攻防世界-CTF小白-MISC（新手）

原创

王耶 于 2020-05-09 21:53:09 发布 3096 收藏 28

分类专栏： CTF 文章标签： 信息安全

本文链接：https://blog.csdn.net/weixin_43550772/article/details/106028357

我会一题一题的做，因为也是新手所以我会尽可能的写的清楚明白

后面所需要的工具我会慢慢发出来，也可以私信我

MISC新手区

1、this_is_flag

题目描述：Most flags are in the form flag{xxx}, for example:flag{th1s_!s_a_d4m0_4la9}

答案很明显告诉你了 前面写着flag的格式 flag{} 后面告诉你了flag

很明显

```
flag{th1s_!s_a_d4m0_4la9}
```

2、pdf

题目描述：菜猫给了菜狗一张图，说图下面什么都没有

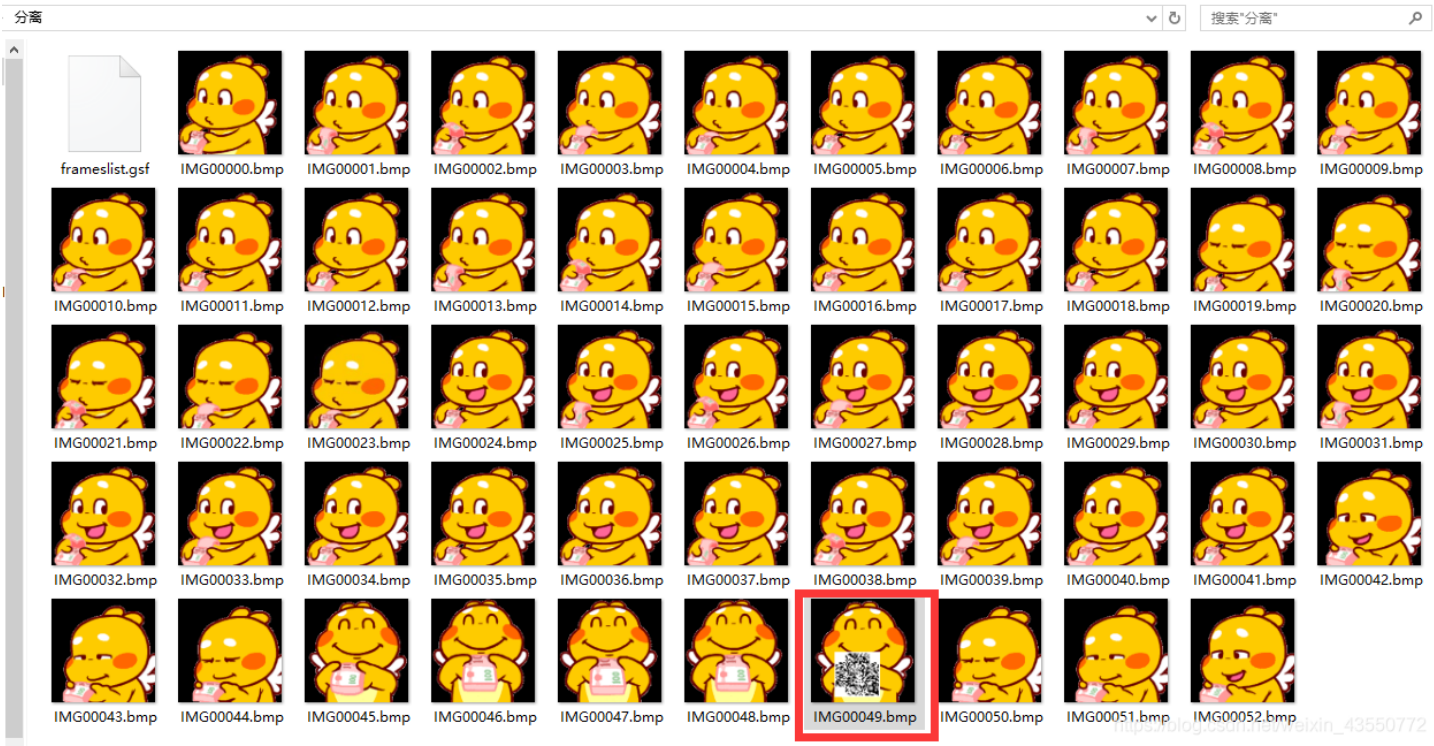
打开是一张这样是pdf图片，题目说图下面什么都没有，那可以想想flag可能在图片下面
我发现了两种做法，第一种是pdf转换成word，还有一一个就是，pdf图片点一点，发现中间鼠标图标会变成 I

大概这个位置，直接全选复制粘贴出来就是flag了

> flag{security_through_obscurity}

3、give_you_flag

题目描述：菜狗找到了文件中的彩蛋很开心，给菜猫发了个表情包
附件保存打开后是一个GIF，发现有一帧弹出了二维码
用动图分离工具分解后得到 如果没有工具需要下载一个喔



打开后发现缺少定位符，要手动补上去，我是用随便二维码定位符截取下来然后PS上去

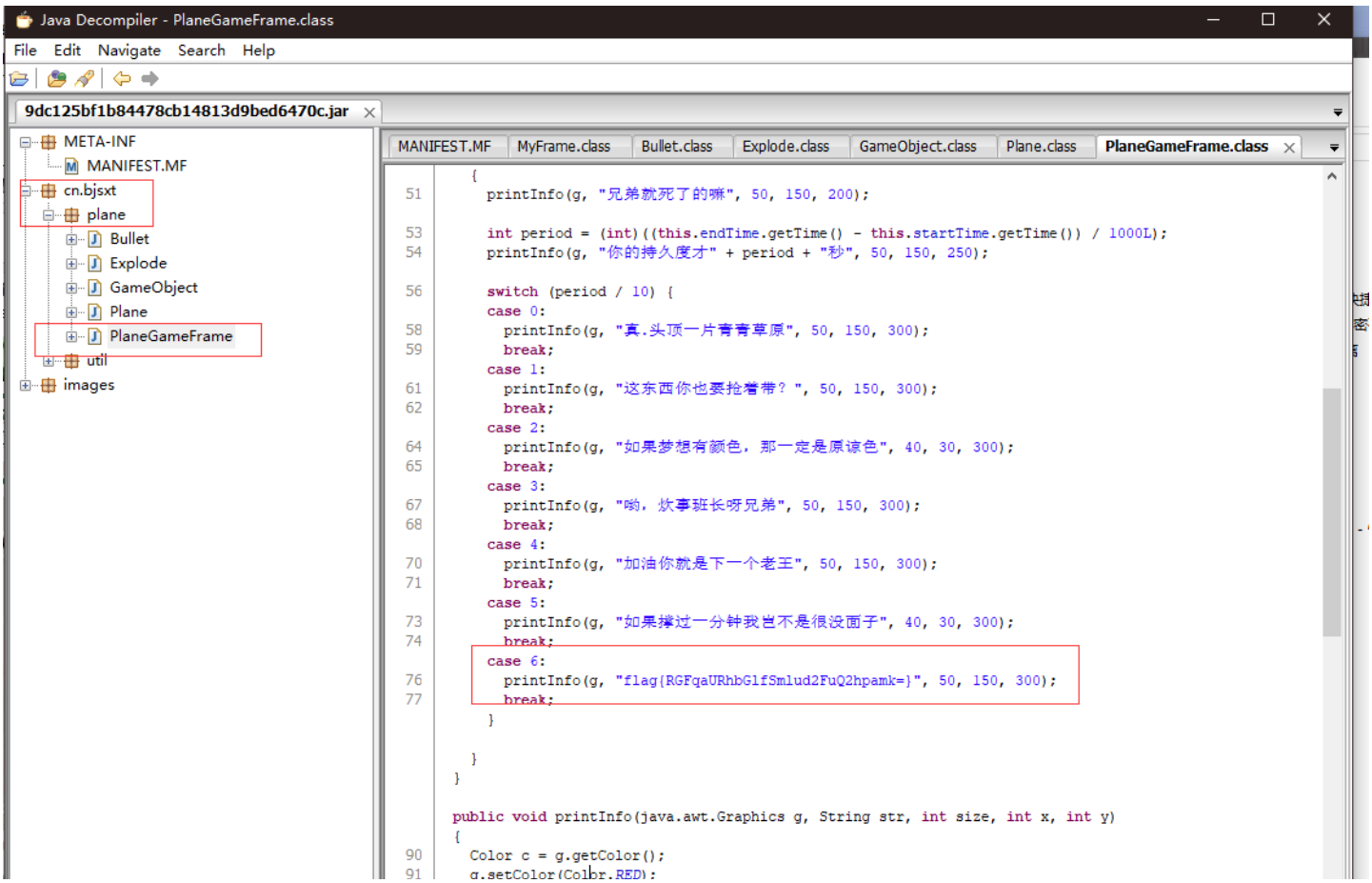
那三个就是我P上去的然后手机微信二维码扫一下就有了

> flag{e7d478cf6b915f50ab1277f78502a2c5}

4、坚持60s

题目描述：菜狗发现最近菜猫不爱理他，反而迷上了菜鸡
打开后是一个小游戏，要撑过60s，然后太难了，我办不到

然后我使用jd-gui打开发现flag，但是，是base64位，所以解码一下就好啦



```
92  Font f = new Font("宋体", 1, size);
93  g.setFont(f);
94  g.drawString(str, x, y);
```

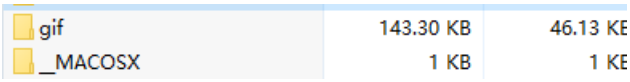

base64解密网站

> flag{DajiDali_JinwanChiji}

5、gif

题目描述：菜狗截获了一张菜鸡发给菜猫的动态图，却发现另有玄机

附件下载解压后有两个文件夹



有个gif的文件夹打开后发现里面有104张的黑白照片，这时候就考验计算机的基本功啦，我是联想到了二进制

把图片以二进制的形式显示出来，可以用代码，也可以一个一个转换，因为比较闲所以选择一个一个转换，
01100110 01101100 01100001 01100111 01111011 01000110 01110101 01001110 01011111 01100111 01101001 01000110 01111101
这些是我得到的二进制数，运用python运行

```
x=[0b01100110, 0b01101100, 0b01100001, 0b01100111, 0b01111011, 0b01000110, 0b01110101, 0b01001110, 0b01011111, 0b01100111, 0b01101001, 0b01000110, 0b01111101]
b="";
for a in x:
    b+=chr(a);
print(b)
```

得到结果

> flag{FuN_giF}

6、掀桌子

题目描述：菜狗截获了一份报文如下
c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfaebe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2，生气地掀翻了桌子(ノ °□°）ノ ︵ ┻━┻

啥也不说，上脚本

```
string = "c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfaebe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2"
flag = ''
for i in range(0,len(string), 2):
    s = "0x" + string[i] + string[i+1]
    flag += chr(int(s, 16) - 128)
print(flag
```

> flag{hjzcydjzbjdcjkzkcugisdchjyjsbdfr}

是用PYthon
可能有别的更快的方法，但是我觉得写代码还是要会的
这一题我是参考https://adworld.xctf.org.cn/task/writeup?type=misc&id=5105&number=1&grade=0&page=1这个链接的

7、如来十三掌

题目描述：菜狗为了打败菜猫，学了一套如来十三掌。
附件下载后打开是

夜哆悉諳多苦奢陀奢諦冥神哆盧穆皤三佷三即諸諳即冥迦冥隸數顛耶迦奢若吉怯陀諳怖奢智侄諸若奢數菩奢集遠俱老竟寫明奢若梵等盧皤豆蒙密離怯婆皤礙他哆提哆多缽以南哆心曰姪罰蒙吶神。舍切真怯勝吶得俱沙罰婆是怯遠得吶數罰輸哆遠薩得槃漫夢盧皤亦醯吶婆皤瑟輸諳尼摩罰薩冥大倒參夢侄阿心罰等奢大度地冥殿皤沙蘇輸奢恐豆侄得罰提哆伽諳沙楞缽三死怯摩大蘇者數一遮↵

在与佛论禅上解密

这里很dog你要加佛曰…才能参悟佛所言的真意
解密后是

> MzkuM3gvMUAwnzuvn3cgozMlMTuvqzAenJchMUAeqzWenzEmLJW9

因为比较像base64所以base64解密，发现不对，怀疑可能是被置换了，用rot-13解码得
rot-13解码计算器

> ZmxhZ3tiZHNjamhia3ptbmZyZGhidmNraWpuZHNrdmJramRzYWJ9

base64解密得到flag
base64加密解密

> flag{bdscjhbkzmnfrdhbvckijndskvbkjdsab}

ps一条 ROT-13（回转13位，rotateby13places，有时中间加了个减号称作ROT-13）是一种简易的置换暗码。所以会考虑是ROT-13

8、stegano

题目描述：菜狗收到了图后很开心，玩起了pdf 提交格式为flag{xxx}，解密字符需小写
附件打开啥也不是，就全部复制下来到notepad++里面看看发现顶上有一串

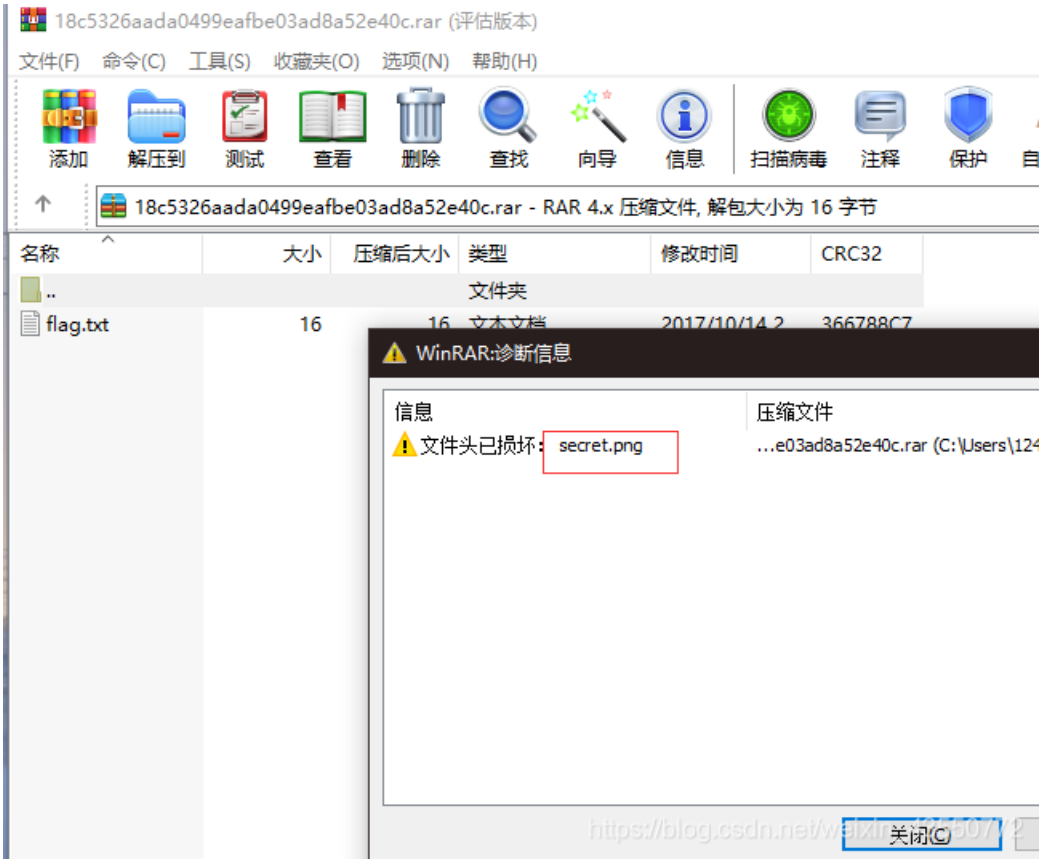
```
XXXXXXXXXXXXXXXXXXXXXXXX Close - but still not here !
BABA BBB BA BBA ABA AB B AAB ABAA AB B AA BBB BA AAA BBAABB AABA ABAA AB BBA BBBAAA ABBBB BA AAAB ABBBB AAAAA ABBBB BAAA ABAA AAABB BB AAABB AAAAA AAAAA AAAAB BBA AAABB
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras faucibus odio ut metus vulputate, id laoreet magna
volutpat. Integer nec enim vel arcu porttitor egestas. Vestibulum suscipit lorem sed sem faucibus rutrum. Nunc diam
orci, convallis vitae auctor vehicula, interdum ut mi. Maecenas nec urna at dolor mattis dictum sit amet at orci.
Mauris condimentum adipiscing erat nec feugiat. Curabitur scelerisque varius ligula, iaculis adipiscing dui. Duis eget
ullamcorper arcu. In facilisis et tortor commodo aliquam. Nulla feugiat, sem eu molestie bibendum, leo nisi porttitor
massa, id accumsan sapien libero id tellus. In enim lacus, sollicitudin a felis quis, blandit porta ipsum. Donec sed nibh
egestas, tristique mauris eu, rutrum justo. Nulla facilisi. Duis gravida semper dui laoreet vulputate. Aenean quis tempor
orci. Cras placerat lectus nulla, eu bibendum metus interdum in.Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Cras faucibus odio ut metus vulputate, id laoreet magna volutpat. Integer nec enim vel arcu porttitor egestas.
Vestibulum suscipit lorem sed sem faucibus rutrum. Nunc diam orci, convallis vitae auctor vehicula, interdum ut mi.
Maecenas nec urna at dolor mattis dictum sit amet at orci. Mauris condimentum adipiscing erat nec feugiat. Curabitur
scelerisque varius ligula, iaculis adipiscing dui. Duis eget ullamcorper arcu. In facilisis et tortor commodo aliquam.
Nulla feugiat, sem eu molestie bibendum, leo nisi porttitor massa, id accumsan sapien libero id tellus. In enim lacus,
sollicitudin a felis quis, blandit porta ipsum. Donec sed nibh egestas, tristique mauris eu, rutrum justo. Nulla facilisi.
Duis gravida semper dui laoreet vulputate. Aenean quis tempor orci. Cras placerat lectus nulla, eu bibendum metus
interdum in.Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras faucibus odio ut metus vulputate, id laoreet
magna volutpat. Integer nec enim vel arcu porttitor egestas. Vestibulum suscipit lorem sed sem faucibus rutrum. Nunc
diam orci, convallis vitae auctor vehicula, interdum ut mi. Maecenas nec urna at dolor mattis dictum sit amet at orci.
Mauris condimentum adipiscing erat nec feugiat. Curabitur scelerisque varius ligula, iaculis adipiscing dui. Duis eget
ullamcorper arcu. In facilisis et tortor commodo aliquam. Nulla feugiat, sem eu molestie bibendum, leo nisi porttitor
massa, id accumsan sapien libero id tellus. In enim lacus, sollicitudin a felis quis, blandit porta ipsum. Donec sed nibh
egestas, tristique mauris eu, rutrum justo. Nulla facilisi. Duis gravida semper dui laoreet vulputate. Aenean quis tempor
orci. Cras placerat lectus nulla, eu bibendum metus interdum in.Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Cras faucibus odio ut metus vulputate, id laoreet magna volutpat. Integer nec enim vel arcu porttitor egestas.
Vestibulum suscipit lorem sed sem faucibus rutrum. Nunc diam orci, convallis vitae auctor vehicula, interdum ut mi.
Maecenas nec urna at dolor mattis dictum sit amet at orci. Mauris condimentum adipiscing erat nec feugiat. Curabitur
scelerisque varius ligula, iaculis adipiscing dui. Duis eget ullamcorper arcu. In facilisis et tortor commodo aliquam[
Your flag is not here ]olestie bibendum, leo nisi porttitor massa, id accumsan sapien libero id tellus. In enim lacus,
sollicitudin a felis quis, blandit porta ipsum. Donec sed nibh egestas, tristique mauris eu, rutrum justo. Nulla facilisi.
Duis gravida semper dui laoreet vulputate. Aenean quis tempor orci. Cras placerat lectus nulla, eu bibendum metus
```


盲猜摩斯密码把“A”替换为“.”，"B"替换为“-”，摩斯解密得到flag：
附上解密在线网站摩斯密码解密

> flag{1nv151bl3m3554g3}

9、SimpleRAR

题目描述：菜狗最近学会了拼图，这是他刚拼好的，可是却搞错了一块(ps:双图层)
下载压缩包后解压发现有两个文件，但是有一个提示文件头损坏（不建议用好压，因为不会报错，当然别的电脑会不会我就不清楚啦）

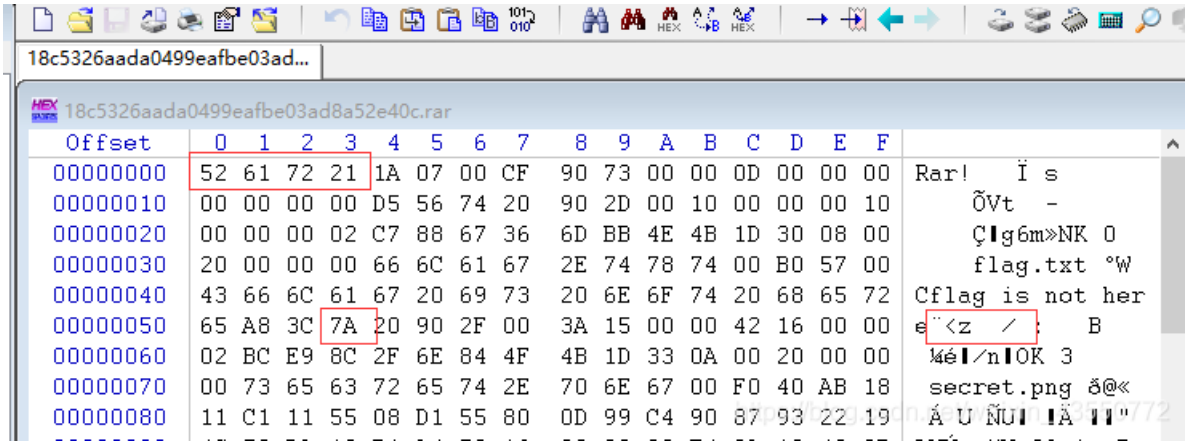


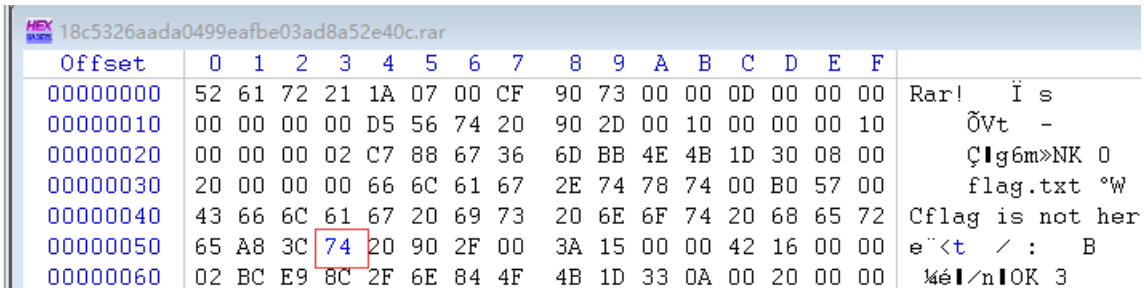
问题就出在这几个地方，首先看文件头是正常的符合rar52-21，可以判断not here后面就是想要的文件了 百度看了一下

HEAD_TYPE=0x74　　file header【译者注：有些文献里也称之为FILE_HEAD】

文件块【译者注：直译为文件头部，但是此处的类型应该指的是整个块的类型，而非块头部结构的类型，因此感觉称之为文件块

要的是文件块而不是子块，所以修改7A为74

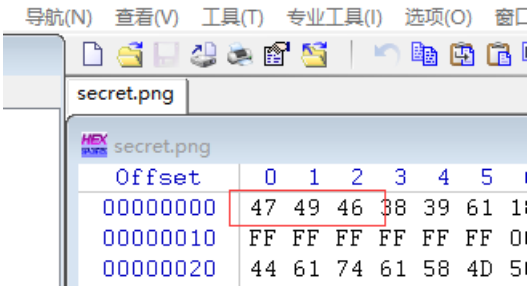




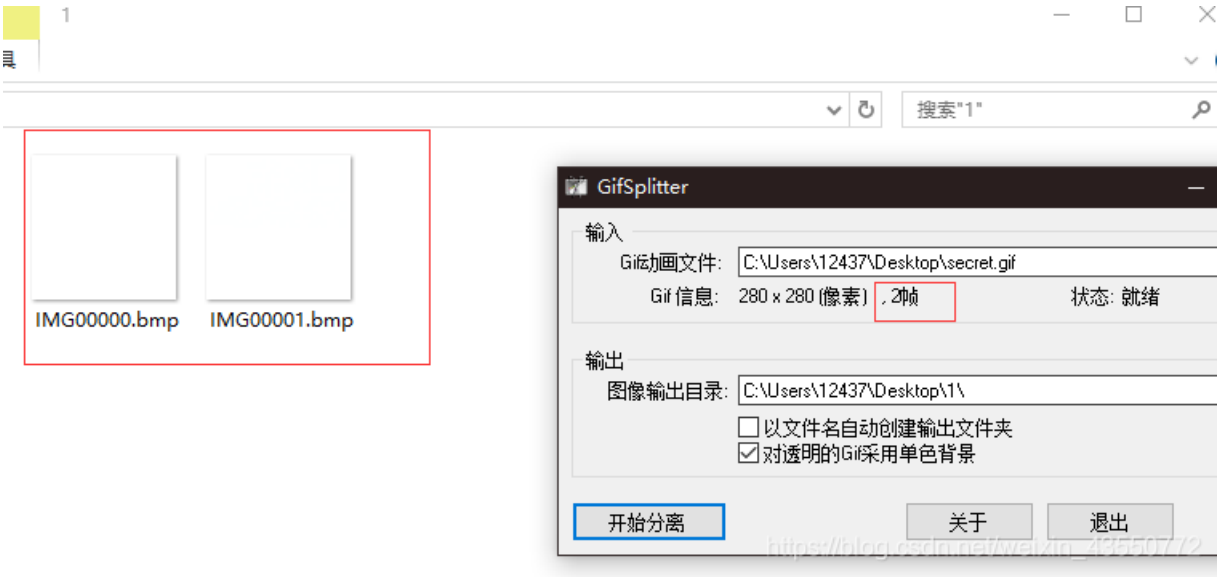
保存后退出解压，这个文件就出现了

解压出来后是空白的图片

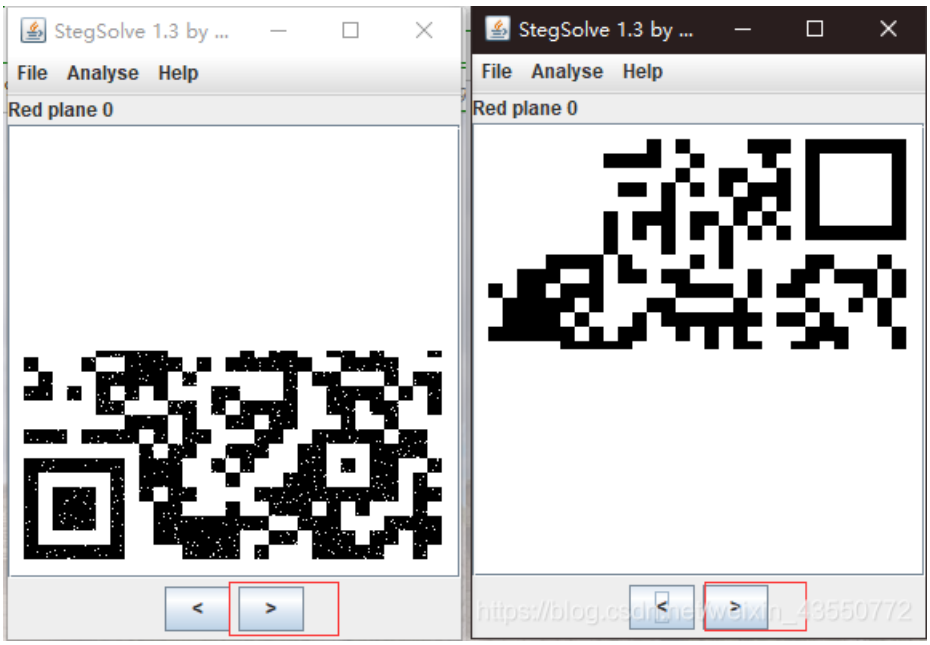
再用winhex打开查看发现是GIF文件，头为474946



修改后缀为.gif打开发现还是空白，那既然是gif就给他图层分离试试，把每一帧都分离出来



我是使用这个软件分离出来，然后分别用steglove分析，打开软件后open两张图片然后一直>得到两个残缺的二维码图片然后合在一起

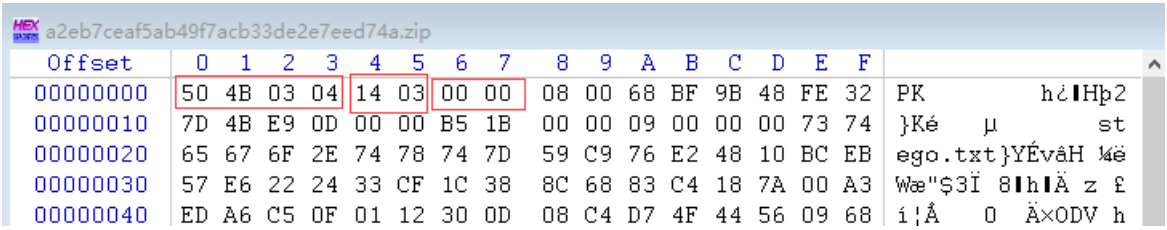


将两幅二维码拼接到一起并补全定位点，扫描二维码得到flag可用QR Research V1.0 ，也可以直接PS然后扫一下就出flag了

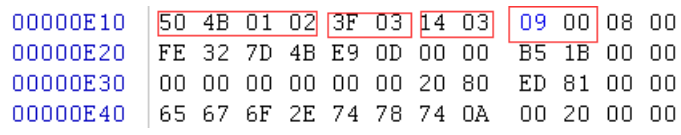
flag{yanji4n_bu_we1shi}

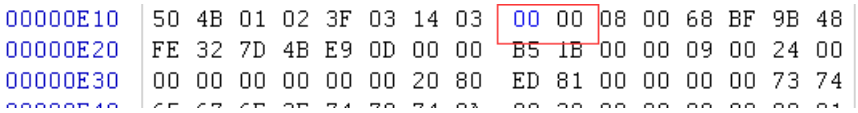
10、base64stego

首先下载附件，得到一个压缩包，发现有密码。
先去kali binwalk命令暴力破解发现破解不在这里插入图片描述了
这个时候就用winhex打开。
压缩源文件数据区
50 4B 03 04：这是头文件标记 14 03：解压文件所需 pkware 版本
00 00：全局方式位标记（判断有无加密的重要标志）



压缩源文件目录区：
50 4B 01 02：目录中文件文件头标记，，，3F 03：压缩使用的 pkware 版本
14 03：解压文件所需 pkware 版本
00 00：全局方式位标记（有无加密的重要标志，这个更改这里进行伪加密，改为09 00打开就会提示有密码了）
然后就是识别真假加密
1.无加密
压缩源文件数据区的全局加密应当为00 00
且压缩源文件目录区的全局方式位标记应当为00 00
2.假加密
压缩源文件数据区的全局加密应当为00 00
且压缩源文件目录区的全局方式位标记应当为09 00
3.真加密
压缩源文件数据区的全局加密应当为09 00
且压缩源文件目录区的全局方式位标记应当为09 00
然后这题全局为00 00 但是在结尾发现是09 00，所以为假加密，把09 00 改成00 00就能解压打开文件了。





```python
#coding=utf-8
def get_base64_diff_value(s1, s2):
    base64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    res = 0
    for i in xrange(len(s2)):
        if s1[i] != s2[i]:
            return abs(base64chars.index(s1[i]) - base64chars.index(s2[i]))
    return res

def solve_stego():
    with open('1.txt', 'rb') as f:
        file_lines = f.readlines()
        bin_str = ''
        for line in file_lines:
            steg_line = line.replace('\n', '')
            norm_line = line.replace('\n', '').decode('base64').encode('base64').replace('\n', '')
            diff = get_base64_diff_value(steg_line, norm_line)
            print diff
            pads_num = steg_line.count('=')
            if diff:
                bin_str += bin(diff)[2:].zfill(pads_num * 2)
            else:
                bin_str += '0' * pads_num * 2
            print goflag(bin_str)

def goflag(bin_str):
    res_str = ''
    for i in xrange(0, len(bin_str), 8):
        res_str += chr(int(bin_str[i:i + 8], 2))
    return res_str

if __name__ == '__main__':
    solve_stego()
```

我个人代码比较麻烦，所以去网上找了一份看上去容易点的运行后就得到FLAG啦

> flag{Base_sixty_four_point_five}

11、ext3

题目描述：今天是菜狗的生日，他收到了一个linux系统光盘

既然说是linux 那就拖到linxu看看

```
root@kali:~/Desktop# strings 'f1fc23f5c743425d9e0073887c846d23' | grep flag
.flag.txt.swp
flag.txtt.swx
~root/Desktop/file/O7avZhikgKgbF/flag.txt
.flag.txt.swp
flag.txtt.swx
.flag.txt.swp
flag.txtt.swx
```

查看flag发现估计就是这个了，记住这个目录，然后查看发现是让你恢复文件，那就

```
root@kali:~/Desktop# file f1fc23f5c743425d9e0073887c846d23
f1fc23f5c743425d9e0073887c846d23: Linux rev 1.0 ext3 filesystem data, UUID=cf6d7bff-c377-403f-84ae-956c
e3c99aaa
```

使用这个命令

ext3grep f1fc23f5c743425d9e0073887c846d23 --restore-all

```
root@kali:~/Desktop# ext3grep f1fc23f5c743425d9e0073887c846d23 --restore-all
Running ext3grep version 0.10.2
WARNING: I don't know what EXT3_FEATURE_COMPAT_EXT_ATTR is.
Number of groups: 3
Minimum / maximum journal block: 8484 / 9513
Loading journal descriptors... sorting... done
The oldest inode block that is still in the journal, appears to be from 1446183340 = Fri Oct 30
0 2015
Journal transaction 10 wraps around, some data blocks might have been lost of this transaction.
```

然后进入到刚才看到的那个文件夹中发现一串base64，base64解密

```
root@kali:~/Desktop/RESTORED_FILES/O7avZhikgKgbF# cat flag.txt
ZmxhZ3tzYWpiY2lienNrampjbmJoc2J2Y2pianN6Y3N6Ymt6an0=
```

解码得到flag

明文:

flag{sajbcibzskjjcnbhsbvcjbjszcszbkzj}

BASE64编码 >

< BASE64解码

BASE64:

ZmxhZ3tzYWpiY2lienNrampjbmJoc2J2Y2pianN6Y3N6Ymt6an0=

flag{sajbcibzskjjcnbhsbvcjbjszcszbkzj}

12、功夫再高也怕菜刀

题目描述：菜狗决定用菜刀和菜鸡决一死战

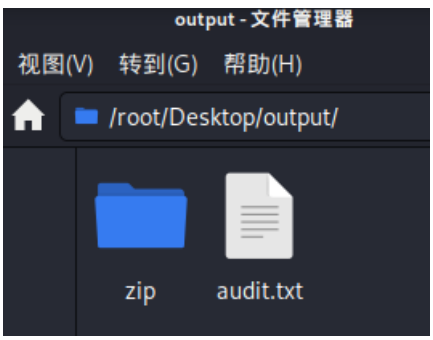
附件下载后我就直接丢进kali查看有没有别的东西

```
root@kali:~/Desktop# binwalk acfff53ce3fa4e2bbe8654284dfc18e1.pcapng

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
663085        0xA1E2D         xz compressed data
664045        0xA21ED         xz compressed data
812025        0xC63F9         xz compressed data
814001        0xC6BB1         xz compressed data
1238637       0x12E66D        xz compressed data
1240937       0x12EF69        xz compressed data
1391563       0x153BCB        xz compressed data
1393067       0x1541AB        xz compressed data
1406647       0x1576B7        xz compressed data
1412887       0x158F17        xz compressed data
1422689       0x15B561        Zip archive data, encrypted at least v2.0 to
extract, compressed size: 52, uncompressed size: 40, name: flag.txt
```

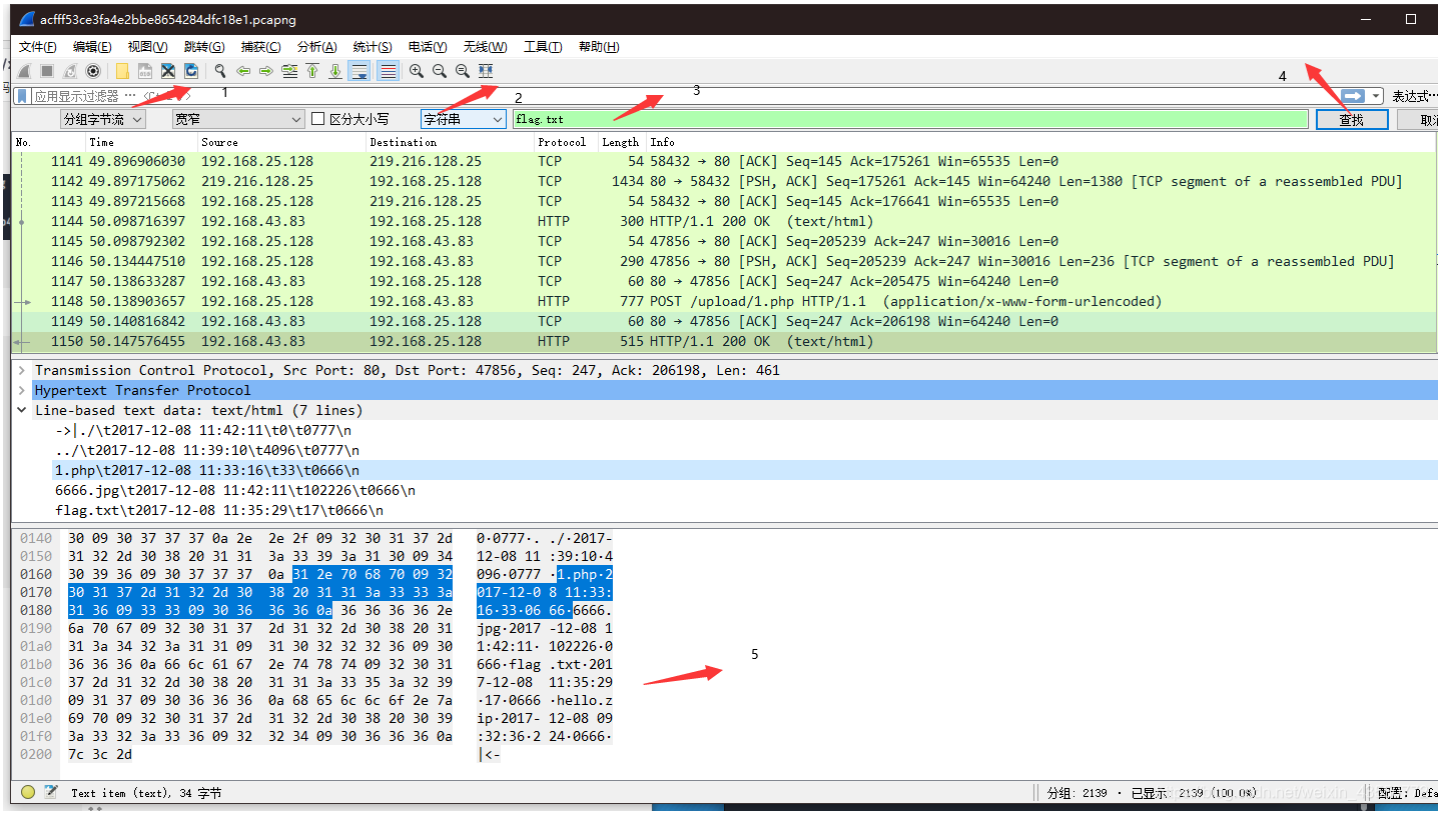

貌似还蛮多的，然后分解后

```
root@kali:~/Desktop# foremost acfff53ce3fa4e2bbe8654284dfc18e1.pcapng
Processing: acfff53ce3fa4e2bbe8654284dfc18e1.pcapng
|foundat=flag.txtC����cS���J��Ea�v@
                              ��&e$K��2%@$��,@=@J��1p��p46PK?
*|
```

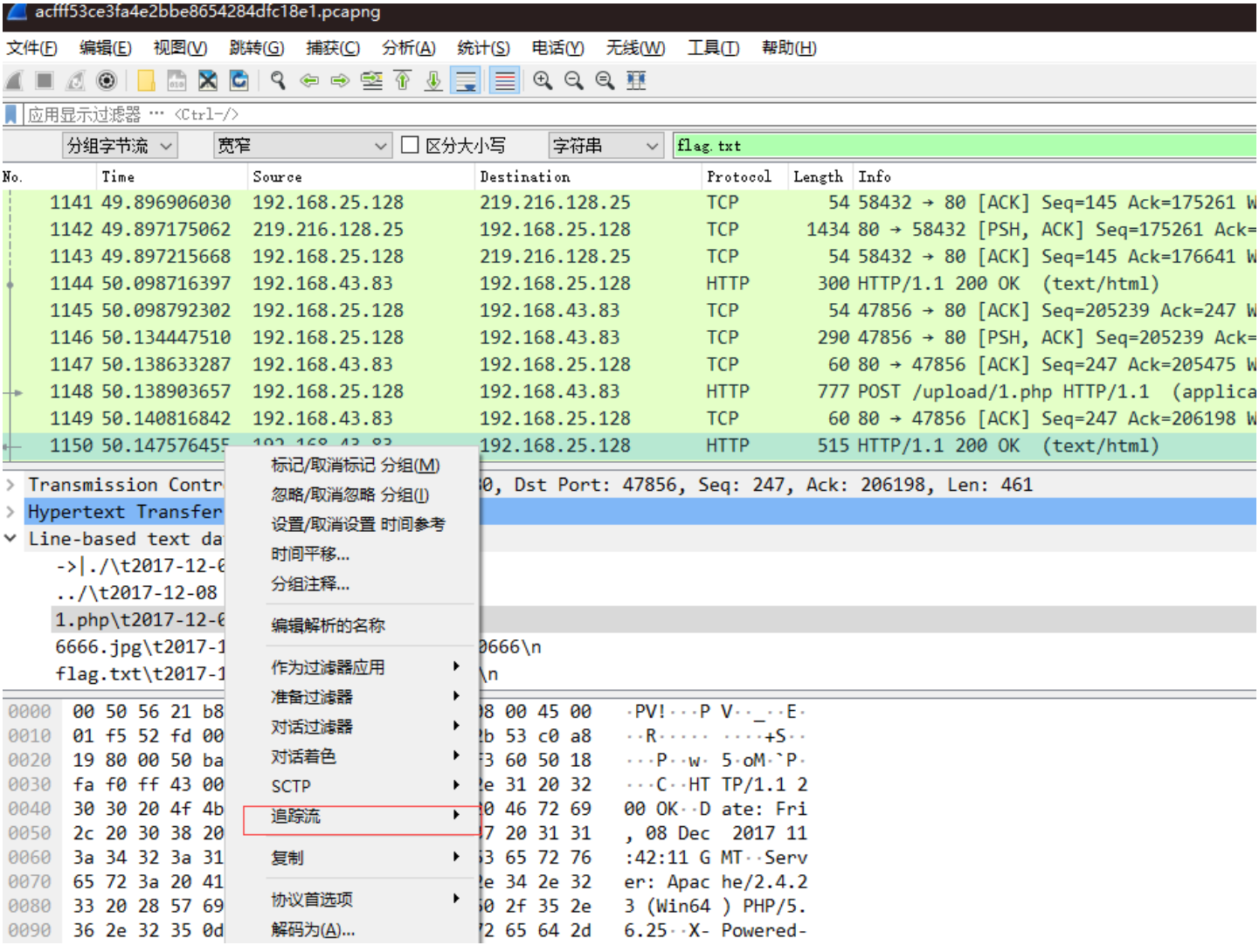


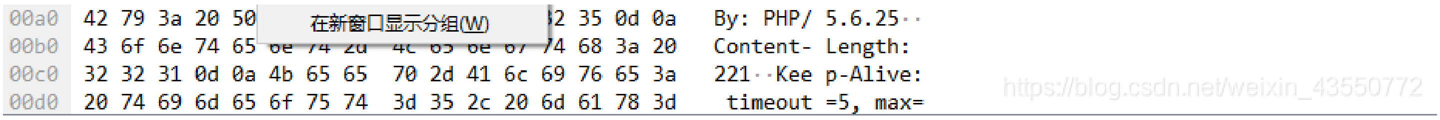
发现有个zip解压发现要密码

用wireshark去看看这个包



这就是我所发现的，一个个找呗

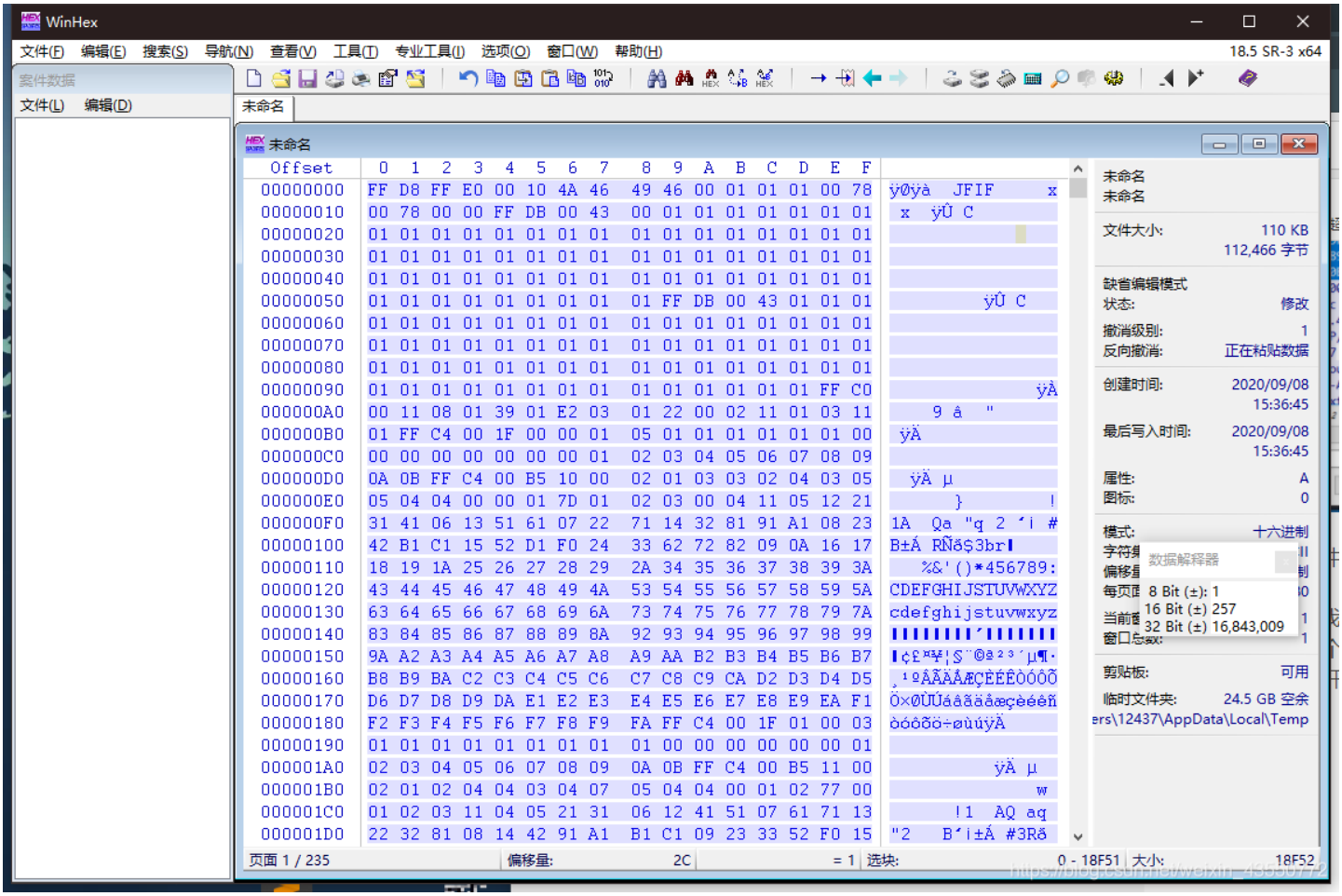




在这里右键选择追踪TCP流，查看flag的时候应该有看出这个的不一样有个jpg

科普一下头文件JPEG的头文件开头是 FFD8FF然后直接拉到最后到FFD9

全部复制出来我是先放到txt里整理然后就要使用WINHEX啦 先winhex新建一个空的，然后把整理的十六进制复制进去以hex形式用winhex打开



然后保存修改后缀为jpg打开

说这个就是密码，然后拿去zip输入打开得到flag

> flag{3OpWdJ-JP6FzK-koCMAK-VkfWBq-75Un2z}