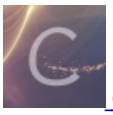
BJDCTF-writeup

原创

夏日 の blog 于 2020-03-23 20:18:43 发布 1562 收藏 1

分类专栏： CTF杂项

本文链接：https://blog.csdn.net/zss192/article/details/105055352



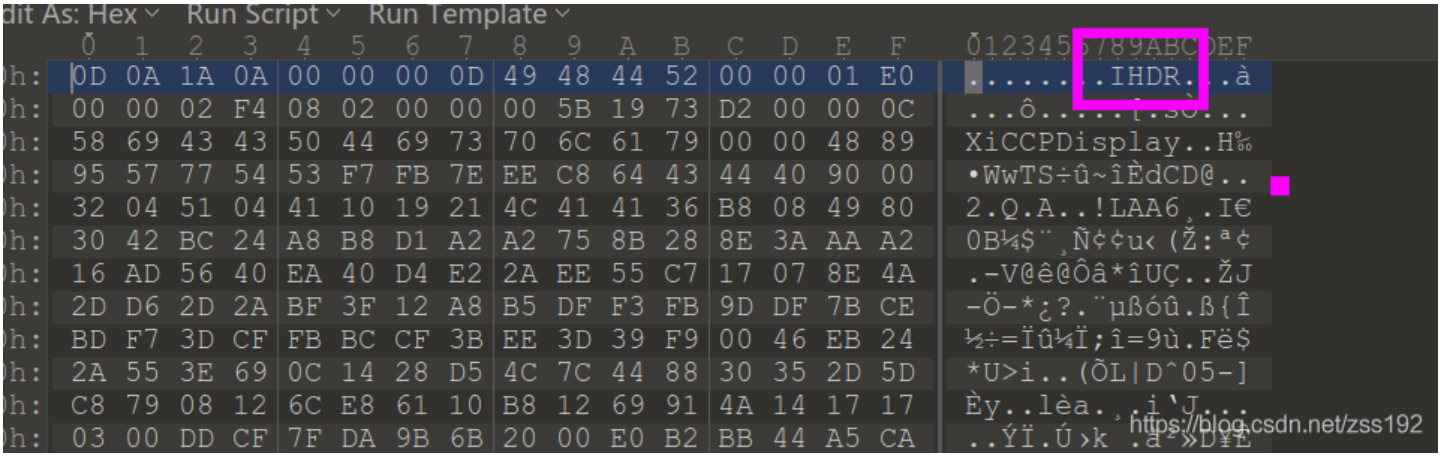
> 本篇文章为个人做题时的部分wp，如有错误，请联系我更正。

目录

杂项

最简单的misc-y1ng

题目是一个zip包，尝试解压出错，用AZR压缩包修复工具修复后，解压得到一个secret文件，由于没有后缀名，放入010editor查看，发现含有IHDR



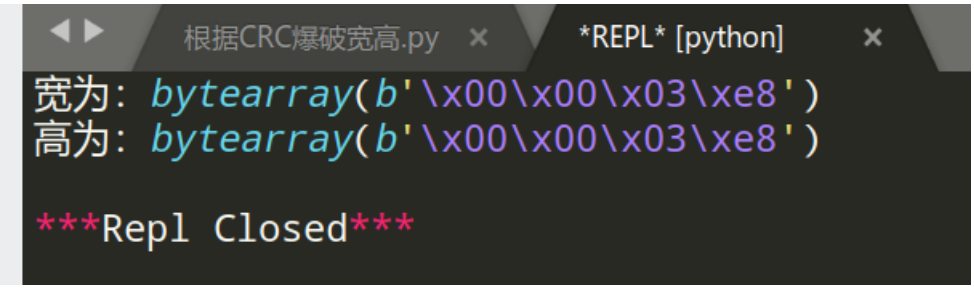
可能是缺少文件头，加上png的文件头89504E47后成功打开图片。发现424A44...等一列数字，猜测是十六进制，在线网站解密得到flag。附上网站：十六进制转文本

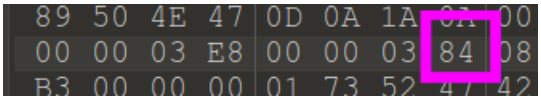
A_Beautiful_Picture

题目是一个图片，直接打开打不开(我用的linux系统)，放到Windows下面可以打开。由此可推断是图片宽高错误，可根据CRC爆破宽高。附上Python脚本

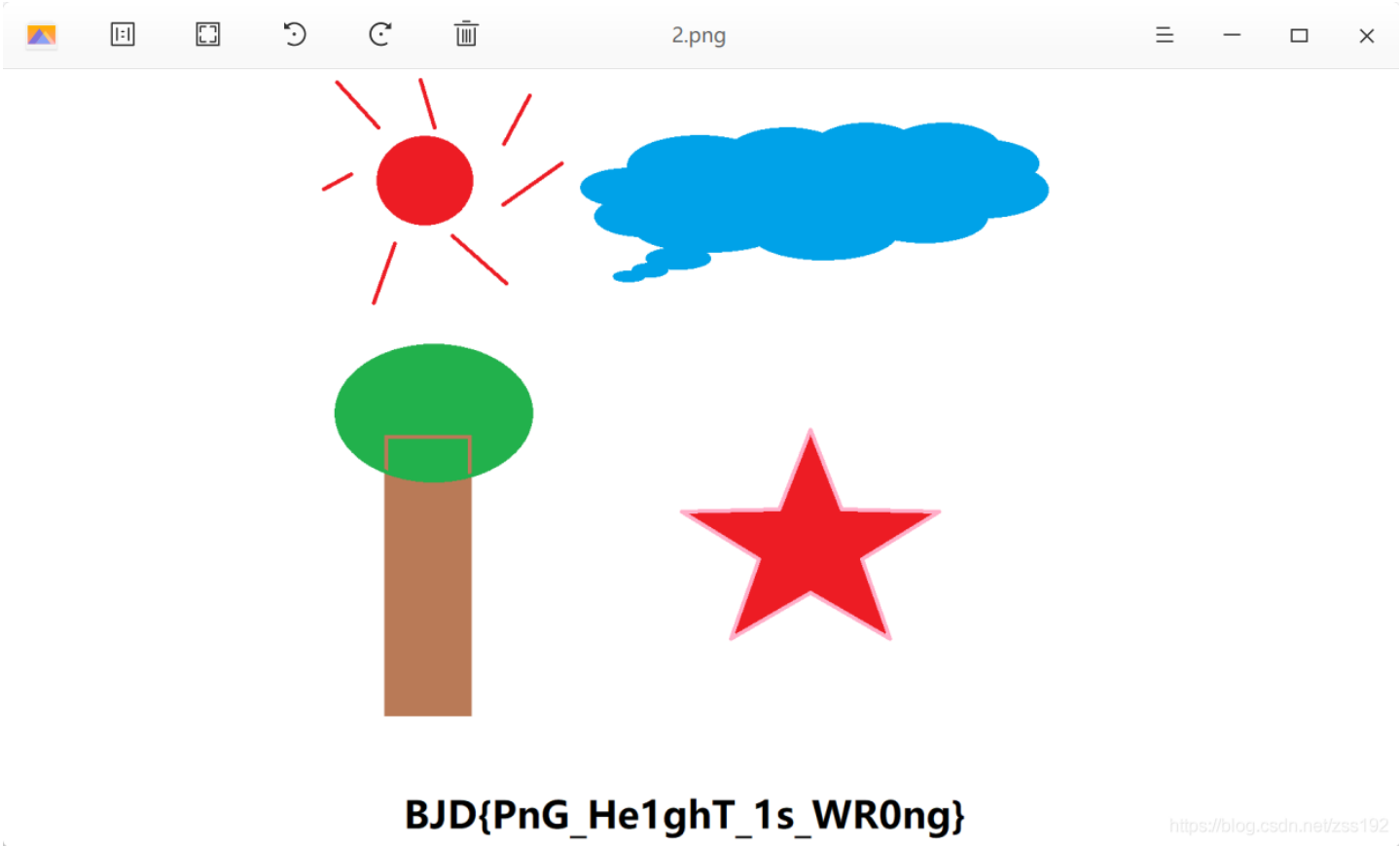
```
import zlib
import struct

filename = 'test.png'
with open(filename, 'rb') as f:
    all_b = f.read()
    crc32key = int(all_b[29:33].hex(),16)
    data = bytearray(all_b[12:29])
    n = 4095            #理论上0xffffffff,但考虑到屏幕实际/cpu，0x0fff就差不多了
    for w in range(n):          #高和宽一起爆破
        width = bytearray(struct.pack('>i', w))     #q为8字节，i为4字节，h为2字节
        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)
            if crc32result == crc32key:
                print("宽为：",end="")
                print(width)
                print("高为：",end="")
                print(height)
                exit(0)
```

根据跑出来的结果更改宽高





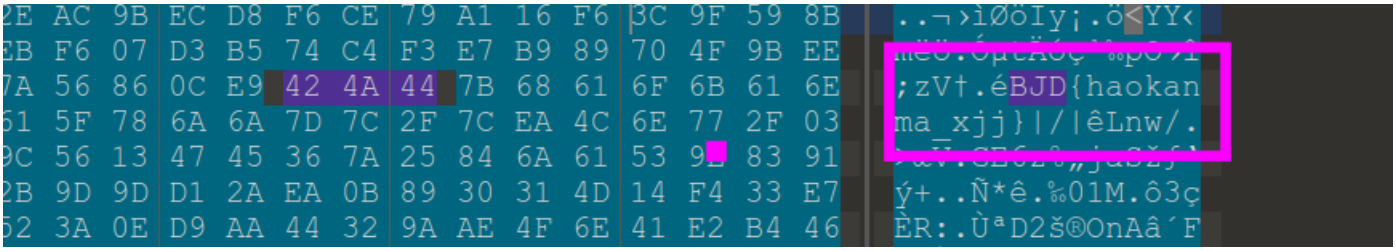
此时就可正常打开图片，发现flag



> 这个考点技巧就是图片在linux下不能打开(打开一片空白)，在Windows上可正常打开。

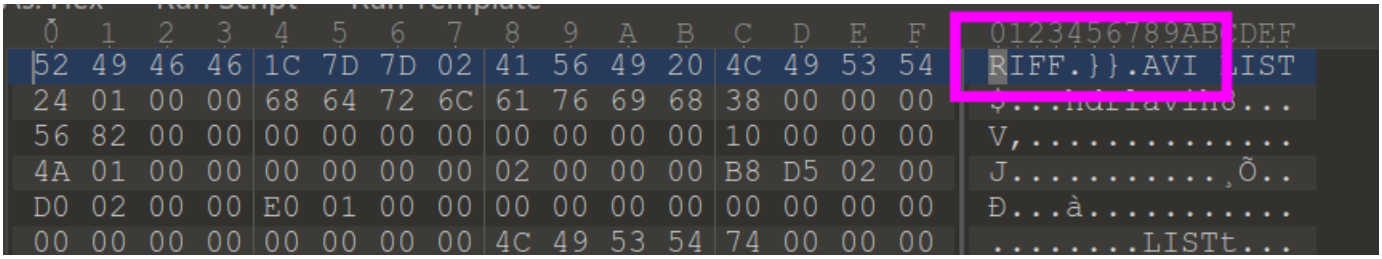
小姐姐-y1ng

这题很奇怪，其实很简单，但好多人没写出来，当时我也刚开始没写出来，就是因为大家都把它想复杂化了。图片放到十六进制编辑器搜索BJD(因为这次比赛名称是BJDCTF)即可找到flag



EasyBaBa

题目是一个图片，但却有19m，那一定是里面藏东西了。使用binwalk查看确实有东西，用foremost分离得到zip文件解压后是名字为里面都是出题人的jpg图片。一看大小39m，信他个鬼，肯定不是图片。放到十六进制编辑器发现文件头为RIFF后面又是AVI，更改后缀为.avi



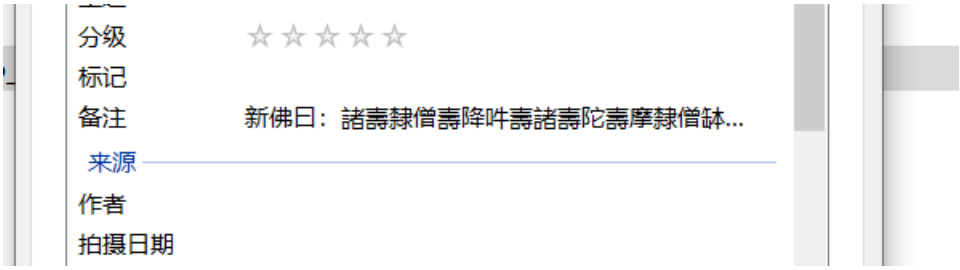
得到一个视频文件，打开发现在第四秒左右有大量画面闪过，很可疑，用ae打开一帧一帧的查看发现多个二维码。扫描的结果是十六进制，转文本拼接后即可得到flag。但是有的二维码死活扫不上，只能截图用ps加深了下颜色才扫描成功。

> 我也是问了一朋友才发现用ae等大型软件看那个二维码会更清楚点，推荐使用ae或pr。

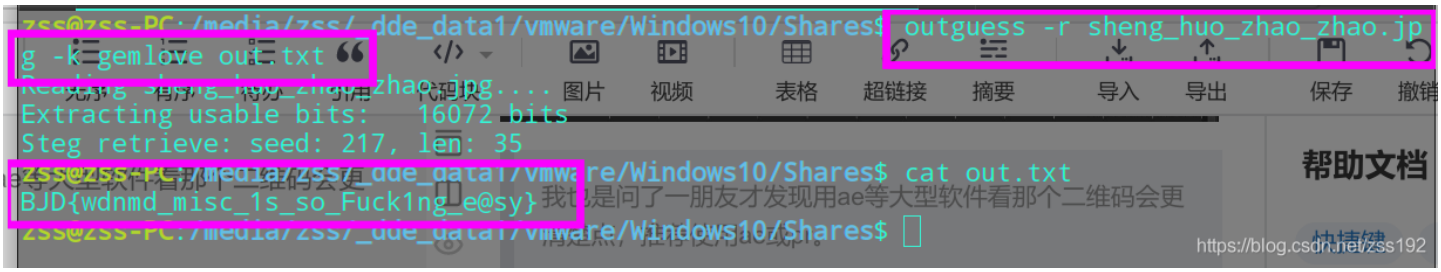
圣火昭昭-y1ng

这题刚开始没做出来，后来官方说出题人给的key出错了，才做出来。

在备注上发现新佛曰，在线解密新约佛论禅得到gemlovecom，官方后来说去掉com,即为gemlove。



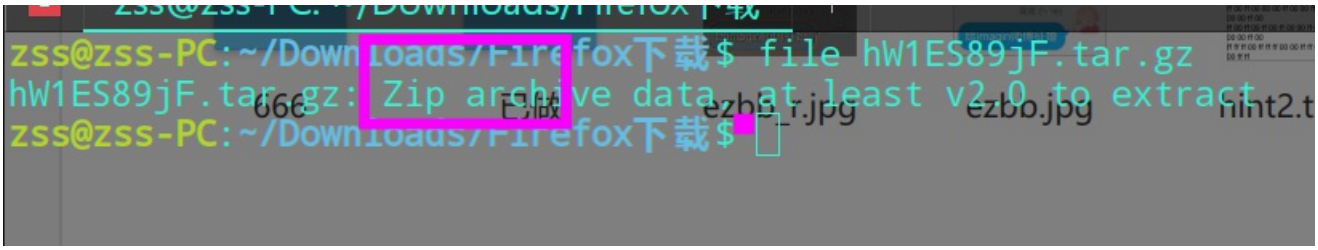
根据gemlove猜测是某种加密，题目描述为：开局一张图，flag全靠猜，猜加重了，猜测是outguess加密。解密得flag。



> 其实这题当时我也没想到是outguess加密，但加密的就是那几个，outguess，f5...这种试一下就出来了。

TARGZ-y1ng

题目是一个tar文件。题目描述为：哎？ 我的tar zxvf怎么不好使了？解压密码不需要爆破。猜测文件并不是tar文件。放到十六进制编辑器发现文件头是504B0304,这是zip文件头，也可用file命令发现这是一个zip文件。



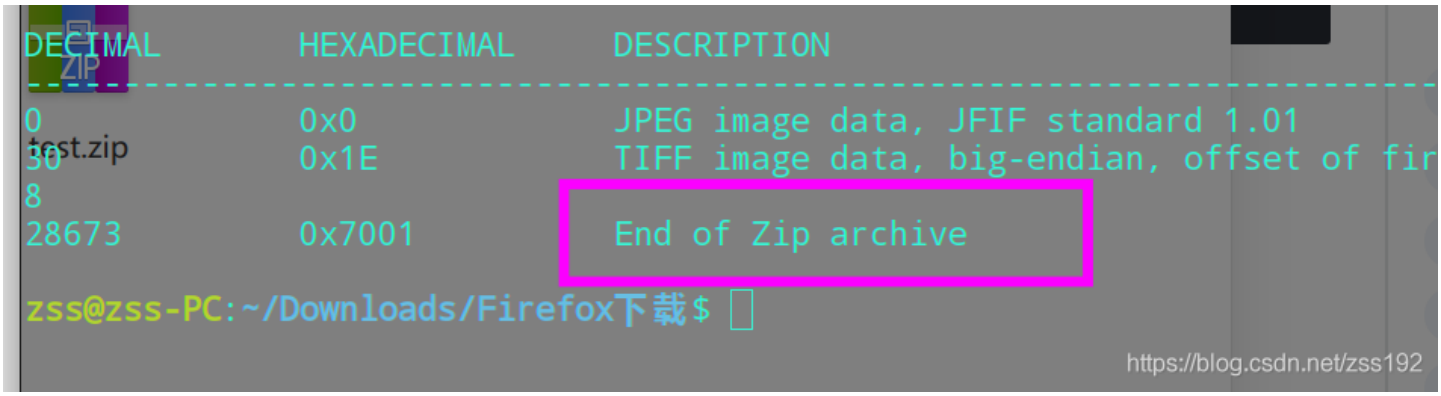
改后缀名为zip，根据题目描述不需要密码，再看文件名很奇怪，尝试用文件名解密成功解密，但这题没有那么简单。后面解密都是用文件名解密，且解密得到的都是tar文件，需要更改后缀再解密。连续解密几次发现规律，应该得用脚本跑。
这里附上自己写的脚本（水平较菜，仅供参考）

```
import zipfile
import os

name = 'hW1ES89jF'
tmp="" #解压过的文件
while True:
    fz = zipfile.ZipFile(name + '.tar.gz', 'r')
    fz.extractall(pwd=bytes(name, 'utf-8'))
    tmp=name+".tar.gz"
    os.remove(tmp)
    name = fz.filelist[0].filename[0:9]
    fz.close(
```

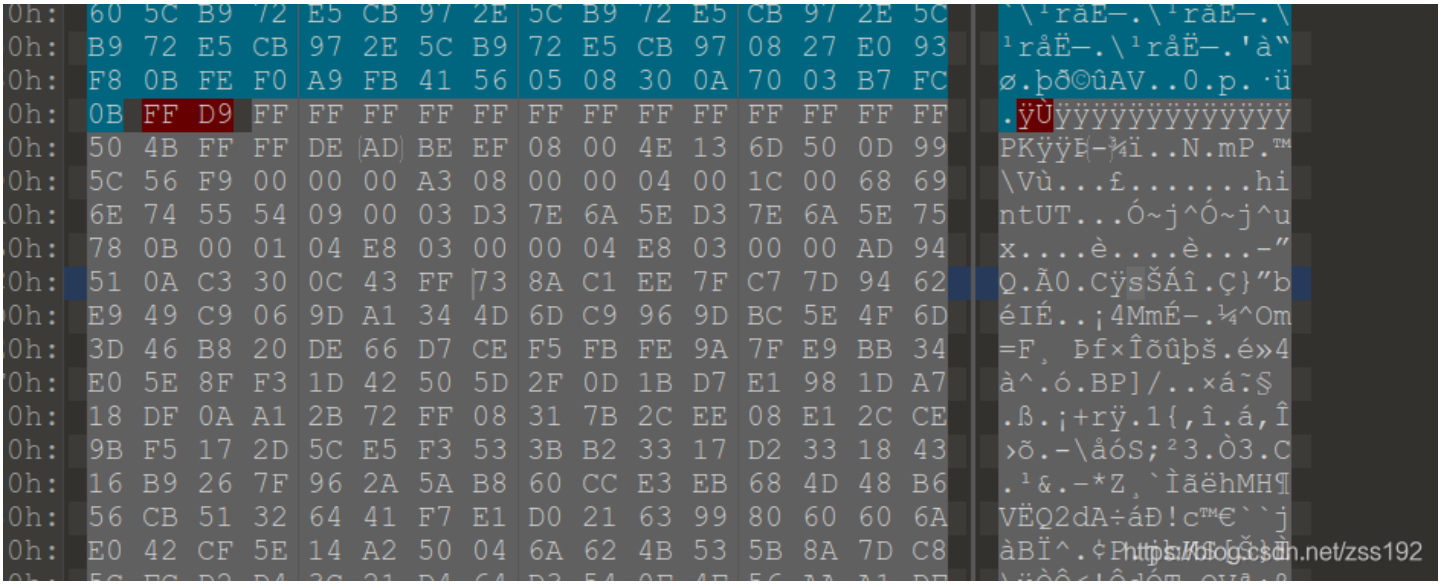
最后会报错(为了适应多种情况不加判断了)，出现flag文件

Real_EasyBaBa

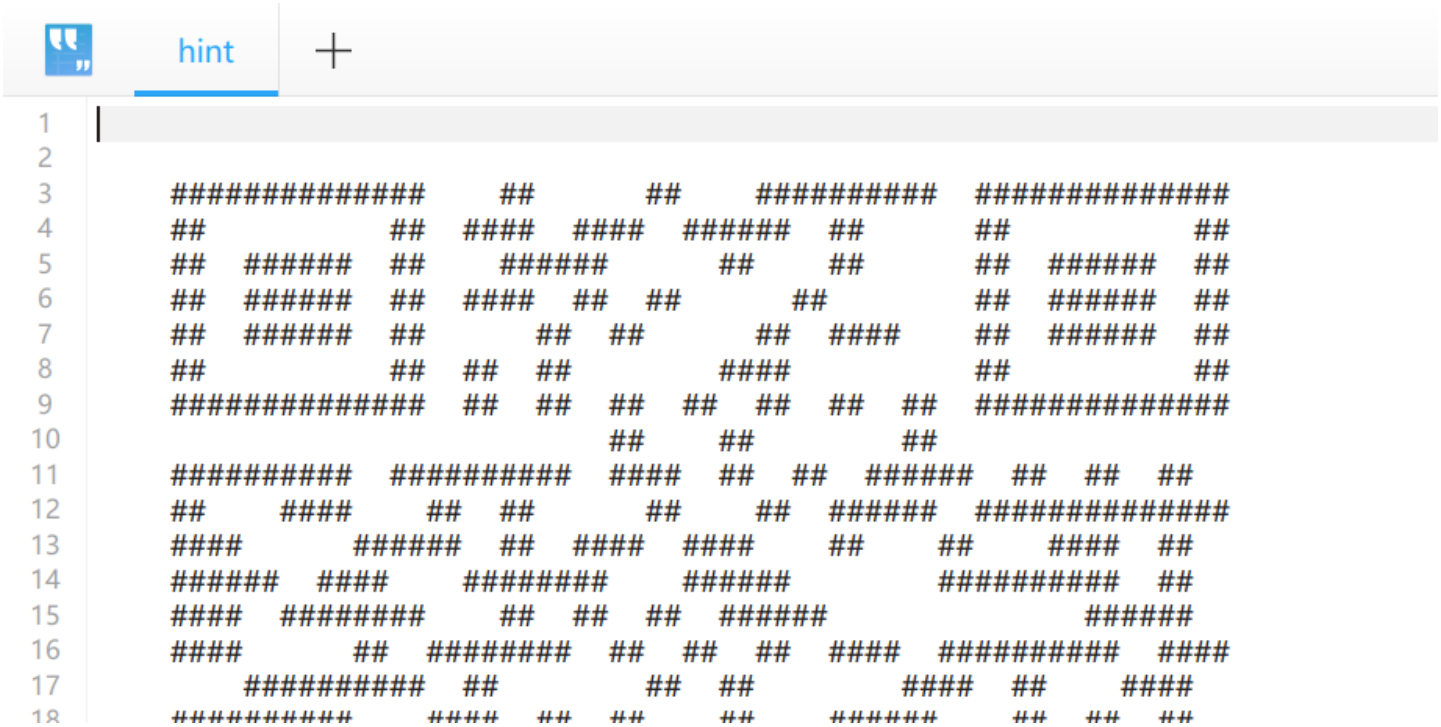
题目是一个图片，binwalk下发现含有zip的结尾，但没有zip头

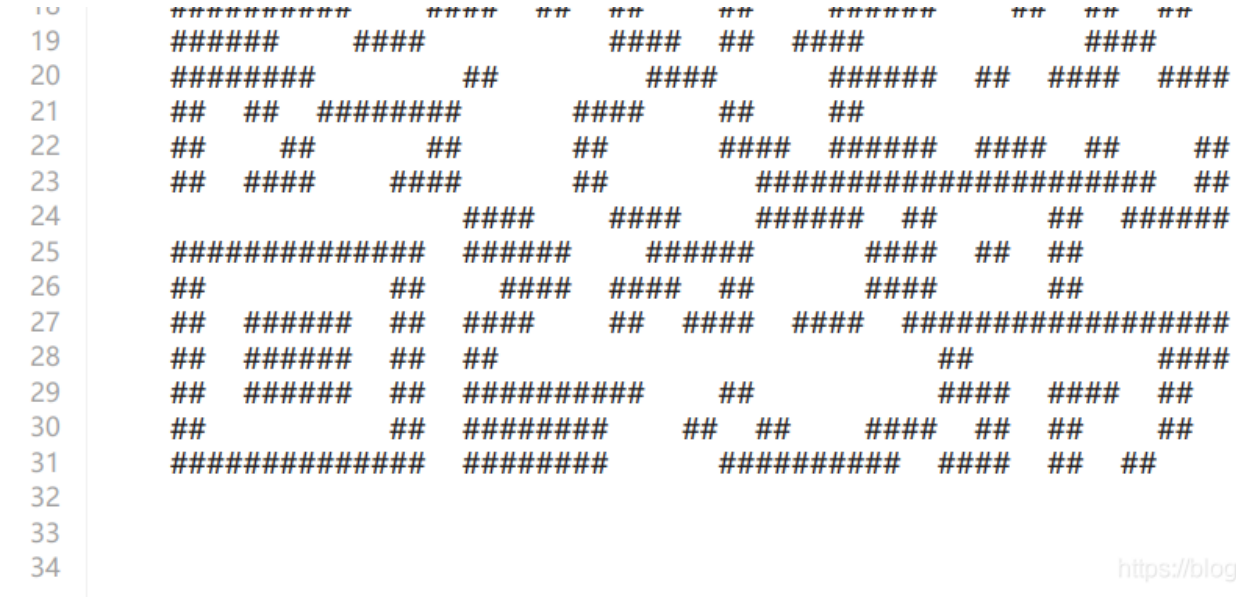


这里就推荐下010editor，里面的模板能直接看到图片结构

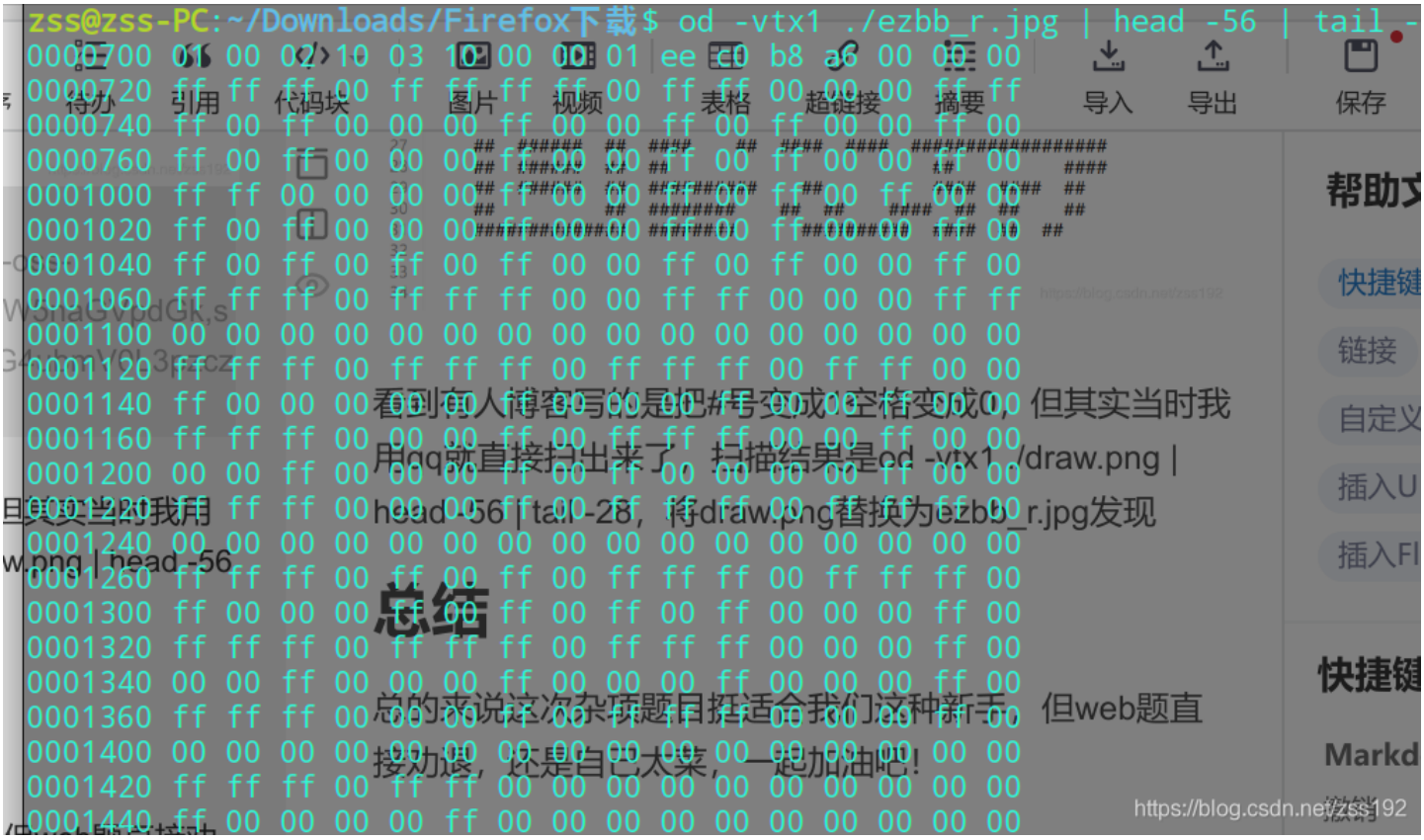


可以看到504BFFFF，将这以及以后的代码另存为一个zip文件并修改头为504B0304，解压后发现一个类似二维码的文档

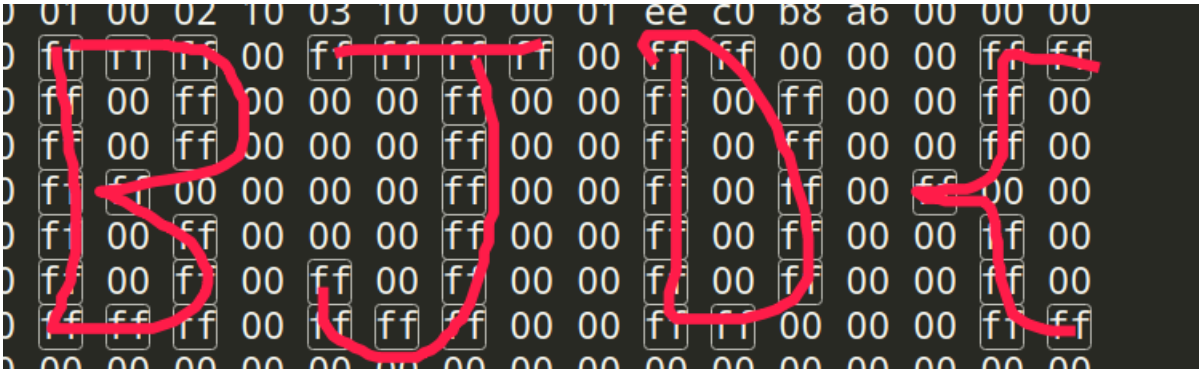


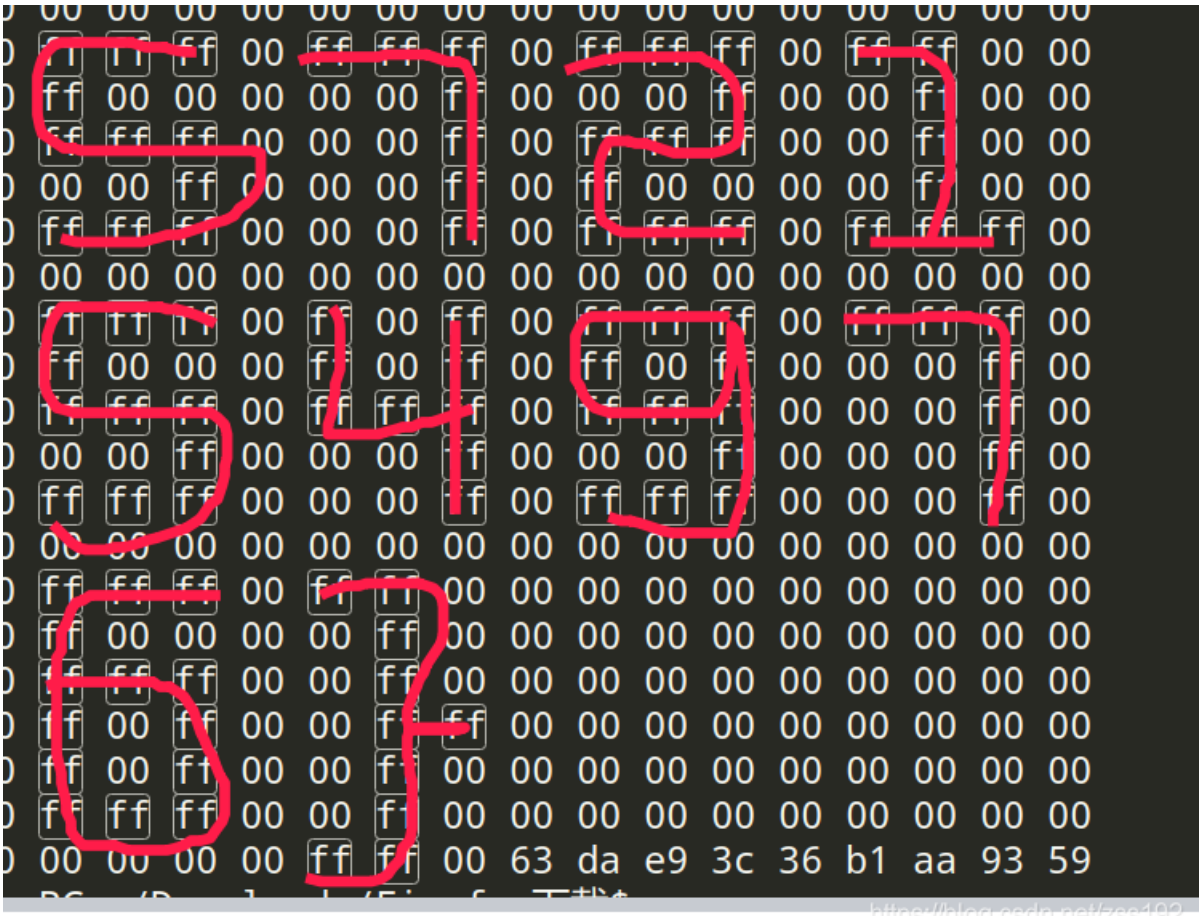


看到有人博客写的是把#号变成1空格变成0，但其实当时我用qq就直接扫出来了，扫描结果是od -vtx1 ./draw.png | head -56 | tail -28，将draw.png替换为ezbb_r.jpg发现这个东西



当时我就是卡到这了，后来看别人的wp发现我们把ff高亮就会发现这就是flag。emm,脑洞很好，可惜我没想出来。





总结

总的来说这次杂项题目挺适合我们这种新手，但web题直接劝退，还是自己太菜，一起加油吧！