实验吧——WriteUp&&涨姿势（3）

原创

浅零半泣 于 2017-05-20 14:55:09 发布 2587 收藏 3

分类专栏： CTF

本文链接：https://blog.csdn.net/sinat_34200786/article/details/72576910

WriteUP

1. Fonts
2. 大雄和哆啦A梦
3. 水果
4. 无处不在的广告
5. 想看正面？那就要看仔细了！
6. 多啦a梦
7. 打不开的文件
8. 复杂的QR_code

涨姿势

1. 刷新 刷新 快刷新
2. guess
3. 小苹果
4. 男神一般都很低调很低调的！！
5. 最低位的亲吻

Fonts

原题

解题思路

```
打开文件什么都没发现，用HxD看看
```

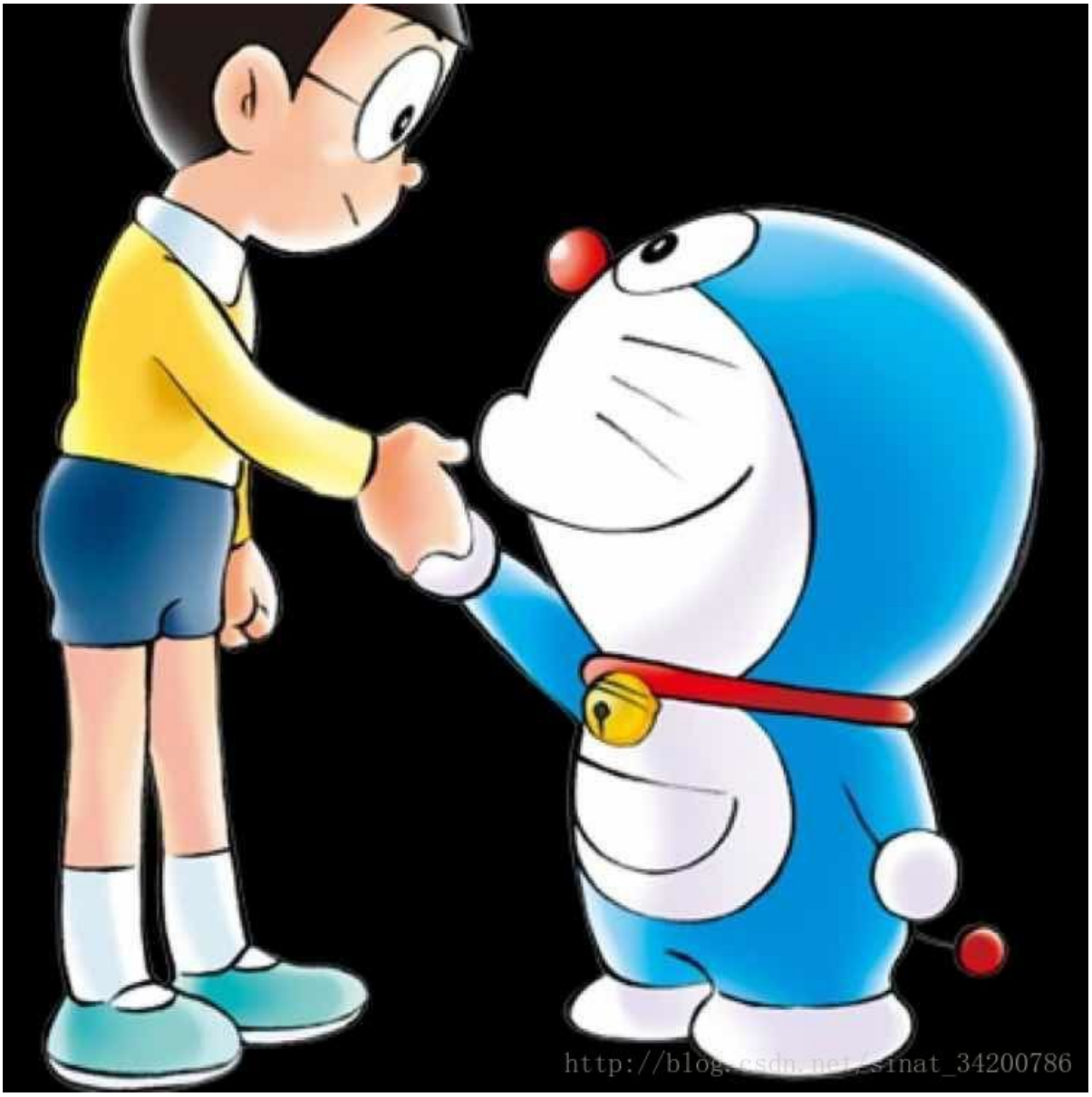
WriteUP

```
HxD搜索关键字直接得到flag
```

```
000009D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000009E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000009F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000A00  43 54 46 7B 46 5F 21 6F 21 5F 21 6E 21 5F 74 5F  CTF{F_!o!_!n!_t_
00000A10  73 7D 0D 00 00 00 00 00 00 00 00 00 00 00 00 00  s}..............
00000A20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000A30  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

大雄和哆啦A梦

原题



解题思路

```
binwalk 发现隐藏数据，分离数据，关键字做密码，得flag
```

WriteUp

```
binwalk发现隐藏数据 ： 一个Rar(箭头所指为隐藏数据的偏移)
```

```
root@kali:~# binwalk base.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
88            0x58            TIFF image data, big-endian, offset of first image
 directory: 8
38906         0x97FA          RAR archive data, first volume type: MAIN_HEAD

root@kali:~#
```

分离数据有两张方法

1. binwalk直接分离
2. HxD选择数据块分离

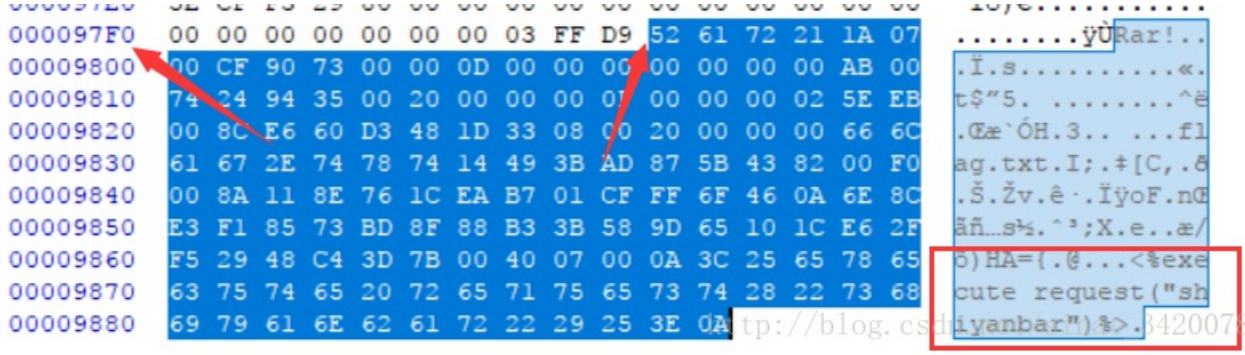
这里采用第二种方法，因为分离出来的数据还有隐藏数据

binwalk 直接分离

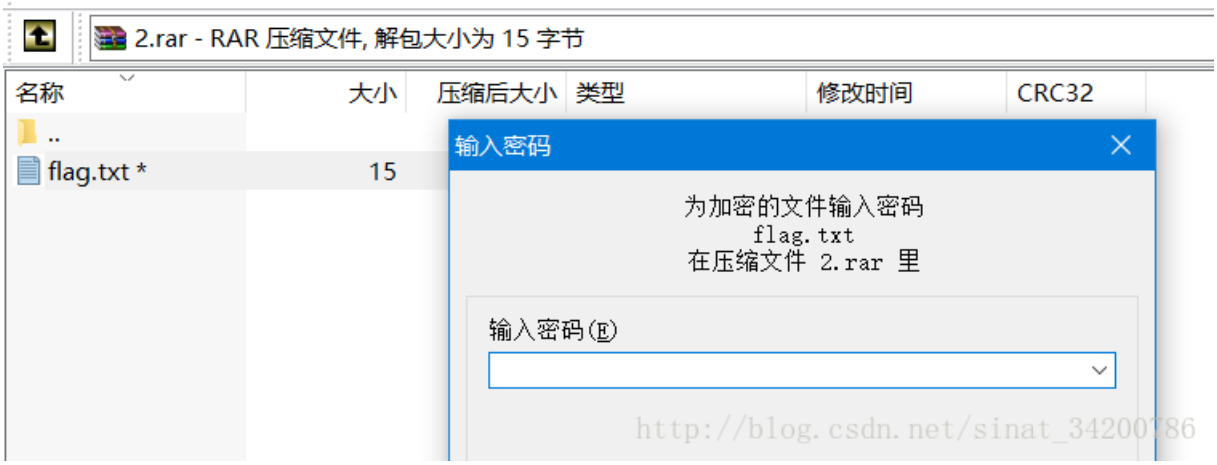
```
root@kali:~# binwalk -e base.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
88            0x58            TIFF image data, big-endian, offset of first image
 directory: 8
```


HxD分离数据(红框处为另一个隐藏数据)



新建->另存->改后缀，打开Rar却发现要密码



联想到图片名为base.jpg，那么应该涉及base64编码，将第二个隐藏数据编码即得密码

加密/解密 散列/哈希 BASE64 图片/BASE64转换

明文:

shiyanbar

BASE64编码 >

< BASE64解码

BASE64:

c2hpeWFuYmFy

解压后直接得flag

flag.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

CTF{simpleware}

水果

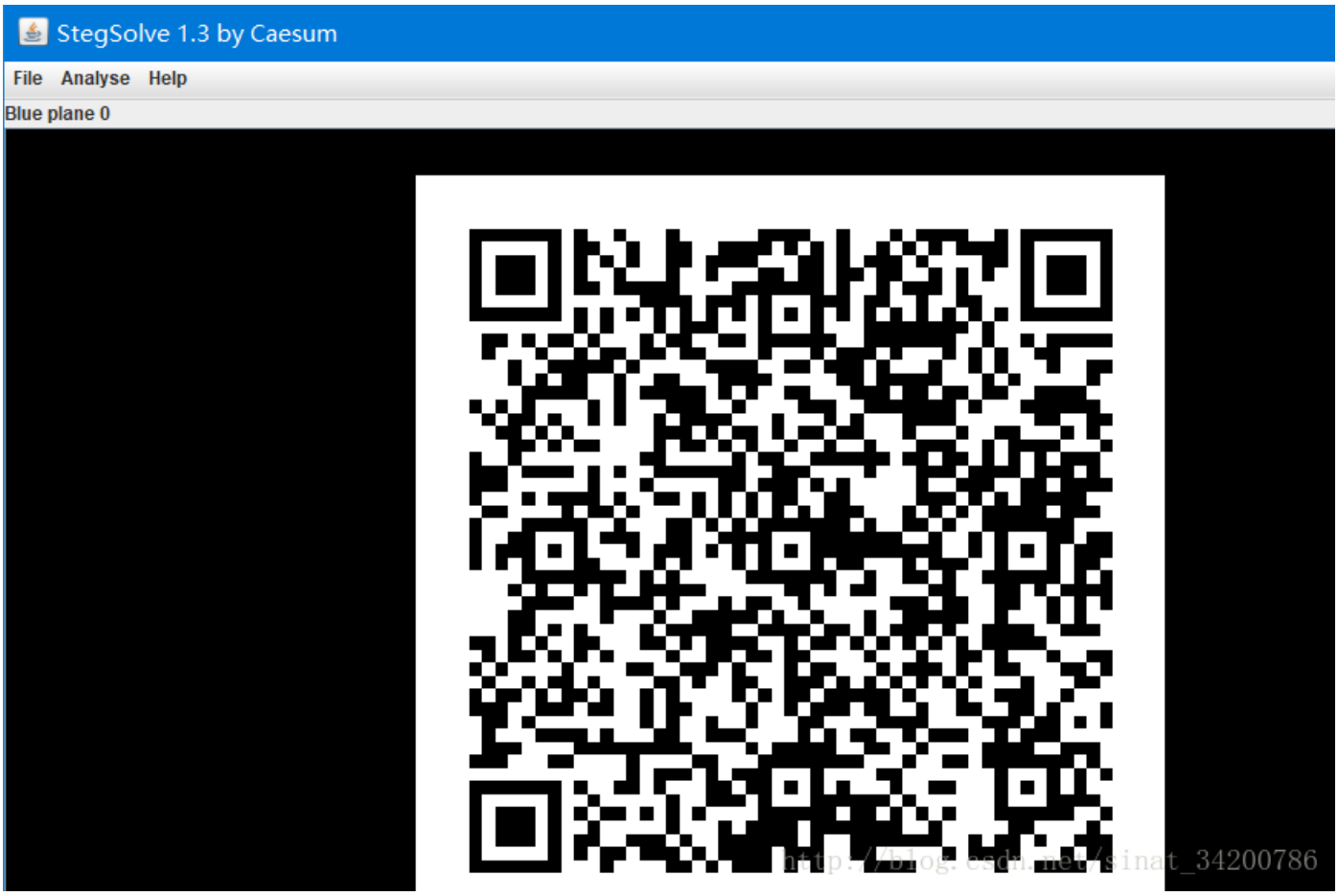
原题

解题思路

Stegsolve发现隐藏二维码，扫码得出的数据很像某种密码？

WriteUp

Stegsolve打开图片，左右看看，发现一张二维码



扫码后发现一串数字，对比ASCii发现这串数字可能是摩斯电码

用Python处理成莫斯电码的形式，解码即可

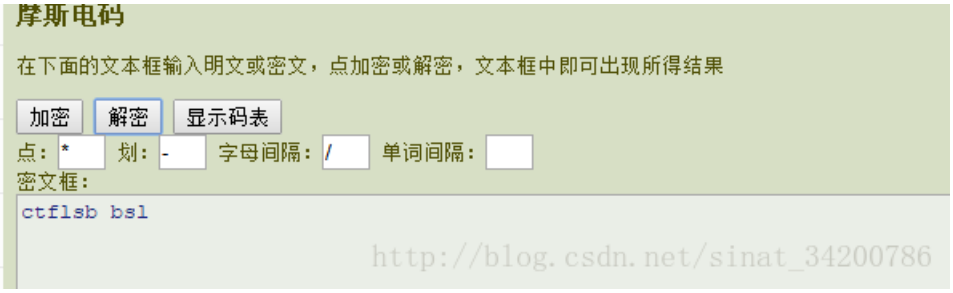
```
lst = [45, 46, 45, 46 ,32 ,45 ,32, 46, 46, 45, 46 ,32, 46, 45,
       46 ,46 ,32, 46 ,46 ,46, 32 ,45, 46, 46 ,46 ,32, 46, 46,
       45, 45, 46 ,45 ,32, 45 ,46, 46, 46, 32, 46, 46 ,46, 32,
       46 ,45, 46 ,46 ,32]

ss = ''
for n in lst:
if n==46:        #此处转换是因为所用解码网站用 * 代替 .
    ss += chr(42)
else:
    ss += chr(n)

print(ss)
```

```
[redacted]\Python36-32\python.exe E:
-*-* - **-* *-** *** -*** **--*- -*** *** *-**

Process finished with exit code 0
```

无处不在的广告

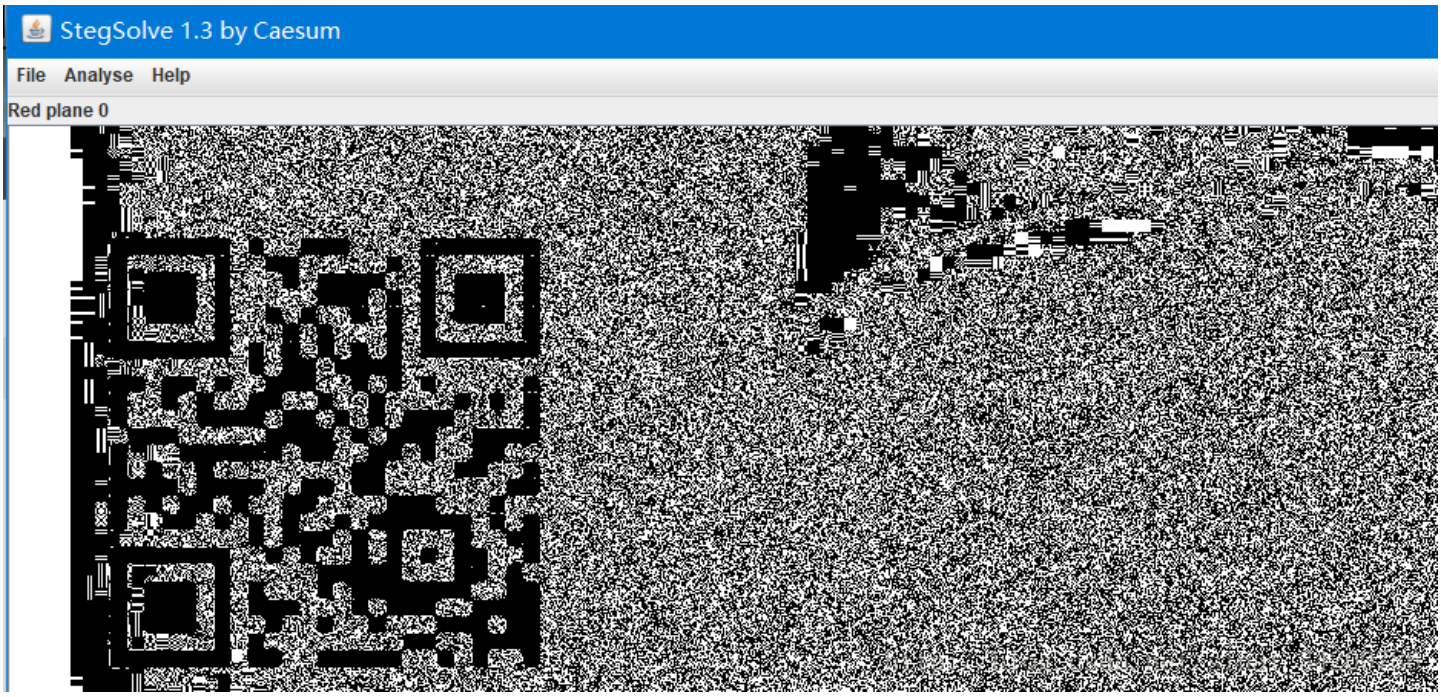
原题

解题思路

```
Stegsolve下隐藏数据无所遁形
```

WriteUp

```
Stegsolve发现隐藏的二维码
```




```
扫码送flag
```

```
----------------------------------------------------------------

已解码数据 1:

----------------------------------------------------------------
位置:(9.8,11.6)-(236.5,8.7)-(18.1,231.8)-(237.2,232.9)
颜色正常,正像
版本:2
纠错等级:L,掩码:6
内容:
FLAG:this is a new word
----------------------------------------------------------------
```

想看正面？那就要看仔细了！

原题

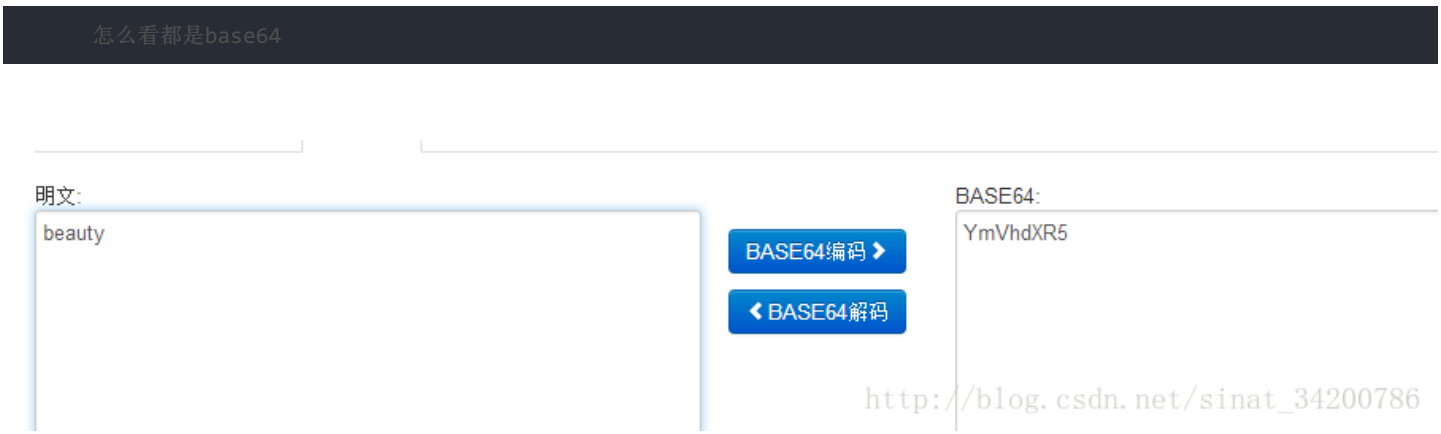
解题思路

脑洞题，不过题目也有提示“背面”

WriteUp

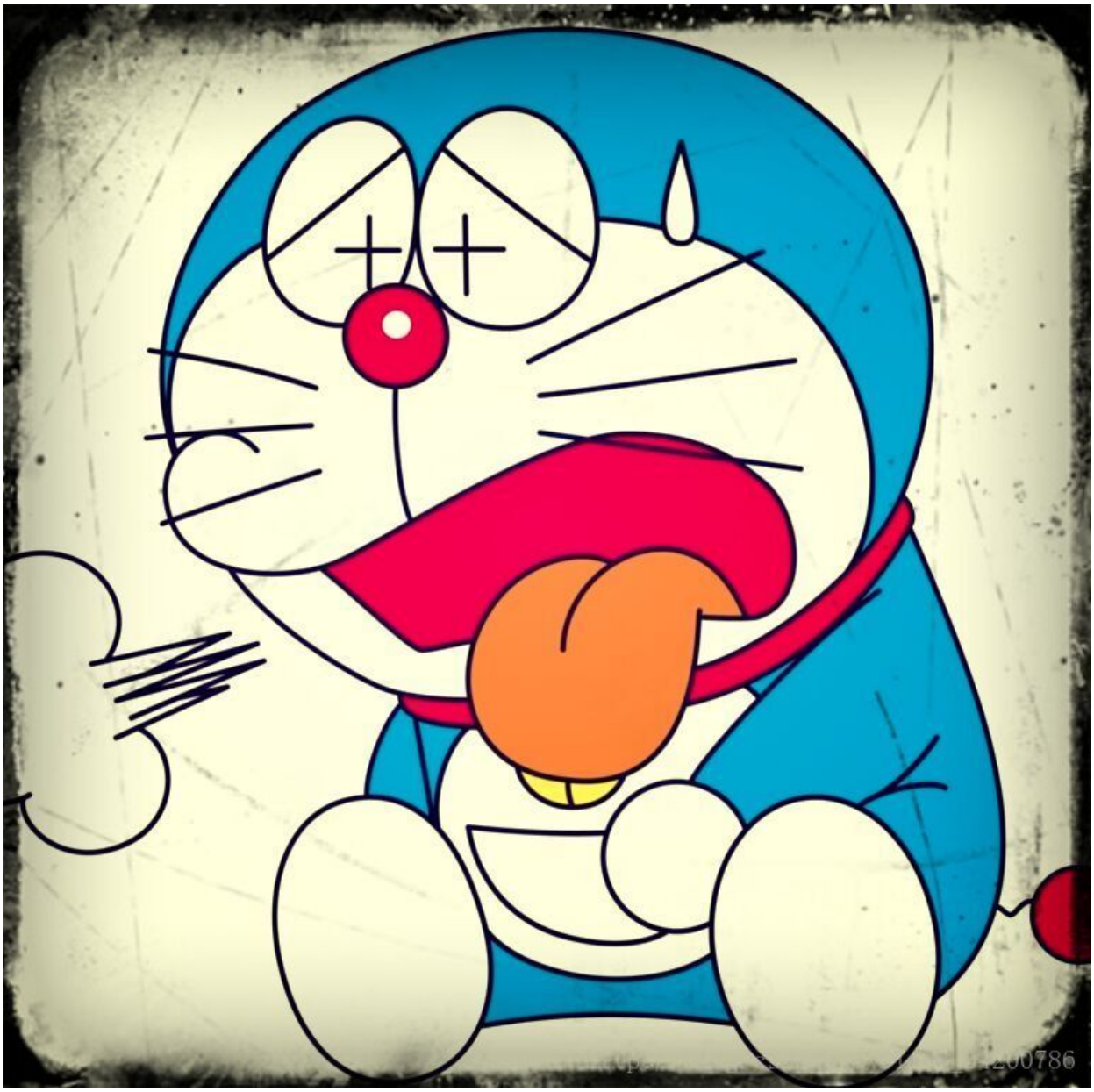
看下图片的背面


meinv.jpg 属性
常规
安全
详细信息
以前的版本
属性
值
说明
标题
想看正面？那就要看仔细了！
主题
好吧，真的被你发现了！
分级
标记
key is YmVhdXR5
备注
key is YmVhdXR5
来源
作者
http://blog.csdn.net/sinat_34200786


怎么看都是base64
明文:
beauty
BASE64编码
BASE64解码
BASE64:
YmVhdXR5
http://blog.csdn.net/sinat_34200786

多啦a梦

原题



解题思路

```
binwalk发现第二张图片,分离图片,得flag
```

WriteUp

```
root@kali:~# binwalk ameng.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
88665         0x15A59         JPEG image data, JFIF standard 1.01
```



```
HxD分离数据，新建->另存->改后缀->得flag
```

KEY:SimCTF{ctfstega}

打不开的文件

原题

解题思路

直接打开链接发现图片无法显示而且无法另存，一开始还以为是故意给一张不存在的图片，要自己抓包分析正确图片的url。其实只要

WriteUp

下载途径很多，我选的是百度云离线

无法直接打开，HxD发现是文件头残缺，补全即可

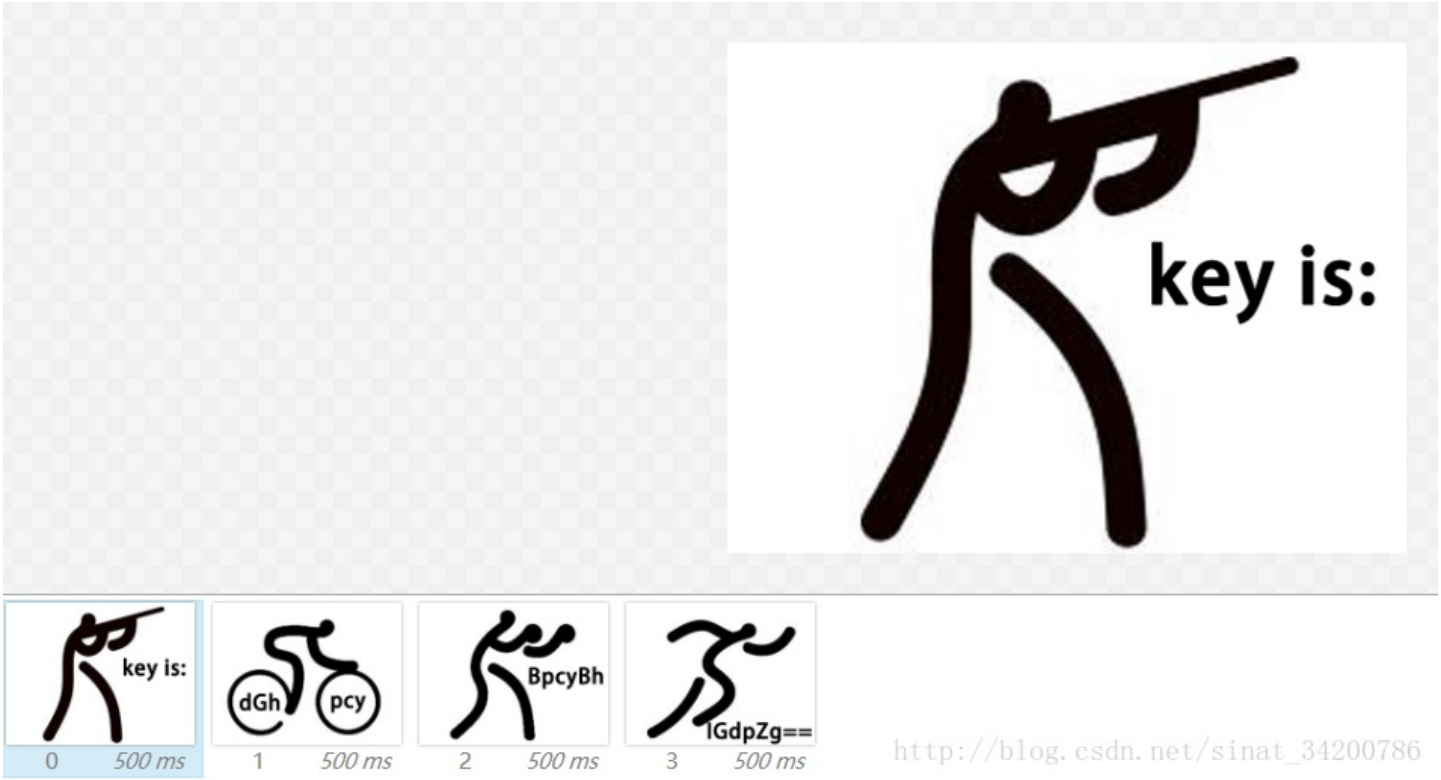
xx.gif

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000   39 61 C2 01 52 01 F7 FF 00 ED ED EC F7 F7 F7 9C  9aÂ.R.÷ÿ.ííì÷÷÷œ
00000010   9C 9C 8C 8C 8C 03 03 03 A3 A3 A3 3F 3F 3F 5C 5B  œœŒŒŒ...£££???\[
00000020   5B 6C 6C 6C E9 E9 E9 93 93 93 F9 F9 F6 DE DE DD  [lllééé“““ùùöÞÞÝ
00000030   12 01 02 F4 F4 F4 74 74 74 FA F5 F4 83 83 83 FE  ...ôôôtttúõôfffþ
```

```
(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
)0   47 49 46 38 39 61 C2 01 52 01 F7 FF 00 ED ED EC  GIF89aÂ.R.÷ÿ.í
10   F7 F7 F7 9C 9C 9C 8C 8C 8C 03 03 03 A3 A3 A3 3F  ÷÷÷œœœŒŒŒ...£££
20   3F 3F 5C 5B 5B 6C 6C 6C E9 E9 E9 93 93 93 F9 F9  ??\[[lllééé“““ù
30   F6 DE DE DD 12 01 02 F4 F4 F4 74 74 74 FA F5 F4  öÞÞÝ...ôôôtttú
```

补全后

有点快，还是用编辑器看吧



复杂的QR_code

原题

解题思路

```
扫不出有用数据，binwalk看看吧
```

WriteUp

```
直接扫，扫出无用数据
```

已解码数据 1:

位置:(10.3,10.3)-(268.4,10.3)-(10.3,268.4)-(268.7,268.7)
颜色正常,正像
版本: 2
纠错等级:H,掩码:7
内容:
secret is here

---

binwalk发现隐藏数据，分离出来看看

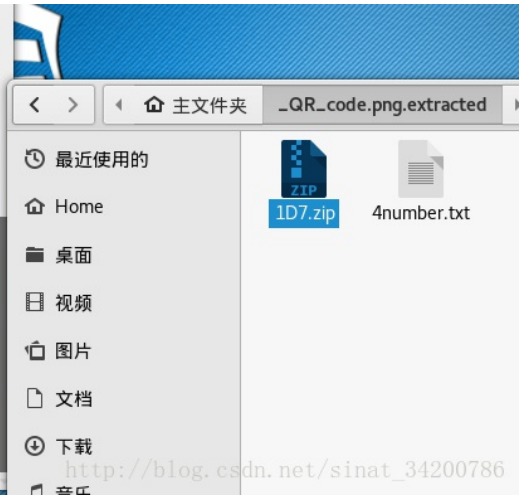
```
root@kali:~# binwalk QR_code.png

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

root@kali:~# binwalk -e QR_code.png

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

root@kali:~#
```

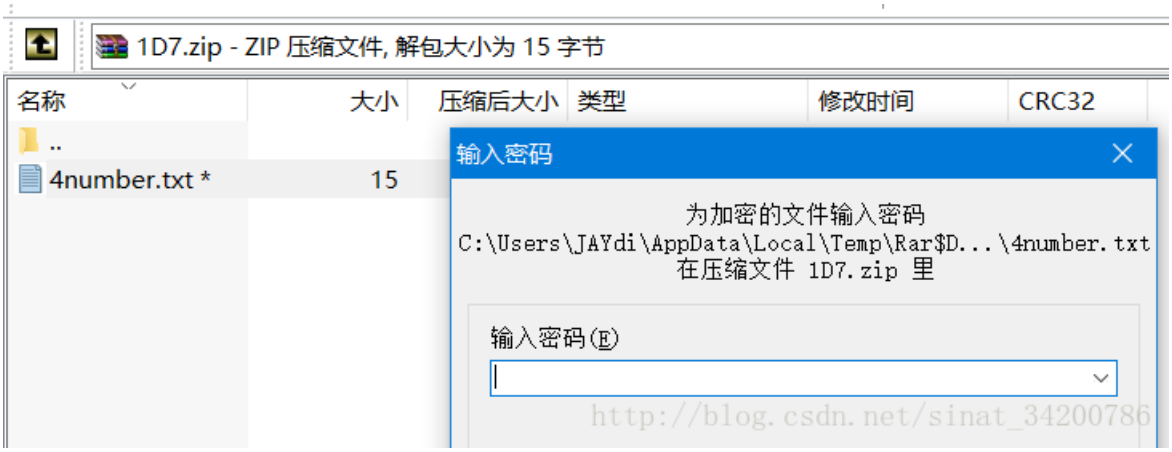


---

txt文件直接打开空白，应该是Zip有密码导致的

---

win下打开Zip发现确实有密码，不过根据提示只有4个数字，爆破它



解压后轻松得flag

刷新 刷新 快刷新

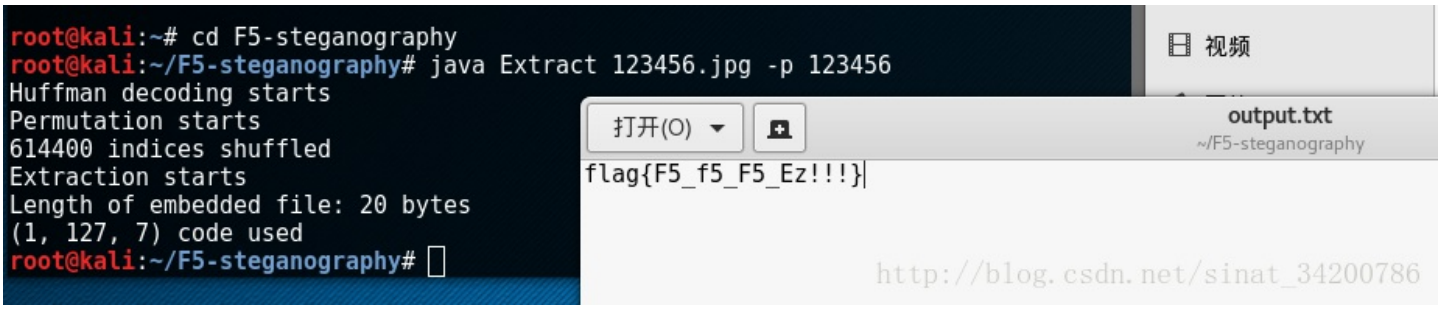
原题

解题思路

```
有一种隐写算法叫 F5
```

WriteUp

```
F5算法解密是需要密码的，密码哪里来？图片名啊
```



涨姿势点

```
F5除了是键盘的一个键位还是一种隐写算法
```

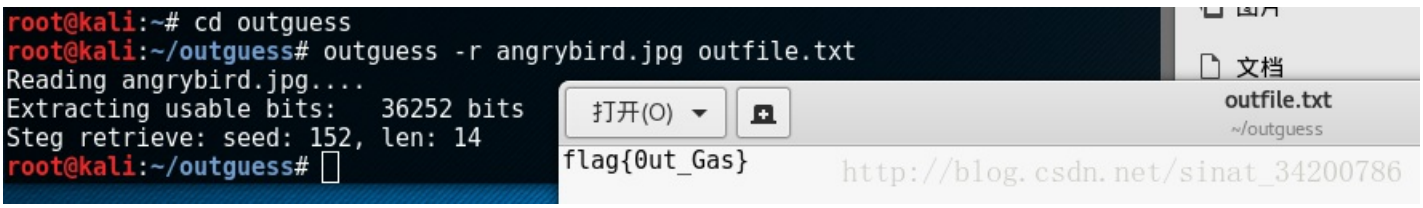
guess

原题

解题思路

```
题目名提示了一种隐写算法
```

WriteUp



涨姿势点

```
又知道了一种叫guess的隐写算法
```

小苹果

原题

解题思路

```
当铺密码？ Mp3stego?
```

WriteUp

```
扫出一串Unicode编码，解码看看
```

已解码数据 1:

位置:(199.1,90.6)-(294.4,185.6)-(104.3,185.5)-(199.3,280.3)
颜色正常,正像
版本:7
纠错等级:H,掩码:3
内容:
\u7f8a\u7531\u5927\u4e95\u592b\u5927\u4eba\u738b\u4e2d\u5de5

```
解码后发现是当铺密码，解密得到 9158753624
```

当铺密码

当前位置： 站长工具 > Unicode编码转换

Unicode编码 | UTF-8编码 | URL编码/解码 | Unix时间戳 | Ascii/Native编码互转 | base64图片在线转换工具

\u7f8a\u7531\u5927\u4e95\u592b\u5927\u4eba\u738b\u4e2d\u5de5

羊由大井夫大人王中工

用binwalk查看后发现隐藏数据，分离数据

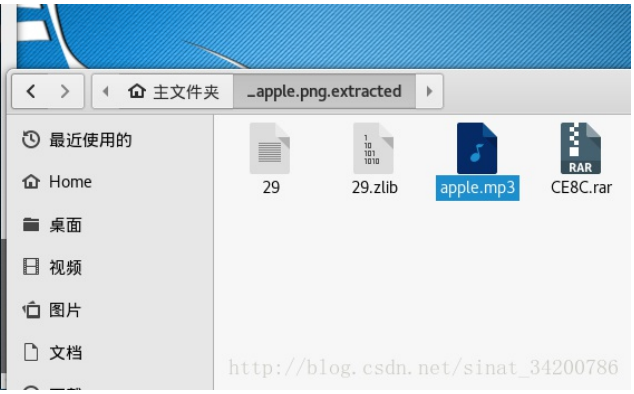
```
root@kali:~# binwalk apple.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 400 x 400, 8-bit/color RGBA, non-interl
aced
41            0x29            Zlib compressed data, compressed
52876         0xCE8C          RAR archive data, first volume type: MAIN_HEAD

root@kali:~# binwalk -e apple.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 400 x 400, 8-bit/color RGBA, non-interl
aced
41            0x29            Zlib compressed data, compressed
52876         0xCE8C          RAR archive data, first volume type: MAIN_HEAD

root@kali:~#
```



Audacity一顿操作后还是没发现异常，猜想可能是Mp3stego，试试，密码就是当铺密码解出的数据

```
MP3Stego>decode -X -P 9158753624 apple.mp3
MP3StegoEncoder 1.1.17
See README file for copyright info
Input file = 'apple.mp3'  output file = 'apple.mp3.pcm'
Will attempt to extract hidden information. Output: apple.mp3.txt
the bit stream file apple.mp3 is a BINARY file
HDR: s=FFF, id=1, l=3, ep=off, br=9, sf=0, pd=1, pr=0, m=0, js=0, c=0, o=0, e=0
alg.=MPEG-1, layer=III, tot bitrate=128, sfrq=44.1
mode=stereo, sblim=32, jsbd=32, ch=2
[Frame 1213]Avg slots/frame = 417.617; b/smp = 2.90; br = 127.895 kbps
Decoding of "apple.mp3" is finished
The decoded PCM output file name is "apple.mp3.pcm"
```


解出的数据应该是base64加密,试试

文字加密解密 | MD5加密/解密 | URL加密 | JS加/解密 | JS混淆加密压缩 | ESCAPE加/解密 | BASE64 | 散列/哈希 | 迅雷，快车

CTF{xiao_ping_guo}

Q1RGe3hpYW9fcGluZ19ndW99

涨姿势点

```
知道了当铺密码的存在，还有Mp3stego这种音频隐写算法
```

男神一般都很低调很低调的！！

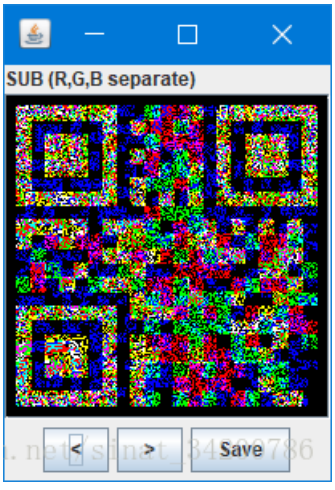
原题

解题思路

```
两张图片？当然是Stegsolve的 Image combine了
```

WriteUp

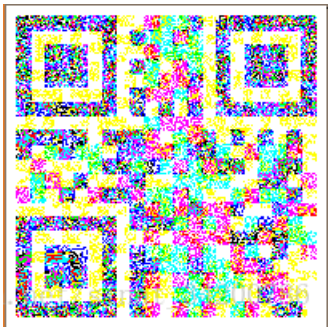
```
Stegosolve Image combine一下看看，发现二维码
```



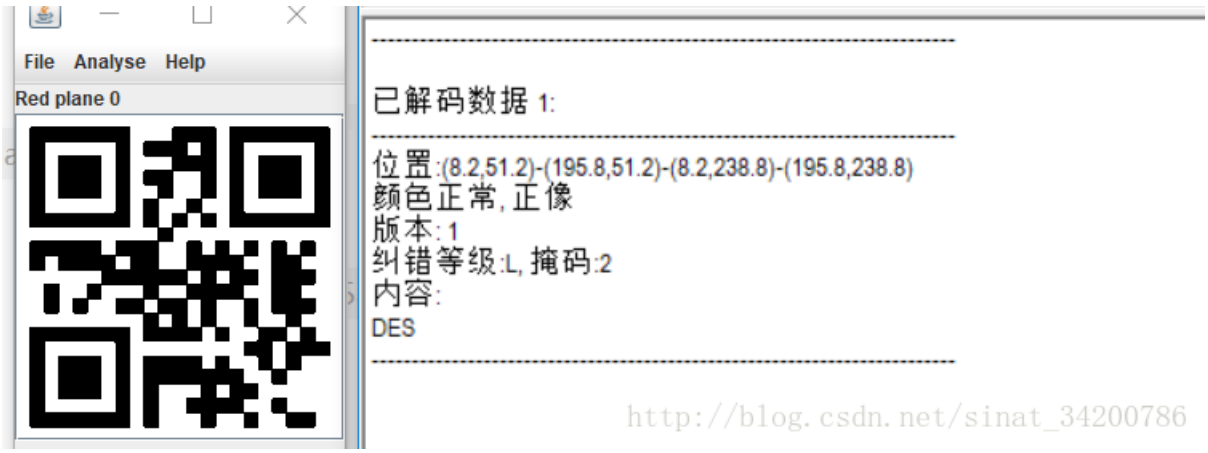
不过扫不出来，有两种处理方法

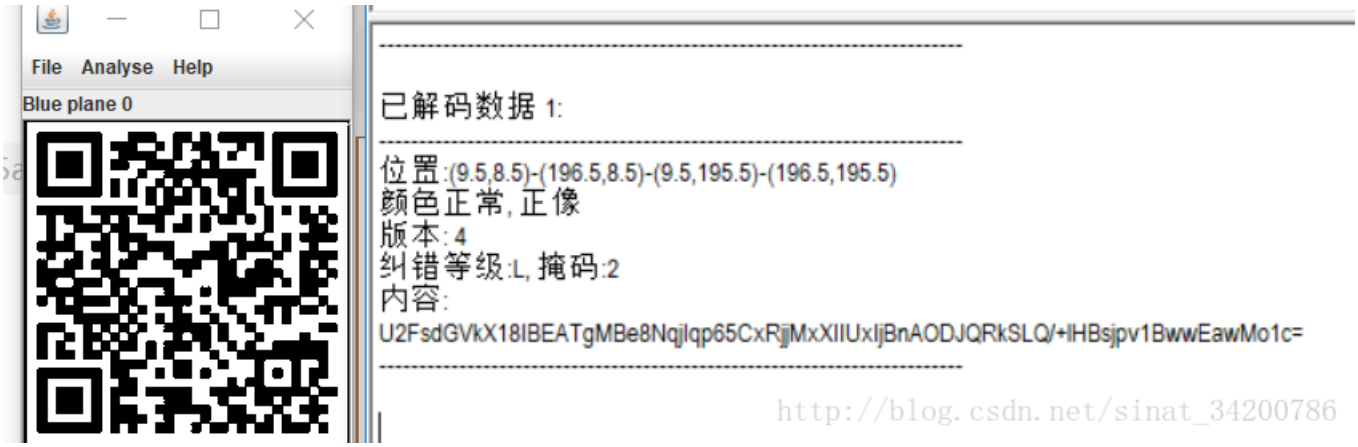
1. 反色后stegosolve查看并扫描二维码
2. ps处理通道后扫码二维码

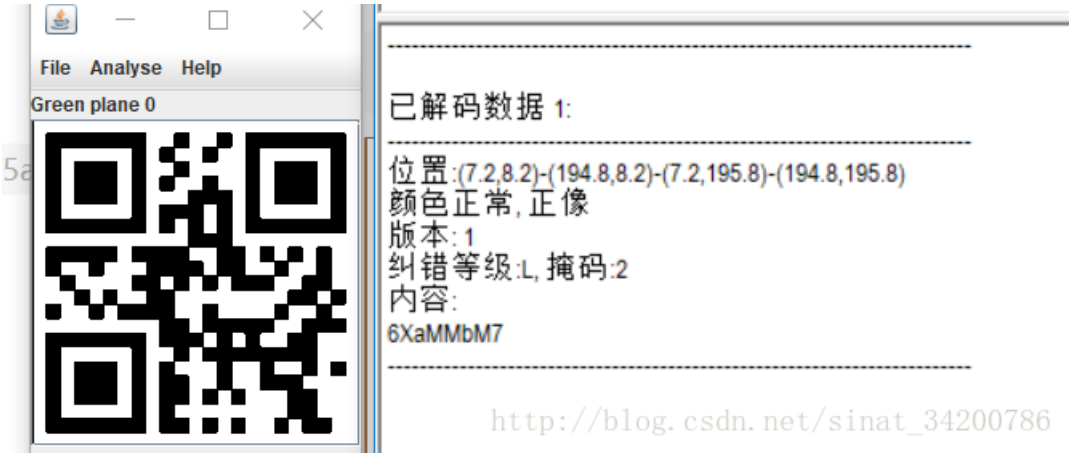
第一种：先反色，用win自带画图即可，shirt + i



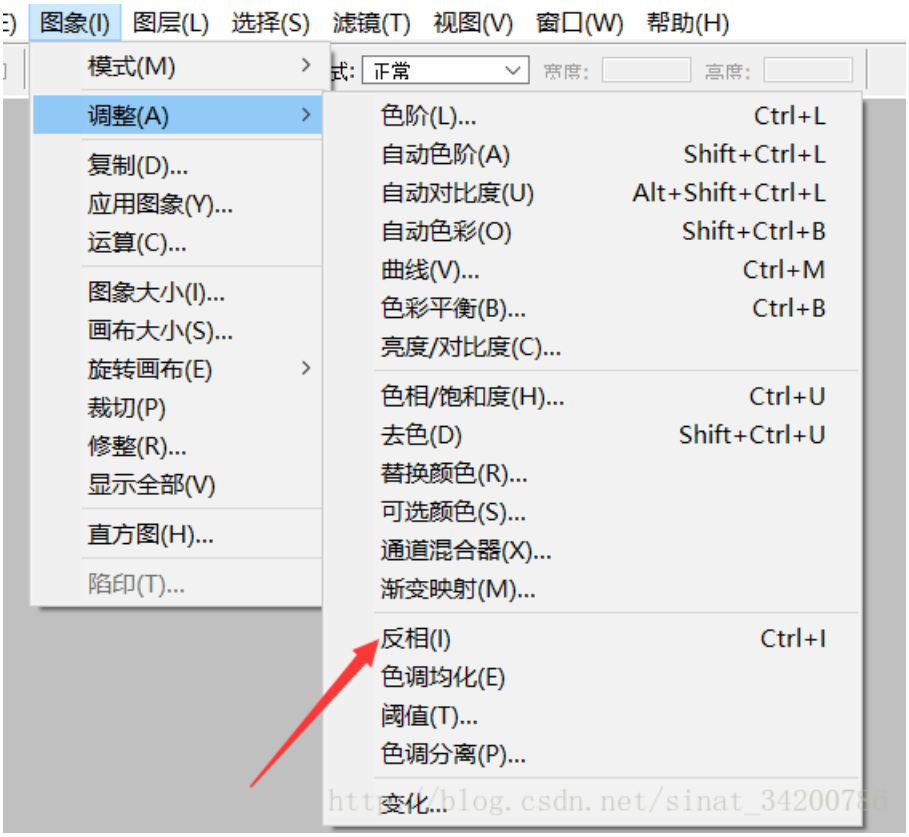
第一种：这时Stegosolve可以查看到三张二维码

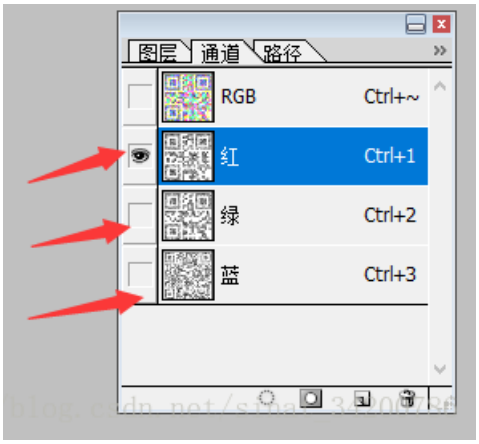


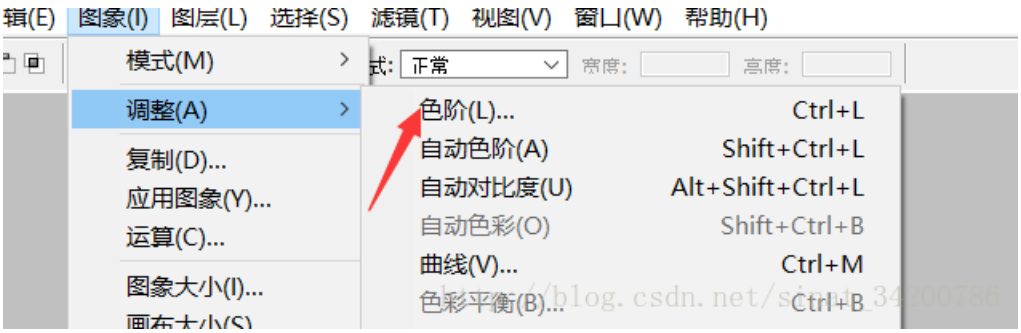


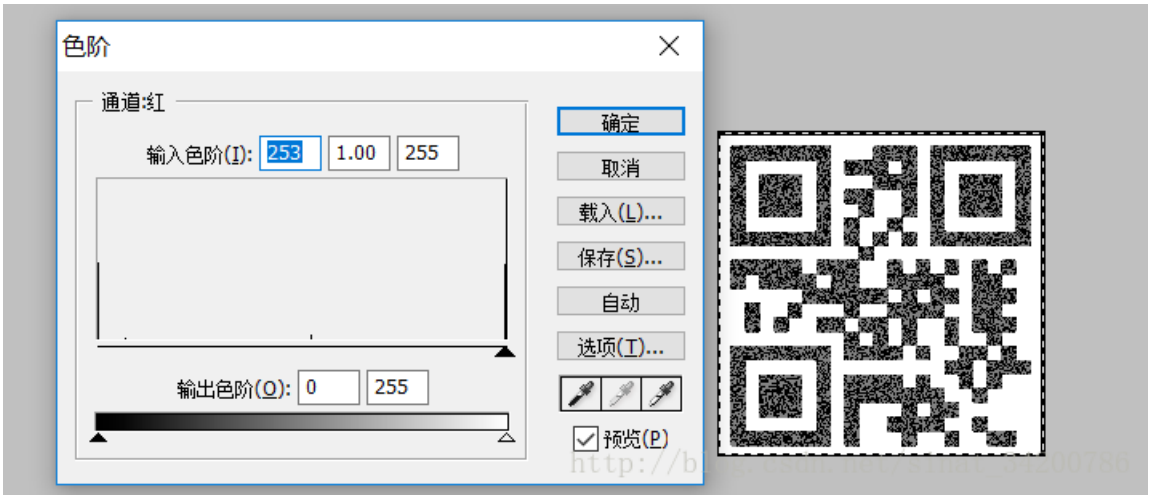


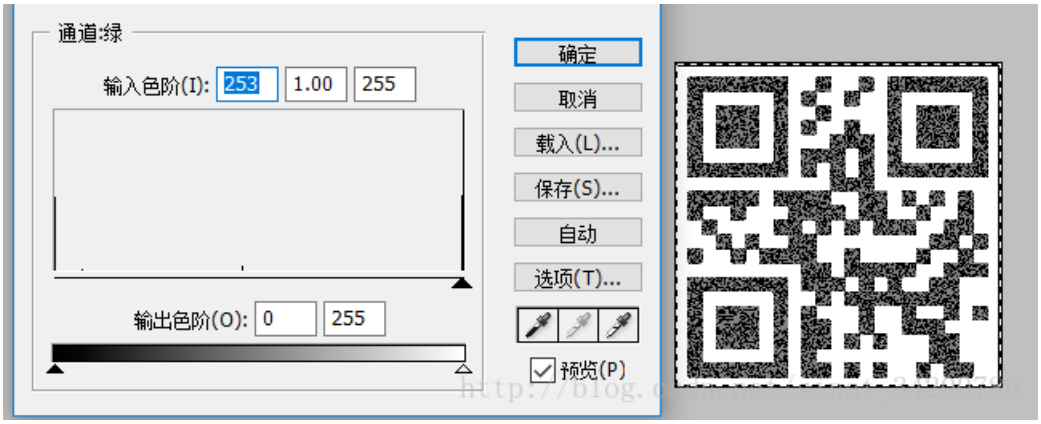
第二种：扔进ps里，反相，每个通道分别调色阶即可

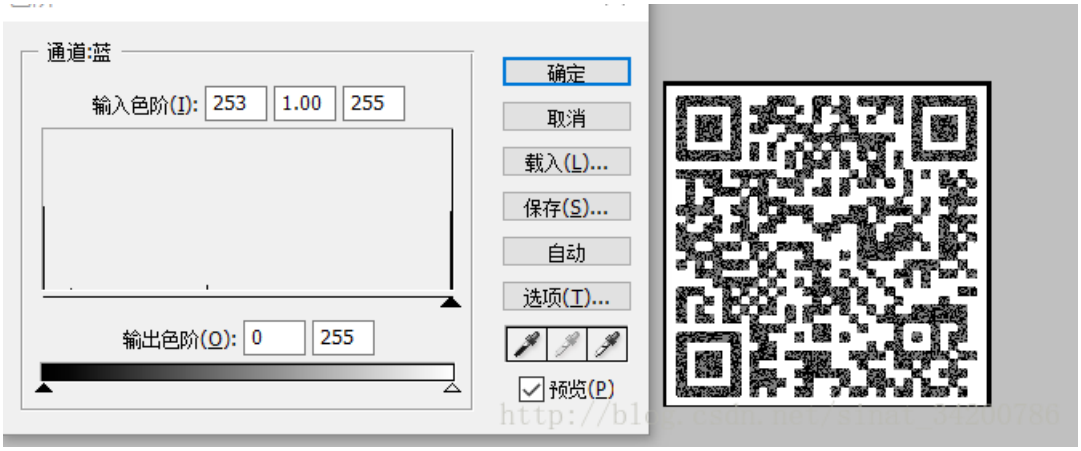



图层
通道
路径
RGB Ctrl+~
红 Ctrl+1
绿 Ctrl+2
蓝 Ctrl+3


辑(E) 图象(I) 图层(L) 选择(S) 滤镜(T) 视图(V) 窗口(W) 帮助(H)
模式(M)
调整(A)
复制(D)...
应用图象(Y)...
运算(C)...
图象大小(I)...
式: 正常
宽度:
高度:
色阶(L)... Ctrl+L
自动色阶(A) Shift+Ctrl+L
自动对比度(U) Alt+Shift+Ctrl+L
自动色彩(O) Shift+Ctrl+B
曲线(V)... Ctrl+M
色彩平衡(B)... Ctrl+B


色阶
通道:红
输入色阶(I): 253 1.00 255
输出色阶(O): 0 255
确定
取消
载入(L)...
保存(S)...
自动
选项(T)...
预览(P)


通道:绿
输入色阶(I): 253 1.00 255
输出色阶(O): 0 255
确定
取消
载入(L)...
保存(S)...
自动
选项(T)...
预览(P)



两种方法都可以得到清晰的二维码，扫码数据分别为

1. DES
2. 6XaMMbM7
3. U2FsdGVkX18IBEATgMBe8NqjIqp65CxRjjMxXIIUxIjBnAODJQRkSLQ/+lHBsjpv1BwwEawMo1c=

那么 很明显就是DES 加密了，第二个是密钥，第三个是密文

涨姿势点

了解了DES加解密，顺便学了一波Ps操作哈哈

最低位的亲吻

原题

解题思路

题目名也算是提示了吧，处理最低位就好了

WreteUp

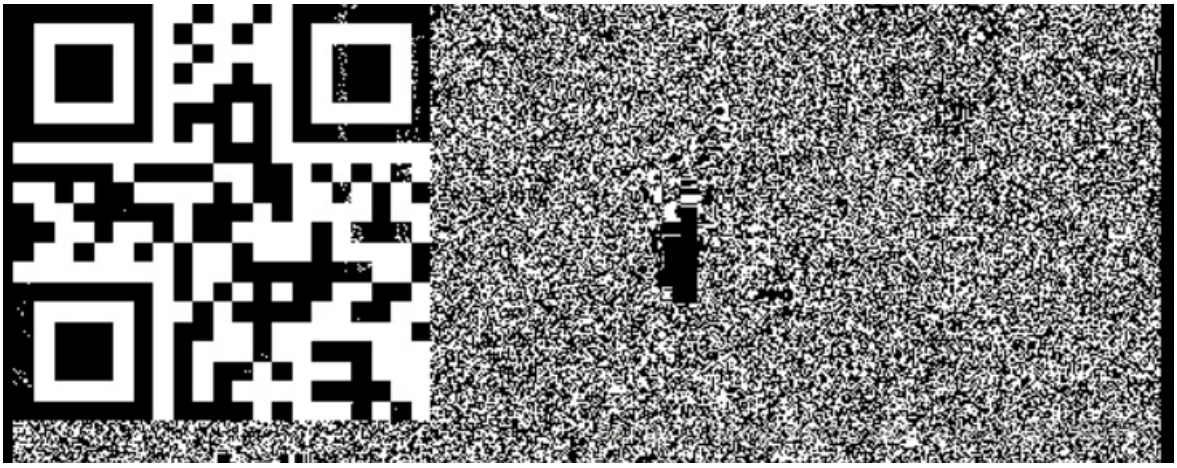
```
from PIL import Image        #引用自实验吧@pcat

def foo():
    im=Image.open('D://01.bmp')
    im2=im.copy()

    pix=im2.load()
    width,height=im2.size

    for x in range(0,width):
        for y in range(0,height):
            #LSB
            if pix[x,y]&0x1==0:
                pix[x,y]=0 #黑
            else:
                pix[x,y]=255
    im2.show()

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



已解码数据 1:

位置:(11.3,-3.6)-(271.8,-2.2)-(9.1,257.7)-(268.8,258.0)
颜色正常, 正像
版本: 1
纠错等级:L, 掩码:2
内容:
i love u

涨姿势点

通过图片像素点的最低有效位进行数据隐藏