攻防世界--MISC高手进阶区（持续更新）

原创

IMWJING 于 2020-03-26 14:30:32 发布 6862 收藏 21

分类专栏： CTF

本文链接：https://blog.csdn.net/qq_43081170/article/details/104989305

版权

CTF 专栏收录该内容

5 篇文章 0 订阅

订阅专栏

目录

base64÷4

base16解码：`https://www.qqxiuzi.cn/bianma/base.php?type=16`

wireshark-1

下载附件解压缩后为一个.pcap的数据包，用wireshark打开，筛选HTTP数据包，找到post请求包，追踪TCP流找到password字段得到flag。

Training-Stegano-1

下载附件为一个图片，图片特别小，直接用winhex打开，看到passwd:steganoI，直接提交steganoI，成功。

János-the-Ripper

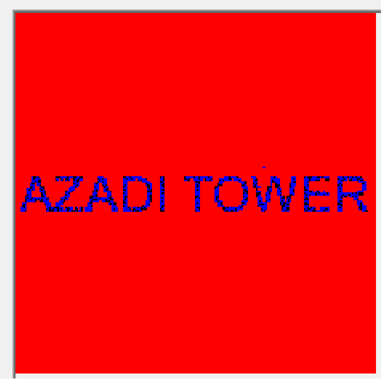
下载附件解压缩为一个“misc100”的文件，用winhex打开，文件头为PK，文件为一个zip压缩包，添加.zip后缀，解压缩需要密码，文件中无密码提示，采用暴力破解，得出密码为“fish”，解压缩后得到flag。

Test-flag-please-ignore

下载附件解压缩为一个“misc10”的文件，直接打开为“666c61677b68656c6c6f5f776f726c647d”，猜想为16进制转字符串，通过`https://tool.lu/hexstr`进行转换得到flag。

What-is-this

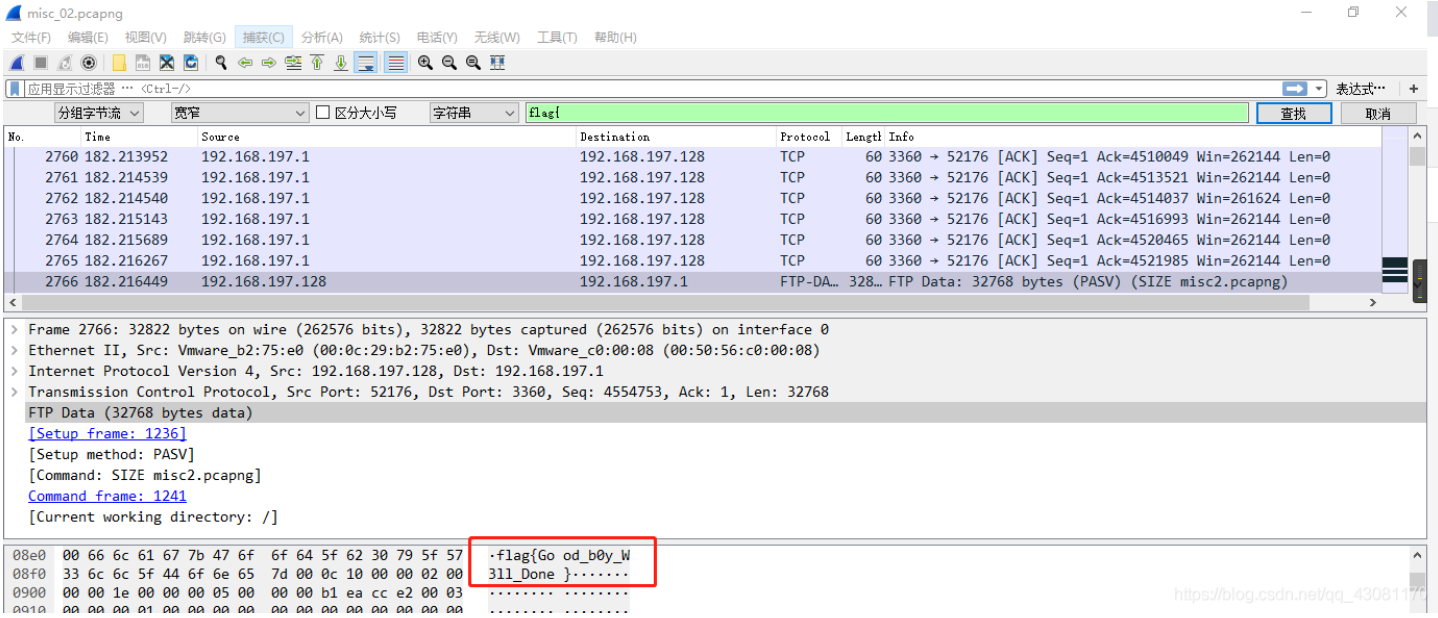
下载附件解压缩得到两个图片，将两张图片进行合并，



得到flag。

embarrass

下载附件解压缩为.pcapng的文件，用wireshark打开，按照分组字节流搜索字符串flag{得到flag。


misc_02.pcapng
文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)
应用显示过滤器 … <Ctrl-/>
表达式…
查找
取消

分组字节流　宽窄　□ 区分大小写　字符串　flag{

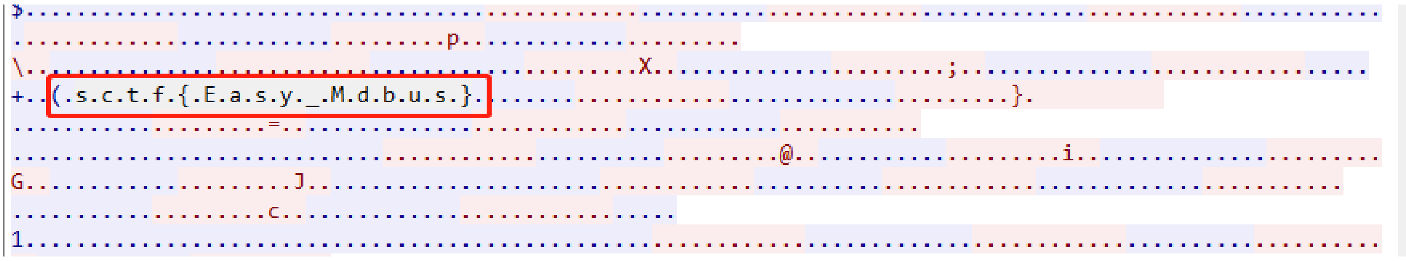
```
No.   Time        Source           Destination      Protocol  Length Info
 2760 182.213952  192.168.197.1    192.168.197.128  TCP       60 3360 → 52176 [ACK] Seq=1 Ack=4510049 Win=262144 Len=0
 2761 182.214539  192.168.197.1    192.168.197.128  TCP       60 3360 → 52176 [ACK] Seq=1 Ack=4513521 Win=262144 Len=0
 2762 182.214540  192.168.197.1    192.168.197.128  TCP       60 3360 → 52176 [ACK] Seq=1 Ack=4514037 Win=261624 Len=0
 2763 182.215143  192.168.197.1    192.168.197.128  TCP       60 3360 → 52176 [ACK] Seq=1 Ack=4516993 Win=262144 Len=0
 2764 182.215689  192.168.197.1    192.168.197.128  TCP       60 3360 → 52176 [ACK] Seq=1 Ack=4520465 Win=262144 Len=0
 2765 182.216267  192.168.197.1    192.168.197.128  TCP       60 3360 → 52176 [ACK] Seq=1 Ack=4521985 Win=262144 Len=0
 2766 182.216449  192.168.197.128  192.168.197.1    FTP-DA…   328… FTP Data: 32768 bytes (PASV) (SIZE misc2.pcapng)
```

```
> Frame 2766: 32822 bytes on wire (262576 bits), 32822 bytes captured (262576 bits) on interface 0
> Ethernet II, Src: Vmware_b2:75:e0 (00:0c:29:b2:75:e0), Dst: Vmware_c0:00:08 (00:50:56:c0:00:08)
> Internet Protocol Version 4, Src: 192.168.197.128, Dst: 192.168.197.1
> Transmission Control Protocol, Src Port: 52176, Dst Port: 3360, Seq: 4554753, Ack: 1, Len: 32768
  FTP Data (32768 bytes data)
  [Setup frame: 1236]
  [Setup method: PASV]
  [Command: SIZE misc2.pcapng]
  Command frame: 1241
  [Current working directory: /]
```

```
08e0  00 66 6c 61 67 7b 47 6f  6f 64 5f 62 30 79 5f 57   ·flag{Go od_b0y_W
08f0  33 6c 6c 5f 44 6f 6e 65  7d 00 0c 10 00 00 02 00   3ll_Done }·······
0900  00 00 1e 00 00 00 05 00  00 00 b1 ea cc e2 00 03   ········ ········
0910  00 00 00 01 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
```

神奇的Modbus

下载附件是一个后缀为.pcapng的文件，用wireshark打开，根据题目“神奇的Modbus”搜索协议为”Modbus/TCP“协议的数据流，寻找flag无果，追踪TCP流，看到sctf{Easy_Mdbus}

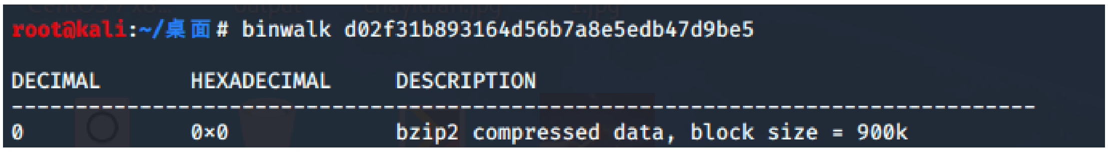


```
Þ...........................................................................................................
...................................p.....................
\.................................................X.......................;................................
+..(.s.c.t.f.{.E.a.s.y._.M.d.b.u.s.}...........................................}.
....................=..................................................
.............................................................@.....................i........................
G.....................J..................................................................................
....................c...............................
1...........................................................................................................
```

直接提交显示错误，根据modbus猜想flag为sctf{Easy_Modbus}，提交成功。

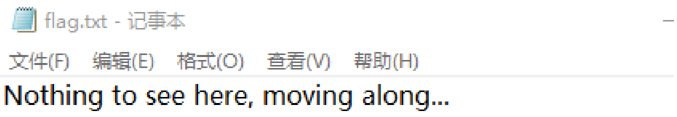
MISCall

下载附件，是一个不知道什么后缀的文件，放进kali，binwalk查看文件信息，是一个zip压缩包，



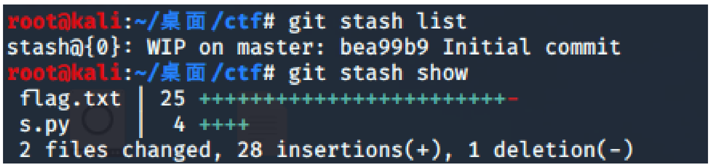
```
root@kali:~/桌面# binwalk d02f31b893164d56b7a8e5edb47d9be5

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             bzip2 compressed data, block size = 900k
```

解压缩得到一个`ctf`的文件夹，文件夹里有两个文件，一个`.git`的文件夹和一个`flag.txt`的文本文档，打开`flag.txt`没找到flag，



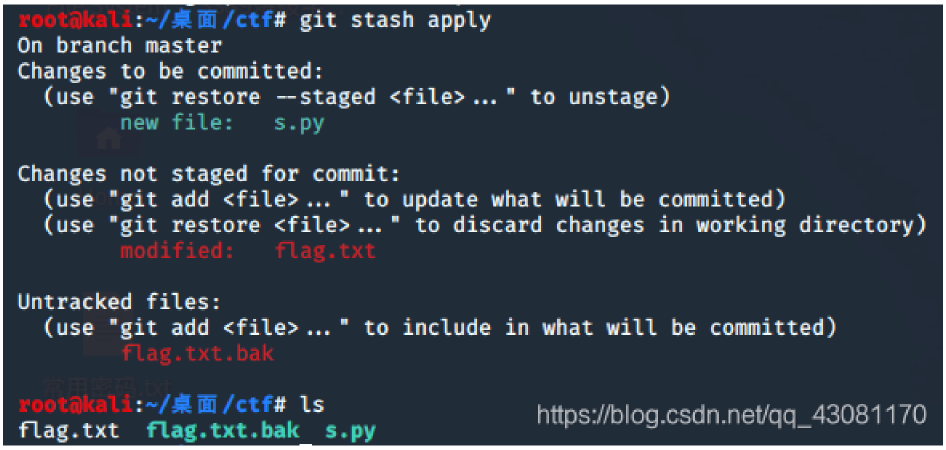
Nothing to see here, moving along...

只能从`.git`下手，使用`git stash list：查看stash了哪些存储`
`git stash show：显示做了哪些改动`；



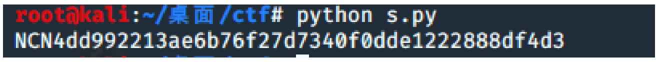
```
root@kali:~/桌面/ctf# git stash list
stash@{0}: WIP on master: bea99b9 Initial commit
root@kali:~/桌面/ctf# git stash show
 flag.txt | 25 ++++++++++++++++++++++++-
 s.py     |  4 ++++
 2 files changed, 28 insertions(+), 1 deletion(-)
```

`git stash apply：:应用某个存储,恢复之前的存储，但不会把存储从存储列表中删除`（将原来的flag.txt重命名或删除）。



```
root@kali:~/桌面/ctf# git stash apply
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   s.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   flag.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        flag.txt.bak

root@kali:~/桌面/ctf# ls
flag.txt  flag.txt.bak  s.py
```

运行s.py得到flag。



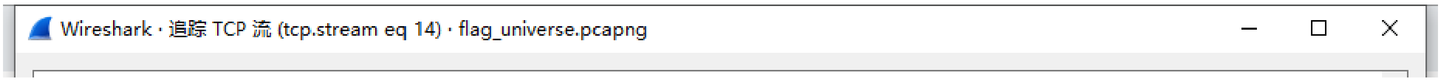
```
root@kali:~/桌面/ctf# python s.py
NCN4dd992213ae6b76f27d7340f0dde1222888df4d3
```

flag_universe

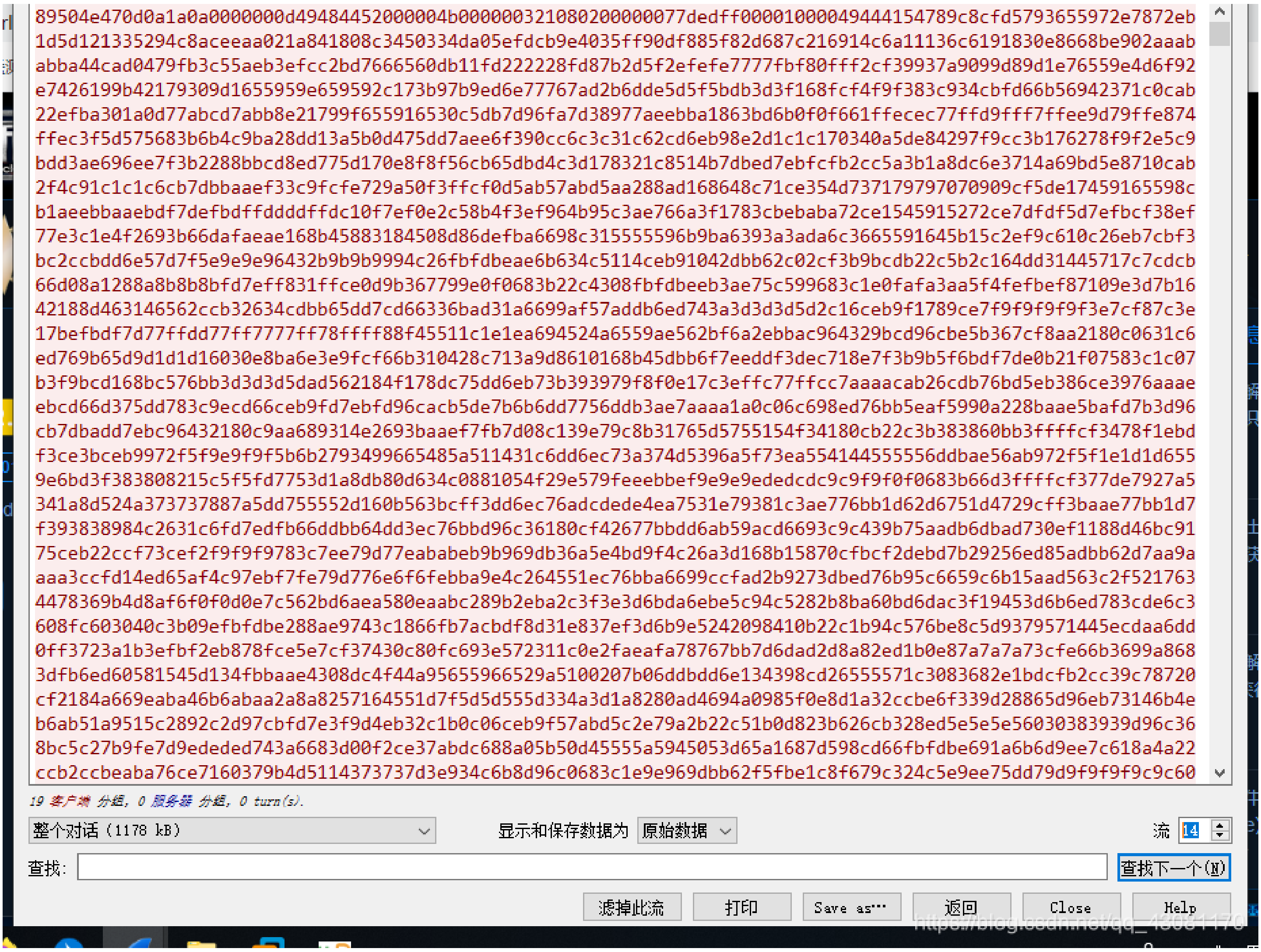
下载附件解压缩是一个pcapng文件，用wireshark打开，追踪TCP数据流，

```
ZmxhZ3tUaGlzIGlzIGZha2UgZmxhZyBoYWhhaGF9
```

base64解密后flag{This is fake flag hahaha}，是一个假的flag，继续追踪TCP数据流，universe.png中也没有flag，当追踪到第14个TCP数据流时，有一个new_universe.png，将原始数据保存到txt文本文档中，



Wireshark · 追踪 TCP 流 (tcp.stream eq 14) · flag_universe.pcapng



```
89504e470d0a1a0a0000000d49484452000004b00000032108020000077dedff00001000049444154789c8cfd5793655972e7872eb
1d5d121335294c8aceeaa021a841808c3450334da05efdcb9e4035ff90df885f82d687c216914c6a11136c6191830e8668be902aaab
abba44cad0479fb3c55aeb3efcc2bd7666560db11fd222228fd87b2d5f2efefe7777fbf80fff2cf39937a9099d89d1e76559e4d6f92
e7426199b42179309d1655959e659592c173b97b9ed6e77767ad2b6dde5d5f5bdb3d3f168fcf4f9f383c934cbfd66b56942371c0cab
22efba301a0d77abcd7abb8e21799f655916530c5db7d96fa7d38977aeebba1863bd6b0f0f661ffecec77ffd9fff7ffee9d79ffe874
ffec3f5d575683b6b4c9ba28dd13a5b0d475dd7aee6f390cc6c3c31c62cd6eb98e2d1c1c170340a5de84297f9cc3b176278f9f2e5c9
bdd3ae696ee7f3b2288bbcd8ed775d170e8f8f56cb65dbd4c3d178321c8514b7dbed7ebfcfb2cc5a3b1a8dc6e3714a69bd5e8710cab
2f4c91c1c1c6cb7dbbaaef33c9fcfe729a50f3ffcf0d5ab57abd5aa288ad168648c71ce354d737179797070909cf5de17459165598c
b1aeebbaaebdf7defbdffddddffdc10f7ef0e2c58b4f3ef964b95c3ae766a3f1783cbebaba72ce1545915272ce7dfdf5d7efbcf38ef
77e3c1e4f2693b66dafaeae168b45883184508d86defba6698c315555596b9ba6393a3ada6c3665591645b15c2ef9c610c26eb7cbf3
bc2ccbdd6e57d7f5e9e9e96432b9b9b9994c26fbfdbeae6b634c5114ceb91042dbb62c02cf3b9bcdb22c5b2c164dd31445717c7cdcb
66d08a1288a8b8b8bfd7eff831ffce0d9b367799e0f0683b22c4308fbfdbeeb3ae75c599683c1e0fafa3aa5f4fefbef87109e3d7b16
42188d463146562ccb32634cdbb65dd7cd66336bad31a6699af57addb6ed743a3d3d3d5d2c16ceb9f1789ce7f9f9f9f9f3e7cf87c3e
17befbdf7d77ffdd77ff7777ff78ffff88f45511c1e1ea694524a6559ae562bf6a2ebbac964329bcd96cbe5b367cf8aa2180c0631c6
ed769b65d9d1d1d16030e8ba6e3e9fcf66b310428c713a9d8610168b45dbb6f7eeddf3dec718e7f3b9b5f6bdf7de0b21f07583c1c07
b3f9bcd168bc576bb3d3d3d5dad562184f178dc75dd6eb73b393979f8f0e17c3effc77ffcc7aaaacab26cdb76bd5eb386ce3976aaae
ebcd66d375dd783c9ecd66ceb9fd7ebfd96cacb5de7b6b6dd7756ddb3ae7aaaa1a0c06c698ed76bb5eaf5990a228baae5bafd7b3d96
cb7dbadd7ebc96432180c9aa689314e2693baaef7fb7d08c139e79c8b31765d5755154f34180cb22c3b383860bb3ffffcf3478f1ebd
f3ce3bceb9972f5f9e9f9f5b6b2793499665485a511431c6dd6ec73a374d5396a5f73ea554144555556ddbae56ab972f5f1e1d1d655
9e6bd3f383808215c5f5fd7753d1a8db80d634c0881054f29e579feeebbef9e9e9ededcdc9c9f9f0f0683b66d3ffffcf377de7927a5
341a8d524a373737887a5dd755552d160b563bcff3dd6ec76adcdede4ea7531e79381c3ae776bb1d62d6751d4729cff3baae77bb1d7
f393838984c2631c6fd7edfb66ddbb64dd3ec76bbd96c36180cf42677bbdd6ab59acd6693c9c439b75aadb6dbad730ef1188d46bc91
75ceb22ccf73cef2f9f9f9783c7ee79d77eababeb9b969db36a5e4bd9f4c26a3d168b15870cfbcf2debd7b29256ed85adbb62d7aa9a
aaa3ccfd14ed65af4c97ebf7fe79d776e6f6febba9e4c264551ec76bba6699ccfad2b9273dbed76b95c6659c6b15aad563c2f521763
4478369b4d8af6f0f0d0e7c562bd6aea580eaabc289b2eba2c3f3e3d6bda6ebe5c94c5282b8ba60bd6dac3f19453d6b6ed783cde6c3
608fc603040c3b09efbfdbe288ae9743c1866fb7acbdf8d31e837ef3d6b9e5242098410b22c1b94c576be8c5d9379571445ecdaa6dd
0ff3723a1b3efbf2eb878fce5e7cf37430c80fc693e572311c0e2faeafa78767bb7d6dad2d8a82ed1b0e87a7a7a73cfe66b3699a868
3dfb6ed60581545d134fbbaae4308dc4f44a95655966529a5100207b06ddbdd6e134398cd26555571c3083682e1bdcfb2cc39c78720
cf2184a669eaba46b6abaa2a8a8257164551d7f5d5d555d334a3d1a8280ad4694a0985f0e8d1a32ccbe6f339d28865d96eb73146b4e
b6ab51a9515c2892c2d97cbfd7e3f9d4eb32c1b0c06ceb9f57abd5c2e79a2b22c51b0d823b626cb328ed5e5e5e56030383939d96c36
8bc5c27b9fe7d9ededed743a6683d00f2ce37abdc688a05b50d45555a5945053d65a1687d598cd66fbfdbe691a6b6d9ee7c618a4a22
ccb2ccbeaba76ce7160379b4d5114373737d3e934c6b8d96c0683c1e9e969dbb62f5fbe1c8f679c324c5e9ee75dd79d9f9f9f9c9c60
```

19 客户端 分组, 0 服务器 分组, 0 turn(s).

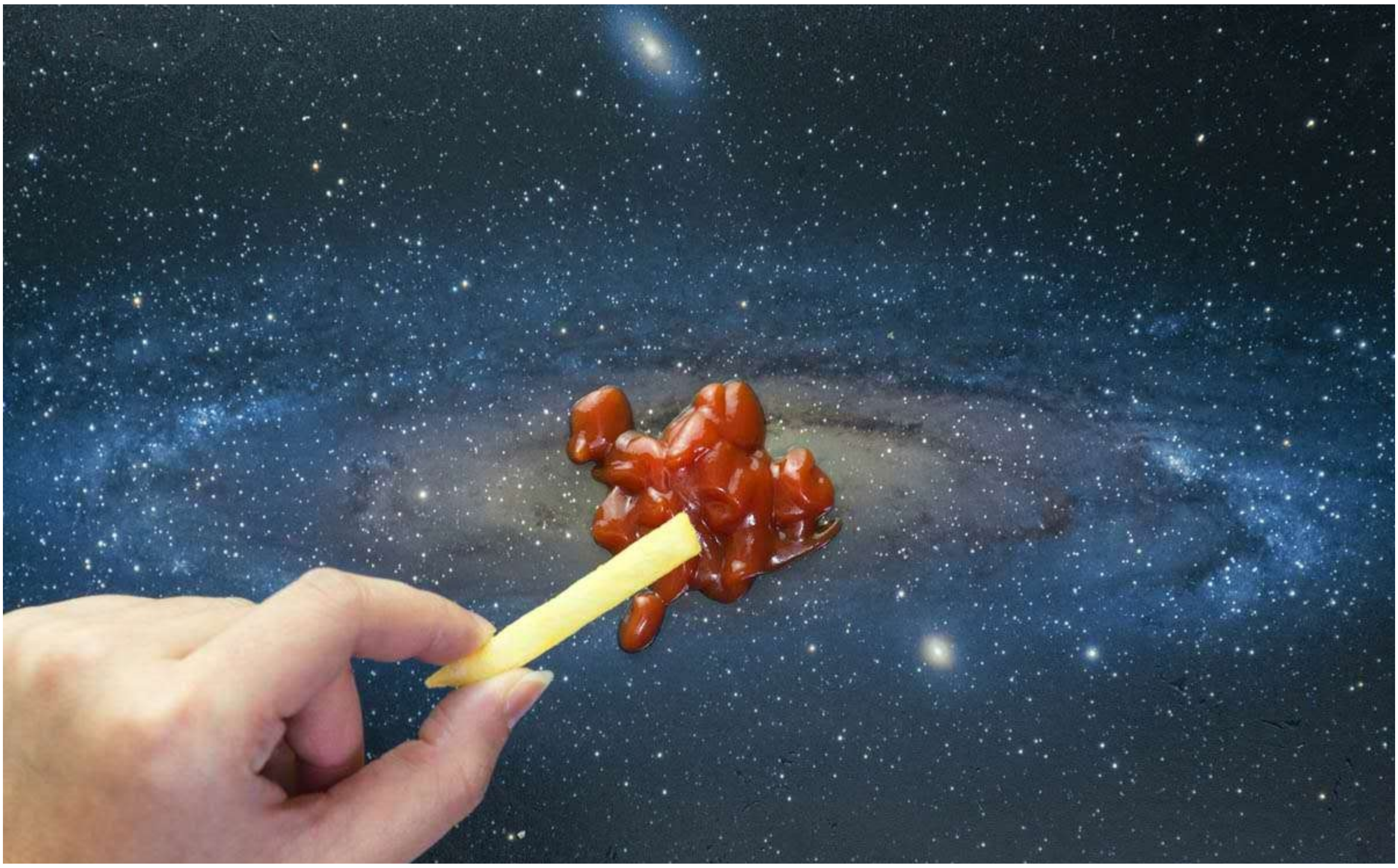
整个对话 (1178 kB)

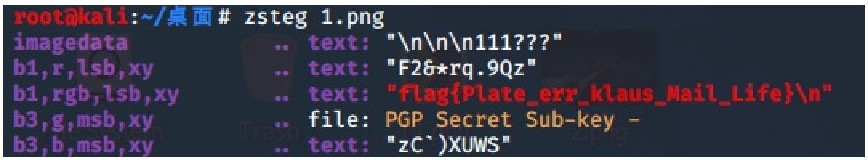
显示和保存数据为 原始数据

流 14

查找:

再复制到winhex中保存为png图片，利用zsteg命令得到flag。

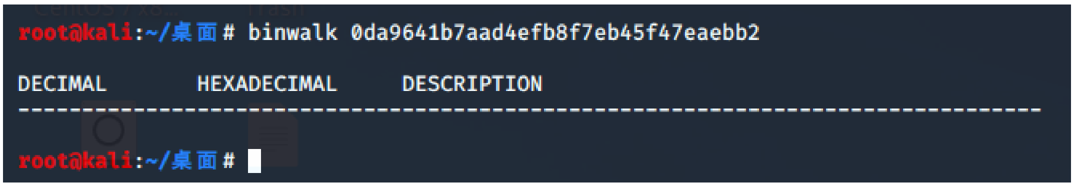




```
root@kali:~/桌面# zsteg 1.png
imagedata           .. text: "\n\n\n111???"
b1,r,lsb,xy         .. text: "F2&*rq.9Qz"
b1,rgb,lsb,xy       .. text: "flag{Plate_err_klaus_Mail_Life}\n"
b3,g,msb,xy         .. file: PGP Secret Sub-key -
b3,b,msb,xy         .. text: "zC`)XUWS"
```

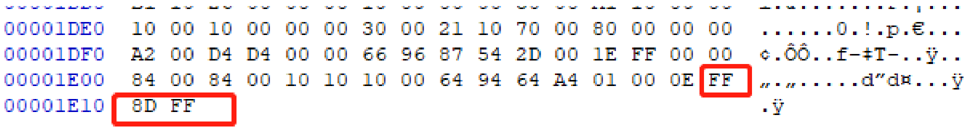
Reverse-it

下载附件，用binwalk分析文件，无隐藏文件



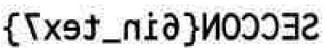
```
root@kali:~/桌面# binwalk 0da9641b7aad4efb8f7eb45f47eaebb2

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------

root@kali:~/桌面# 
```

用winhex分析，文件末尾时jpg文件头的倒序，根据题目reverse-it，将文件内容进行反转，

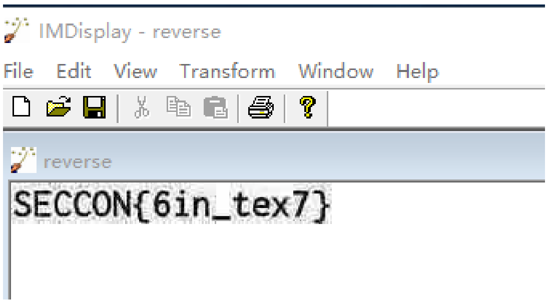


```
00001DE0  10 00 10 00 00 00 30 00 21 10 70 00 80 00 00 00  ......0.!.p.€...
00001DF0  A2 00 D4 D4 00 00 66 96 87 54 2D 00 1E FF 00 00  ¢.ÔÔ..f–‡T-..ÿ..
00001E00  84 00 84 00 10 10 10 00 64 94 64 A4 01 00 0E FF  „.„.....d”d¤...ÿ
00001E10  8D FF                                            .ÿ
```

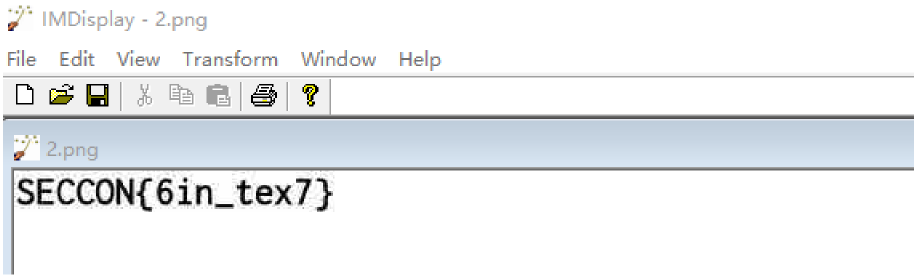
通过Linux命令行工具将文件内容进行反转，
`xxd -p 原文件| tr -d '\n' | rev | xxd -r -p > 反转后的文件名`，得到一张图片，



再利用ImageMagick工具将图片进行反转，得到flag。



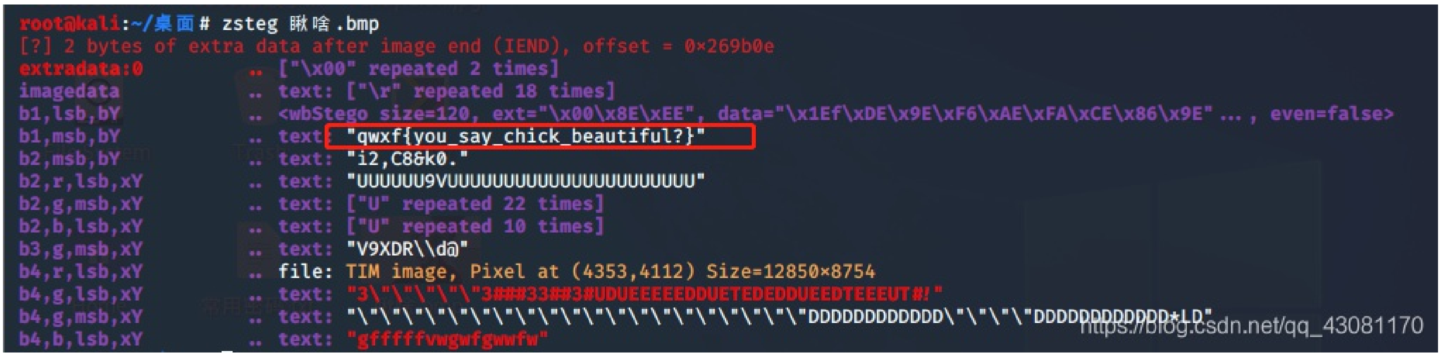
SECCON{6in_tex7}



SECCON{6in_tex7}

打野

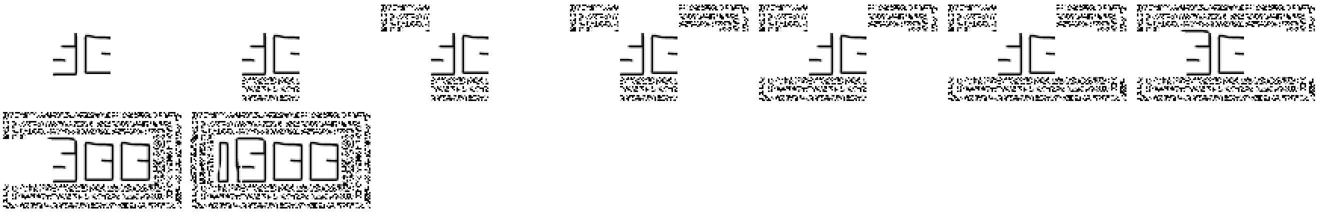
下载附件解压缩是一个.bmp的文件，放到kali，使用`zsteg 愁啥.bmp`，得到flag。



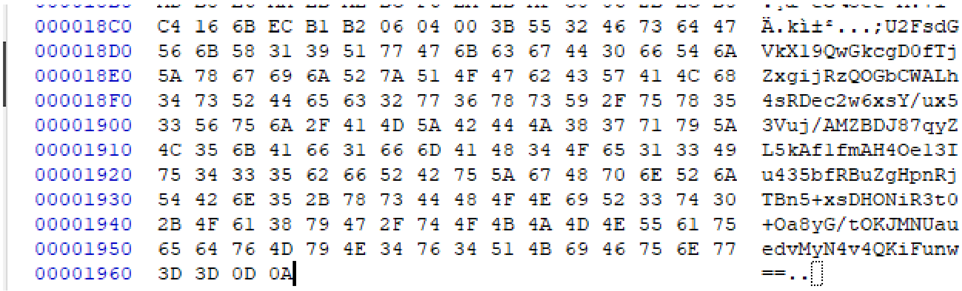
```
root@kali:~/桌面# zsteg 瞅啥.bmp
[?] 2 bytes of extra data after image end (IEND), offset = 0x269b0e
extradata:0         .. ["\x00" repeated 2 times]
imagedata           .. text: ["\r" repeated 18 times]
b1,lsb,bY           .. <wbStego size=120, ext="\x00\x8E\xEE", data="\x1Ef\xDE\x9E\xF6\xAE\xFA\xCE\x86\x9E"... , even=false>
b1,msb,bY           .. text: "qwxf{you_say_chick_beautiful?}"
b2,msb,bY           .. text: "i2,C8&k0."
b2,r,lsb,xY         .. text: "UUUUUU9VUUUUUUUUUUUUUUUUUUUUUU"
b2,g,msb,xY         .. text: ["U" repeated 22 times]
b2,b,lsb,xY         .. text: ["U" repeated 10 times]
b3,g,msb,xY         .. text: "V9XDR\\d@"
b4,r,lsb,xY         .. file: TIM image, Pixel at (4353,4112) Size=12850x8754
b4,g,lsb,xY         .. text: "3\"\"\"\"3###33##3#UDUEEEEEDDUETEDEDDUEEDTEEEUT#!"
b4,g,msb,xY         .. text: "\"\"\"\"\"\"\"\"\"\"\"\"\"\"\"\"DDDDDDDDDDDD\"\"\"DDDDDDDDDDDD*LD"
b4,b,lsb,xY         .. text: "gffffvwgwfgwwfw"
```

Aesop_secret

下载附件解压缩，打开是一张动态图片图片时分块显示的，将图片进行分解，拼图，图片在线分解：`https://tu.sioe.cn/gj/fenjie/`得到ISCC。

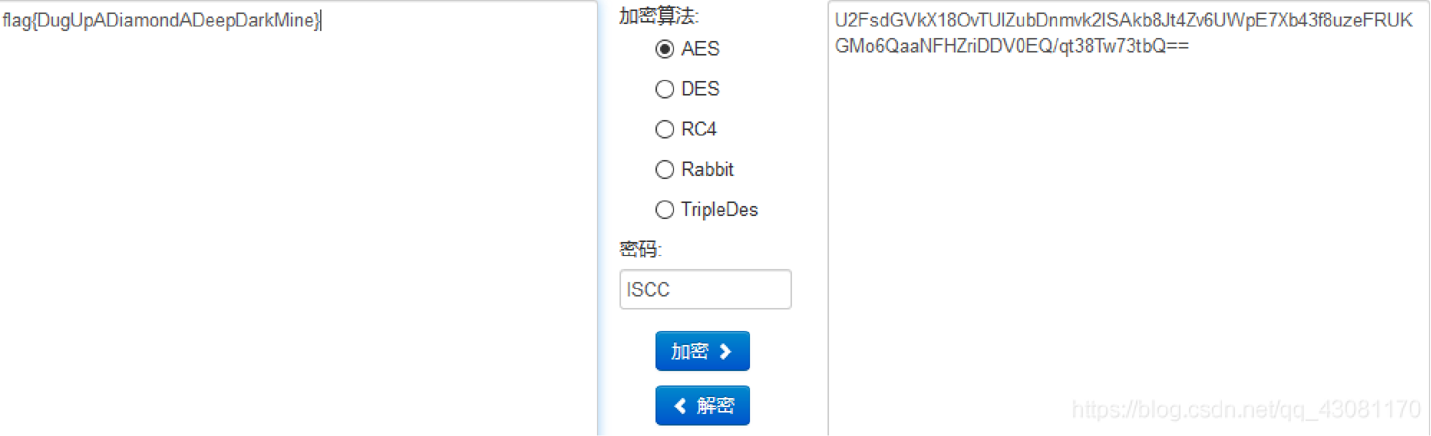


将图片放进winhex中，看到一串字符串，



```
000018C0  C4 16 6B EC B1 B2 06 04 00 3B 55 32 46 73 64 47  Ä.kì±²...;U2FsdG
000018D0  56 6B 58 31 39 51 77 47 6B 63 67 44 30 66 54 6A  VkX19QwGkcgD0fTj
000018E0  5A 78 67 69 6A 52 7A 51 4F 47 62 43 57 41 4C 68  ZxgijRzQOGbCWALh
000018F0  34 73 52 44 65 63 32 77 36 78 73 59 2F 75 78 35  4sRDec2w6xsY/ux5
00001900  33 56 75 6A 2F 41 4D 5A 42 44 4A 38 37 71 79 5A  3Vuj/AMZBDJ87qyZ
00001910  4C 35 6B 41 66 31 66 6D 41 48 34 4F 65 31 33 49  L5kAf1fmAH4Oe13I
00001920  75 34 33 35 62 66 52 42 75 5A 67 48 70 6E 52 6A  u435bfRBuZgHpnRj
00001930  54 42 6E 35 2B 78 73 44 48 4F 4E 69 52 33 74 30  TBn5+xsDHONiR3t0
00001940  2B 4F 61 38 79 47 2F 74 4F 4B 4A 4D 4E 55 61 75  +Oa8yG/tOKJMNUau
00001950  65 64 76 4D 79 4E 34 76 34 51 4B 69 46 75 6E 77  edvMyN4v4QKiFunw
00001960  3D 3D 0D 0A                                      ==..
```

根据题目Aesop_secret为aes加密，图片分解拼图的ISCC是解密的密码，解密两次得到flag。



flag{DugUpADiamondADeepDarkMine}

● AES
○ DES
○ RC4
○ Rabbit
○ TripleDes

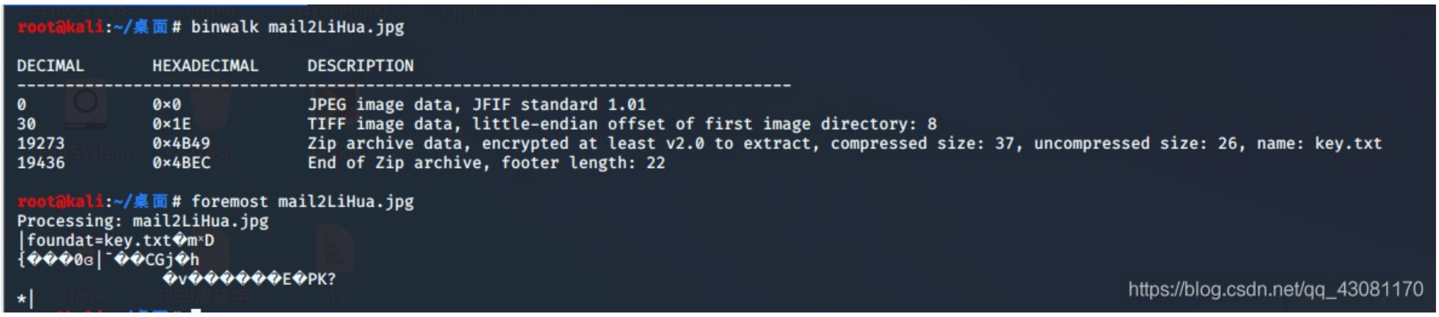
密码: ISCC

U2FsdGVkX18OvTUlZubDnmwk2ISAkb8Jt4Zv6UWpE7Xb43f8uzeFRUKGMo6QaaNFHZriDDV0EQ/qt38Tw73tbQ==

再见李华

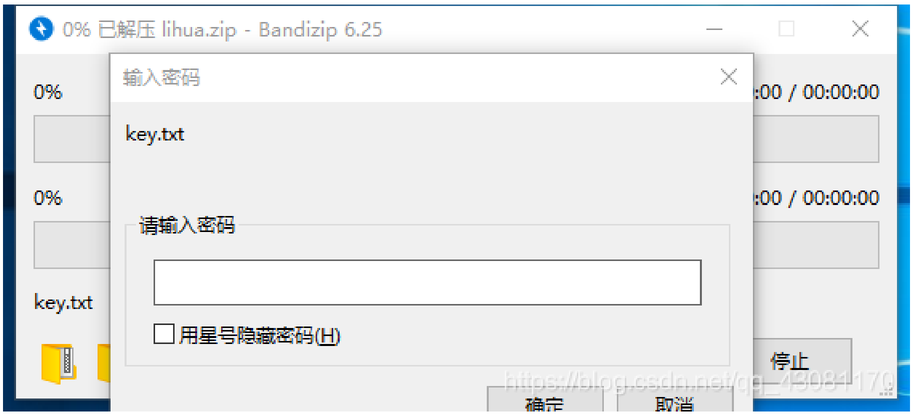
下载附件解压缩是一个图片，图片中有一个MD5值，进行MD5解密，解不出来，

将图片放进kali中，利用binwalk看到图片中有一个压缩包，利用foremost进行文件分离，得到一个压缩包，

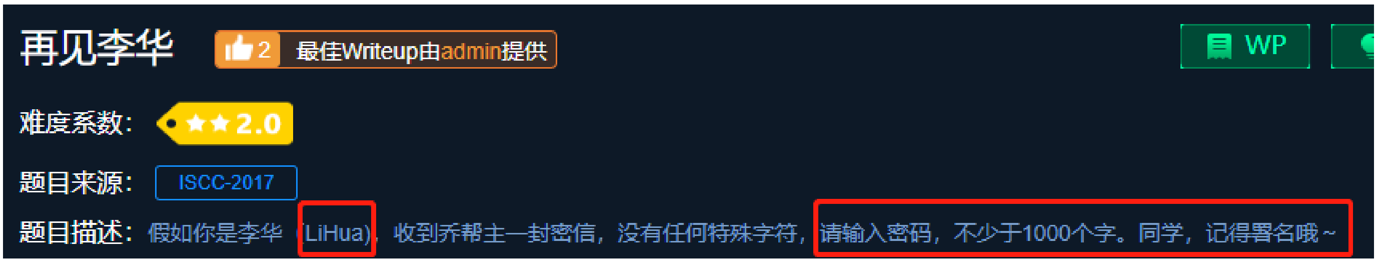


```
root@kali:~/桌面# binwalk mail2LiHua.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
30            0x1E            TIFF image data, little-endian offset of first image directory: 8
19273         0x4B49          Zip archive data, encrypted at least v2.0 to extract, compressed size: 37, uncompressed size: 26, name: key.txt
19436         0x4BEC          End of Zip archive, footer length: 22

root@kali:~/桌面# foremost mail2LiHua.jpg
Processing: mail2LiHua.jpg
|foundat=key.txt�m׳D
{���0ɞ|¯��CGj�h
        �v������E�PK?
*|
```

进行解压缩需要密码，



根据题目描述，不少于1000个字，记得署名，猜测1000个字是四位，署名LiHua，密码组成为四位字符+LiHua



再见李华

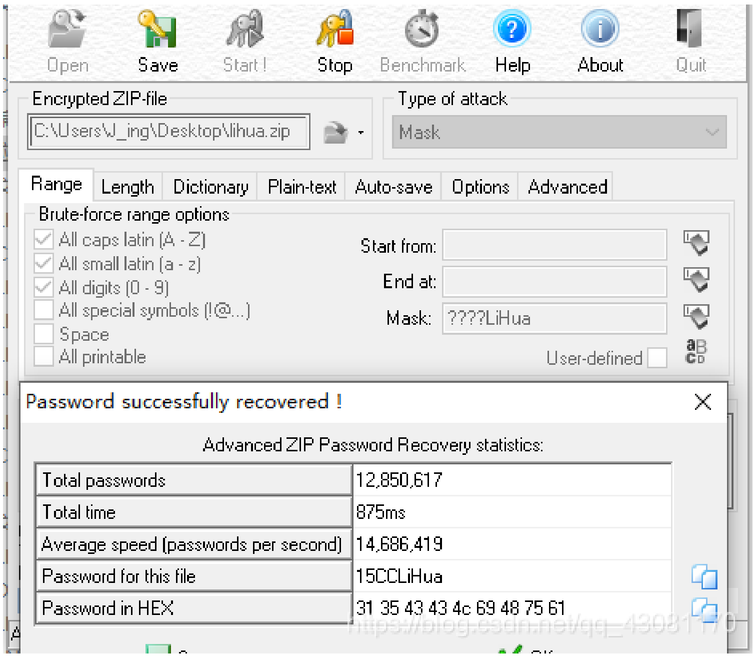
2 最佳Writeup由admin提供

难度系数：★★2.0

题目来源：ISCC-2017

题目描述：假如你是李华（LiHua），收到乔帮主一封密信，没有任何特殊字符，请输入密码，不少于1000个字。同学，记得署名哦～

利用AZPR进行爆破，得到密码。



Encrypted ZIP-file: C:\Users\J_ing\Desktop\lihua.zip

Type of attack: Mask

Brute-force range options

- ☑ All caps latin (A - Z)
- ☑ All small latin (a - z)
- ☑ All digits (0 - 9)
- □ All special symbols (!@...)
- □ Space
- □ All printable

Start from:

End at:

Mask: ????LiHua

User-defined □

Password successfully recovered !

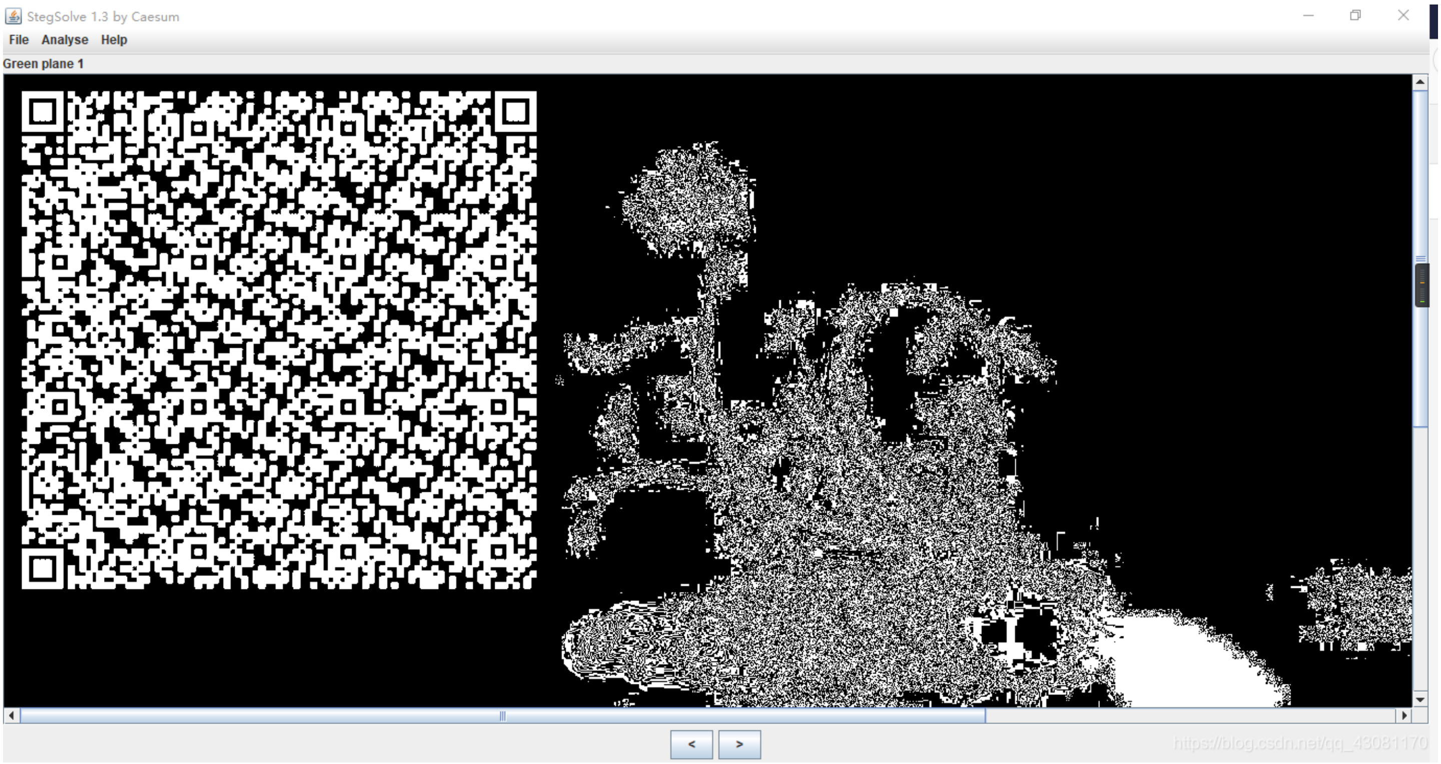
Advanced ZIP Password Recovery statistics:

Total passwords	12,850,617
Total time	875ms
Average speed (passwords per second)	14,686,419
Password for this file	15CCLiHua
Password in HEX	31 35 43 43 4c 69 48 75 61

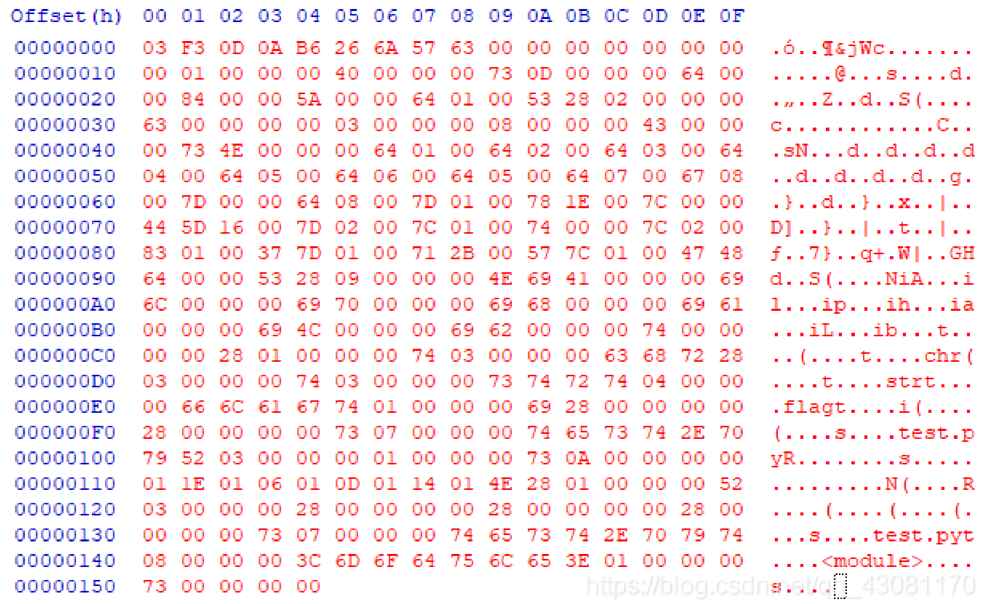
解压缩得到flag。

stage1

下载附件，将png图片用stegsolve打开，Green plane1得到一个二维码，



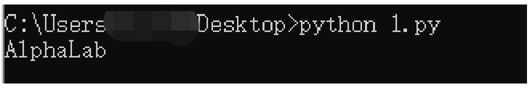
扫描二维码时十六进制字符，复制到winhex中，



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  03 F3 0D 0A B6 26 6A 57 63 00 00 00 00 00 00 00  .ó..¶&jWc.......
00000010  00 01 00 00 00 40 00 00 00 73 0D 00 00 00 64 00  .....@...s....d.
00000020  00 84 00 00 5A 00 00 64 01 00 53 28 02 00 00 00  .„..Z..d..S(....
00000030  63 00 00 00 00 03 00 00 00 08 00 00 00 43 00 00  c............C..
00000040  00 73 4E 00 00 00 64 01 00 64 02 00 64 03 00 64  .sN...d..d..d..d
00000050  04 00 64 05 00 64 06 00 64 05 00 64 07 00 67 08  ..d..d..d..d..g.
00000060  00 7D 00 00 64 08 00 7D 01 00 78 1E 00 7C 00 00  .}..d..}..x..|..
00000070  44 5D 16 00 7D 02 00 7C 01 00 74 00 00 7C 02 00  D]..}..|..t..|..
00000080  83 01 00 37 7D 01 00 71 2B 00 57 7C 01 00 47 48  ƒ..7}..q+.W|..GH
00000090  64 00 00 53 28 09 00 00 00 4E 69 41 00 00 00 69  d..S(....NiA...i
000000A0  6C 00 00 00 69 70 00 00 00 69 68 00 00 00 69 61  l...ip...ih...ia
000000B0  00 00 00 69 4C 00 00 00 69 62 00 00 00 74 00 00  ...iL...ib...t..
000000C0  00 00 28 01 00 00 00 74 03 00 00 00 63 68 72 28  ..(....t....chr(
000000D0  03 00 00 00 74 03 00 00 00 73 74 72 74 04 00 00  ....t....strt...
000000E0  00 66 6C 61 67 74 01 00 00 00 69 28 00 00 00 00  .flagt....i(....
000000F0  28 00 00 00 00 73 07 00 00 00 74 65 73 74 2E 70  (....s....test.p
00000100  79 52 03 00 00 00 01 00 00 00 73 0A 00 00 00 00  yR........s.....
00000110  01 1E 01 06 01 0D 01 14 01 4E 28 01 00 00 00 52  .........N(....R
00000120  03 00 00 00 28 00 00 00 00 28 00 00 00 00 28 00  ....(....(....(.
00000130  00 00 00 73 07 00 00 00 74 65 73 74 2E 70 79 74  ...s....test.pyt
00000140  08 00 00 00 3C 6D 6F 64 75 6C 65 3E 01 00 00 00  ....<module>....
00000150  73 00 00 00 00                                   s....
```

保存为`.pyc`格式，利用Easy Python Decompiler进行反编译，用文本工具打开得到源码，整理后运行得到flag。

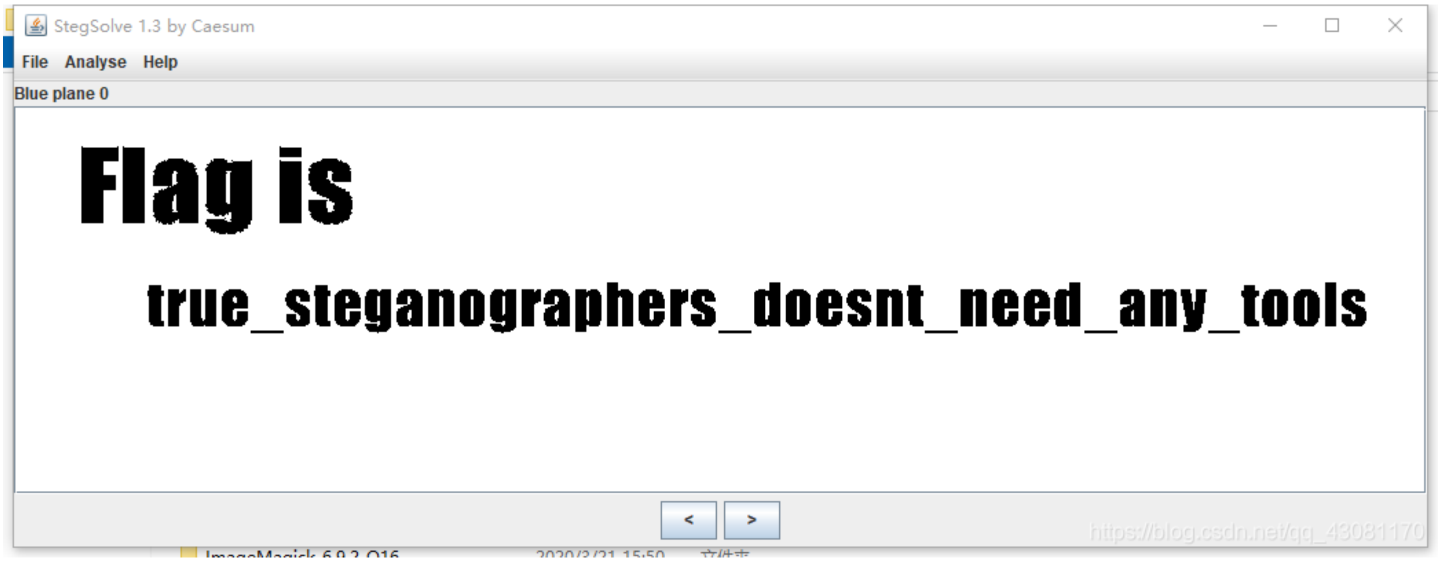
```
# Embedded file name: test.py
str = [65,108, 112,104,97,76,97,98]
flag = ''
for i in str:
    flag += chr(i)
print flag
```



```
C:\Users[redacted]Desktop>python 1.py
AlphaLab
```

pure_color

下载附件，用stegsolve打开，Blue plane 0得到flag（注意格式flag{xxxxxx}）。



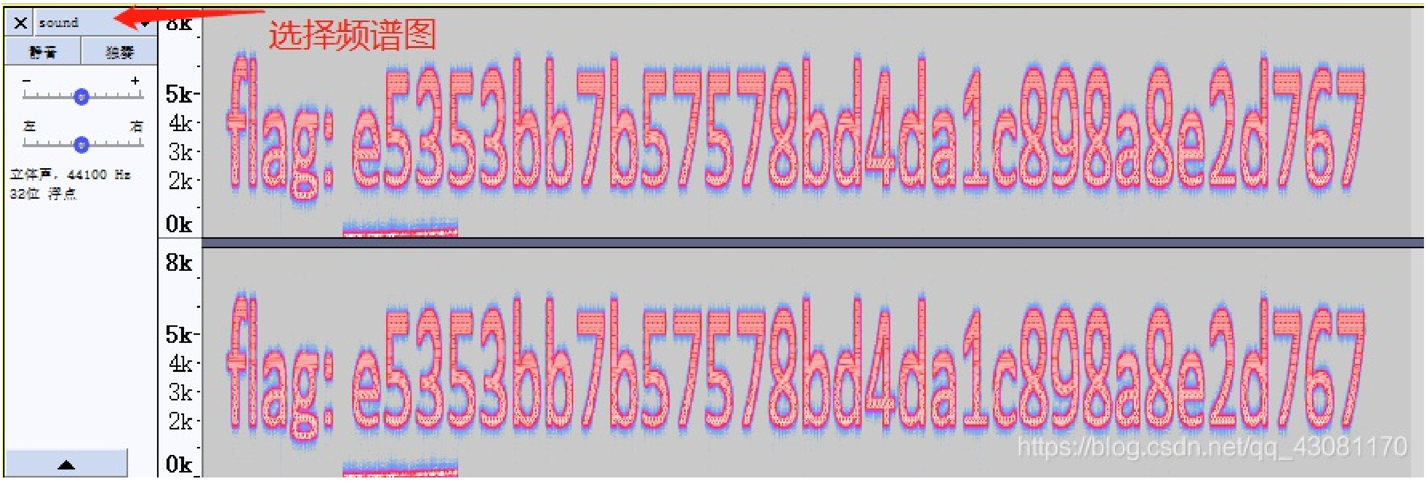
Blue plane 0

Flag is

true_steganographers_doesnt_need_any_tools

Hear-with-your-Eyes

下载附件解压缩，是一个音频，用Audacity打开，选择频谱图得到flag。

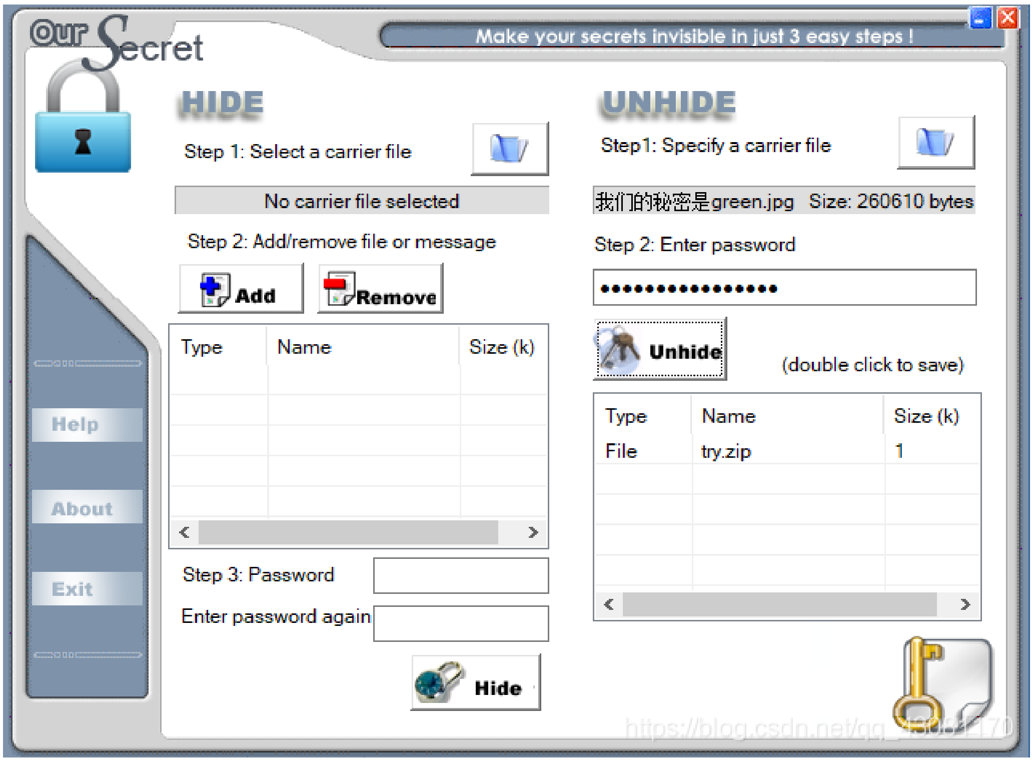


flag: e5353bb7b57578bd4da1c898a8e2d767

flag: e5353bb7b57578bd4da1c898a8e2d767

我们的秘密时绿色的

下载附件解压缩，是一张jpg图片，根据题目我们的秘密是绿色，用`OurSecret`工具分离处隐藏的文件，密码是jpg中的绿色的数字`0405111218192526`



HIDE

Step 1: Select a carrier file

No carrier file selected

Step 2: Add/remove file or message

Type	Name	Size (k)

Step 3: Password

Enter password again

UNHIDE

Step1: Specify a carrier file

我们的秘密是green.jpg Size: 260610 bytes

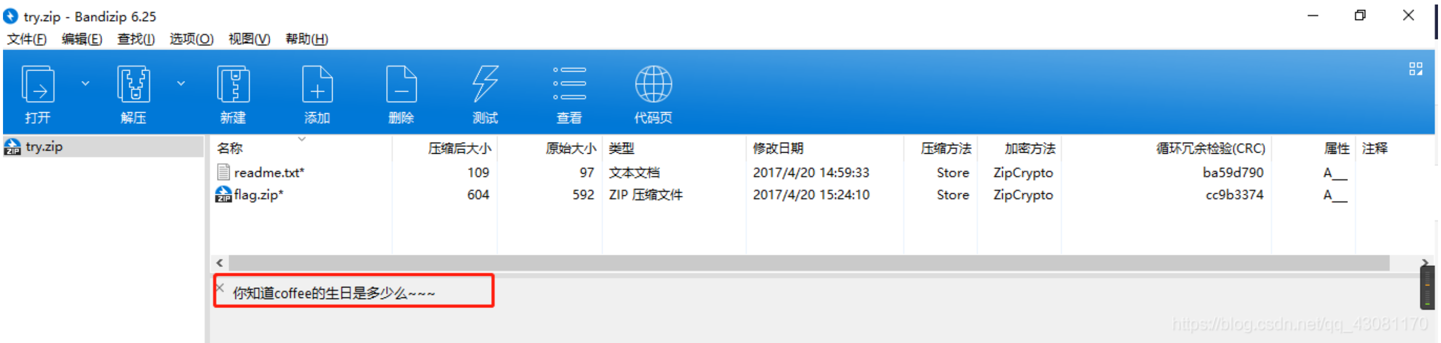
Step 2: Enter password

••••••••••••••••

(double click to save)

Type	Name	Size (k)
File	try.zip	1

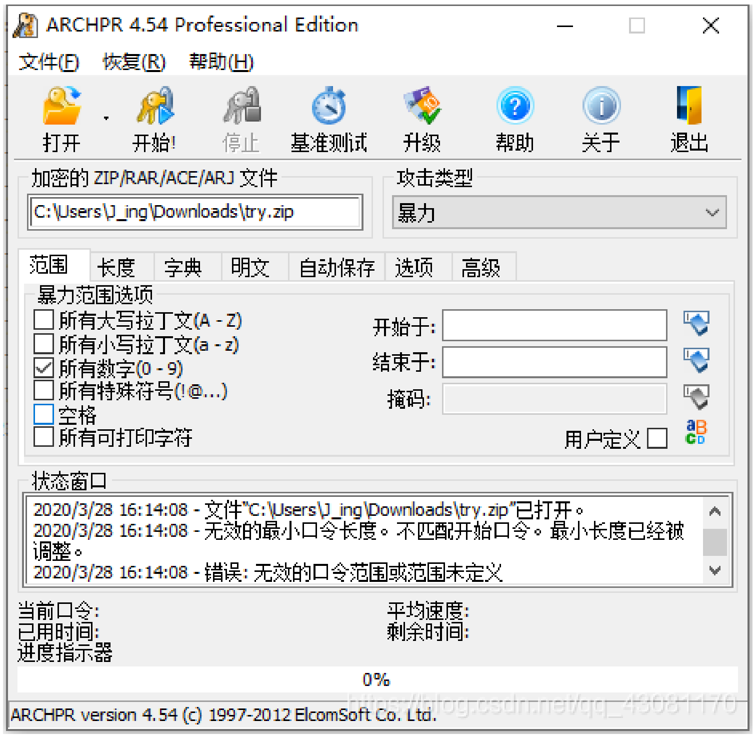
分离出来一个try.zip的压缩包，双击打开，



名称	压缩后大小	原始大小	类型	修改日期	压缩方法	加密方法	循环冗余检验(CRC)	属性	注释
readme.txt*	109	97	文本文档	2017/4/20 14:59:33	Store	ZipCrypto	ba59d790	A__	
flag.zip*	604	592	ZIP 压缩文件	2017/4/20 15:24:10	Store	ZipCrypto	cc9b3374	A__	

你知道coffee的生日是多少么~~~

提示`你知道coffee的生日是多少么~~~`，利用ARCHPR进行爆破，范围选择所有数字(0~9)，长度选择8位，



加密的 ZIP/RAR/ACE/ARJ 文件: C:\Users\J_ing\Downloads\try.zip

攻击类型: 暴力

暴力范围选项

□ 所有大写拉丁文(A - Z)
□ 所有小写拉丁文(a - z)
☑ 所有数字(0 - 9)
□ 所有特殊符号(!@...)
□ 空格
□ 所有可打印字符

开始于:

结束于:

掩码:

用户定义 □

状态窗口

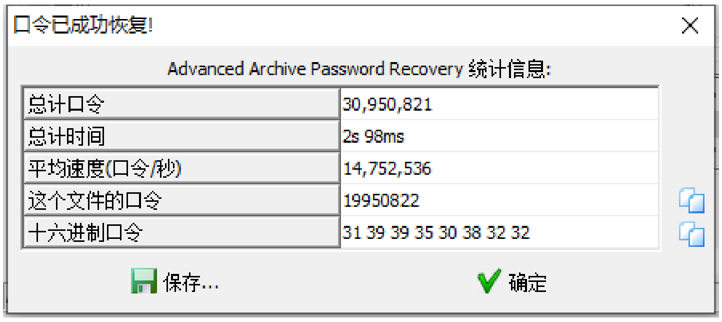
```
2020/3/28 16:14:08 - 文件"C:\Users\J_ing\Downloads\try.zip"已打开。
2020/3/28 16:14:08 - 无效的最小口令长度。不匹配开始口令。最小长度已经被调整。
2020/3/28 16:14:08 - 错误: 无效的口令范围或范围未定义
```

当前口令:
已用时间:
进度指示器
平均速度:
剩余时间:

0%

ARCHPR version 4.54 (c) 1997-2012 ElcomSoft Co. Ltd.

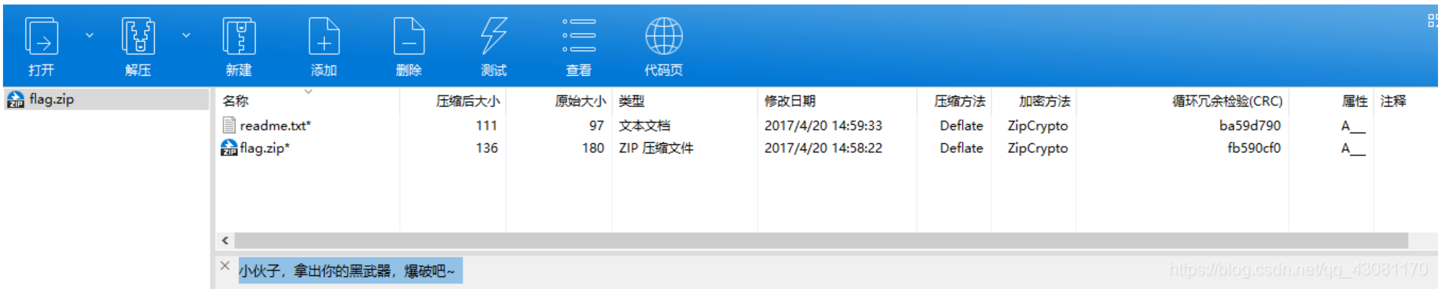
得到密码为`19950822`，


口令已成功恢复!

Advanced Archive Password Recovery 统计信息:

总计口令	30,950,821
总计时间	2s 98ms
平均速度(口令/秒)	14,752,536
这个文件的口令	19950822
十六进制口令	31 39 39 35 30 38 32 32

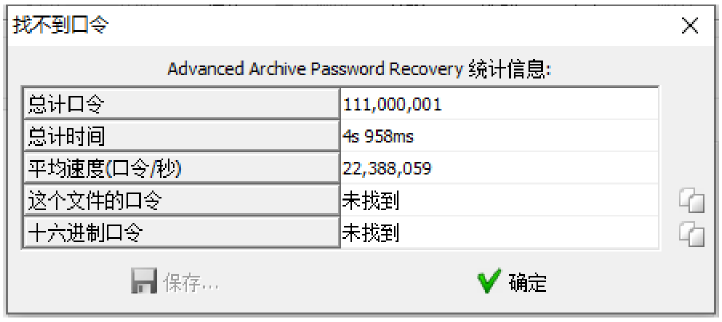
解压缩得到flag.zip和readme.txt，双击打开flag.zip，提示`小伙子，拿出你的黑武器，爆破吧~`，



名称	压缩后大小	原始大小	类型	修改日期	压缩方法	加密方法	循环冗余检验(CRC)	属性	注释
readme.txt*	111	97	文本文档	2017/4/20 14:59:33	Deflate	ZipCrypto	ba59d790	A__	
flag.zip*	136	180	ZIP 压缩文件	2017/4/20 14:58:22	Deflate	ZipCrypto	fb590cf0	A__	

小伙子，拿出你的黑武器，爆破吧~

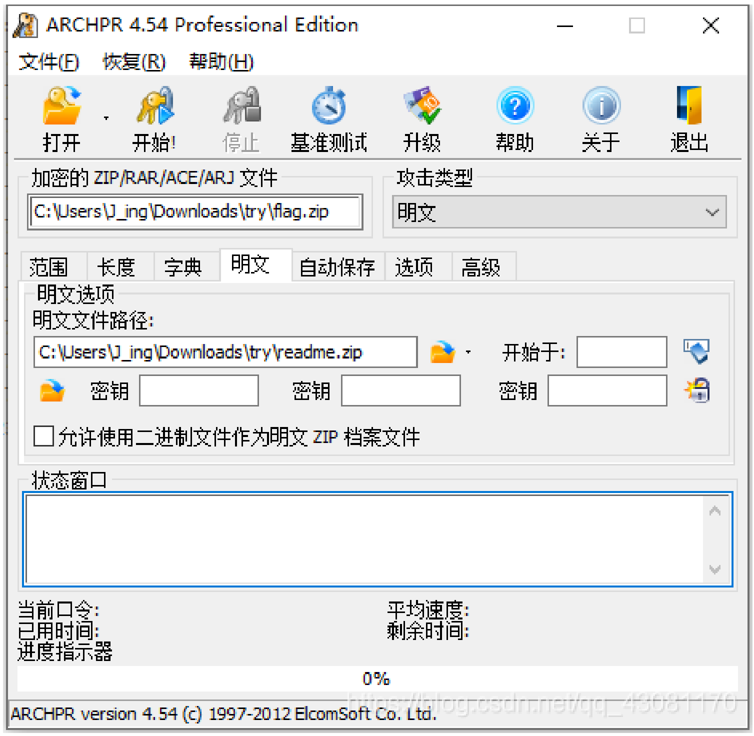
进行爆破，


找不到口令

Advanced Archive Password Recovery 统计信息:

总计口令	111,000,001
总计时间	4s 958ms
平均速度(口令/秒)	22,388,059
这个文件的口令	未找到
十六进制口令	未找到

将readme.txt压缩，查看到和flag.txt的循环冗余校验(CRC)值相同，应该为明文攻击，利用ARCHPR进行明文攻击，



加密的 ZIP/RAR/ACE/ARJ 文件: C:\Users\J_ing\Downloads\try\flag.zip

攻击类型: 明文

明文文件路径: C:\Users\J_ing\Downloads\try\readme.zip

□允许使用二进制文件作为明文 ZIP 档案文件

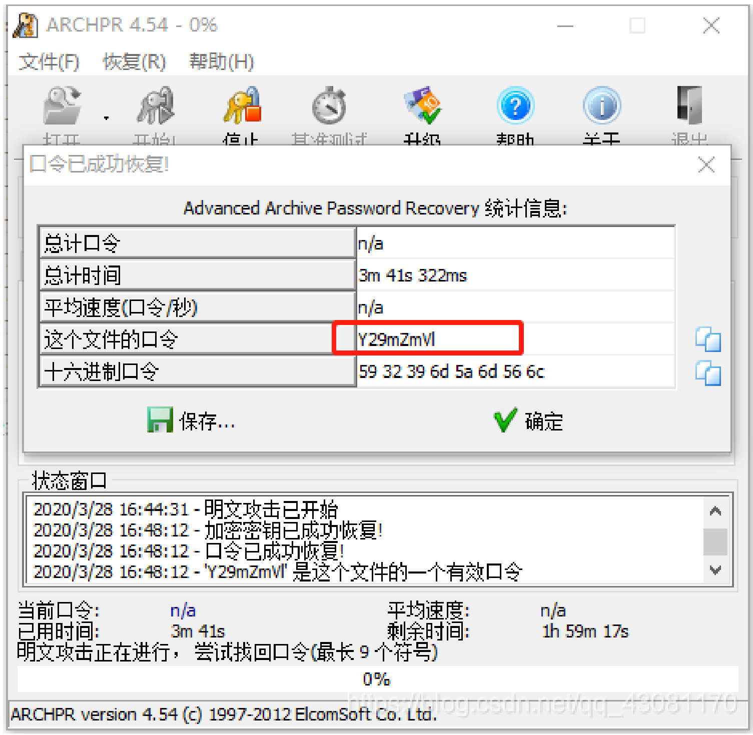
当前口令:
已用时间:
进度指示器

平均速度:
剩余时间:

0%

ARCHPR version 4.54 (c) 1997-2012 ElcomSoft Co. Ltd.

得到解压密码，



Advanced Archive Password Recovery 统计信息:

总计口令	n/a
总计时间	3m 41s 322ms
平均速度(口令/秒)	n/a
这个文件的口令	Y29mZmVl
十六进制口令	59 32 39 6d 5a 6d 56 6c

```
2020/3/28 16:44:31 - 明文攻击已开始
2020/3/28 16:48:12 - 加密密钥已成功恢复!
2020/3/28 16:48:12 - 口令已成功恢复!
2020/3/28 16:48:12 - 'Y29mZmVl' 是这个文件的一个有效口令
```

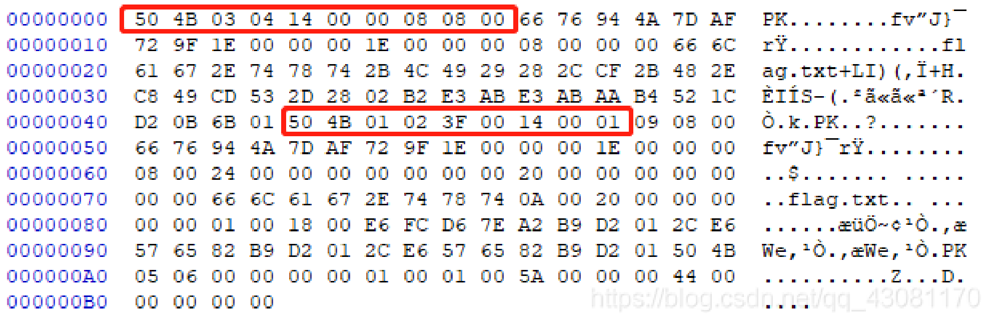
当前口令: n/a
已用时间: 3m 41s
明文攻击正在进行，尝试找回口令(最长 9 个符号)

平均速度: n/a
剩余时间: 1h 59m 17s

0%

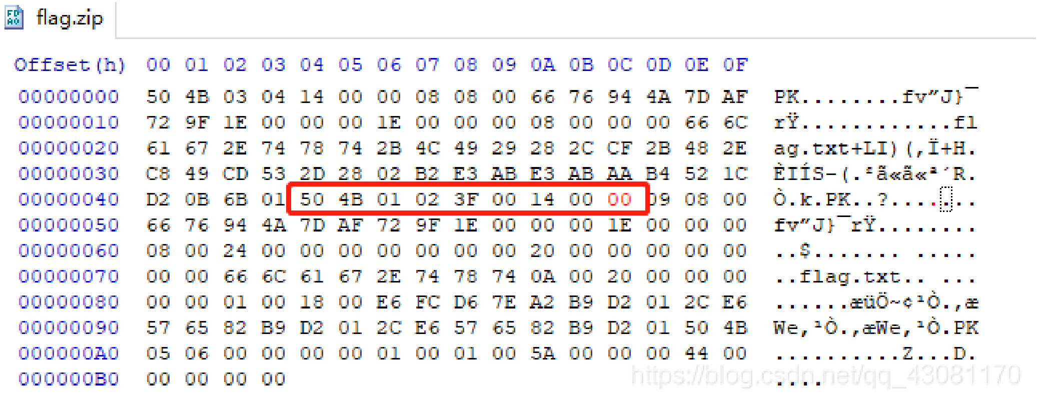
ARCHPR version 4.54 (c) 1997-2012 ElcomSoft Co. Ltd.

解压缩又得到一个flag.zip和readme.txt，解压缩仍需要密码，双击打开没有提示，放进winhex中，



```
00000000  50 4B 03 04 14 00 00 08 08 00 66 76 94 4A 7D AF   PK........fv”J}¯
00000010  72 9F 1E 00 00 00 1E 00 00 00 08 00 00 00 66 6C   rŸ............fl
00000020  61 67 2E 74 78 74 2B 4C 49 29 28 2C CF 2B 48 2E   ag.txt+LI)(,Ï+H.
00000030  C8 49 CD 53 2D 28 02 B2 E3 AB E3 AB AA B4 52 1C   ÈIÍS-(.²ã«ã«ª´R.
00000040  D2 0B 6B 01 50 4B 01 02 3F 00 14 00 01 09 08 00   Ò.k.PK..?.......
00000050  66 76 94 4A 7D AF 72 9F 1E 00 00 00 1E 00 00 00   fv”J}¯rŸ........
00000060  08 00 24 00 00 00 00 00 00 00 20 00 00 00 00 00   ..$....... .....
00000070  00 00 66 6C 61 67 2E 74 78 74 0A 00 20 00 00 00   ..flag.txt.. ...
00000080  00 00 01 00 18 00 E6 FC D6 7E A2 B9 D2 01 2C E6   ......æüÖ~¢¹Ò.,æ
00000090  57 65 82 B9 D2 01 2C E6 57 65 82 B9 D2 01 50 4B   We‚¹Ò.,æWe‚¹Ò.PK
000000A0  05 06 00 00 00 00 01 00 01 00 5A 00 00 00 44 00   ..........Z...D.
000000B0  00 00 00 00                                       ....
```

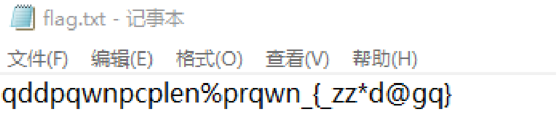
为zip伪加密，将`00 01`改为`00 00`保存后解压缩，



flag.zip

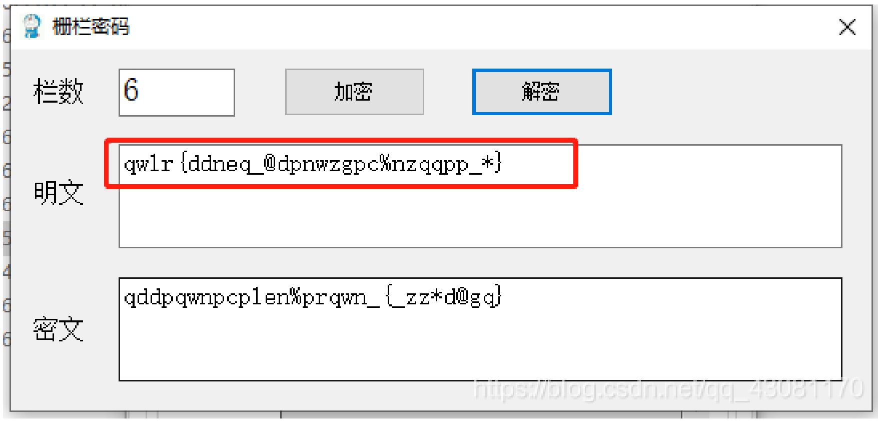
```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  50 4B 03 04 14 00 00 08 08 00 66 76 94 4A 7D AF  PK........fv”J}¯
00000010  72 9F 1E 00 00 00 1E 00 00 00 08 00 00 00 66 6C  rŸ............fl
00000020  61 67 2E 74 78 74 2B 4C 49 29 28 2C CF 2B 48 2E  ag.txt+LI)(,Ï+H.
00000030  C8 49 CD 53 2D 28 02 B2 E3 AB E3 AB AA B4 52 1C  ÈIÍS-(.²ã«ã«ª´R.
00000040  D2 0B 6B 01 50 4B 01 02 3F 00 14 00 00 09 08 00  Ò.k.PK..?.......
00000050  66 76 94 4A 7D AF 72 9F 1E 00 00 00 1E 00 00 00  fv”J}¯rŸ........
00000060  08 00 24 00 00 00 00 00 00 00 20 00 00 00 00 00  ..$....... .....
00000070  00 00 66 6C 61 67 2E 74 78 74 0A 00 20 00 00 00  ..flag.txt.. ...
00000080  00 00 01 00 18 00 E6 FC D6 7E A2 B9 D2 01 2C E6  ......æüÖ~¢¹Ò.,æ
00000090  57 65 82 B9 D2 01 2C E6 57 65 82 B9 D2 01 50 4B  We‚¹Ò.,æWe‚¹Ò.PK
000000A0  05 06 00 00 00 00 01 00 01 00 5A 00 00 00 44 00  ..........Z...D.
000000B0  00 00 00 00                                      ....
```

解压得到flag.txt，



qddpqwnpcplen%prqwn_{_zz*d@gq}

利用栅栏解密，

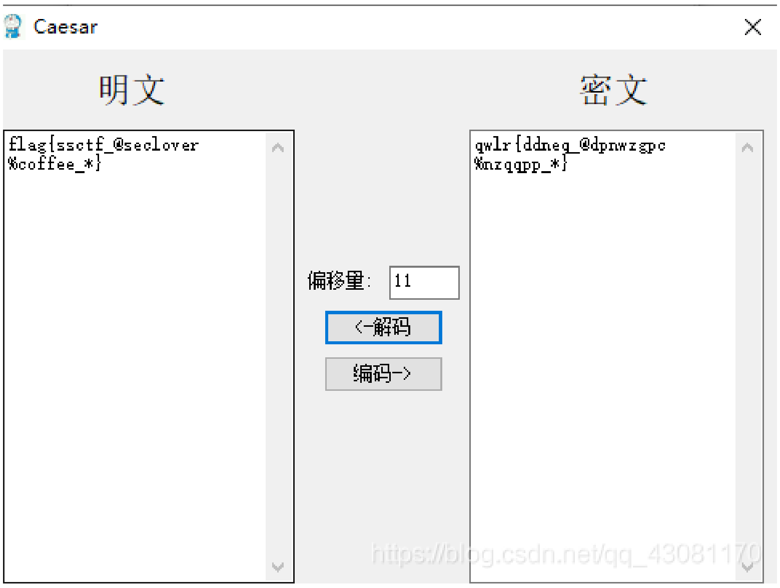


栏数 6

明文 qwlr{ddneq_@dpnwzgpc%nzqqpp_*}

密文 qddpqwnpcplen%prqwn_{_zz*d@gq}

再进行凯撒解密得到flag。



明文

flag{ssctf_@seclover
%coffee_*}

密文

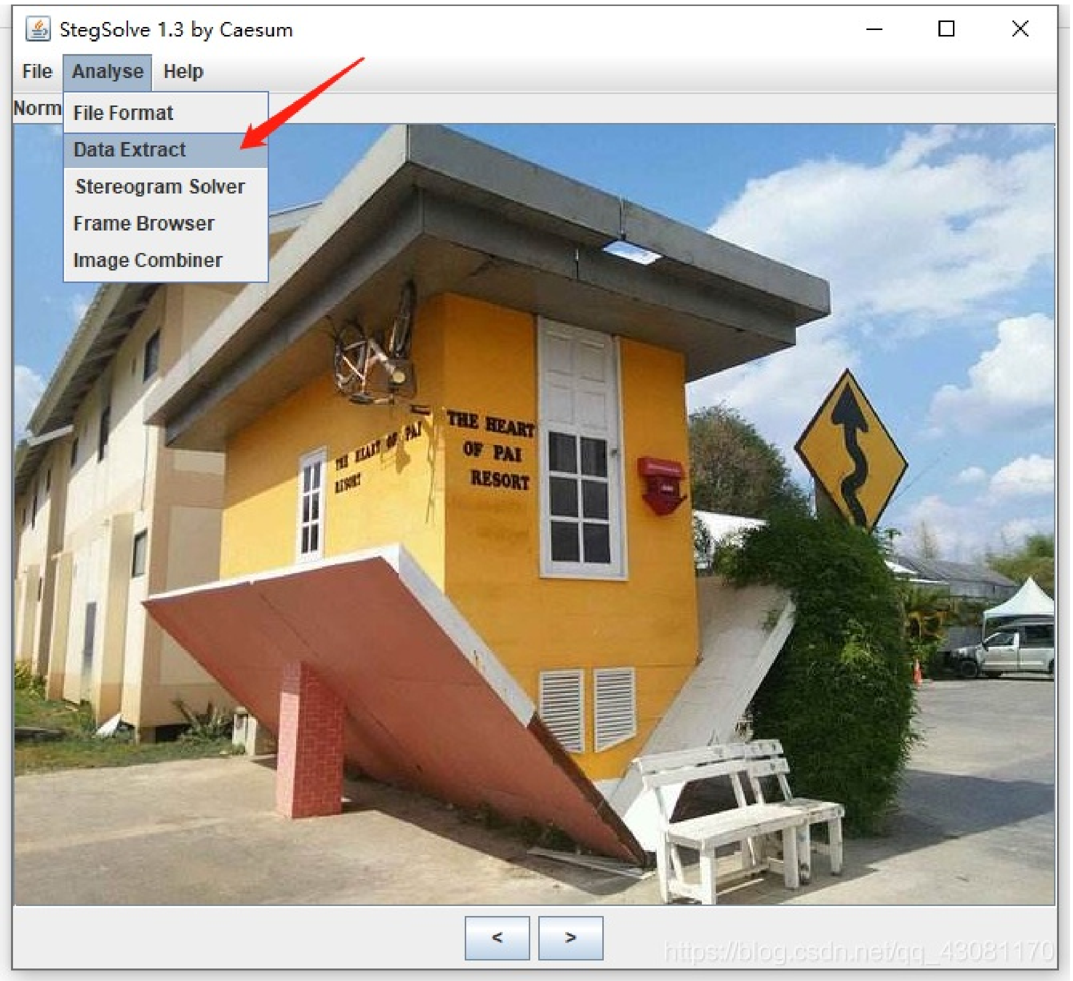
qwlr{ddneq_@dpnwzgpc
%nzqqpp_*}

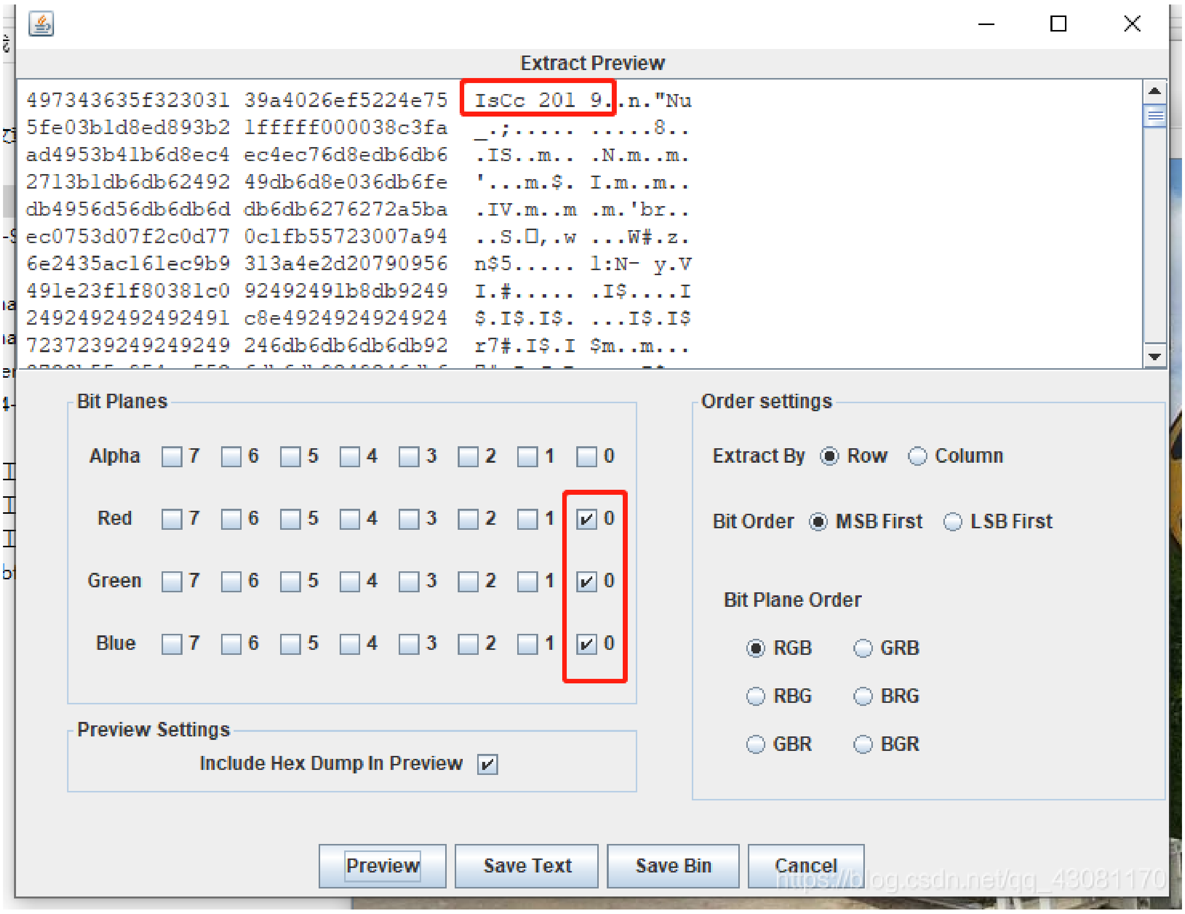
偏移量：11

倒立屋

下载附件解压缩出来一张png图片，

放进StegSolve，





Extract Preview

```
497343635f323031 39a4026ef5224e75  IsCc 201 9..n."Nu
5fe03b1d8ed893b2 1fffff000038c3fa  _.;..... .....8..
ad4953b41b6d8ec4 ec4ec76d8edb6db6  .IS..m.. .N.m..m.
2713b1db6db62492 49db6d8e036db6fe  '...m.$. I.m..m..
db4956d56db6db6d db6db6276272a5ba  .IV.m..m .m.'br..
ec0753d07f2c0d77 0c1fb55723007a94  ..S.□,.w ...W#.z.
6e2435ac161ec9b9 313a4e2d20790956  n$5..... 1:N- y.V
491e23f1f80381c0 92492491b8db9249  I.#..... .I$....I
2492492492492491 c8e4924924924924  $.I$.I$. ...I$.I$
7237239249249249 246db6db6db6db92  r7#.I$.I $m..m...
```

Bit Planes

Alpha	□ 7	□ 6	□ 5	□ 4	□ 3	□ 2	□ 1	□ 0
Red	□ 7	□ 6	□ 5	□ 4	□ 3	□ 2	□ 1	☑ 0
Green	□ 7	□ 6	□ 5	□ 4	□ 3	□ 2	□ 1	☑ 0
Blue	□ 7	□ 6	□ 5	□ 4	□ 3	□ 2	□ 1	☑ 0

Preview Settings

Include Hex Dump In Preview ☑

Order settings

Extract By ● Row ○ Column

Bit Order ● MSB First ○ LSB First

Bit Plane Order

● RGB ○ GRB

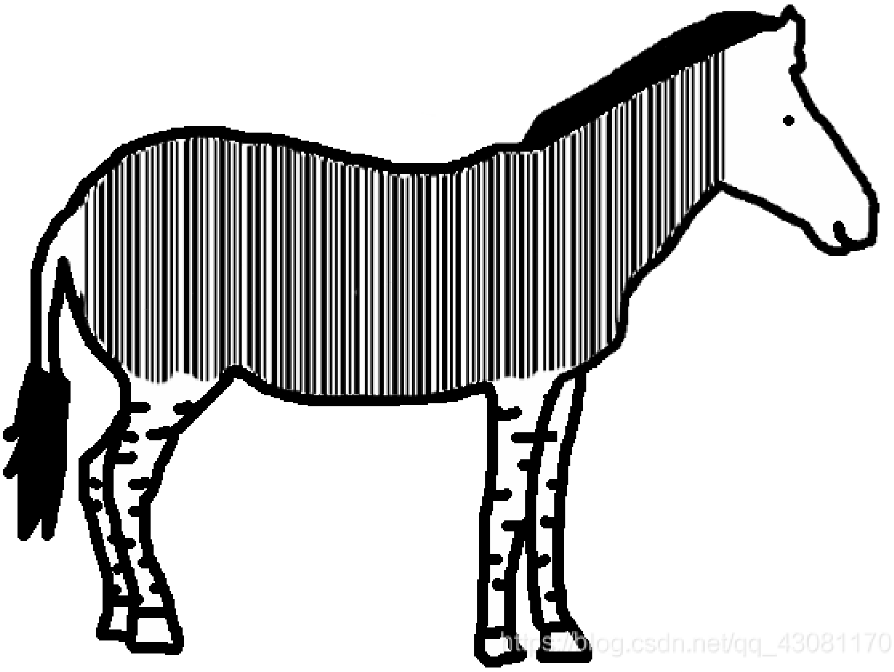
○ RBG ○ BRG

○ GBR ○ BGR

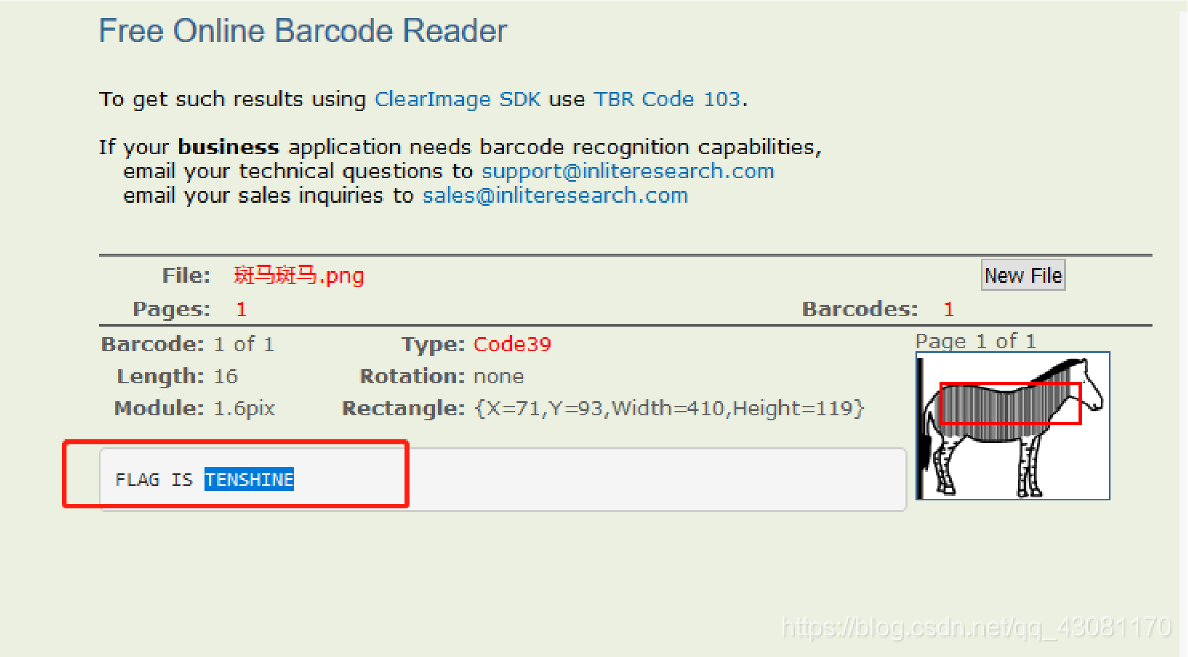
将IsCc_2019倒过来就是flag：`flag{9102_cCsI}`。

Banmabanma

下载附件，是一个斑马的图片，



中间看着像条形码，利用在线条码扫描工具`https://online-barcode-reader.inliteresearch.com/`扫描得到flag。



Free Online Barcode Reader

To get such results using ClearImage SDK use TBR Code 103.

If your **business** application needs barcode recognition capabilities,
email your technical questions to support@inliteresearch.com
email your sales inquiries to sales@inliteresearch.com

File: 斑马斑马.png

Pages: 1

Barcodes: 1

Barcode: 1 of 1

Type: Code39

Length: 16

Rotation: none

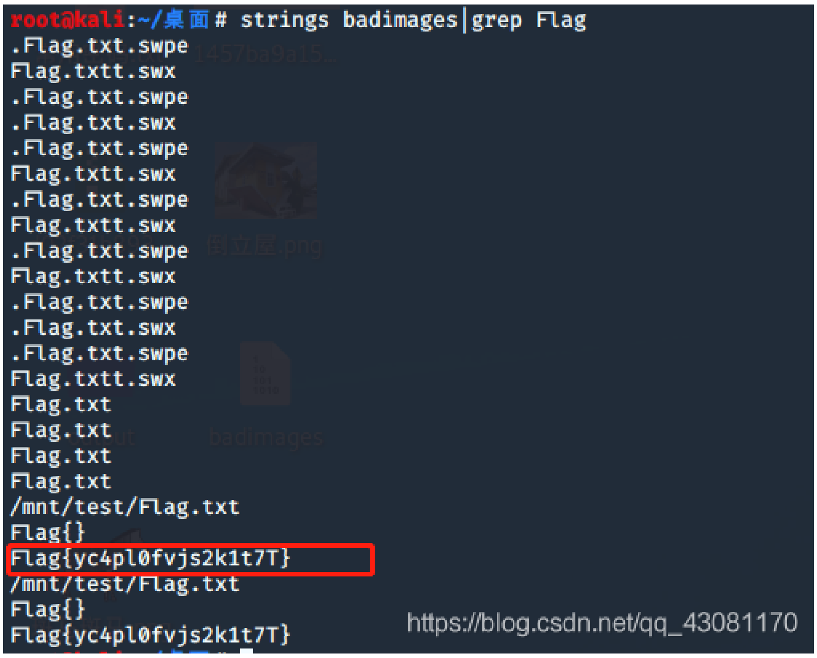
Module: 1.6pix

Rectangle: {X=71,Y=93,Width=410,Height=119}

```
FLAG IS TENSHINE
```

something_in_image

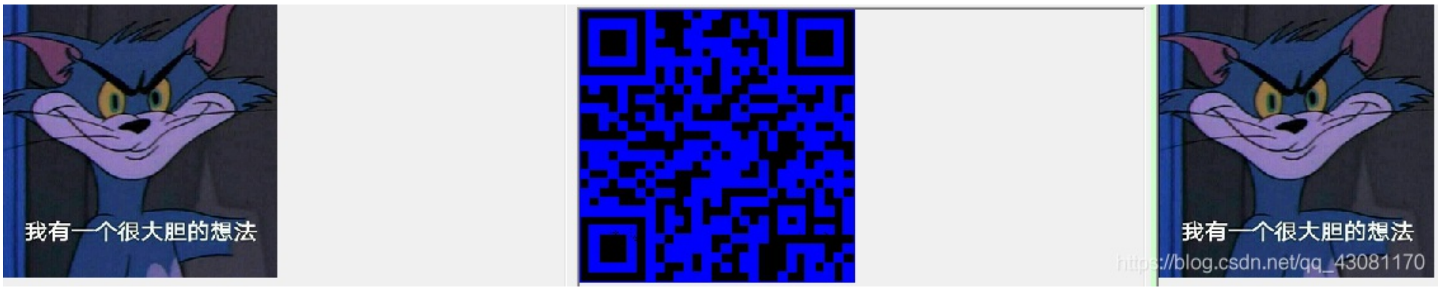
下载附件，利用flie命令分析文件为 Linux EXT filesystem，直接 `strings badimages |grep Flag` 得到flag。



```
root@kali:~/桌面# strings badimages|grep Flag
.Flag.txt.swpe
Flag.txtt.swx
.Flag.txt.swpe
.Flag.txt.swx
.Flag.txt.swpe
Flag.txtt.swx
.Flag.txt.swpe
Flag.txtt.swx
.Flag.txt.swpe
Flag.txtt.swx
.Flag.txt.swpe
.Flag.txt.swx
.Flag.txt.swpe
Flag.txtt.swx
Flag.txt
Flag.txt
Flag.txt
Flag.txt
/mnt/test/Flag.txt
Flag{}
Flag{yc4pl0fvjs2k1t7T}
/mnt/test/Flag.txt
Flag{}
Flag{yc4pl0fvjs2k1t7T}
```

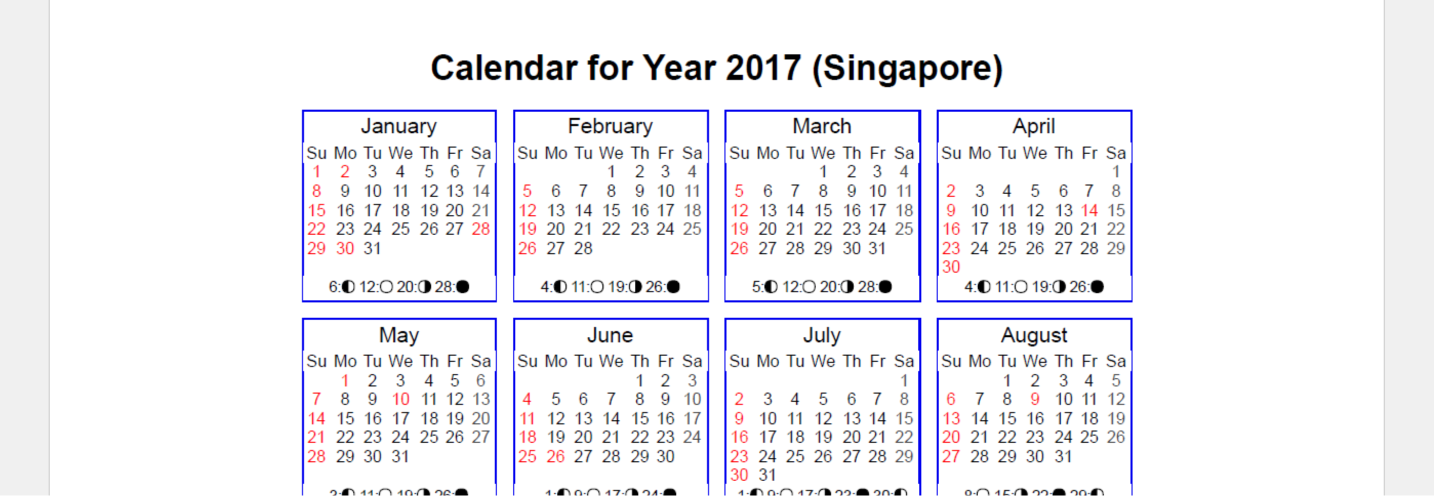
a_good_idea

下载附件解压缩得到一张图片，利用binwalk分析文件，发现隐藏有压缩包，foremost解压缩得到一个txt文件和两个看似一样的图片，查看txt文件，内容为 `try to find the secret of pixels`，尝试找到像素的秘密，两张看似一样的图片利用Beyond Compare进行对比得到一个二维码，扫码得到flag。



2017_Dating_in_Singapore

下载附件解压缩是一张pdf图片，打开是一个日历，



Calendar for Year 2017 (Singapore)

January

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

6:◐ 12:○ 20:◑ 28:●

February

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

4:◐ 11:○ 19:◑ 26:●

March

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

5:◐ 12:○ 20:◑ 28:●

April

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

4:◐ 11:○ 19:◑ 26:●

May

Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

3:◐ 11:○ 19:◑ 26:●

June

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

1:◐ 9:○ 17:◑ 24:●

July

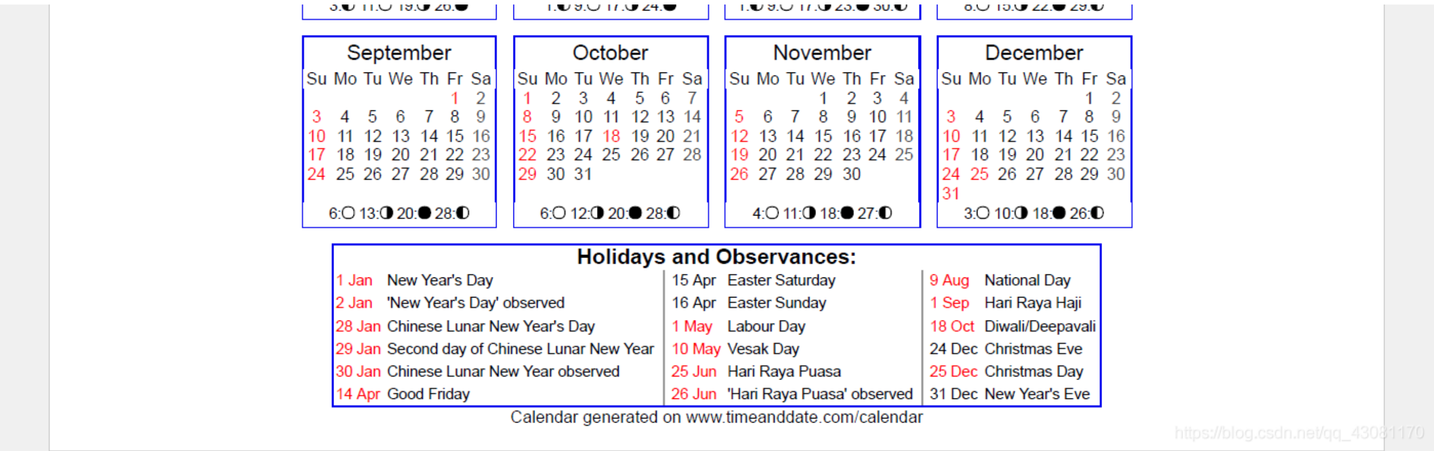
Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

1:◐ 9:○ 17:◑ 23:● 30:◐

August

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

8:○ 15:◑ 22:● 29:◐



3:◐ 11:○ 19:◑ 26:● 1:◐ 9:○ 17:◑ 24:● 1:◐ 9:○ 17:◑ 23:● 30:◐ 8:○ 15:◑ 22:● 29:◐

September

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

6:○ 13:◑ 20:● 28:◐

October

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

6:○ 12:◑ 20:● 28:◐

November

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

4:○ 11:◑ 18:● 27:◐

December

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

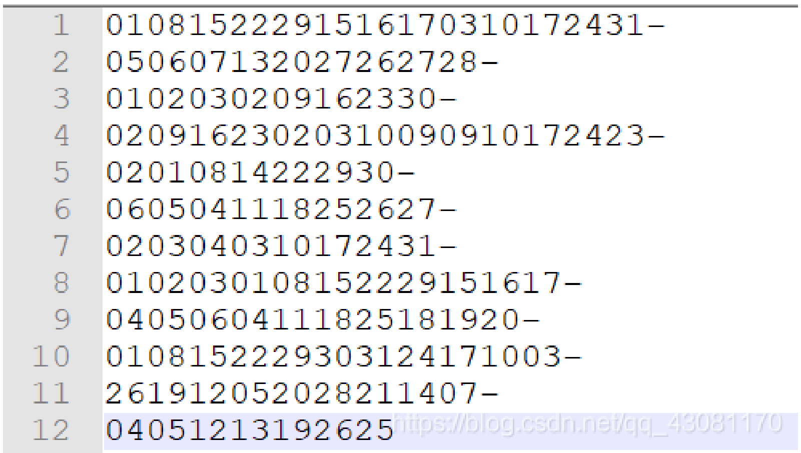
3:○ 10:◑ 18:● 26:◐

Holidays and Observances:

1 Jan	New Year's Day	15 Apr	Easter Saturday	9 Aug	National Day
2 Jan	'New Year's Day' observed	16 Apr	Easter Sunday	1 Sep	Hari Raya Haji
28 Jan	Chinese Lunar New Year's Day	1 May	Labour Day	18 Oct	Diwali/Deepavali
29 Jan	Second day of Chinese Lunar New Year	10 May	Vesak Day	24 Dec	Christmas Eve
30 Jan	Chinese Lunar New Year observed	25 Jun	Hari Raya Puasa	25 Dec	Christmas Day
14 Apr	Good Friday	26 Jun	'Hari Raya Puasa' observed	31 Dec	New Year's Eve

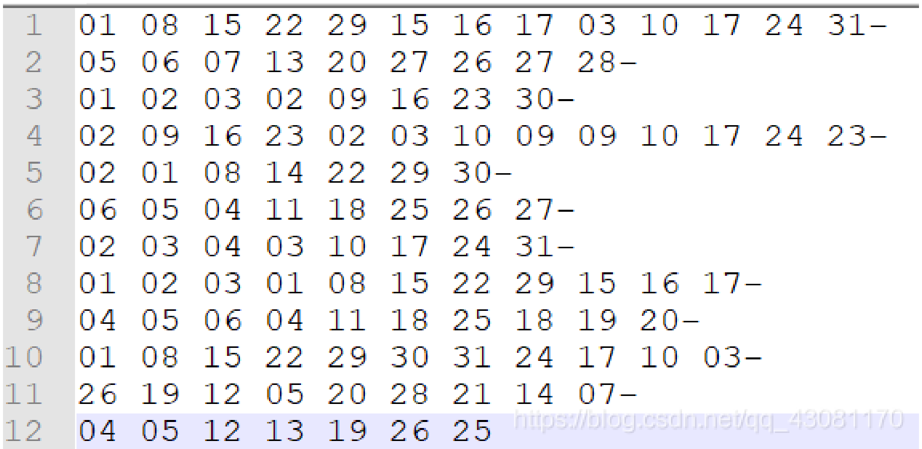
Calendar generated on www.timeanddate.com/calendar

题目描述中还有一串数字，数字间有-，将数字按照-分割。



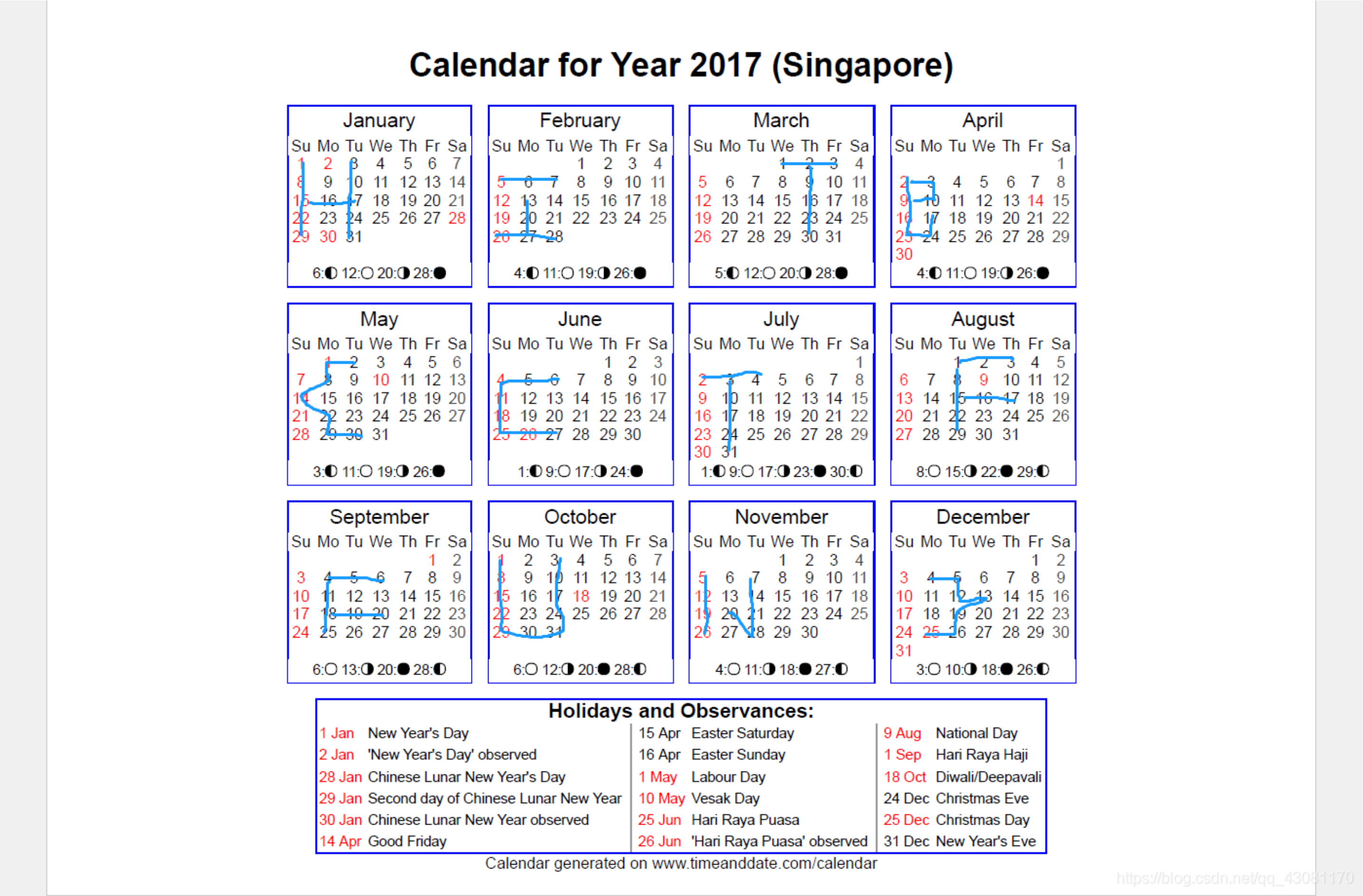
```
0108152229151617031017243 1-
0506071320272627 28-
0102030209162330-
0209162302031009091017242 3-
02010814222930-
06050411182526 27-
02030403101724 31-
0102030108152229151617-
04050604111825181920-
0108152229303124171003-
261912052028211407-
04051213192625
```

分割之后正好有12行对应12个月份，没两位一组，



```
01 08 15 22 29 15 16 17 03 10 17 24 31-
05 06 07 13 20 27 26 27 28-
01 02 03 02 09 16 23 30-
02 09 16 23 02 03 10 09 09 10 17 24 23-
02 01 08 14 22 29 30-
06 05 04 11 18 25 26 27-
02 03 04 03 10 17 24 31-
01 02 03 01 08 15 22 29 15 16 17-
04 05 06 04 11 18 25 18 19 20-
01 08 15 22 29 30 31 24 17 10 03-
26 19 12 05 20 28 21 14 07-
04 05 12 13 19 26 25
```

按照顺序连线得到flag。



Calendar for Year 2017 (Singapore)

January

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

6:◐ 12:○ 20:◑ 28:●

February

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

4:◐ 11:○ 19:◑ 26:●

March

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

5:◐ 12:○ 20:◑ 28:●

April

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

4:◐ 11:○ 19:◑ 26:●

May

Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

3:◐ 11:○ 19:◑ 26:●

June

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

1:◐ 9:○ 17:◑ 24:●

July

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

1:◐ 9:○ 17:◑ 23:● 30:◐

August

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

8:○ 15:◑ 22:● 29:◐

September

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

6:○ 13:◑ 20:● 28:◐

October

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

6:○ 12:◑ 20:● 28:◐

November

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

4:○ 11:◑ 18:● 27:◐

December

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

3:○ 10:◑ 18:● 26:◐

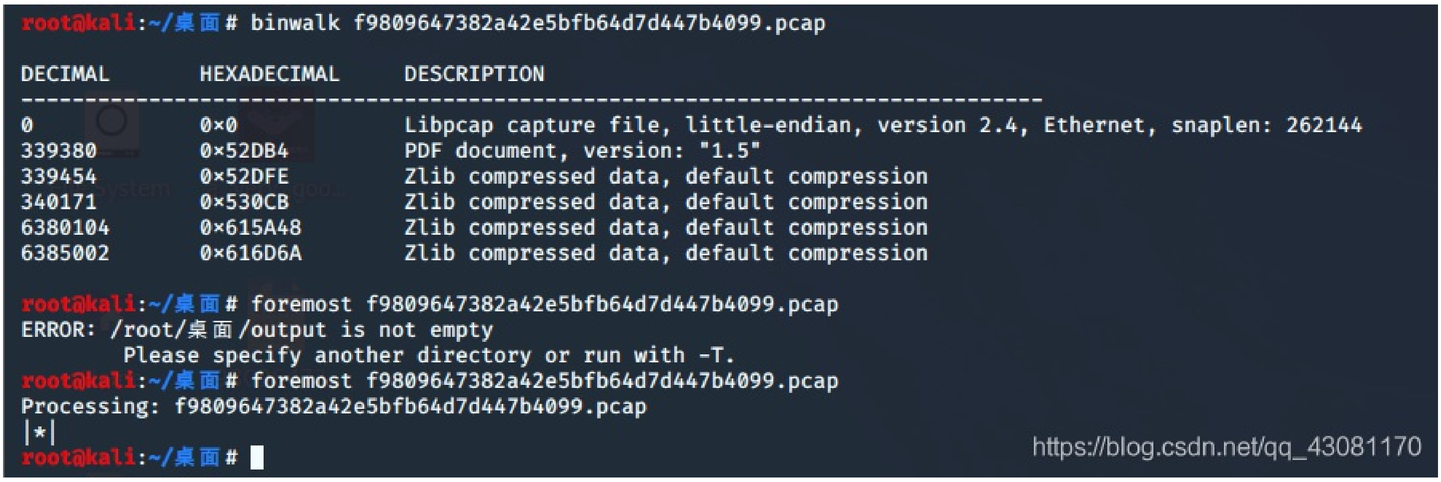
Holidays and Observances:

1 Jan	New Year's Day	15 Apr	Easter Saturday	9 Aug	National Day
2 Jan	'New Year's Day' observed	16 Apr	Easter Sunday	1 Sep	Hari Raya Haji
28 Jan	Chinese Lunar New Year's Day	1 May	Labour Day	18 Oct	Diwali/Deepavali
29 Jan	Second day of Chinese Lunar New Year	10 May	Vesak Day	24 Dec	Christmas Eve
30 Jan	Chinese Lunar New Year observed	25 Jun	Hari Raya Puasa	25 Dec	Christmas Day
14 Apr	Good Friday	26 Jun	'Hari Raya Puasa' observed	31 Dec	New Year's Eve

Calendar generated on www.timeanddate.com/calendar

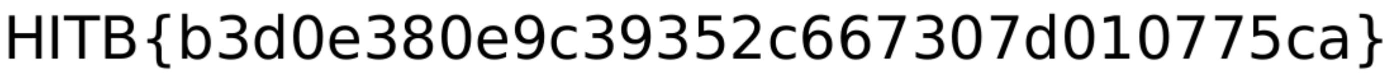
simple_transfer

下载附件，是一个pcap格式的文件，用binwalk分心文件，文件中隐藏有pdf文件 ，利用foremost进行文件分离，得到pdf文件。



```
root@kali:~/桌面# binwalk f9809647382a42e5bfb64d7d447b4099.pcap

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             Libpcap capture file, little-endian, version 2.4, Ethernet, snaplen: 262144
339380        0×52DB4         PDF document, version: "1.5"
339454        0×52DFE         Zlib compressed data, default compression
340171        0×530CB         Zlib compressed data, default compression
6380104       0×615A48        Zlib compressed data, default compression
6385002       0×616D6A        Zlib compressed data, default compression

root@kali:~/桌面# foremost f9809647382a42e5bfb64d7d447b4099.pcap
ERROR: /root/桌面/output is not empty
        Please specify another directory or run with -T.
root@kali:~/桌面# foremost f9809647382a42e5bfb64d7d447b4099.pcap
Processing: f9809647382a42e5bfb64d7d447b4099.pcap
|*|
root@kali:~/桌面# 
```

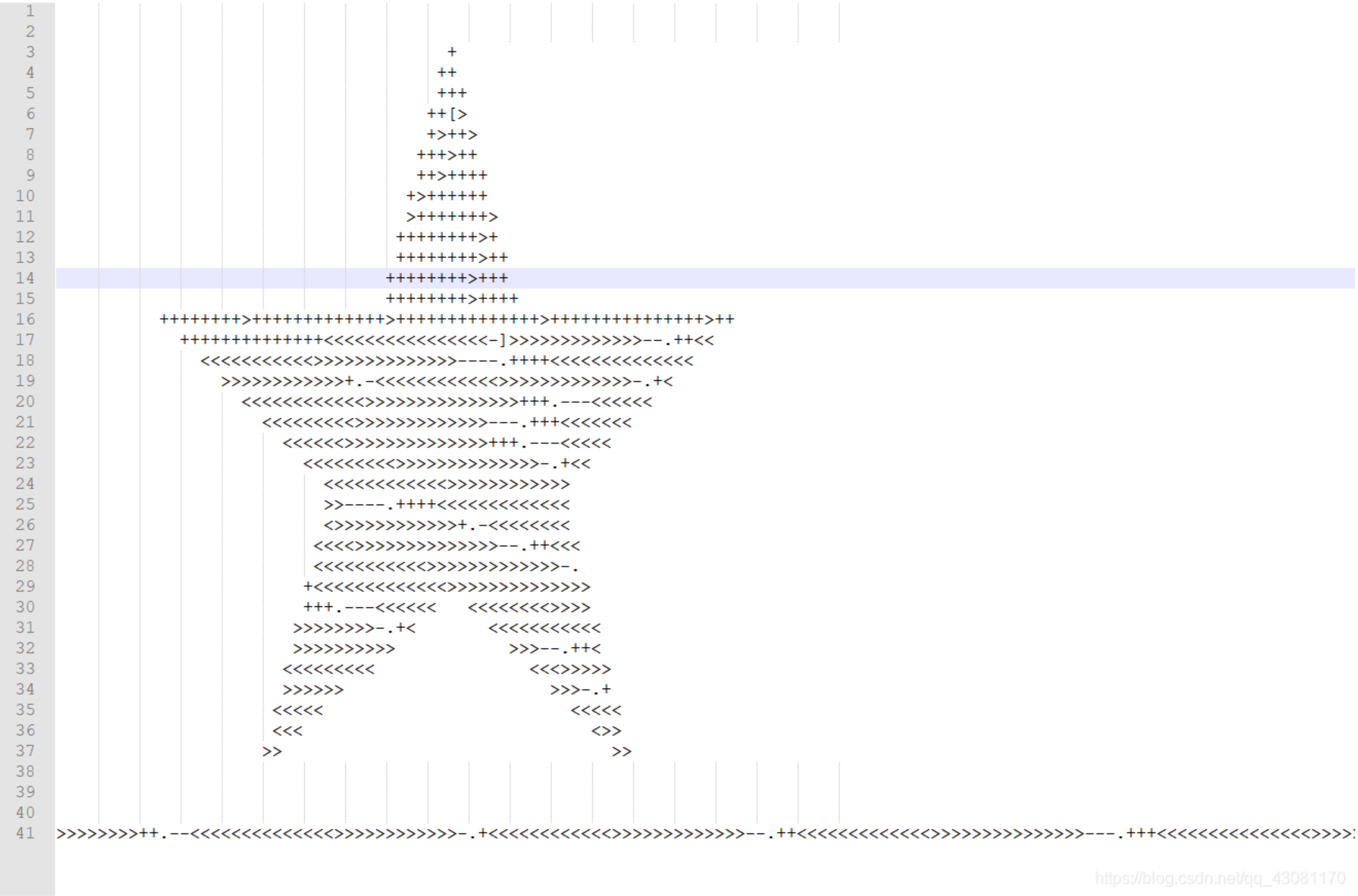
打开pdf文件得到flag。



HITB{b3d0e380e9c39352c667307d010775ca}

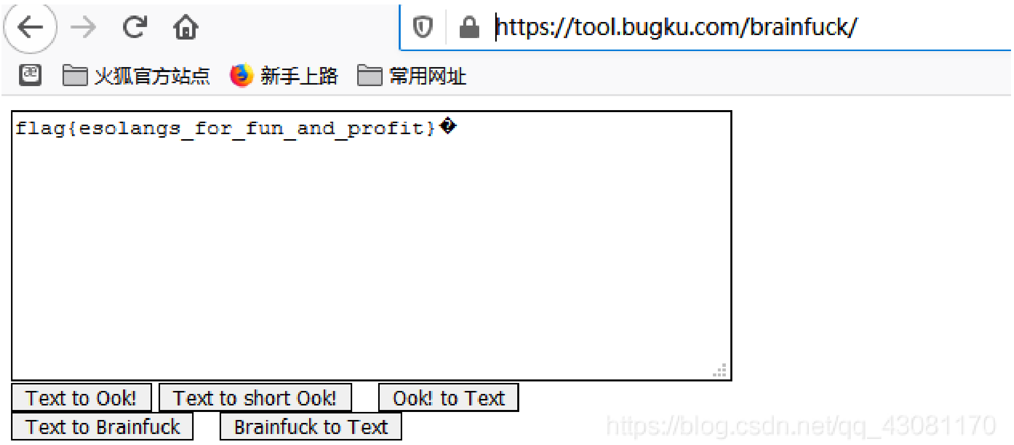
can_has_stdio?

下载附件，用Notepad++打开，



```


                                      +
                                     ++
                                     +++
                                    ++[>
                                    +>++>
                                   +++>++
                                   ++>++++
                                  +>++++++
                                  >+++++++>
                                 ++++++++>+
                                 ++++++++>++
                                ++++++++>+++
                                ++++++++>++++
          ++++++++>+++++++++++++>++++++++++++++>+++++++++++++++>++
            ++++++++++++++<<<<<<<<<<<<<<<<-]>>>>>>>>>>>>>--.++<<
              <<<<<<<<<<<>>>>>>>>>>>>>>----.++++<<<<<<<<<<<<<<
                >>>>>>>>>>>>+.-<<<<<<<<<<<<>>>>>>>>>>>>>-.+<
                  <<<<<<<<<<<<>>>>>>>>>>>>>>>+++.---<<<<<<
                    <<<<<<<<<>>>>>>>>>>>>>---.+++<<<<<<<
                      <<<<<<>>>>>>>>>>>>>>+++.---<<<<<
                        <<<<<<<<>>>>>>>>>>>>>>>-.+<<
                          <<<<<<<<<<<<>>>>>>>>>>>>
                          >>----.++++<<<<<<<<<<<<<
                          <>>>>>>>>>>>>+.-<<<<<<<<
                         <<<<>>>>>>>>>>>>>>--.++<<<
                         <<<<<<<<<<<>>>>>>>>>>>>>-.
                        +<<<<<<<<<<<<<>>>>>>>>>>>>>>
                        +++.---<<<<<<   <<<<<<<<>>>>
                       >>>>>>>>-.+<       <<<<<<<<<<<
                       >>>>>>>>>>           >>>--.++<
                      <<<<<<<<<               <<<>>>>>
                      >>>>>>                    >>>-.+
                     <<<<<                        <<<<<
                     <<<                            <>>
                    >>                                >>



>>>>>>>>>++.--<<<<<<<<<<<<<<>>>>>>>>>>>>>-.+<<<<<<<<<<<<<>>>>>>>>>>>>>>--.++<<<<<<<<<<<<<<>>>>>>>>>>>>>>>---.+++<<<<<<<<<<<<<<<<<<>>>>
```

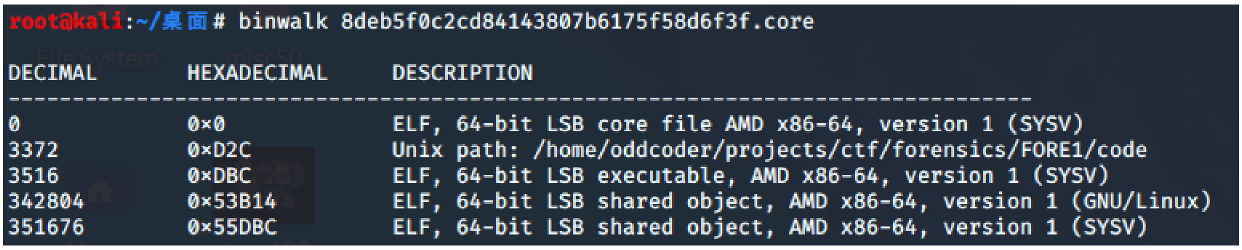
利用brainfuck编码在线解码`https://tool.bugku.com/brainfuck/`得到flag。



```
flag{esolangs_for_fun_and_profit}�
```

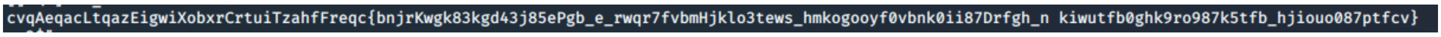
hit-the-core

下载附件，binwalk分析文件，是Linux文件，



```
root@kali:~/桌面# binwalk 8deb5f0c2cd84143807b6175f58d6f3f.core

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             ELF, 64-bit LSB core file AMD x86-64, version 1 (SYSV)
3372          0xD2C           Unix path: /home/oddcoder/projects/ctf/forensics/FORE1/code
3516          0xDBC           ELF, 64-bit LSB executable, AMD x86-64, version 1 (SYSV)
342804        0x53B14         ELF, 64-bit LSB shared object, AMD x86-64, version 1 (GNU/Linux)
351676        0x55DBC         ELF, 64-bit LSB shared object, AMD x86-64, version 1 (SYSV)
```

那个strings查看，有一串很长的字符串，从第三位开始，每5位进行提取得到flag。



```
cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrKwgk83kgd43j85ePgb_e_rwqr7fvbmHjklo3tews_hmkogooyf0vbnk0ii87Drfgh_n kiwutfb0ghk9ro987k5tfb_hjiouo087ptfcv}
```

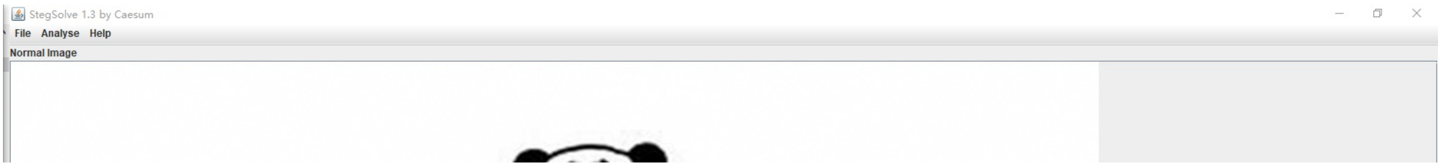
附上提取脚本：

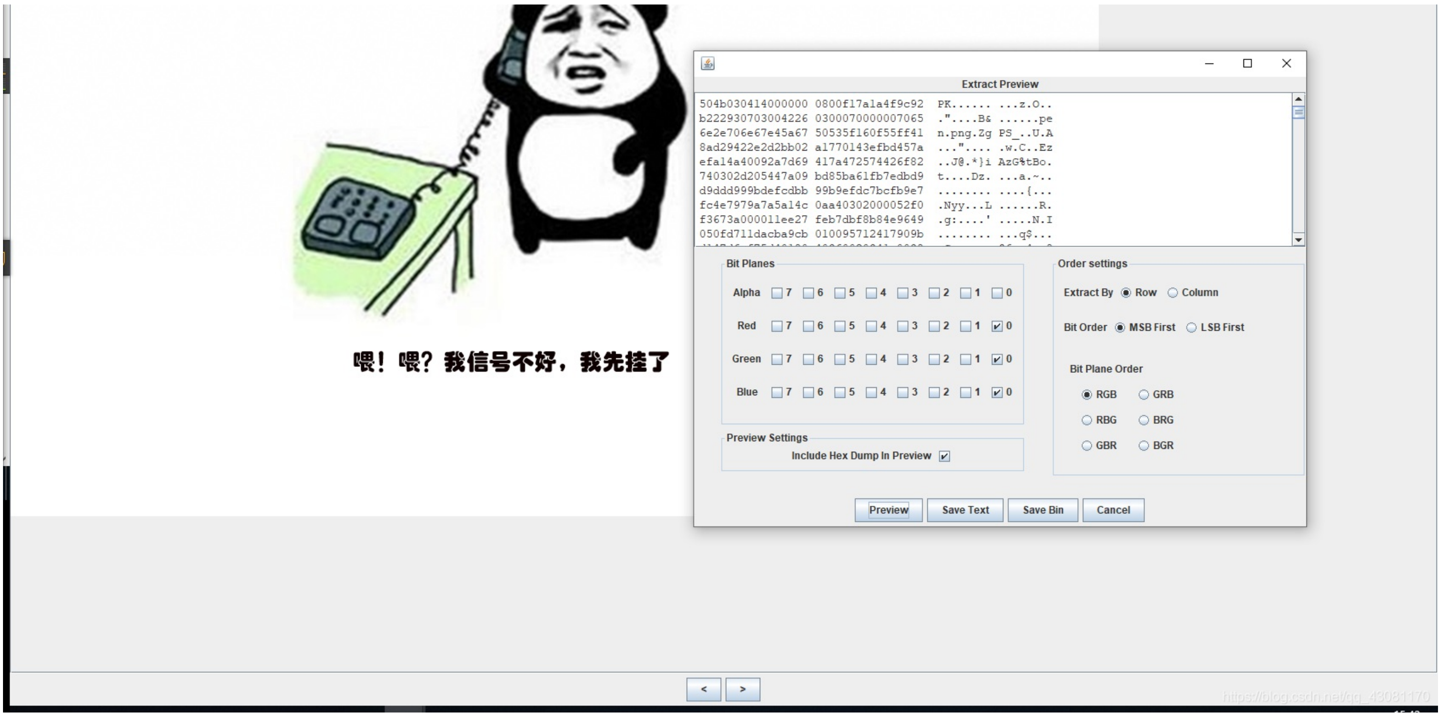
```
str = 'cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrKwgk83kgd43j85ePgb_e_rwqr7fvbmHjklo3tews_hmkogooyf0vbnk0ii87Dr
fgh_n kiwutfb0ghk9ro987k5tfb_hjiouo087ptfcv}'
flag = ''
for i in range(3,len(str),5):
 flag += str[i]
print flag
```

信号不好先挂了

下载附件解压缩得到一张图片apple.png，

用Stegsolve打开，





Extract Preview

```
504b030414000000 0800f17a1a4f9c92  PK...... ...z.O..
b222930703004226 0300070000007065  ."....B& ......pe
6e2e706e67e45a67 50535f160f55ff41  n.png.Zg PS_..U.A
8ad29422e2d2bb02 a1770143efbd457a  ..."....  .w.C..Ez
efa14a40092a7d69 417a472574426f82  ..J@.*}i AzG%tBo.
740302d205447a09 bd85ba61fb7edbd9  t....Dz. ...a.~..
d9ddd999bdefcdbb 99b9efdc7bcfb9e7  ........ ....{...
fc4e7979a7a5a14c 0aa40302000052f0  .Nyy...L ......R.
f3673a00011ee27 feb7dbf8b84e9649   .g:....' .....N.I
050fd711dacba9cb 010095712417909b  ........ ...q$...
```

Bit Planes

Alpha □ 7 □ 6 □ 5 □ 4 □ 3 □ 2 □ 1 □ 0

Red □ 7 □ 6 □ 5 □ 4 □ 3 □ 2 □ 1 ☑ 0

Green □ 7 □ 6 □ 5 □ 4 □ 3 □ 2 □ 1 ☑ 0

Blue □ 7 □ 6 □ 5 □ 4 □ 3 □ 2 □ 1 ☑ 0

Preview Settings

Include Hex Dump In Preview ☑

Order settings

Extract By ● Row ○ Column

Bit Order ● MSB First ○ LSB First

Bit Plane Order

● RGB ○ GRB

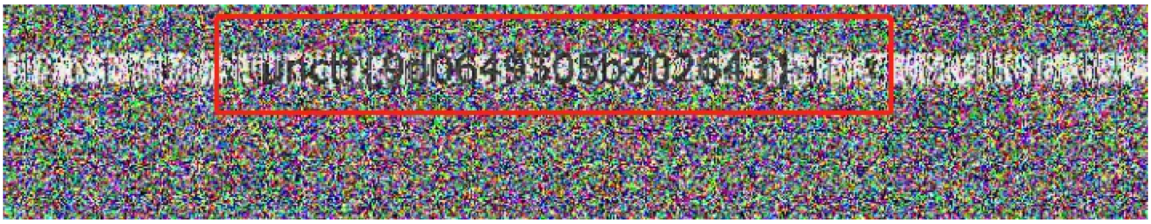
○ RBG ○ BRG

○ GBR ○ BGR

得到一个压缩包，解压缩，得到一个看似和apple.png的图片，

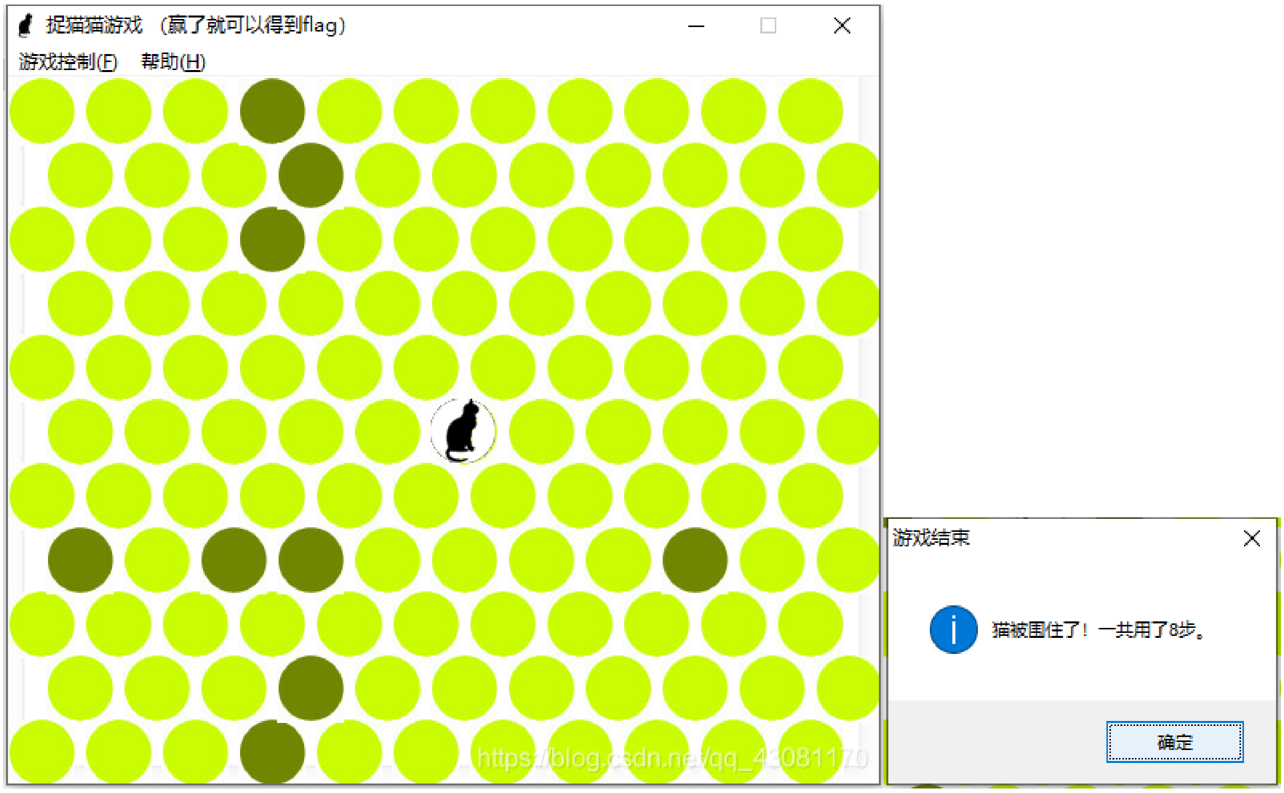
然后利用盲水印脚本得到flag。

```
python bwn.py decode apple.png pen.png apple_pen.png
```

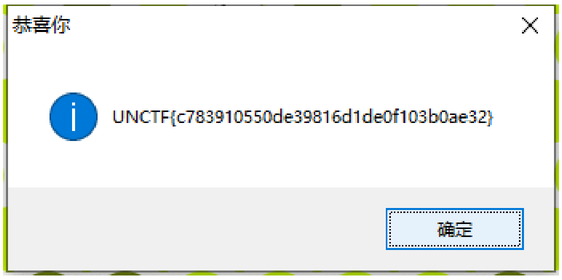


快乐游戏题

下载附件解压缩，是一个exe的文件，运行，将猫围住即可得到flag。



猫被围住了！一共用了8步。

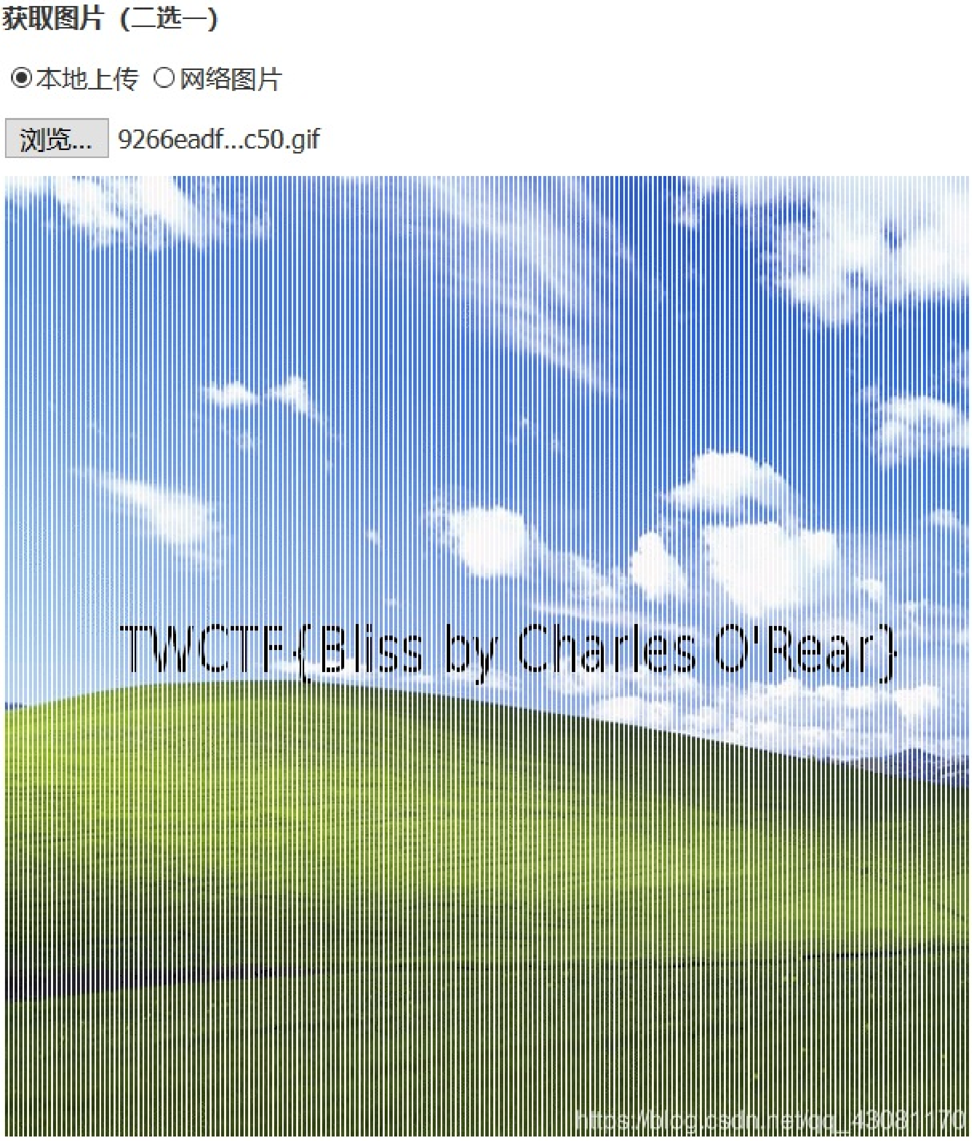


UNCTF{c783910550de39816d1de0f103b0ae32}

glance-50

下载附件是一个动态图片，

利用在线动图分帧合并工具，得到flag。



获取图片（二选一）

● 本地上传 ○ 网络图片

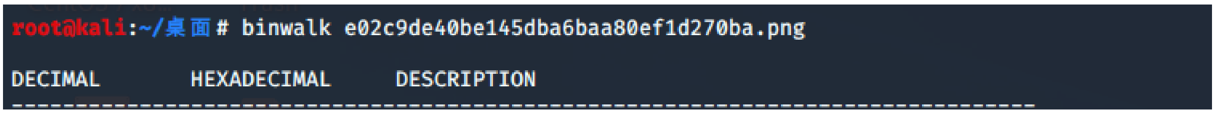
浏览... 9266eadf...c50.gif

TWCTF{Bliss by Charles O'Rear}
https://blog.csdn.net/qq_43081170

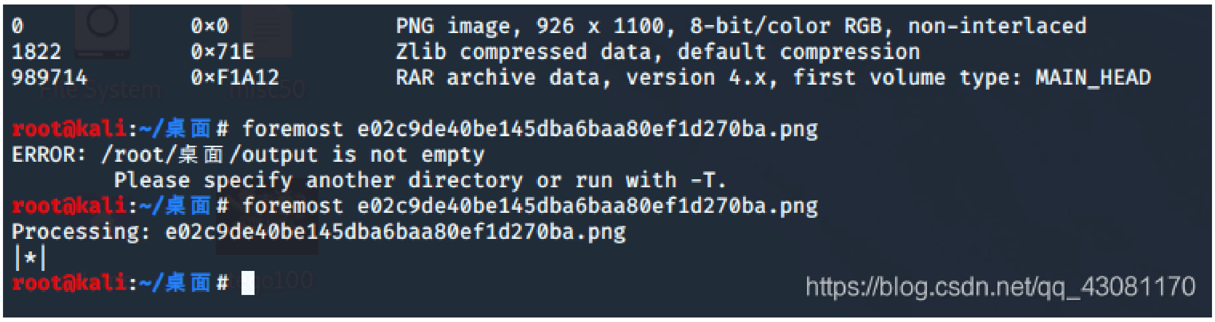
Ditf

下载附件是一张png图片，

图片有点大怀疑有隐藏文件，用binwalk分析文件，发现一个隐藏有一个压缩包，foremost文件分离，



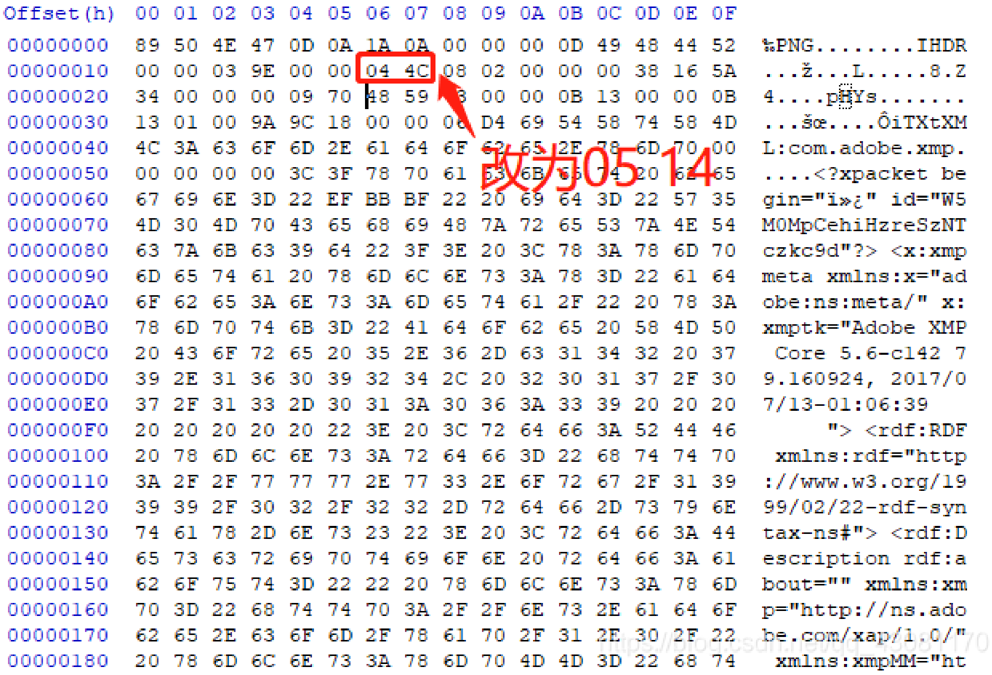
```
root@kali:~/桌面# binwalk e02c9de40be145dba6baa80ef1d270ba.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
```



```
0             0x0             PNG image, 926 x 1100, 8-bit/color RGB, non-interlaced
1822          0x71E           Zlib compressed data, default compression
989714        0xF1A12         RAR archive data, version 4.x, first volume type: MAIN_HEAD

root@kali:~/桌面# foremost e02c9de40be145dba6baa80ef1d270ba.png
ERROR: /root/桌面/output is not empty
        Please specify another directory or run with -T.
root@kali:~/桌面# foremost e02c9de40be145dba6baa80ef1d270ba.png
Processing: e02c9de40be145dba6baa80ef1d270ba.png
|*|
root@kali:~/桌面# 
```

解压缩需要密码，

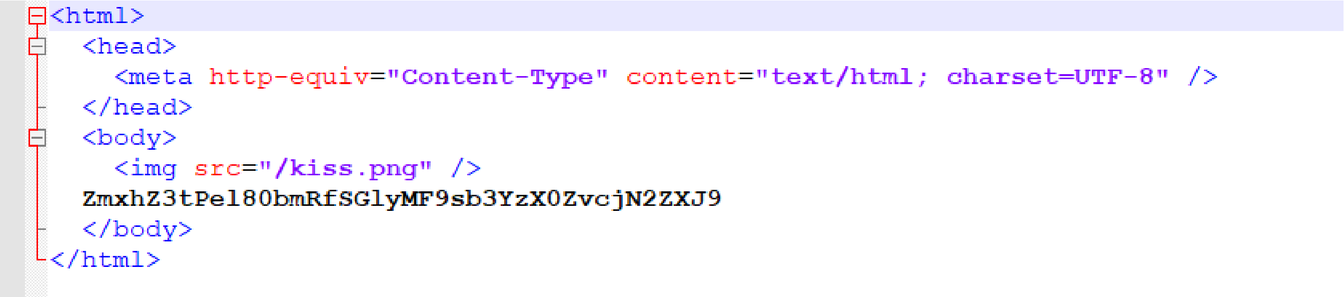
修改图片像素高度，得到解压缩密码，



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52  ‰PNG........IHDR
00000010  00 00 03 9E 00 00 04 4C 08 02 00 00 00 38 16 5A  ...ž...L.....8.Z
00000020  34 00 00 00 09 70 48 59 73 00 00 0B 13 00 00 0B  4....pHYs.......
00000030  13 01 00 9A 9C 18 00 00 00 D4 69 54 58 74 58 4D  ...šœ....ÔiTXtXM
00000040  4C 3A 63 6F 6D 2E 61 64 6F 62 65 2E 78 6D 70 00  L:com.adobe.xmp.
00000050  00 00 00 00 3C 3F 78 70 61 63 6B 65 74 20 62 65  ....<?xpacket be
00000060  67 69 6E 3D 22 EF BB BF 22 20 69 64 3D 22 57 35  gin="ï»¿" id="W5
00000070  4D 30 4D 70 43 65 68 69 48 7A 72 65 53 7A 4E 54  M0MpCehiHzreSzNT
00000080  63 7A 6B 63 39 64 22 3F 3E 20 3C 78 3A 78 6D 70  czkc9d"?> <x:xmp
00000090  6D 65 74 61 20 78 6D 6C 6E 73 3A 78 3D 22 61 64  meta xmlns:x="ad
000000A0  6F 62 65 3A 6E 73 3A 6D 65 74 61 2F 22 20 78 3A  obe:ns:meta/" x:
000000B0  78 6D 70 74 6B 3D 22 41 64 6F 62 65 20 58 4D 50  xmptk="Adobe XMP
000000C0  20 43 6F 72 65 20 35 2E 36 2D 63 31 34 32 20 37   Core 5.6-c142 7
000000D0  39 2E 31 36 30 39 32 34 2C 20 32 30 31 37 2F 30  9.160924, 2017/0
000000E0  37 2F 31 33 2D 30 31 3A 30 36 3A 33 39 20 20 20  7/13-01:06:39   
000000F0  20 20 20 20 20 22 3E 20 3C 72 64 66 3A 52 44 46       "> <rdf:RDF
00000100  20 78 6D 6C 6E 73 3A 72 64 66 3D 22 68 74 74 70   xmlns:rdf="http
00000110  3A 2F 2F 77 77 77 2E 77 33 2E 6F 72 67 2F 31 39  ://www.w3.org/19
00000120  39 39 2F 30 32 2F 32 32 2D 72 64 66 2D 73 79 6E  99/02/22-rdf-syn
00000130  74 61 78 2D 6E 73 23 22 3E 20 3C 72 64 66 3A 44  tax-ns#"> <rdf:D
00000140  65 73 63 72 69 70 74 69 6F 6E 20 72 64 66 3A 61  escription rdf:a
00000150  62 6F 75 74 3D 22 22 20 78 6D 6C 6E 73 3A 78 6D  bout="" xmlns:xm
00000160  70 3D 22 68 74 74 70 3A 2F 2F 6E 73 2E 61 64 6F  p="http://ns.ado
00000170  62 65 2E 63 6F 6D 2F 78 61 70 2F 31 2E 30 2F 22  be.com/xap/1.0/"
00000180  20 78 6D 6C 6E 73 3A 78 6D 70 4D 4D 3D 22 68 74   xmlns:xmpMM="ht
```

改为0514

解压缩得到`Ditf.pcapng`，用wireshark打开，将HTTP对象导出，有一个html的文件，



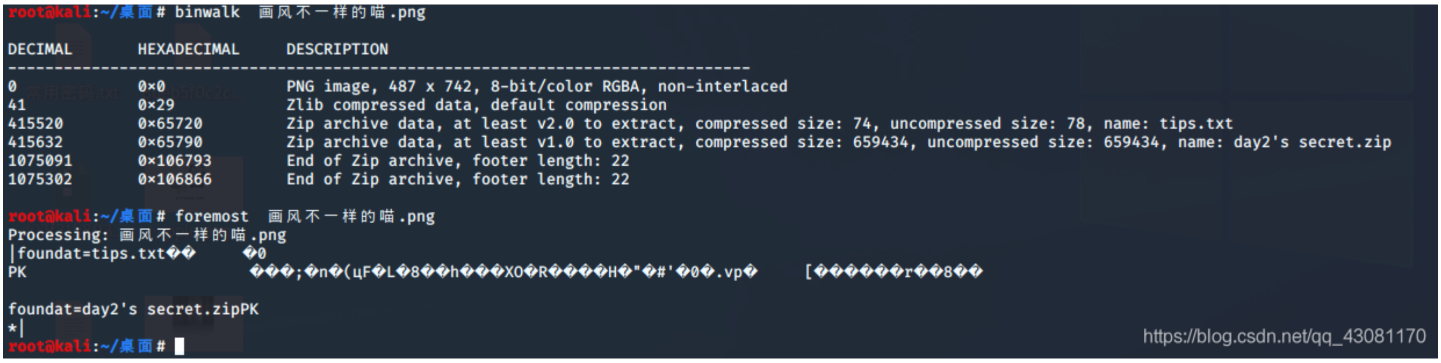
```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  </head>
  <body>
    <img src="/kiss.png" />
  ZmxhZ3tPel80bmRfSGlyMF9sb3YzX0ZvcjN2ZXJ9
  </body>
</html>
```

将`ZmxhZ3tPel80bmRfSGlyMF9sb3YzX0ZvcjN2ZXJ9`base64解码得到flag。

4-1

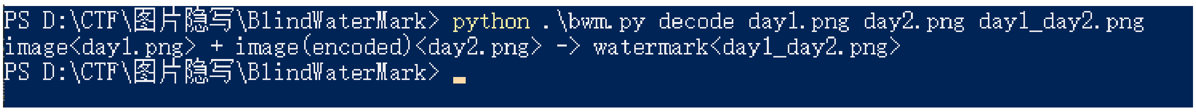
下载附件解压缩，是一张图片，

用binwalk分析文件，发现隐藏有zip压缩包，foremost分离，



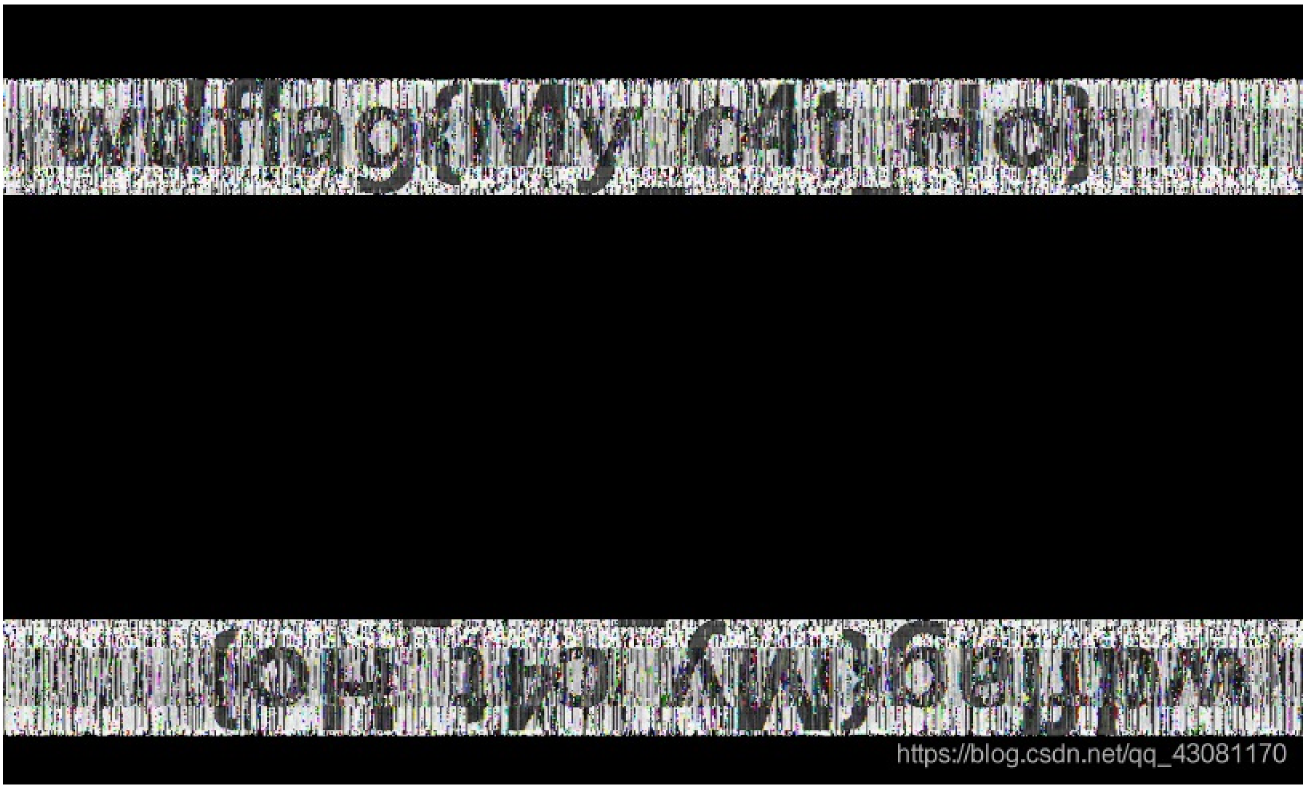
```
root@kali:~/桌面# binwalk  画风不一样的喵.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 487 x 742, 8-bit/color RGBA, non-interlaced
41            0x29            Zlib compressed data, default compression
415520        0x65720         Zip archive data, at least v2.0 to extract, compressed size: 74, uncompressed size: 78, name: tips.txt
415632        0x65790         Zip archive data, at least v1.0 to extract, compressed size: 659434, uncompressed size: 659434, name: day2's secret.zip
1075091       0x106793        End of Zip archive, footer length: 22
1075302       0x106866        End of Zip archive, footer length: 22

root@kali:~/桌面# foremost  画风不一样的喵.png
Processing: 画风不一样的喵.png
|foundat=tips.txt��     �0
PK                       ���;�n�(цF�L�8��h���XO�R����H�"�#'�0�.vp�     [������r��8��

foundat=day2's secret.zipPK
*|
root@kali:~/桌面# 
```

解压缩后又得到一个压缩包和txt文件，再一次解压缩得到两张看似相同的图片，
用脚本跑盲水印得到图片flag。



```
PS D:\CTF\图片隐写\BlindWaterMark> python .\bwm.py decode day1.png day2.png day1_day2.png
image<day1.png> + image(encoded)<day2.png> -> watermark<day1_day2.png>
PS D:\CTF\图片隐写\BlindWaterMark> _
```

[redacted]



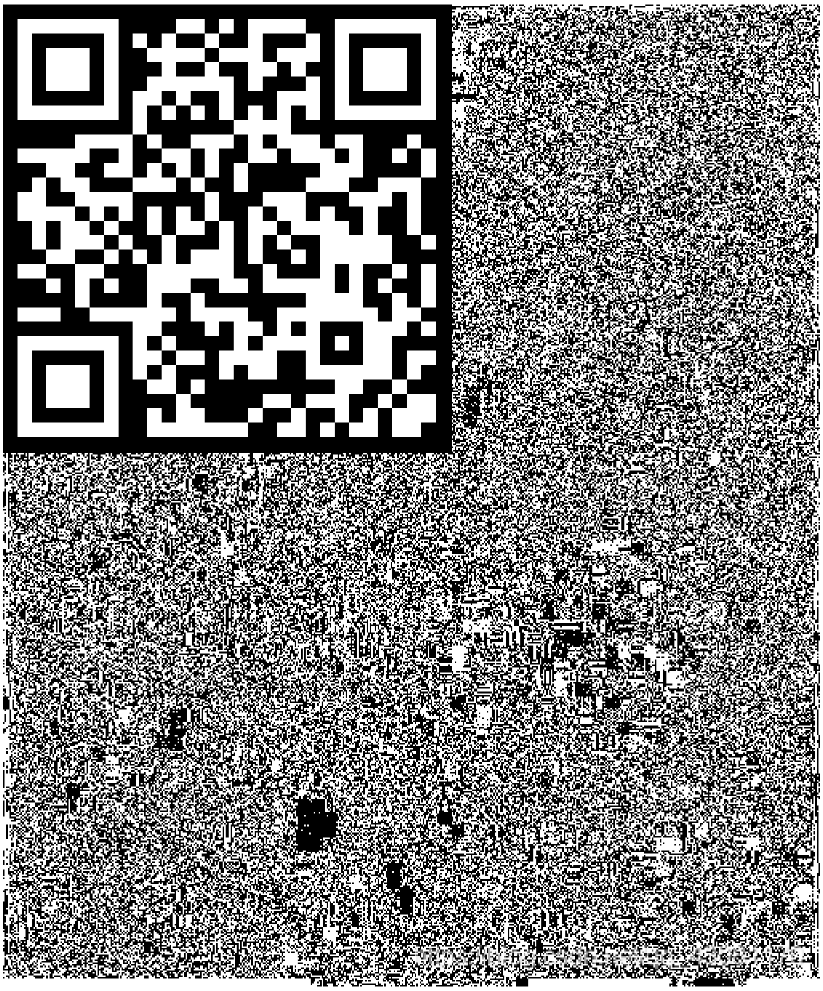
low

下载附件，解压缩得到一个bmp图片，

LSB隐写，脚本如下：

```
import PIL.Image as Image    #需安装PIL模块，如果安装失败可以安装pillow模块
img = Image.open('low.bmp')
img_tmp = img.copy()
pix = img_tmp.load()
width,height = img_tmp.size
for w in range(width):
   for h in range(height):
      if pix[w,h]&1 == 0:
         pix[w,h] = 0
      else:
         pix[w,h] = 255
img_tmp.show()
```

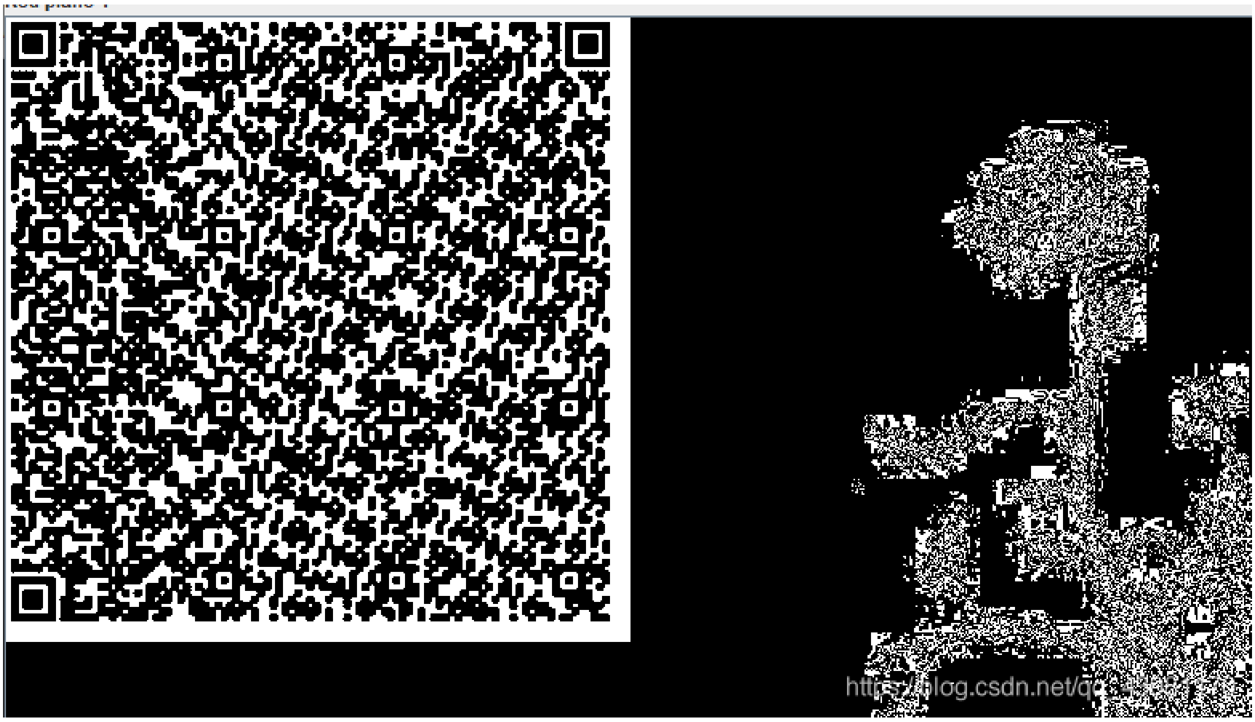
生成一个二维码图片，扫描二维码得到flag。



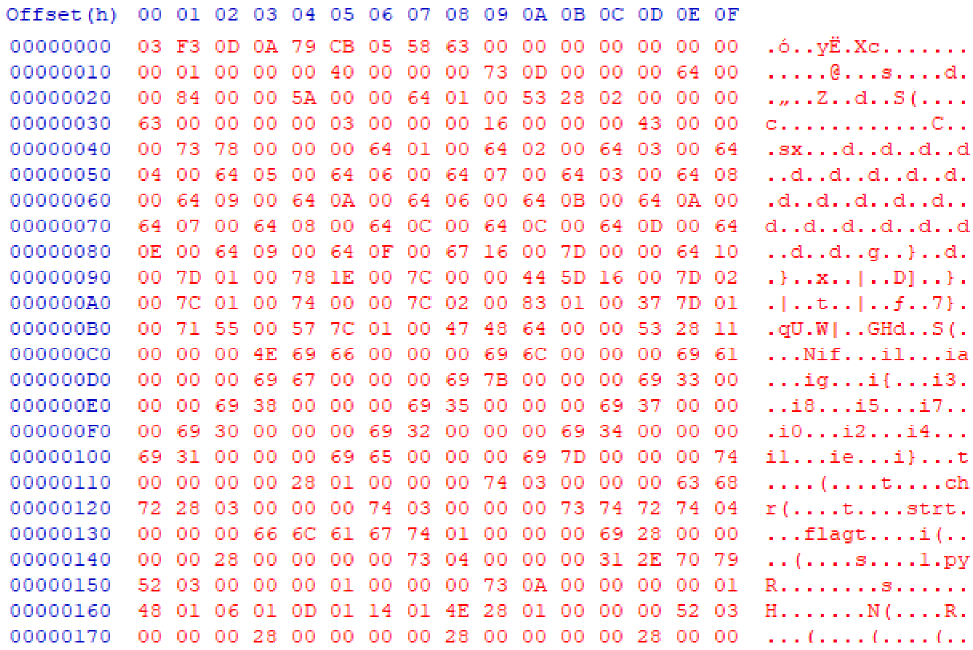
适合作为桌面

下载附件解压缩得到一张图片，

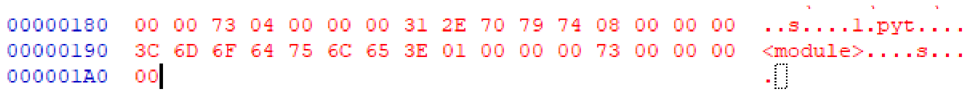
用StegSolve打开得到一张二维码，



扫描二维码得到一串十六进制字符串，放到winhex中，保存为`.pyc`格式，



```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   03 F3 0D 0A 79 CB 05 58 63 00 00 00 00 00 00 00   .ó..yË.Xc.......
00000010   00 01 00 00 00 40 00 00 00 73 0D 00 00 00 64 00   .....@...s....d.
00000020   00 84 00 00 5A 00 00 64 01 00 53 28 02 00 00 00   .„..Z..d..S(....
00000030   63 00 00 00 00 03 00 00 00 16 00 00 00 43 00 00   c............C..
00000040   00 73 78 00 00 00 64 01 00 64 02 00 64 03 00 64   .sx...d..d..d..d
00000050   04 00 64 05 00 64 06 00 64 07 00 64 03 00 64 08   ..d..d..d..d..d.
00000060   00 64 09 00 64 0A 00 64 06 00 64 0B 00 64 0A 00   .d..d..d..d..d..
00000070   64 07 00 64 08 00 64 0C 00 64 0C 00 64 0D 00 64   d..d..d..d..d..d
00000080   0E 00 64 09 00 64 0F 00 67 16 00 7D 00 00 64 10   ..d..d..g..}..d.
00000090   00 7D 01 00 78 1E 00 7C 00 00 44 5D 16 00 7D 02   .}..x..|..D]..}.
000000A0   00 7C 01 00 74 00 00 7C 02 00 83 01 00 37 7D 01   .|..t..|..ƒ..7}.
000000B0   00 71 55 00 57 7C 01 00 47 48 64 00 00 53 28 11   .qU.W|..GHd..S(.
000000C0   00 00 00 4E 69 66 00 00 00 69 6C 00 00 00 69 61   ...Nif...il...ia
000000D0   00 00 00 69 67 00 00 00 69 7B 00 00 00 69 33 00   ...ig...i{...i3.
000000E0   00 00 69 38 00 00 00 69 35 00 00 00 69 37 00 00   ..i8...i5...i7..
000000F0   00 69 30 00 00 00 69 32 00 00 00 69 34 00 00 00   .i0...i2...i4...
00000100   69 31 00 00 00 69 65 00 00 00 69 7D 00 00 00 74   i1...ie...i}...t
00000110   00 00 00 00 28 01 00 00 00 74 03 00 00 00 63 68   ....(....t....ch
00000120   72 28 03 00 00 00 74 03 00 00 00 73 74 72 74 04   r(....t....strt.
00000130   00 00 00 66 6C 61 67 74 01 00 00 00 69 28 00 00   ...flagt....i(..
00000140   00 00 28 00 00 00 00 73 04 00 00 00 31 2E 70 79   ..(....s....1.py
00000150   52 03 00 00 00 01 00 00 00 73 0A 00 00 00 00 01   R........s......
00000160   48 01 06 01 0D 01 14 01 4E 28 01 00 00 00 52 03   H.......N(....R.
00000170   00 00 00 28 00 00 00 00 28 00 00 00 00 28 00 00   ...(....(....(..
```



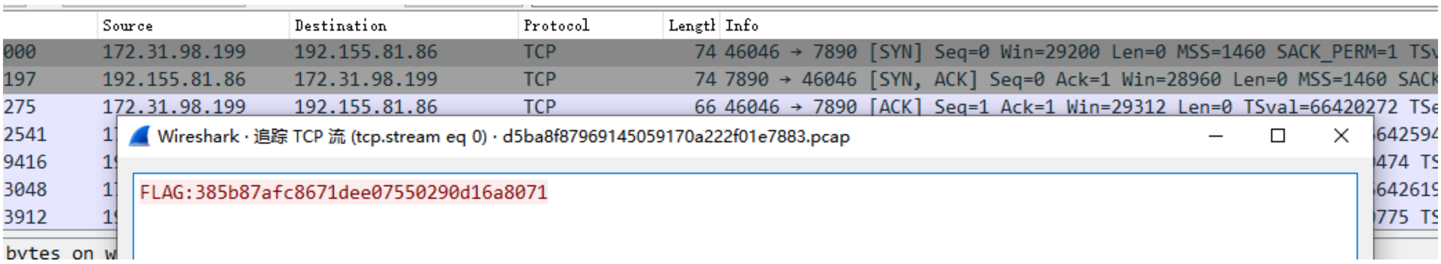
```
00000180  00 00 73 04 00 00 00 31 2E 70 79 74 08 00 00 00  ..s....1.pyt....
00000190  3C 6D 6F 64 75 6C 65 3E 01 00 00 00 73 00 00 00  <module>....s...
000001A0  00                                               .
```

用EasyPythonDecompiler转化为py格式文件。整理运行得到flag。

```
# Embedded file name: 1.py
str = [102,108,97,103,123,51,56,97,53,55,48,51,50,48,56,53,52,52,49,101,55,125]
flag = ''
for i in str:
    flag += chr(i)
print flag
```

easycap

下载附件用wireshark打开，全部为TCP协议的数据包，追踪TCP流，得到flag。



```
     Source         Destination     Protocol  Length Info
000  172.31.98.199  192.155.81.86   TCP       74 46046 → 7890 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSv
197  192.155.81.86  172.31.98.199   TCP       74 7890 → 46046 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK
275  172.31.98.199  192.155.81.86   TCP       66 46046 → 7890 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=66420272 TSe
2541 17                                                                                                 642594
9416 19                                                                                                 474 TS
3048 17                                                                                                 642619
3912 19                                                                                                 775 TS
bytes on w
```

Wireshark · 追踪 TCP 流 (tcp.stream eq 0) · d5ba8f87969145059170a222f01e7883.pcap

```
FLAG:385b87afc8671dee07550290d16a8071
```