




# CTF杂项-BUUCTF竞赛真题WriteUp(2)

原创

Tr0e  于 2021-05-05 23:44:11 发布  1448  收藏 24

分类专栏: [CTF之路](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/weixin\\_39190897/article/details/116430110](https://blog.csdn.net/weixin_39190897/article/details/116430110)

版权



[CTF之路](#) 专栏收录该内容

17 篇文章 27 订阅

订阅专栏

文章目录

## 前言

No.1 Git动图分解提取信息

No.2 隐藏文件提取与爆破

No.3 Base64编码还原图片

No.4 winhex修改图片大小

No.5 编辑器查看图片隐写

No.6 LSB隐写的信息导出

No.7 图片属性中隐藏信息

No.8 LSB隐写的数据抽取

No.9 RAR文件加密的爆破

No.10 二进制编码转Ascll

No.11 Winhex搜索获得flag

No.12 ZIP的16进制文件头

No.13 盲文与摩斯密码读取

No.14 Brainfuck编码与解码

No.15 后门查杀之后门识别

No.16 路由器信息数据查看

No.17 base64流量导出图片

No.18 脑洞大开歌名猜密钥

No.19 Steghide隐写工具的使用

No.20 ZIP伪加密与多重编码转换

No.21 F5隐写与ZIP文件头的识别

No.22 Py脚本16进制坐标绘二维码

No.22 HTTP流量分析与文件转换

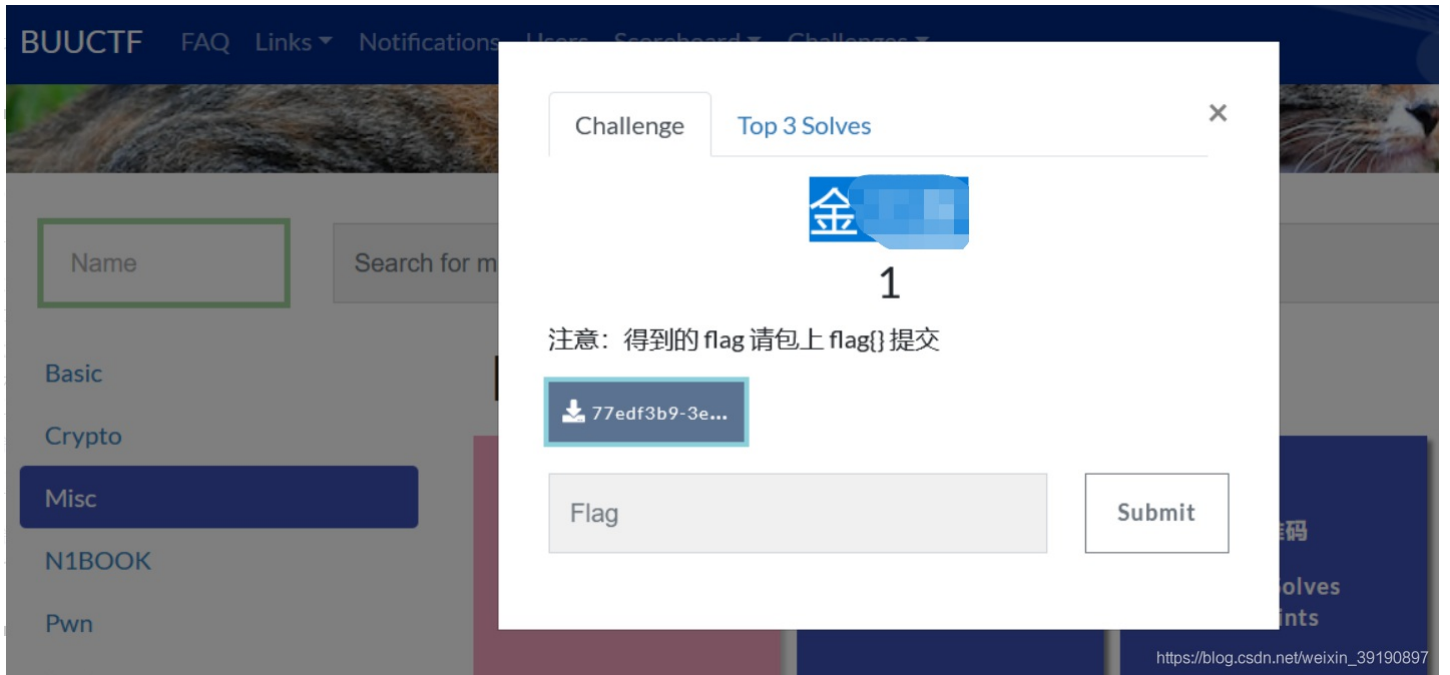
No.23 HTTP流量分析与文件提取

总结

## 前言

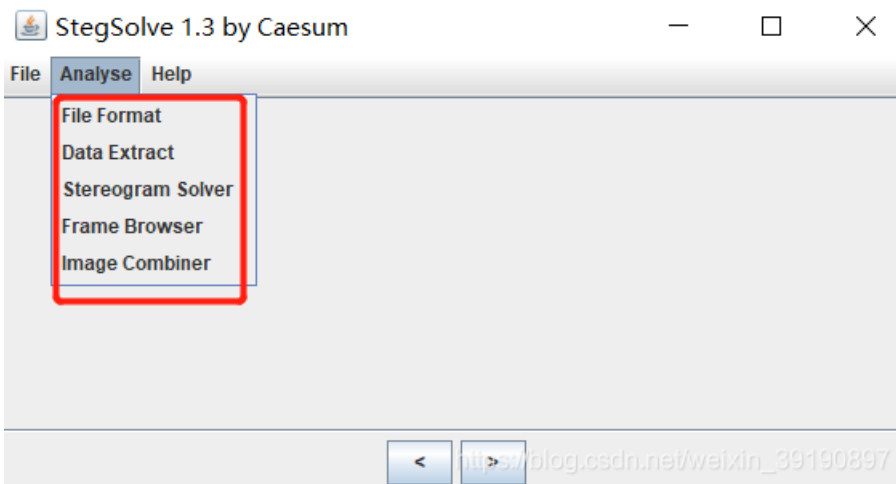
为了划水过几天的“红帽杯”网络安全大赛，学习并记录下 BUUCTF 平台的杂项部分题目，因为去年参加“强网杯”网络安全大赛发现杂项类型的题目还是可以争取得分的，也希望过几天运气好吧...

## No.1 Git动图分解提取信息



下载附件发现是一个动图，闪烁着红色 flag 字体。该题需要用到隐写图片查看的神器——[stegsolve](#)，下载地址。

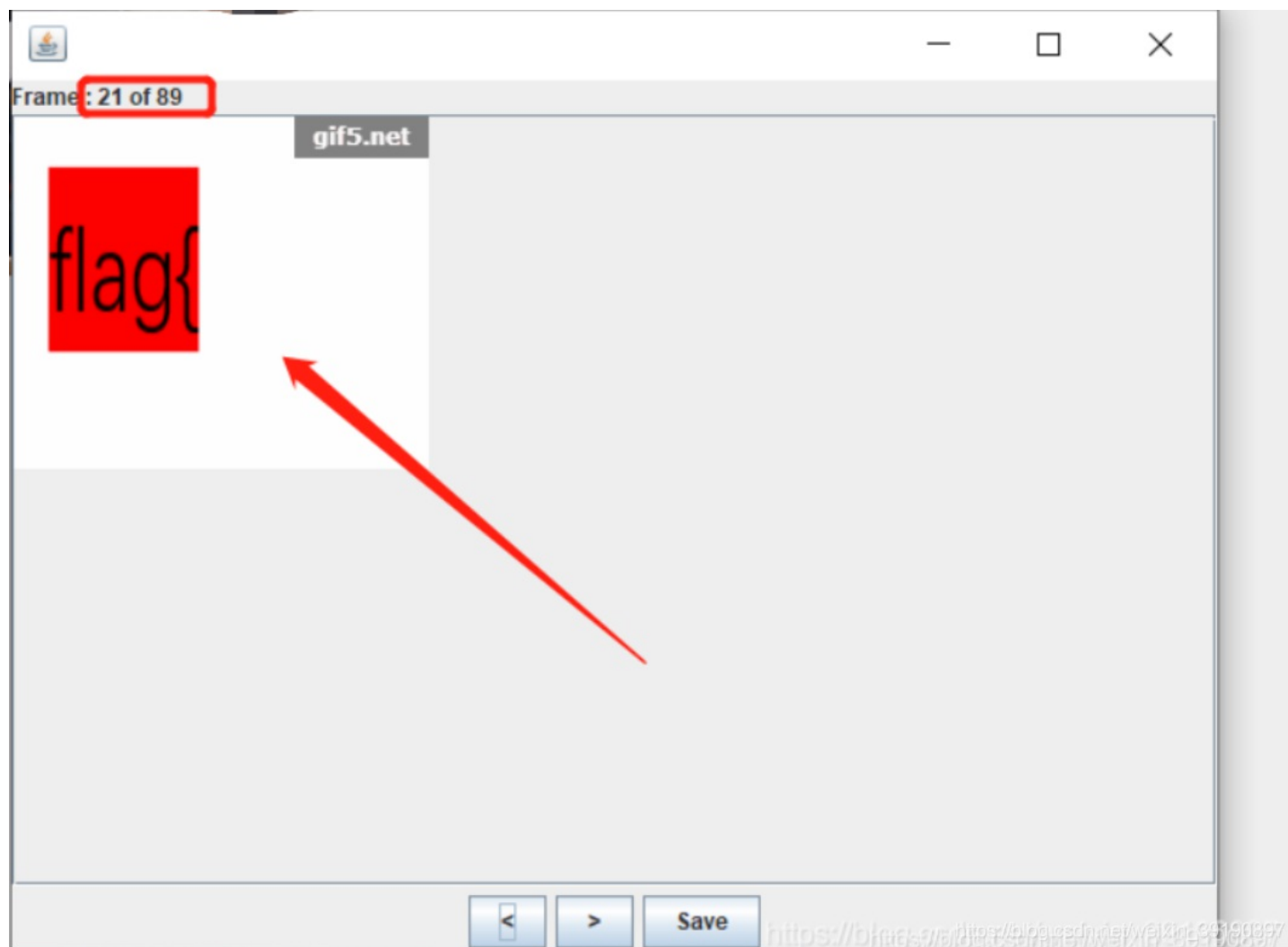
Stegsolve 功能简介：



上面是软件打开的界面，界面简单。主要供能为 analyse，下面对 Analyse 下面几个功能键作简单介绍：

功能键	作用
File Format	文件格式，这个主要是查看图片的具体信息
Data Extract	数据抽取，图片中隐藏数据的抽取
Frame Browser	帧浏览器，主要是对GIF之类的动图进行分解，动图变成一张张图片，便于查看
Image Combiner	拼图，图片拼接

此题是动图中闪烁着 flag，故使用 Frame Browser 功能对 gif 动图进行分解，获得 89 张分解后的静图，查阅获得 flag:

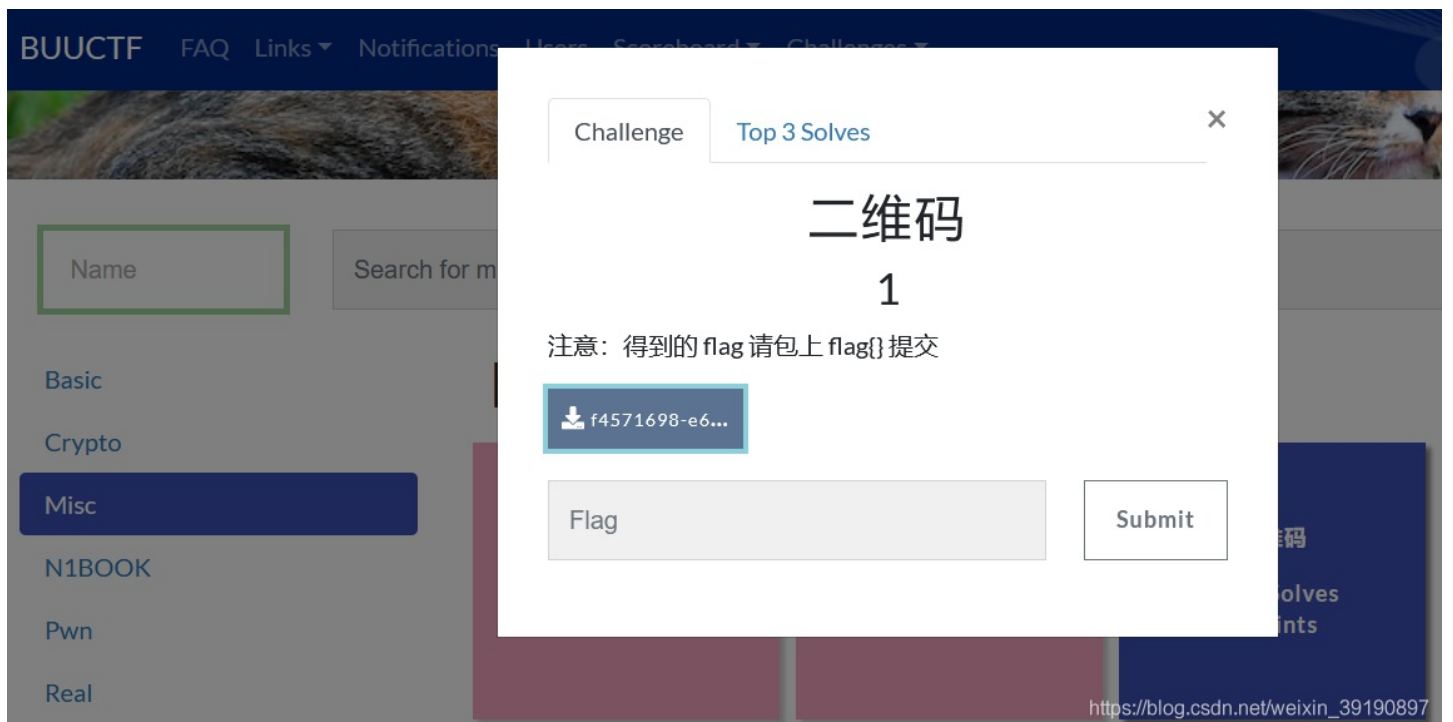






获得 flag: `flag{he11ohongke}`，本入门题完。

## No.2 隐藏文件提取与爆破



1、下载完打开是个二维码，用微信扫描只能看见一句话——“secret is here”。



2、猜测二维码图片中隐藏了别的文件，可以使用 Binwalk 工具进行文件分析和分离。

## 1. Binwalk 工具

Binwalk 是 Linux 下用来分析和分离文件的工具，可以快速分辨文件是否由多个文件合并而成，并将文件进行分离。会在目标文件的目录。

同目录下生成一个形如\_文件名\_extracted的文件目录，目录中有分离后的文件。

用法：

分析文件：`binwalk filename`

分离文件：`binwalk -e filename`

```
root@kali2: ~/ctf# binwalk ans.jpg
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, JFIF standard 1.01
8232	0x2028	TIFF image data, big-endian
19610	0x4C9A	Copyright string: " (c) 1998 Hewlett-Packard Company"

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

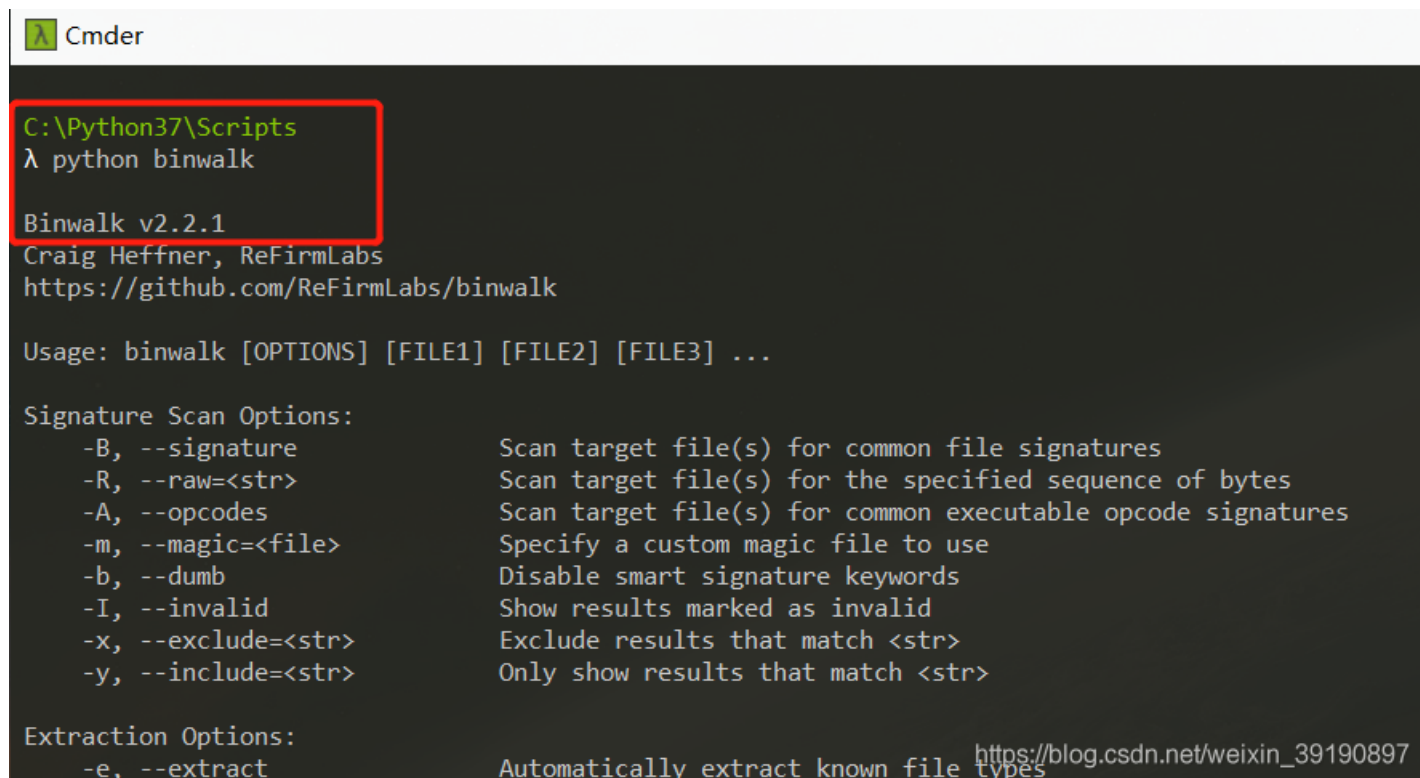
Kali Linux 自带了 Binwalk 工具，但我不想开虚拟机.....所以在 Windows 下安装，Github 下载源码，本地安装好 python 并进入 Binwalk 所在文件夹，执行命令如下，程序将会自动安装：

```
Cmdr
D:\CTF\杂项Misc\binwalk-master\binwalk-master
λ ls
API.md  deps.sh*  Dockerfile  images/  INSTALL.md  LICENSE  NOTICE.md  README.md  setup.py*  src/  testing/
D:\CTF\杂项Misc\binwalk-master\binwalk-master
λ python setup.py install
running install
running bdist_egg
```

```
Running build_egg
running egg_info
creating src\binwalk.egg-info
writing src\binwalk.egg-info\PKG-INFO
writing dependency_links to src\binwalk.egg-info\dependency_links.txt
writing top-level names to src\binwalk.egg-info\top_level.txt
writing manifest file 'src\binwalk.egg-info\SOURCES.txt'
reading manifest file 'src\binwalk.egg-info\SOURCES.txt'
```

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

安装完成后在 python 安装目录下的 Script 目录里有个名字叫做 binwalk 的文件，然后在该文件夹启动命令行就可以使用 Binwalk 了：



```
Cmder
C:\Python37\Scripts
λ python binwalk

Binwalk v2.2.1
Craig Heffner, ReFirmLabs
https://github.com/ReFirmLabs/binwalk

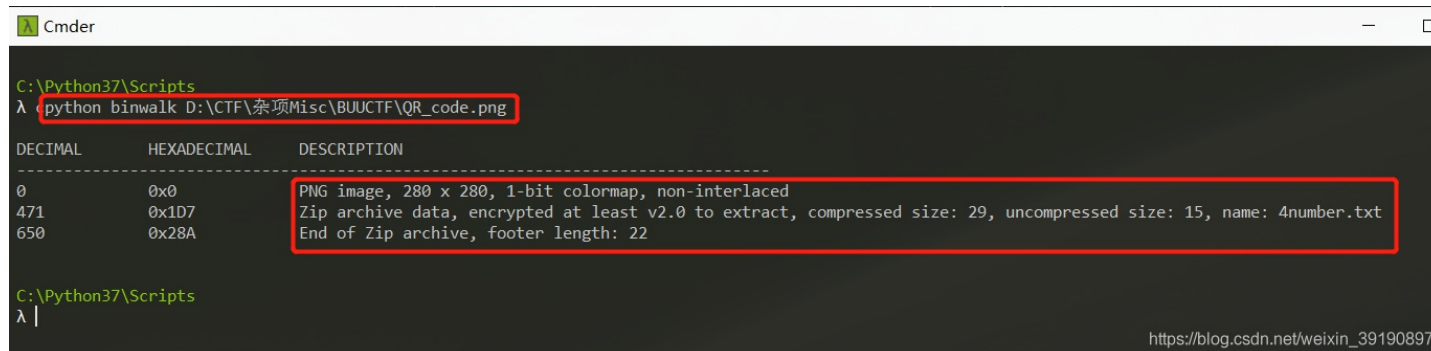
Usage: binwalk [OPTIONS] [FILE1] [FILE2] [FILE3] ...

Signature Scan Options:
-B, --signature          Scan target file(s) for common file signatures
-R, --raw=<str>          Scan target file(s) for the specified sequence of bytes
-A, --opcodes            Scan target file(s) for common executable opcode signatures
-m, --magic=<file>       Specify a custom magic file to use
-b, --dumb               Disable smart signature keywords
-I, --invalid            Show results marked as invalid
-x, --exclude=<str>     Exclude results that match <str>
-y, --include=<str>     Only show results that match <str>

Extraction Options:
-e, --extract            Automatically extract known file types
```

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

3、使用 Binwalk 分析目标图片，可以看到隐藏着几个文件：



```
Cmder
C:\Python37\Scripts
λ python binwalk D:\CTF\杂项Misc\BUUCTF\QR_code.png

DECIMAL      HEXADECIMAL    DESCRIPTION
-----
0            0x0            PNG image, 280 x 280, 1-bit colormap, non-interlaced
471          0x1D7         Zip archive data, encrypted at least v2.0 to extract, compressed size: 29, uncompressed size: 15, name: 4number.txt
650          0x28A         End of Zip archive, footer length: 22

C:\Python37\Scripts
λ |
```

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

4、使用 Binwalk 进一步分离出二维码图片中隐藏的文件：

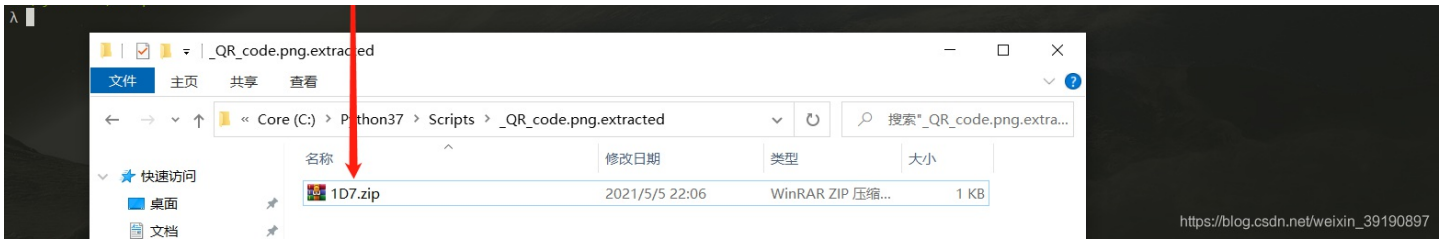


```
C:\Python37\Scripts
λ python binwalk -e D:\CTF\杂项Misc\BUUCTF\QR_code.png

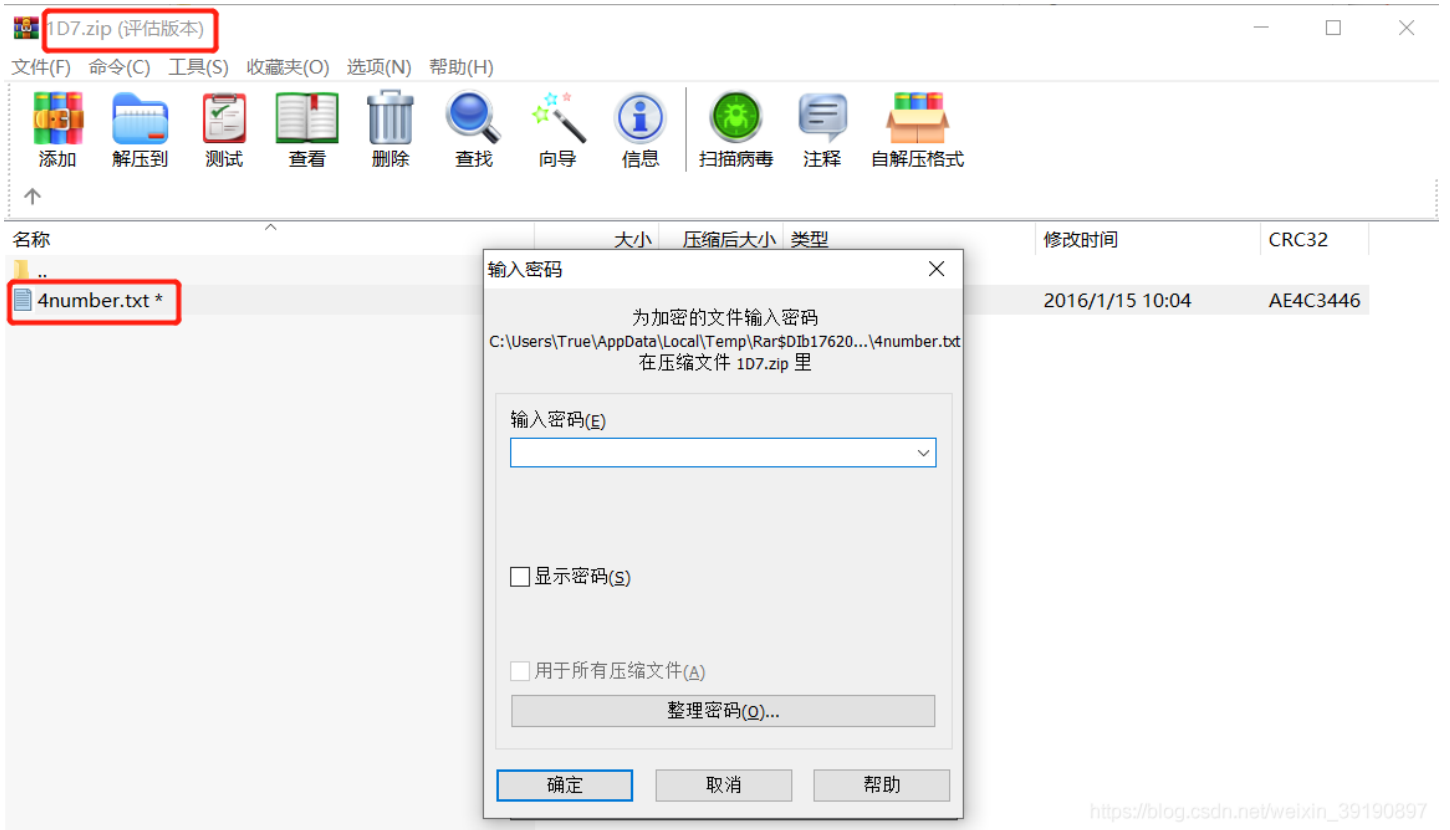
DECIMAL      HEXADECIMAL    DESCRIPTION
-----
0            0x0            PNG image, 280 x 280, 1-bit colormap, non-interlaced

WARNING: Extractor.execute failed to run external extractor '7z x -y '%e' -p ''': [WinError 2] 系统找不到指定的文件。; '7z x -y '%e' -p '' might not be installed correctly
471          0x1D7         Zip archive data, encrypted at least v2.0 to extract, compressed size: 29, uncompressed size: 15, name: 4number.txt
650          0x28A         End of Zip archive, footer length: 22

C:\Python37\Scripts
```



5、打开分离出来的压缩文件，提示需要进行密码才能查看文件：



6、根据提示是 4 位数的密码，直接使用 Ziperello 工具对 ZIP 文件进行暴力破解（下载链接：[https://pan.baidu.com/s/17TxQVXE8oZZA\\_dsyJglaqQ](https://pan.baidu.com/s/17TxQVXE8oZZA_dsyJglaqQ)；提取码：2333）：



字符类型: 固定字符集

字符集

- 数字 (0-9)
- 小写字母 (a-z)
- 大写字母 (A-Z)
- 特殊符号 (!@...)
- 空格
- 所有印刷字符

最小密码长度 = 4

最大密码长度 = 4

起始密码

7000|



步骤 3

1. 选择字符集类型: 固定或自设。
- 定义可能存在于密码中的字符集。
3. 设置最小及最大密码长度。
4. 输入起始密码 (非必需)

BACK

步骤 3 / 4: 暴力破解设定

NEXT

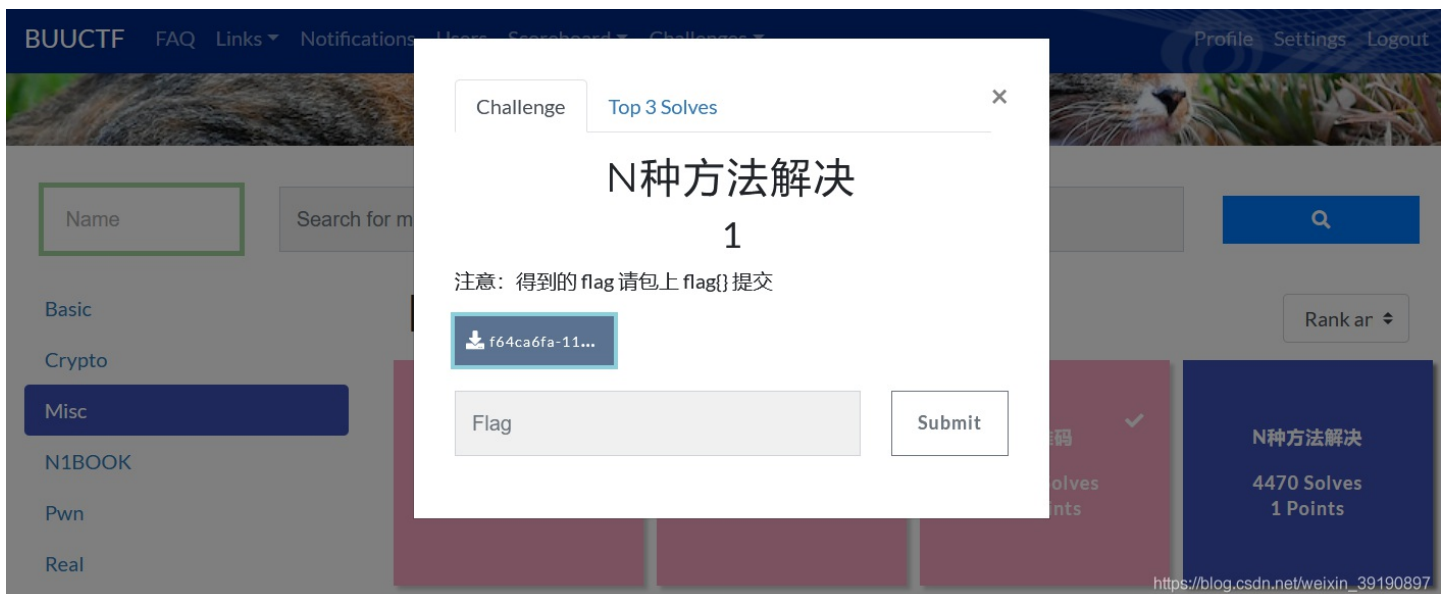
获得密码为 7639:



7、打开加密文件，获得 Flag:



### N0.3 Base64编码还原图片

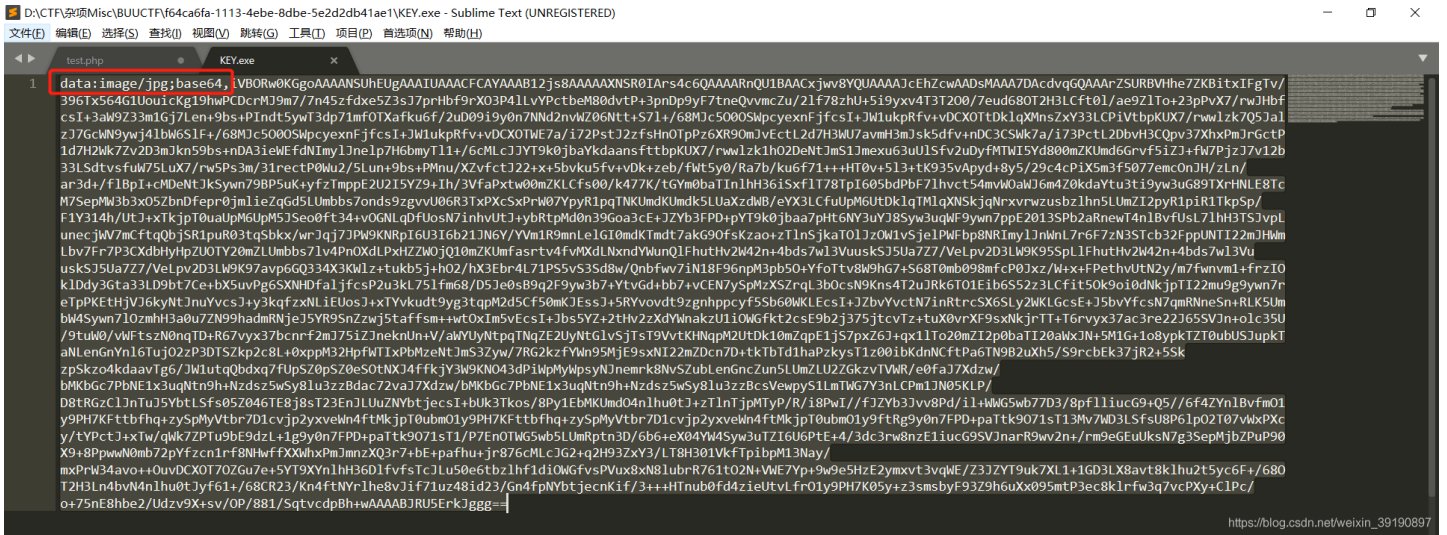


1、下载后是一个 KEY.exe 文件:



名称	修改日期	类型	大小
KEY.exe	2015/10/30 10:52	应用程序	4 KB

### 2、使用 Sublime 打开，发现是图片转换成 Base 64编码：



### 3、使用站长之家在线转换该 base 64 编码并还原成图片：



### 4、接着使用微信扫描二维码或者在线识别二维码图片，可获得 flag：



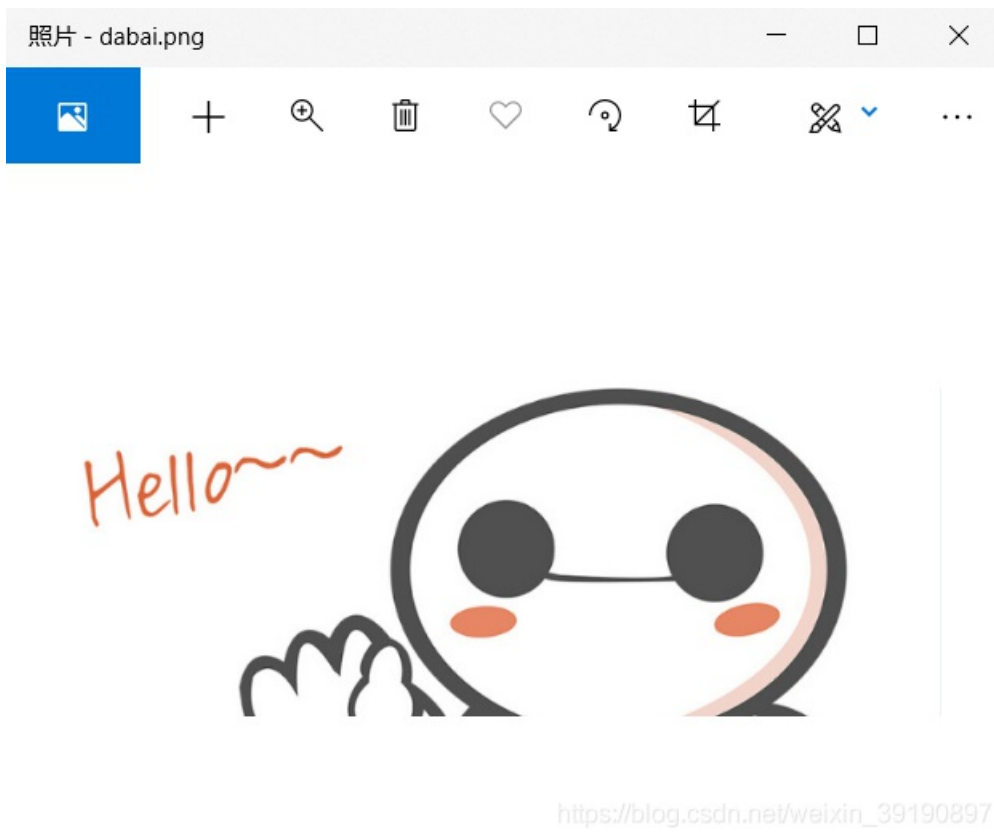
[生成二维码](#)[再解一个](#)[快速美化器](#)[高级美化器](#)[https://blog.csdn.net/qq\\_3940897](https://blog.csdn.net/qq_3940897)

## No.4 winhex修改图片大小

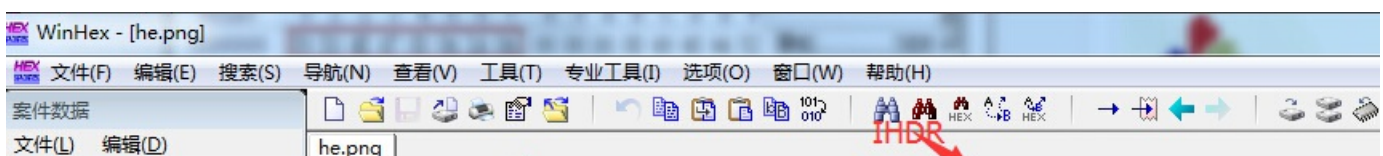




1、下载后打开图片如下：



2、根据题目提示，需要将图片变大，可使用 winhex 打开图片并修改图片的宽、高。先来介绍下相关知识：



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52
00000010	00	00	01	F4	00	00	01	F4	08	02	00	00	00	44	B4	48
00000020	DD	00	00	00	19	74	45	58	74	53	6F	66	74	77	61	72
00000030	65	00	41	64	6F	62	65	20	49	6D	61	67	65	52	65	61
00000040	64	79	71	C9	65	3C	00	00	03	22	69	54	58	74	58	4D
00000050	4C	3A	63	6F	6D	2E	61	64	6F	62	65	2E	78	6D	70	00
00000060	00	00	00	00	3C	3F	78	70	61	63	6B	65	74	20	62	65
00000070	67	69	6E	3D	22	EF	BB	BF	22	20	69	64	3D	22	57	35

CRC

### 5.2.3 分析PNG图像文件结构(2)

表5-15归纳了pic1.png图像文件中文件头数据块(IHDR)中各字段的含义。由于PNG文件使用Big-Endian顺序存储数据，所以不需要反转字节数据理解。

表5-15 pic1.png图像文件中文件头数据块(IHDR)的各字段含义

十六进制值	描述
00 00 00 0D	文件头的数据长度, 00 00 00 0D =13
49 48 44 52	数据块类型标志, 49 48 44 52的ASCII值等于IHDR
00 00 00 C8	图像的宽度, 00 00 00 C8 = 200
00 00 00 96	图像的高度, 00 00 00 96 = 150
08	色深, 表示2的8次幂等于256色
03	03表示索引图像
00	00表示使用Deflate压缩编码压缩图像数据
00	00表示为将来使用更好的压缩方法预留
00	00表示非隔行扫描
AC 02 37 2B	AC 02 37 2B表示CRC

从表5-14看到pic1.png文件的文件头数据块(IHDR)结构中的CRC字段的值为AC 02 37 2B, 这个CRC值是按照从数据块类型标志字段到CRC字段前一字节的数据计算而来的, 即使用数据49 48 44 52 00 00 00 C8 00 00 00 96 08 03 00 00 00计算, CRC的计算代码如下:

3、故使用 winhex 工具打开图片, 找到宽、高位置, 修改其值:

The screenshot shows the WinHex interface with the file 'dabai.png' open. The main window displays the hex and ASCII data of the PNG header. The IHDR chunk is visible, with the width field (00 00 00 C8) and height field (00 00 00 96) highlighted in red. The CRC field (AC 02 37 2B) is also visible. The right sidebar shows file metadata: 'dabai.png', 'D:\CTF\杂项\Misc\BUUCTF\37914', 'File size: 147 KB (150,560 bytes)', 'Default Edit Mode: original', 'State: original', 'Undo level: 0', 'Undo reverses: n/a', and 'Creation time: 2021/05/05 22:43:23'.

```

000000E0 62 D1 E5 34 BC 34 0D AD 56 CB 1A 8D 86 35 9B 4D bÑã44 -VÈ t5>M
000000F0 17 B3 B3 B3 36 37 37 E7 72 98 21 70 87 DC 6E B7 **677qr~!p+Un·
00000100 3D FC 90 01 E1 11 0F 61 22 CA E5 B2 8B 4A A5 62 =ù á a"Êâ&lt;Jwb
00000110 A5 52 C9 D5 B5 5A CD AA D5 AA CB 88 7A BD EE 22 ¥RÉÖpzi*ô*E"z%1"
00000120 DC 45 3A FB 11 E9 24 3E 80 1E B7 91 87 34 2F 81 ÛE:ú é$>e ·'+4/
00000130 30 8B F4 A1 DE 9D DB 7E F4 BB D9 1B 3F 39 72 1C 0<ô;P Û~ô»Ù ?9r
00000140 0E D8 3E D9 B4 9C 9C E6 C8 91 23 C7 41 8C 18 C2 Ø>Ù'öææÈ'#çAG Å
00000150 FB 89 4D 6A 9E 12 1F D4 88 7E 32 99 CA B8 1D 14 û&Mjž ô~2=è;
00000160 5E 6A D6 6F 0F A1 1C 1F 1F B7 6D DB B6 D9 8E 1D ^j0o ; m0gÜž
00000170 3B 6C E7 CE 9D 2E 63 36 3D 3D 6D 93 93 93 2E 26 ;lçf .c6==m""".&
00000180 26 26 9C 90 4E 4D 4D CD 93 53 08 21 E1 47 1C 91 &æ NMMI"S !âG '

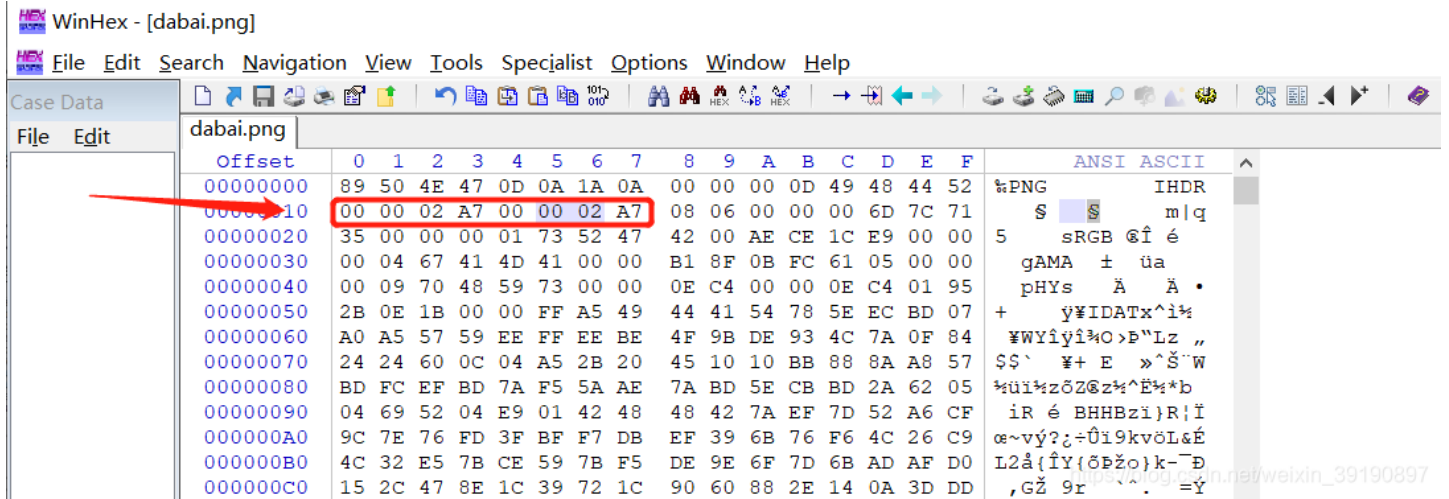
```

```

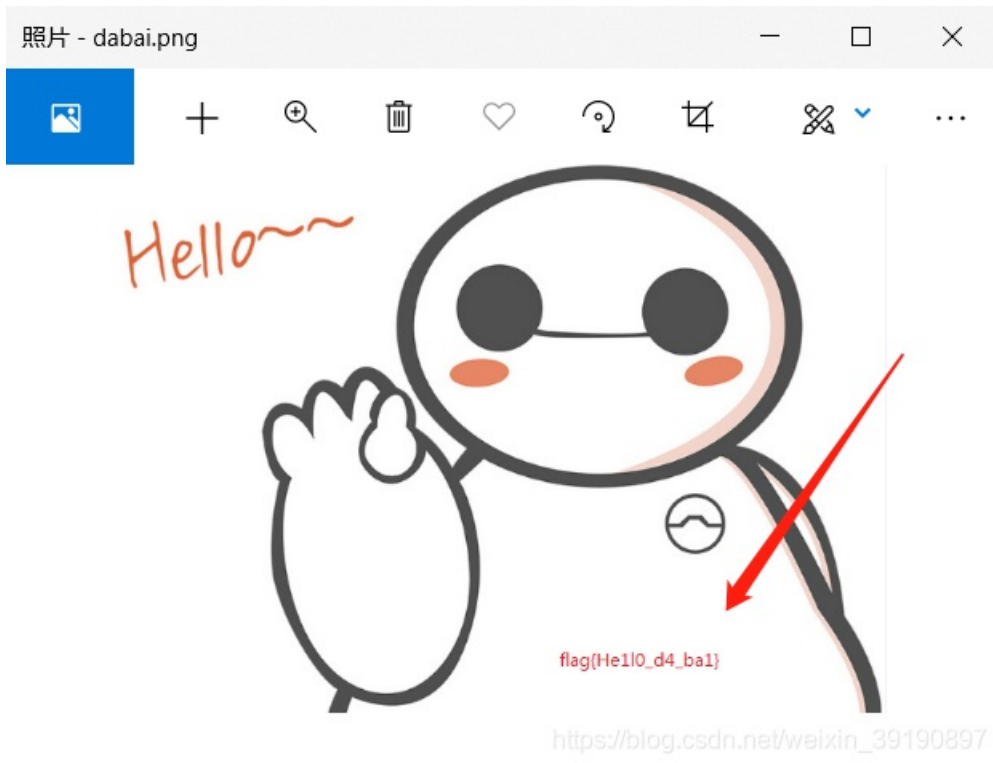
Last write time: 2016/08/31
                  13:56:22
Attributes:      A
Icons:           0
Mode:            hexadecimal
Offsets:         hexadecimal
Bytes per page: 46x16=736
Window #:       1
No. of windows: 2in_39190891

```

第 18、19 位是长，22、23是宽，我们把宽度设置大一点，就设置和长度一样好了：



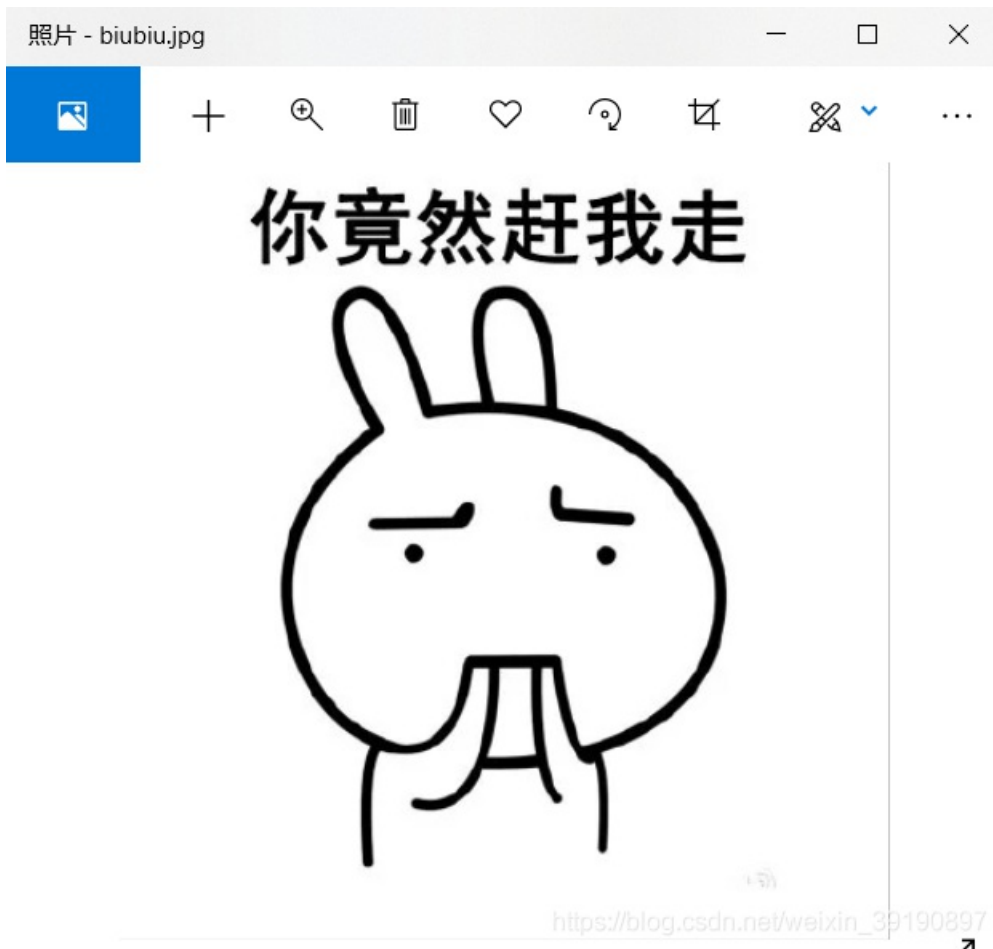
4、保存后重新打开图片，可获得 flag:



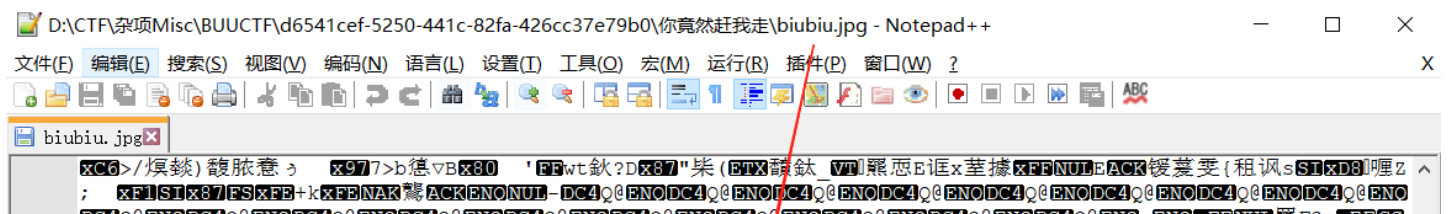
## No.5 编辑器查看图片隐写



1、下载后打开是一张图片:

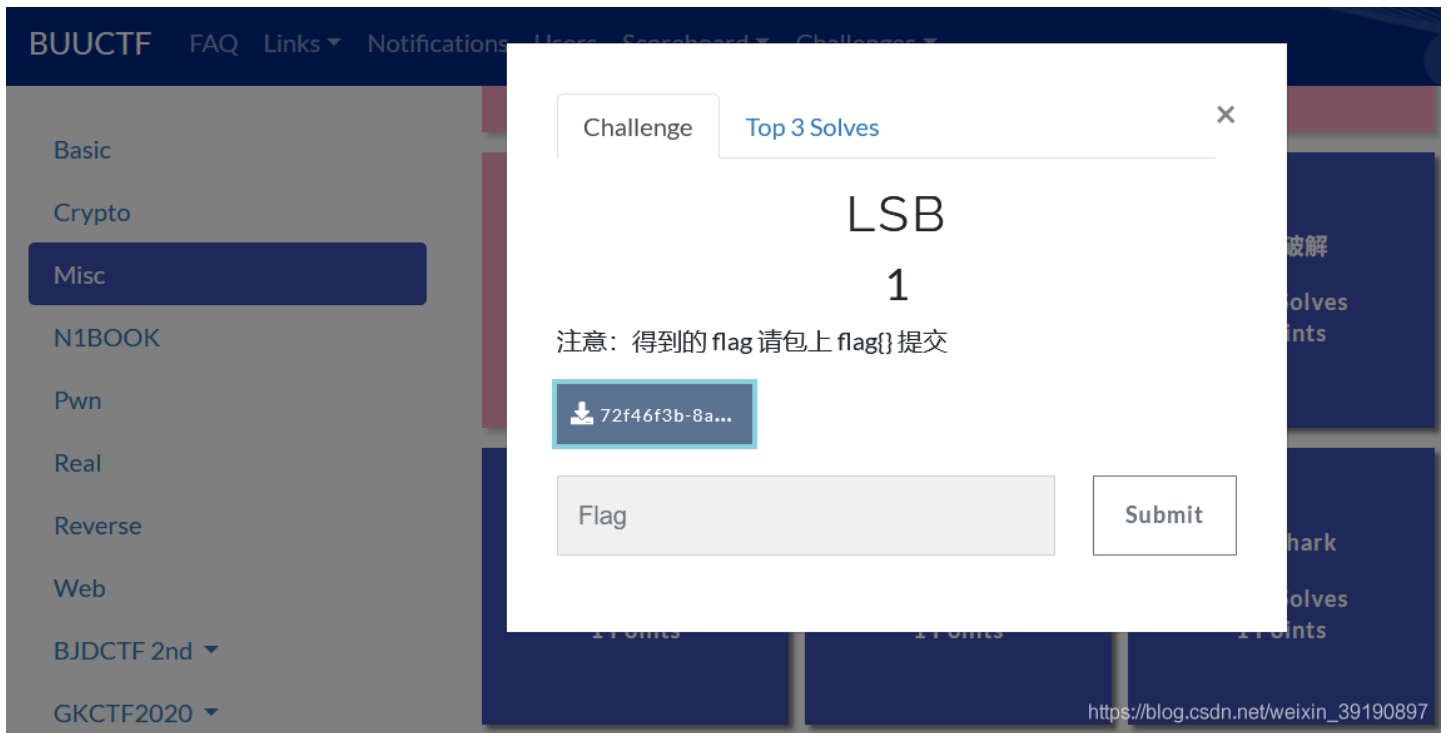


2、使用 Notepad 编辑器打开查看图片，搜索 flag，可在末尾获得 flag:

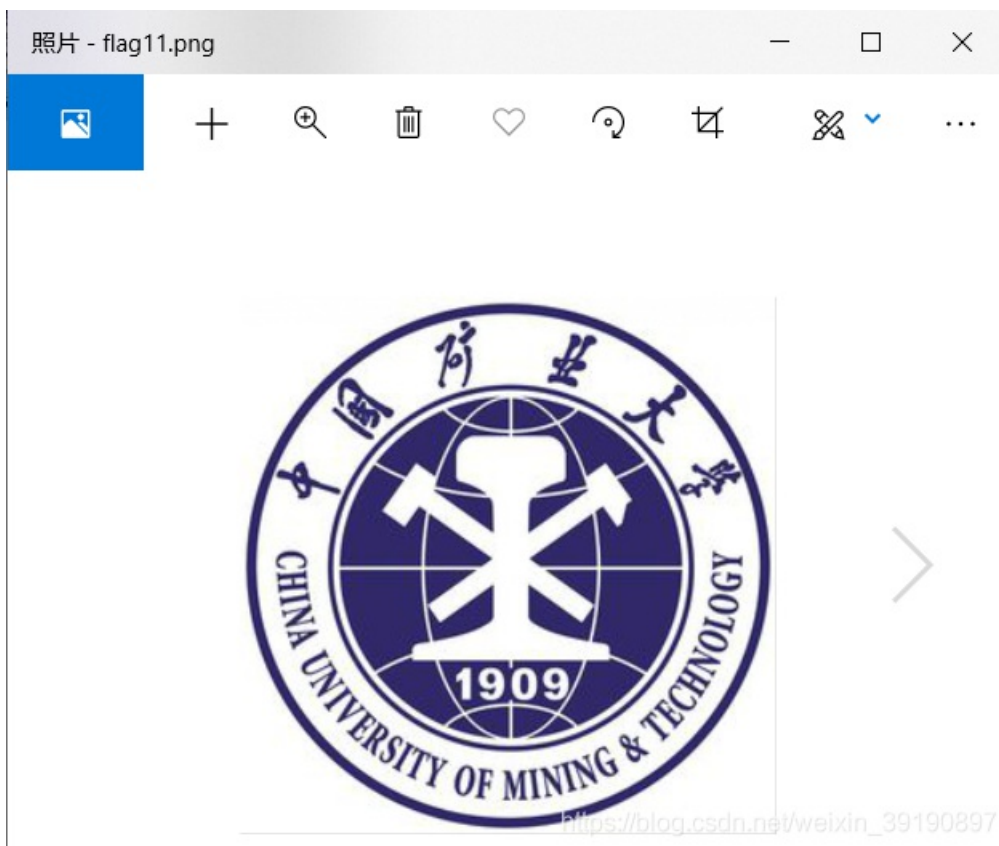








1、下载后是一张 png 图片：



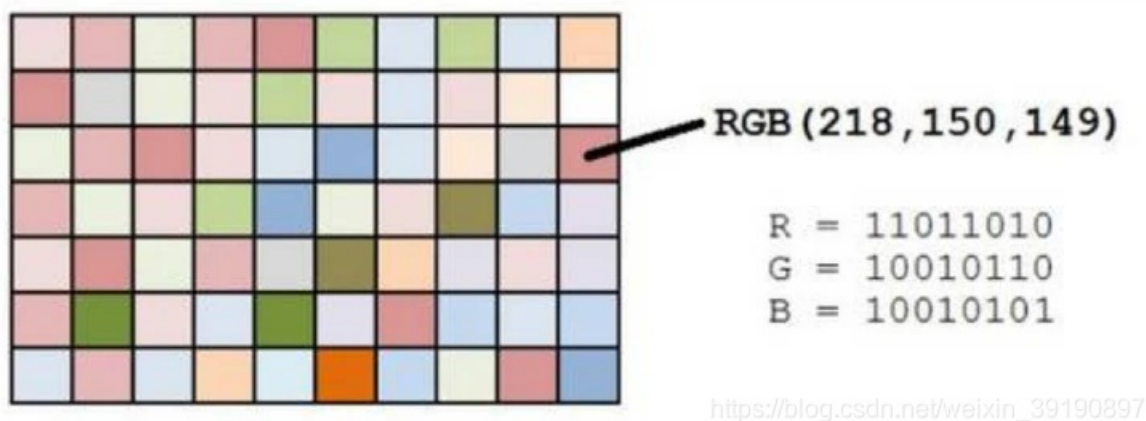
题目的提示很明显了，这是一道 LSB 隐写题。

LSB 全称为 least significant bit，是最低有效位的意思。Lsb 图片隐写是基于 lsb 算法的一种图片隐写术，以下统称为 lsb 隐写，这是一种常见的信息隐藏方法。lsb 隐写很实用，算法简单，能存储的信息量也大，是 CTF 比赛中的常客。




#### 【原理介绍】

png 图片是一种无损压缩的位图格式，也只有在不压缩或者无压缩的图片（BMP）上实现 lsb 隐写。如果图像是 jpg 图片的话，就没法使用 lsb 隐写了，原因是 jpg 图片对像素进行了有损压缩，我们修改的信息就可能会在压缩的过程中被破坏。而 png 图片虽然也有压缩，但却是无损压缩，这样我们修改的信息也就能得到正确的表达，不至于丢失。BMP 的图片也是一样的，是没有经过压缩的。BMP 图片一般是特别的大，因为 BMP 把所有的像素都按原样储存，没有进行压缩。

png 图片中的图像像素一般是由 RGB 三原色（红绿蓝）组成，每一种颜色占用 8 位，取值范围为  $0x00\sim0xFF$ ，即有 256 种颜色，一共包含了 256 的 3 次方的颜色，即 16777216 种颜色。而人类的眼睛可以区分约 1000 万种不同的颜色，这就意味着人类的眼睛无法区分余下的颜色大约有 6777216 种。



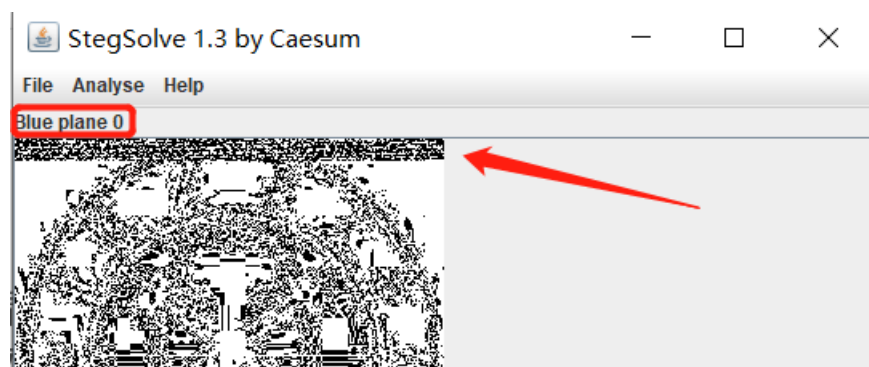
LSB 隐写就是修改 RGB 颜色分量的最低二进制位也就是最低有效位（LSB），而人类的眼睛不会注意到这前后的变化，每个像素可以携带 3 比特的信息。

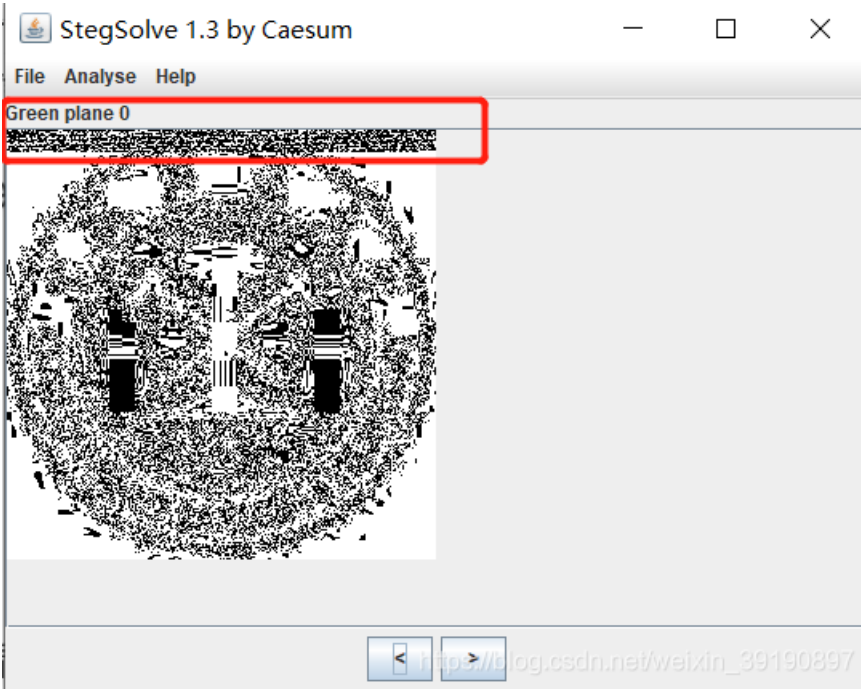
Color (Green)	Base 10	Binary	Change
	238	11101110	+3
	235	11101011	(base)
	232	11101000	-3

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

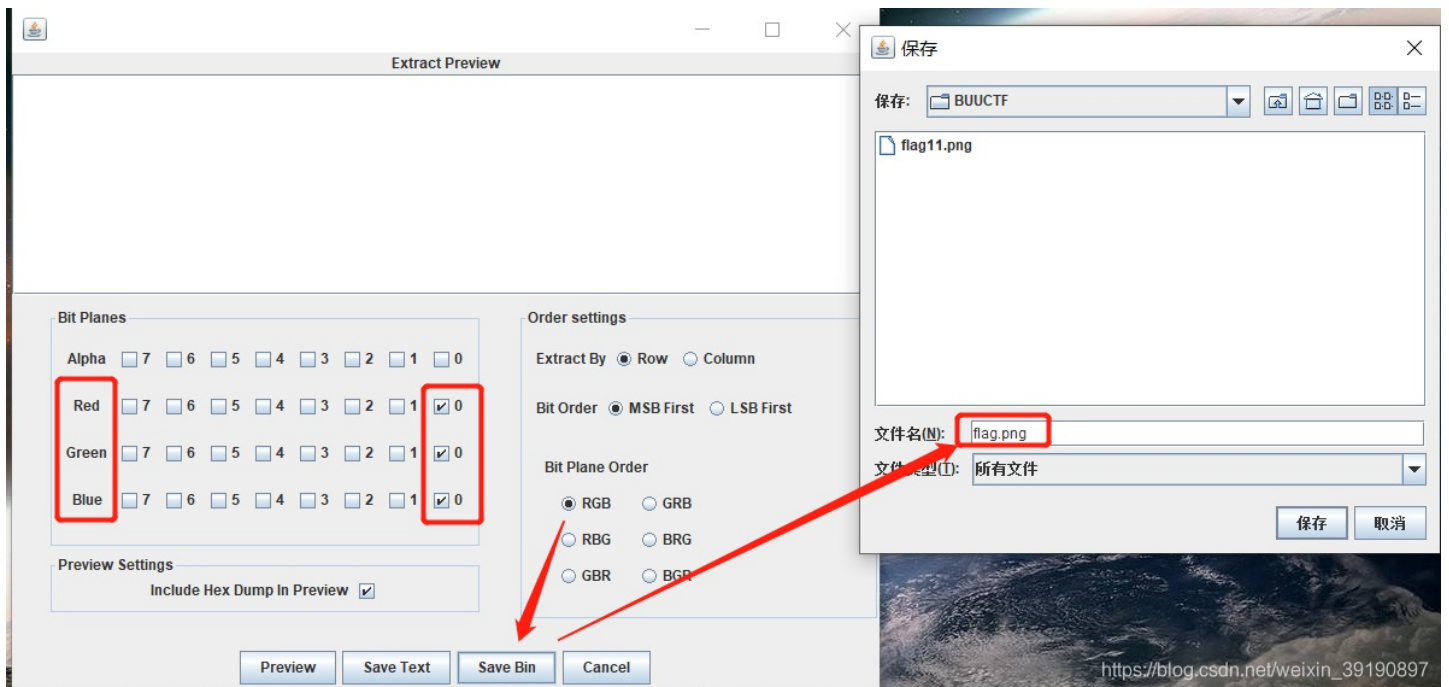
上图我们可以看到，十进制的 235 表示的是绿色，我们修改了在二进制中的最低位，但是颜色看起来依旧没有变化。我们就可以修改最低位中的信息，实现信息的隐写。

2、回到这道题目中来，针对 LSB 隐写，同样可以使用隐写图片查看的神器——**stegsolve**来分析，用 stegsolve 打开这张图片，发现 `red0`, `green0`, `blue0` 这三个通道的图片上方有异样：





3、因此用 Analyse 菜单栏里的 Data Extract 功能（数据抽取，图片中隐藏数据的抽取）查看图片，将三个位置的 0 通道勾选并导出保存为 png 格式的文件：





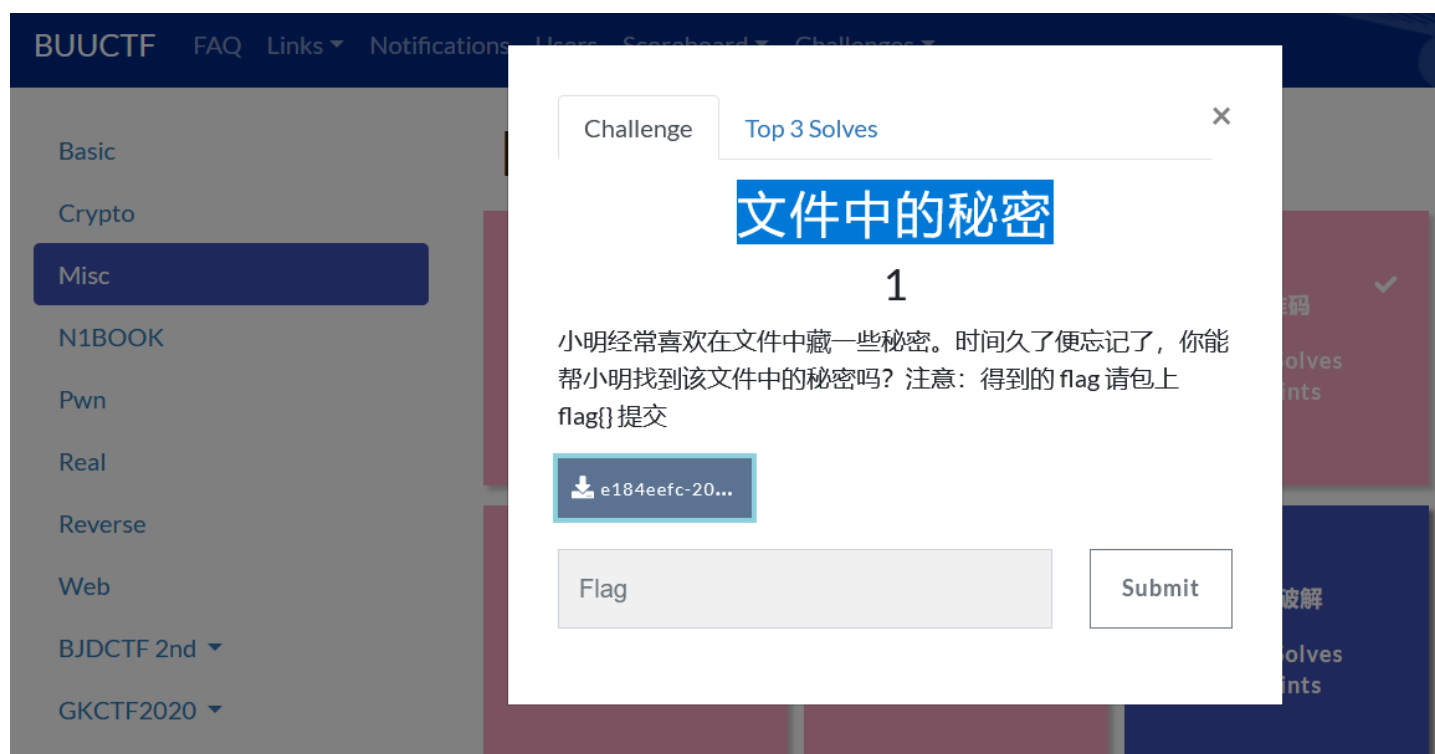
4、获得 flag.png 图片，发现是个二维码：



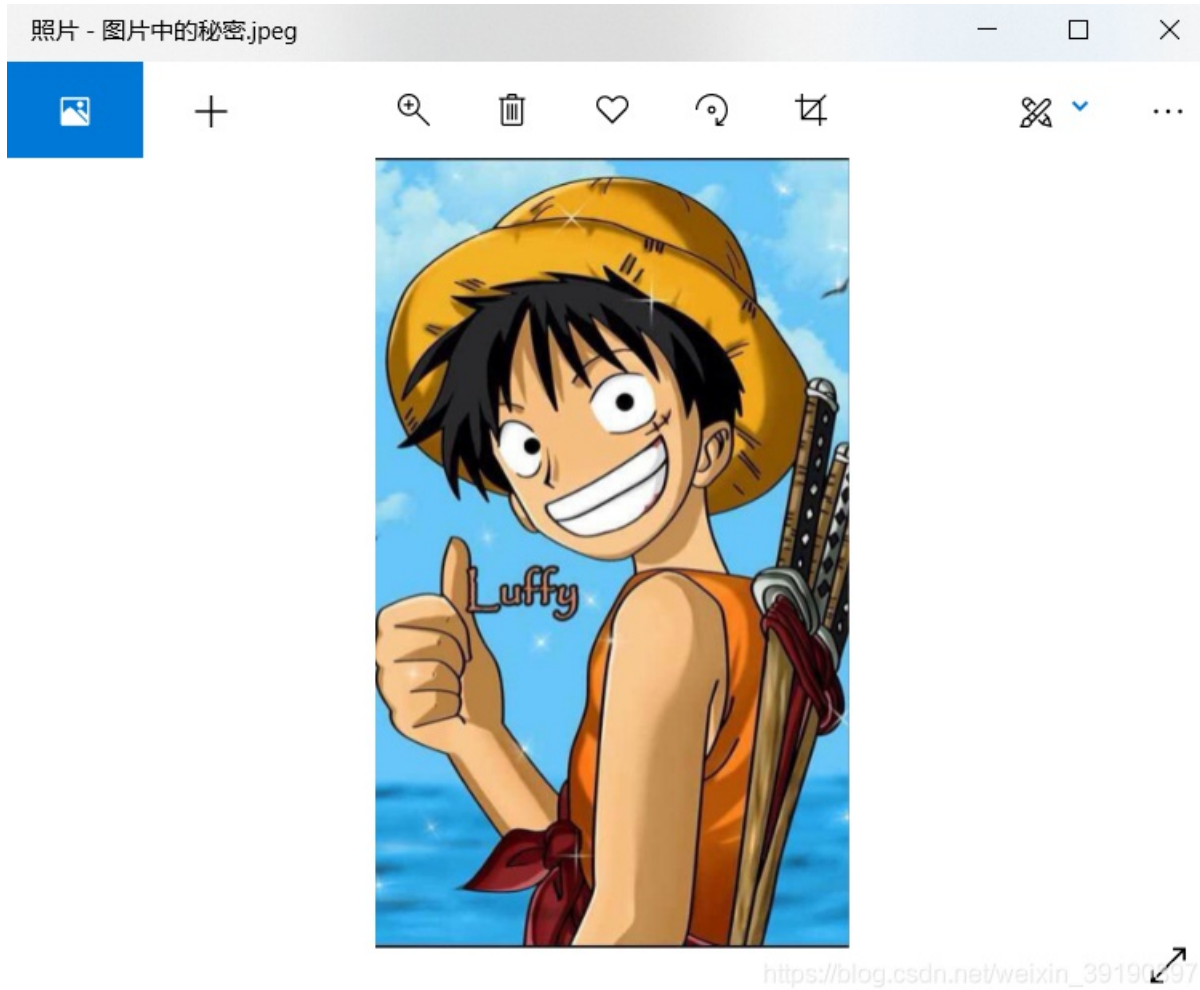
5、在线解码，获得 flag：



## No.7 图片属性中隐藏信息



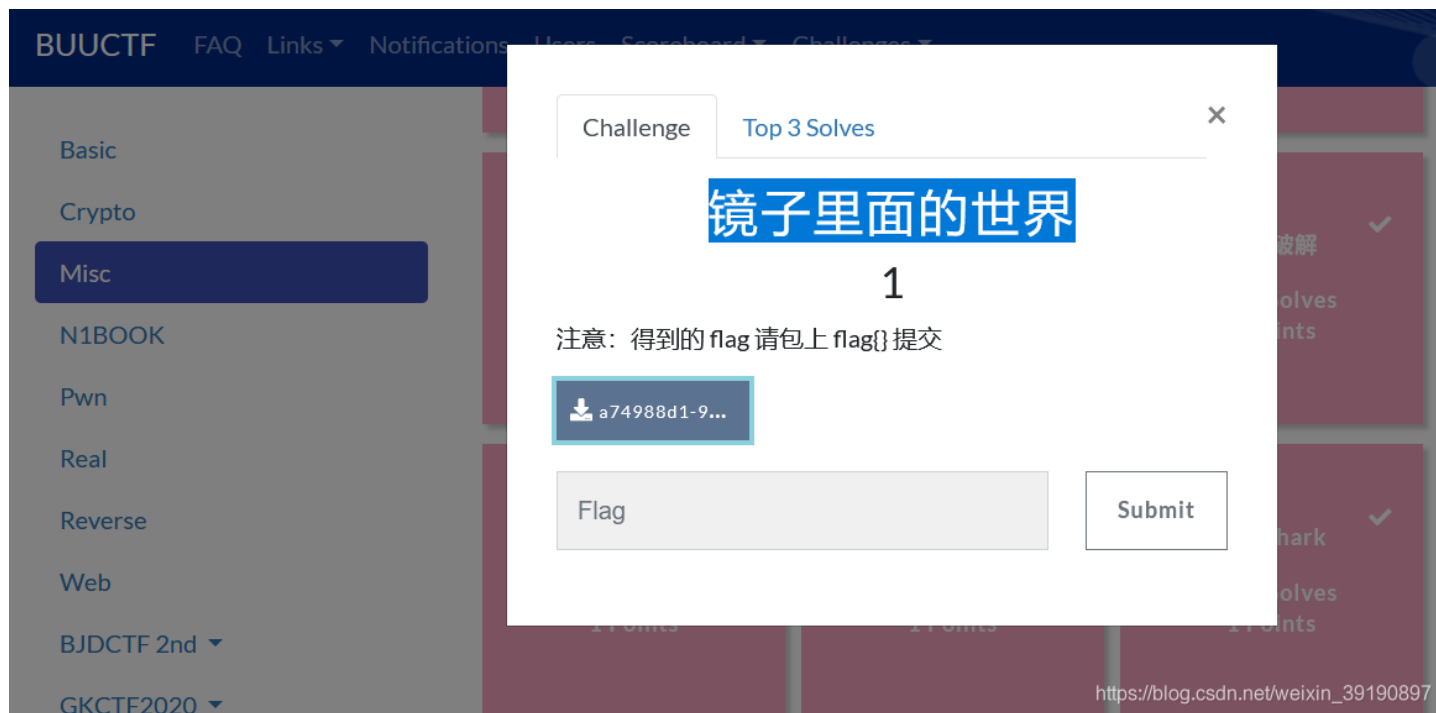
1、打开是一张路飞的图：



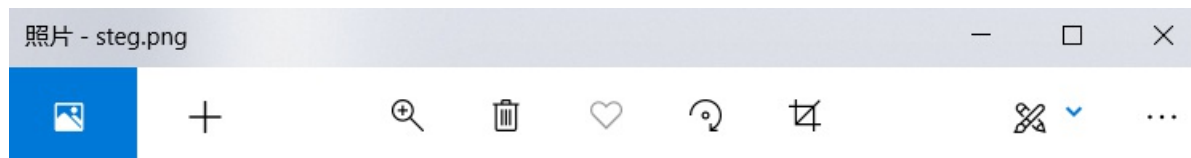
2、查看图片属性可直接获得 flag:



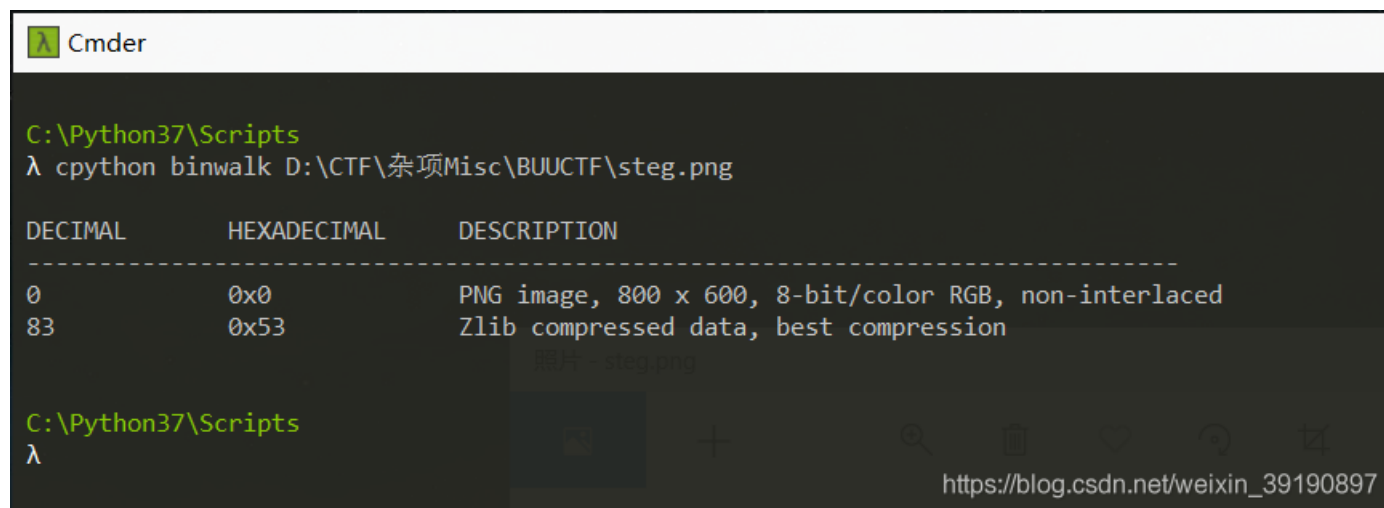
## No.8 LSB隐写的数据抽取



1、下载后打开是一张图片：



2、猜测是隐写，使用 binwalk，查看是否有隐藏文件，然而并没有：



```
C:\Python37\Scripts
λ cpython binwalk D:\CTF\杂项Misc\BUUCTF\steg.png

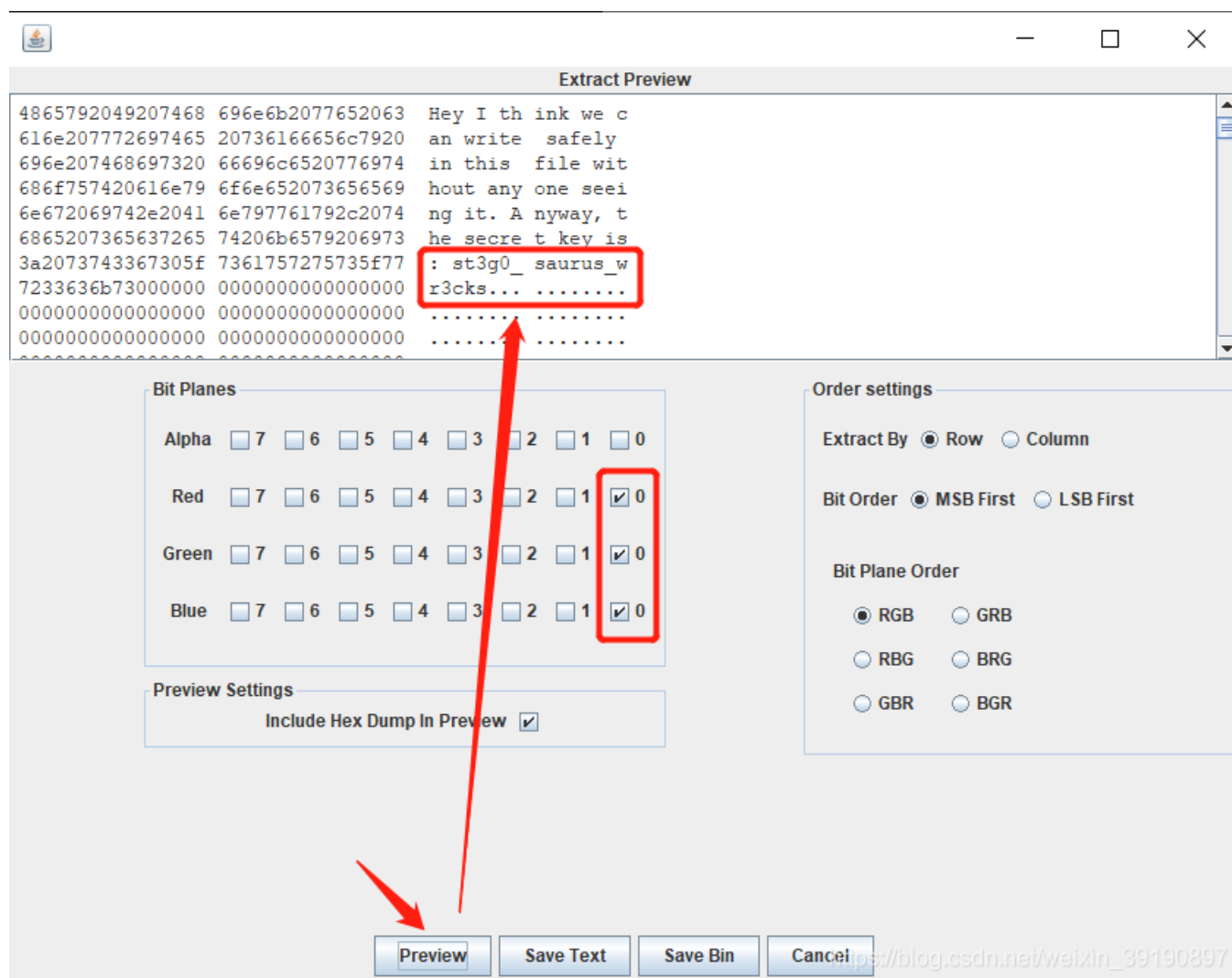
DECIMAL      HEXADECIMAL    DESCRIPTION
-----
0            0x0           PNG image, 800 x 600, 8-bit/color RGB, non-interlaced
83          0x53          Zlib compressed data, best compression

C:\Python37\Scripts
λ
```

3、使用 Stegsolve，查看是否存在 LSB 隐写，发现 `red0`，`green0`，`blue0` 这三个通道的图片均为异样的黑色：



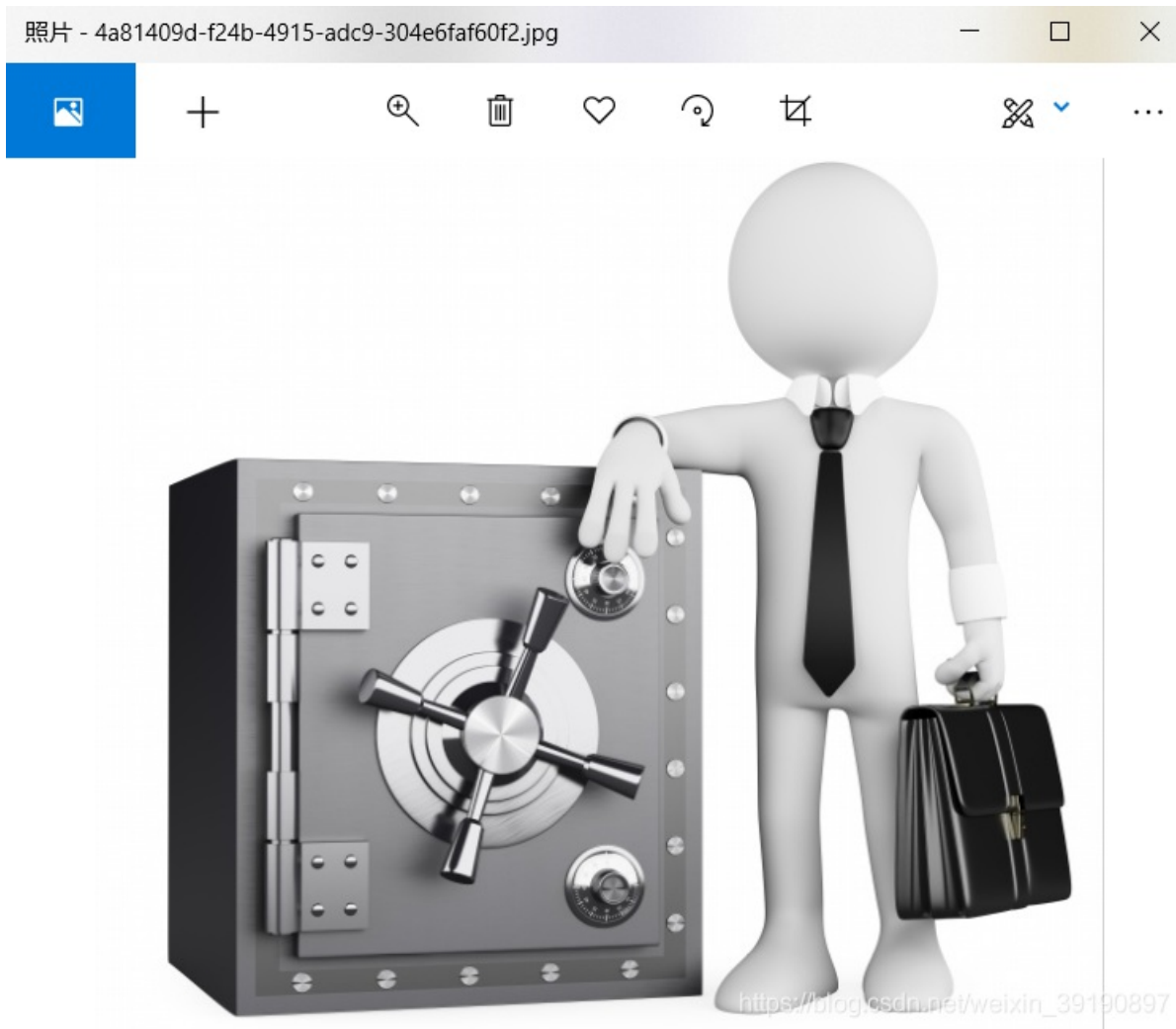
4、因此用 Analyse 菜单栏里的 Data Extract 功能（数据抽取，图片中隐藏数据的抽取）查看图片，将三个位置的 0 通道勾选并预览，发现 flag:



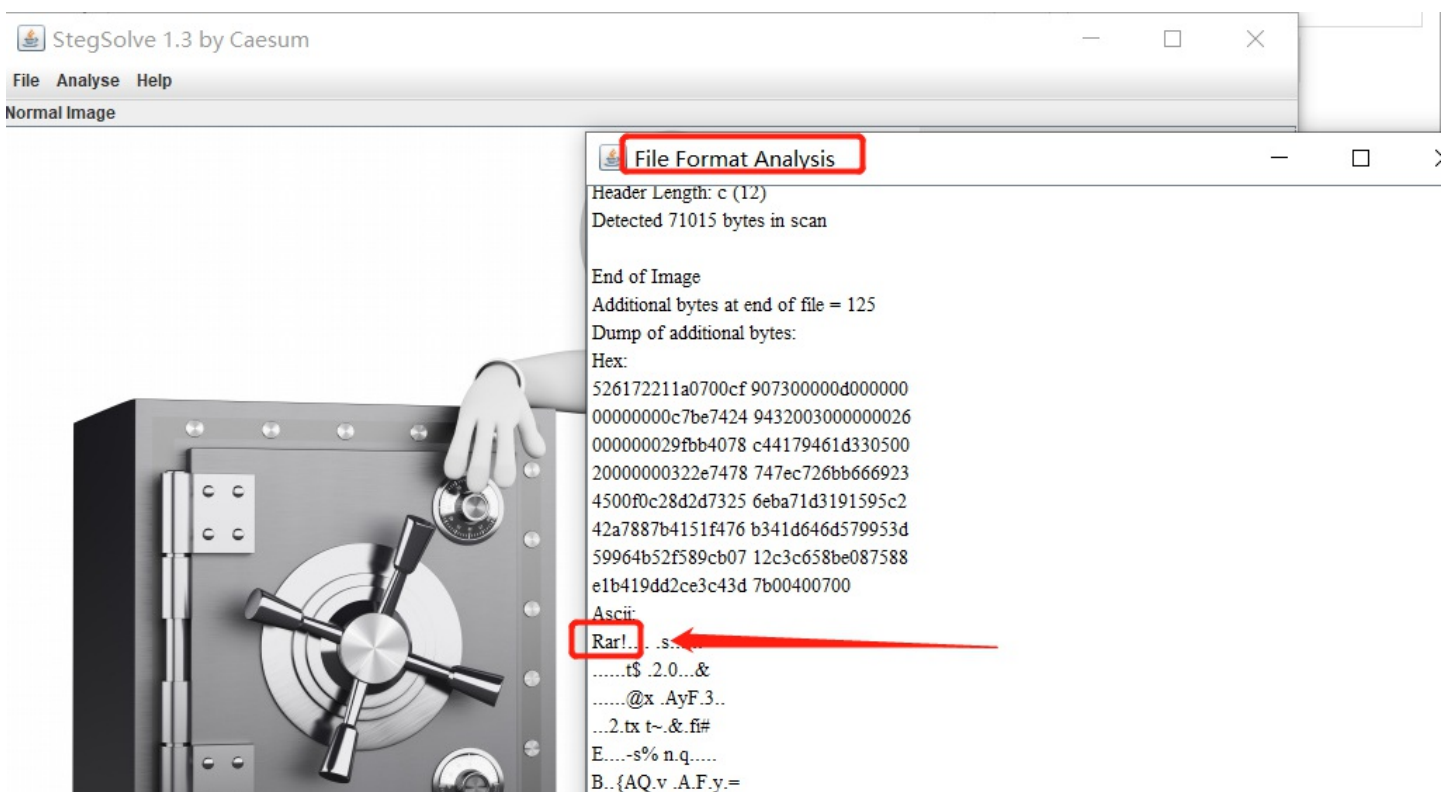
## No.9 RAR文件加密的爆破



1、下载发现是一张图片：



2、使用 **stegsolve** 的格式分析功能，发现 RAR 关键词：







Y.KR.....X.u.  
.....= {.@..

OK

https://blog.csdn.net/weixin\_39190897

### 3、动用 Binwalk 工具分析，分离出 rar 压缩文件：

```
C:\Python37\Scripts
λ python binwalk D:\CTF\BUUCTF\4a81409d-f24b-4915-adc9-304e6faf60f2.jpg

DECIMAL      HEXADECEMIAL  DESCRIPTION
-----
0            0x0           JPEG image data, JFIF standard 1.01
30          0x1E          TIFF image data, big-endian, offset of first image directory: 8
79903      0x1381F      RAR archive data, version 4.x, first volume type: MAIN_HEAD

C:\Python37\Scripts
λ python binwalk -e D:\CTF\BUUCTF\4a81409d-f24b-4915-adc9-304e6faf60f2.jpg

DECIMAL      HEXADECEMIAL  DESCRIPTION
-----
0            0x0           JPEG image data, JFIF standard 1.01
30          0x1E          TIFF image data, big-endian, offset of first image directory: 8

WARNING: Extractor.execute failed to run external extractor 'unrar e '%e': [WinError 2] 系统找不到指定的文件。 , 'unrar e '%e'' might not be installed correctly
WARNING: Extractor.execute failed to run external extractor 'unrar -x '%e': [WinError 2] 系统找不到指定的文件。 , 'unrar -x '%e'' might not be installed correctly
79903      0x1381F      RAR archive data, version 4.x, first volume type: MAIN_HEAD

C:\Python37\Scripts
λ
```

The file explorer window shows the following file:

名称	修改日期	类型	大小
1381F.rar	2021/5/7 21:38	WinRAR 压缩文件	1 KB

https://blog.csdn.net/weixin\_39190897

### 4、打开压缩文件，发现加密了：

1381F.rar (评估版本)

文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)

添加 解压到 测试 查看 删除 查找 向导 信息 扫描病毒 注释 保护 自解压格式

名称	大小	压缩后大小	类型	修改时间	CRC32
2.txt *				2015/3/25 8:14	7840BB9F

输入密码

为加密的文件输入密码  
2.txt  
在压缩文件 1381F.rar 里

输入密码(E)

显示密码(S)

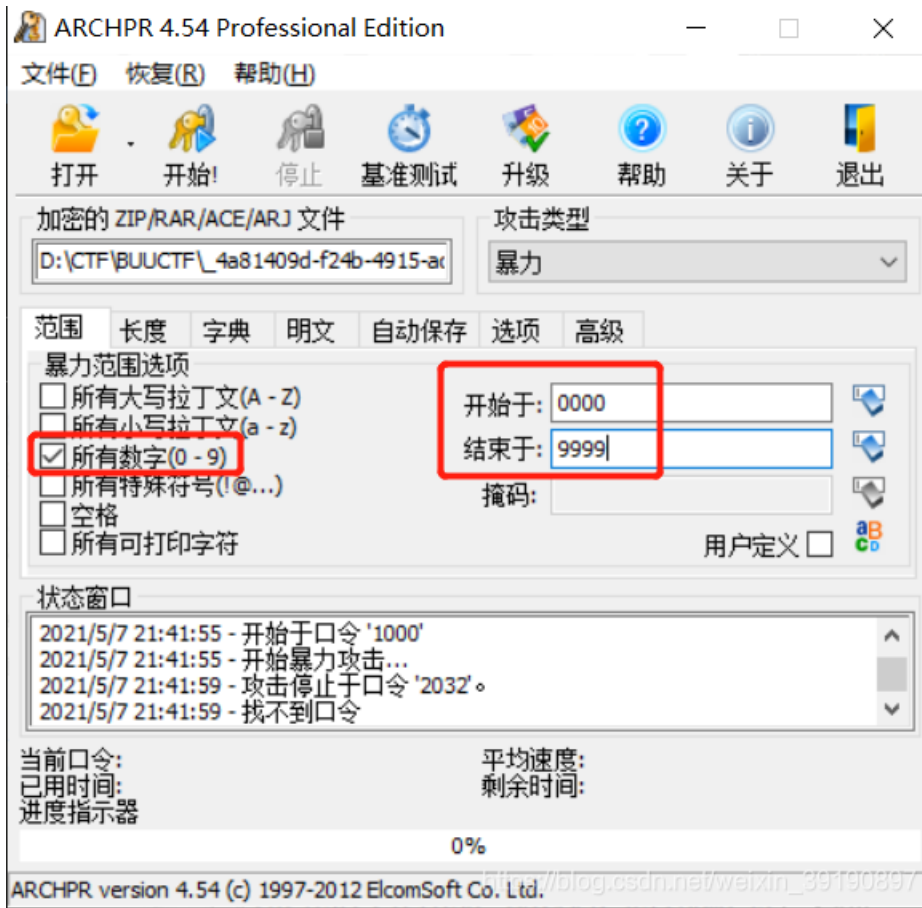
用于所有压缩文件(A)

整理密码(O)...

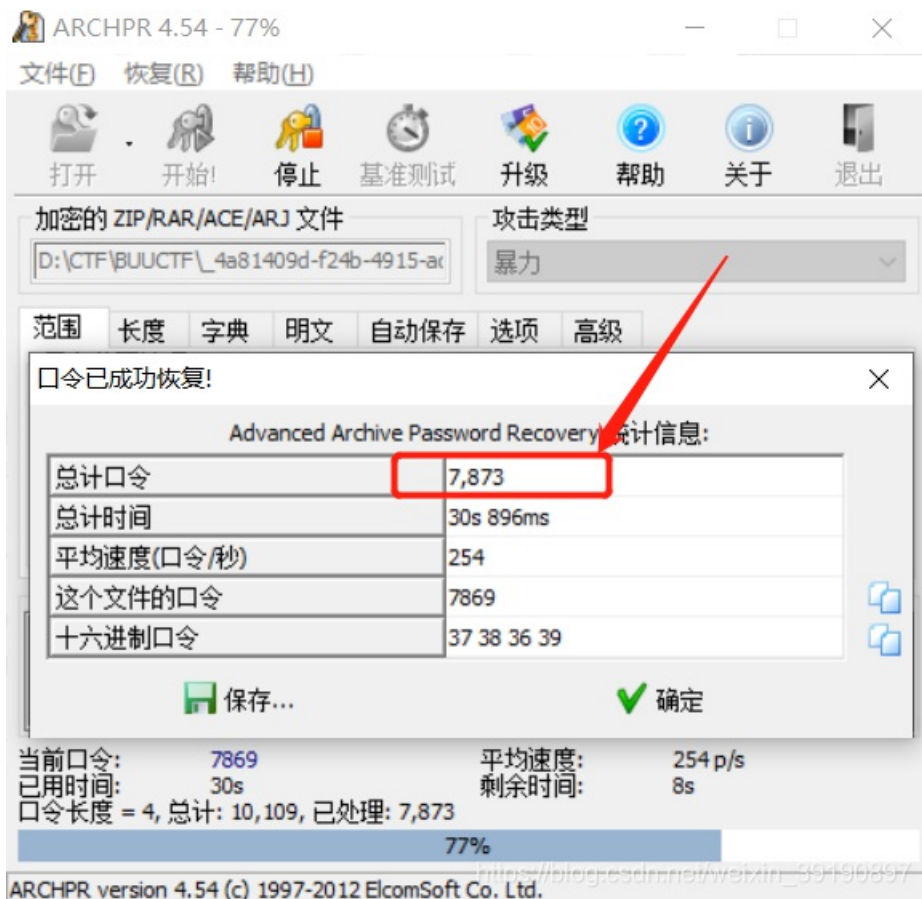
确定 取消 帮助

https://blog.csdn.net/weixin\_39190897

5、使用 ARCHPR 爆破工具尝试进行 4 位数密码的破解：



6、成功破解密码，获得 Flag：







```

0 F9 D3 7C B3 40 71 4F D1 C6 5D F4 2D B5 D4 94 A1 ùÓ|°@qOÑÆ]ô-μô";
0 BD AF 64 80 91 21 3D 1A B3 3C 3C 2B A1 6E 1E EE %¯de`!= °<<+;n î
0 5C EE AE EE 33 B2 AE AE 9B B9 79 0E B8 E1 71 A0 \i@i3°@@>¹y ,áq
0 40 21 46 78 F7 D5 A0 FE F0 79 E7 51 35 FB F5 7F @!Fx÷õ bðvcQ5ûö
0 48 F8 0A 59 E4 94 BB 1A 29 30 31 31 30 31 30 31 Hø Yä"» )0110101
0 31 30 31 31 30 31 31 31 31 30 31 31 30 30 31 30 1011011110110010
0 31 30 31 31 30 31 30 31 31 30 31 31 30 31 30 31 1011010110110101
0 30 30 30 31 31 30 30 31 31 30 31 31 31 30 30 31 0001100110111001
0 31 2E 00 00 00 1.

```

3、猜测可用 8 位二进制一组，转换成 ASCII 编码，脚本如下：

```

import binascii

a = ''01101011
01101111
01100101
01101011
01101010
00110011
01110011'''
b = ''
for ii in a.split('\n'):
    b += chr(int(ii,2))
print(b)

```

运行脚本，获得 flag：



## No.11 Winhex搜索获得flag

Challenge Top 3 Solves

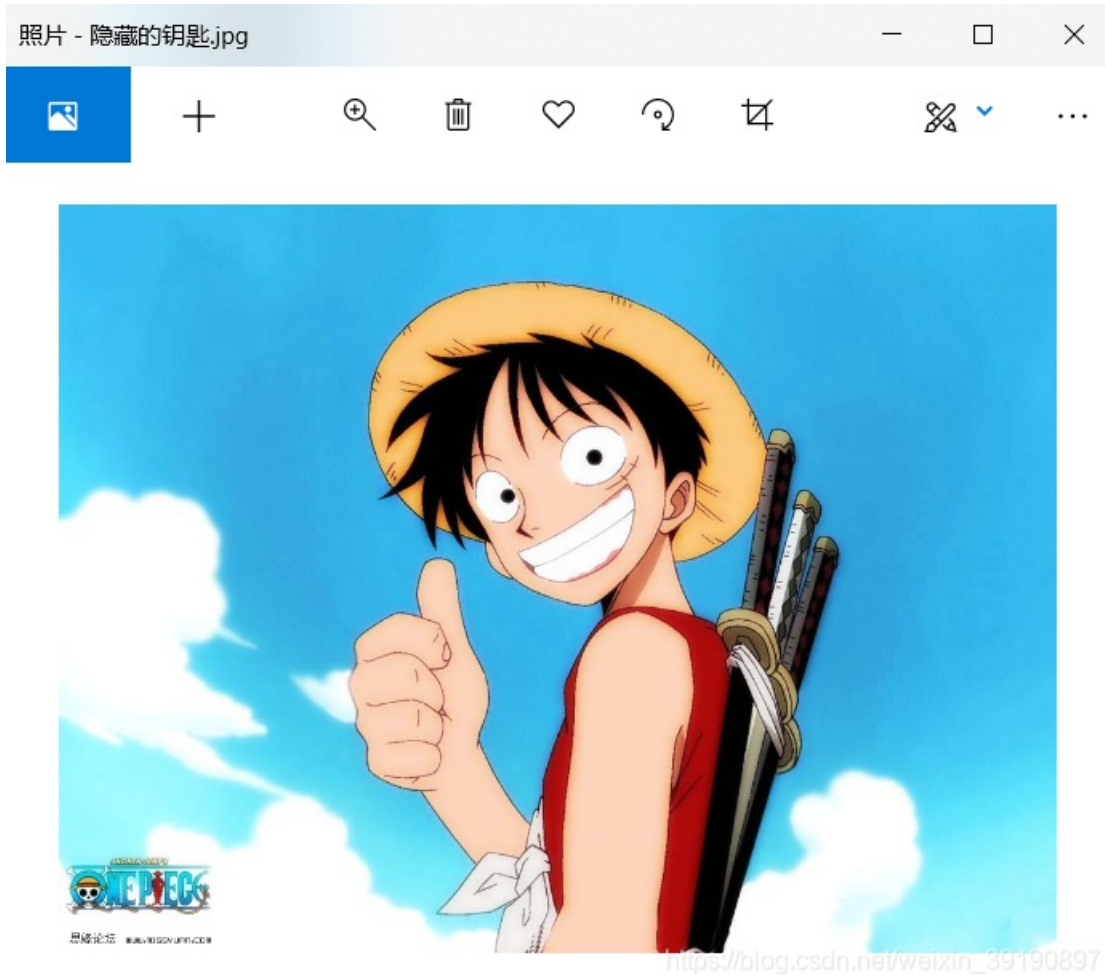
### 隐藏的钥匙

1

路飞一行人千辛万苦来到了伟大航道的终点，找到了传说中的 One piece，但是需要钥匙才能打开 One Piece 大门，钥匙就隐藏在下面的图片中，聪明的你能帮路飞拿到钥匙，打开 One Piece 的大门吗？注意：得到的 flag 请包上 flag{} 提交

8f6577ab-40...

1、下载后打开如下：

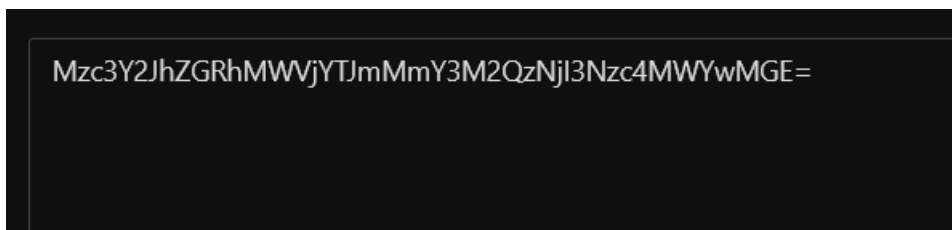


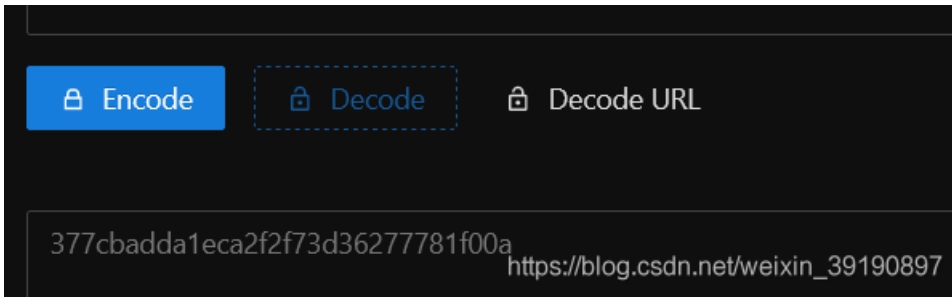
2、使用 winhex 打开图片并搜索 flag，看到 base64 编码字符串：

```

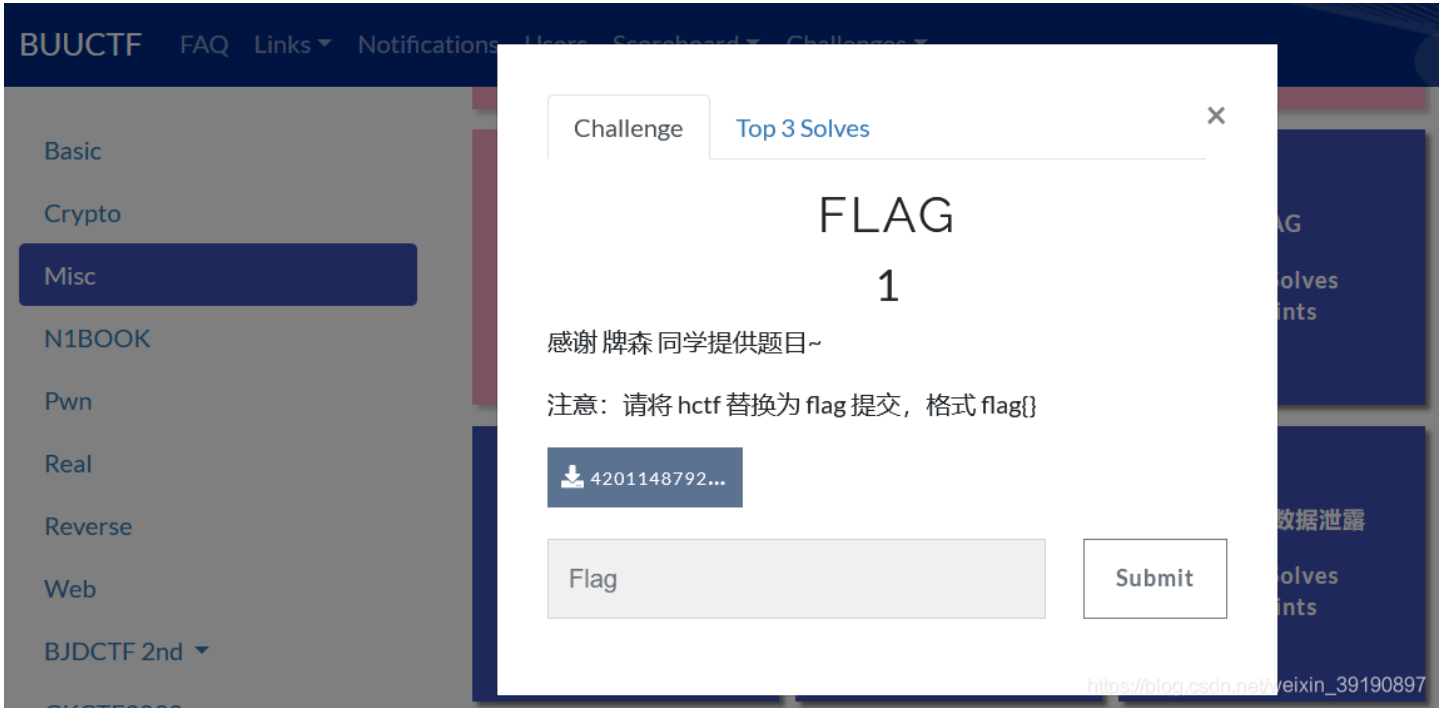
2A 2A 2A 2A 2A 2A 2A 2A | *****
2A 2A 0D 0A 66 6C 61 67 | ***** flag
28 4D 7A 63 33 59 32 4A | :base64:(Mzc3Y2J
6A 59 54 4A 6D 4D 6D 59 | hZGRhMWVjYTJmMmY
33 4E 7A 63 34 4D 57 59 | 3M2QzNjI3Nzc4MWY
2A 2A 2A 2A 2A 2A 2A 2A | wMGE=) *****
2A 2A 2A 2A 2A 2A 2A 2A | *****
2A 2A 2A 2A 2A 2A 2A 2A | *****
2A 2A 2A 2A 2A 2A 2A 2A | *****
2A 0D 0A DD D9 D0 C5 FE | ***** YÜÐÀp
  
```

3、base64 解码获得 flag：

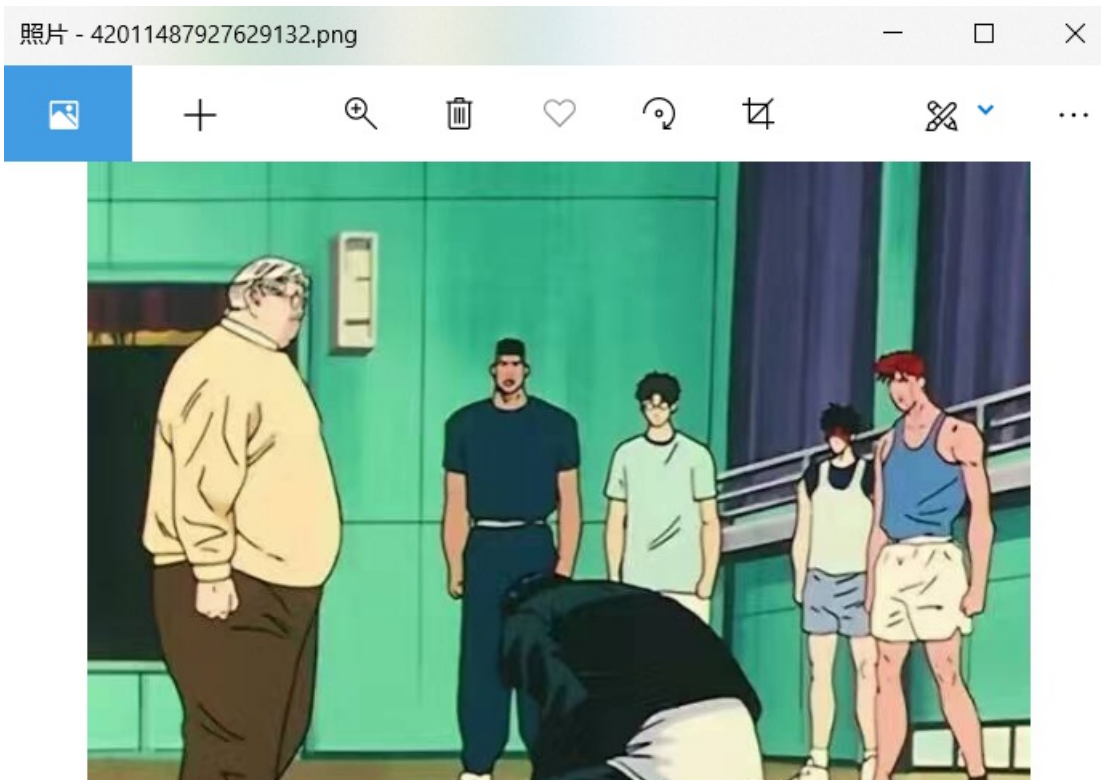




## No.12 ZIP的16进制文件头



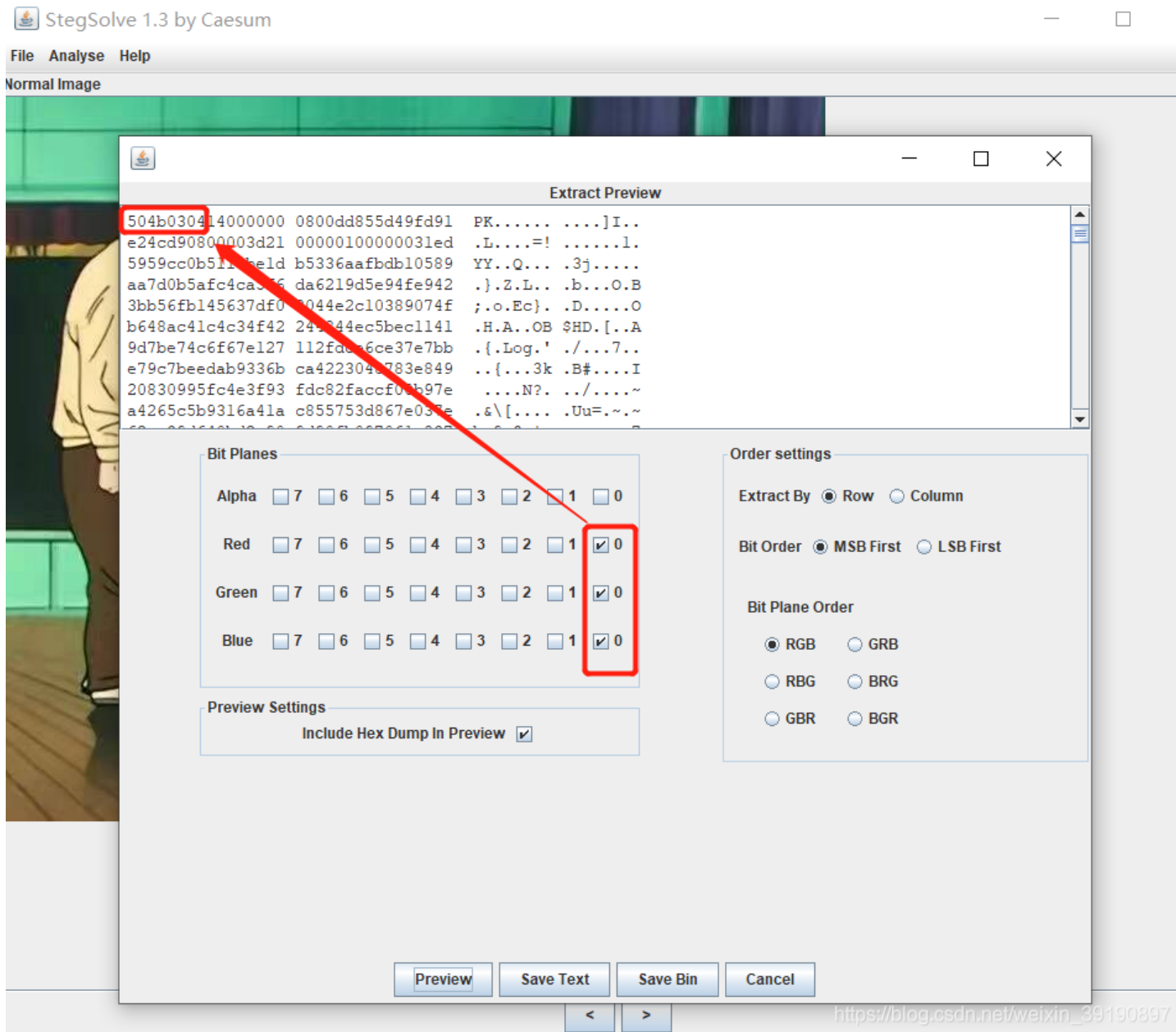
下载后是一张图片:





[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

1、使用 stegsolve 打开图片，发现在 RGB 模式下 0 通道预览是个 ZIP 文件（文件头 `504b0304`），存在 LSB 隐写，Save Bin 保存为 flag.zip:



[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

**【注意】**一定要对常见的文件头高度敏感:

JPEG (jpg),	文件头: FFD8FF	文件尾: FF D9
PNG (png),	文件头: 89504E47	文件尾: AE 42 60 82
GIF (gif),	文件头: 47494638	文件尾: 00 3B
Archive (zip),	文件头: 504B0304	文件尾: 50 4B
TIFF (tif),	文件头: 49492A00	文件尾:
Windows Bitmap (bmp),	文件头: 424D	文件尾:



CAD (dwg),	文件头: 41433130	文件尾:
Adobe Photoshop (psd),	文件头: 38425053	文件尾:
Rich Text Format (rtf),	文件头: 7B5C727466	文件尾:
XML (xml),	文件头: 3C3F786D6C	文件尾:
HTML (html),	文件头: 68746D6C3E	
Email [thorough only] (eml),	文件头: 44656C69766572792D646174653A	
Outlook Express (dbx),	文件头: CFAD12FEC5FD746F	
Outlook (pst),	文件头: 2142444E	
MS Word/Excel (xls.or.doc),	文件头: D0CF11E0	
MS Access (mdb),	文件头: 5374616E64617264204A	
WordPerfect (wpd),	文件头: FF575043	
Adobe Acrobat (pdf),	文件头: 255044462D312E	
Quicken (qdf),	文件头: AC9EBD8F	
Windows Password (pwl),	文件头: E3828596	
RAR Archive (rar),	文件头: 52 61 72 21 1A 07 00	文件尾: 3D 7B 00 40 07 00
Wave (wav),	文件头: 57415645	
AVI (avi),	文件头: 41564920	
Real Audio (ram),	文件头: 2E7261FD	
Real Media (rm),	文件头: 2E524D46	
MPEG (mpg),	文件头: 000001BA	
MPEG (mpg),	文件头: 000001B3	
Quicktime (mov),	文件头: 6D6F6F76	
Windows Media (asf),	文件头: 3026B2758E66CF11	
MIDI (mid),	文件头: 4D546864	

[https://blog.csdn.net/weixin\\_39190892](https://blog.csdn.net/weixin_39190892)

2、360解压缩导出来的 flag.zip 文件，获得一个没有后缀的文件：

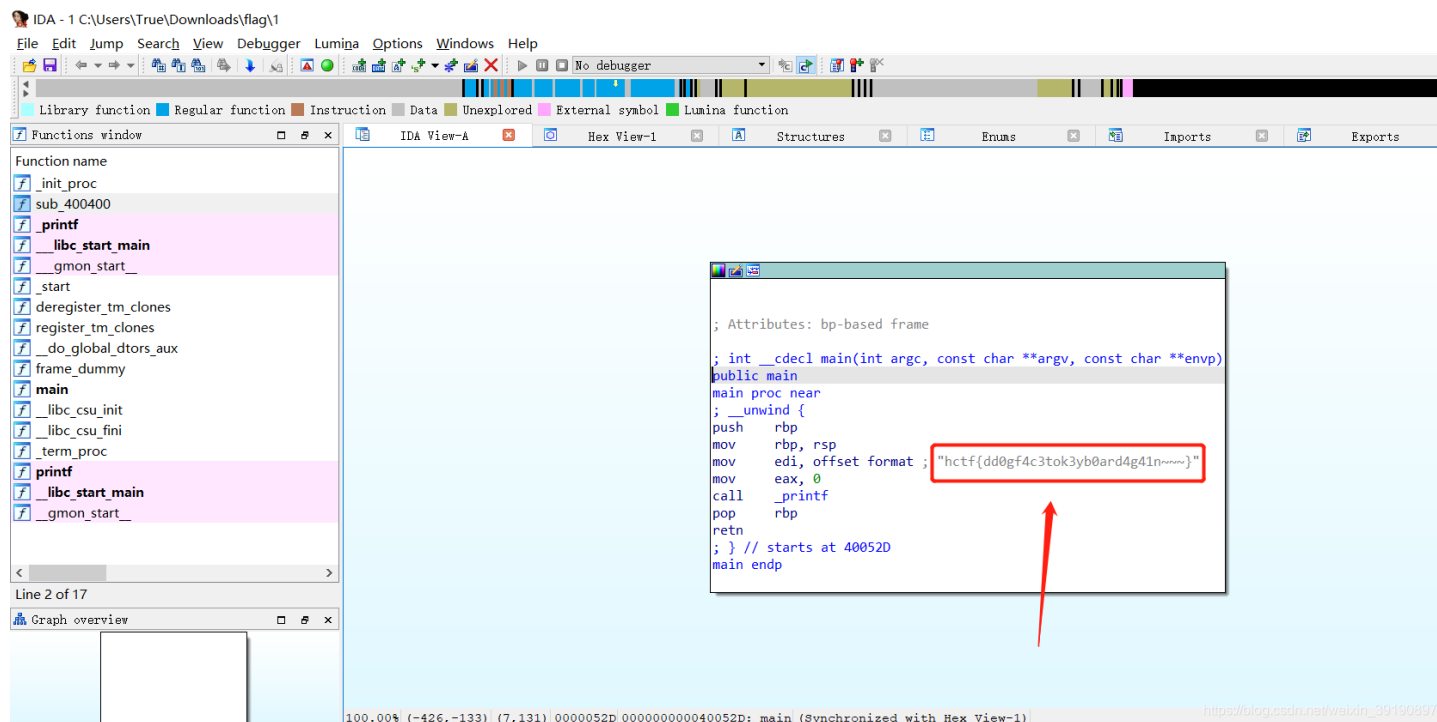


3、编辑器或者 Winhex 打开发现是 ELF 文件：

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
00000000	7F	45	4C	46	02	01	01	00	00	00	00	00	00	00	00	00	ELF	
00000010	02	00	3E	00	01	00	00	00	40	04	40	00	00	00	00	00	>	@ @
00000020	40	00	00	00	00	00	00	00	70	11	00	00	00	00	00	00	@	p @
00000030	00	00	00	00	40	00	38	00	09	00	40	00	1E	00	1B	00	@ @	@ @
00000040	06	00	00	00	05	00	00	00	40	00	00	00	00	00	00	00	@	@
00000050	40	00	40	00	00	00	00	00	40	00	40	00	00	00	00	00	@ @	@ @
00000060	F8	01	00	00	00	00	00	00	F8	01	00	00	00	00	00	00	ø	ø
00000070	08	00	00	00	00	00	00	00	03	00	00	00	04	00	00	00	ø	ø @
00000080	38	02	00	00	00	00	00	00	38	02	40	00	00	00	00	00	ø	ø @
00000090	38	02	40	00	00	00	00	00	1C	00	00	00	00	00	00	00	ø @	
000000A0	1C	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00		
000000B0	01	00	00	00	05	00	00	00	00	00	00	00	00	00	00	00		
000000C0	00	00	40	00	00	00	00	00	00	00	40	00	00	00	00	00	@	@
000000D0	24	07	00	00	00	00	00	00	24	07	00	00	00	00	00	00	\$	\$
000000E0	00	00	20	00	00	00	00	00	01	00	00	00	06	00	00	00		
000000F0	10	0E	00	00	00	00	00	00	10	0E	60	00	00	00	00	00		
00000100	10	0E	60	00	00	00	00	00	30	02	00	00	00	00	00	00	,	0
00000110	38	02	00	00	00	00	00	00	00	00	20	00	00	00	00	00	ø	
00000120	02	00	00	00	06	00	00	00	28	0E	00	00	00	00	00	00	(	,
00000130	28	0E	60	00	00	00	00	00	28	0E	60	00	00	00	00	00	(	,

[https://blog.csdn.net/weixin\\_39190892](https://blog.csdn.net/weixin_39190892)

#### 4、使用 IDA 打开可以获得 Flag:



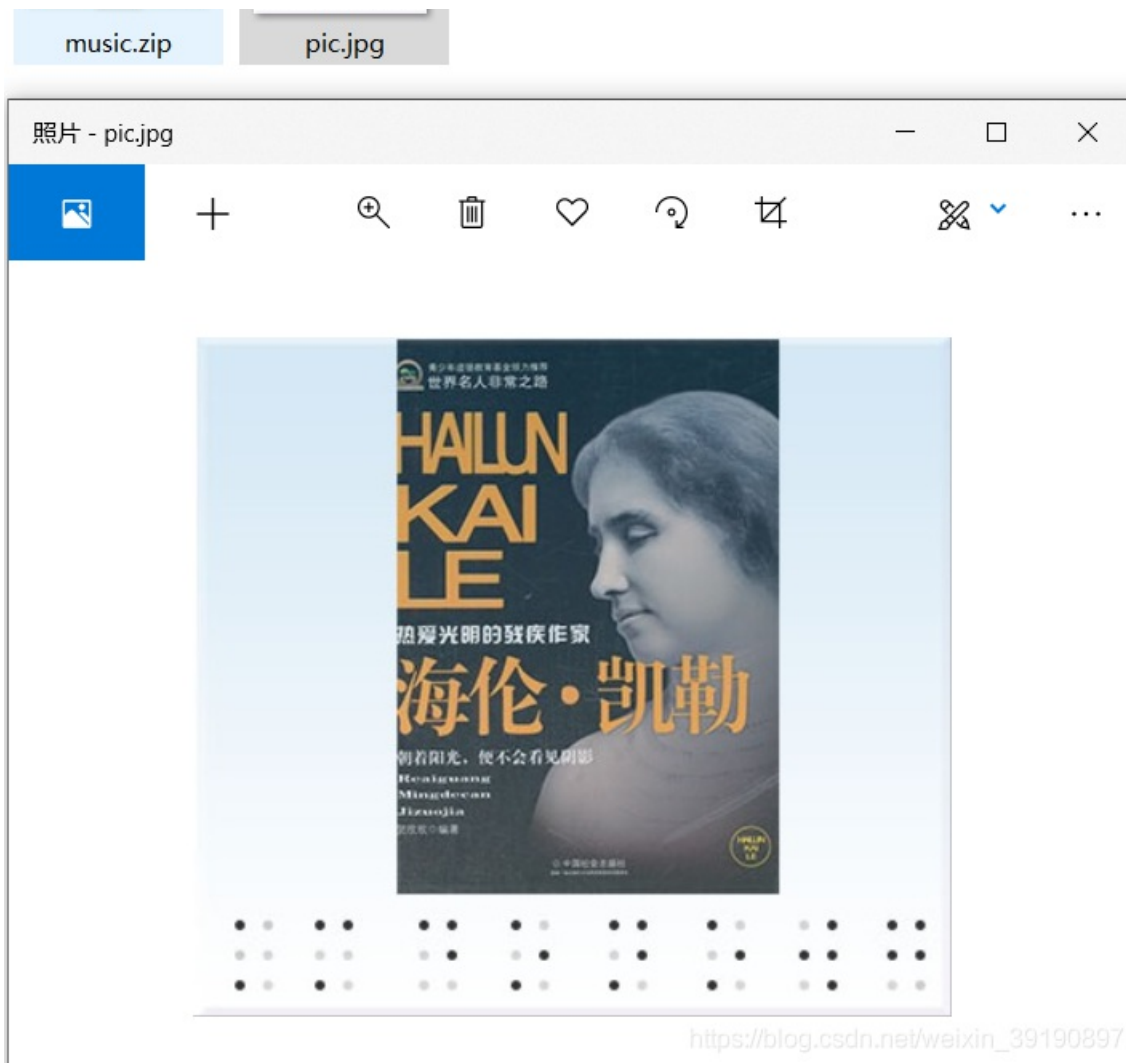
## No.13 盲文与摩斯密码读取



1、下载后是一张图片和压缩文件:

d46-5ada-4821-8664-d7bc26be142a > 假如给我三天光明





[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

2、对照盲文解密图片中隐藏的字符串 `kmdonowg` :

a/1 b/2 c/3 d/4 e/5 f/6 g/7 h/8 i/9 j/0

●○●○●●●●●○●●●●●○●○●○●

○○●○●○○○●○●●●●●●●●●●●●

○○○○○○○○○○○○○○○○○○○○○○○○

k l m n o p q r s t

●○●○●●●●●○●○●●●●●○●○●○●

○○●○●○○○●○●●●●●●●●●●●●

●○●○●○●○●○●○●○●○●○●○●○●○

u v w x y z

●○●○●○●○●○●○●○●○●○●○●○●○

○○●○●●●○○○●○●○●○●○●○●○

●●●●●○●●●●●●●●●●●●●●●●●●

① ④

② ⑤

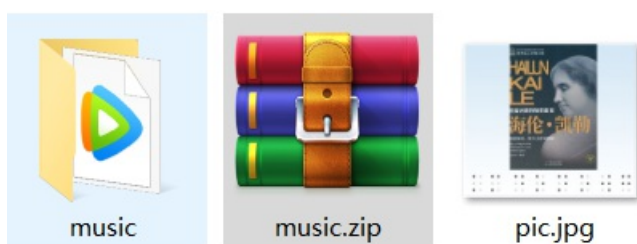
③ ⑥

盲文的六点排列图

盲文的数字和英文字母图

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

3、解压缩 music.zip 文件发现需要密码，输入上面解密的字符串正好可以解开：



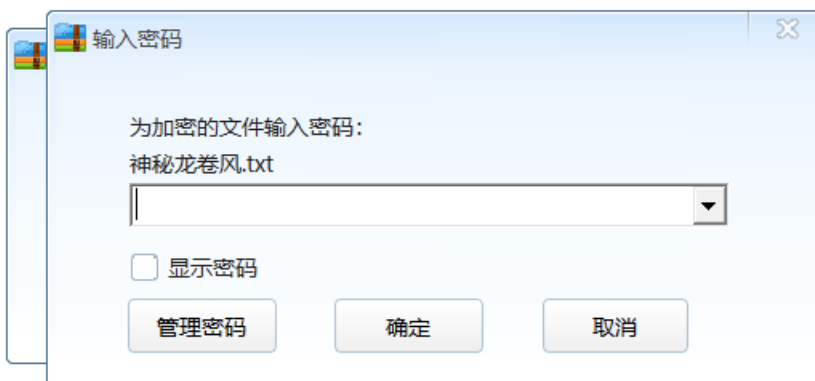






1、下载后解压需要密码：

神秘龙卷风	2021/5/7 23:41	文件夹
神秘龙卷风.rar	2015/12/8 20:57	RAR 文件



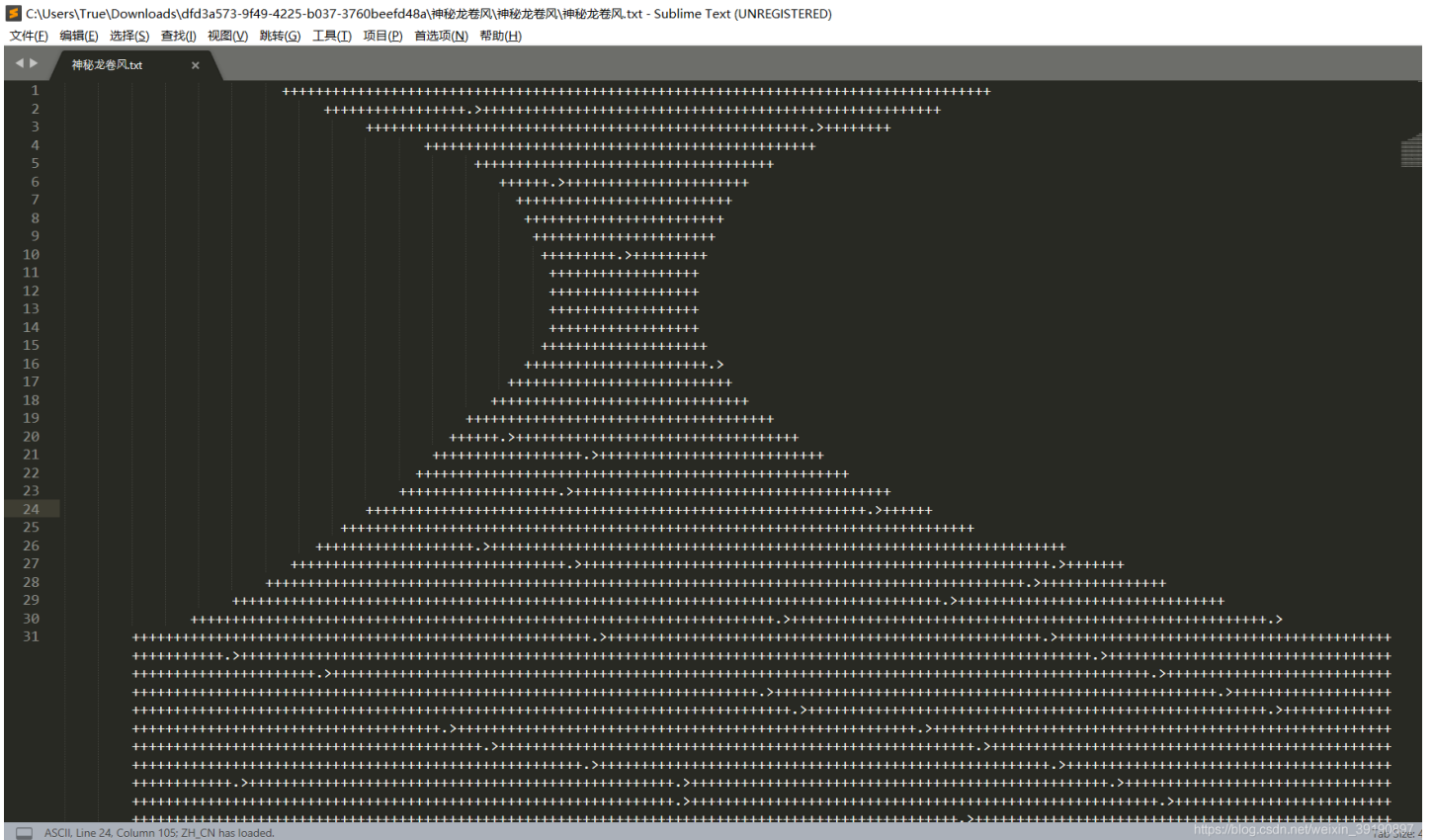
[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

2、尝试使用 ARCHPR 进行四位数字的爆破：

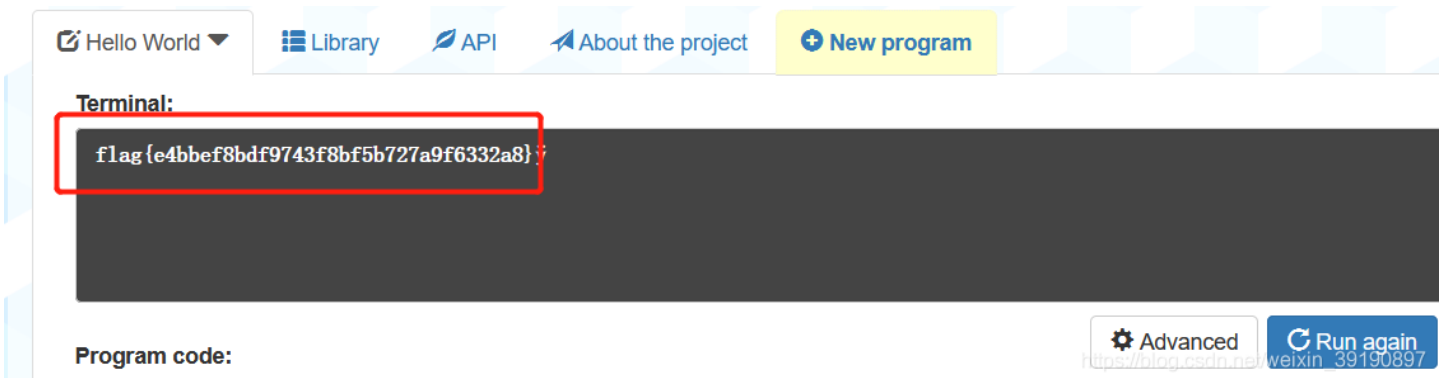




### 3、解压后打开文件：



### 4、发现是 brainfuck 代码，使用 在线执行网站 运行即可得到 flag：



科普下 brainfuck 代码：

Müller的目标是建立一种简单的、可以用最小的编译器来实现的、符合图灵完全思想的编程语言。这种语言由八种状态构成，为Amiga机器编写的编译器（第二版）只有240个字节大小！

就象它的名字所暗示的，brainfuck程序很难读懂。尽管如此，brainfuck图灵机一样可以完成任何计算任务。虽然brainfuck的计算方式如此与众不同，但它确实能够正确运行。

这种语言基于一个简单的机器模型，除了指令，这个机器还包括：一个以字节为单位、被初始化为零的数组、一个指向该数组的指针（初始时指向数组的第一个字节）、以及用于输入输出的两个字节流。

这种语言，是一种按照“Turing complete（图灵完备）”思想设计的语言，它的主要设计思路是：用最小的概念实现一种“简单”的语言，BrainF\*\*k语言只有八种符号，所有的操作都由这八种符号的组合来完成。

## 字符标识

 编辑

下面是这八种状态的描述，其中每个状态由一个字符标识：

字符	含义
>	指针加一
<	指针减一
+	指针指向的字节的值加一
-	指针指向的字节的值减一
.	输出指针指向的单元内容（ASC II 码）
,	输入内容到指针指向的单元（ASC II 码）
[	如果指针指向的单元值为零，向后跳转到对应的]指令的次一指令处
]	如果指针指向的单元值不为零，向前跳转到对应的[指令的次一指令处

（按照更节省时间的简单说法，“]”也可以说成“向后跳转到对应的[”状态”。这两解释是一样的。[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

## No.15 后门查杀之后门识别

BUUCTF    FAQ    Links    Notifications    Home    Searchboard    Challenges

- Basic
- Crypto
- Misc**
- N1BOOK
- Pwn
- Real
- Reverse
- Web
- BJDCTF 2nd
- GKCTF2020
- V&N2020 公开赛
- VNCTF2021

Challenge    **Top 3 Solves**    ×

## 后门查杀

### 1

小白的网站被小黑攻击了，并且上传了Webshell，你能帮小白找到这个后门么？(Webshell中的密码(md5)即为答案)。注意：得到的 flag 请包上 flag{} 提交

↓ 10b1cf9b-cf...

Flag    Submit

- 数据包中的线索  
1750 Solves
- 九连环  
1542 Solves
- 面具下的flag  
1500 Solves  
[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

1、下载后解压缩如下：

名称	修改日期	类型	大小
admin	2015/7/9 17:05	文件夹	
cache	2015/7/9 17:05	文件夹	
data	2015/7/9 17:05	文件夹	
images	2015/7/9 17:05	文件夹	
include	2015/7/9 17:07	文件夹	
install	2015/7/9 17:05	文件夹	
languages	2015/7/9 17:05	文件夹	
theme	2015/7/9 17:05	文件夹	
upload	2015/7/9 17:05	文件夹	
.htaccess	2013/9/4 21:15	HTACCESS 文件	2 KB
article.php	2013/9/5 4:48	PHP 文件	2 KB
article_category.php	2013/9/1 6:16	PHP 文件	3 KB
attacktest.sql	2013/9/5 6:49	SQL 文件	121 KB
captcha.php	2013/8/29 2:26	PHP 文件	2 KB
download.php	2013/9/5 3:31	PHP 文件	1 KB
error.php	2013/9/5 4:58	PHP 文件	1 KB
favicon.ico	2013/8/31 0:23	图标	2 KB
index.php	2013/9/1 6:23	PHP 文件	3 KB
page.php	2013/9/1 6:12	PHP 文件	2 KB
phpinfo.php	2013/9/5 1:32	PHP 文件	1 KB
product.php	2013/9/1 6:12	PHP 文件	2 KB
product_category.php	2013/9/1 6:16	PHP 文件	3 KB
robots.txt	2013/8/24 8:50	文本文档	1 KB
sitemap.php	2013/6/30 10:39	PHP 文件	5 KB
web.php	2013/9/5 1:31	PHP 文件	1 KB

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

2、找 Webshell，直接上火绒扫描该文件夹：

**火绒安全** 病毒查杀

共发现风险项目1个，建议立即处理 全部忽略 立即处理

扫描已完成

风险项目	状态
<input checked="" type="checkbox"/> C:\Users\True\Downloads\10b1cf9b-cfb1-40f2-8340-c1bf78b37c9d\html\include\include.php <b>后门病毒</b> Backdoor/PHP.WebShell.h	待处理 <a href="#">详情</a>

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

3、打开该文件，发现 flag: `flag{6ac45fb83b3bc355c024f5034b947dd3}` :

```
include.php x
1 <?php
2 //ini_set('display_errors',1);
3 @error_reporting(7);
4 @session_start();
5 @set_time_limit(0);
6 @set_magic_quotes_runtime(0);
7 if( strpos( strtolower( $_SERVER['HTTP_USER_AGENT'] ), 'bot' ) !== false ) {
8     header('HTTP/1.0 404 Not Found');
9     exit;
10 }
11 ob_start();
12 $mtime = explode(' ', microtime());
13 $starttime = $mtime[1] + $mtime[0];
14 define('SA_ROOT', str_replace('\\', '/', dirname(__FILE__)).'/');
15 define('SELF', $_SERVER['PHP_SELF'] ? $_SERVER['PHP_SELF'] : $_SERVER['SCRIPT_NAME']);
16 define('IS_WIN', DIRECTORY_SEPARATOR == '\\');
17 define('IS_GPC', get_magic_quotes_gpc());
18 $dis_func = get_cfg_var('disable_functions');
19 define('IS_PHPINFO', (!ereg("phpinfo",$dis_func)) ? 1 : 0);
20
21 if( IS_GPC ) {
22     $_POST = s_array($_POST);
23 }
24 $P = $_POST;
25 unset($_POST);
26 /*===== 程序配置 =====*
27
28 //echo encode_pass('angel');exit;
29 // 如果需要密码验证,请修改登陆密码,留空为不需要验证
30 $pass = '6ac45fb83b3bc355c024f5034b947dd3'; //angel
31
32 //如您对 cookie 作用范围有特殊要求,或登录不正常,请修改下面变量,否则请保持默认
33 // cookie 前缀
34 $cookiepre = '';
35 // cookie 作用域
```

https://blog.csdn.net/weixin\_39190897

## No.16 路由器信息数据查看

BUUCTF FAQ Links Notifications Users Searchboard Challenges

Basic  
Crypto  
Misc  
N1BOOK  
Pwn  
Real  
Reverse  
Web  
BJDCTF 2nd  
GKCTF2020

Challenge Top 3 Solves

**荷兰宽带数据泄露**

1

注意: 得到的 flag 请包上 flag{} 提交

200fd993-69...

Flag Submit

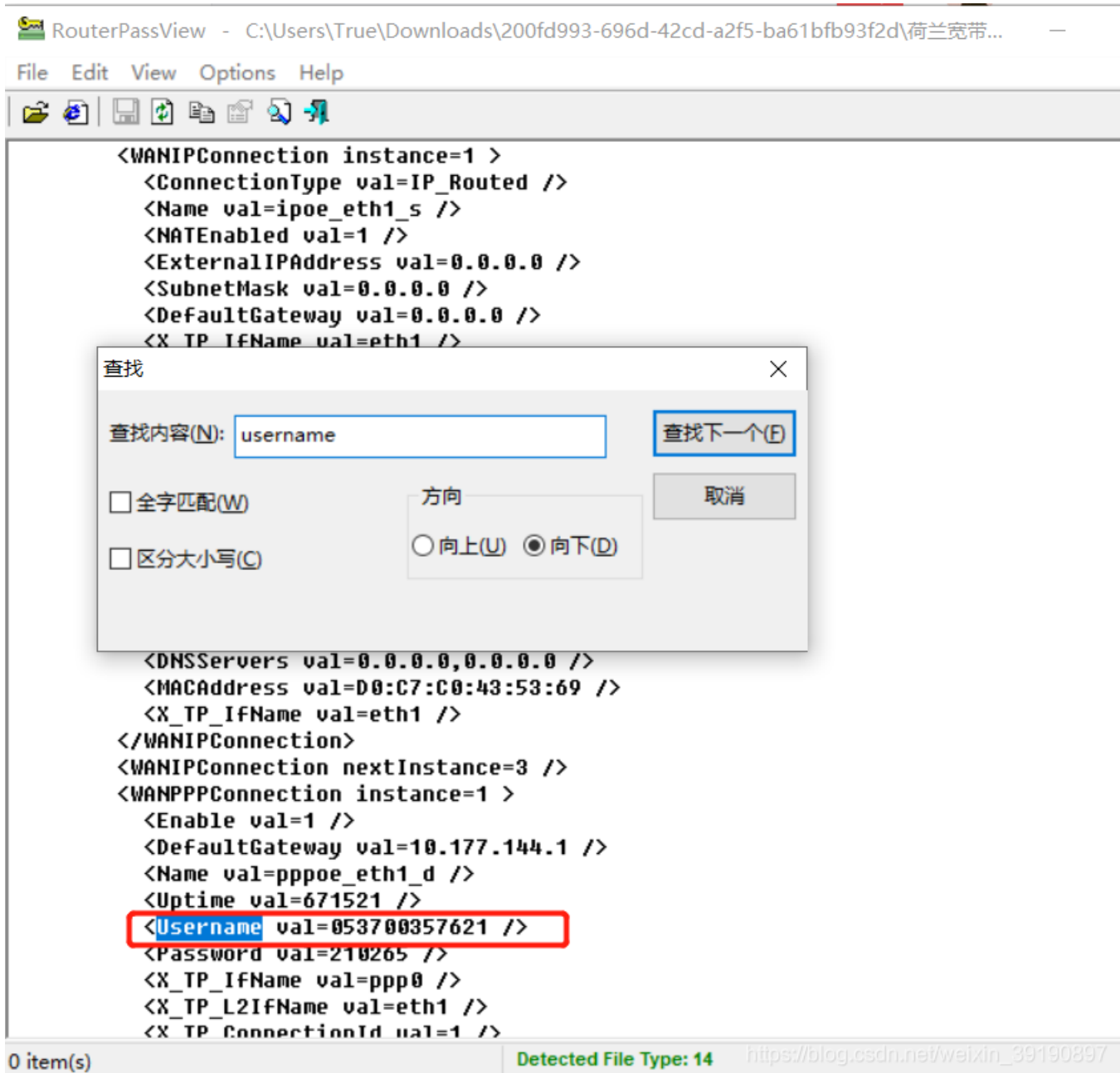
1 Points 1 Points 1 Points

https://blog.csdn.net/weixin\_39190897

1、下载后解压缩是个 conf.bin 文件，路由器信息数据，一般包含账号密码，题目并没有提示 flag 是什么，猜测是账号或者密码加上格式为最终 flag:



2、无法使用编辑器直接打开阅读该类文件，需要用 RouterPassView 工具查看，打开后搜索 username 或者 password:



最后发现是用户名为 flag，加上格式提交即可 `flag{053700357621}`。

## No.17 base64流量导出图片



BUUCTF FAQ Links Notifications Users Searchboard Challenges

Basic  
Crypto  
Misc  
N1BOOK  
Pwn  
Real  
Reverse  
Web  
BJDCTF 2nd  
GKCTF2020  
V&N2020 公开赛  
VNCTF2021

Challenge Top 3 Solves ✕

## 数据包中的线索

### 1

公安机关近期截获到某网络犯罪团伙在线交流的数据包，但无法分析出具体的交流内容，聪明的你能帮公安机关找到线索吗？注意：得到的 flag 请包上 flag{} 提交

[📄 94b9c18e-d5...](#)

数据包中的线索

1750 Solves  
1 Points

九连环

1543 Solves  
1 Points

面具下的flag

1509 Solves  
1 Points

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

1、下载后用 wireshark 打开，大部分都是 TCP 的包，直接过滤出 http 的包：

流量中的线索.pcapng

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(T) 无线(W) 工具(I) 帮助(H)

No.	Time	Source	Destination	Protocol	Length	Data	Data	Info
7	0.420363	172.16.66.100	172.16.80.5	HTTP	129			GET /sdk/vimService?wsdl HTTP/1.1
8	0.420766	172.16.80.5	172.16.66.100	HTTP	296			HTTP/1.1 301 Moved Permanently (text/html)
60	10.082308	172.16.66.100	172.16.80.120	HTTP	439			GET /fenxi.php HTTP/1.1
142	10.091579	172.16.80.120	172.16.66.100	HTTP	444	2f396a2f34414...	2f396a2f34414...	HTTP/1.1 200 OK (text/html)

<

> Frame 7: 129 bytes on wire (1032 bits), 129 bytes captured (1032 bits) on interface 0  
 > Ethernet II, Src: CompalIn\_32:73:c5 (20:89:84:32:73:c5), Dst: Hangzhou\_9e:cc:85 (0c:da:41:9e:cc:85)  
 > Internet Protocol Version 4, Src: 172.16.66.100, Dst: 172.16.80.5  
 > Transmission Control Protocol, Src Port: 1882, Dst Port: 80, Seq: 1, Ack: 1, Len: 75  
 > Hypertext Transfer Protocol  
 > GET /sdk/vimService?wsdl HTTP/1.1\r\n  
 Host: 172.16.80.5\r\n  
 Connection: Close\r\n  
 \r\n  
 [Full request URI: http://172.16.80.5/sdk/vimService?wsdl]  
 [HTTP request 1/1]  
 [Response in frame: 8]

```

0000 0c da 41 9e cc 85 20 89 84 32 73 c5 08 00 45 00  ..A... 2s...E.
0010 00 73 45 bd 40 00 40 06 00 00 ac 10 42 64 ac 10  -SE@@...Bd..
0020 50 05 07 5a 00 50 b6 2f 38 32 92 ff 4a f1 50 18  P-Z·P·/ 82·J·P·
0030 40 29 ea ef 00 00 47 45 54 20 2f 73 64 6b 2f 76  @)....GE T /sdk/v
0040 69 6d 53 65 72 76 69 63 65 3f 77 73 64 6c 20 48  imServic e?wsdl H
0050 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 31  TTP/1.1 ·Host: 1
0060 37 32 2e 31 36 2e 38 30 2e 35 0d 0a 43 6f 6e 6e  72.16.80 .5·Conn
0070 65 63 74 69 6f 6e 3a 20 43 6c 6f 73 65 0d 0a 0d  ection: Close...
0080 0a
  
```

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

2、追踪 http 流：

Wireshark · 追踪 HTTP 流 (tcp.stream eq 7) · 流量中的线索.pcapng

GET /fenxi.php HTTP/1.1

```
GET /tenx1.php HTTP/1.1
Host: 172.16.80.120
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
```

```
HTTP/1.1 200 OK
Date: Tue, 18 Aug 2015 09:09:39 GMT
Server: Apache
X-Powered-By: PHP/5.3.3
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
```

```
/9j/4AAQSkZJRgABAQEAYABgAAD/2wBDAAIBAQIBAQICAgICAgICAwUDAwMDAwYEBAMFBwYHBwCG
BwcICQsJCAgKCACg8CKGgsMDAwMBwKODwMDgsMDAz/2wBDAQICAgMDAwYDAwYMCAcIDAwMDAwM
DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAz/wAARCAHgAkQDASIA
AhEBAXEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3
ODk6Q0RFRkdISUpTVFVWV1hZmNkZWZnaGlqc3R1dnd4eXQhIWhGh4iJipKTlJWWl5iZmqKjpKWm
p6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erX8vP09fb3+Pn6/8QAHwEAA
wEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSEx
BhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElK
U1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsr00tba3
uLm6wsPExcbhYmNk0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD9/KKK
KACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoooo
AKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigA
ooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAC
iiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKK
KACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAoooo
AKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigA
ooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAC
iiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKK
KACiiigAooooAKKKKACiiigDjfix+0V8Pvgvr2j6V4x8deDfCeqeIRM2lWes61bWfXqYhUNMYEld
wL8tSC2wHaCCCV0nhrxNpvjTw7Yaxo+oW0raTqlvHd2V7ZrPb3kmihk1jkULXR1IIZSQQQa/N/
```

分组 00。1 客户端 分组, 1 服务器 分组, 1 turn(s)。点击选择。

整个对话 (86 kB)

3、传输内容好像是 base64 密文，使用 [在线网站](#) 解码 base64:

## Base64 在线解码、编码



常规Base64 CSS Base64 DES加密/解密 3DES加密/解密 AES加密/解密 RSA加密/解密

```
n+00x+S1bDkV0vXwht0uizKdJiK00g/wComGc/r/rK9Brai+AjorjpuuENLrVt+z000jml  
4X/dzO7JHXjml2D2sczt9+4nd6v3ms6lLb21t9pnkto7pJ3R3/ufPXL/AGFD6pKHJ70gr5hiZ5jS  
rRre7D3f8X83/gR798BfBen/ALowYuNW1iJkv7iEXV8+z95/sQp/n7710c/xwm8PwXVxrXh/VNM  
WOFJrYnZN9q3vsSH5P8AltV2fjXz94o+MPibxTps1jdX0clvI8Lx74/40ff/AOyVm+NvjB4k1zTp  
JtW1aNlrR4blERP3cDo+9K6o0atKn7DDfZPm62Bjiq08Tjp+9Kb5r83923KfQmvtHW/guG7/tzT  
W0y+tLX+0LW2e5R/tn+wmz7j769i/Z7+M998JfBmgeArLwF4gm1WbS5NdjE1zbL9tM03mXM0vz/u  
S88zH5+7143+yj+wlrHxx8GyeN/FI7DaS69HB/ZcTjz2Sy89Hmhf/bng3oif8s/Mevr7UvgVeS+P  
ff3iC11WC3u9a0KDRNMYw/8AIPKGZmdv7+WdP++K/UeE8lzXDw9v8HP/AIf73/2v3n4/xZmWRzqe  
wjyz5d/i5fs7a/3pc3+HTodN8E/ivbGL4XaP4mt7Wayh1iHzlhm4ePBK4P/AHzRVj4SfDyH4VfD  
PQ/DsMnmR6NZRWiv/f2IFz+OKK+/p/2lyLn3tr6n5djpq7rz9gvcu+X0vp+B/9k=
```

编码源格式:  文本  Hex 解码结果: 自动检测 中文编码: UTF-8 编码 解码

该内容已经被插件识别为二进制数据。  
但未提供可供阅读的文本信息，且数据量较大，故不在此处显示hex内容。  
如需查看hex内容，请关闭自动模式!

插件【Jpeg】Jpeg Image(JFIF)  
另存为: jpg文件  
附加信息:  
Size: 580x480  
format: JFIF  
显示内容非原始信息  
数据长度: 63,089 Bytes  
插件数: 18, 耗时: 0ms  
[https://blog.csdn.net/weixin\\_39190497](https://blog.csdn.net/weixin_39190497)

4、发现解码出来的内容是 jpg 图片格式并以自动转为图片，下载即可得到 flag:

照片 - from\_the-x.jpg

flag{209acebf6324a09671abc31c869de72c}

[https://blog.csdn.net/weixin\\_39190497](https://blog.csdn.net/weixin_39190497)

## No.18 脑洞大开歌名猜密钥

BUUCTF FAQ Links Notifications Users Scoreboard Challenges

Challenge Top 3 Solves

# snake

1

注意: 得到的 flag 请包上 flag{} 提交

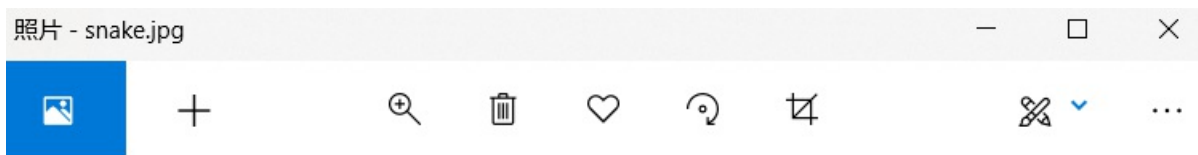
8033a99b-d...

Flag Submit

1071 Solves 1 Points 1022 Solves 1 Points 1016 Solves 1 Points

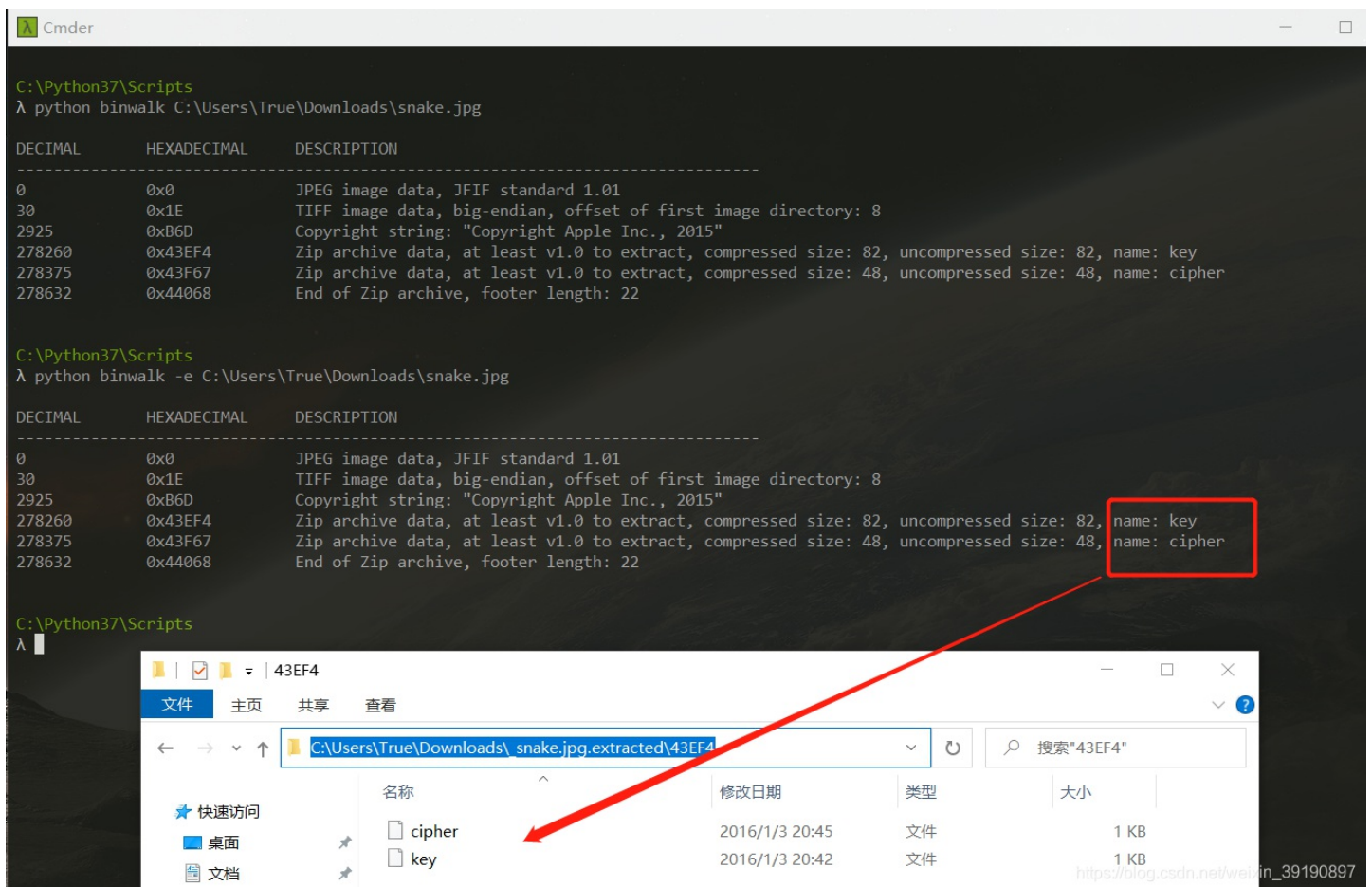
[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

下载后是一张蛇的照片:

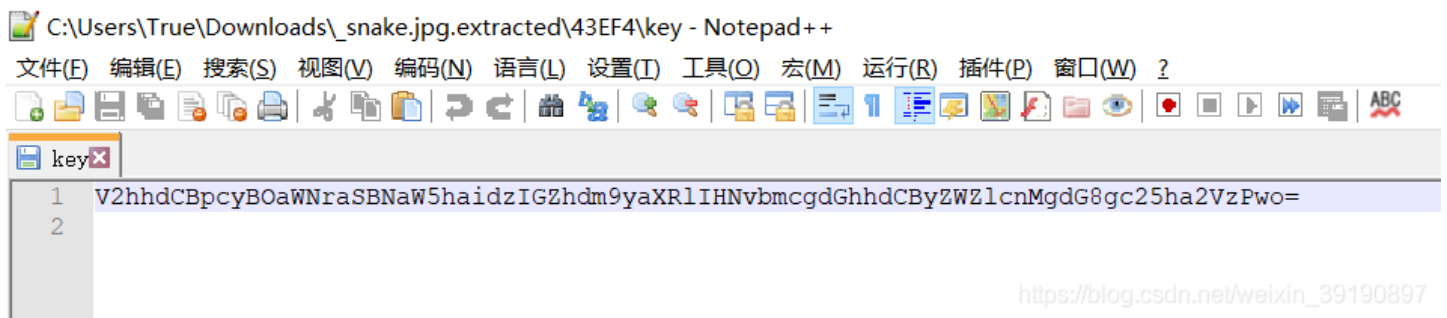




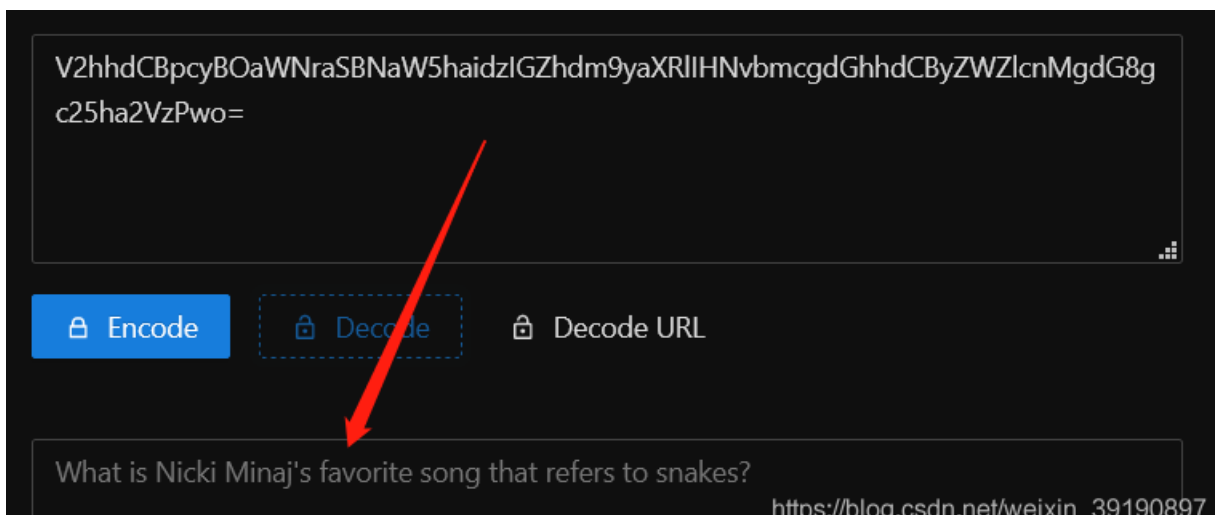
1、使用 Binwalk 分离提取出两个隐藏文件：



2、打开 Key 文件发现 Base64 编码：



3、解码如下：



What is Nicki Minaj's favorite song that refers to snakes?

跟蛇有关，于是我们百度 Nicki Minaj 然后有她的百度百科，观看一下，发现：



nicki minaj

进入词条

## 基本信息

中文名	妮琪·米娜	身高	157 cm
外文名	Nicki Minaj	毕业院校	纽约拉瓜迪亚高中表演艺术学校 <sup>[1]</sup>
别名	Onika Tanya Maraj（原名）	职业	歌手、作词人、作曲人、演员
国籍	美国	经纪公司	Young Money、Cash Money、环球音乐
民族	特立尼达和多巴哥	代表作品	Super Bass, Anaconda, Starships, Pound The Alarm、The Night Is Still Young、Moment 4 Life
出生地	特立尼达和多巴哥西班牙港	主要成就	全美音乐奖2012年度最受欢迎饶舌嘻哈歌手奖
出生日期	1982年12月8日	三围	89/66/114 cm
星座	射手座	音乐类型	Hip-hop、R&B、dance-pop
血型	A型	粉丝名称	Barbz, Kenz, Dante

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

## anaconda - 百度翻译

### anaconda

英[.ænə'kɒndə] 美[.ænə'kɑːndə]

n. 水蚺(南美洲蟒蛇);

[例句] Operation **Anaconda**'s now in its second week.

蟒蛇行动现在已进行到第二周。

[其他] 复数: anacondas

[进行更多翻译](#)

扫码下载百度翻译APP

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

Anaconda 是关于蛇的作品，这是不是我们的 key 呢？



于是我们用 Serpent 工具解密，Serpent 是一个加密算法（为何猜测是这种加密算法.....不知道.....），<http://serpent.online-domain-tools.com/>，我们用这个然后key是 Anaconda:

Input type: File

File: C:\fakepath\cipher Browse

Function: SERPENT

Mode: ECB (electronic codebook)

Key: anaconda  
(plain)

Plaintext  Hex

> Encrypt! > Decrypt! ▶ 🔗

0%  
File was uploaded.

Decrypted text:

00000000	43 54 46 7b 77 68 6f 5f 6b 6e 65 77 5f 73 65 72	CTF{who_knew_serpent_cipher_existed}.....
00000010	70 65 6e 74 5f 63 69 70 68 65 72 5f 65 78 69 73	
00000020	74 65 64 7d 00 00 00 00 00 00 00 00 00 00 00 00	

[\[Download as a binary file\] \[?\]](#) Active [https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

获得 flag: `flag{who_knew_serpent_cipher_existed}`。

## No.19 Steghide隐写工具的使用

BUUCTF [FAQ](#) [Links](#) [Notifications](#) [Users](#) [Searchbox](#) [Challenges](#)

Basic  
Crypto  
Misc  
N1BOOK  
Pwn  
Real  
Reverse  
Web  
BJDCTF 2nd

Challenge Top 3 Solves

# 九连环

1

注意：得到的 flag 请包上 flag{} 提交

[📄 389a0c11-d0...](#)

Flag Submit

下载后是一张图片：



1、Binwalk 查看发先隐藏了一个 ZIP 文件，进行提取：

```

C:\Python37\Scripts
λ python binwalk C:\Users\True\Downloads\389a0c11-d0df-4180-829a-b529e6b0a1bc\123456cry.jpg

DECIMAL      HEXADECIMAL    DESCRIPTION
-----
0             0x0            JPEG image data, JFIF standard 1.01
19560        0x4C68         Zip archive data, at least v1.0 to extract, name: asd/
48454        0xBD46         Zip archive data, at least v1.0 to extract, compressed size: 184, uncompressed size: 184, name: asd/qwe.zip
48657        0xBE11         End of Zip archive, footer length: 22
48962        0xBF42         End of Zip archive, footer length: 22

C:\Python37\Scripts
λ python binwalk -e C:\Users\True\Downloads\389a0c11-d0df-4180-829a-b529e6b0a1bc\123456cry.jpg

DECIMAL      HEXADECIMAL    DESCRIPTION
-----
0             0x0            JPEG image data, JFIF standard 1.01
19560        0x4C68         Zip archive data, at least v1.0 to extract, name: asd/
48454        0xBD46         Zip archive data, at least v1.0 to extract, compressed size: 184, uncompressed size: 184, name: asd/qwe.zip
48657        0xBE11         End of Zip archive, footer length: 22
48962        0xBF42         End of Zip archive, footer length: 22

C:\Python37\Scripts
λ

```

2、提取后发现两个文件：

« `_123456cry.jpg.extracted` » asd



good-已合并.jpg

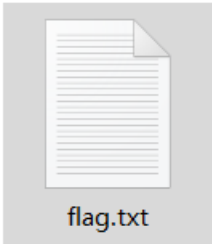


qwe.zip

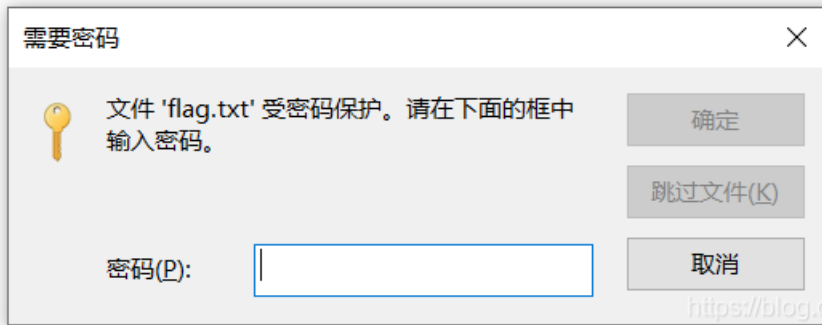
[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

3、直接打开 qwe.zip 里面的文件发现需要密码，那大概率就是回到另一个图片 `good-已合并.jpg` 进行信息提取了：

123456cry.jpg.extracted > asd > qwe.zip



flag.txt



[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

4、使用 Binwalk 查看 `good-已合并.jpg` 是否由多个文件组合而成，无果：

```

C:\Python37\Scripts
λ python binwalk C:\Users\True\Downloads\389a0c11-d0df-4180-829a-b529e6b0a1bc\_123456cry.jpg.extracted\asd\good-已合并.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0         JPEG image data, JFIF standard 1.01

C:\Python37\Scripts
λ

```

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

5、此时只能用尝试用隐写工具 **Steghide**（[官方下载地址](#)）进行信息提取。这是一个可以将文件隐藏到图片或音频中的工具，其使用如下：

目的	命令
Linux安装	apt-get install steghide
隐藏文件	steghide embed -cf 1.jpg（图片文件载体） -ef 1.txt（待隐藏文件） -p 123456（-p参数用于添加密码，非必选）

目的	命令
查看图片中嵌入的文件信息	steghide info 1.jpg
提取图片中隐藏的文件	steghide extract -sf 1.jpg -p 123456

我下载的是 Windows 版本（懒得打开虚拟机...），其使用方法为命令行 cd 到 `steghide.exe` 所在的文件夹后，与 Linux 版本的参数和命令结构都差不多，也是无密码直接回车就好：

```
Cmder
D:\CTF\杂项Misc\steghide-0.5.1-win32\steghide
λ ls
about-nls.txt  copying.txt  cygiconv-2.dll*  cygjpeg-62.dll*  cygmhash-2.dll*  cygz.dll*  LEAME.txt  manual.pdf  README.txt  todo.txt
bugs.txt      credits.txt  cygintl-2.dll*  cygmcrypt-4.dll*  cygwin1.dll*    history.txt  locale/    manual_es.pdf  steghide.exe*

D:\CTF\杂项Misc\steghide-0.5.1-win32\steghide
λ steghide info C:\Users\True\Downloads\389a0c11-d0df-4180-829a-b529e6b0a1bc\_123456cry.jpg.extracted\asd\good-已合并.jpg
format: jpeg
capacity: 1.2 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
embedded file "ko.txt":
size: 48.0 Byte
encrypted: rijndael-128, cbc
compressed: yes

D:\CTF\杂项Misc\steghide-0.5.1-win32\steghide
λ |
```

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

6、上面已经发现图片中隐藏了 ko.txt 文件了，进一步进行信息提取：

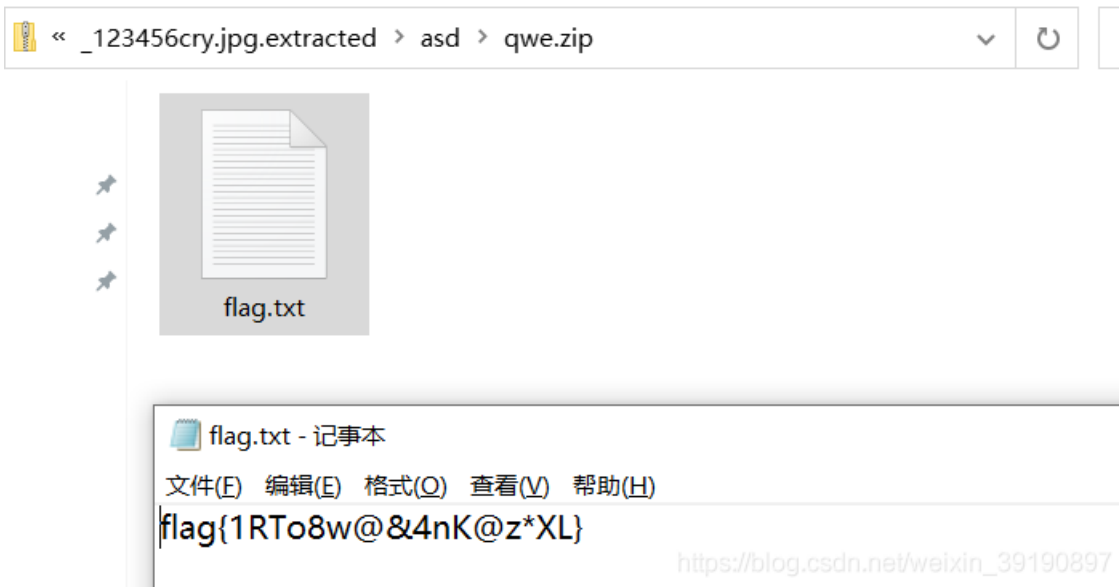
```
D:\CTF\杂项Misc\steghide-0.5.1-win32\steghide
λ steghide extract -sf C:\Users\True\Downloads\389a0c11-d0df-4180-829a-b529e6b0a1bc\_123456cry.jpg.extracted\asd\good-已合并.jpg
Enter passphrase:
wrote extracted data to "ko.txt".

D:\CTF\杂项Misc\steghide-0.5.1-win32\steghide
λ |
```



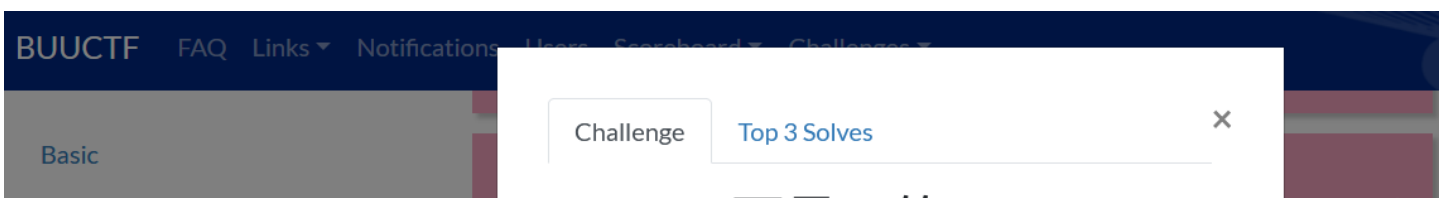
[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

7、拿着上述压缩包密码解密 flag.txt，获得 flag：



[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

## No.20 ZIP伪加密与多重编码转换





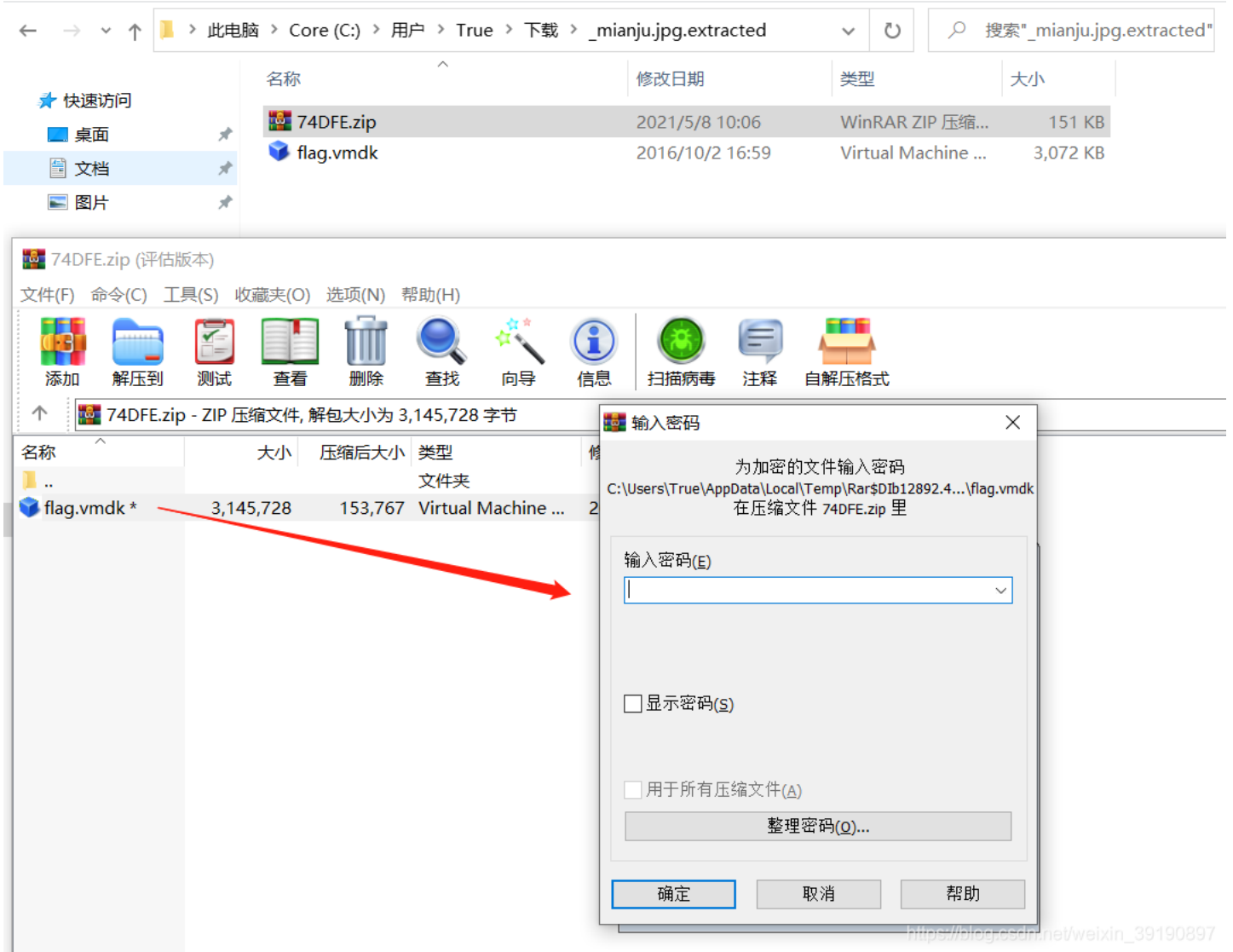


DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, EXIF standard
12	0xC	TIFF image data, little-endian offset of first image directory: 8
478718	0x74DFE	Zip archive data, at least v2.0 to extract, compressed size: 153767, uncompressed size: 3145728, name: flag.vmdk
632615	0x9A727	End of Zip archive, footer length: 22

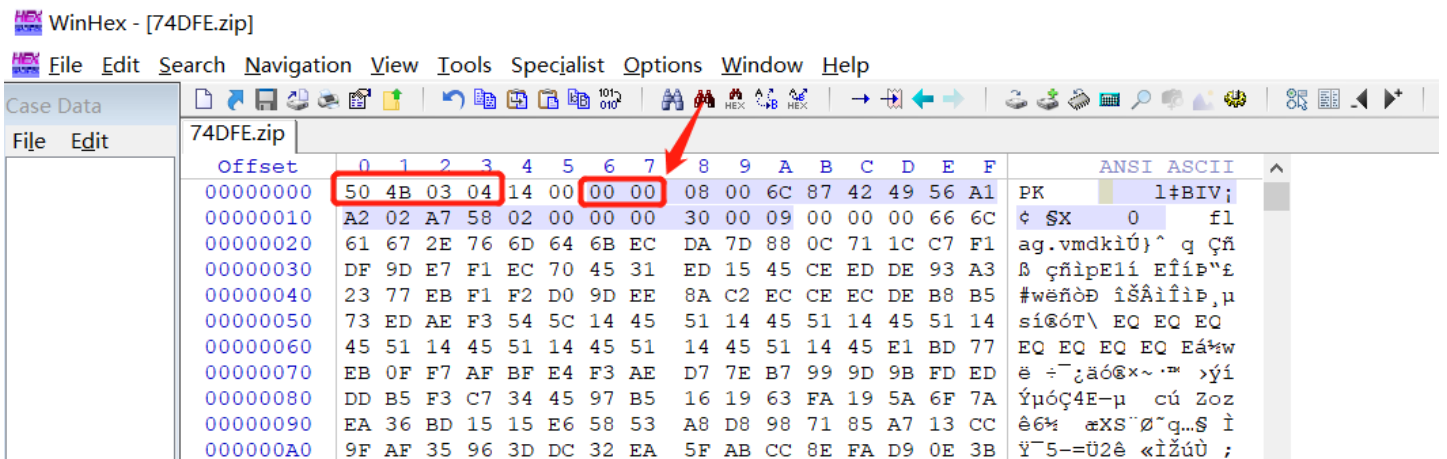
C:\Python37\Scripts  
λ

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

## 2、分离出来的 ZIP 文件需要密码：



## 3、使用 Winhex 打开 ZIP 压缩文件，发现是 ZIP 伪加密（CTF 杂项-ZIP 伪加密与 Base64 隐写实战），如图可见压缩源文件数据区的全局加密为 00 00：



```

000000B0 EA 65 E3 19 BF 33 17 64 16 FA 29 CF EA F2 32 59      3 d  )I  2Y
000000C0 3F 48 D7 87 2D 2F 1D 0F 5C 3F 9D AC 0F 2D 6A 68 ?H*+~/ \? - -jh
000000D0 0A 59 8D 4B A2 F5 61 D7 A9 AB AE 4B 78 56 A7 93 Y K  a*   KxV$  
000000E0 F1 D2 B9 FC A6 44 21 CB CF 2E 77 72 7E 97 D7 92     ;D!E!r    

```

weixin\_39190897

## 真假加密识别

解密情况	识别方法
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
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
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

https://blog.csdn.net/weixin\_39190897

搜索十六进制数值 504B0102（压缩源文件目录区头标志），发现全局方式位标记为 09 00，断定为伪加密：

74DFE.zip

Position Manager (General)

Offset	Search hits	Time
258CE	504B0102	2021/05/08 10:47:27

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
00025810	D4	C9	FF	16	E6	4D	FC	BF	4D	FC	7F	F2	7F	31	FE	EF	���y �M�;M� � lpi	
00025820	2D	FE	DF	25	FE	3F	FB	BF	18	FF	F7	16	FF	EF	13	FF	-pB%b?�; �+ �i �	
00025830	5F	FC	5F	8C	FF	7B	8B	FF	0F	89	FF	AF	FE	2F	C6	FF	_u_Gy{<y %y_p/�y	
00025840	BD	C5	FF	C7	C4	FF	83	FF	8B	F1	7F	6F	F1	FF	90	F8	%�y��yfy<� �ny �	
00025850	FF	E6	FF	62	FC	DF	5B	FC	3F	99	8E	FF	7F	F7	7F	31	y�yb�� [�?��z� + 1	
00025860	FE	EF	2D	FE	9F	25	FE	7F	F8	BF	18	FF	F7	16	FF	CF	pi-pY%b �; �+ �i	
00025870	13	FF	3F	FD	5F	8C	FF	7B	8B	FF	17	89	FF	5F	FE	2F	�?y_Gy{<y %y_p/	
00025880	C6	FF	BD	C5	FF	CB	C4	FF	6F	FF	17	E3	FF	DE	E2	FF	�y%�y��yoy �y��y	
00025890	55	E2	FF	8F	FF	8B	F1	7F	6F	F1	FF	3A	F1	FF	D7	FF	U�y �<� �ny:�y�y	
000258A0	C5	F8	BF	B7	F8	7F	93	F8	FF	E7	FF	62	FC	DF	9B	FE	�; � "�y�yb� >p	
000258B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
000258C0	00	00	00	00	00	00	00	00	00	00	00	90	F1	07	50	4B		� �K
000258D0	01	02	3F	00	14	00	09	00	08	00	6C	87	42	49	56	A1	?	l+BIV;
000258E0	A2	02	A7	58	02	00	00	00	30	00	09	00	24	00	00	00	� \$X	0 \$
000258F0	00	00	00	00	20	00	00	00	00	00	00	00	66	6C	61	67		flag
00025900	2E	76	6D	64	6B	0A	00	20	00	00	00	00	00	01	00	18	.vmdk	
00025910	00	B8	22	67	44	8B	1C	D2	01	70	51	53	E0	85	1C	D2	,"gD< � pQS�... �	
00025920	01	70	51	53	E0	85	1C	D2	01	50	4B	05	06	00	00	00	pQS�... � PK	
00025930	00	01	00	01	00	5B	00	00	00	CE	58	02	00	00	00	00	[ �x	

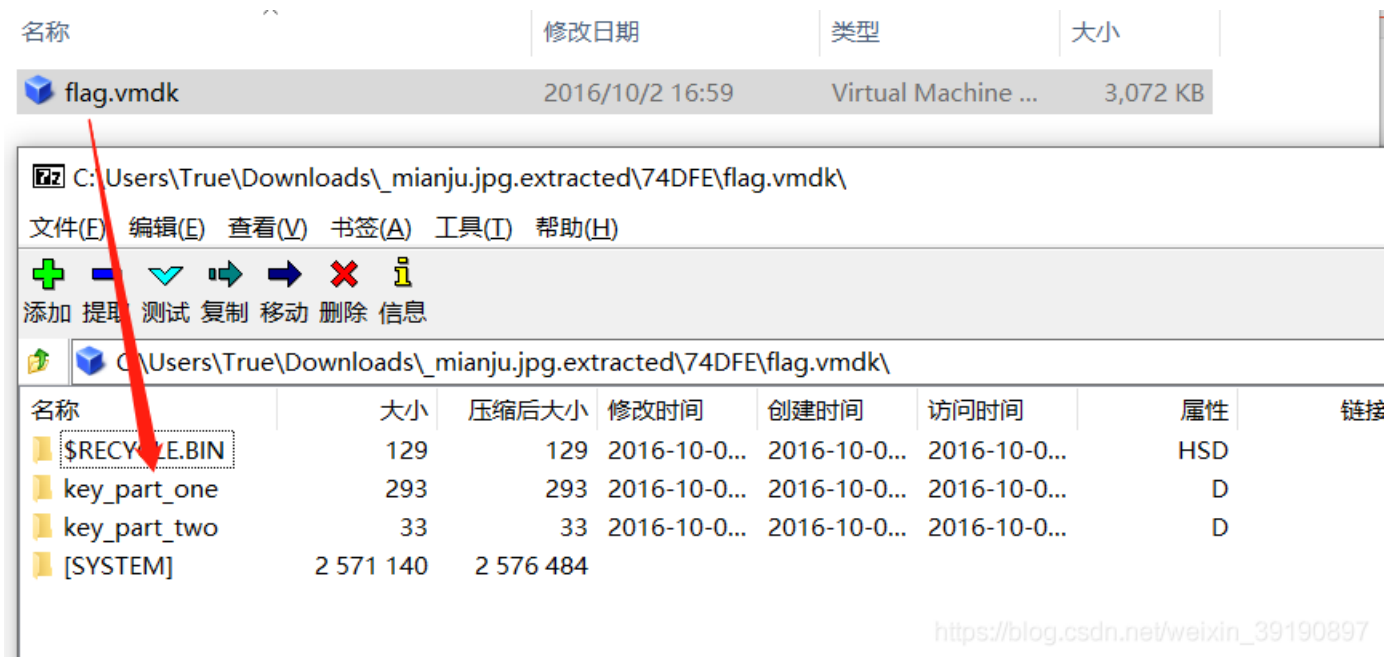
https://blog.csdn.net/weixin\_39190897

4、修改 09 00 为 00 00 后保存，即可解压缩该伪加密的 ZIP 文件，获得如下文件：

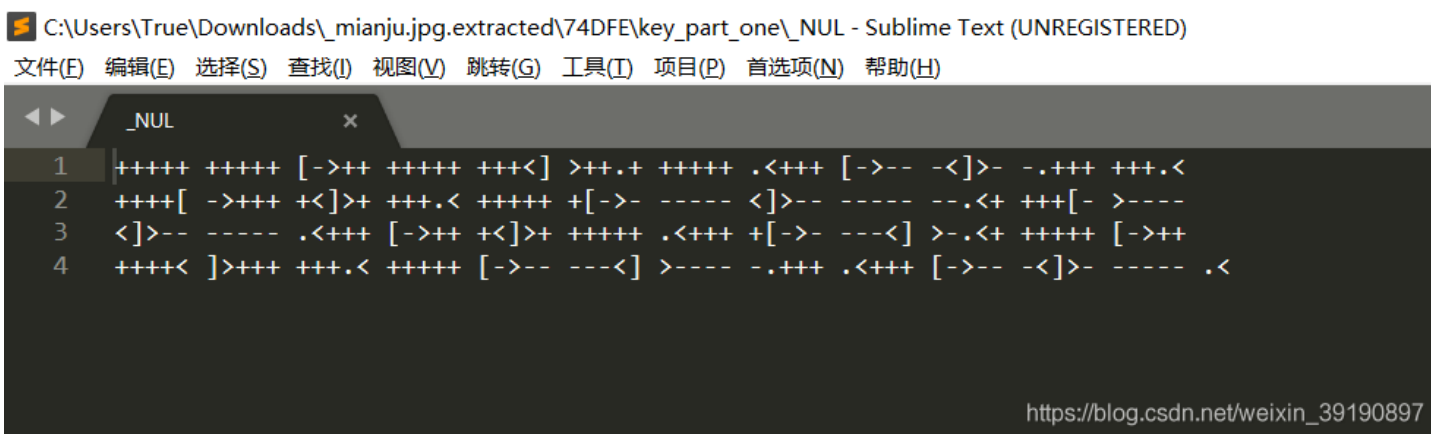
C:\Users\True\Downloads\ mianju.jpg.extracted\74DFE

名称	修改日期	类型	大小
flag.vmdk	2016/10/2 16:59	Virtual Machine ...	3,072 KB

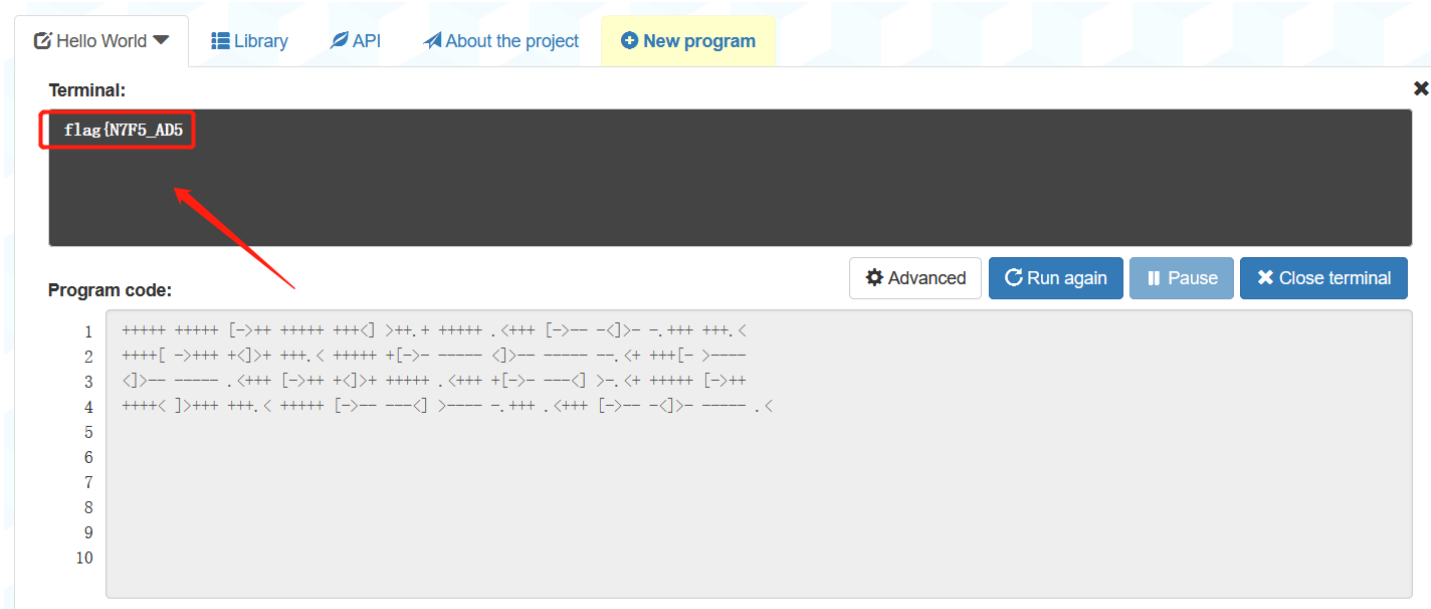
5、对于 vmdk 文件，需要使用 [7-ZIP](#) 工具解压缩（[官网下载地址](#)），安装后打开后压缩包，如下：



6、打开 Key\_part\_one 文件夹里面的文件：



Brainfuck 编码，[Brainfuck在线解密](#)：



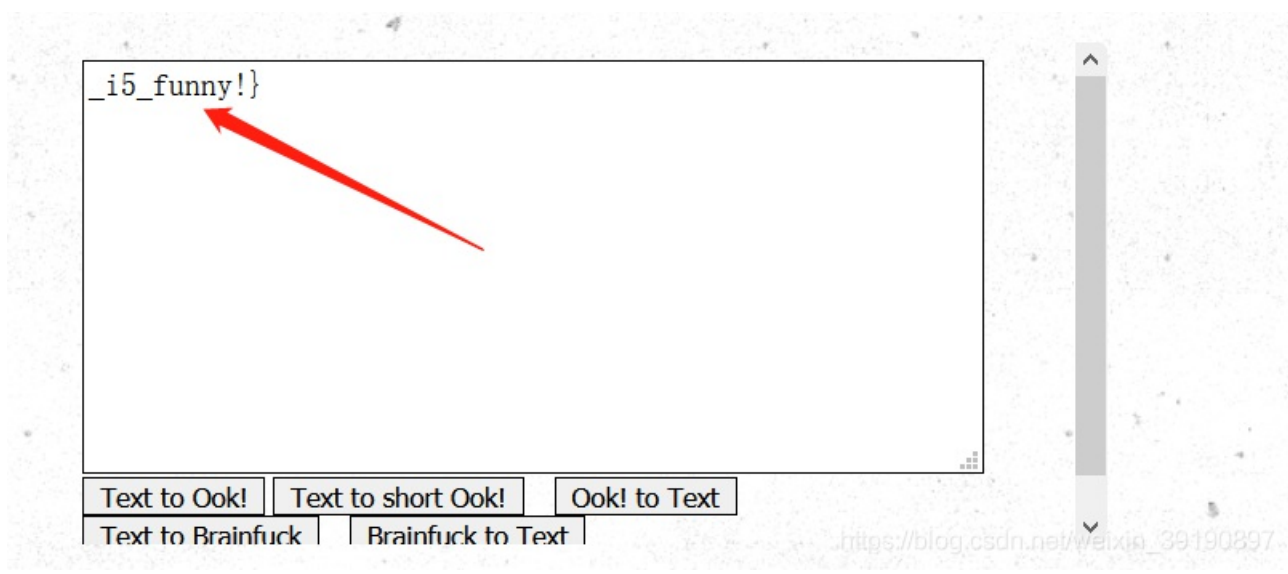
7、打开 Key\_part\_one 文件夹里面的文件：

```

0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok.
0ok. 0ok. 0ok. 0ok! 0ok? 0ok! 0ok! 0ok. 0ok? 0ok. 0ok. 0ok. 0ok. 0ok. 0ok.
0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok? 0ok. 0ok?
0ok! 0ok. 0ok? 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok.
0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok.
0ok. 0ok! 0ok. 0ok? 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok! 0ok? 0ok! 0ok!
0ok. 0ok? 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok? 0ok. 0ok? 0ok! 0ok. 0ok? 0ok.
0ok. 0ok! 0ok. 0ok? 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok.
0ok. 0ok. 0ok. 0ok. 0ok! 0ok? 0ok! 0ok! 0ok. 0ok? 0ok! 0ok! 0ok! 0ok! 0ok!
0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok? 0ok. 0ok? 0ok! 0ok. 0ok?
0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok. 0ok? 0ok. 0ok. 0ok. 0ok. 0ok. 0ok.
0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok! 0ok? 0ok! 0ok! 0ok. 0ok? 0ok. 0ok.
0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok? 0ok. 0ok? 0ok! 0ok.
0ok? 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok! 0ok.
0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok!
0ok. 0ok? 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok! 0ok? 0ok! 0ok! 0ok. 0ok?
0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok? 0ok. 0ok? 0ok! 0ok. 0ok? 0ok. 0ok. 0ok.
0ok. 0ok? 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok.
0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok! 0ok? 0ok! 0ok! 0ok. 0ok? 0ok! 0ok! 0ok!
0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok!
0ok? 0ok. 0ok? 0ok! 0ok. 0ok? 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok!
0ok! 0ok! 0ok! 0ok! 0ok! 0ok! 0ok. 0ok? 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok.
0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok! 0ok? 0ok!
0ok! 0ok. 0ok? 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok.
0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok? 0ok. 0ok? 0ok! 0ok. 0ok? 0ok. 0ok. 0ok.
0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok. 0ok.
0ok. 0ok. 0ok. 0ok. 0ok! 0ok. 0ok? 0ok.

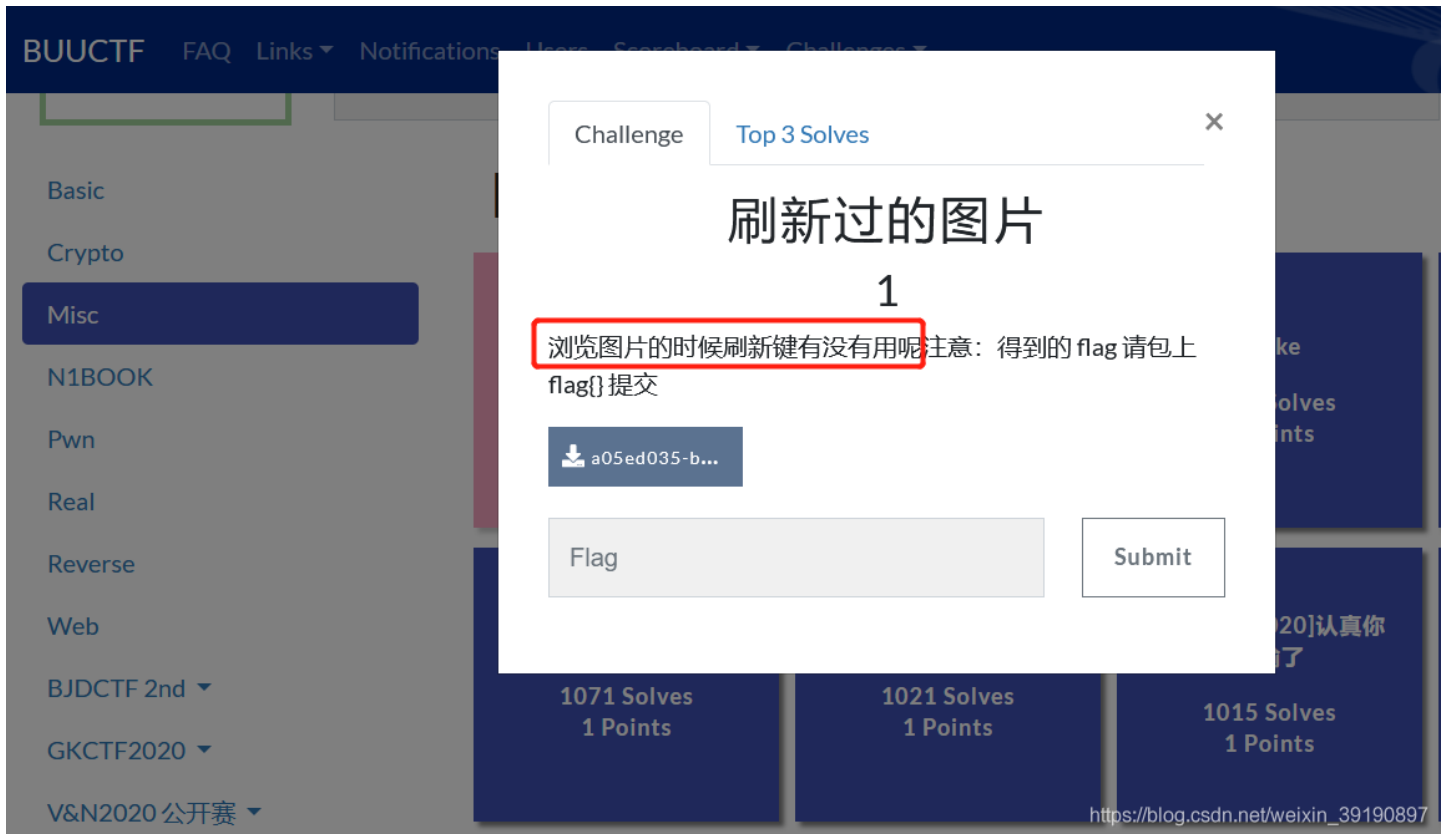
```

Ook! 编码, [在线解密](#):



拼接获得 flag: `flag{N7F5_AD5_i5_funny!}`。

## No.21 F5隐写与ZIP文件头的识别



The screenshot shows a CTF challenge page with a modal window. The modal window has a title bar with "Challenge" and "Top 3 Solves". The main content of the modal is:

### 刷新过的图片

1

浏览图片的时候刷新键有没有用呢 注意：得到的 flag 请包上 flag{} 提交

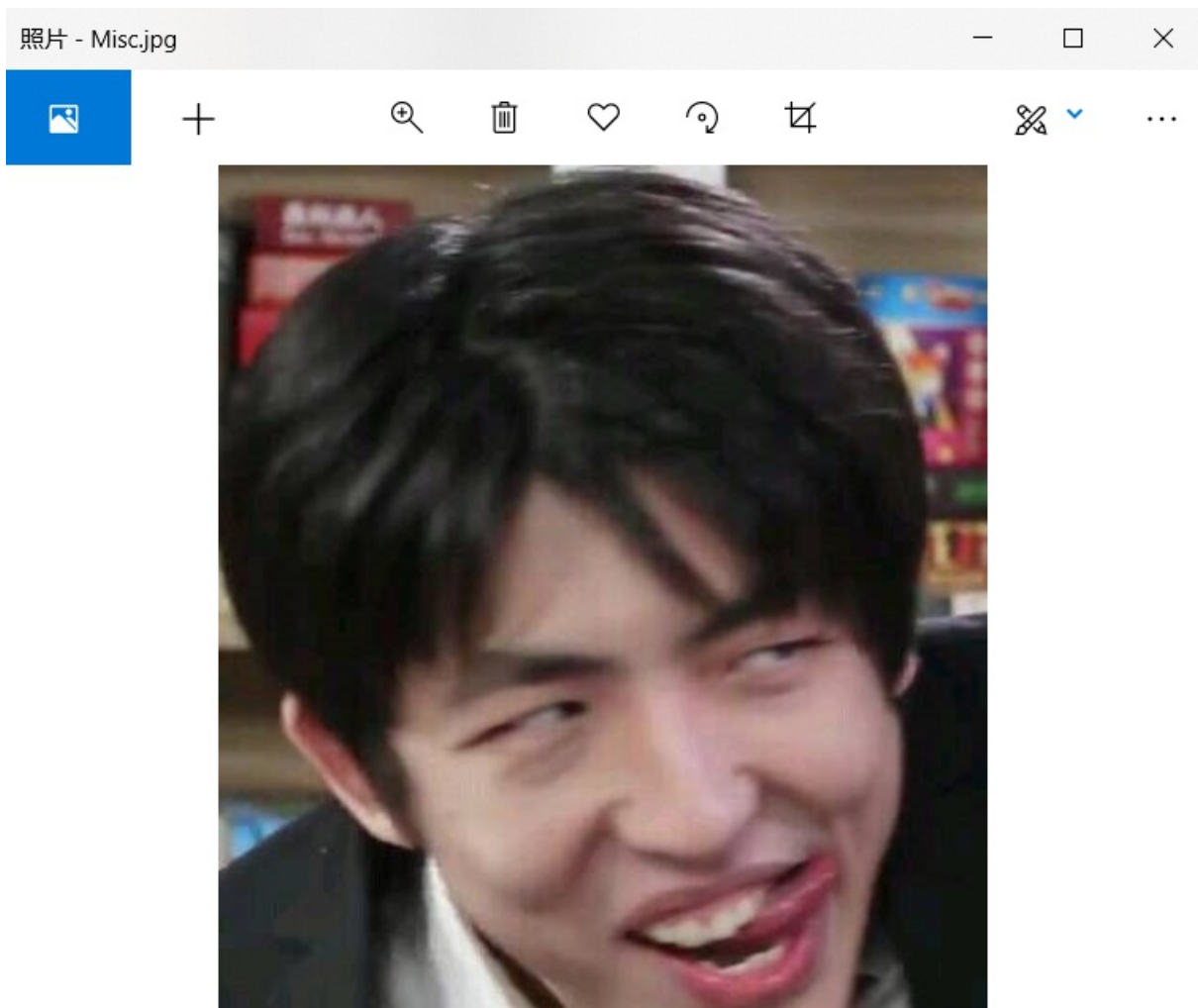
Download button: a05ed035-b...

Flag input field: Flag

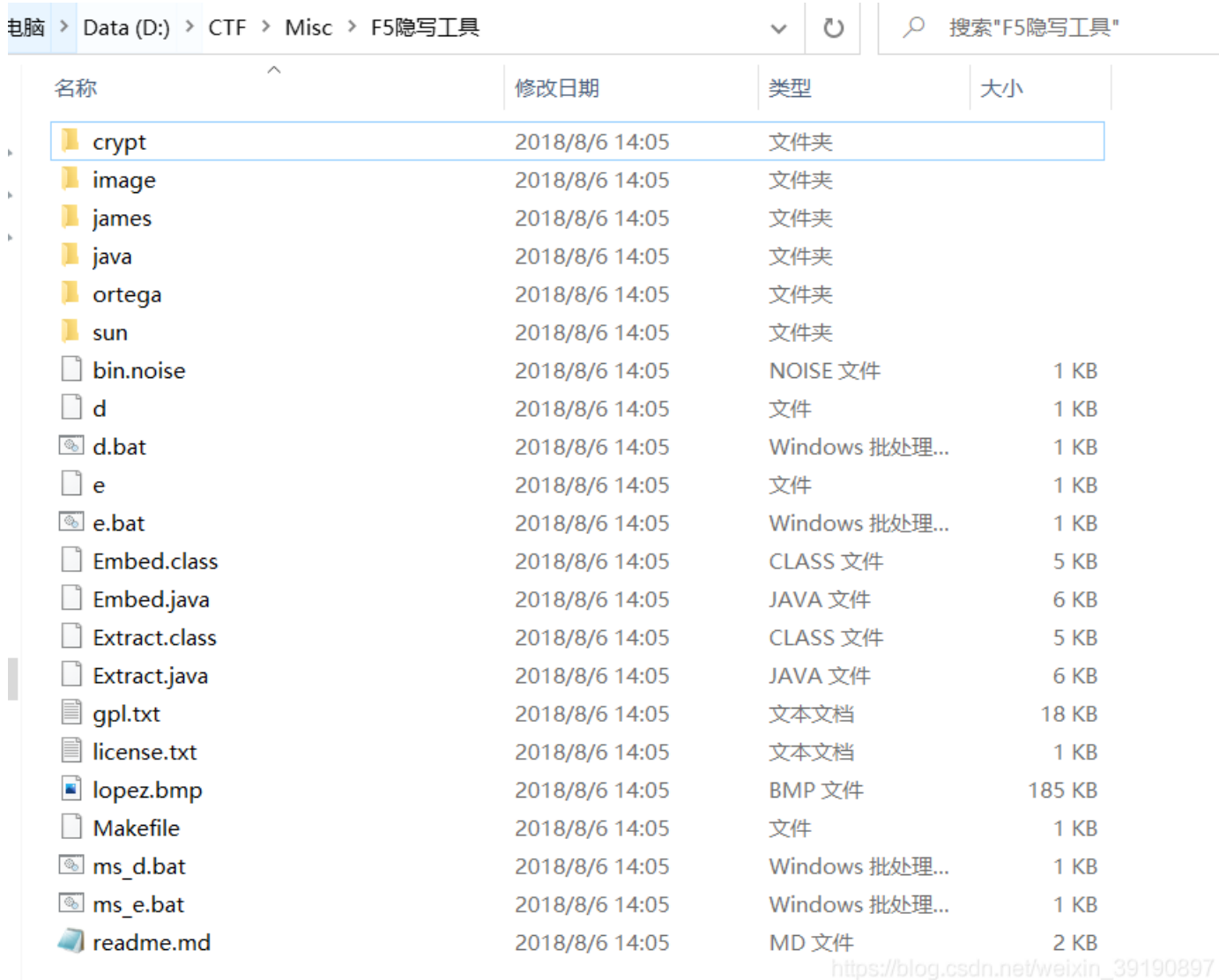
Submit button: Submit

The background shows a sidebar with categories: Basic, Crypto, Misc (selected), N1BOOK, Pwn, Real, Reverse, Web, BJDCTF 2nd, GKCTF2020, and V&N2020 公开赛. At the bottom, there are statistics for "Top 3 Solves": 1071 Solves 1 Points, 1021 Solves 1 Points, and 1015 Solves 1 Points. A URL is visible at the bottom right: [https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

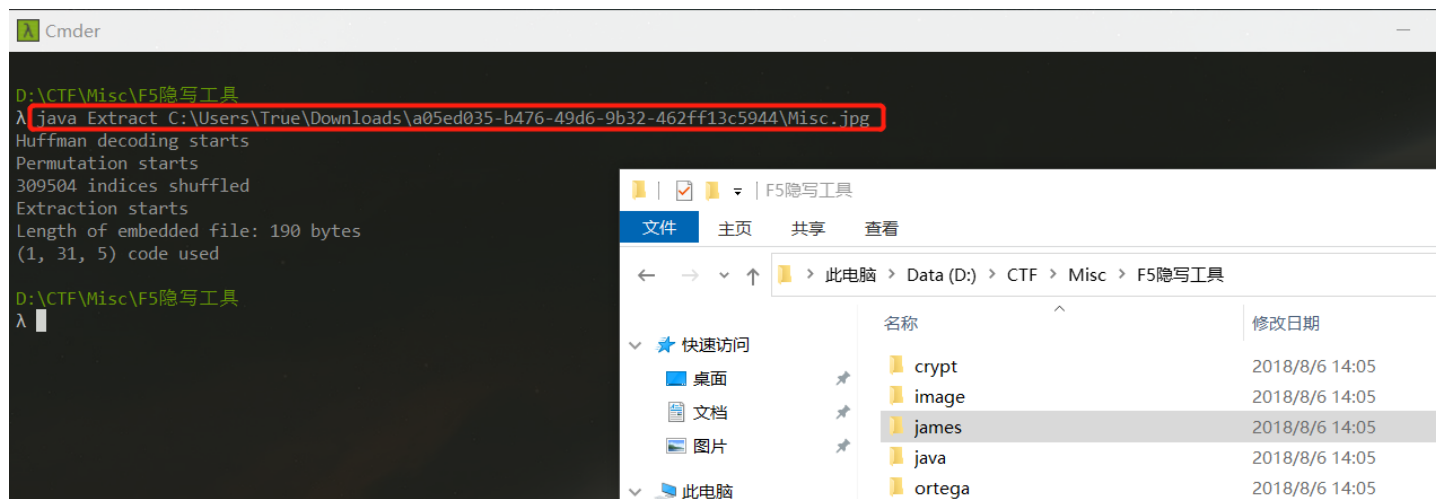
下载后是一张图片：



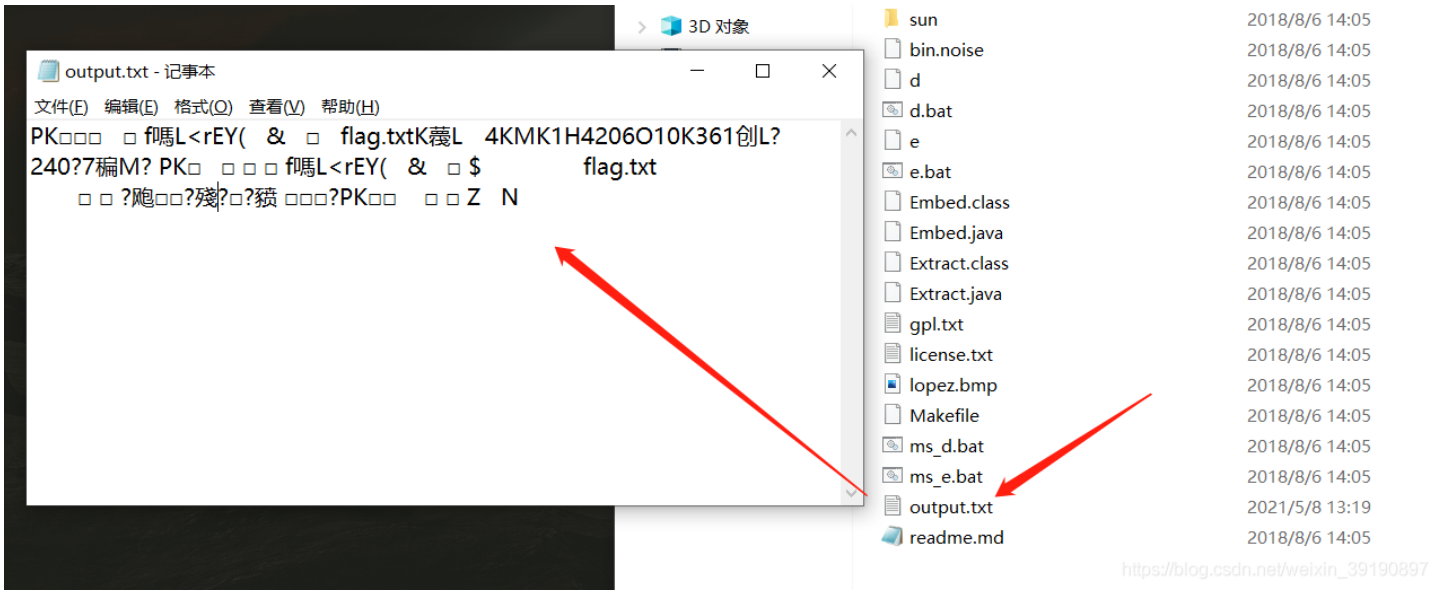
使用各种解密软件（Stegsolve、steghide、binwalk）都没有找到问题所在，回到题目提示：“浏览图片的时候刷新键有没有用呢”，刷新，F5??? 在百度的帮助下了解了 F5 隐写（小白啊，真的蒙了）.....还真有！F5 隐写的原理可参见：F5 隐写算法及其隐写分析研究，此处直接上 F5 隐写利用工具（F5-steganography的Github地址）：



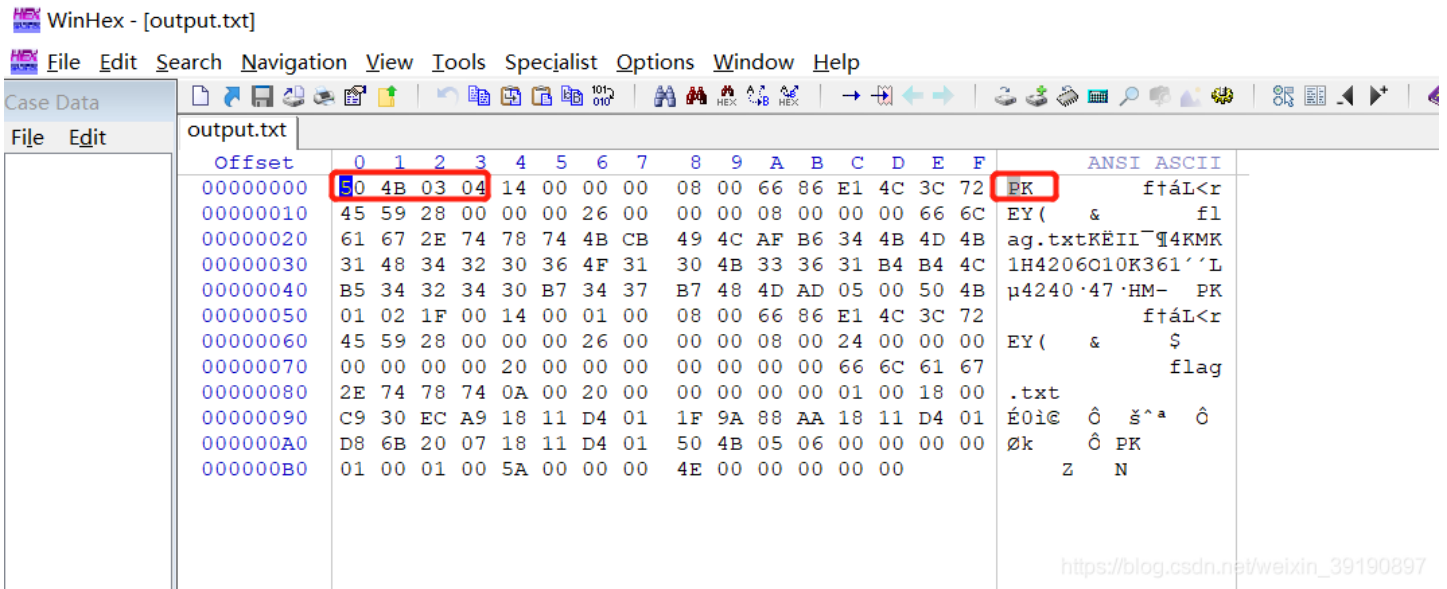
1、在该工具的文件夹路径下执行命令 `java Extract 图片路径` 即可在输出 `output.txt` 文件记录了图片隐写的信息：



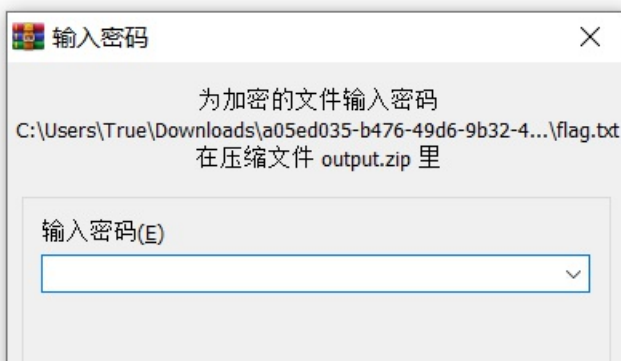


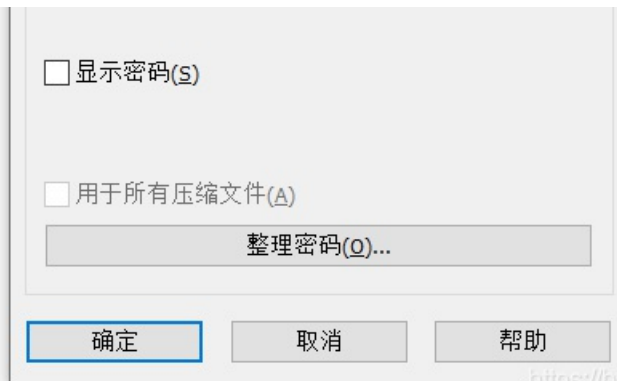


2、使用 Winhex 打开 output.txt 发现 ZIP 文件的文件头（实际上根据打开的 txt 文件的 PK 首部字符也可猜测是 ZIP 文件）：



3、故修改 output.txt 后缀获得 output.zip 压缩文件，解压发现加密：





[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

4、伪加密，将 01 00 改为 00 00 后保存，即可解压：

output.txt output.zip

Position Manager (General)

Offset	Search hits ▲	Time
258CE 504B0102		2021/05/08 10:54:05
4E 504B0102		2021/05/08 13:32:12

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
00000000	50	4B	03	04	14	00	00	00	08	00	66	86	E1	4C	3C	72	PK	f t á L < r
00000010	45	59	28	00	00	00	26	00	00	00	08	00	00	00	66	6C	EY (	& f l
00000020	61	67	2E	74	78	74	4B	CB	49	4C	AF	B6	34	4B	4D	4B	ag.txt	K È I I ~ ¶ 4 K M K
00000030	31	48	34	32	30	36	4F	31	30	4B	33	36	31	B4	B4	4C	1H4206010K361' 'L	
00000040	B5	34	32	34	30	B7	34	37	B7	48	4D	AD	05	00	50	4B	µ4240·47·HM-	PK
00000050	01	02	1F	00	14	00	01	00	08	00	66	86	E1	4C	3C	72		f t á L < r
00000060	45	59	28	00	00	00	26	00	00	00	08	00	24	00	00	00	EY (	& \$
00000070	00	00	00	00	20	00	00	00	00	00	00	00	66	6C	61	67		flag
00000080	2E	74	78	74	0A	00	20	00	00	00	00	00	01	00	18	00	.txt	
00000090	C9	30	EC	A9	18	11	D4	01	1F	9A	88	AA	18	11	D4	01	É 0 i @	Ô š ^ a Ô
000000A0	D8	6B	20	07	18	11	D4	01	50	4B	05	06	00	00	00	00	ø k	Ô PK
000000B0	01	00	01	00	5A	00	00	00	4E	00	00	00	00	00	00	00	Z	N

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

5、解压缩文件后获得 flag：

> a05ed035-b476-49d6-9b32-462ff13c5944 > output

名称	修改日期	类型	大小
flag.txt	2018/7/1 16:51	文本文档	

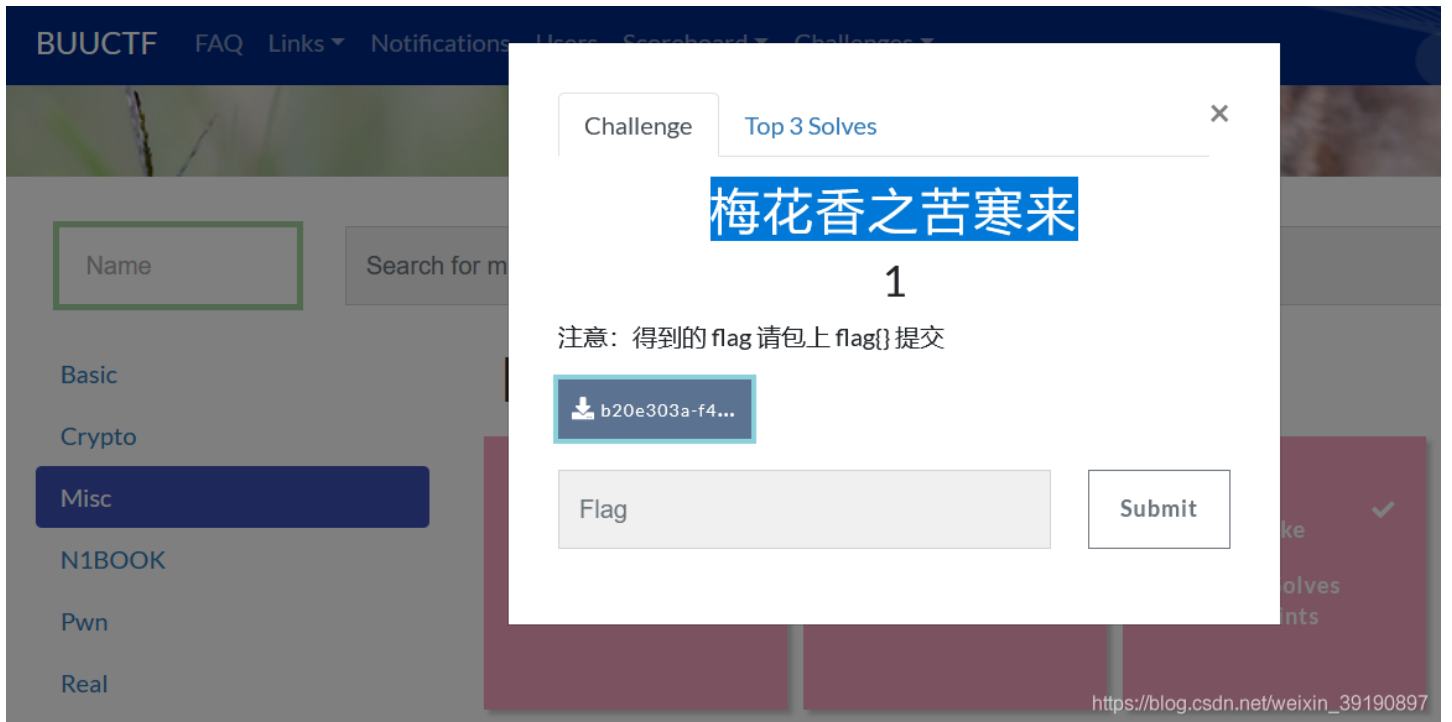
flag.txt - 记事本

文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)

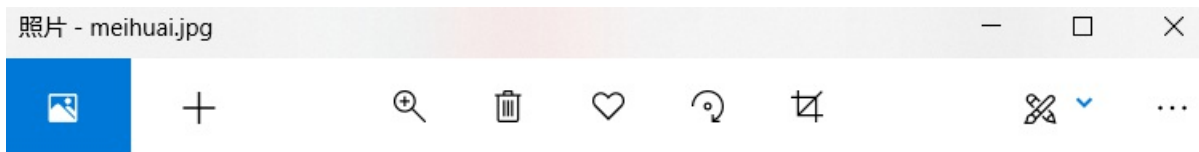
flag{96efd0a2037d06f34199e921079778ee}

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

## No.22 Py脚本16进制坐标绘二维码



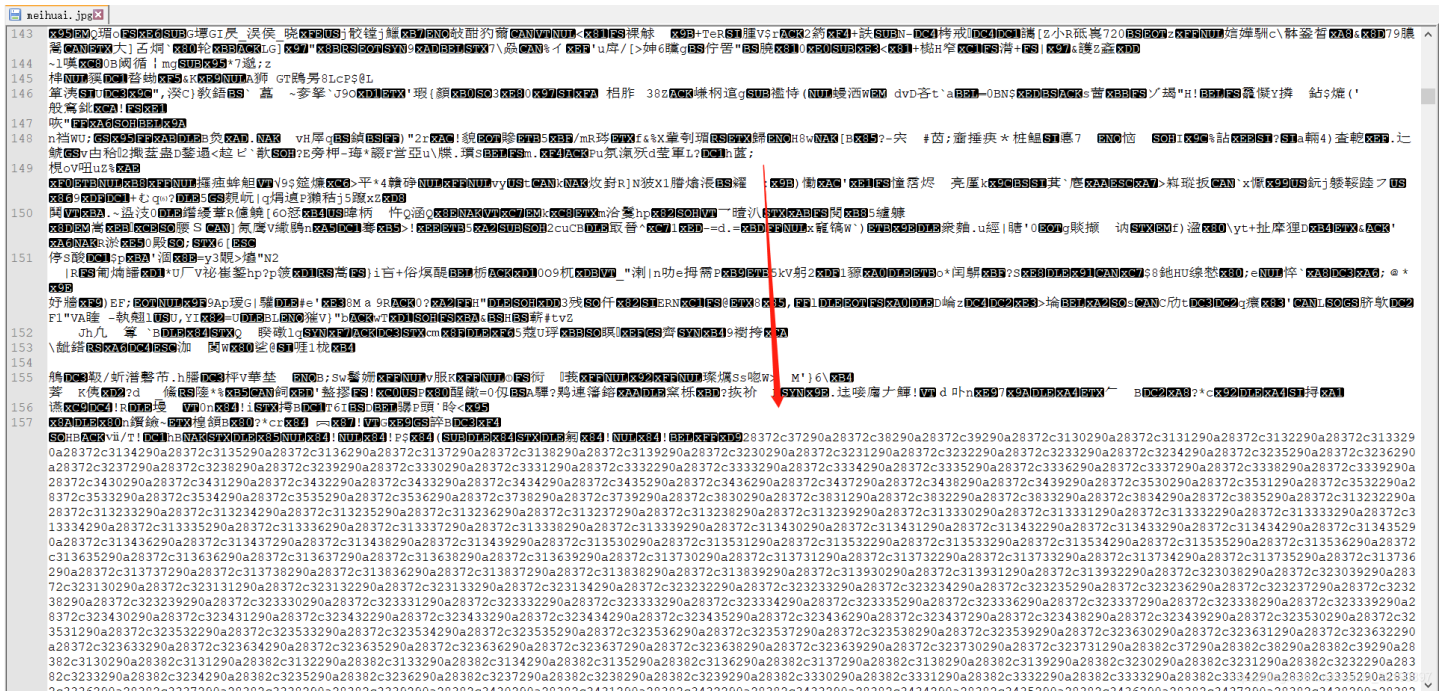
下载后是一张图片：



[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

1、编辑器或者 winhex 打开图片，发现末尾存在大量十六进制：

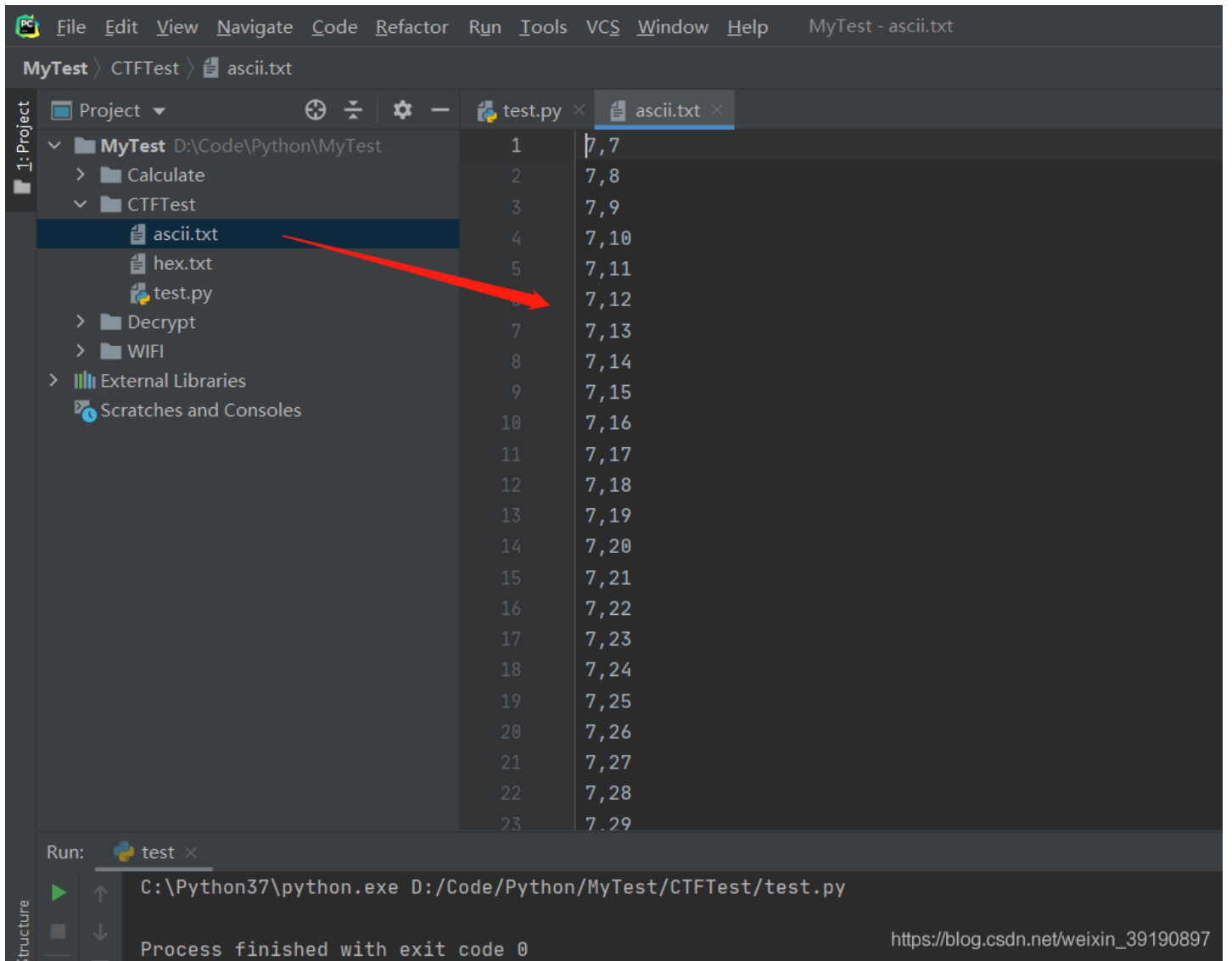




2、将 16 进制数据复制到 hex.txt 利用脚本转化为 ascii 获取坐标点：

```
with open('hex.txt', 'r') as h:  
    h = h.read()  
with open('./ascii.txt', 'a') as a:  
    for i in range(0, len(h), 2):  
        tmp = '0x'+h[i]+h[i+1]  
        tmp = int(tmp, base=16)  
        if chr(tmp) != '(' and chr(tmp) != ')':  
            a.write(chr(tmp))
```

执行脚本:



The screenshot shows an IDE window titled "MyTest - ascii.txt". The project structure on the left includes "MyTest", "Calculate", "CTFTest", "hex.txt", "test.py", "Decrypt", "WIFI", "External Libraries", and "Scratches and Consoles". The main editor area displays a list of coordinates from 1 to 23, each followed by a pair of numbers separated by a comma. A red arrow points from the "test.py" file in the project structure to the list of coordinates. The Run console at the bottom shows the command "C:\Python37\python.exe D:/Code/Python/MyTest/CTFTest/test.py" and the message "Process finished with exit code 0".

1	7,7
2	7,8
3	7,9
4	7,10
5	7,11
6	7,12
7	7,13
8	7,14
9	7,15
10	7,16
11	7,17
12	7,18
13	7,19
14	7,20
15	7,21
16	7,22
17	7,23
18	7,24
19	7,25
20	7,26
21	7,27
22	7,28
23	7,29

Run: test ×  
C:\Python37\python.exe D:/Code/Python/MyTest/CTFTest/test.py  
Process finished with exit code 0

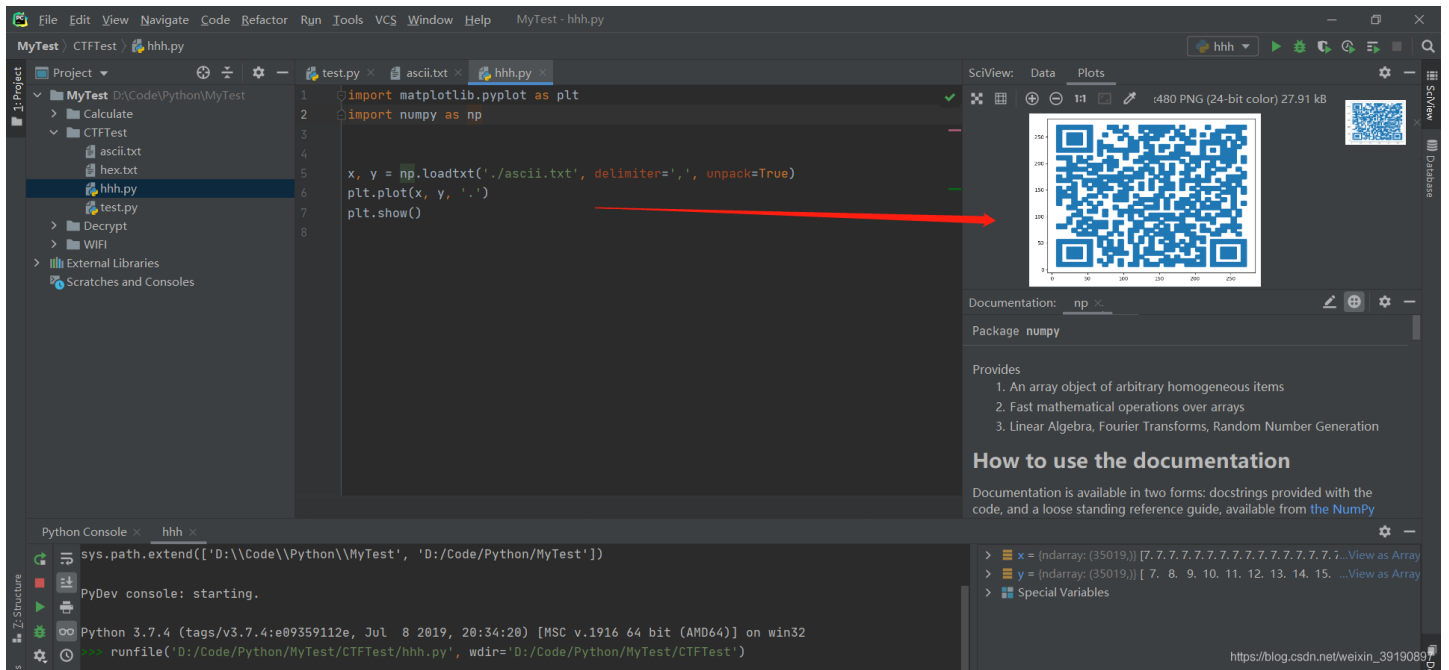
[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

3、用 matplotlib 绘图得到二维码:

```
import matplotlib.pyplot as plt
import numpy as np

x, y = np.loadtxt('./ascii.txt', delimiter=',', unpack=True)
plt.plot(x, y, '.')
plt.show()
```

生成二维码:



扫描二维码获得 flag:



最终脚本:

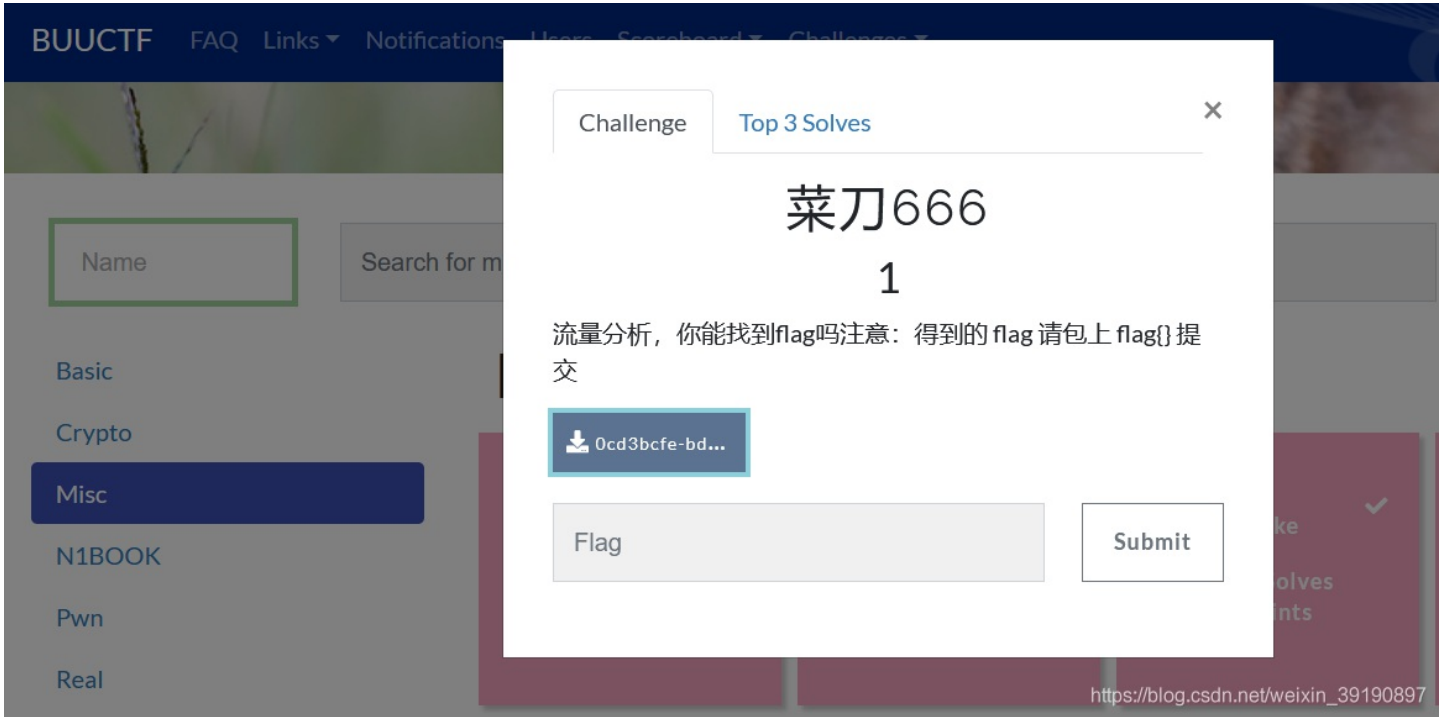
```
import matplotlib.pyplot as plt
import numpy as np

with open('hex.txt', 'r') as h:
    h = h.read()
with open('./ascii.txt', 'a') as a:
    for i in range(0, len(h), 2):
        tmp = '0x'+h[i]+h[i+1]
        tmp = int(tmp, base=16)
        if chr(tmp) != '(' and chr(tmp) != ')':
            a.write(chr(tmp))

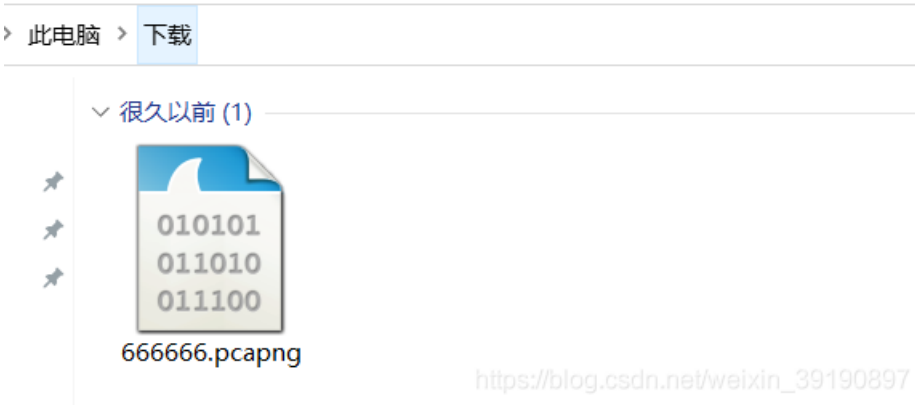
x, y = np.loadtxt('./ascii.txt', delimiter=',', unpack=True)
plt.plot(x, y, '.')
plt.show()
```

## No.22 HTTP流量分析与文件转换

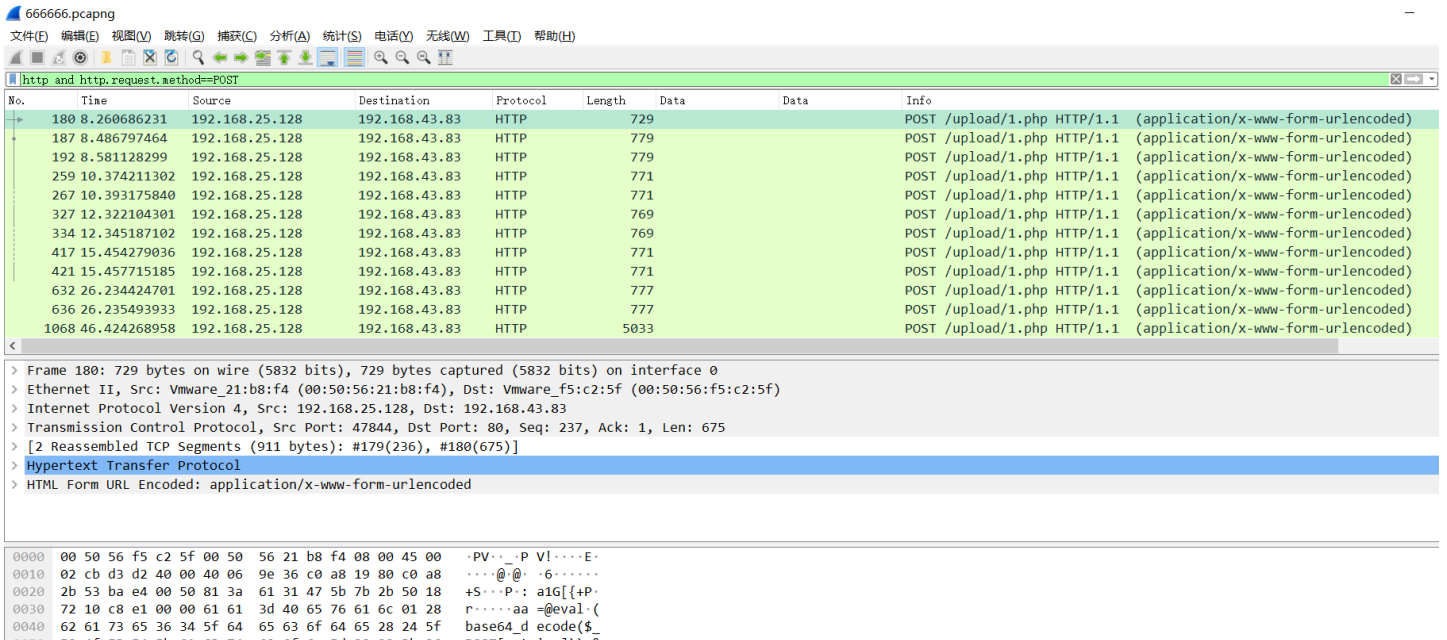


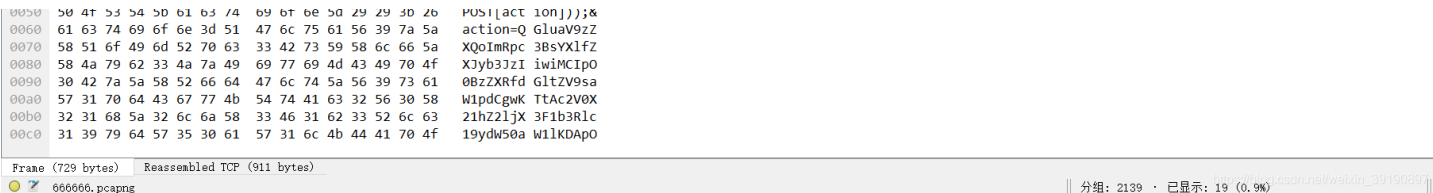


下载后是一个流量文件:

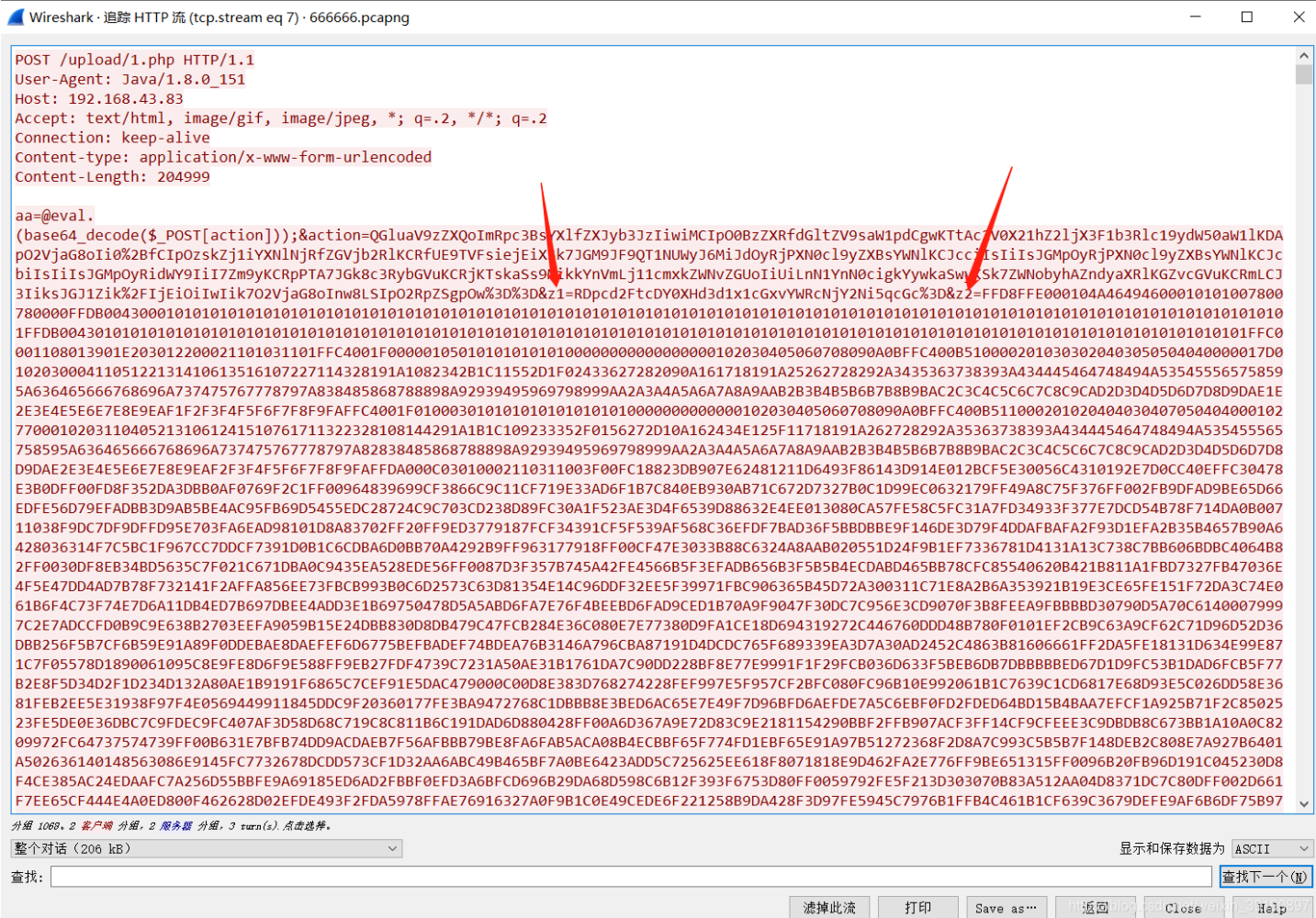


1、直接过滤出 HTTP 流量，题目因为是菜刀吗？一般都是 post 连接，于是我们过滤 post 数据 `http.request.method==POST`，然后分析流量：

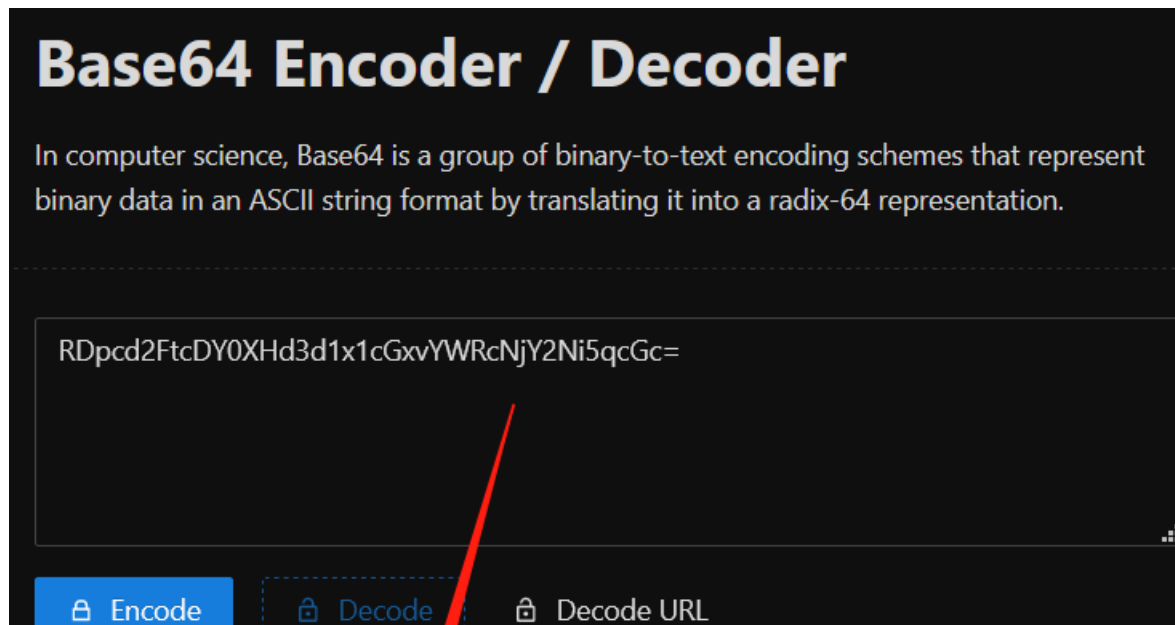




2、当我们分析到流7时，发现了base64 编码，解码一看是上传的图片的地址：



z1 的值拿去解码得到：





D:\wamp64\www\upload\6666.jpg

[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

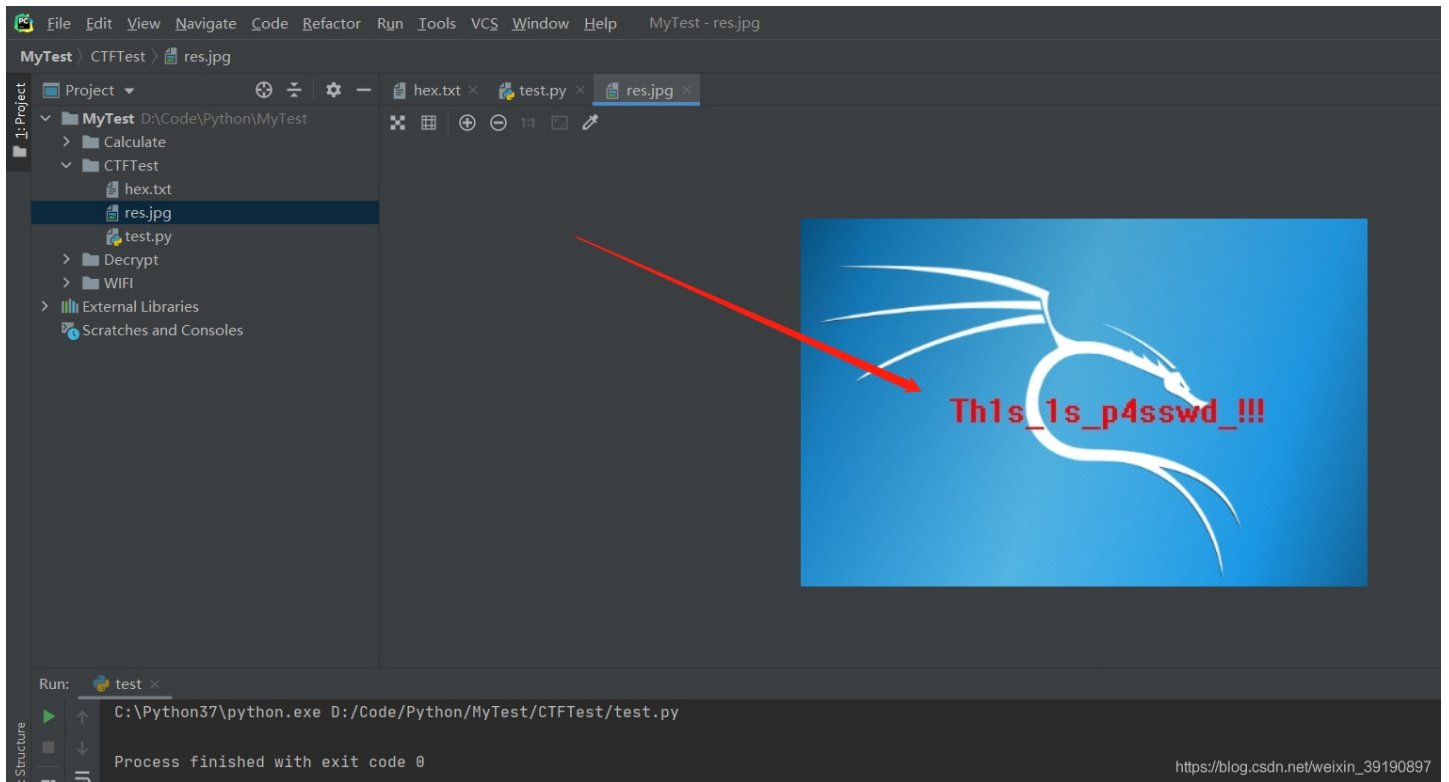
后面跟了个 z2 参数 FF D8 开头 FF D9 结尾，判断为 jpg 图片，将这些十六进制复制出来，以原始文件流写入文件：

```
import struct

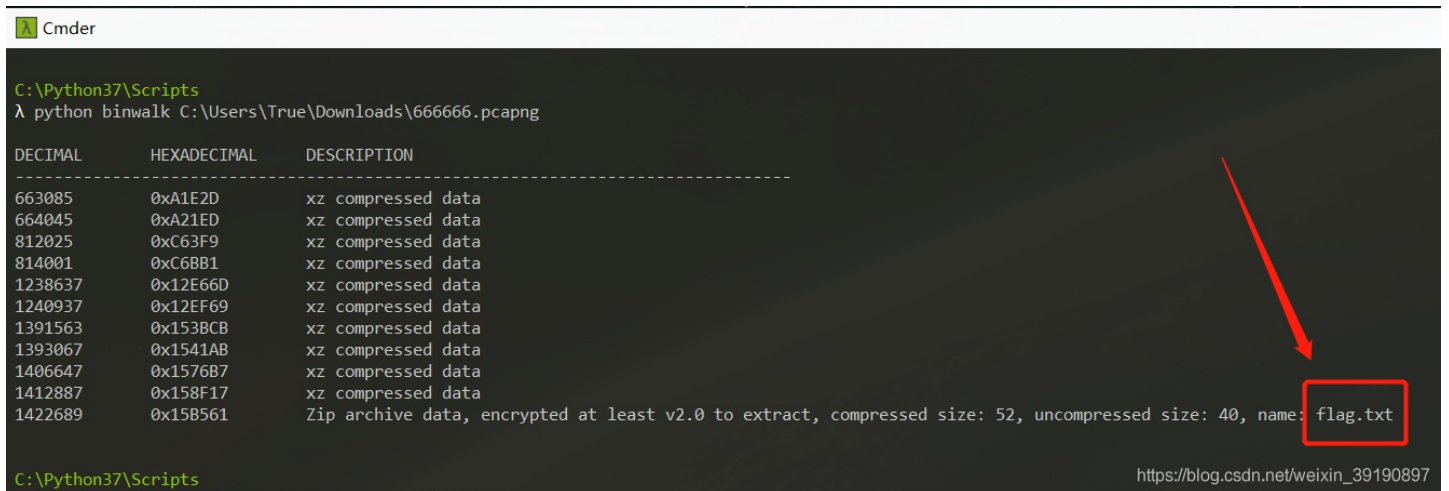
a = open("hex.txt", "r") # 十六进制数据文件
lines = a.read()
res = [lines[i:i+2] for i in range(0, len(lines), 2)]

with open("res.jpg", "wb") as f:
    for i in res:
        s = struct.pack('B', int(i, 16))
        f.write(s)
```

执行脚本获得图片：



2、使用 Binwalk 分析 666666.pcapng 文件，发现隐藏着一个 flag.txt，提取：



分离出来一个压缩包，需要密码，难道这就是压缩包的密码，然后输入密码，打开了文件，然后获得了flag: `f1ag{30pWdJ-JP6FzK-koCMAK-VkFwBq-75Un2z}`。

## No.23 HTTP流量分析与文件提取

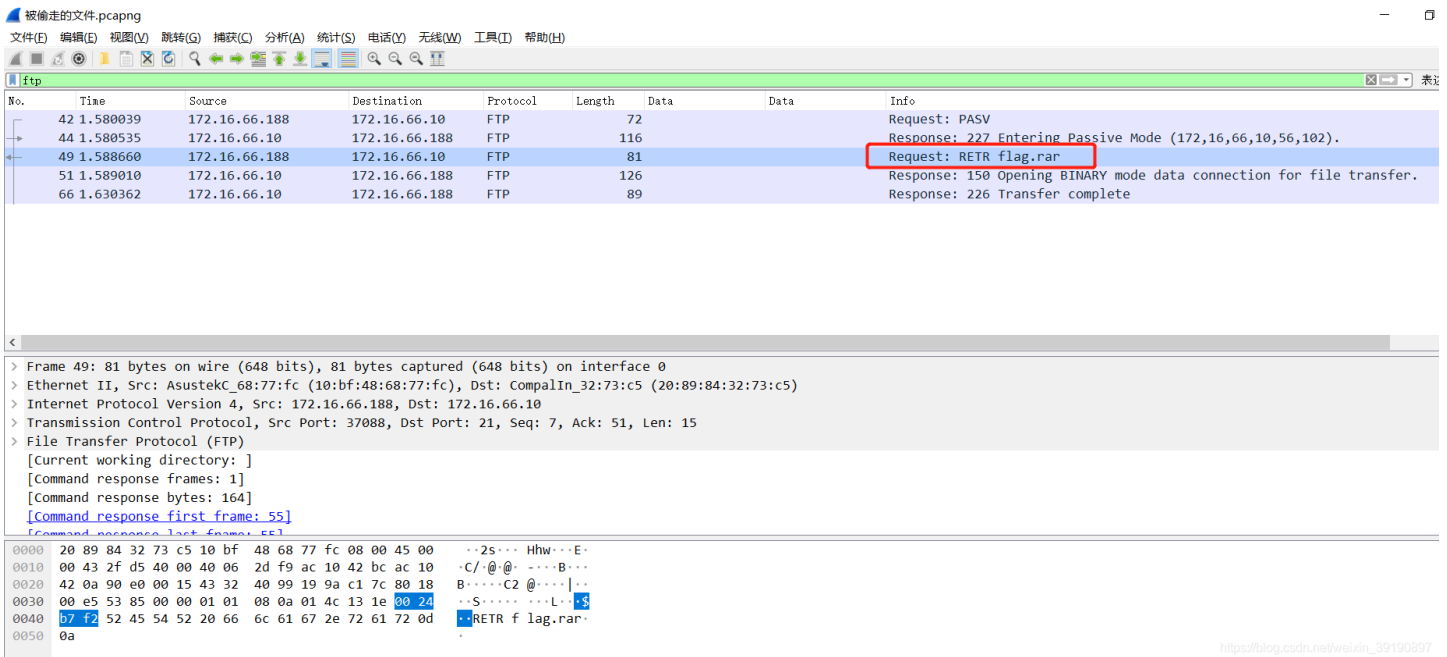




下载后如下:



1、传输文件看到了 ftp 协议，过滤一下，发现 flag.rar 文件:



2、使用 Binwalk 分析流量文件 `被偷走的文件.pcapng`，发现隐藏的文件，进行提取：

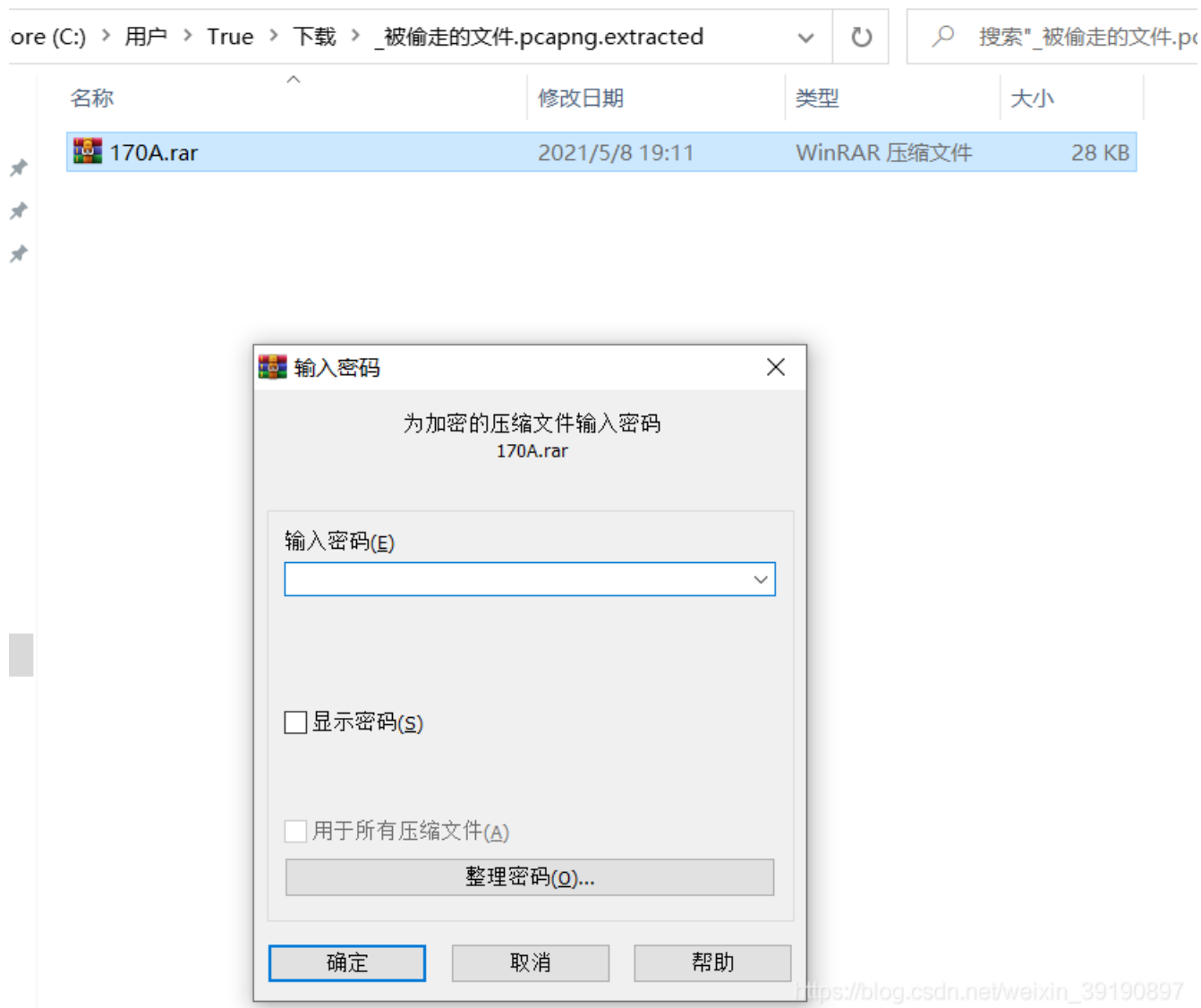
```
Cmder
C:\Python37\Scripts
λ python binwalk C:\Users\True\Downloads\被偷走的文件.pcapng

DECIMAL      HEXADECIMAL  DESCRIPTION
-----
5898         0x170A      RAR archive data, version 4.x, first volume type: MAIN_HEAD

C:\Python37\Scripts
λ |
```

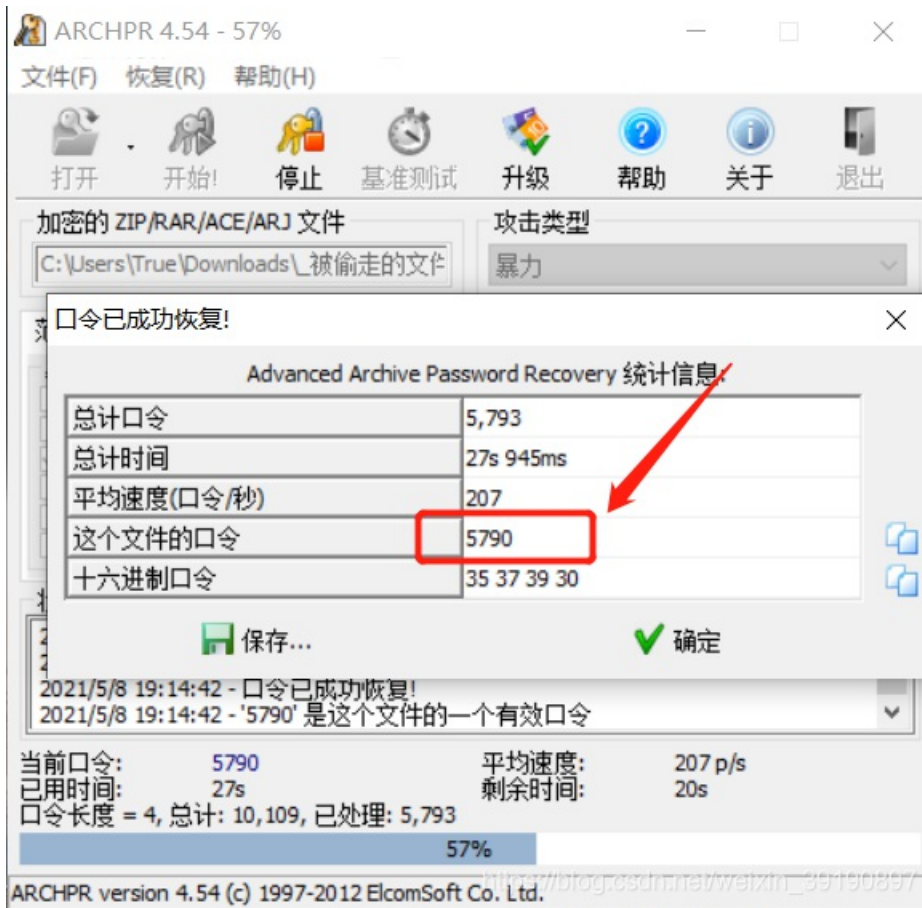
[https://blog.csdn.net/weixin\\_39190897](https://blog.csdn.net/weixin_39190897)

3、获得 170A.rar 文件，需要密码：





4、尝试 ARCHPR 四位数字密码爆破：



5、输入密码获得 flag:



## 总结

本文是 BUUCTF 较为简单的杂项题目的汇集，总结下本文所涉及的一些杂项题目解题思路，主要还是图片隐写：

1. 拿到文件可先查看属性是否有特殊信息；
2. 可使用 Winhex 或者 Notepad++ 编辑器查看文件是否有特殊信息，同时关注是否隐藏常见的文件头；
3. 必要时可使用 Winhex 修改图片的长、宽数值获得 flag；
4. 隐写图片查看的神器-----stegsolve，可以帮助查看文件信息、分离动图、处理 LSB 隐写；
5. Binwalk 工具可快速分辨文件是否由多个文件合并而成，并可将文件进行分离提取；
6. 对于压缩文件的加密，可看看是不是 ZIP 伪加密或者尝试使用 ARCHPR 工具进行四位数字的爆破；
7. 在 Binwalk 无法分析出存在合并的隐藏文件的情况下，可以考虑使用 Steghide 隐写工具查看是否隐写了字符串。

CTF 杂项题目还很多知识和套路不懂，路漫漫其修远兮.....