

# Bugku Misc 整合（进行中。。。。。）

原创

S40D1  于 2019-04-06 12:36:47 发布  614  收藏 2

分类专栏: [ctf](#) 文章标签: [ctf misc](#)

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\\_37516675/article/details/89054161](https://blog.csdn.net/weixin_37516675/article/details/89054161)

版权



[ctf 专栏收录该内容](#)

1 篇文章 0 订阅

订阅专栏

## Table of Contents

[Bugku Misc 想蹭网先解开密码](#)

[Bugku Misc 账号被盗了](#)

[Bugku Misc 细心的大象](#)

[Bugku Misc 爆照\(08067CTF\)](#)

[bugku misc 猫片\(安恒\)](#)

[BugKu misc 多彩](#)

[BugKu misc 旋转跳跃](#)

---

## Bugku Misc 想蹭网先解开密码

题目如下:

开局一张图。故事全靠编, hahhahahah

Challenge 1382 Solves ×

# 想蹭网先解开密码

## 100

flag格式: flag{你破解的WiFi密码}

tips: 密码为手机号, 为了不为难你, 大佬特地让我悄悄地把前七位告诉你

1391040\*\*

Goodluck!!

作者@NewBee

wifi.cap

Flag

Submit

[https://blog.csdn.net/weixin\\_37516675](https://blog.csdn.net/weixin_37516675)

下载附件:

为cat文件。

给了数据包破解WiFi密码, 基本上都是爆破了:

1. 创建密码字典:

```
crunch 11 11 -t 1391040%% -o password.txt
```

2. 爆破:

```
aircrack-ng -a2 wifi.cap -w password.txt
```

提示需要爆破的对象:

```
root@kali: ~/桌面
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@kali:~/桌面# crunch 11 11 -t 1391040%% -o password.txt
Crunch will now generate the following amount of data: 120000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 10000
crunch: 100% completed generating output
root@kali:~/桌面# aircrack-ng -a2 wifi.cap -w password.txt
Opening wifi.cap please wait...
Read 4257 packets.
# BSSID          ESSID          Encryption
1 3C:E5:A6:20:91:60 CATR           WPA (0 handshake)
2 3C:E5:A6:20:91:61 CATR-GUEST    WPA (0 handshake)
3 BC:F6:85:9E:4E:A3 D-Link_DIR-600A WPA (1 handshake, with PMKID)
Index number of target network ?
```

看到第三个存在握手包，噢哈哈哈哈哈哈哈哈，就是它了！！！！

呵呵，真鸡儿快！

```
Index number of target network ? 3
Opening wifi.cap please wait...
Read 4257 packets.
1 potential targets
Aircrack-ng 1.5.2
[00:00:01] 9904/9999 keys tested (5181.70 k/s)
Time left: 0 seconds 99.05%
KEY FOUND! [ 13910407686 ]
Master Key : DD C5 F1 5E 32 BE 14 EC 68 F8 64 7C C5 BD FE 8F
11 2A 83 76 A8 DF F4 17 D0 A8 DA 2F 45 AB 1E 67
Transient Key : 0F 2E 92 B1 C6 FF 02 8A 1F 01 BE 1D 71 77 3E 7B
00 9E 8C 45 FB 3B FE 4E 4D C8 BD 21 9C C6 B0 E9
CA B2 CF 17 64 0C 6A 58 00 DC 94 91 8F F5 37 A0
75 48 D8 37 69 D1 FB FC D8 9E 92 EE F0 18 1A 0B
EAPOL HMAC : 31 EB 87 12 00 BE B4 F4 5B A9 26 E6 27 EC 5C 56
root@kali:~/桌面#
```

那么：

flag{13910407686}

嗯，100分到手！





点开链接下载文件解压得到一个.jpg文件。

顺手点开属性一看：

这家伙还有一个备注：TVNEUzQ1NkFTRDEyM3p6



看起来挺像base64编码，base64解码搞一哈嗨：

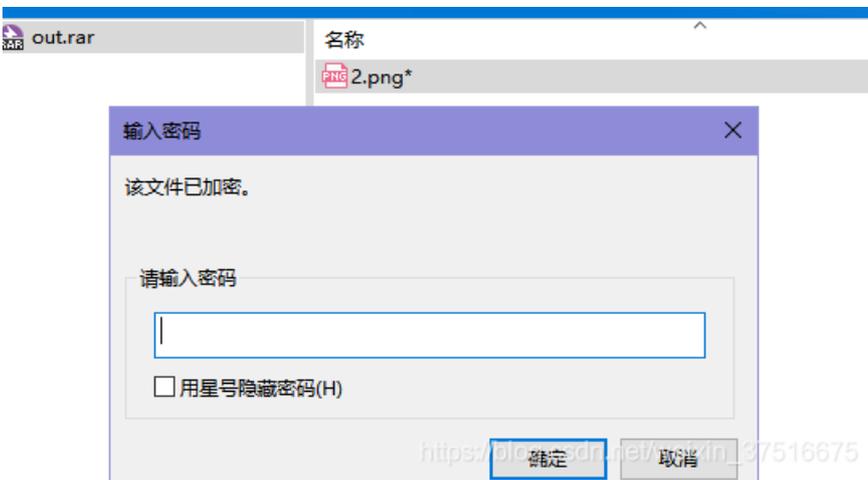
解密得到：MSDS456ASD123zz

看起来也不像flag，试了一下确实不是，那就先放这。

再看看图片大小，6M多，贼吉尔大！用binwalk分析一下试试：

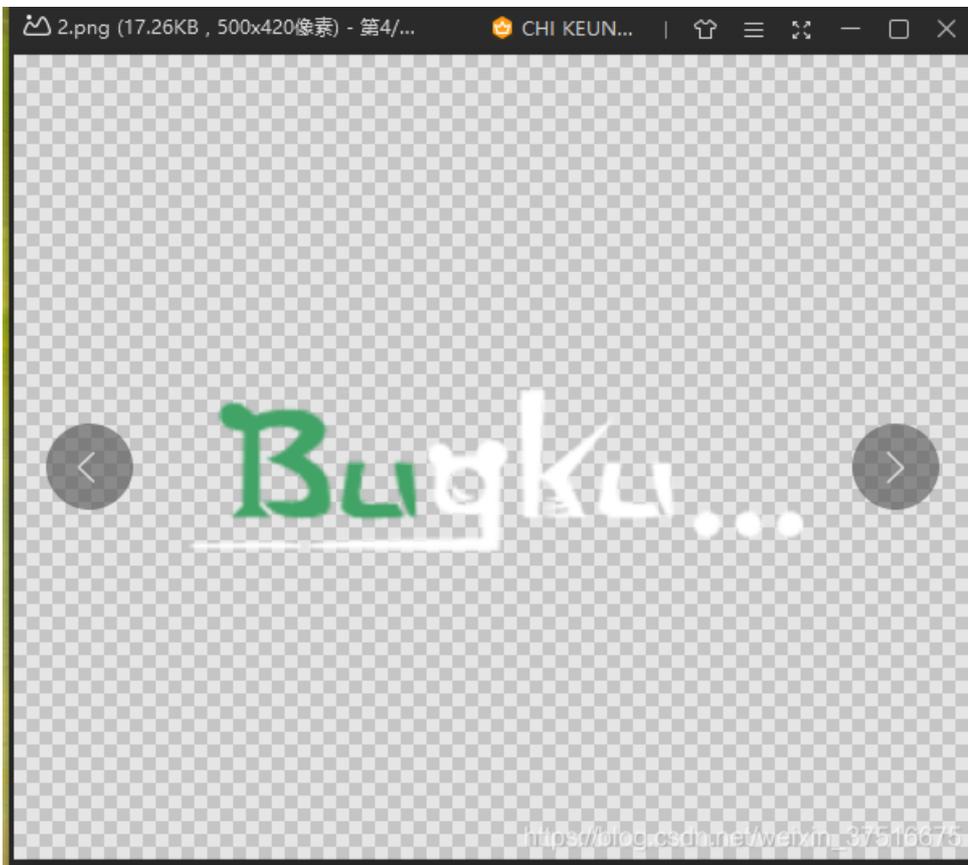
```
root@kali: ~/桌面
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@kali:~/桌面# binwalk 1.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0 主目录      0x0          out.rJPEG image data, EXIF standard
12           0xC         TIFF image data, big-endian, offset of first image
directory: 8
5005118      0x4C5F3E    PARity archive data
6391983      0x6188AF    RAR archive data, version 4.x, first volume type: M
AIN_HEAD     原文件名
有个RAR压缩包
设置读写缓存区的字节数
root@kali:~/桌面# dd if=1.jpg of=out.rar bs=1 skip=6391983
记录了16301+0 的读入
记录了16301+0 的写出
16301 bytes (16 kB, 16 KiB) copied, 0.0331621 s, 492 kB/s
root@kali:~/桌面#
```

打开分离出来的out.rar压缩包:



哎呀卧槽，需要密码。。。。那会不会。。。。嘿嘿，试一下哈（备注base64解码后的数据）

啧啧啧，还真是，解压后的文件为：



问题是flag还是不见啊。。。。

但是，，，这不是回到隐写2了吗？

用winhex修改宽高：把第二行第六列的01改成02保存

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG	IHDR
00000010	00	00	01	F4	00	00	02	A4	08	06	00	00	00	CB	D6	DF	ó	✖
00000020	8A	00	00	00	09	70	48	59	73	00	00	12	74	00	00	12	Š	pHYs t
00000030	74	01	DE	66	1F	78	00	00	0A	4D	69	43	43	50	50	68	t	Ëf x MiCCPPh

然后：



就没有有然后了。。。100分

## Bugku Misc 爆照(08067CTF)

Challenge 1268 Solves ×

爆照(08067CTF)

100

flag格式 flag{xxx\_xxx\_xxx}

8.jpg

Flag

Submit

[https://blog.csdn.net/weixin\\_37516675](https://blog.csdn.net/weixin_37516675)

下载附件: 8.jpg



用binwalk查看文件：内含9个文件

```

root@kali:~/桌面/CTF# binwalk 8.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0          JPEG image data, JFIF standard 1.01
40499       0x9E33       Zip archive data, encrypted at least v2.0 to extract, compressed size: 8362, uncompressed size: 92278, name: 8
48892       0xBEFC       Zip archive data, at least v2.0 to extract, compressed size: 14906, uncompressed size: 15739, name: 88
63830       0xF956       Zip archive data, at least v2.0 to extract, compressed size: 11129, uncompressed size: 18479, name: 888
74992       0x124F0      Zip archive data, at least v2.0 to extract, compressed size: 10371, uncompressed size: 11782, name: 8888
85397       0x14D95      Zip archive data, at least v2.0 to extract, compressed size: 6945, uncompressed size: 92278, name: 88888
92377       0x168D9      Zip archive data, at least v2.0 to extract, compressed size: 6824, uncompressed size: 92278, name: 888888
99237       0x183A5      Zip archive data, at least v2.0 to extract, compressed size: 7076, uncompressed size: 92278, name: 8888888
106350      0x19F6E      Zip archive data, at least v2.0 to extract, compressed size: 8219, uncompressed size: 92278, name: 88888888
168452      0x29204      End of Zip archive, footer length: 22

```

用foremost提取出来：



用binwalk挨个进行分析：88,888,8888三个文件内含信息

```

root@kali:~/桌面/CTF# binwalk 8
DECIMAL      HEXADECIMAL     DESCRIPTION
-----
0            0x0             PC bitmap, Windows 3.x format,, 303 x 300 x 8
root@kali:~/桌面/CTF# binwalk 88
DECIMAL      HEXADECIMAL     DESCRIPTION
-----
0            0x0             JPEG image data, JFIF standard 1.01
40499       0x9E33         Zip archive data, encrypted at least v2.0
compressed size: 92278, name: 8
48892       0xBEFC         Zip archive data, at least v2.0 to extract
0            0x0             JPEG image data, JFIF standard 1.01
30         63830         0xF956         TIFF image data, big-endian, offset of first image
directory: 8
compressed size: 11129, uncompressed size: 18479, name: 888
74992       0x124F0        Zip archive data, at least v2.0 to extract
root@kali:~/桌面/CTF# binwalk 888
DECIMAL      HEXADECIMAL     DESCRIPTION
-----
0            0x0             JPEG image data, JFIF standard 1.01
30         85397         0x14095        Zip archive data, at least v2.0 to extract
compressed size: 11782, name: 8888
root@kali:~/桌面/CTF# binwalk 8888
DECIMAL      HEXADECIMAL     DESCRIPTION
-----
0            0x0             JPEG image data, JFIF standard 1.01
30         99237         0x183A5        TIFF image data, big-endian, offset of first image
directory: 8
compressed size: 7076, uncompressed size: 92278, name: 8888888
106350      0x19F6E        Zip archive data, at least v2.0 to extract
compressed size: 92278, name: 88888888
root@kali:~/桌面/CTF# binwalk 88888
DECIMAL      HEXADECIMAL     DESCRIPTION
-----
0            0x0             JPEG image data, JFIF standard 1.01
30         168452        0x29204        End of Zip archive, footer length: 22
gif
https://blog.csdn.net/weixin_37516675

```

```

root@kali:~/桌面/CTF# binwalk 88888
DECIMAL      HEXADECIMAL      DESCRIPTION
-----
0 0x0 PC bitmap, Windows 3.x format,, 303 x 300 x 8
root@kali:~/桌面/CTF# binwalk 888888
DECIMAL      HEXADECIMAL      DESCRIPTION
-----
0 0x0 PC bitmap, Windows 3.x format,, 303 x 300 x 8
root@kali:~/桌面/CTF# binwalk 8888888
DECIMAL      HEXADECIMAL      DESCRIPTION
-----
0 0x0 PC bitmap, Windows 3.x format,, 303 x 300 x 8
root@kali:~/桌面/CTF# binwalk 88888888
DECIMAL      HEXADECIMAL      DESCRIPTION
-----
0 0x0 PC bitmap, Windows 3.x format,, 303 x 300 x 8

```

可以看见 88 上面有个二维码：



扫一下得内容为：bilibili

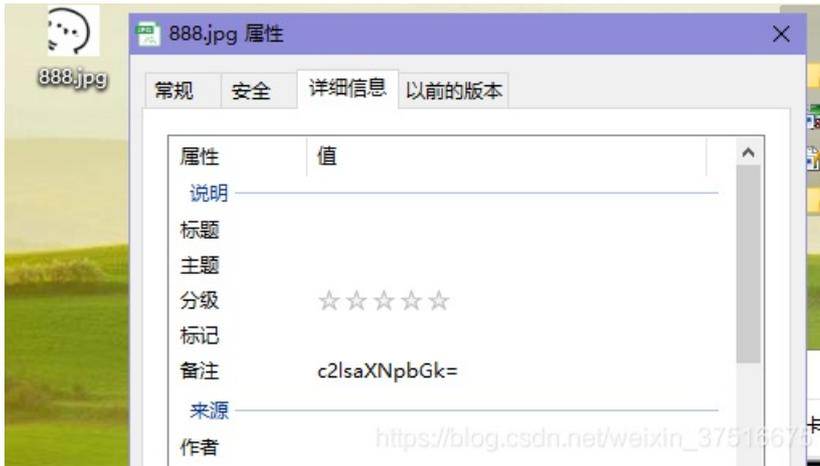
用foremost提取8888：可以得到一个二维码



扫一下得到信息：panama

那么888内的信息是什么呢？

给其加上jpg后缀，查看属性，可以看见备注信息：c2lsaXNpbGk=



base64解密得：silisili

至此三个含有信息的图像所含信息都出来了

尝试了几个组合，最终发现了flag为：flag{bilibili\_silisili\_panama}

## bugku misc 猫片(安恒)



打开附件添加png后缀：



将文件输出保存的格式bin(txt格式用winhex和010Editor打开都是乱码), 然后我们修改文件的后缀为png, 然后发现图片打不开, 用winhex打开后删除前面的FFFE保存:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG	IHDR
00000010	00	00	01	18	00	00	00	8C	08	02	00	00	00	08	EC	7E	G	i~
00000020	DB	00	00	05	9C	49	44	41	54	78	9C	ED	DD	51	6A	1C	U	aIDATxaiYQj

得到半张二维码:



然后改写图片高度:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG	IHDR
00000010	00	00	01	18	00	00	00	8C	08	02	00	00	00	08	EC	7E	G	i~
00000020	DB	00	00	05	9C	49	44	41	54	78	9C	ED	DD	51	6A	1C	U	aIDATxaiYQj
00000030	3B	14	40	C1	38	64	FF	5B	76	16	10	14	50	38	37	92	; @A8dy[v	P87'
00000040	EC	AA	DF	37	AF	DD	EE	F1	41	90	8B	D4	3F	7E	00	00	i*B7-YiñA	<O?~

保存得到:



诶呀卧槽, 这二维码看起来贼吉尔别扭!

拿画图工具反色一哈子:



用QR research 看一下：



看到了百度网盘的链接：<https://pan.baidu.com/s/1pLT2J4f>

进入是一个flag.rar的压缩包：



下载下来，解压打开：



什么鬼？魔鬼吗？

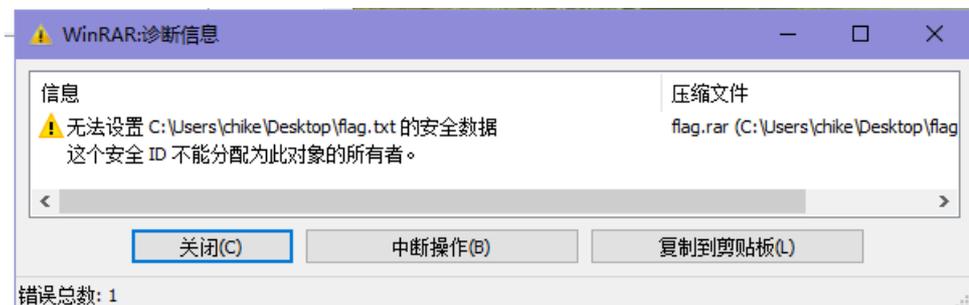
不知道怎么下手了。。。

度娘搜了下，发现另一位老铁写的writeup

<https://www.jianshu.com/p/abc44c54857a>

最后根据hint里面的提示“NTFS”，根据大佬的说法，这是一种流隐写，需要用到工具

ntfstreamseditor，然而。。这里还有一个坑就是，这压缩文件一定要用winrar来解压才会产生这样的效果



接着用ntfstreamseditor，查看解压的文件夹里面的数据流，然后把他导出来：

NtfStreamsEditor2

NtfStreamsEditor  
Ntf数据流处理工具

<http://blog.sina.com.cn/advnetsoft>  
advnetsoft@sina.com  
by XGQ

搜索 编辑 记录 信息

选择搜索类型  
 全部NTFS磁盘  
 自定义磁盘/文件夹 C:\Users\chike\Desktop

数据流名称匹配  
\* 搜索 停止

搜索结果: 共62个; 用时0.578 s

* 文件	数据流名称	大小(字节)	可疑度(0-5)
<input type="checkbox"/> C:\Users\chike\Desktop\f.txt:QQPcDocManager	QQPcDocManager	22	1
<input type="checkbox"/> C:\Users\chike\Desktop\flag.rar:Zone.Identifier	Zone.Identifier	969	1
<input checked="" type="checkbox"/> C:\Users\chike\Desktop\flag.txt:flag.pyc	flag.pyc	755	1
<input type="checkbox"/> C:\Users\chike\Desktop\jd-gui-1.4.1.jar:Zone.Identifier	Zone.Identifier	598	1
<input type="checkbox"/> C:\Users\chike\Desktop\Matlab7_清华大学教程.ppt:Q...	QQPcDocManager	22	1
<input type="checkbox"/> C:\Users\chike\Desktop\Matlab7_清华大学教程.ppt:Zo...	Zone.Identifier	197	0
<input type="checkbox"/> C:\Users\chike\Desktop\png.png:Zone.Identifier	Zone.Identifier	146	0

[https://blog.csdn.net/weixin\\_37516675](https://blog.csdn.net/weixin_37516675)

将里面的flag.pyc文件导出：这是一个py文件编译后的文件



拿去在线反编译一下：

得到的是一个加密脚本:



搜索工具头似间早

正则 运行代码 icon 搜索

所有 开发类 站长类 极客类 HR 其它 码农文库 奇淫巧技 软件推荐 网址导航 Wiki

请选择pyc文件进行解密。支持所有Python版本

选择文件 未选择任何文件

```
#!/usr/bin/env python
# encoding: utf-8
# 如果觉得不错，可以推荐给你的朋友！http://tool.lu/pyc
import base64

def encode():
    flag = '*****'
    ciphertext = []
    for i in range(len(flag)):
        s = chr(i ^ ord(flag[i]))
        if i % 2 == 0:
            s = ord(s) + 10
        else:
            s = ord(s) - 10
        ciphertext.append(str(s))

    return ciphertext[::-1]

ciphertext = ['96', '65', '93', '123', '91', '97', '22', '93', '70', '102',
              '94', '132', '46', '112', '64', '97', '88', '80', '82', '137',
              '90', '109', '99', '112']

import base64
```

美化(Beautify)

下载(Download)

[https://blog.csdn.net/weixin\\_37516675](https://blog.csdn.net/weixin_37516675)

再编辑如下解密脚本代码:

```
def decode():
    ciphertext = [
        '96',
        '65',
        '93',
        '123',
        '91',
        '97',
        '22',
        '93',
        '70',
        '102',
        '94',
        '132',
        '46',
        '112',
        '64',
        '97',
        '88',
        '80',
        '82',
        '137',
        '90',
        '109',
        '99',
        '112']
    ciphertext.reverse()
    flag = ''
    for i in range(len(ciphertext)):
        if i % 2 == 0:
            s = int(ciphertext[i]) - 10
        else:
            s = int(ciphertext[i]) + 10
        s=chr(i^s)
        flag += s
    return flag

def main():
    flag = decode()
    print(flag)

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

去运行一下：

```
Python 保存(Save) 我的代码 嵌入博客(Embed) 执行(Run) +
11 '70',
12 '102',
13 '94',
14 '132',
15 '46',
16 '112',
17 '64',
18 '97',
19 '88',
20 '80',
21 '82',
22 '137',
23 '90',
24 '109',
25 '99',
26 '112']
27 ciphertext.reverse()
28 flag = ''
29 for i in range(len(ciphertext)):
30     if i % 2 == 0:
31         s = int(ciphertext[i]) - 10
32     else:
33         s = int(ciphertext[i]) + 10
34     s=chr(i^s)
35     flag += s
36 return flag
37
38 def main():
39     flag = decode()
40     print(flag)
41
42 if __name__ == '__main__':
43     main()
flag{Y@e_Cl3veR_C1Ever!}
sandbox> exited with status 0
https://blog.csdn.net/weixin_37516675
```

哎呀卧槽：flag终于出来了。。。。。

flag{Y@e\_Cl3veR\_C1Ever!}

## BugKu misc 多彩

### BugKu misc 旋转跳跃

Challenge
359 Solves
×

## 旋转跳跃

### 100

熟悉的声音中貌似又隐藏着啥，key: syclovergeek  
 题目来源：第七季极客大挑战

sycgeek-mp3\_2...

Flag

Submit

https://blog.csdn.net/weixin\_37516675

打开题目是一个压缩包，解压是一个MP3音频文件，根据题目的提示的key

使用mp3stego 和提示的key 进行操作:

```
PS C:\Users\chike\Desktop\MP3Stego_1_1_19\MP3Stego> .\Decode.exe -X -P syclovergeek C:\Users\chike\Desktop\sycgeek-mp3_2\sycgeek-mp3.mp3
MP3StegoEncoder 1.1.19
See README file for copyright info
Input file = 'C:\Users\chike\Desktop\sycgeek-mp3_2\sycgeek-mp3.mp3' output file = 'C:\Users\chike\Desktop\sycgeek-mp3_2\sycgeek-mp3.mp3.pcm'
Will attempt to extract hidden information. Output: C:\Users\chike\Desktop\sycgeek-mp3_2\sycgeek-mp3.mp3.txt
the bit stream file C:\Users\chike\Desktop\sycgeek-mp3_2\sycgeek-mp3.mp3 is a BINARAY 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=111, tot bitrate=128, sfrq=44.1
mode=stereo, sbim=32, jsbd=32, ch=2
[Frame 5932]Avg slots/frame = 417.889; b/smp = 2.90; br = 127.979 kbps
Decoding of "C:\Users\chike\Desktop\sycgeek-mp3_2\sycgeek-mp3.mp3" is finished
The decoded PCM output file name is "C:\Users\chike\Desktop\sycgeek-mp3_2\sycgeek-mp3.mp3.pcm"
PS C:\Users\chike\Desktop\MP3Stego_1_1_19\MP3Stego>
```

[https://blog.csdn.net/weixin\\_37516675](https://blog.csdn.net/weixin_37516675)

在MP3文件同目录下多出两个文件:



打开TXT文件得到flag:



[创作打卡挑战赛](#)  
[赢取流量/现金/CSDN周边激励大奖](#)