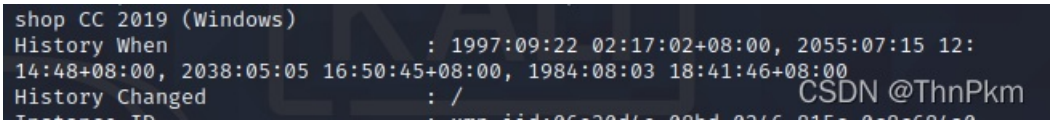


在这里获得信息，Timestamp指的是时间戳，DECtoHEX是十进制转十六进制



时间戳(Unix timestamp)转换工具 - 在线工具 (tool.lu)

用这个网站来获取时间戳，

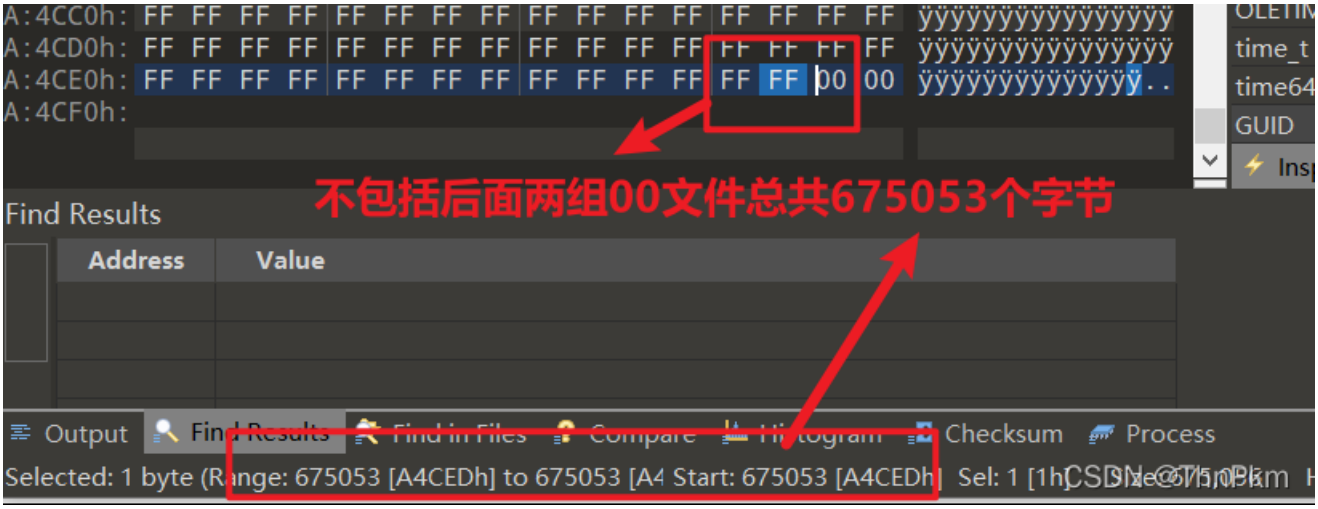


再将十进制数转为16进制的，把上面的4组时间戳都这样搞就行了 ，最后拼接

misc24

Hint:flag在图片上面

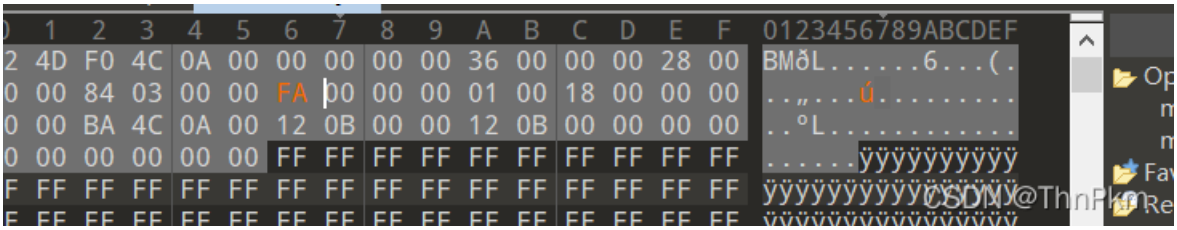
给了个bmp文件，是图片高度的问题，但我试了瞎改还不行



bmg文件计算图片宽高 文件头占了53个字节,一个像素由3个字节构成

像素数 = （675053-53）/ 3 =

我们知道宽是900，所以正确高是 225000/900=250, 十六进制是 FA



就出来了

misc25

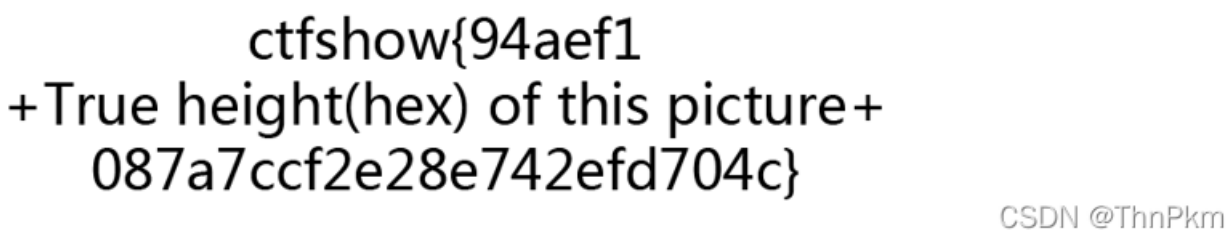
Hint：flag在图片下面

这是一个png文件 ，010随便改高一点就出来了

misc26

Hint：flag还是在图片下面，但到底有多下面？。

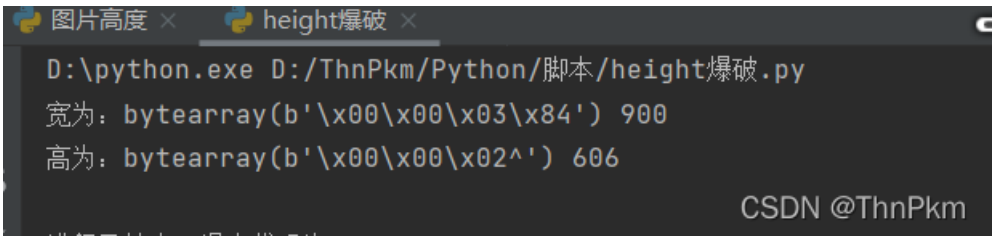
010改高度后发现，flag条件 需要知道真实的高度



脚本爆破一下真实高度，只需将其中的crc替换即可

```
import zlib
import struct

# 同时爆破宽度和高度
filename = "C:/Users/达/Desktop/misc26.png"
with open(filename, 'rb') as f:
    all_b = f.read()
    data = bytearray(all_b[12:29])
    n = 4095
    for w in range(n):
        width = bytearray(struct.pack('>i', w))
        for h in range(n):
            height = bytearray(struct.pack('>i', h))
            for x in range(4):
                data[x+4] = width[x]
                data[x+8] = height[x]
            crc32result = zlib.crc32(data)
            #替换成图片的crc
            if crc32result == 0xEC9CCBC6:
                print("宽为：", end = '')
                print(width, end = ' ')
                print(int.from_bytes(width, byteorder='big'))
                print("高为：", end = '')
                print(height, end = ' ')
                print(int.from_bytes(height, byteorder='big'))
```

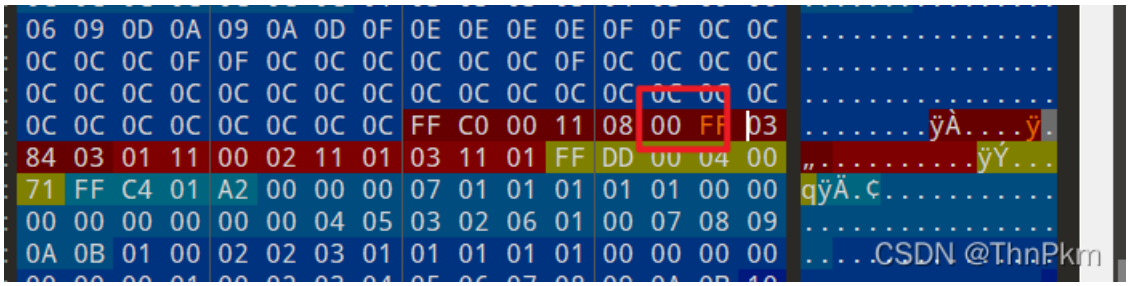


16进制没有体现好， 606转16进制是25e

misc27

Hlnt:flag在图片下面

找到正确的地方修改即可

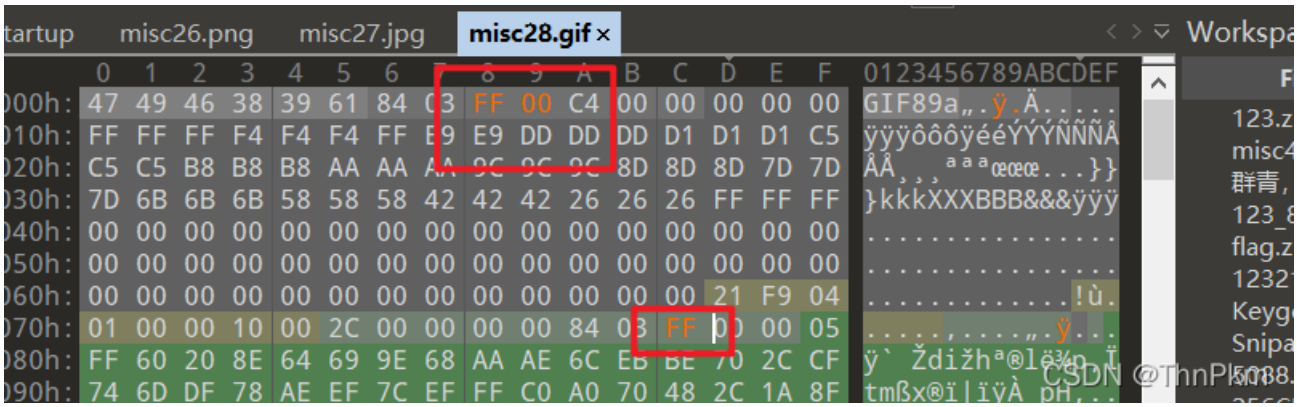


misc28

Hint：**flag在图片下面**

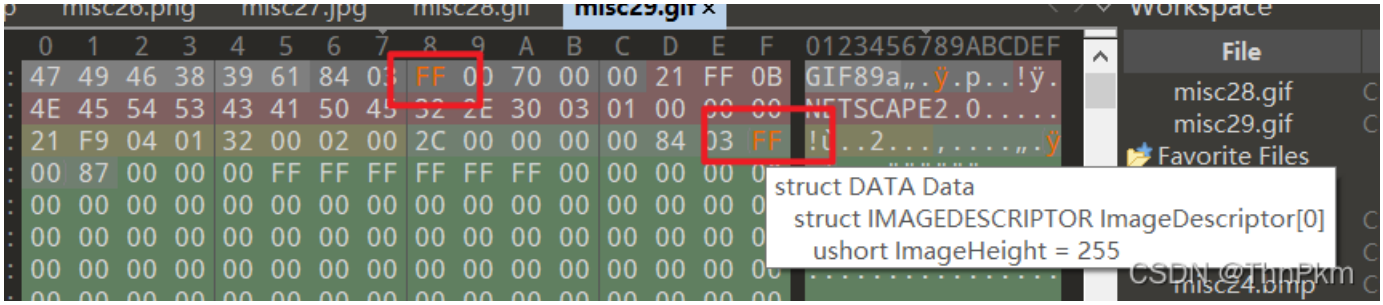
是一个gif 比较懵

gif的每一帧都有宽高所以修改的地方不止一处,



misc29

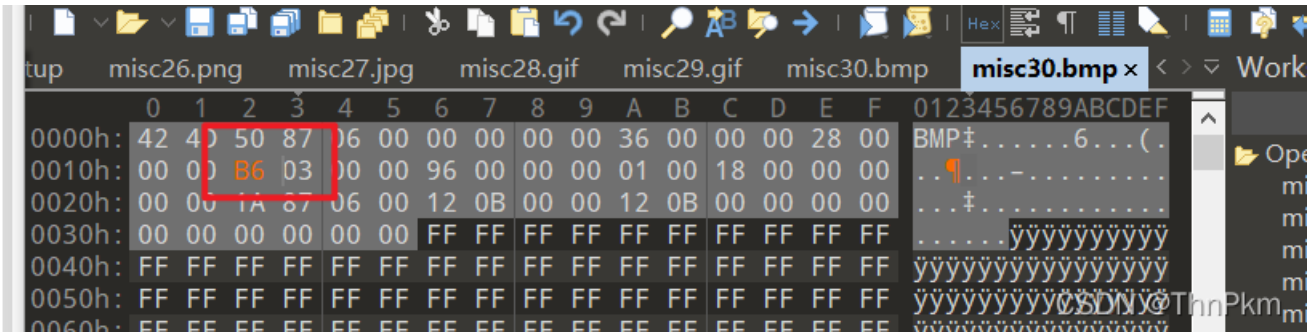
与上一题一样的思路，不过要改很多个 把每一帧都改掉，因为某一帧会出现flag



misc30

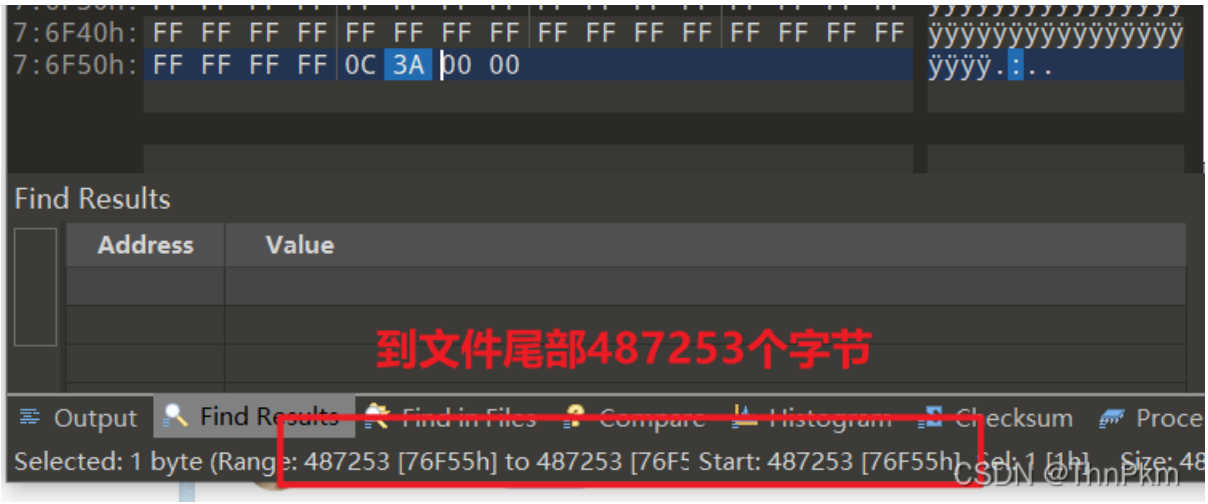
Hint：正确的宽度是950。

950转为16进制是3B6
010修改宽，注意写法



misc31

Hint：高度是正确的，但正确的宽度是多少呢。



bmg文件计算图片宽高 文件头占了53个字节,一个像素由3个字节构成

像素数 = （487253-53）/ 3 =162400

我们知道高是150，所以正确宽是 162400/150=1082, 十六进制是 43A

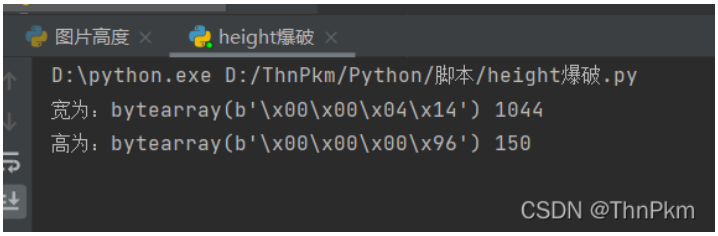
misc32

Hint：高度是正确的，但正确的宽度是多少呢

这题是png ，要用脚本爆破，还是msic26的脚本

```
import zlib
import struct

# 同时爆破宽度和高度
filename = "C:/Users/达/Desktop/misc32.png"
with open(filename, 'rb') as f:
    all_b = f.read()
    data = bytearray(all_b[12:29])
    n = 4095
    for w in range(n):
        width = bytearray(struct.pack('>i', w))
        for h in range(n):
            height = bytearray(struct.pack('>i', h))
            for x in range(4):
                data[x+4] = width[x]
                data[x+8] = height[x]
            crc32result = zlib.crc32(data)
            #替换成图片的crc
            if crc32result == 0xE14A4C0B:
                print("宽为：", end = '')
                print(width, end = ' ')
                print(int.from_bytes(width, byteorder='big'))
                print("高为：", end = '')
                print(height, end = ' ')
                print(int.from_bytes(height, byteorder='big'))
```



正确修改即可

misc33

Hint：出题人丧心病狂，把高度也改了

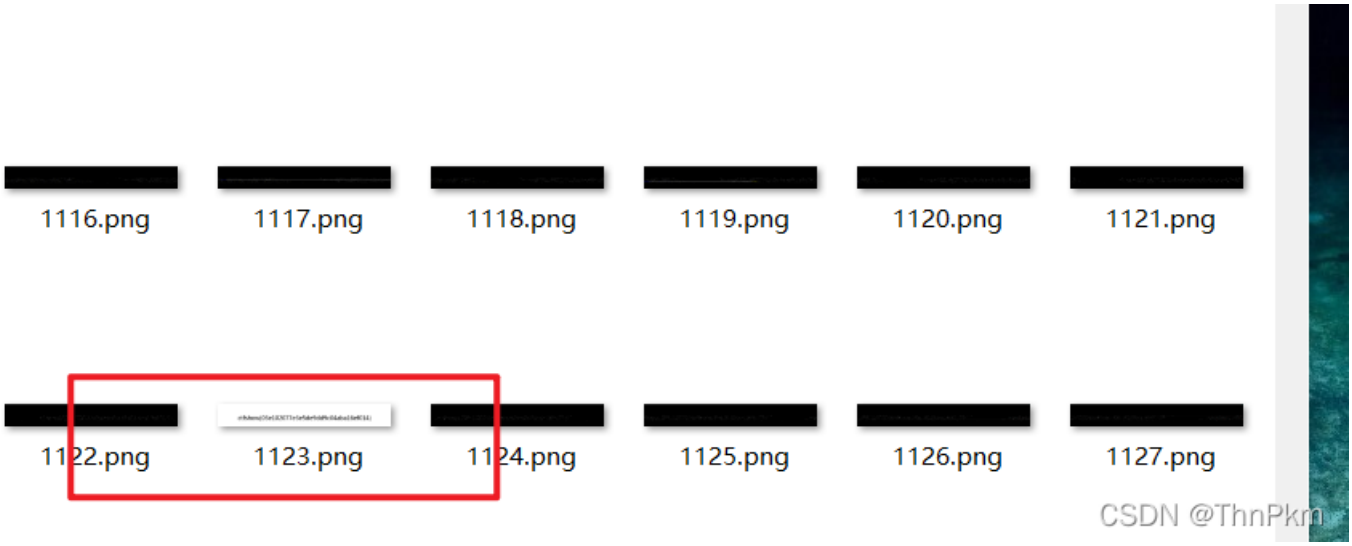
与上一题一样的脚本，修改图片路径 crc

misc34

Hint：出题人狗急跳墙，把IHDR块的CRC也改了，但我们知道正确宽度肯定大于900

也借用大佬的脚本

```
import zlib
import struct
filename = "C:/Users/达/Desktop/misc34.png"
with open(filename, 'rb') as f:
    all_b = f.read()
    for i in range(901,1200):
        name = str(i) + ".png"
        f1 = open(name,"wb")
        im = all_b[:16]+struct.pack('>i',i)+all_b[20:]  #w = all_b[16:20]
        f1.write(im)                                    #h = all_b[20:24]
        f1.close()
```



生成很多 png，拿肉眼看哪个是正常的

misc35

Hint：出题人负隅顽抗，但我们知道正确宽度肯定大于900

依然与上题一样，但是要先将高度拉长点，要能看全乱码

```
import zlib
import struct
filename = "C:/Users/达/Desktop/misc35.jpg"
with open(filename, 'rb') as f:
    all_b = f.read()
    #w = all_b[159:161]
    #h = all_b[157:159]
    for i in range(901,1200):
        name = str(i) + ".jpg"
        f1 = open(name,"wb")
        im = all_b[:159]+struct.pack('>h',i)+all_b[161:]
        f1.write(im)
        f1.close()
```

大概994左右

misc36

Hint：出题人坦白从宽，正确的宽度在920-950

gif，先修改高度，继续脚本遍历

```
import zlib
import struct
filename = "C:/Users/达/Desktop/misc36.gif"
with open(filename, 'rb') as f:
    all_b = f.read()
    for i in range(920,951):
        name = str(i) + ".gif"
        f1 = open(name,"wb")
        im = all_b[:38]+struct.pack('>h',i)[::-1]+all_b[40:]
        f1.write(im)
        f1.close()
```

正确宽度941

misc37

Hint：flag在图片里

gif 逐帧查看即可

misc38

Hint：flag在图片里

一个png，打开跟gif一样会动，（我是用honeyview打开得，也可以逐帧看）

与上题一样

misc39

Hint：flag就像水，忽快忽慢地流

一个 gif， 287 帧图片，但是并没有 flag 的内容。

又是新的知识点

这题是 gif 帧数间隔隐写，就是动图每一帧播放的速度时间都是不一样的。
需要用到kali的工具 identify ，
安装命令：

```
sudo apt-get install imagemagick
```

提取命令：identify -format "%T " misc39.gif > 1.txt

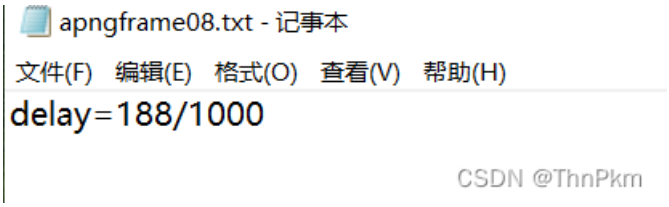
打开 1.txt 里面一串 36 和 37 的数字，把 37 换成1、36 换成 0，就得到长度为 287 的二进制字符串。
然后 flag 有 41 个字符，287/41 = 7 ，所以把每 7 位转一个字符。（正常是 8 位二进制转一个字符）
这里用代码实现：

```
s="11000111110100110011011100111101000110111111101111111011011010101100100111000011000101100101100110110011
flag=""
for i in range(41): #287//7
    flag += chr(int(s[7*i:7*(i+1)],2))
print(flag)
```

misc40

Hint：flag就像歌，有长有短仿佛岁月悠悠

这里用到一个工具 APNG Disassembler 。
APNG图片分解器(APNG Disassembler)是一个用来分解APNG图片的软件，使用这个工具你可以把APNG动画图片中的每一帧都分解出来，并且把帧导出保存为图片文件。
工具处理图片后，除了每一帧图片外，每张图片还带有一个 txt 文件，



大佬说前28个信息没用，用 python 代码脚本提取每个 txt 文件的内容。（从每个 txt 文本中的第七个字符开始取）

```
flag=""
for i in range(28,69): #flag内容从28位开始
    f = open('C:/Users/达/Desktop/apngframe'+str(i)+'.txt')
    s = f.read()
    flag += chr(int(s.split("/")[0][6:]))
print(flag)
```