

# BUGKU-杂项

原创

超级神兽小金刚 于 2020-02-09 22:13:35 发布 1353 收藏 2

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/wojiushilsy/article/details/104084620>

版权

## BugKu

[BugKu 专栏收录该内容](#)

2 篇文章 0 订阅

订阅专栏


1.这是一张单纯的图片

2.隐写

3.telnet







## 4.眼见非实(ISCCCTF)

下载文件zip(无后缀)，推测为zip包。补全后缀并打开

名称	压缩后大小	原始大小	类型
 眼见非实.docx	10,244	13,996	Microsoft Word 文档

果然如此，尝试打开报错，用010Editor查看文件头为**504B0304**

文件为压缩包文件修改后缀打开

名称	修改日期	类型	大小
 _rels	2020/1/25 20:52	文件夹	
 customXml	2020/1/25 20:52	文件夹	
 docProps	2020/1/25 20:52	文件夹	
 word	2020/1/25 20:52	文件夹	
 .project	2020/1/25 20:53	PROJECT 文件	1 KB
 [Content_Types].xml		XML 文档	2 KB

创建日期: 2016/8/15 4:00  
大小: 42.2 KB  
文件: \_rels, theme, document.xml, fontTable.xml, settings.xml, styles.xml, ...

看不懂，百度XML文件了解一下，没得头绪。

尝试直接查看文件，直接搜索出现flag。

## 5.啊哒

首先查看属性，得到：

```
照相机型号 73646E6973635F32303138
```

感觉有用，解码得：sdnisc\_2018

接着继续分析，010Editor打开，文件头为**FFD8FF**，文件尾应为**FFD9**。

很明显看到结尾不是FFD9，向上翻一点可以看到文件尾。

猜测为两个文件拼接，**binwalk ada.jpg**

```
[root@kaliDesktop]#binwalk ada.jpg
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, JFIF standard 1.01
30	0x1E	TIFF image data, big-endian, offset of first image directory: 8
5236	0x1474	Copyright string: "Copyright Apple Inc., 2018"
218773	0x35695	Zip archive data, encrypted at least v2.0 to extract, compressed size: 34, uncompressed size: 22, name: flag.txt
218935	0x35737	End of Zip archive, footer length: 22

猜测正确。**binwalk -e ada.jpg**

分离文件，得到压缩包。解压需要密码，尝试输入刚开始得到的 **sdnisc\_2018** 密码正确，得到flag。

## 6.又一张图片，还单纯吗

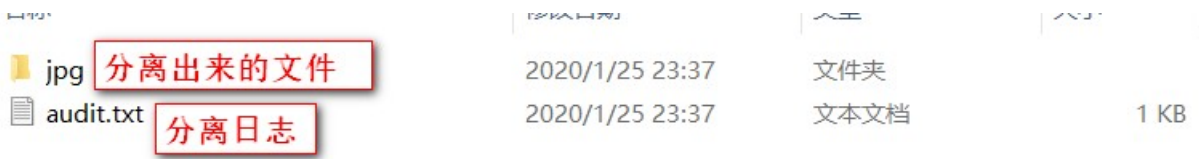


拖入kali binwalk分析

```
[root@kaliDesktop]#binwalk 2.jpg
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, EXIF standard
12	0xC	TIFF image data, big-endian, offset of first image directory: 8
158792	0x26C48	JPEG image data, JFIF standard 1.02
158822	0x26C66	TIFF image data, big-endian, offset of first image directory: 8
159124	0x26D94	JPEG image data, JFIF standard 1.02
162196	0x27994	JPEG image data, JFIF standard 1.02
168370	0x291B2	Copyright string: "Copyright (c) 1998 Hewlett-Packard Company"

不晓得为啥binwalk -e 没有分离图片  
这里使用了foremost分离结果如下:



打开jpg文件夹即可得到flag

## 7.猜

随便一个识图网站即可

360识图

百度识图

## 8.宽带信息泄露

用routerpassview打开.bin文件




```
</WANIPConnection/
<WANIPConnection nextInstance=3 />
<WANPPPPConnection instance=1 >
  <Enable val=1 />
  <DefaultGateway val=10.177.144.1 />
  <Name val=pppoe_eth1_d />
  <Uptime val=671521 />
  <Username val=053700357621 />
  <Password val=210265 />
  <X_TP_IFName val=ppp0 />
  <X_TP_L2IFName val=eth1 />
  <X_TP_ConnectionId val=1 />
  <ExternalIPAddress val=10.177.150.82 />
  <RemoteIPAddress val=10.177.144.1 />
  <DNSServers val=202.102.152.3,202.102.154.3 />
  <MACAddress val=D0:C7:C0:43:53:69 />
</WANPPPPConnection>
<WANPPPPConnection nextInstance=2 />
</WANConnectionDevice>
<WANConnectionDevice nextInstance=2 />
```



## 9.隐写2



binwalk分离文件。结果如下：

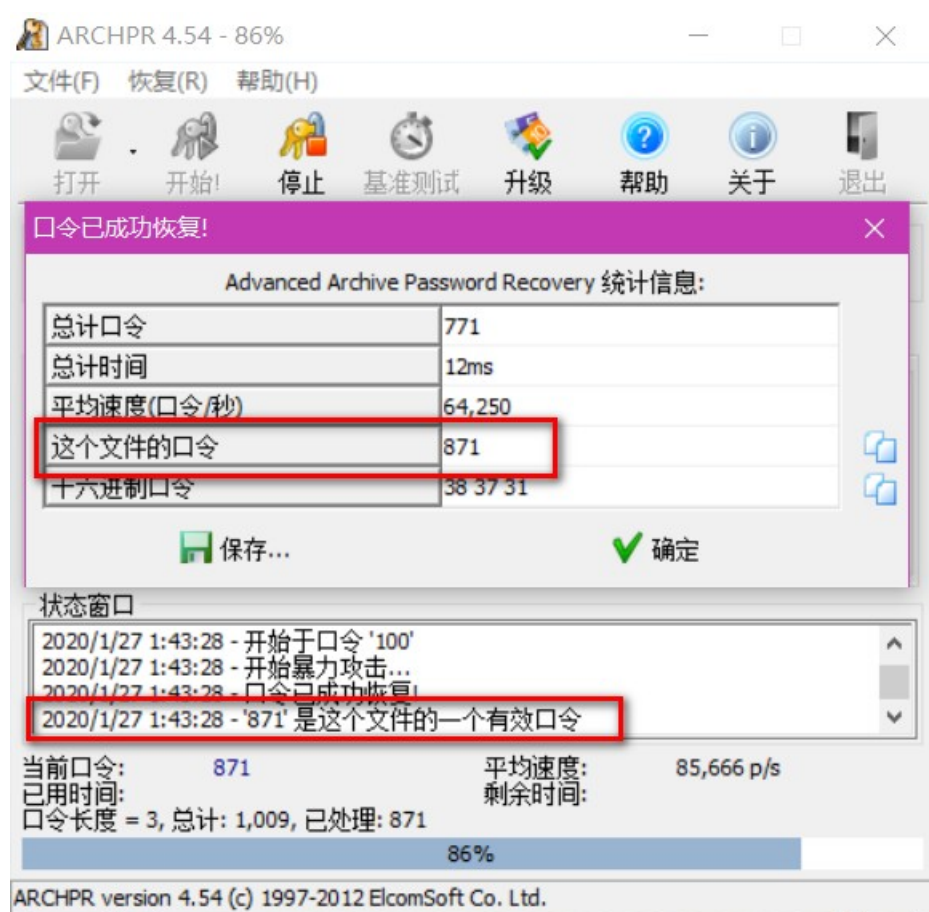
 CD24.zip	2020/1/27 0:54	ZIP 压缩文件	94 KB
 flag.rar	2020/1/27 0:54	RAR 压缩文件	7 KB
 提示.jpg	2020/1/27 0:54	JPG 文件	91 KB

### 得到flag.rar解压密码——方法一：

通过百度可以知道红桃k为查理曼，雅典娜为黑桃Q,梅花J代表着兰斯洛特。观察键盘可以发现对应数字为8、1、7。(如若不理解，搜一下打字指法，大概就可以明白为什么KQJ对应8、1、7了)然后就把817排列组合，挨个试就可以得到解压密码了。

### 得到flag.rar解压密码——方法二：

使用ARCHPR进行爆破



### 得到flag.rar解压密码——方法三：

fcrackzip暴力破解

fcrackzip是破解zip压缩包的，题目中虽然为rar后缀，但文件头为zip，故可以使用fcrackzip暴力破解

```
[root@kali11]#fcrackzip -b -c1 -l3-3 -v flag.rar  
found file '3.jpg', (size cp/uc 6588/ 6769, flags 801, chk 102c)  
possible pw found: 035 ()  
possible pw found: 337 ()  
possible pw found: 728 ()  
possible pw found: 871 ()
```

```

USAGE: fcrackzip
  [-b|--brute-force]      use brute force algorithm
  [-D|--dictionary]      use a dictionary
  [-B|--benchmark]       execute a small benchmark
  [-c|--charset characterset] use characters from charset
  [-h|--help]            show this message
  [--version]            show the version of this program
  [-V|--validate]        sanity-check the algorithm
  [-v|--verbose]         be more verbose
  [-p|--init-password string] use string as initial password/file
  [-l|--length min-max]  check password with length min to max
  [-u|--use-unzip]        use unzip to weed out wrong passwords
  [-m|--method num]       use method number "num" (see below)
  [-2|--modulo r/m]       only calculate 1/m of the password
  file...                 the zipfiles to crack

methods compiled in (* = default):

0: cpmask
1: zip1
*2: zip2, USE_MULT_TAB

```

参数选项	英文解释	中文解释
[-b --brute-force]	use brute force algorithm	爆破模式
[-D --dictionary]	use a dictionary	使用一个自定义字典
[-B --benchmark]	execute a small benchmark	速度测试
[-c --charset characterset]	use characters from charset	指定字符类型
[-h --help]	show this message	显示消息
[--version]	show the version of this program	显示版本
[-V --validate]	sanity-check the algortihm	健全性检查
[-v --verbose]	be more verbose	更详细
[-p --init-password string]	use string as initial password/file	指定开始字符(比如要只记得密码是3开头的6位纯数字, 可以指定从300000开始破解, 节约时间)
[-l --length min-max]	check password with length min to max	指定密码长度区间
[-u --use-unzip]	use unzip to weed out wrong passwords	使用unzip清除错误的密码
[-m --method num]	use method number "num" (see below)	(指定破解类型) 下面的数字选项
[-2 --modulo r/m]	only calculcate 1/m of the password	只计算 1/m 的密码
file...	the zipfiles to crack	
methods compiled in (* = default):		
0: cpmask		
1: zip1		
*2: zip2, USE_MULT_TAB		
下面简单介绍下 -c下的参数		
a	include all lowercase characters [a-z]	包含小写a-z
A	include all uppercase characters [A-Z]	包含大写A-Z
1	include the digits [0-9]	包含数字0-9
!	include [ ! : \$ % & / ( ) = ? { [ ] } * ~ #]	包含特殊字符

比如: fcrackzip -b -c a -l 1-4 -u test.zip 破解4位纯小写字母的test.zip文件, 这里需要注意一定要指定-u参数, 不然显示不出来密码

解压得到一张图片, winhex打开, 最后可以发现 flag(BASE64 加密的)。解密即可。

Base64密文有如下特点:

1. 字符串只可能包含A-Z, a-z, 0-9, +, /, =字符
2. 字符串长度是4的倍数
3. =只会出现在字符串最后, 可能没有或者一个等号或者两个等号

## 10.多种方法解决

下载直接打开



emmmm 010editor打开

```

0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160
data:image/jpg;base64,iVBORw0KGgoAAAANSUHEUgAAATUAAACFCAAAAA12js8AAAAAXNSR0IARs4c6QAAARnQUIBAAACXjv8YQURAAAjCv8ZcwaAADSMAAA7DacdvGQAAARZSURBVHne7ZKBitxIFgTv/
396Tx564G1UouicKg19nWpDcrMj9m7/7n45zfdxe5z3sJ7prHbf9rXO3P41LvYpctbeM80dvtP+3pnDp9yF7tneQvvcZu/21f78zhU+519yxv4T3T200/7eud68OT2H3Lcft01/ae9Z1To+23PvX7/rwJHbf
csI+3aW9233m1Gj7Len+9bsPIndt5ywT3dp71mfOTXaFku6F/2uD09i9y0n7Nnd2nvWZ06Nt+S7l+/68MjC500S0WpcyexnFj fcsI+JWlupkRfv+vDCXOTWE7a/i72PstJ2zfsHnOTpZ6XR9OmJvEctL2d7H3WU7avmH3mJsK5dfv+nDC3CSWk7a/i73PctL2DbvH3CQpv37XhPmJrGctP
1d7H2Wk7Zv2D3mJkn59bs+nDAeWefdnImy1Jne1p7H6bmyT11+/6cMLcJYJY9k0jbaYkdaansfttbpKUX7/rwlzklh02DeNtJms1Jmexu63uUlsfv2uDYfMTWI5Yd800mZKUm6Grvf5iZJ+FW7FjzJ7v12b
33LsdvsvfuW75LuX7/rw5Ps3m/rectP0Wu2/5Lun+9bs+PMnu/XZvfctJ22+x+5bvku5fv+vDk+zeb/Fwt5y0/Ra7b/ku6f71+++HT0v+513+tK935vApyd+8y5/29c4cPiX5m3f5077emcOnJH/zLn/ar3d+
/flBpI+cMDeNtJkSywn79BP5uK+yZTmpe2U2I5YZ9+Ih/3VfaPxtw00mZKLCfs00/k477K/tGYm0baTInlhH36iSxflT78TpI605bdEbF7lhvct54mwOaWJ6m4ZkdaYtu3ti9yw3uG89TXrHNLE8TcM7Sep
MW3b3xO52bnDfep0jmlieZg6d5Lmbs7onds9zgvv006R3TxPXSxPrW07YpyR1pgTNKUmKUm6d5LUAxzdwB/eYX3LcfuUpM6UtdKlqTmLgXNSkj qNrxvrvzusbzLhn5LmZi2pyR1PiR1TkpSp/F1Y314h/
UtJ+xtKjpt0UaPm6UpM5Jseo0ft34rOGNLgDfUos7inhvUtJ+ybRtpM4n039coa3cE+JZYb3FPD+pYTY9k0jbaa7pHt6NY3uYJ8Syw3uqWf9ywn7ppE201338Pb2aRnewT4nlBvFUsL7lh3T3JvpLunecjWV7
mcftqQbj9R1puR03tqSbkx/wrJqj7JPW9KRNrPi6U3I6b21JN6Y/YVmlR9mnlElG10mdKtmdt7akG90fsKzao+zTlnSjkaTolJzow1vsJelFwFbp8NRImy1JnWnlr6F7zN3Tcb32FppUNTI22mJHmLbv7Fz7P
3CXdbHyPzUOTY20mZLmbs71v4PnOXdlPxHZ2Wojq10mZKUmfasrtv4fvMXdLXndYwunQLFhutHv2W42n+4bds7w13VuuskS5Ua7Z7/VeLpv2D3LW9K95SpL1FhutHv2W42n+4bds7w13VuuskS5Ua7Z7/
VeLpv2D3LW9K97avp6GQ334X3Kw1z+tuks5j+h02/hX3Ebr4L71FS5vS3Sd8w/Qnbfwv7iN18F96npM3pb50+YfoTtv8W9H7+568T0mb098mfcP0Jxz/W+x+PfeThvUtN2y/m7fwnvml+frzIOkLDdy3Gta33L
D9bt7Ce+bx5UvPg6SXNHDfalj fcsP2u3kL75fm68/d5Je0sB9g2F9ywb7+YtvGd+bb7+vCEN7YspMzXSzrqL3b0csN9Kns4T2uJRk6TolE1b6S52z3Lcft50k9oi0dNkjpTI22mu9g9ywn7reTpPKtHjVJ6
kyNtJnuYvcsJ+y3kgfzxNliEUos+xtYVkuDtvg3tqpM2d5CF50mKJESsJ+5RYvovdt9zgnhppyf5Sb60WKLecsI+J2bvYvctN7inRtrcSX6SLy2WKLGcsE+J5bvYfcsN7gmRNneSn+RLK5UmbW4Symn710zm
hH3a0u72N99hadmRNjeJ5Y9Sn2zjw5taffsm+vtOxIm5vEcsI+Jbs5Y2+2tHv2zXyWnakzUliOWGfkt2cs9b2j375jtctv2+tuX0vXF9sXNkj rTT+T6rvxy37ac3re22J6S5VJn+olc35U/9tuW0/vWFts
zN0ngPD+R67vyx37bcnrf2mu75iZJneknUn+v/awUyNtpgTNq2E2UyNtGlvSjTs9VvtKHnqM2UtDkl0m2gppEljS7px26J+qxllTo20mZi2p0baTI20awxJN+5M1G+1o8ypkT2T0ubUSJupkTanLenGnVn16T
uj02zP3DTS2kp2c8L+0xppM32HpfWTlXPhMzeNtJm32yW/7RG2kzfYwn95Mj9esxNI22mZDcn7D+tkTbTdlhaPzKysTlZ00ibKdnNcftPa6Tn9B2uXh5/S9rcbEk37jR2+5SkzpSkzo4kdaavTg6/Jw1utqQbd
xg7Eupsz0pS20eSotMX74fkljY3W9KN043dPiWpMyWeyNemrk8NvS2ubLenGncZun5LmZLU2ZGkzvVWR/eo7a7Xdzw/bMKbGc7EbNE1x3uqNtn9+Nzdsz5w5y81u3z2Bdac72vau7Xdzw/bMKbGc7EbN
E13uqNtn9+Nzdsz5w5y81u3z2BcsVewpys1LmTW6Y1nLcPmlJN05KLP/D8tRGZclJnTu5YbtLSfs052046T8j3st23EnJLUuZNYbtjecsI+buk3Tkos/8Py1EbmKUm6d4nlhu0tJ+zTlnTjMTyP/R/18P
wI//fJZyB3Jvv8Pd/1l+WWG5wb77D3/8pfl1iucG9+Q5/76f4ZynlBvfmOly9PH/Kftbthq+zySpMyvtbr/Dlcvj2yxveWn4ftMkjpT0ubm0ly9PH/Kftbthq+zySpMyvtbr/Dlcvj2yxveWn4ftMkjpT0u
bm0ly9ftRg9y0n7FPD+paTtk9071sT13Mv7WD3LSfsU8P6po2T07vWxPXcy/tYpctJ+xtw/gWk7ZPTu9B9dZL+ig9y0n7FPD+paTtk9071sT1/P7EnoTWG5wb5LmRptn3D/6b6+eX04Yw4S9y3uTzi6U6PtE
+4/3dc3rwn8E1iucG9SVJnarR9w2n+/zm9eGUksN7g3SepMjBzPuP90X9+8PpwN0mb72Pxfzcn1rf8NHwfEXXWhxPmJmnzXQ3r7+bb+paFhu+jr876cMLJG2+g2H93ZxY3/LT8H301VkfTpbpM13Nay/
mxPrW34avo+OouDCXOT70ZGu7e+5Y9XynlhH36d1fvfsTcJn50e6tbzlhfidOWGfvsPVux8xN8lubr761to2N+VWE7Yp+9w9e5HzE2ymxvt3vgWE/Z3JZYT9uk7XL1+1GD3LX8avt8klhu2t5yc6F+/68O
T2H3Ln4bvN4nlhu0tJyF61+/68CR23/Kn4ftNylrhe8vJif71u248id23/Gn4fpNYbtjecnKif/3+++HTnub0fd4zieUtvLfr01y9PH7K05y+z3smsbyF9329h6uXx095mtP3ec8k1rfw3q7vcyXclPc/o+75
n8Hbe2/Udzv9X+sv/OP/881/SgtvcdbH+WAAABJR5ErkJgg==

```

很明显BASE64加密的, 还原为图片。得到二维码。使用小米手机\_(:3|\_<)\_扫一扫即可得到flag。

## 11.闪的好快

使用Stegsolve逐帧查看, 同时使用屏幕二维码扫描器得到二维码包含的信息 即可得到flag。

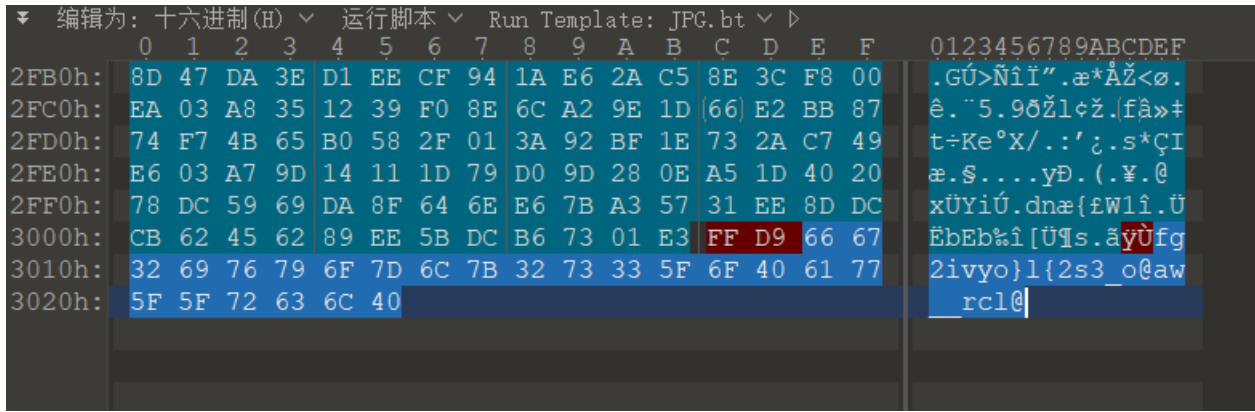
## 12.come\_game

下载解压之后发现是一个游戏  
 试玩一下会发现多了几个文件  
 其中有一个文件名为save1 (关卡)  
 修改即可

## 13.白哥的鸽子

查看属性，未得到有用信息。010editor 打开

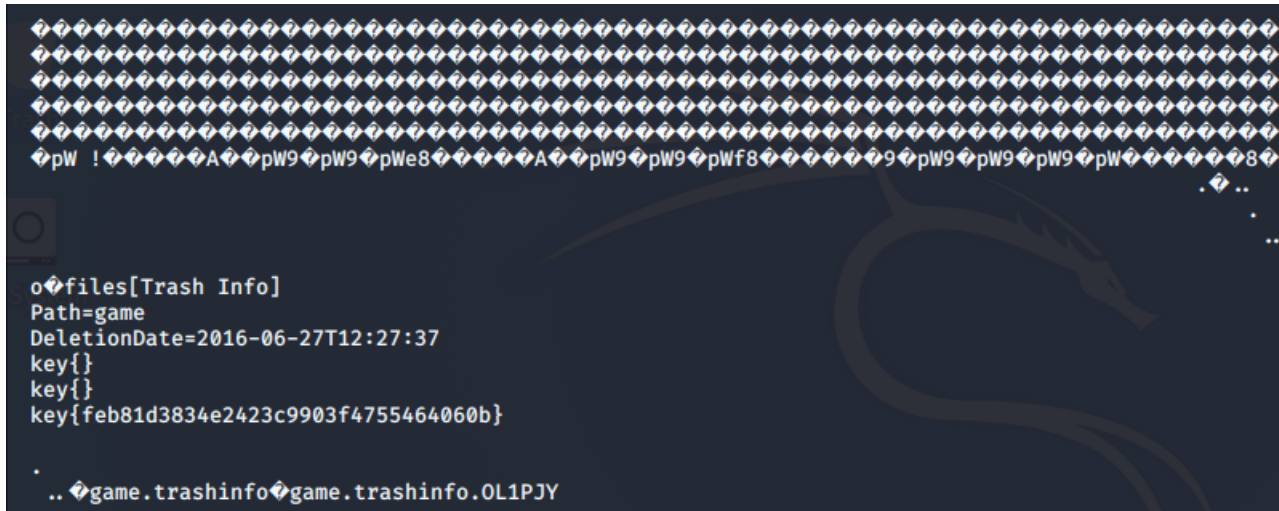
在结尾处发现：



可以看出有 **flag** 和 **{}|**，明文信息未改变，只是打乱了顺序，推测为栅栏加密法加密，解密即可。

## 14.linux

方法一 cat flag





方法二 cat flag |grep '{' -a -C 1 flag

查找 { 字符

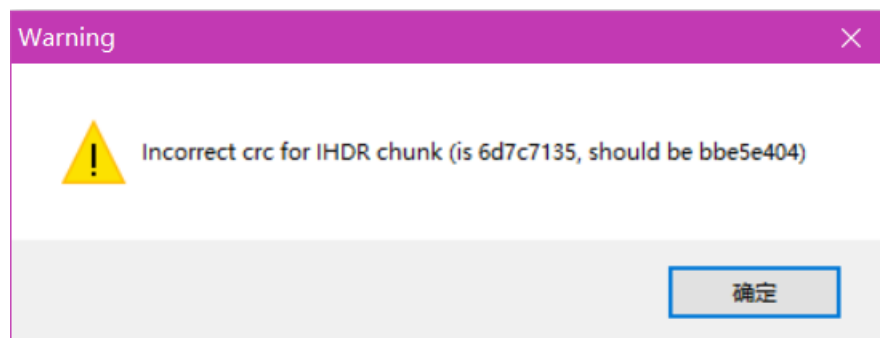
```
info@fites:~/trash_info$ cat flag |grep '{' -a -C 1 flag
--
DeletionDate=2016-06-27T12:27:37
key{}
key{}
key{feb81d3834e2423c9903f4755464060b}
.
.. game.trashinfo game.trashinfo.0L1PJY
.
.. game
```

方法三 strings flag

```
fites
[Trash Info]
Path=game
DeletionDate=2016-06-27T12:27:37
key{}
key{}
key{feb81d3834e2423c9903f4755464060b}
game.trashinfo
game.trashinfo.0L1PJY
game
```

## 15.隐写3

tweakpng打开



CRC校验错误，宽度或者高度出错。修复即可得到flag。

## 16.做个游戏(08067CTF)

```

编辑为: 十六进制(H) 运行脚本 运行模板
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0000h: 50 4B 03 04 14 00 08 08 08 00 AC 86 52 4B 00 00 PK.....-+RK..
0010h: 00 00 00 00 00 00 00 00 00 00 14 00 04 00 4D 45 .....ME
0020h: 54 41 2D 49 4E 46 2F 4D 41 4E 49 46 45 53 54 2E TA-INF/MANIFEST.
0030h: 4D 46 FE CA 00 00 F3 4D CC CB 4C 4B 2D 2E D1 0D MFpÊ..óMÏËLK-.Ñ.
0040h: 4B 2D 2A CE CC CF B3 52 30 D4 33 E0 E5 72 CE 49 K-*îîî³R0Ô3àârîI
0050h: 2C 2E D6 0D 48 2C C9 B0 52 D0 E3 E5 F2 4D CC CC ,.Ö.H,É°RÐãàòMÏÏ
0060h: D3 05 8B 59 29 24 E7 E9 25 65 15 57 94 E8 15 E4 Ó.<Y)şçé%e.W"è.ä
0070h: 24 E6 A5 EA 05 80 48 F7 C4 DC 54 B7 22 20 C1 CB şæ¥è.€H=ÄÜT·" ÁË
0080h: C5 CB 05 00 50 4B 07 08 81 91 F9 36 4E 00 00 00 ÄË..PK...`ù6N...
0090h: 53 00 00 00 50 4B 03 04 0A 00 00 08 00 00 35 86 S...PK.....5†
00A0h: 52 4B 00 00 00 00 00 00 00 00 00 00 00 00 03 00 RK.....

```

文件头为504B 0304(zip压缩文件)，修改后缀名。

解压缩粗略查看后，flag应在cn文件夹

windows下cmd

```

\heiheihei\cn findstr /s /i "flag" *.*
bjsxt\plane\PlaneGameFrame.class: ??? toString()Ljava/lang/String? [鏽?渣撒 《涓€鋼回瀟闊技嶠劍?? [鈔嚮瓊夸絳湑
燭 錫(瀾甯) ? -濡俗濼姊€先鍾夙 鑛詫紅間d 溥淪氣楸劍燭啤鑛? [錕摺紅錫聿鑛鑛 眠鍛€鑛勳? [鏢豹补浣羽氮錫 燮涓€
涓 €仨 ? -濡俗濼錕戮繡涓€€錄賺挖鋃賊職涓嘈楸衰塚病闊(-) ? flag{RGFqaURhbGlfSmIud2FuQ2hpamk=} [ [ [ [java/awt/Graphics;
StackMapTable [cn/bjsxt/plane/Bullet; [peng [ [period
FINDSTR: 写入错误

```

flag为base64加密，解密即可。

## 17.想蹭网先解开密码

已经提示了前七位密码。所以我们可以暴力破解。

破解工具我们选择aircrack-ng这里没得问题。但是字典怎么弄，我不晓得。

然后就问度娘了。发现了一个很好的字典生成工具crunch

```
[root@kaliDesktop]#crunch 11 11 -t 1391040%%-% -o hhh.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@kaliDesktop]#aircrack-ng wifi.cap -w hhh.txt
Opening wifi.cap please wait ...
Read 4257 packets.

# BSSID          ESSID          Encryption
1 3C:E5:A6:20:91:60 CATR           No data - WEP or WPA
2 3C:E5:A6:20:91:61 CATR-GUEST     None (10.2.28.31)
3 BC:F6:85:9E:4E:A3 D-Link_DIR-600A WPA (1 handshake, with PMKID)

Index number of target network ? y

Opening wifi.cap please wait ...
Read 4257 packets.

1 potential targets

Aircrack-ng 1.5.2

[00:00:02] 9472/9999 keys tested (4604.53 k/s)

Time left: 0 seconds          94.73%

KEY FOUND! [ 13910407686 ]

Master Key      : 80 25 44 FF 65 4F B4 16 A0 AD 85 00 53 C9 81 09
                  D6 0B EE 75 D8 1F 1A 44 2D 50 91 29 55 9D CF 39

Transient Key   : 2B 38 37 CC EF 0F BB 9F 0E 01 20 A9 26 52 8D 7C
                  5F D5 9C 32 D3 99 21 EE 08 44 3C 10 25 B5 AB B6
                  EA 97 39 25 85 4D E9 59 11 DF 96 52 2A 85 ED 00
                  3C 19 73 5D CB B6 7F CC A3 67 6B 5C 9B 4F C5 5D

EAPOL HMAC     : 7C D2 2B 0E 2F 72 90 CB 21 48 66 86 28 87 DE 6B
```

## 18.Linux2

```
[root@kaliDesktop]#cat brave |grep 'KEY{' -a -i -C 1 brave
HKEY{24f3627a86fc740a7f36ee2c7a1c124a}
```

## 19.账号被盗了

思路

## 20.细心的大象

打开图片看到SHOT ON MI 6

感觉属性里应该有信息，果不其然在备注中发现了信息

base64加密的，先解密出来

然后放到kali里

```
[root@kaliDesktop]#binwalk -e 1.jpg
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, EXIF standard
12	0xC	TIFF image data, big-endian, offset of first image directory: 8
5005118	0x4C5F3E	PARity archive data

得到一个加密的rar文件，输入刚刚解密得到的字符串

里面是一张图片，额...

和这是一张单纯的图片一毛一样的解法。得到flag

## 21.爆照(08067CTF)



首先查看图片属性，无果  
放到kali里，binwalk分析图片

```
[root@kaliDesktop]#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

分离出一个压缩包，打开发现有一张gif图和一些文件。首先根据文件头将文件的后缀名补齐。



很明显88.jpg有一张二维码。定位符好像有残缺，一直扫不到，给它补齐扫码的到bilibili。  
然后查看各个图片的属性在888.jpg的备注得到base64加密的字符串，解密得到silisili。  
8888.jpg也是binwalk得到一张二维码，扫码得panama  
结合题目提示flag格式：flag{xxx\_xxx\_xxx}  
将上面的字符串组合即可得到flag。

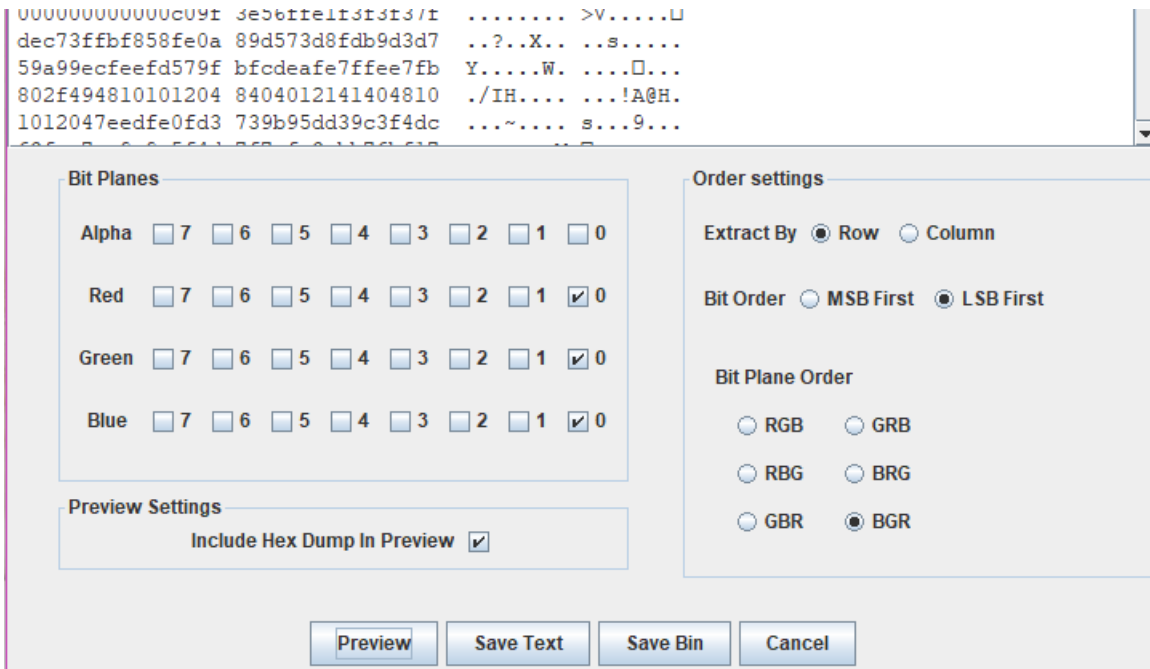
## 22.猫片(安恒)

题目提示 hint:LSB BGR NTFS

LSB为最低有效位；BGR为最低有效位的顺序；根据提示得：

```
Extract Preview
```

ffffe89504e470d0a	1a0a0000000d4948	...PNG..	.....IH
4452000001180000	008c080200000008	DR.....	.....
ec7edb0000059c49	444154789ceddd51	..~.....I	DATx...Q
6alc3b1440c13864	ff5b761610145038	j.;.@.8d	.[v...P8
3792ecaadf37afdd	eef141908bd43f7e	7.....7..	..A....?~



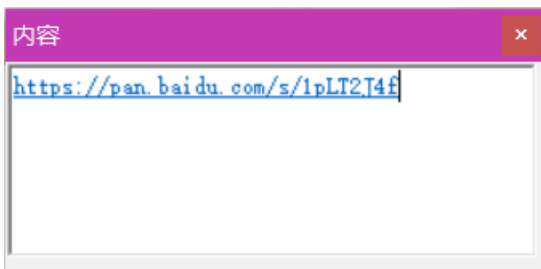
应该是一张png格式的图片，但是文件头不对，将前面多出来的 **ffe** 删去，保存得到半张二维码。



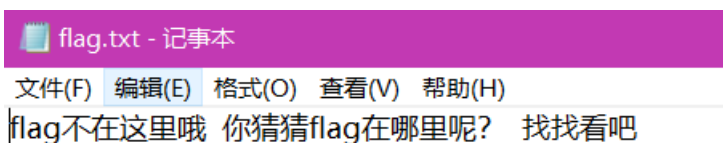
应该是高度做了手脚，修正即可。

补全后扫的时候发现扫不到。仔细观察发现颜色是反的，将其反色得到正确的二维码。

扫描完整的二维码，得到一个百度网盘的地址。



下载后得到 **flag.rar** 文件。解压得到**flag.txt**。然而...



按理说有三个提示，但是目前为止只用到两个，剩下的NTFS死活不知道有啥用。  
后来百度了大佬的writeup，才晓得还有一个**NTFS**交换数据流隐藏文件的骚操作。  
然后将隐藏的文件弄出来，是一个pyc文件。  
反编译为py文件。

```
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] #取从后向前（相反）的元素

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']
```

py文件是加密函数，写一个解密函数

```

import base64
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']
    flag = ""
    ciphertext.reverse()
    for i in range(len(ciphertext)):
        s = int(ciphertext[i])
        if i%2 ==0:
            s = s-10
        else:
            s = s+10
        flag += chr(i ^ s)
    print(flag)
decode()

```

得到flag

## 23.多彩(未做)

## 24.旋转跳跃

使用MP3Stego,使用方法百度。

```

C:\Users\... \Desktop\临时文件存放处\MP3Stego_1_1_19\MP3Stego>Decode.exe -X -P syclovergeek sycgeek-mp3.mp3
MP3StegoEncoder 1.1.19
See README file for copyright info
Input file = 'sycgeek-mp3.mp3' output file = 'sycgeek-mp3.mp3.pcm'
Will attempt to extract hidden information. Output: sycgeek-mp3.mp3.txt
the bit stream file sycgeek-mp3.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 5932]Avg slots/frame = 417.889; b/smp = 2.90; br = 127.979 kbps
Decoding of "sycgeek-mp3.mp3" is finished
The decoded PCM output file name is "sycgeek-mp3.mp3.pcm"

```

结果如下:

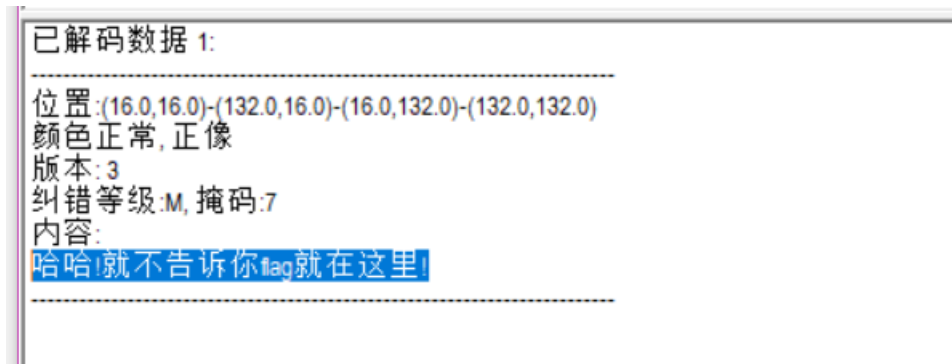
 sycgeek-mp3.mp3.txt	2020/2/7 23:55	文本文档	1 KB
 sycgeek-mp3.mp3.pcm	2020/2/7 23:55	PCM 文件	26,697 KB

打开sycgeek-mp3.mp3.txt即可得到flag。

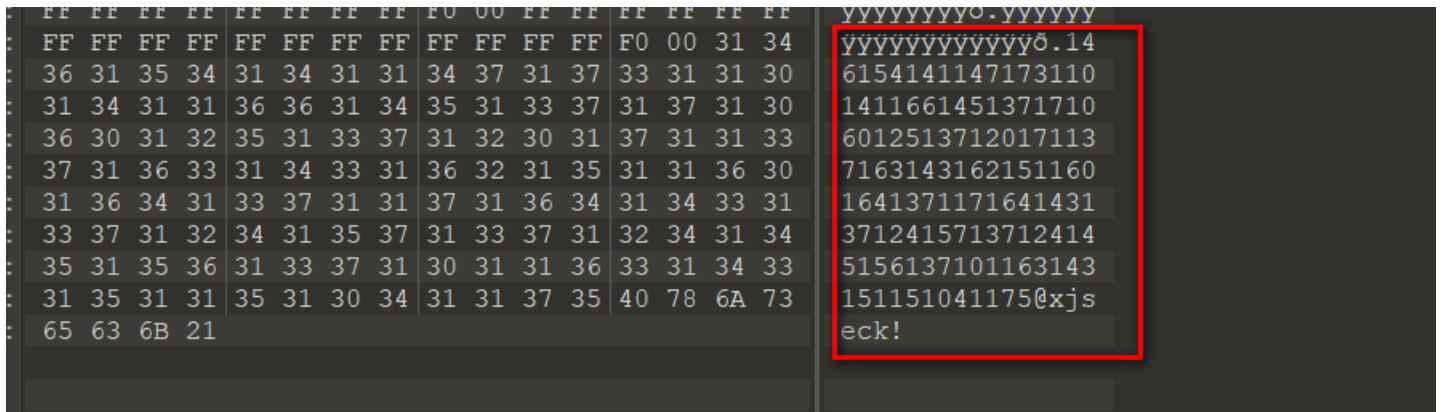
## 25.普通的二维码



解压缩得到一张二维码，扫码得：



010Editor打开二维码图片：



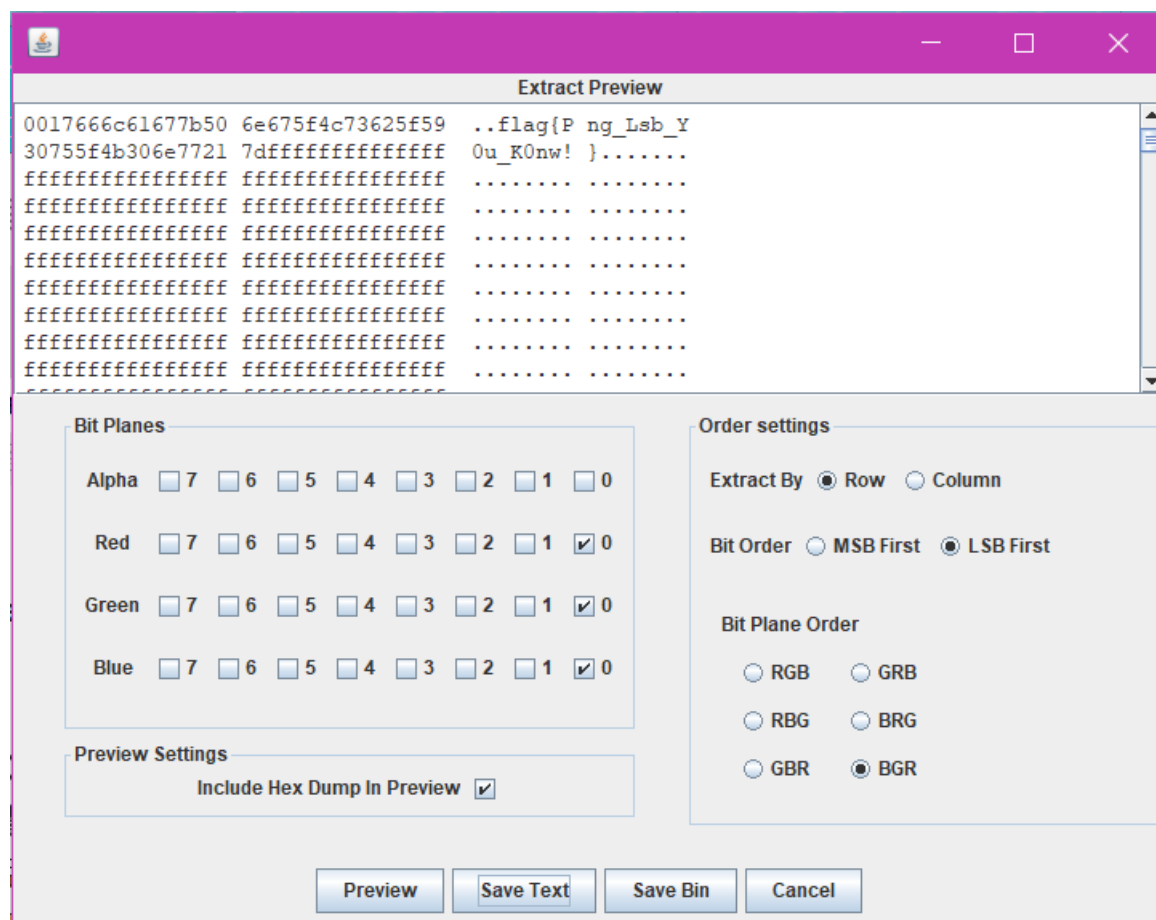
这一串数字最大为7，应该是8进制。跑脚本

```
with open(r"C:\Users\xxx\Desktop\CTF\25.txt", "r") as temp:  
    res = ""  
    for i in range(42):  
        s=eval('0o'+temp.read(3))  
        res+=chr(int(s))  
print(res)
```

得到flag

## 26.乌云邀请码

解压出图片后，查看属性、010Editor打开、binwalk、CRC校验高宽度。  
均无任何发现，可能是LSB 和 图层隐写。  
果然是LSB隐写



## 27.神秘的文件

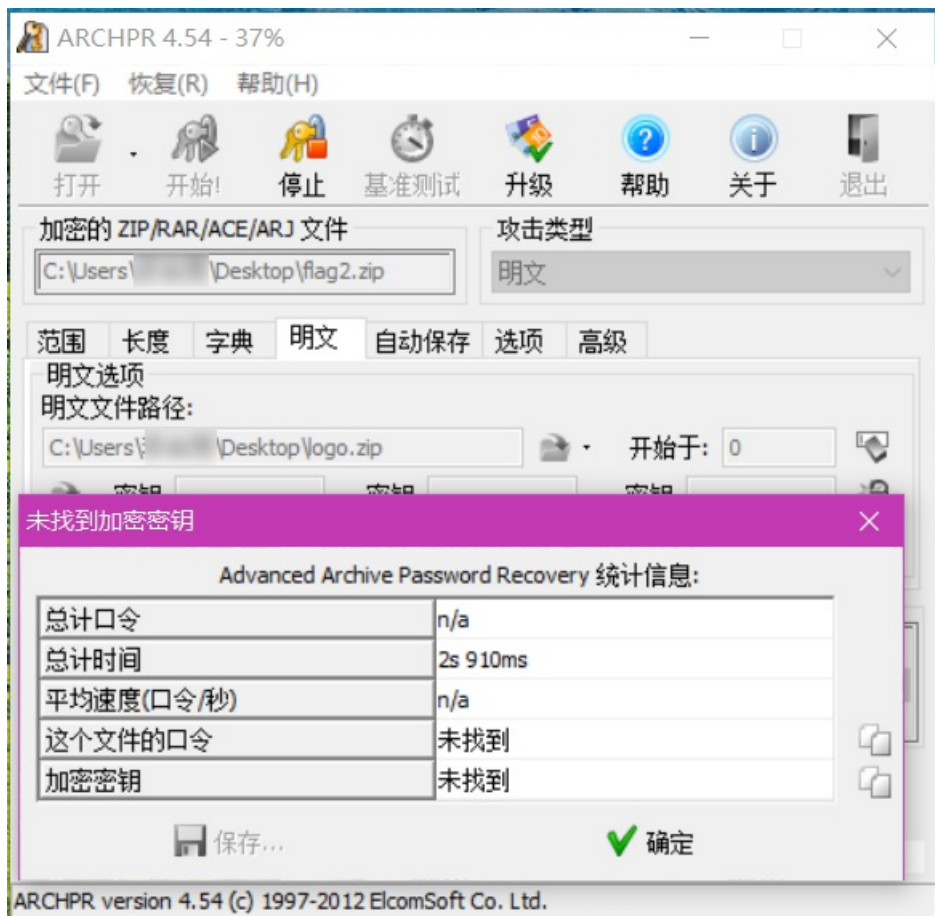
解压得到 **flag.zip** 和 **logo.png**

解压**flag.zip**发现它是加密的

预览发现**flag.zip** 包含一张相同的 **logo.png**

所以可以采用明文攻击破解密码

注意：明文攻击时采用的算法应该相同，并且明文攻击需要两个压缩包都只含一个文件

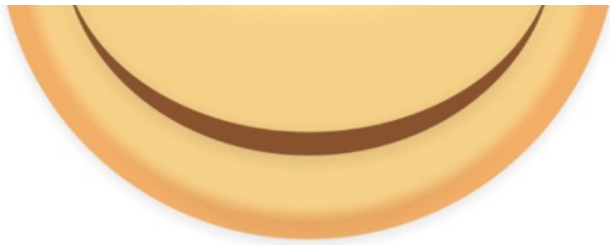


我的 ARCHPR 不可以得到密码。网上说是版本的原因

最终得到密码 q1w2e3r4

解压得到的一份word文档，文档里面是一张笑脸。





哪有什么 WriteUP，别想了，老老实实做题吧！

我们没发现什么有价值的东西。接下来我们把word文档放到 kali 里 binwalk 分析一下。

```
[root@kaliDesktop]#binwalk -e 233.docx
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	Zip archive data, at least v2.0 to extract, compressed size: 362, uncompressed size: 695, name: docProps/app.xml
672	0x2A0	Zip archive data, at least v2.0 to extract, compressed size: 387, uncompressed size: 773, name: docProps/core.xml
1370	0x55A	Zip archive data, at least v1.0 to extract, compressed size: 36452, uncompressed size: 36452, name: docProps/thumbnail.jpeg
37875	0x93F3	Zip archive data, at least v2.0 to extract, compressed size: 1285, uncompressed size: 4056, name: word/document.xml
39207	0x9927	Zip archive data, at least v2.0 to extract, compressed size: 476, uncompressed size: 1529, name: word/fontTable.xml
39731	0x9B33	Zip archive data, at least v1.0 to extract, compressed size: 222845, uncompressed size: 222845, name: word/media/image1.png
262627	0x401E3	Zip archive data, at least v2.0 to extract, compressed size: 1117, uncompressed size: 2847, name: word/settings.xml
263791	0x4066F	Zip archive data, at least v2.0 to extract, compressed size: 2920, uncompressed size: 29509, name: word/styles.xml
266756	0x41204	Zip archive data, at least v2.0 to extract, compressed size: 1512, uncompressed size: 6803, name: word/theme/theme1.xml
268319	0x4181F	Zip archive data, at least v2.0 to extract, compressed size: 287, uncompressed size: 529, name: word/webSettings.xml
268656	0x41970	Zip archive data, at least v2.0 to extract, compressed size: 266, uncompressed size: 949, name: word/_rels/document.xml.rels
269244	0x41BBC	Zip archive data, at least v2.0 to extract, compressed size: 362, uncompressed size: 1414, name: [Content_Types].xml
270175	0x41F5F	Zip archive data, at least v2.0 to extract, compressed size: 255, uncompressed size: 735, name: _rels/.rels
270991	0x4228F	Zip archive data, at least v2.0 to extract, compressed size: 34, uncompressed size: 32, name: docProps/flag.txt
272048	0x426B0	End of Zip archive, footer length: 22

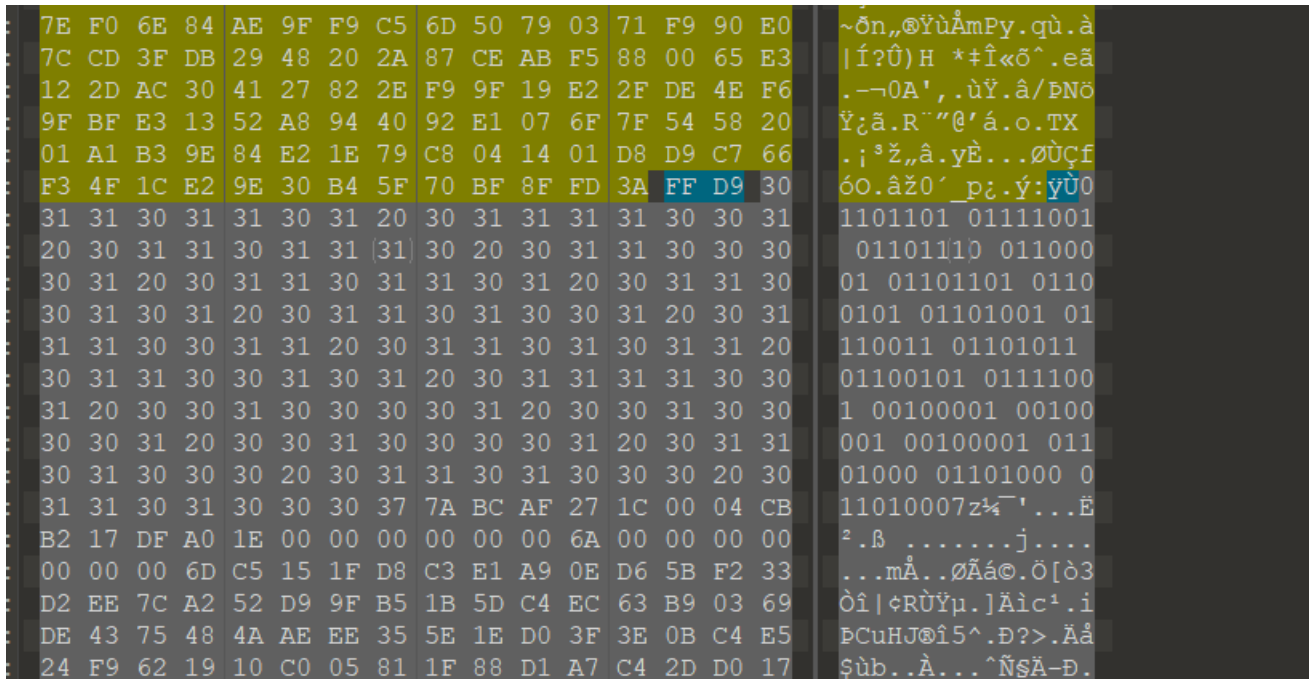
分离出来的文件加中有一份 **flag.txt** 文档。打开以后是一串BASE64加密的字符串。解密即可得到flag。

## 28.论剑



下载得到一张图片，属性 -> binwalk -> 颜色通道 -> LSB(均未得到线索)

010Editor打开。详细查看：



FF D9 是jpg文件的尾部。紧挨着是一串二进制数。将其转换为字符。

```
s = input("输入要转换的字符串：")
list = s.split()
str1 = '0b'
for i in list:
    str2 = str1 + i
    char1 = chr(eval(str2))
    print(char1,end = "")
```

```
=====
输入要转换的字符串: 01101101 01111001 01101110 01100001 01101101 01100101 011010
01 01110011 01101011 01100101 01111001 00100001 00100001 00100001 01101000 01101
000 01101000
mynameiskey!!!hhh
>>>
```

应该是有用的。翻到最

下面发现还是**FF D9**结尾。那应该还有一张图片。

搜索jpg的文件头。又发现了一张图片。然而这张图片并没有什么用。

后来实在是没有思路

看了大佬的WP，又了解到一个新的文件头(7z压缩包的头:37 7A BC AF 27 1C)

将原来错误的文件头38 7B BC AF 27 1C修改为正确的文件头，然后分离出来

再将7Z压缩包里的图片和原图片的高度修改可以发现：

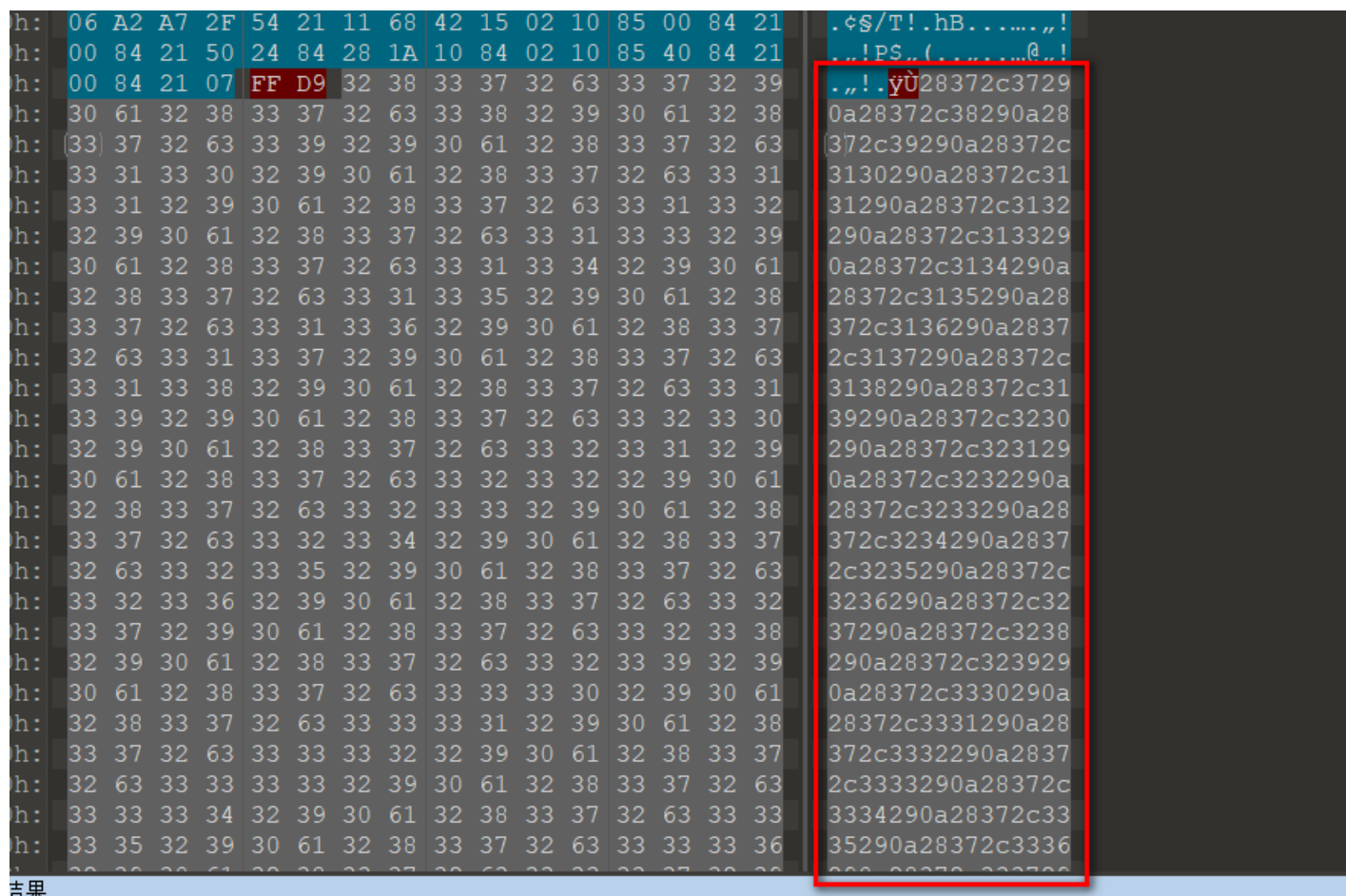


结合起来得到 not flag{666C61677B6D795F6E616D655F482121487D}

BASE16解密得到flag。

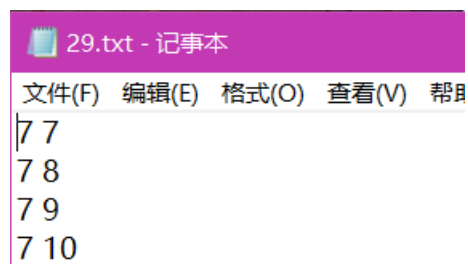
## 29.图穷匕见

010Editor打开下载的图片，可以发现后面是一串二进制文本。



将其转换为ASCII码

得到许多的坐标，应该是鼠标的流量数据包，转换为Gnuplot可以识别的格式









下载下来是一个txt文档，打开是一行行RGB值(共61366行)，每行代表一个像素点。

分解因数

数值	分解质因数 (结果)
61366	$2 * 61 * 503$

长和宽 就这三个数的组合。(例：长 **2**，宽 **30683**...)

跑一下脚本就可以还原出来图片。

跑脚本前需安装pillow库

python安装pillow(转载)

脚本(转载)

### 33.很普通的数独(ISCCCTF)

下载解压缩是25张数独图片

仔细观察发现这是同一道数独题。有的格子有数字，有的没有

有数字的代表1，没有的代表0。再把得到的数字用python画出二维码。

**1.png,5.png,21.png**仔细看看就是是二维码的定位形状，三个角上的方形块，但是按排列的画，这三个图的顺序不对，需要将图片1.png,5.png,21.png重命名成:5.png,21.png,1.png，然后把01提取出来：

111111010101000101000011111000010111111  
10000101100111101010011101100011001001000001  
101110101110011111010011111101000101001011101  
101110101101100010001010000011110001101011101  
10111010001110010000111110111111011101011101  
10000010110010000001100010000111010001000001  
11111110101010101010101010101010101110111111  
00000000001100110100100011010011001110000000  
11001110010010000111111100100101000000101111  
10100100101111111101110101011110101101001100  
100000111100100100000110001101001101010001010  
001100010011010001010011000100000010110010000  
010110101010001111110100011101001110101101111  
100011000100011100111011101101100101101110001  
001100110100000000010010000111100101101011010  
101000001011010111110011011111101001110100011  
110111110111011001101100010100001110000100000  
110101000010101000011101101101110101101001100  
010011111110001011111010001000011011101101100  
011001011001010101100011110101001100001010010  
010111111111010111111101101101111111111100  
011110001100000100001000101000100100100011110  
111110101110011100111010110100110100101010010  
110010001011101011101000111100000011100010000  
1010111101110011101111111100001010111110010  
110100011000111000100111101101111101000100010  
111101111110001001000011010110001111110111110  
011001010101000110010100010001000101101010001  
011101110101101101100100001101101000111101001  
110110001001101100010101101111110100101100110  
000011100111000000000100001010101111100010010  
111010010011110011101110010100001011111010010  
101001100010111111110100000100001010101010100  
000010011001001101110101001111100101111101101  
000010111101110001101011000001000101110100110  
011110011010100010100000011011000001110010000  
10011010010000110111111101100101110111110011  
000000001111110101101000101011100100100011010  
111111100011111011011010101101110011101011110  
100000101110101101101000111110010001100010001  
10111010101110000111111101101001000111111011  
101110100110111101101000001001101100011101101  
101110100000011101100001101010110010010010001  
100000101011001011111011001011000011010110000  
111111101010101001111011110101101110000101101

写脚本把图画出来:

```

from PIL import Image
x = 45
y = 45
im = Image.new("RGB", (x,y)) # 创建图片
with open("1.txt") as file:
    list1 = file.readline()
    for i in range(x):
        for j in range(y):
            if list1[i * x + j] == '1':
                im.putpixel((i,j), (0, 0, 0))
            else :
                im.putpixel((i,j), (255, 255, 255))
im.show()

```

扫码得到Base64密文

```

Vm0xd1NtUXlWa1pPVI doVFIUSINjRIJVVGtOamJGWnIWMjFHVIUxV1ZqTldNakZlWVcxS1lxTnNhRmhoTVZweVdWUkdXbVZHWkhOWGJGc
HBWa1paZWxacIpEUmhNVXBYVW14V2FHVnFRVGs9

```

解密好几次后得到flag。

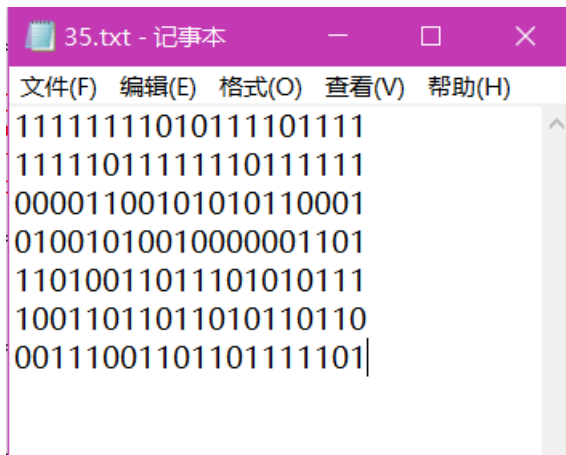
## 34.PEN\_AND\_APPLE(未做)

## 35.color

下载解压得到七张图片

用tweakpng打开得到的图片，CRC校验出错，改一下高度。(其实在每张图片的颜色通道有提示信息，连起来就是**Make Me Tall**把图片加高)

可以看到下面有黑白相间的小方块，记录下来(黑色代表1，白色代表0)



后面我就想不通20个二进制可以表示个啥？

万万没想到是竖着看(所以不要被思维局限 0.0)，跑个脚本就可以得到flag了。

```
row1 = '11111111010111101111'
row2 = '11111011111110111111'
row3 = '00001100101010110001'
row4 = '01001010010000001101'
row5 = '11010011011101010111'
row6 = '10011011011010110110'
row7 = '00111001101101111101'
for i in range(0, 20):
    str1 = row1[i] + row2[i] + row3[i] + row4[i] + row5[i] + row6[i] + row7[i]
    print(chr(int(str1, 2)),end = '')
```

## 36.怀疑人生

解压得到一个压缩包、一张图片、一张二维码

- 二维码扫码得——12580}
- 图片用010Editor 打开在末尾可以发现一个zip文件  
解压得到txt文件是Ook密文。解密得到3oD54e(base58加密，太难了 0.0)。  
再次解密得misc
- 压缩包有密码，咋都想不到密码，看了WP，密码是password  
压缩包里是一串base64密文，解密得到 Unicode编码  
\\u66\\u6c\\u61\\u67\\u7b\\u68\\u61\\u63\\u6b\\u65\\u72  
之后再解码得flag{hacker

拼接起来flag{hackermisc12580}

## 37.红绿灯(未做)

## 38.不简单的压缩包

大佬的Write up

## 39.一枝独秀

大佬的Write up

## 40.小猪佩奇

思路

(写给自己：注意python会将换行符当作一个字符。跑字典时要注意)

## 41.好多压缩包

## 42.一个普通的压缩包 (xp0intCTF) \*\*

## 43.2B

## 44.QAQ

## 45.apple

## 46.妹子的陌陌

## 47.就五层你能解开吗