

首先看见是一个png图片，有三个提示reverse反转 Gopher esolang是一种编码 Bitmap RGBA bmp图片和RGBA

png图片就先拖到zsteg扫一下试试

```
(root@xxxzyhxxx)-[~/桌面]
# zsteg -a '/var/run/vmblock-fuse/blockdir/LTGwwy/flag.png'
imagedata .. text: "R^TrdXB2*"
b1,g,lsb,xy .. file: Zip archive data, at least v5.1 to extract, compression method=AES Encrypted
b1,g,msb,xy .. text: "9ea^L3KT;^"
b2,g,lsb,xy .. text: "#bWPUz[6"
b2,b,lsb,xy .. file: VISX image file
b3p,rgb,msb,xy .. file: OpenPGP Secret Key
b3p,bgr,lsb,xy .. text: ":8:2:=89"
b3p,bgr,msb,xy .. file: OpenPGP Secret Key
b4,r,lsb,xy .. text: "P3TEB#Be$U"
b4,r,msb,xy .. text: " fB\"b6D\"F\"b$ddD"
b4,g,msb,xy .. text: "s3s53S7=W"
```

有个压缩包，导出来

```
(root@xxxzyhxxx)-[~/桌面]
# zsteg -e b1,g,lsb,xy '/var/run/vmblock-fuse/blockdir/aRjQUc/flag.png' -> 23.zip
```

需要密码并且显示损坏，用winrar修复一下

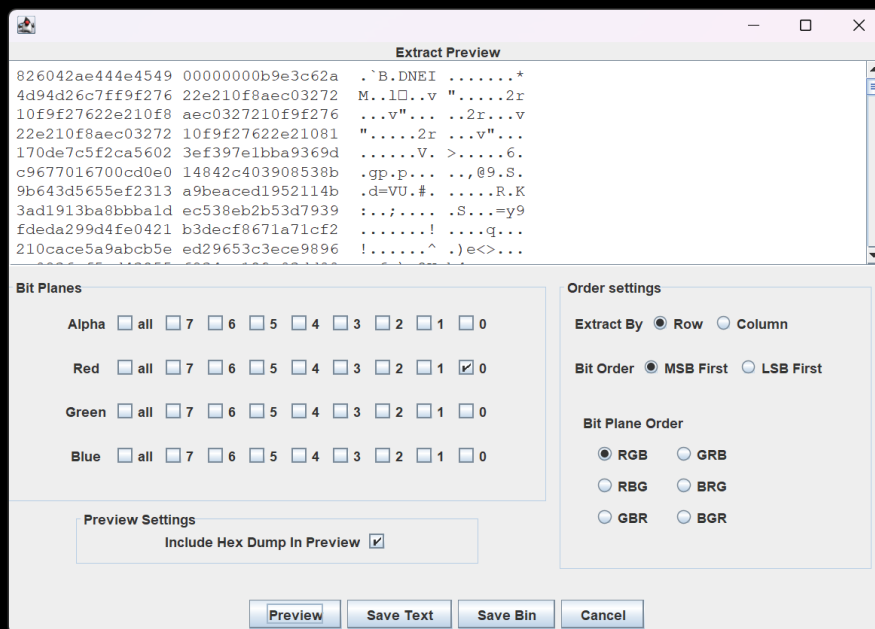


然后拿stegsolve继续分析

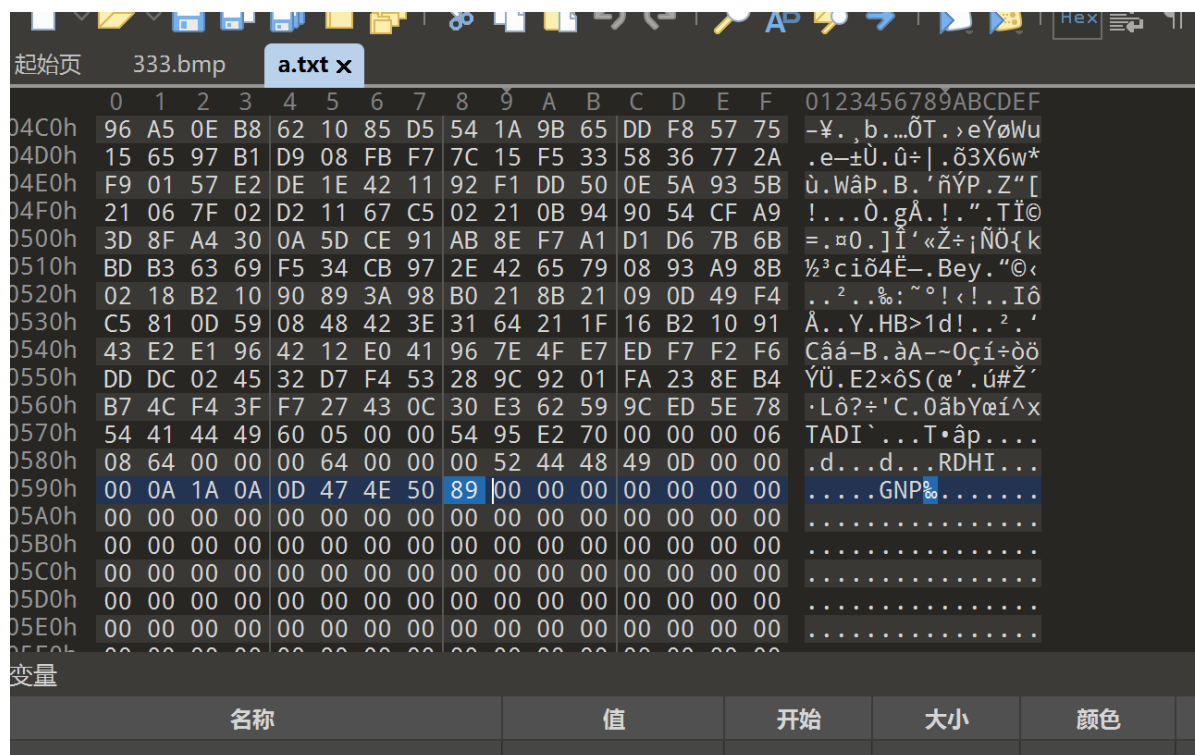
发现在红色最低通道只有这种斑点条纹带，说明这个通道是被修改过的



果然在这里发现了png文件尾的字符IEND而且是反转的，和第一个提示对上了，把它提取出来



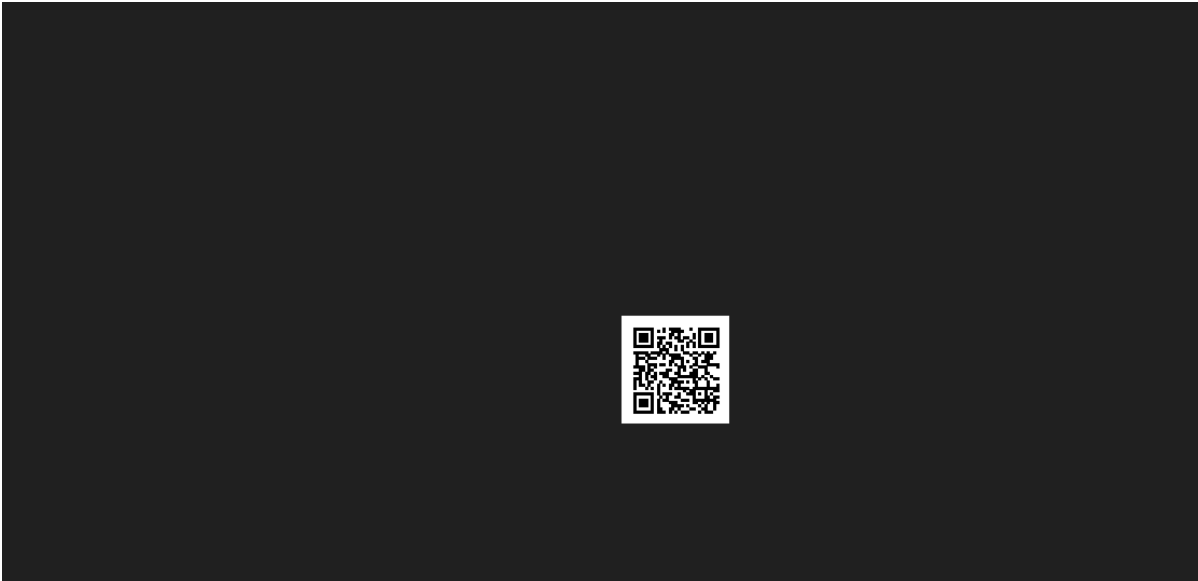
把这些16进制放到010分析一下，文件头尾是反着的，删掉下面多余的0000



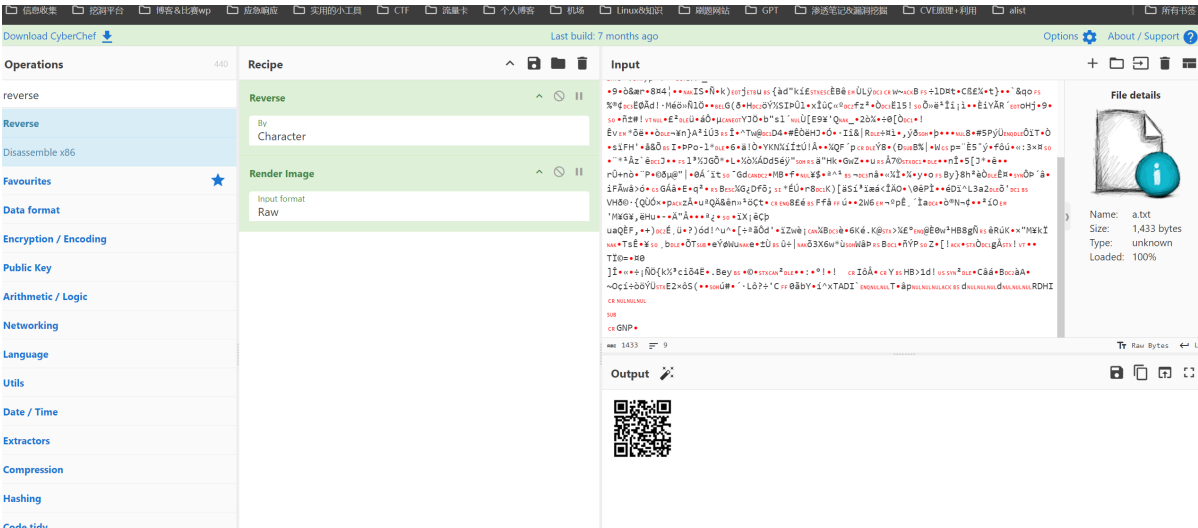
写个脚本逆序一下

```
input = open('a.txt', 'rb')
input_all = input.read()
ss = input_all[::-1]
output = open('1.png', 'wb')
output.write(ss)
input.close()
output.close()
```

得到第一个二维码



cyberchef更方便



扫码得到

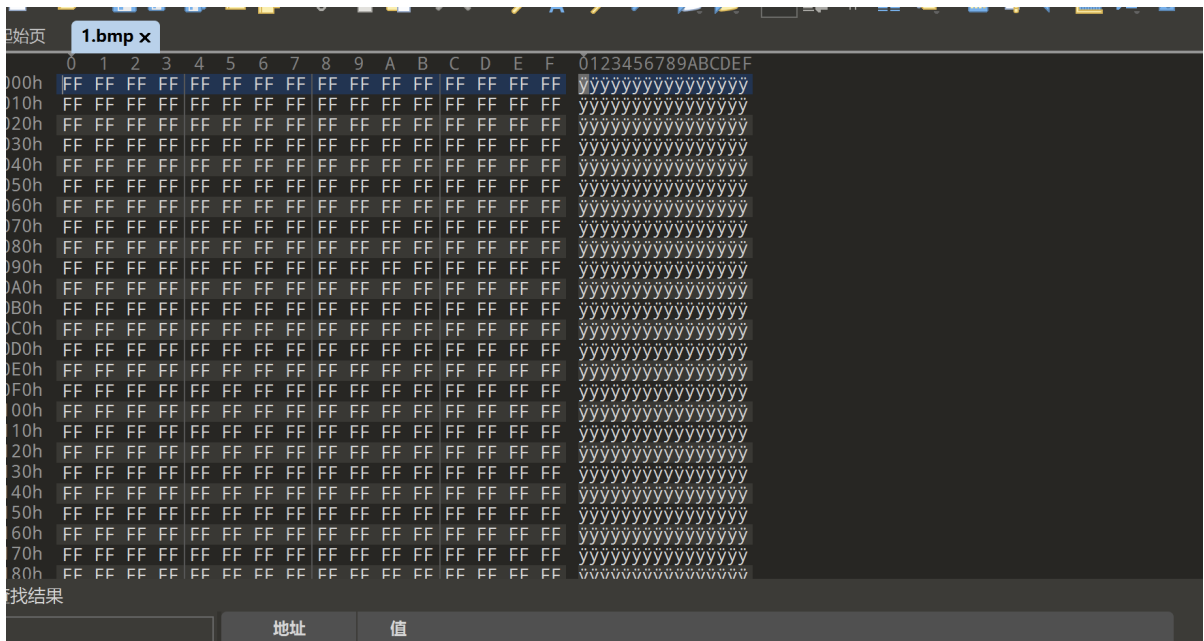
key_part1:0d000721_114514_1919810_

继续分析



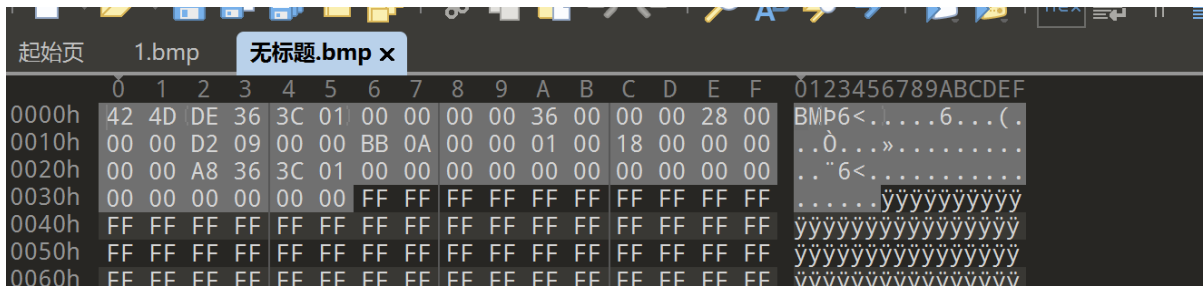
蓝色最低位也有问题

导出save bin

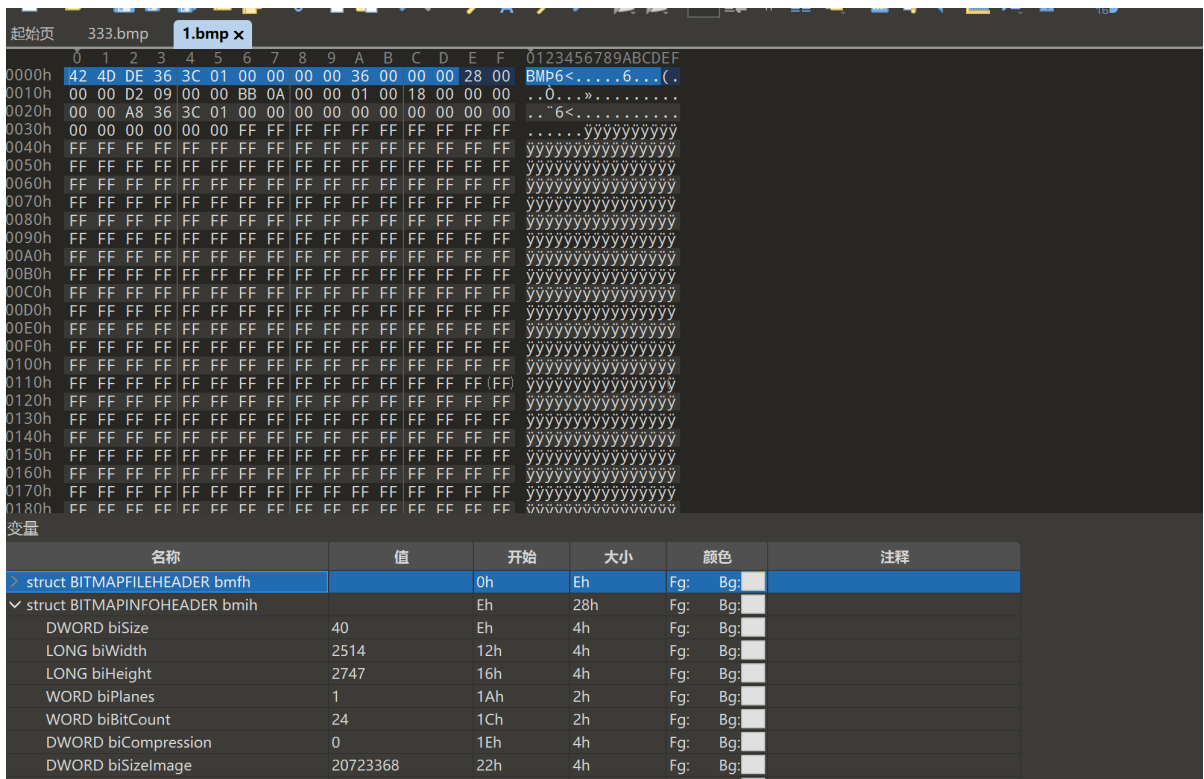


得到这个，这时候需要用提示三了，删掉后面多余的0000（结尾要留四个0000，这是bmp文件的格式）
补上bmp文件头，这里我拿图画生成了一个bmp图片

把它的头部复制到刚才的文件



找到bmih



第三个提示就是bitmap转成RGBA，这里的位深度是24，对应RGB三原色通道，一个通道是8位，RGBA比RGB多了个Alpha的色彩空间也占有8位，所以我们要把这个24修改成32

<https://blog.csdn.net/mochu777777/article/details/120773359>

根据这篇文章公式

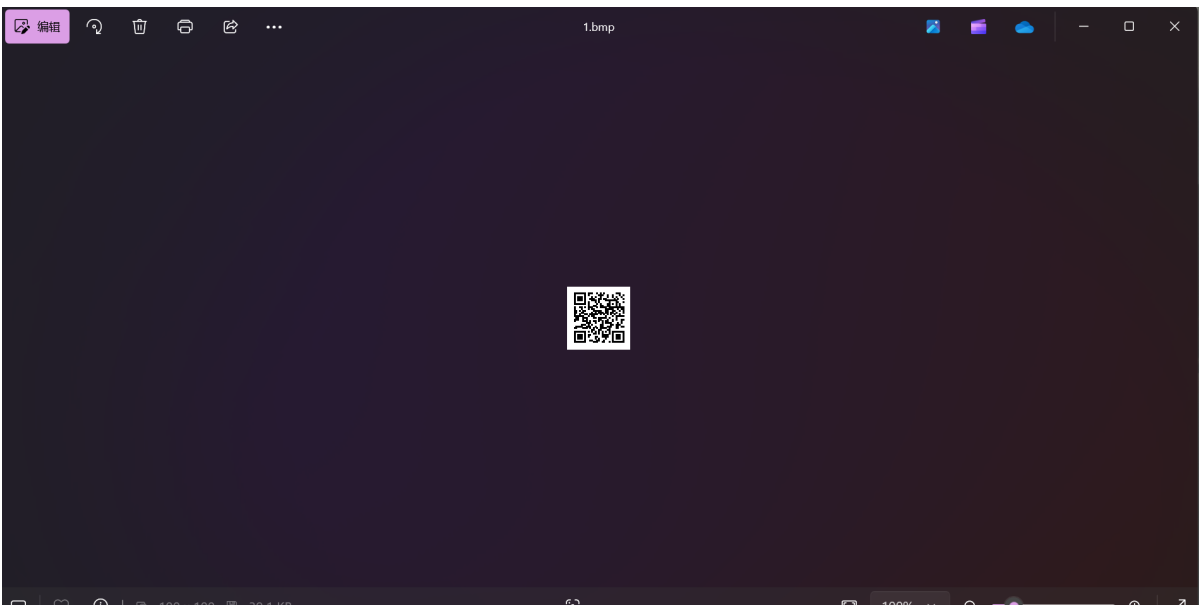
分辨率(width * height) x (颜色深度/8) + bmp文件头(共14Bytes) + 位图信息头(共40Bytes) + 文件尾(共2bytes: 00 00) = 图像文件大小

可以推出分辨率,



分辨率: $(400056-56)/(24/8)=10000$, 组合方式有很多, 但是上面那个是key_part1, 这个肯定是part2

我们注意到上面那个二维码的分辨率就是 $100 \times 100 = 10000$ ，我们把这个bmp也修改成这样试试



成功得到第二张二维码

扫码得到key_part2:089d7147-6786-42b5-b81e-3b7e050304a6

拼一起就是密码key: 0d000721_114514_1919810_089d7147-6786-42b5-b81e-3b7e050304a6

解压得到密文



这时候用第二个提示，网站<https://esolangs.org/wiki/Gopher>

<https://shubshub.github.io/gopher/>

解码

← → ↻ 🔍 shubshub.github.io/gopher/

信息收集 挖洞平台 博客&比赛wp 应急响应 实用的小工具 CTF 流量卡 个人博客 标签

Interpret Gopher! Convert to Gopher! List of Functions

Input:

&+<\$×<×++<@÷+<×++<?×<-. ÷--<'@÷-<+. ^-<-<÷+<÷<^<÷+<@<^<-<-<'<. ×-<@<^-<' ÷--<-<×+<. @++<^×<÷+<÷'+<-<. ÷-<^-<÷<'×<. ×+<÷<÷+<+<+<'<-<-. ×<+<@'×<&÷-<-<. ×-<=

Output:

flag{7a874085-53d5-4baf-96ed-40d73013e67b}|

得到flag:flag{7a874085-53d5-4baf-96ed-40d73013e67b}