XCTF 攻防世界 MISC杂项 高手进阶区

原创

天问_Herbert555 于 2020-02-05 09:24:24 发布 4488 收藏 6

分类专栏： # 各平台题目

https://blog.csdn.net/qq_44657899

本文链接：https://blog.csdn.net/qq_44657899/article/details/104165872

各平台题目 专栏收录该内容

45 篇文章 0 订阅

订阅专栏

文章目录

hit-the-core（找规律）

日期：2020/02/09

这道题虽然很简单但是弄python的环境弄了很久。因为我写的脚本要用到isupper函数，而curses 库不支持 windows，所以要下载whl 包。试了很多个才找到适合我的版本。`pip install curses-2.2.1+utf8-cp27-cp27m-win_amd64.whl`

```
C:\Users\DELL> pip install curses-2.2.1+utf8-cp27-cp27m-win32.whl
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7
s no longer maintained. A future version of pip will drop support for Python 2.7. More details about Python 2 supp
t in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support
WARNING: Requirement 'curses-2.2.1+utf8-cp27-cp27m-win32.whl' looks like a filename, but the file does not exist
ERROR: curses-2.2.1+utf8-cp27-cp27m-win32.whl is not a supported wheel on this platform.

C:\Users\DELL> pip install curses-2.2.1+utf8-cp27-cp27m-win_amd64.whl
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7
s no longer maintained. A future version of pip will drop support for Python 2.7. More details about Python 2 supp
t in pip, can be found at https://pip.pypa.io/en/latest/development/release-process/#python-2-support
WARNING: Requirement 'curses-2.2.1+utf8-cp27-cp27m-win_amd64.whl' looks like a filename, but the file does not exi

Requirement already satisfied: curses==2.2.1+utf8 from file:///C:/Users/DELL/curses-2.2.1%2Butf8-cp27-cp27m-win_am
4.whl in c:\python27\lib\site-packages (2.2.1+utf8)
```


首先在winhex中找到了这段字符串，现在要把flag筛选出来。

```
00      óÃ  Hfì HfÄ Ã
76                    cv
62  qAeqacLtqazEigwiXob
62  xrCrtuiTzahfFreqc{b
50  njrKwgk83kgd43j85eP
6F  gb_e_rwqr7fvbmHjklo
6E  3tews_hmkogooyf0vbn
74  k0ii87Drfgh_n kiwut
68  fb0ghk9ro987k5tfb_h
00  jiouo087ptfcv} %s
01                zR
00  x           $
```

最开始以为是提取大写字母：

```
import re
import sys
from curses.ascii import isupper
a='cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrKwgk83kgd43j85ePgb_e_rwqr7fvbmHjklo3tews_hmkogooyf0vbnk0ii87Drfgh_
n kiwutfb0ghk9ro987k5tfb_hjiouo087ptfcv}'
b=re.findall('.{1}',a)
flag=' '
for i in b:
    if(i.istitle()==True):
        flag=flag+i
    if(i=='{' or i=='}'):
        flag=flag+i
print flag
# ALEXCTF{KPHD}
```

后来才知道是从第三个字符开始，每隔四个提取一个，得到正确flag:

```
import re
import sys
from curses.ascii import isupper
a='cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrKwgk83kgd43j85ePgb_e_rwqr7fvbmHjklo3tews_hmkogooyf0vbnk0ii87Drfgh_
n kiwutfb0ghk9ro987k5tfb_hjiouo087ptfcv}'
flag=' '
for i in range(3,len(a),5):
    flag=flag+a[i]

print flag
#ALEXCTF{K33P_7H3_g00D_w0rk_up}
```

2017_Dating_in_Singapore（脑洞）

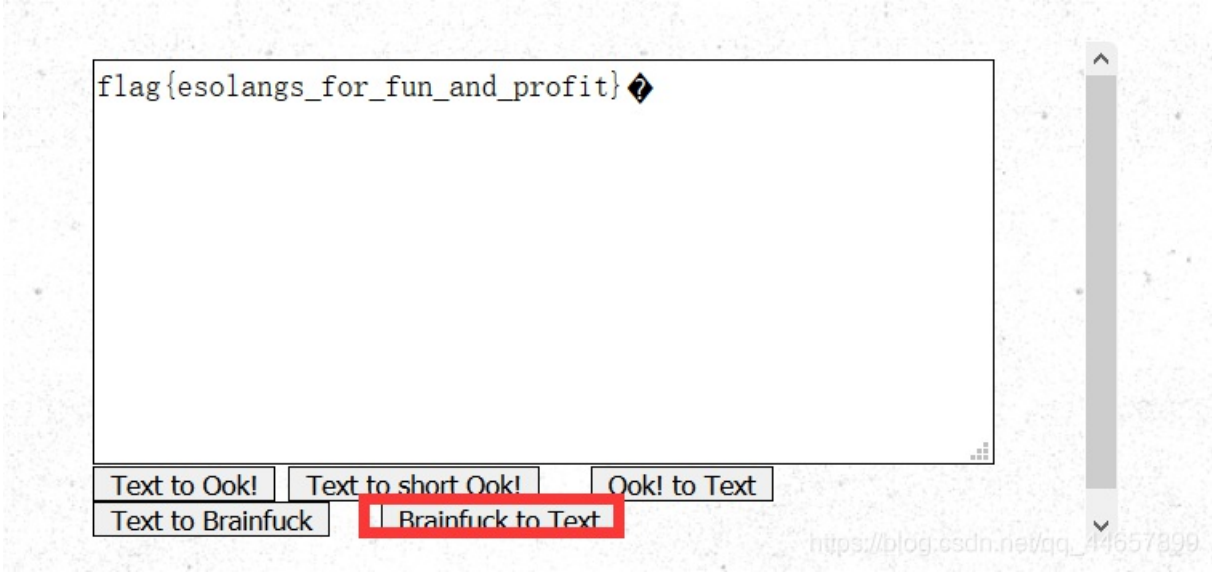
日期：2020/02/09

根据题解知道了是在日历上按数字连线得到flag。

can_has_stdio?（Brainfuck）

日期：2020/02/09

直接用winhex打开全部复制到Brainfuck/Ook!中解码得到flag。



打野（zsteg）

日期：2020/02/08

先下载zsteg，用命令`apt-get install zsteg`下载失败，百度了一下，还可以用`gem install zsteg`下载。

然后直接`zsteg 瞅啥.bmp`得到flag。

a_good_idea（将两张图片不同的像素标记出来）

日期：2020/02/08

这道题一如既往的做不出来，看了看题解之后恍然大悟，顺便学学python对图像的操作，希望下次自己能把脚本写出来。参考：Python图像库PIL的类Image及其方法介绍

PIL

图像类class image

PIL是python里标准的图像处理库，可以对显示后的图像进行多种可视化处理操作。

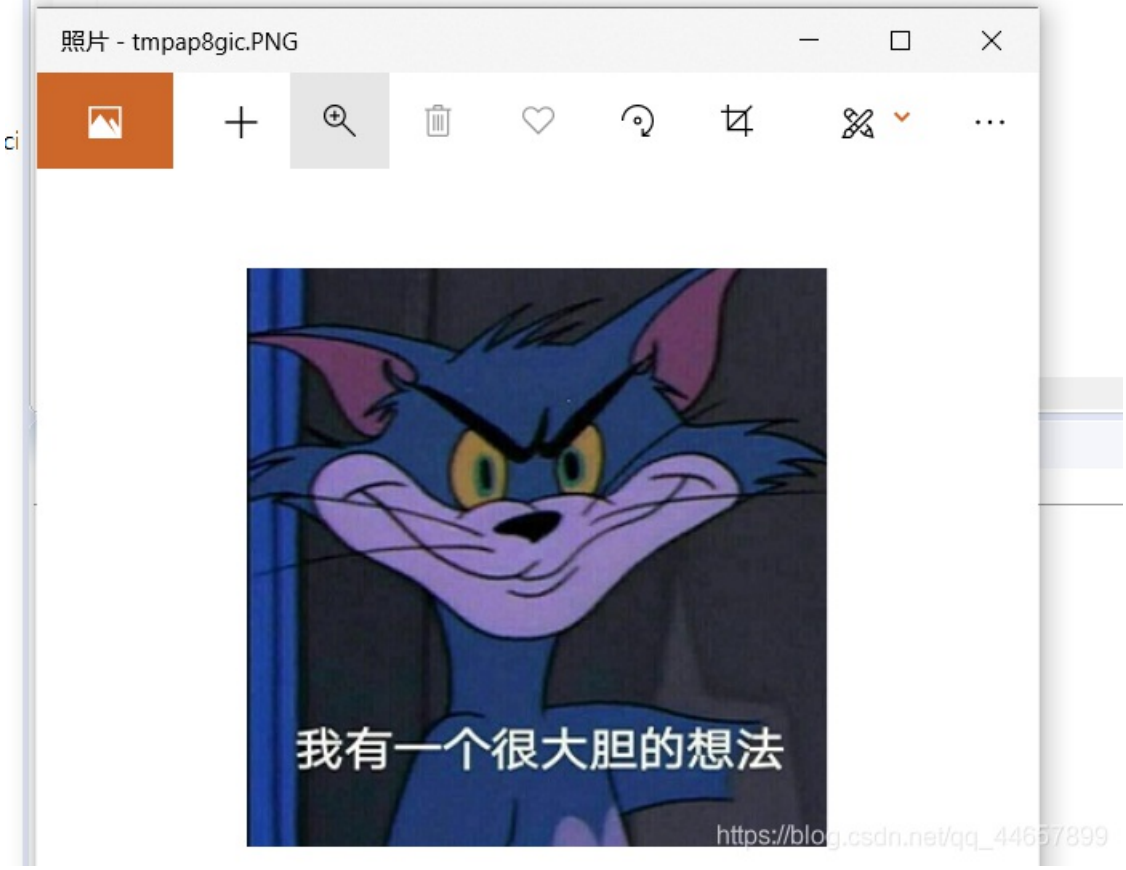
Image类是PIL中的核心类，你有很多种方式来对它进行初始化，比如从文件中加载一张图像，处理其他形式的图像，或者是从头创造一张图像等。Image模块操作的基本方法都包含于此模块内。如open、save、conver、show...等方法。下面是PIL的Image类中常用的方法和属性:

open方法

要从文件加载图像，使用 open() 函数， 在 Image 模块（类）：

```
from PIL import Image                ##调用库，包含图像类
im=Image.open('D:\\desktop\\ctf\\to_do.png')    ##文件存在的路径，如果没有路径就是当前目录下文件
im.show()
```

```
from PIL import Image
im=Image.open('D:\\desktop\\ctf\\to_do.png')
im.show()
```

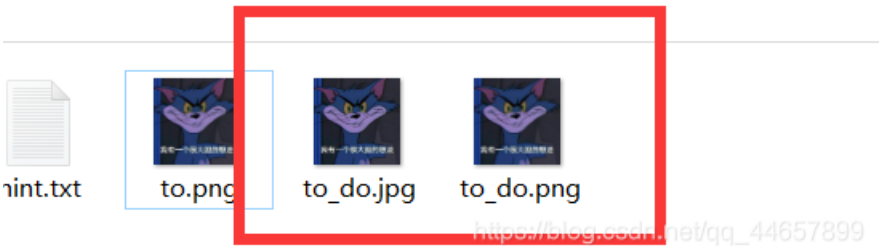


使用save方法将jpg 转换成png

用 Image 类的 save() 方法保存文件的文件，使用给定的文件名保存图像。

```
from PIL import Image
im = Image.open("D:\\desktop\\ctf\\to_do.png")
print(im)
im.save("D:\\desktop\\ctf\\to_do.jpg")      ## 将"3d.png"保存为3d.jpg"
```

```python
from PIL import Image
im = Image.open("D:\\desktop\\ctf\\to_do.png")
print(im)
im.save("D:\\desktop\\ctf\\to_do.jpg")
```



mode属性

im.mode ⇒ string

图像的模式，常见的mode 有 “L” (luminance) 表示灰度图像，“RGB”表示真彩色图像，和 “CMYK” 表示出版图像，表明图像所使用像素格式。如下表为常见的nodes描述：

modes	Description
1	1位像素，黑白图像，存成8位像素
L	8位像素，黑白
P	9位像素，使用调色板映射到任何其他模式
RGB	3*8位像素，真彩
RGBA	4*8位像素，真彩+透明通道
CMYK	4*8位像素，印刷四色模式或彩色印刷模式
YCbCr	3*8位像素，色彩视频格式
I	32位整型像素
F	33位浮点型像素

```python
from PIL import Image
im = Image.open("D:\\desktop\\ctf\\to_do.png")
print(im.mode) ## 打印模式属性
```

```
from PIL import Image
im = Image.open("D:\\desktop\\ctf\\to_do.png")
print(im.mode)
```

```
Console
<terminated> a_good_idea.py [C:\Python27\python.exe]
RGB
```


load方法

为图像分配内存并从文件中加载它（或者从源图像，对于懒操作）。正常情况下，用户不需要调用这个方法，因为在第一次访问图像时，Image类会自动地加载打开的图像。目前的版本，方法load()返回一个用于读取和修改像素的像素访问对象。这个访问对象像一个二维队列，如：

```
pix = im.load()
print pix[x, y]
pix[x, y] =value
```

size属性

im.size ⇒ (width, height)
图像的尺寸，按照像素数计算，它的返回值为宽度和高度的二元组（width, height）。

```
from PIL import Image
im = Image.open("3d.jpg")
print(im.size)
```

```
from PIL import Image
im = Image.open("D:\\desktop\\ctf\\to_do.png")
print(im.size)
```

```
Console
<terminated> a_good_idea.py [C:\Python27\python.exe]
(290, 289)
```



```
from PIL import Image
im = Image.open("D:\\desktop\\ctf\\to_do.png")
print im.size[0]
```

```
1 from PIL import Image
2 im = Image.open("D:\\desktop\\ctf\\to_do.png")
3 print im.size[0]
```

Console

```
<terminated> a_good_idea.py [C:\Python27\python.exe]
290
```



```
1 from PIL import Image
2 im = Image.open("D:\\desktop\\ctf\\to_do.png")
3 print im.size[1]
```

Console

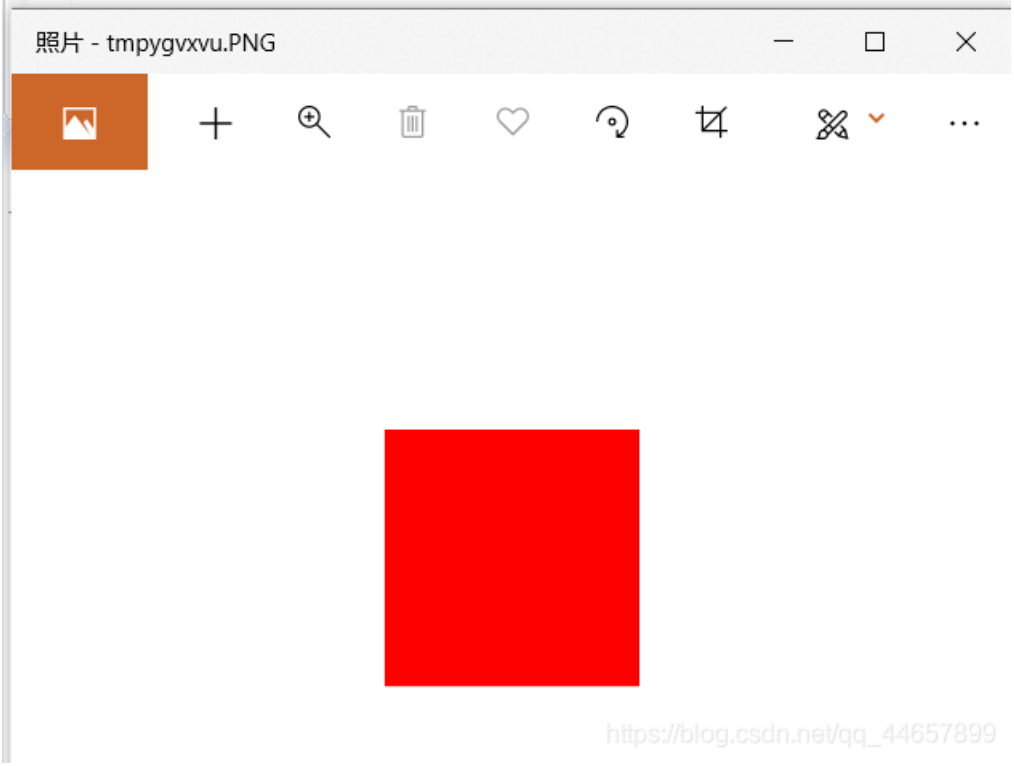
```
<terminated> a_good_idea.py [C:\Python27\python.exe]
289
```


new方法

使用给定的变量mode和size生成新的图像。Size是给定的宽/高二元组，这是按照像素数来计算的。对于单通道图像，变量color只给定一个值；对于多通道图像，变量color给定一个元组（每个通道对应一个值）。在版本1.1.4及其之后，用户也可以用颜色的名称，比如给变量color赋值为“red”。如果没有对变量color赋值，图像内容将会被全部赋值为0（为黑色）。如果变量color是空，图像将不会被初始化，即图像的内容全为0。这对向该图像复制或绘制某些内容是有用的。

```
from PIL import Image
im = Image.open("D:\\desktop\\ctf\\to_do.png")
n_im= Image.new("RGB", (128, 128), "#FF0000")
n_im.show()
```

```
from PIL import Image
im = Image.open("D:\\desktop\\ctf\\to_do.png")
n_im= Image.new("RGB", (128, 128), "#FF0000")
n_im.show()
```



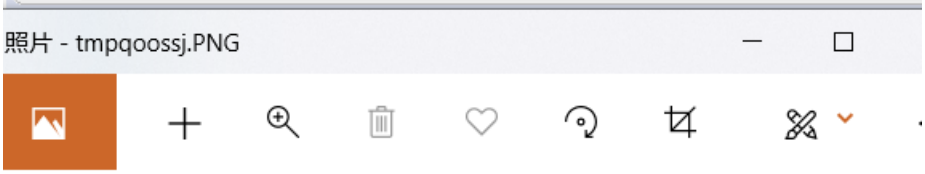
解题脚本

```
from PIL import Image
im1 = Image.open("to.png")
im2 = Image.open("to_do.png")
p1 = im1.load()
p2 = im2.load()
w = im1.size[0]
h = im1.size[1]

im = Image.new('RGB',(290,289))
p = im.load()

for i in range(w):
    cnt = 0
    for j in range(h):
        if p1[i, j] != p2[i, j]:
            p[i,j] = (255,255,255)
im.show()
```

```python
from PIL import Image
im1 = Image.open("D:\\desktop\\ctf\\to.png")
im2 = Image.open("D:\\desktop\\ctf\\to_do.png")
p1 = im1.load()
p2 = im2.load()
w = im1.size[0]
h = im1.size[1]
```



隐藏的信息（八进制转字符串）

日期：2020/02/08

得到字符串：

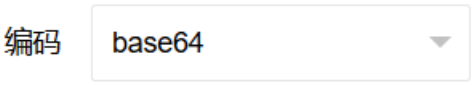
```
0126 062 0126 0163 0142 0103 0102 0153 0142 062 065 0154 0111 0121 0157 0113 0111 0105 0132 0163 0131 0127 0143
066 0111 0105 0154 0124 0121 060 0116 067 0124 0152 0102 0146 0115 0107 065 0154 0130 062 0116 0150 0142 0154 07
1 0172 0144 0104 0102 0167 0130 063 0153 0167 0144 0130 060 0113
```

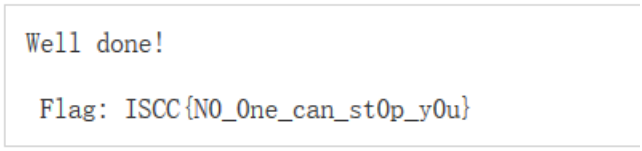
数字到7为止猜测是八进制，写个脚本转为字符串：

```
import re
import sys
s='0126 062 0126 0163 0142 0103 0102 0153 0142 062 065 0154 0111 0121 0157 0113 0111 0105 0132 0163 0131 0127 01
43 066 0111 0105 0154 0124 0121 060 0116 067 0124 0152 0102 0146 0115 0107 065 0154 0130 062 0116 0150 0142 0154
 071 0172 0144 0104 0102 0167 0130 063 0153 0167 0144 0130 060 0113 '
a=re.findall('\d{3,}',s)
for i in a:
    sys.stdout.write(chr(int(i,8)))

#V2VsbCBkb25lIQoKIEZsYWc6IElTQ0N7TjBfMG5lX2Nhbl9zdDBwX3kwdX0K
```

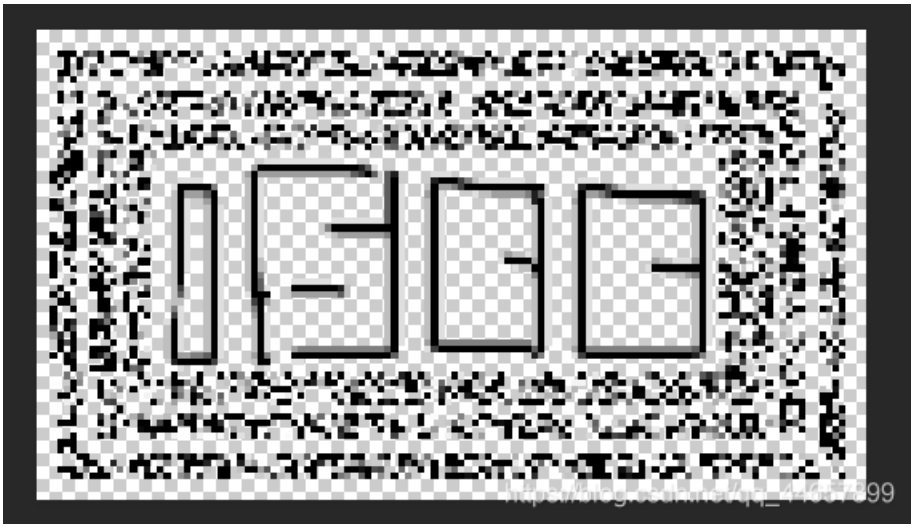
将得到的字符串base64解码得到flag。





Aesop_secret（ps图层叠加）

日期：2020/02/07



我们的秘密是绿色的（新工具oursecret）

日期：2020/02/07

一，oursecret密码为绿色数字

二，zip爆破

三，明文攻击

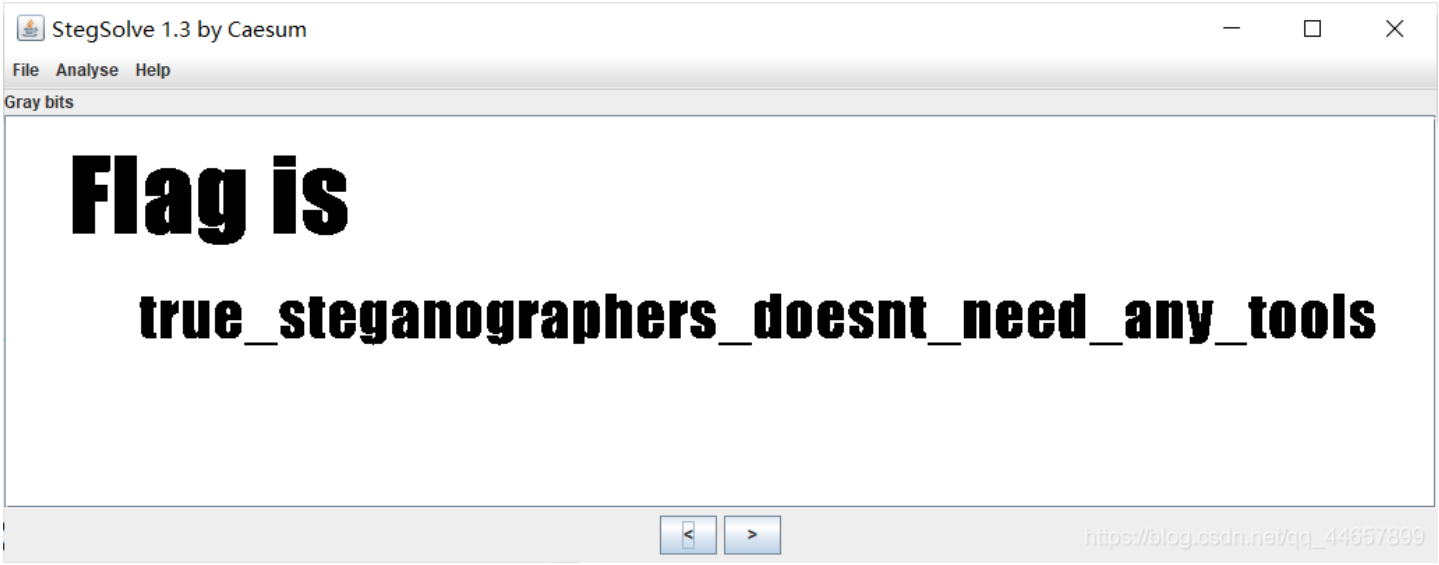
四，伪加密

五，双重加密：栅栏密码，凯撒密码

pure_color（送分）

日期：2020/02/06

Stegsolve打开点右箭头得到flag。



Reverse-it（xxd，convert命令）

日期：2020/02/04

这道题的难点就在于发现文件内容是翻转的，我还特意看了看文件头，没找到对应的文件就没有多想，下次注意点。

方法一，复制十六进制数据，翻转。

```
s = "9DFF700DB6DAFC937263282222BDD218B425D4... ....FF8DFF"

x = s[::-1]

print x
```

python中[-1]、[:-1]、[::-1]、[2::-1]的使用方法

```
a=[1,2,3.4,5]
print(a[-1]) #取最后一个元素 [5]
print(a[:-1]) # 除了最后一个取全部 [ 1 2 3 4 ]
print(a[::-1]) # 取从后向前（相反）的元素 [ 5 4 3 2 1 ]
print(a[2::-1]) #取从下标为2的元素翻转读取 [ 3 2 1 ]
print(a[::-2]) #[5,3,1]
```

将翻转数据保存到winhex中，保存。

0da9641b7aad4efb8f7eb45f... | noname

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	01	00	48	ÿØÿà JFIF H
00000010	00	48	00	00	FF	E1	00	D2	45	78	69	66	00	00	4D	4D	H ÿá ÒExif MM
00000020	00	2A	00	00	00	08	00	07	01	12	00	03	00	00	00	01	*
00000030	00	01	00	00	01	1A	00	05	00	00	00	01	00	00	00	62	b
00000040	01	1B	00	05	00	00	00	01	00	00	00	6A	01	28	00	03	j (
00000050	00	00	00	01	00	02	00	00	01	31	00	02	00	00	00	19	1
00000060	00	00	00	72	01	32	00	02	00	00	00	14	00	00	00	8C	r 2 Œ
00000070	87	69	00	04	00	00	00	01	00	00	00	A0	00	00	00	00	‡i
00000080	00	00	00	48	00	00	00	01	00	00	00	48	00	00	00	01	H H
00000090	73	74	6E	65	6D	65	6C	45	20	70	6F	68	73	6F	74	6F	stnemelE pohsoto
000000A0	68	50	20	65	62	6F	64	41	00	46	32	30	30	31	3A	30	hP ebodA F2001:0
000000B0	30	3A	32	32	20	31	30	3A	34	31	3A	30	32	00	00	03	0:22 10:41:02
000000C0	A0	01	00	03	00	00	00	01	00	01	00	00	A0	02	00	04	

方法二，使用命令行。

```
xxd -p 文件名 | tr -d '\n' | rev | xxd -r -p > 更改后的文件名
```

1，`|`

`|` 表示管道，上一条命令的输出，作为下一条命令参数。

2，xxd命令

`-p` 以一个整块输出所有的hex， 不使用空格进行分割。

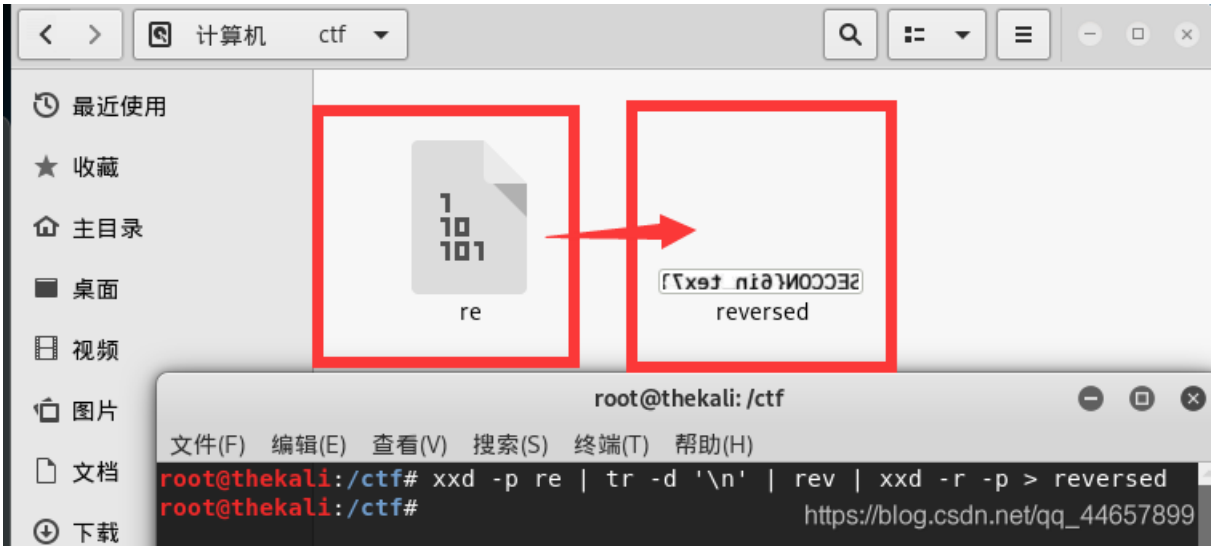
`-r` 反转操作，将16进制专程2进制

3，tr命令

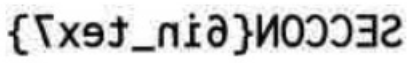
`-d '\n '` 删除字符串中所有' \n '空格。

4，rev命令

`rev` 将文件中的每行内容以字符为单位反序输出，即第一个字符最后输出，最后一个字符最先输出，依次类推。



打开是一张翻转的图片。



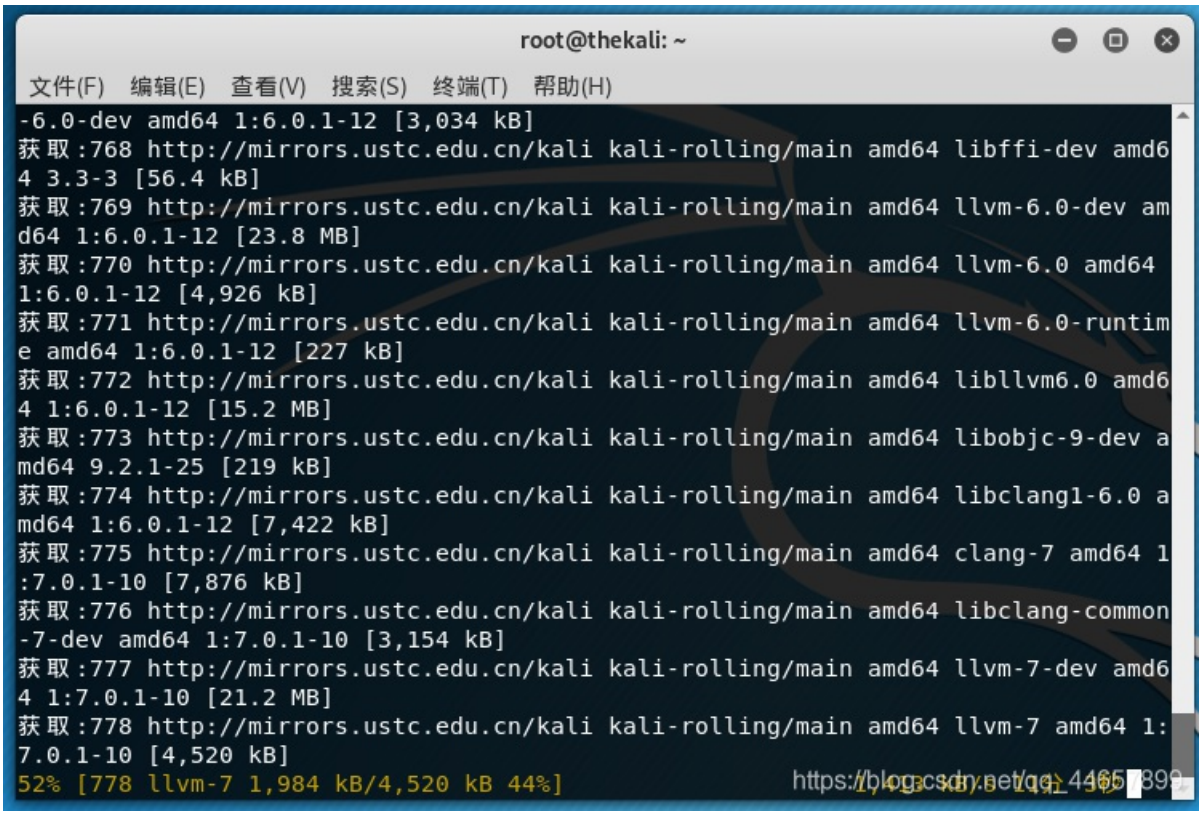
然后对称一下，使用命令：

```
convert -flop reversed reversed.jpg
```

convert命令详解：

> convert -flip 上下翻转。
> convert -flop 左右翻转。

要用convert命令还要先下载ImageMagick软件，使用命令`apt-get install ImageMagick`，但是我这里报错`E:无法定位软件包 ImageMagick`，百度了一下要在etc/apt 的sources.list 修改镜像源，这一更新更新了好久，，，
参考：https://blog.csdn.net/qq_42092076/article/details/88357387



更新之后还是没有解决问题，于是我在windows上安装一个imagemagick使用。
参考：Windows系统安装及初步使用ImageMagick

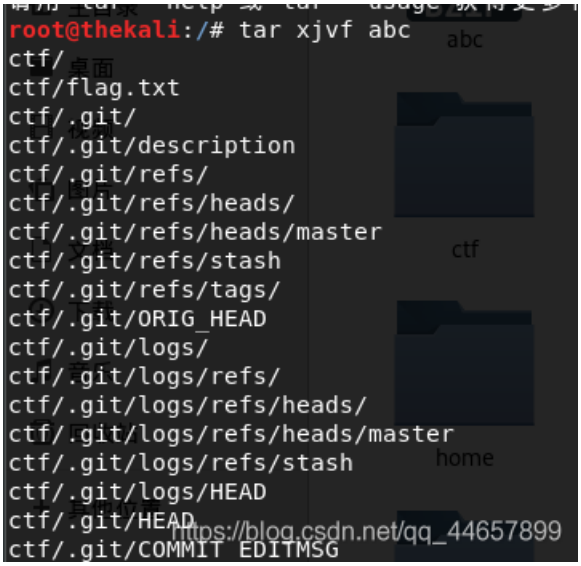
最后发现将 ImageMagick改为imagemagick小写的就可以了，，，，，，，，，晕

MISCall（git stash）

日期：2020/02/04

将文件拖进kali，更改文件名为abc，然后`file abc`查看文件类型。

然后`tar xjvf abc`解压文件。



发现是拿到了.get目录，然后根据网上的方法知道了`git stash`。

```
stash@{0}: WIP on master: bea99b9 Initial commit
root@thekali:/ctf# git stash show
 flag.txt | 25 +++++++++++++++++++++++++-
 s.py     |  4 ++++
 2 files changed, 28 insertions(+), 1 deletion(-)
root@thekali:/ctf# git stash apply
error: 您对下列文件的本地修改将被合并操作覆盖：
        flag.txt
请在合并前提交或贮藏您的修改。
终止中
root@thekali:/ctf# git reset --hard
HEAD 现在位于 bea99b9 Initial commit
root@thekali:/ctf# git stash apply
位于分支 master
要提交的变更：
  (use "git restore --staged <file>..." to unstage)
        新文件：    s.py

尚未暂存以备提交的变更：
  （使用 "git add <文件>..." 更新要提交的内容）
  (use "git restore <file>..." to discard changes in working directory)
        修改：      flag.txt

root@thekali:/ctf#                                  https://blog.csdn.net/qq_44657899
```

然后运行脚本得到falg。

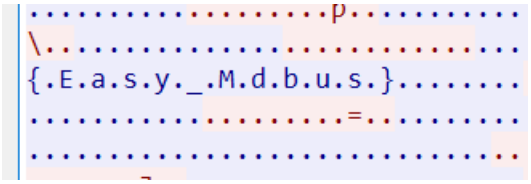
```
root@thekali:/ctf# python s.py
NCN4dd992213ae6b76f27d7340f0dde1222888df4d3
root@thekali:/ctf#
```

神奇的Modbus（送分）

日期：2020/02/04

题目说了Modbus，百度了一下是一个通讯协议。

先在包里搜索flag没有搜到，然后把Modbus协议追踪tcp流在里面找到了flag，但是还要加个o，，，



embarrass（送分）

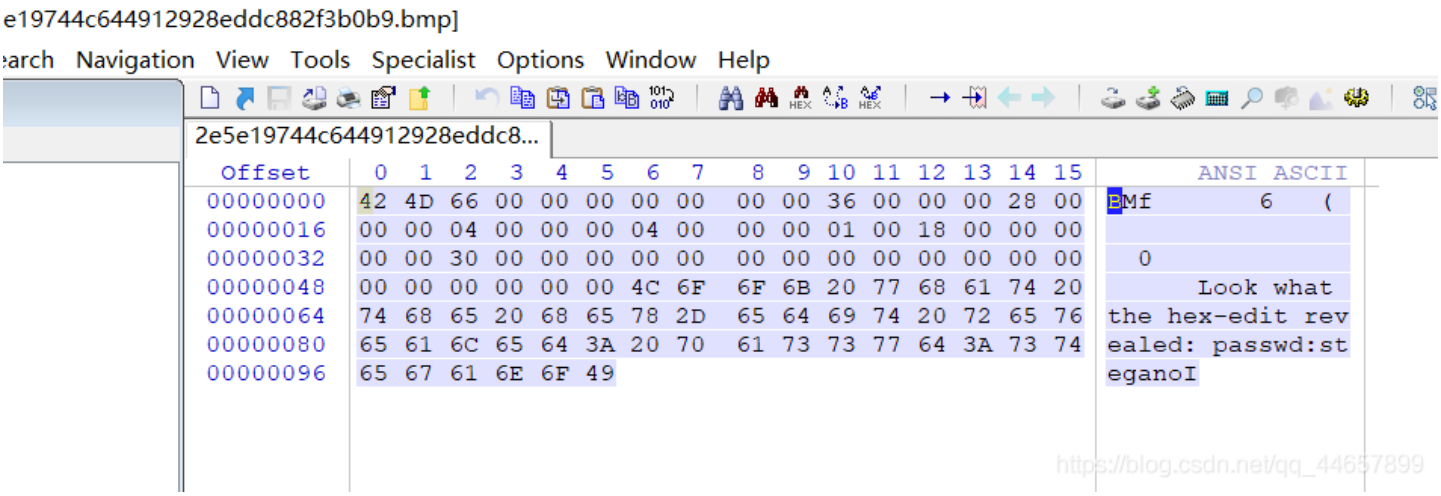
日期：2020/02/04

直接搜索字符串得到flag。

Training-Stegano-1（送分）

日期：2020/01/25

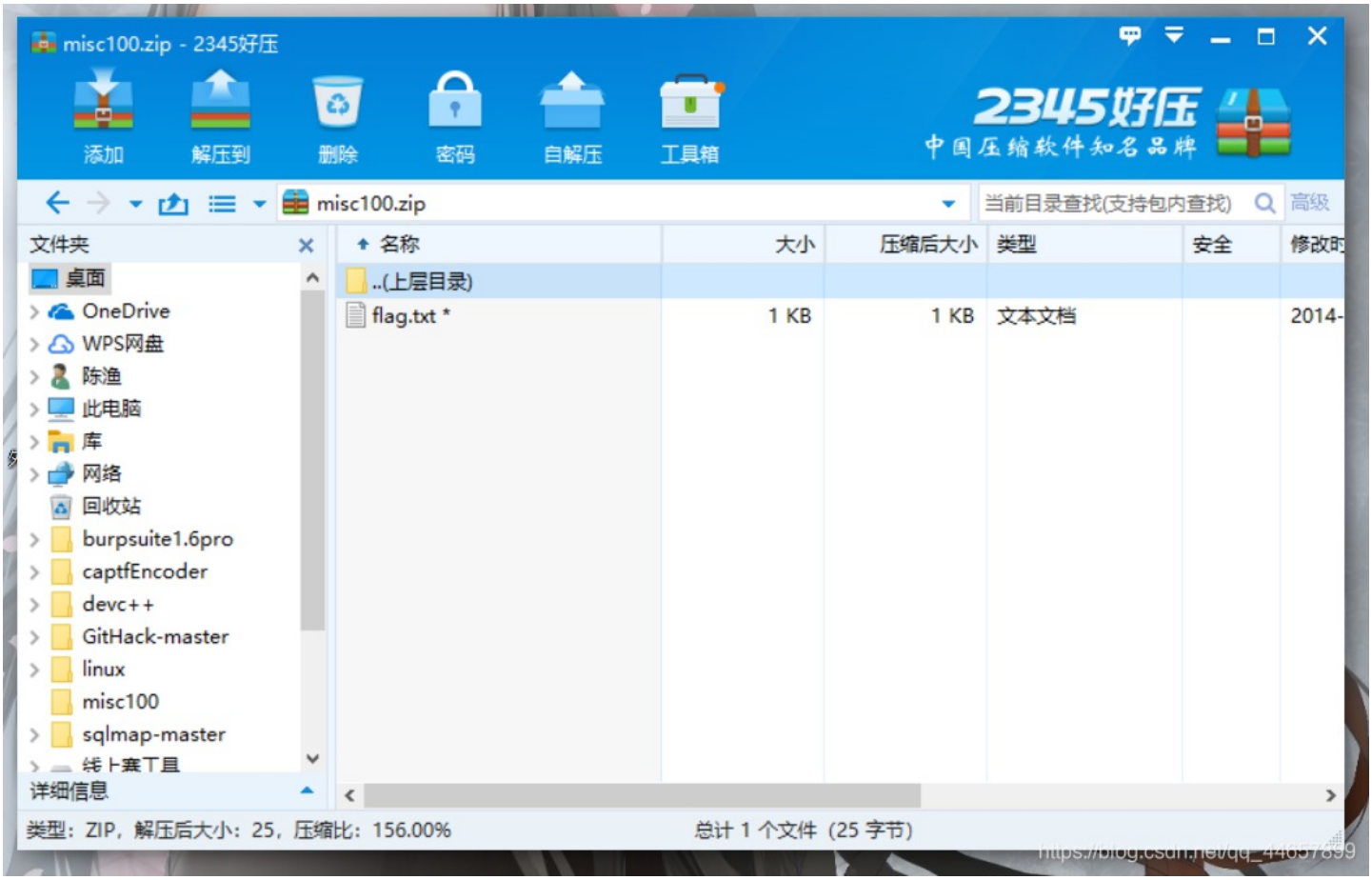
直接用winhex打开得到flag。



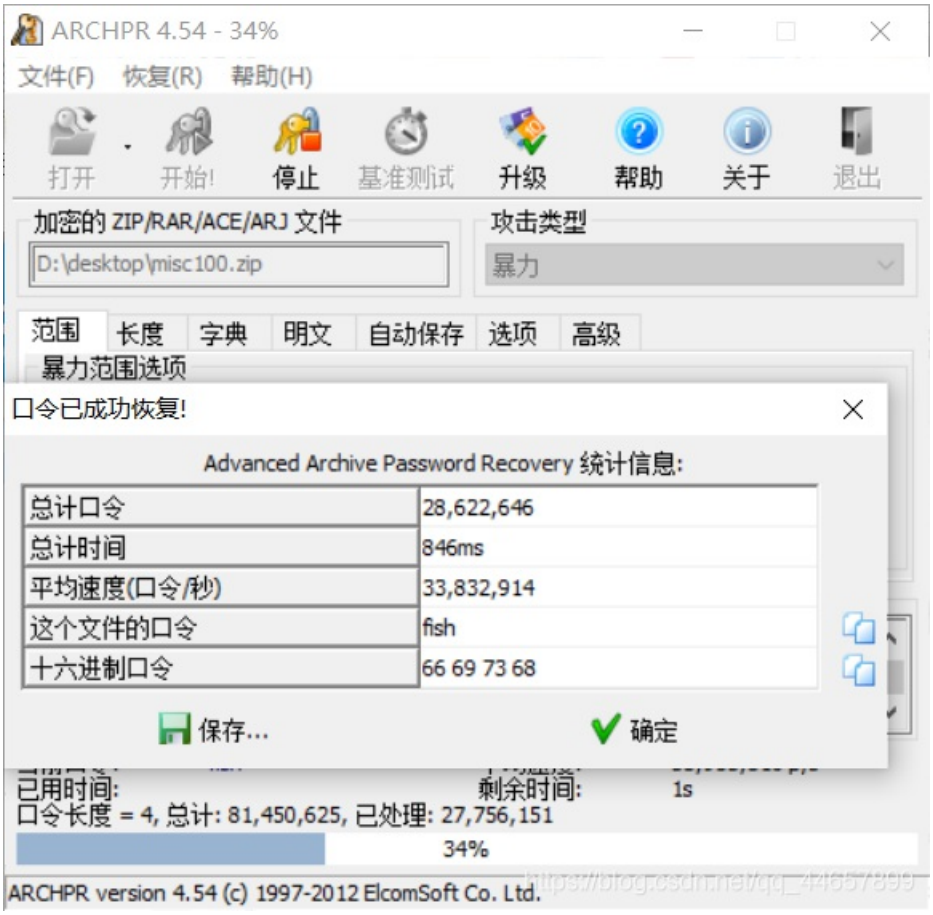
János-the-Ripper（ARCHPR爆破）

日期：2020/01/25

解压出来winhex发现仍是压缩文件，改后缀为zip，打开后是txt加密文件。



然后用ARCHPR爆破



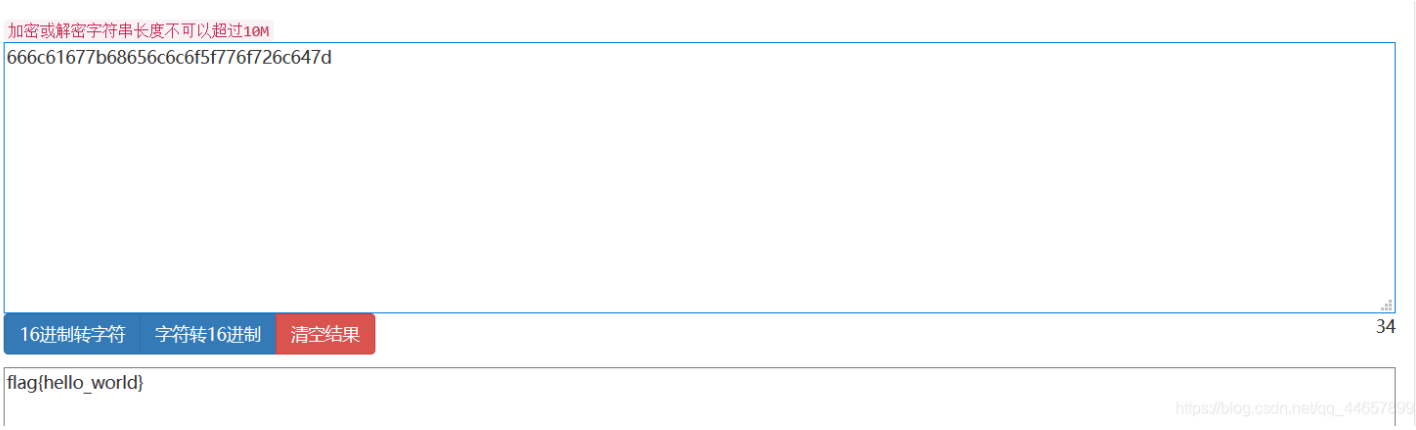
Test-flag-please-ignore（16进制转字符）

日期：2020/01/25

解压后打开是一串字符。

```
666c61677b68656c6c6f5f776f726c647d
```

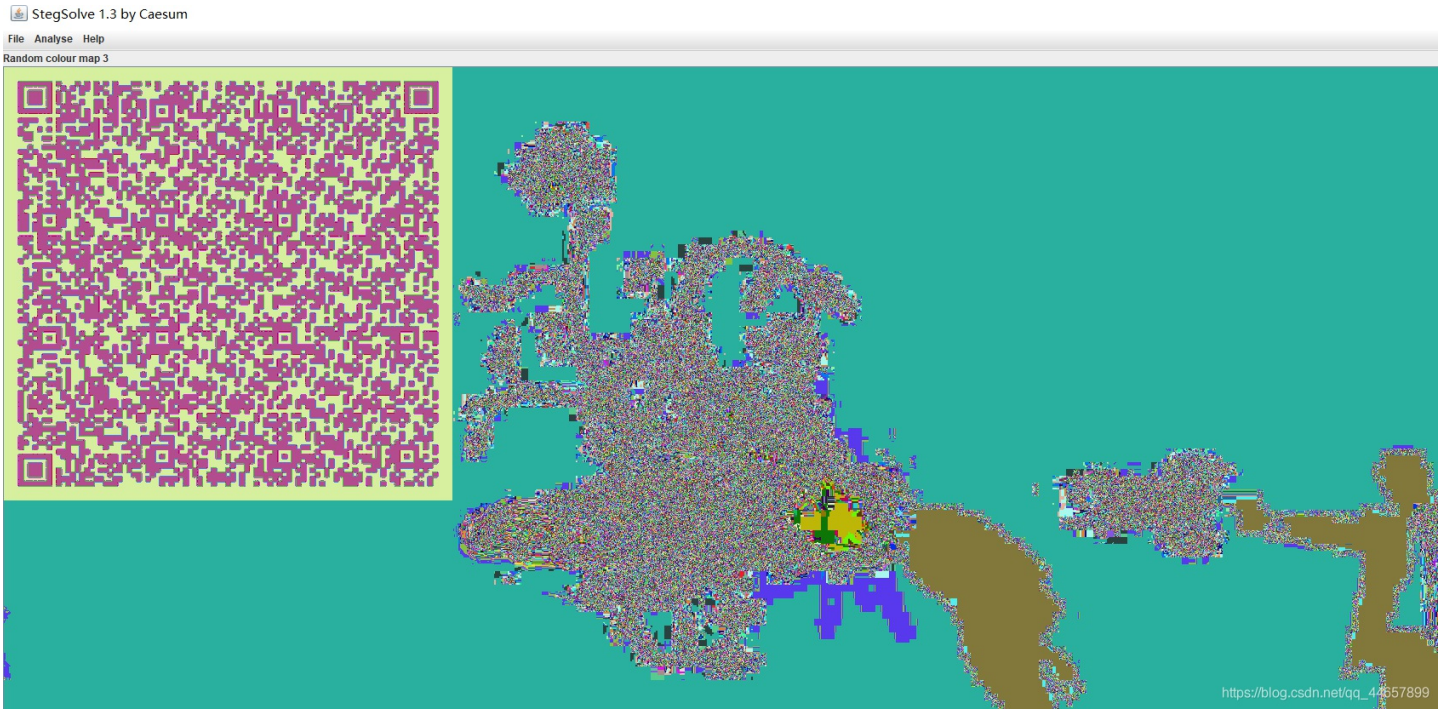
发现没有f以后的字符猜测是16进制，解密以后得到flag。



stage1（图片隐写，反编译）

日期：2020/01/25

下载附件，使用StegSolve打开，切换到左一视图时看到如下图片

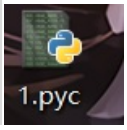


扫描二维码得到一串十六进制数字

```
03F30D0AB6266A5763000000000000000001000000400000007300D000000640000840000
5A0000640100532802000000630000000003000000080000004300000073
4E0000006401006402006403006404006405006406006405006407006708007D00006408007D0100781E007C0000445D16007D02007C01007400007C020083010037
7D0100712B00577C0100474864000053280900000004E6941000000696C00000069700000006968000000069610000006
94C0000006962000000740000000028010000007403000000636872280300000074030000007374727404000000666C6167740100000069280000000028000000007307000000746573742E70795203000000010000007
30A00000000011E0106010D0114014E2801000000520300000028000000002800000000280000000073070000007465737
42E707974080000003C6D6F64756C653E0100000073000000000
```

从开头03F3可知时pyc文件，保存为1.pyc

```
F3 0D 0A B6 26 6A 57 63  00 00 00 00 00 00 00 00  ó  ¶&jWc
01 00 00 00 40 00 00 00  73 0D 00 00 00 64 00 00      @    s    d
84 00 00 5A 00 00 64 01  00 53 28 02 00 00 00 63  „  Z  d  S(    c
00 00 00 00 03 00 00 00  08 00 00 00 43 00 00 00              C
73 4E 00 00 00 64 01 00  64 02 00 64 03 00 64 04  sN   d  d  d  d
00 64 05 00 64 06 00 64  05 00 64 07 00 67 08 00   d  d  d  d  g
7D 00 00 64 08 00 7D 01  00 78 1E 00 7C 00 00 44  }  d  }  x  |  D
5D 16 00 7D 02 00 7C 01  00 74 00 00 7C 02 00 83  ]  }  |  t  |  f
01 00 37 7D 01 00 71 2B  00 57 7C 01 00 47 48 64    7}  q+ W|  GHd
00 00 53 28 09 00 00 00  4E 69 41 00 00 00 69 6C    S(    NiA   il
00 00 00 69 70 00 00 00  69 68 00 00 00 69 61 00     ip   ih   ia
00 00 69 4C 00 00 00 69  62 00 00 00 74 00 00 00    iL   ib   t
00 28 01 00 00 00 74 03  00 00 00 63 68 72 28 03   (    t    chr(
00 00 00 74 03 00 00 00  73 74 72 74 04 00 00 00     t    strt
66 6C 61 67 74 01 00 00  00 69 28 00 00 00 00 28  flagt    i(    (
00 00 00 00 73 07 00 00  00 74 65 73 74 2E 70 79      s    test.py
52 03 00 00 00 01 00 00  00 73 0A 00 00 00 00 01  R         s
1E 01 06 01 0D 01 14 01  4E 28 01 00 00 00 52 03          N(    R
00 00 00 28 00 00 00 00  28 00 00 00 00 28 00 00     (    (    (
00 00 73 07 00 00 00 74  65 73 74 2E 70 79 74 08    s    test.pyt
00 00 00 3C 6D 6F 64 75  6C 65 3E 01 00 00 00 73     <module>    s
00 00 00 00
```



python反编译，介绍一个在线python反编译网站：python反编译在线工具

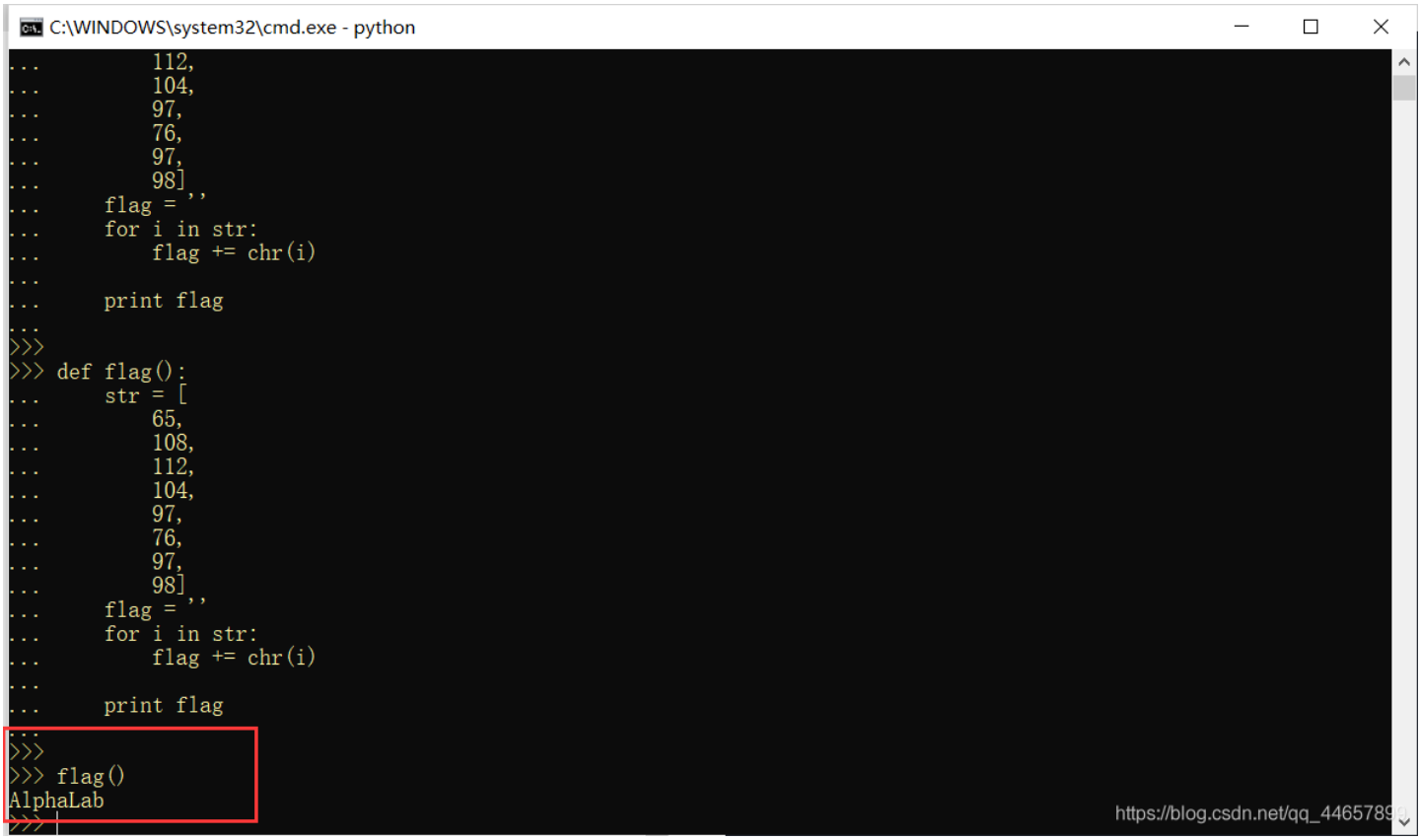
```
def flag():
    str = [
        65,
        108,
        112,
        104,
        97,
        76,
        97,
        98]
    flag = ''
    for i in str:
        flag += chr(i)

    print flag


def flag():
    str = [
        65,
        108,
        112,
        104,
        97,
        76,
        97,
        98]
    flag = ''
    for i in str:
        flag += chr(i)

    print flag
flag()
```

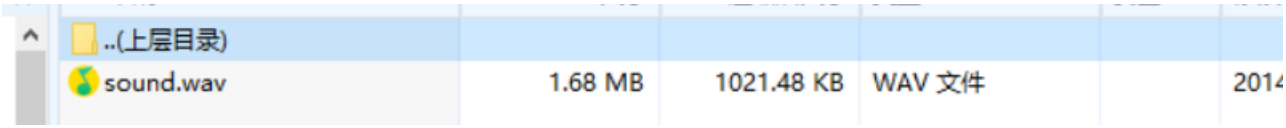
最后要加一个flag()，调用函数，得到flag。



Hear-with-your-Eyes（音频隐写-频谱图）

日期：2020/01/27

下载好文件之后解压，发现里面有个没有后缀的文件，放到winhex里观察，文件头是`1f8b0800`,百度搜了搜发现是gzip文件的文件头，于是把后缀改为zip。打开里面有一首歌。



然后应该是音频隐写类的题目，上网搜了搜这类题怎么做，结果示例的第二题就是原题，，，

CTF中音频隐写的一些整理总结

用audacity 打开这个音频文件，然后调至频谱图，出现flag。