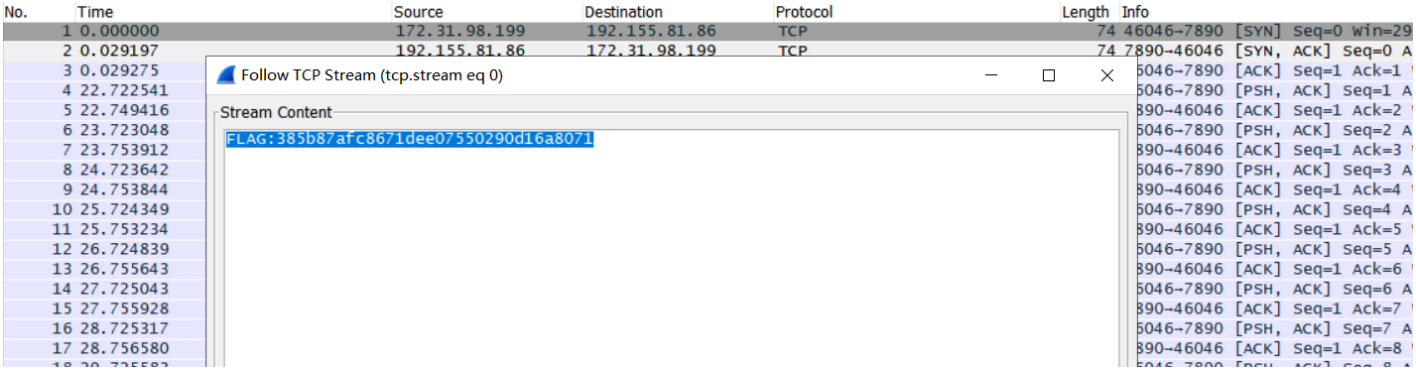
攻防世界misc高手区部分writeup


```
title: 攻防世界misc高手区部分writeup
tags: ctf
categories: ctf
```

1.easycap

下载下来是一个流量包，用wireshark打开，搜索关键字flag，第二行追踪tcp流，



2.Avatar

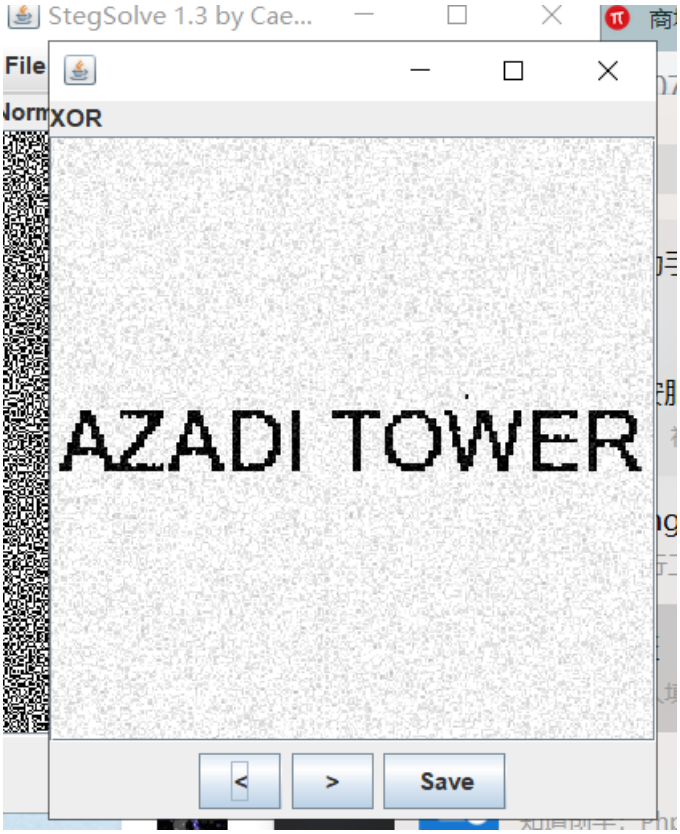
下载下来是一张图片，这里考察outguess算法，在kali下载outguess，`apt-get install outguess`，

"We should blow up the bridge at midnight"即为flag。

```
root@kali:~/Downloads/misc higher# outguess -r b985d98d87b24ff1b17fc02ffc446b2c.jpg out
Reading b985d98d87b24ff1b17fc02ffc446b2c.jpg....
Extracting usable bits:   28734 bits
Steg retrieve: seed: 94, len: 41
root@kali:~/Downloads/misc higher# cat out
We should blow up the bridge at midnight
```

3.What-is-this

下载个解压包，解压后是两张图片，用stegsolve打开其中一张，用image combiner进行xor运算，AZADI TOWER即为flag



4.Get-the-key.txt

下载来一个解压包，解压后的文件不知道是什么，用file命令查看一下，是磁盘文件，那就先挂载下

```
root@kali:~/Downloads/misc higher/4# file forensic100
forensic100: Linux rev 1.0 ext2 filesystem data, UUID=0b92a753-7ec9-4b20-8c0b-79c1fa140869
root@kali:~/Downloads/misc higher/4# mount forensic100 /mnt/
root@kali:~/Downloads/misc higher/4#
```

下载后有一大堆乱七八糟，用grep命令查找关键字“key.txt”，在1文件，cat查看是乱码，file查看是什么文件，是压缩文件，gunzip查看flag

```
root@kali:/mnt# grep -r key.txt
匹配到二进制文件 1
root@kali:/mnt# file 1
1: gzip compressed data, was "key.txt", last modified: Wed Oct  1 06:00:52 2014, from Unix, original
size 30
root@kali:/mnt# gunzip < 1
SECCON{@]NL7n+-s75FrET]vU=7Z}
root@kali:/mnt# ls
1    110  122  134  146  158  17   181  193  204  216  228  24   31   43   55   67   79   90
10   111  123  135  147  159  170  182  194  205  217  229  240  32   44   56   68   8    91
100  112  124  136  148  16   171  183  195  206  218  23   241  33   45   57   69   80   92
101  113  125  137  149  160  172  184  196  207  219  230  242  34   46   58   7    81   93
102  114  126  138  15   161  173  185  197  208  22   231  243  35   47   59   70   82   94
103  115  127  139  150  162  174  186  198  209  220  232  244  36   48   6    71   83   95
104  116  128  14   151  163  175  187  199  21   221  233  25   37   49   60   72   84   96
105  117  129  140  152  164  176  188  2    210  222  234  26   38   5    61   73   85   97
106  118  13   141  153  165  177  189  20   211  223  235  27   39   50   62   74   86   98
107  119  130  142  154  166  178  19   200  212  224  236  28   4    51   63   75   87   99
108  12   131  143  155  167  179  190  201  213  225  237  29   40   52   64   76   88   lost+found
109  120  132  144  156  168  18   191  202  214  226  238  3    41   53   65   77   89
11   121  133  145  157  169  180  192  203  215  227  239  30   42   54   66   78   9
```

5.签到题

将`Z2dRQGdRMWZxaDBvaHRqcHRfc3d7Z2ZoZ3MjfQ==`base64转码为`ggQ@gQ1fqh0ohtjpt_sw{gfhgs#}`,是凯撒+栅栏，解题关键是flag格式ssctf{}，所以先凯撒得到`ssC@sC1rct0atfvbf_ei{srtse#}`栅栏得到`ssctf{ssCtf_seC10ver#@rabit}`

6.Training-Stegano-1

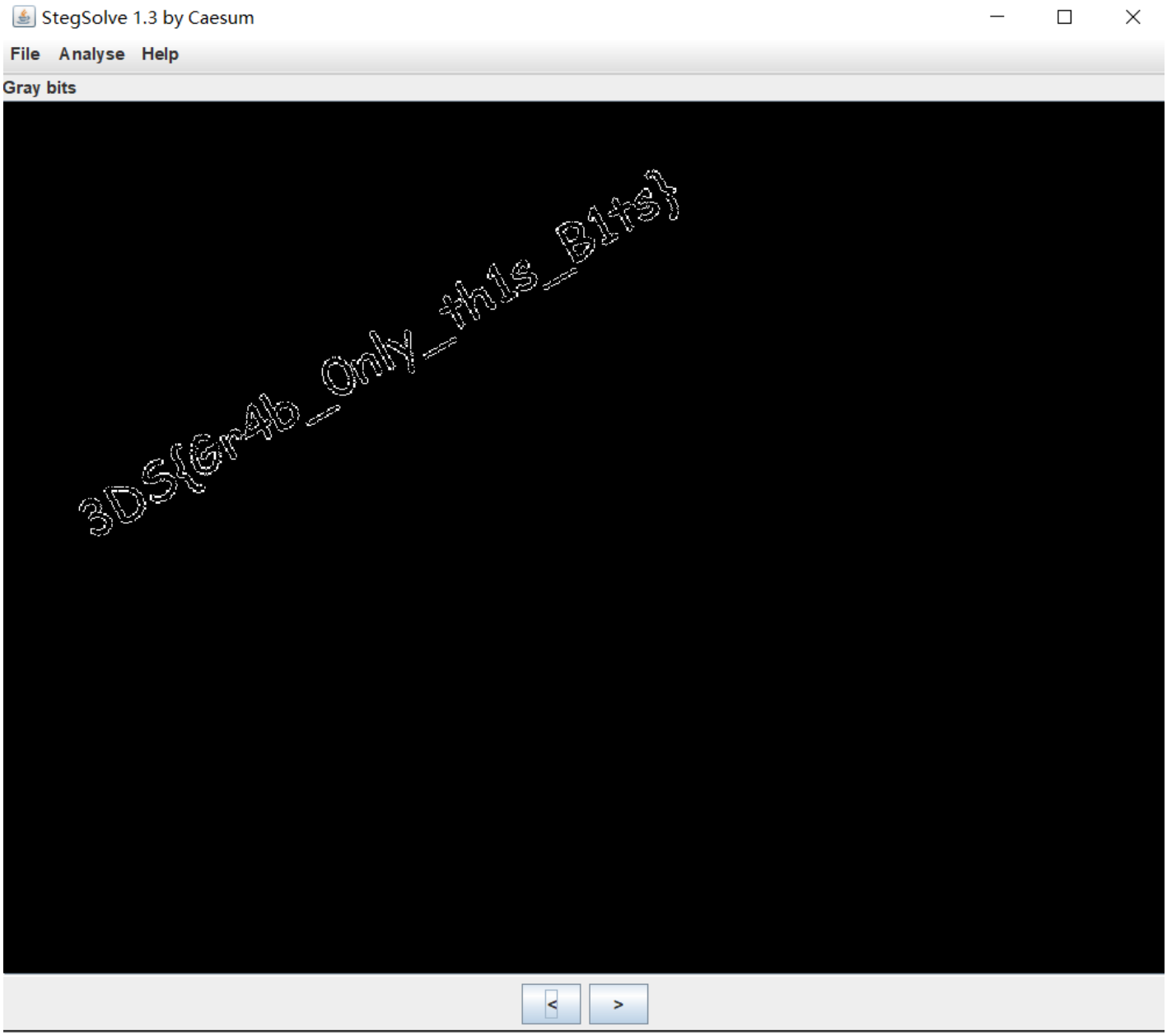
下载是一张图片，记事本打开，文本显示"passwd:steganol",flag即为steganol

7.Test-flag-please-ignore

将`666c61677b68656c6c6f5f776f726c647d`转换为16进制字符串flag{hello_world}

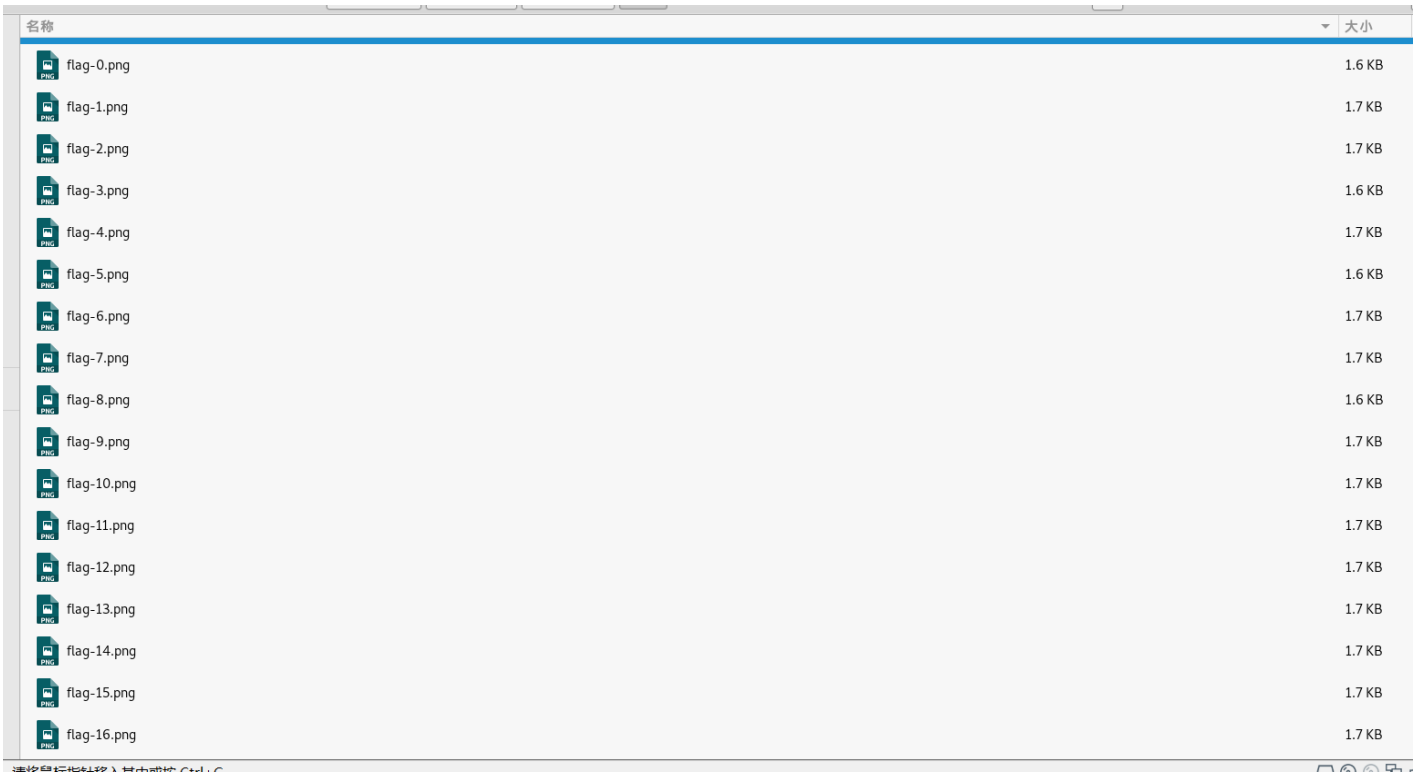
8.Excaliflag

下载图片后，用stegSolve打开图片



9.glance-50

下载来是一张gif图片，分解gif，使用convert命令分解，`convert 33e3d14fb67a44f4ad1378149fff1d9a.gif flag.png`共分解出200张图片



连接这些图片，使用montage 命令，`montage flag*.png -tile x1 -geometry +0+0 flag.png`

-tile是拼接时每行和每列的图片数，这里用x1，就是只一行

-geometry是首选每个图和边框尺寸，我们边框为0，图照原始尺寸即可

*这里参考了：https://blog.csdn.net/zz_Caleb/article/details/89490494

10.4-2

下载后是一个txt文件，里面是不知道是什么，到这个网站https://quipqiup.com/进行词频分析，flag即为flag{classical-cipher_is_not_security_hs}

quipqiup BETA

quipqiup is a fast and automated cryptogram solver by Edwin Olson. It can solve simple substitution ciphers often found in newspapers, including puzzles like cryptoquips (in which word boundaries are preserved) and patristocrats (inwhi chwor dboun darie saren t).

Puzzle:

```
Eg qnlyjtcnzydl z umaujejmjetg qeydsn eu z bsjdtx tw sgqtxegc al kdeqd mgeju tw yrzegjsoj zns nsyrzqsx kejd qeydsnjsoj
Ew ltm fgtk jds kzl tw sgqtxegc m kerr csj jds wrzc kdeqd eu qrzuueqzr-qeydsn_eu_gtj_usqmnejl_du
```

Clues: For example G=R QVW=THE

auto

Solve

0	-1.593	In cryptography a substitution cipher is a ?ethod of encoding by which units of plaintext are replaced with ciphertext If you know the way of encoding u will get the flag which is classical-cipher_is_not_security_hs
1	-1.785	In cryptography a substitution cipher is a ?ethod of encoding by which units of plaintext are replaced with ciphertext If you ?now the way of encoding u will get the flag which is classical-cipher_is_not_security_hs
2	-1.889	In cryptography a sumstitution cipher is a ?ethod of encoding my which units of plaintext are replaced with ciphertext If you ?now the way of encoding u will get the flag which is classical-cipher_is_not_security_hs
3	-1.944	In cryptography a substitution cipher is a ?etho? of enco?ing by which units of plaintext are replace? with ciphertext If you ?now the way of enco?ing u will get the flag which is classical-cipher_is_not_security_hs

11.misc1

misc1 最佳Writeup由admin提供

WP 建议

难度系数: ★1.0

题目来源: DDCTF-2018

题目描述: d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9e1e6b3e3b9e4b3b7b7e2b6b1e4b2b6b9e2b1b1b3b3b7e6b3b3b0e3b9b3b5e6fd

题目场景: 暂无

题目附件: 暂无

猜测16进制转换字符串，直接转失败转不出来，flag为DDCTF{9af3c9d377b61d269b11337f330c935f}

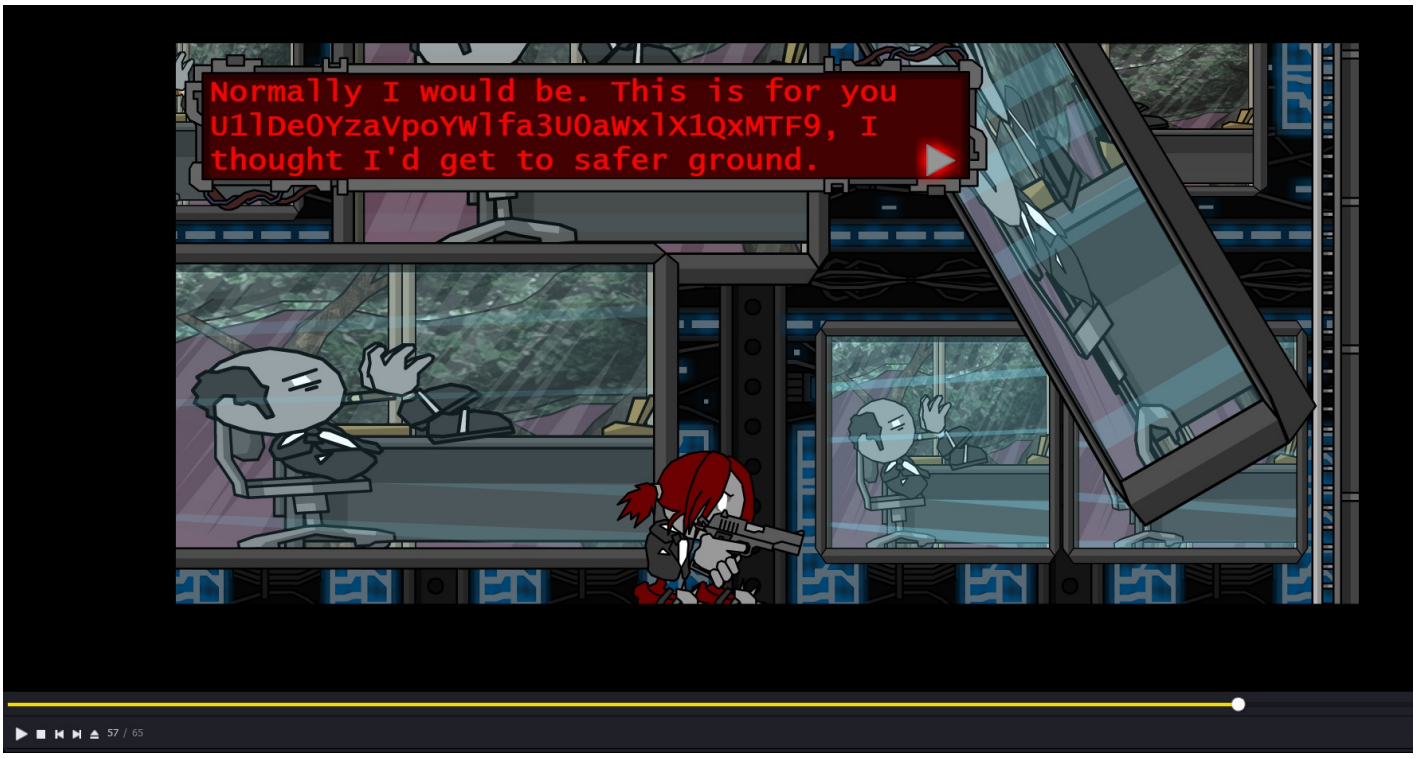
```
string="d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9e1e6b3e3b9e4b3b7b7e2b6b1e4b2b
flag=""
for i in range (0,len(string),2):
    s = "0x" + string[i] + string[i+1]
    flag += chr(int(s,16) -128)
print(flag)
```

12.eczmbarrass

下载压缩包解压后是一个流量包，用wireshark打开，搜索关键词flag后，追踪tcp流，在里面搜索flag，可以看到flag

13.肥宅快乐题

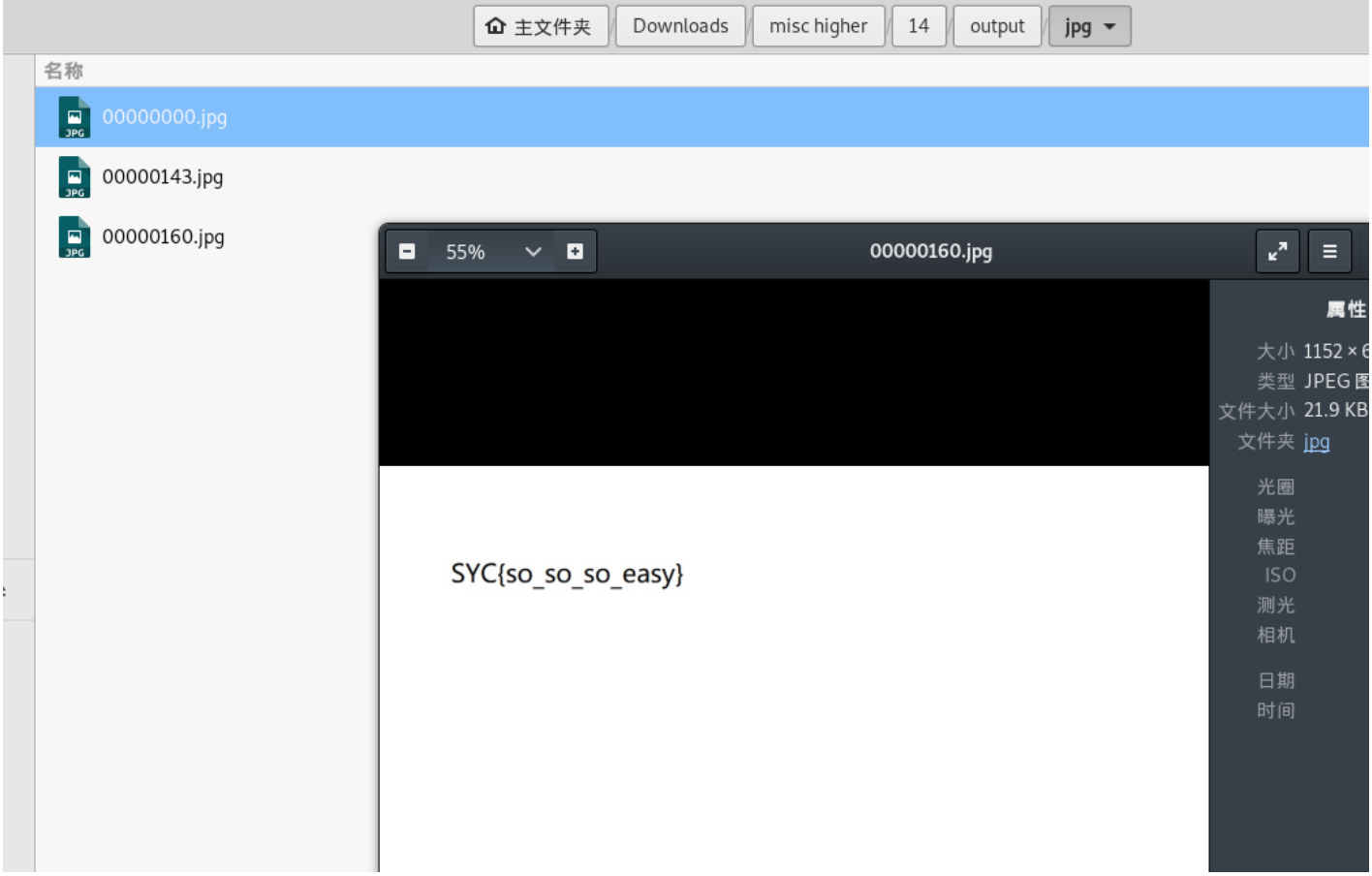
下载后是一个swf文件，是一个游戏，用potplayer打开定位57帧，出现对话里面有U1lDe0YzaVpoYWlfa3U0aWxlX1QxMTF9，base64解码后为SYC{F3iZhai_ku4ile_T111}，即为flag



14.小小的PDF

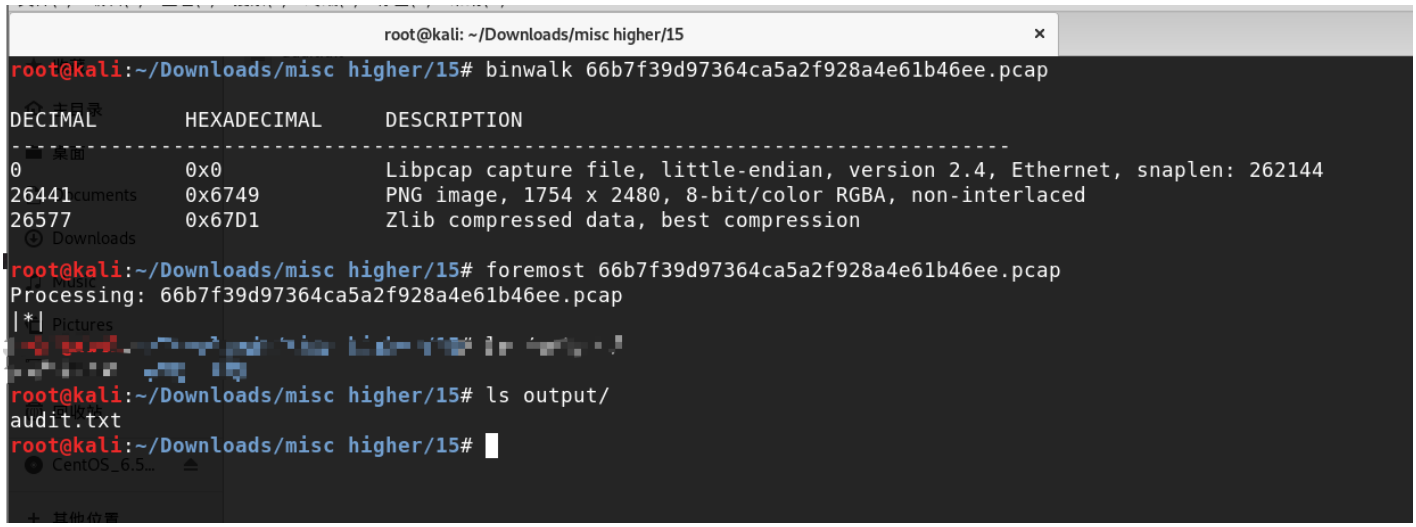
下载后是pdf文件，用binwalk分析，有点东西，用foremost分解东西，flag藏在分解的图片里

```
root@kali:~/Downloads/misc higher/14# binwalk 7e5ab2e7587d4a4abf9c705dfb935a92.pdf

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PDF document, version: "1.4"
452           0x1C4           JPEG image data, JFIF standard 1.01
73254         0x11E26         JPEG image data, JFIF standard 1.01
81606         0x13EC6         Zlib compressed data, default compression
82150         0x140E6         JPEG image data, JFIF standard 1.01
104469        0x19815         Zlib compressed data, default compression
105134        0x19AAE         Zlib compressed data, default compression

root@kali:~/Downloads/misc higher/14# foremost 7e5ab2e7587d4a4abf9c705dfb935a92.pdf
Processing: 7e5ab2e7587d4a4abf9c705dfb935a92.pdf
|*|
root@kali:~/Downloads/misc higher/14#
```



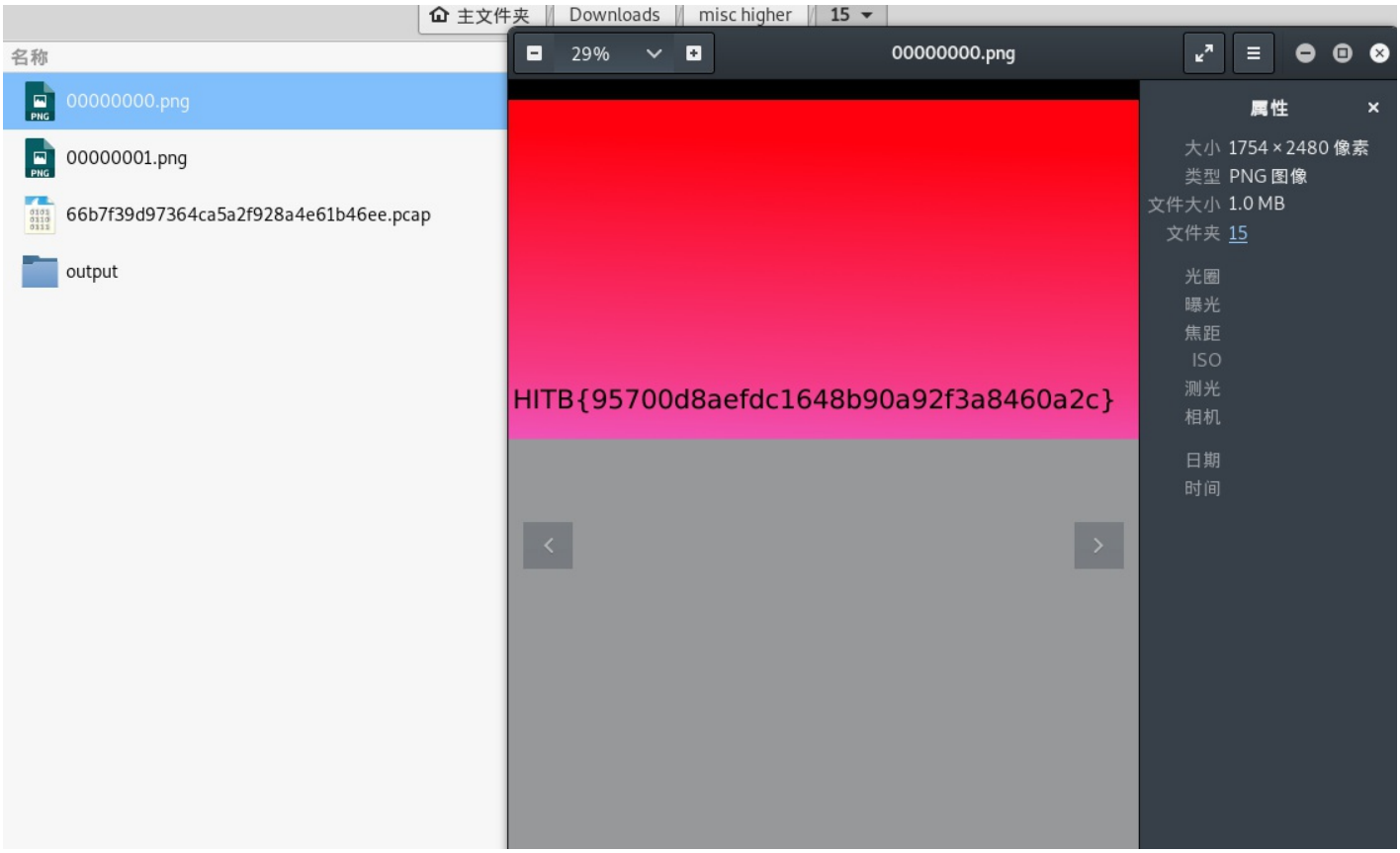
15.Cephalopod

下载后是一个流量包，用wireshark打开，flag关键字找到flag.png，但是却弄不出来图片。用binwalk看一下有点东西，foremost搞不出来图片



了解到有tcpxtract这个工具，Tcpxtract是一种基于文件签名从网络流量中提取文件的工具。安装tcpxtract，并使用分离图片出来，flag值出来了

```
root@kali:~/Downloads/misc higher/15# apt-get install tcpxtract
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列【新】软件包将被安装：
  tcpxtract
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 2120 个软件包未被升级。
需要下载 17.1 kB 的归档。
解压缩后会消耗 56.3 kB 的额外空间。
获取:1 http://mirrors.aliyun.com/kali kali-rolling/main amd64 tcpxtract amd64 1.0.1-13 [17.1 kB]
已下载 17.1 kB，耗时 1秒 (28.0 kB/s)
正在选中未选择的软件包 tcpxtract。
(正在读取数据库 ... 系统当前共安装有 370625 个文件和目录。)
正准备解包 .../tcpxtract_1.0.1-13_amd64.deb  ...
正在解包 tcpxtract (1.0.1-13) ...
正在设置 tcpxtract (1.0.1-13) ...
正在处理用于 man-db (2.8.4-2+b1) 的触发器 ...
root@kali:~/Downloads/misc higher/15# ls
66b7f39d97364ca5a2f928a4e61b46ee.pcap  output
root@kali:~/Downloads/misc higher/15# tcpxtract -f 66b7f39d97364ca5a2f928a4e61b46ee.pcap Found file of type "png" in session
Found file of type "png" in session [10.0.2.7:49818 -> 10.0.2.10:36890], exporting to 00000000.png
Found file of type "png" in session [10.0.2.7:49818 -> 10.0.2.10:36890], exporting to 00000001.png
```



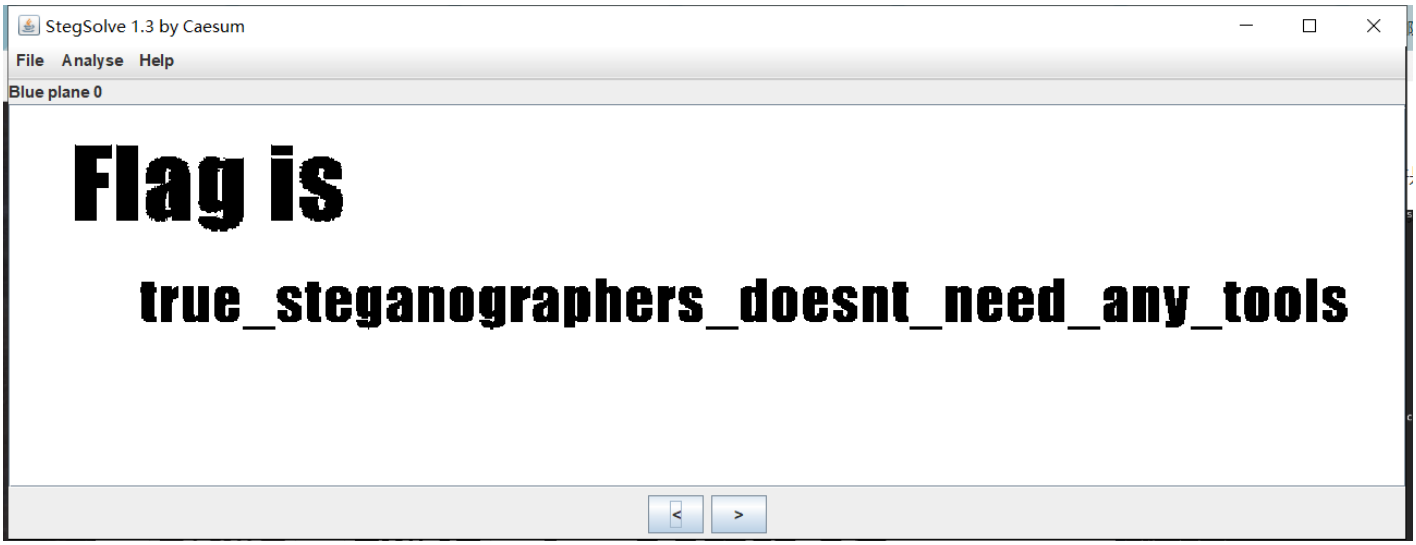
16.hit-the-core

下载后是.core文件，.core文件是Linux的文件，用strings命令查看，看到一段特殊的文段

```
root@kali:~/Downloads/misc higher/16# strings e67ad877258d4454af464202b9ee46c4.core
CORE
code
./code
CORE
CORE
////
////
////
LINUX
////
////
////
IGISCORE
CORE
ELIFCORE
/home/oddcoder/projects/ctf/forensics/FORE1/code
/usr/lib64/libc-2.24.so
/usr/lib64/libc-2.24.so
/usr/lib64/ld-2.24.so
/usr/lib64/ld-2.24.so
/lib64/ld-linux-x86-64.so.2
:Wzc
__isoc99_scanf
GLIBC_2.7
libc.so.6
__libc_start_main
GLIBC_2.2.5
strlen
putchar
_Jv_RegisterClasses
_ITM_registerTMCloneTable
_ITM_deregisterTMCloneTable
__gmon_start__
UH-
ffffff.
AWAVA
AUATL
[]A\A]A^A_
cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrKwgk83kgd43j85ePgb_e_rwqr7fvbmHjklo3tews_hmkogooyf0vbnk0ii87Drfgh_n kiwutfb0ghk9ro987k5tfb_hjiouo087ptfcv}
;*3$"
```

17.pure_color

下载后是一张图片，用stegsolve打开调试几下，即可看到flag

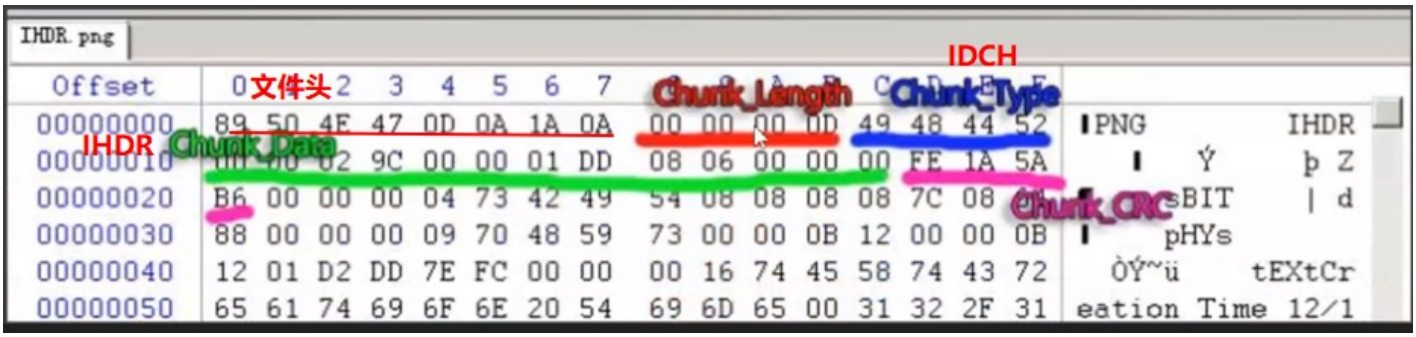


18.2-1

下载后是一张png图片，打开错误，用winhex打开，发现png文件头错误进行修正后发现还是打不开。

```
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 00 00 00 00 02 F8 08 06 00 00 00 93 2F 8A        ø     "/Š
00000020 6B 00 00 00 04 67 41 4D 41 00 00 9C 40 20 0D E4 k    gAMA  œ@  ä
00000030 CB 00 00 00 20 63 48 52 4D 00 00 87 0F 00 00 8C Ë    cHRM  ‡   Œ
00000040 0F 00 00 FD 52 00 00 81 40 00 00 7D 79 00 00 E9    ýR   @  }y  é
00000050 8B 00 00 3C E5 00 00 19 CC 73 3C 85 77 00 00 0A ‹  <å   Ìs<…w
00000060 39 69 43 43 50 50 68 6F 74 6F 73 68 6F 70 20 49 9iCCPPhotoshop I
00000070 43 43 20 70 72 6F 66 69 6C 65 00 00 48 C7 9D 96 CC profile  HÇ –
00000080 77 54 54 D7 16 87 CF BD 77 7A A1 CD 30 D2 19 7A wTT× ‡Ï½wz¡Í0Ò z
00000090 93 2E 30 80 F4 2E 20 1D 04 51 18 66 06 18 CA 00 ".0€ô.   Q f  Ê
000000A0 C3 0C 4D 6C 88 A8 40 44 11 11 01 45 90 A0 80 01 Ã Ml^¨@D   E  €
000000B0 A3 A1 48 AC 88 62 21 28 A8 60 0F 48 10 50 62 30 £¡H¬^b!(¨` H Pb0
000000C0 8A A8 A8 64 46 D6 4A 7C 79 79 EF E5 E5 F7 C7 BD Š¨¨dFÖJ|yyïåå÷Ç½
000000D0 DF DA 67 EF 73 F7 D9 7B 9F B5 2E 00 24 4F 1F 2E ßÚgïs÷Ù{Ÿµ. $O .
000000E0 2F 05 96 02 20 99 27 E0 07 7A 38 D3 57 85 47 D0 / –  ™'à z8ÓW…GÐ
000000F0 B1 FD 00 06 78 80 01 A6 00 30 59 E9 A9 BE 41 EE ±ý  x€ ¦ 0Yé©¾Aî
00000100 C1 40 24 2F 37 17 7A BA C8 09 FC 8B DE 0C 01 48 Á@$/7 z°È ü‹Þ  H
00000110 FC BE 65 E8 E9 4F A7 83 FF 4F D2 AC 54 BE 00 00 ü¾eèéO§ƒÿOÒ¬T¾
00000120 C8 5F C4 E6 6C 4E 3A 4B C4 F9 22 4E CA 14 A4 8A È_ÄælN:KÄù"NÊ ¤Š
00000130 ED 33 22 A6 C6 24 8A 19 46 89 99 2F 4A 50 C4 72 í3"¦Æ$Š F‰™/JPÄr
00000140 62 8E 5B E4 A5 9F 7D 16 D9 51 CC EC 64 1E 5B C4 bŽ[ä¥Ÿ} ÙQÌìd [Ä
00000150 E2 9C 53 D9 C9 6C 31 F7 88 78 7B 86 90 23 62 C4 âœSÙÉl1÷^x{† #bÄ
00000160 47 C4 05 19 5C 4E A6 88 6F 8B 58 33 49 98 CC 15 GÄ  \N¦^o‹X3I˜Ì
00000170 F1 5B 71 6C 32 87 99 0E 00 8A 24 B6 0B 38 AC 78 ñ[ql2‡™  Š$¶ 8¬x
00000180 11 9B 88 98 C4 0F 0E 74 11 F1 72 00 70 A4 B8 2F  ›^˜Ä  t ñr p¤¸/
00000190 38 E6 0B 16 70 B2 04 E2 43 B9 A4 A4 66 F3 B9 71 8æ  p² âC¹¤¤fó¹q
```

)这里涉及了png文件格式



（固定）八个字节89 50 4E 47 0D 0A 1A 0A为png的文件头

（固定）四个字节00 00 00 0D（即为十进制的13）代表数据块的长度为13

（固定）四个字节49 48 44 52（即为ASCII码的IHDR）是文件头数据块的标示（IDCH）

（可变）13位数据块（IHDR)

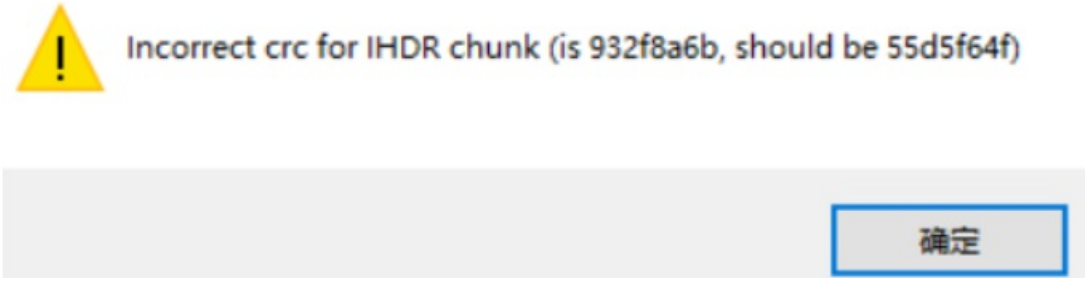
前四个字节代表该图片的宽

后四个字节代表该图片的高

后五个字节依次为： Bit depth、ColorType、Compression method、Filter method、Interlace method

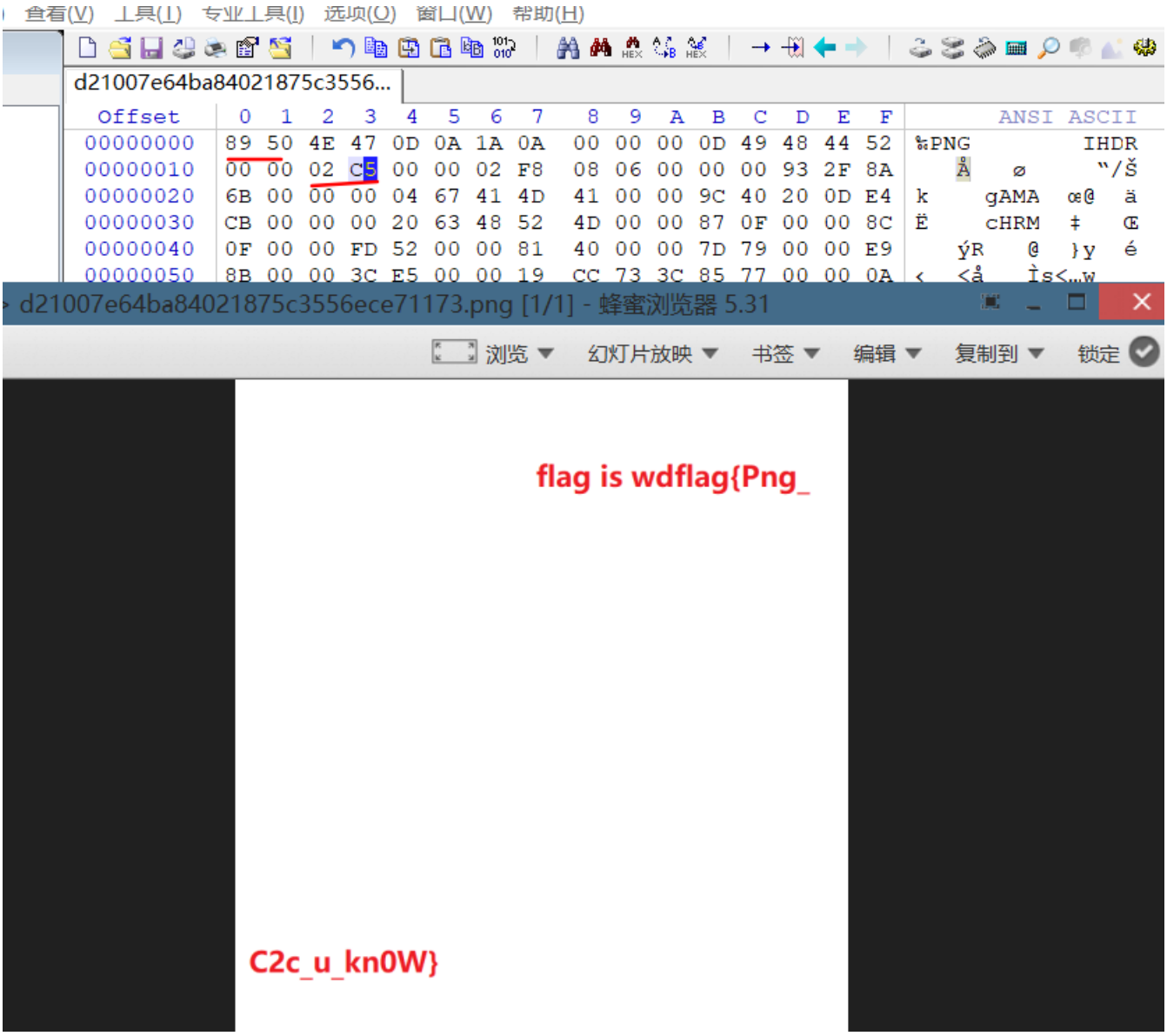
（可变）剩余四字节为该png的CRC检验码，由从IDCH到IHDR的十七位字节进行crc计算得到。

使用tweakpng这个工具计算crc校验码



使用脚本跑出正确的宽709，在windex修改成十六进制为02c5。

```
import struct
import binascii
import os

m = open("misc4.png","rb").read()
for i in range(1024):
    c = m[12:16] + struct.pack('>i', i) + m[20:29]
    crc = binascii.crc32(c) & 0xffffffff
    if crc == 0x932f8a6b:
        print(i)
```



19.János-the-Ripper

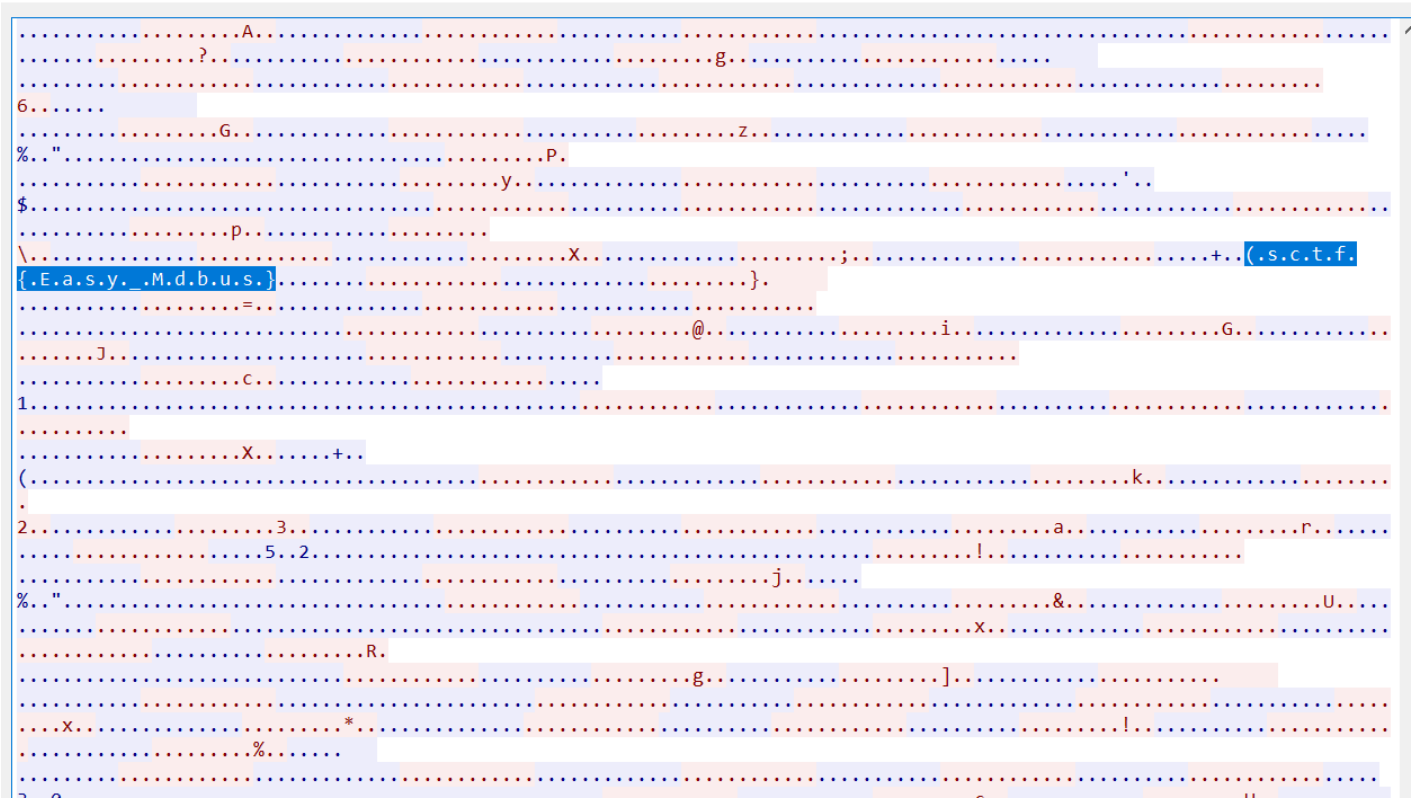
下载压缩吧解压后不知道是什么文件，用file命令查看下是zip文件，改文件后缀名zip，解压需要密码，进行密码爆破为fish，flag为flag{ev3n::y0u::bru7us?!}

20.2017_Dating_in_Singapore

题目为新加坡2017日历，附件解压打开时一张2017新加坡的日历，根据给出的数字进行连线，可得flag

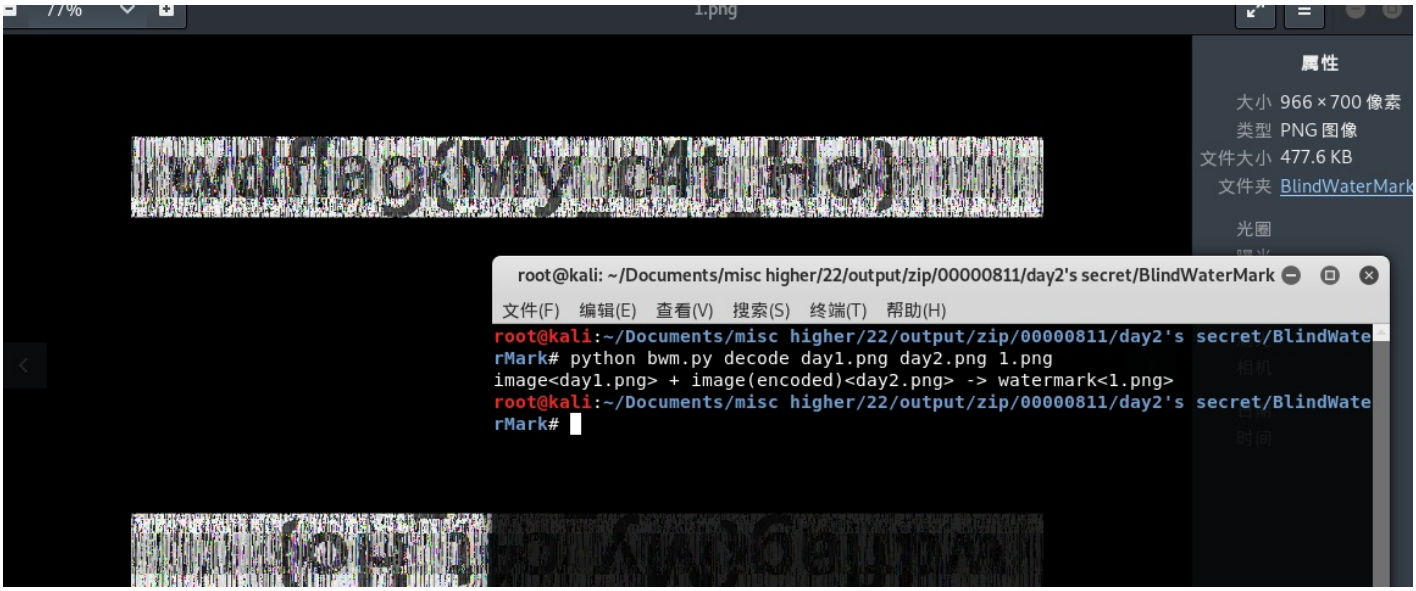
21神奇的Modbus

下载下来是一个流量包，用wireshark打开，搜索关键字flag，追踪数据流，这里搜索关键字没用，全部查看下来发现这里有flag，flag为**sctf{Easy_Modbus}**，要多加一个o



22.4-1

下载下来是一张图片，用binwalk查看一下，有点东西，foremost分离出来两张图片，用盲水印攻击，加密脚本：https://github.com/chishaxie/BlindWaterMark



23.can_has_stdio?

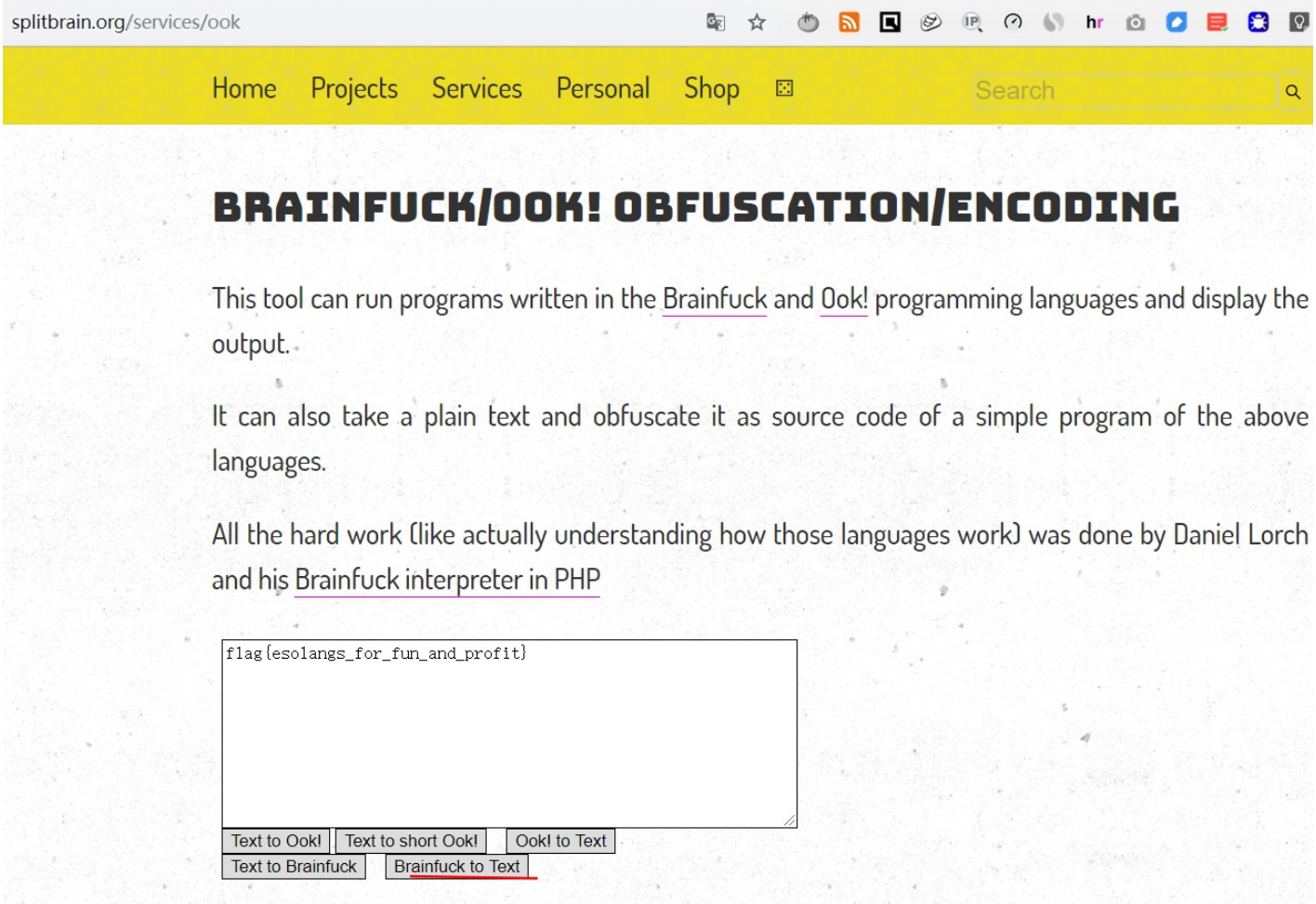
下载解压包解压得文件用记事本打开，看到得<,>,+,-,.,[,]等符号组成得五角星，可以猜测是Brainfuck语言了，

下面是这八种状态的描述，其中每个状态由一个字符标识：

字符	含义
>	指针加一
<	指针减一
+	指针指向的字节的值加一
-	指针指向的字节的值减一
.	输出指针指向的单元内容（ASCⅡ码）
,	输入内容到指针指向的单元（ASCⅡ码）
[	如果指针指向的单元值为零，向后跳转到对应的]指令的次一指令处
]	如果指针指向的单元值不为零，向前跳转到对应的[指令的次一指令处

（按照更节省时间的简单说法，"]"也可以说成"向后跳转到对应的"["状态"。这两解释是一样的。）

到https://www.splitbrain.org/services/ook这个网站翻译Brainfuck为文本



24.5-1

考察xortool工具的使用，安装python库`pip2 install xortool`,猜测最有可能得密钥长度为13，

```
xortool cd2a80e1b48e44b5a830605e684ffb31  -l 13 -c 20 //-l 指定密钥长度 -表示出现频率最高的字符。
```

```
root@kali:~/Downloads/misc higher/24# xortool cd2a80e1b48e44b5a830605e684ffb31
The most probable key lengths:
   2:   12.2%
   5:   11.9%
   9:   9.8%
  13:   22.2%
  20:   6.8%
  22:   6.2%
  26:   12.8%
  30:   4.6%
  39:   7.8%
  52:   5.7%
Key-length can be 3*n
Most possible char is needed to guess the key!
root@kali:~/Downloads/misc higher/24# xortool cd2a80e1b48e44b5a830605e684ffb31  -l 13 -c 20
1 possible key(s) of length 13:
Good\tuckToYou
Found 1 plaintexts with 95.0%+ valid characters
See files filename-key.csv, filename-char_used-perc_valid.csv
root@kali:~/Downloads/misc higher/24# ls
```

使用脚本解密出原文，flag为wdflag{You Are Very Smart}

```
import os
c = open("cd2a80e1b48e44b5a830605e684ffb31",'rb').read()
key = "GoodLuckToYou"
def xor(c,k):
    keylen = len(k)
    res = ""
    for pos,c in enumerate(c):
        res +=chr(ord(c) ^ ord(k[pos % keylen]))
    return res
print xor(c,key)
```

```
root@kali:~/Downloads/misc higher/24# python test.py
The opening line of the novel famously announces: "It is a truth universally acknowledged, that a single man in possession of a good fortune must be in want of a wife." This sets marr
entral subject—and really, a central problem—for the novel generally. Readers are poised to question whether or not these single men are, in fact, in want of a wife, or if such desire
ted by the "neighbourhood" families and their daughters who require a "good fortune". Marriage is a complex social activity that takes political economy, and economy more generally, i
. In the case of Charlotte Lucas, for example, the seeming success of her marriage lies in the comfortable economy of their household, while the relationship between Mr and Mrs Bennet
illustrate bad marriages based on an initial attraction and surface over substance (economic and psychological). The Bennets' marriage is one such example that the youngest Bennet, Ly
ome to re-enact with Wickham, and the results are far from felicitous. wdflag{You Are Very Smart}Though the central characters, Elizabeth and Darcy, begin the novel as hostile acquain
unlikely friends, they eventually work to understand each other and themselves so that they can marry each other on compatible terms personally, even if their "equal" social status re
nt. When Elizabeth rejects Darcy's first proposal, the argument of only marrying when one is in love is introduced. Elizabeth only accepts Darcy's proposal when she is certain she lov
her feelings are reciprocated. Austen's complex sketching of different marriages ultimately allows readers to question what forms of alliance are desirable, especially when it comes t
ng economic, sexual, companionate attraction.
```

25.MISCal

下载解压后不知道是什么文件，用file命令查看一下，是bzip2文件，修改文件后缀名为.bz2,解压文件

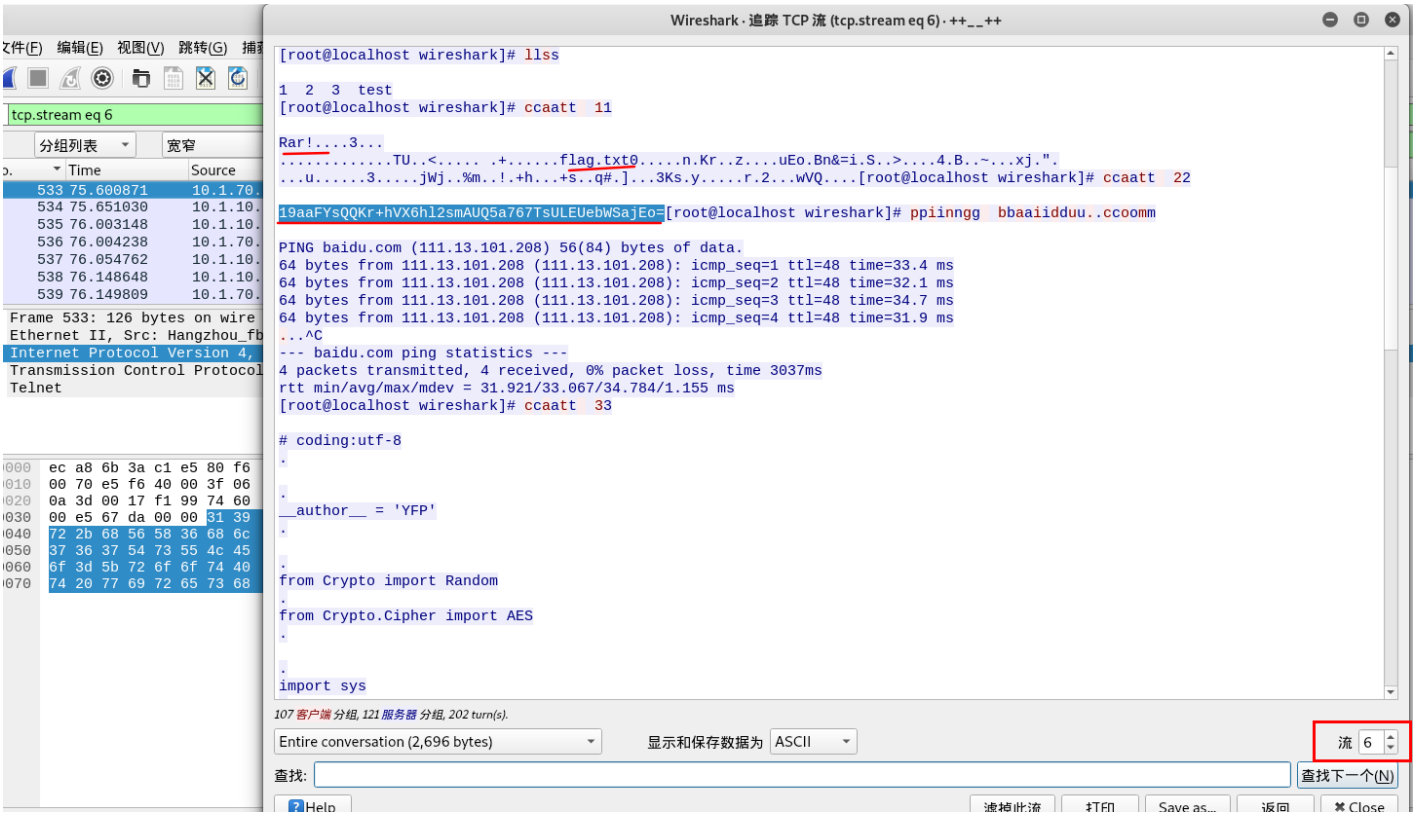
```
root@kali:~/Downloads/misc higher/25# file e4a1278fef074ffd89b5bd9b789527b5
e4a1278fef074ffd89b5bd9b789527b5: bzip2 compressed data, block size = 900k
root@kali:~/Downloads/misc higher/25# tar -xjvf e4a1278fef074ffd89b5bd9b789527b5.bz2
ctf/
ctf/flag.txt
ctf/.git/
ctf/.git/description
ctf/.git/refs/
ctf/.git/refs/heads/
ctf/.git/refs/heads/master
ctf/.git/refs/stash
ctf/.git/refs/tags/
ctf/.git/ORIG_HEAD
ctf/.git/logs/
ctf/.git/logs/refs/
```

发现是git目录，里面有个flag.txt文件不过flag值.`git log`查看git日志，`git stash list`列出所有保存的进度列表，`git stash apply`恢复暂缓区的内容，有s.py文件，运行一下可以得到flag

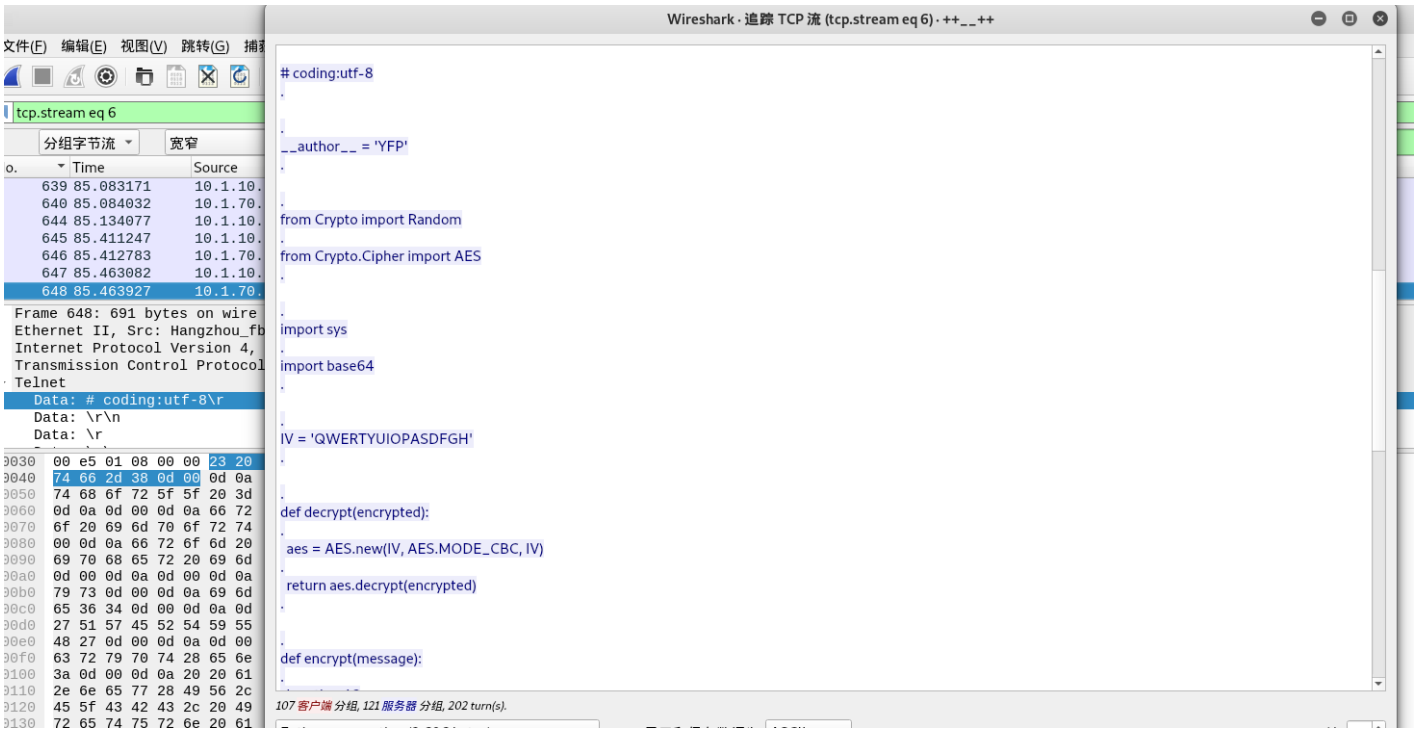
```
root@kali:~/Downloads/misc higher/25# cd ctf/
root@kali:~/Downloads/misc higher/25/ctf# git log
commit bea99b953bef6cc2f98ab59b10822bc42afe5abc (HEAD -> master)
Author: Linus Torvalds <torvalds@klaava.Helsinki.Fi>
Date:   Thu Jul 24 21:16:59 2014 +0200

    Initial commit
root@kali:~/Downloads/misc higher/25/ctf# git stash list
stash@{0}: WIP on master: bea99b9 Initial commit
root@kali:~/Downloads/misc higher/25/ctf# git stash show
 flag.txt | 25 +++++++++++++++++++++++++
 s.py     |  4 ++++
 2 files changed, 28 insertions(+), 1 deletion(-)
root@kali:~/Downloads/misc higher/25/ctf# git stash apply
位于分支 master
要提交的变更：
  （使用 "git reset HEAD <文件>..." 以取消暂存）

        新文件：    s.py

尚未暂存以备提交的变更：
  （使用 "git add <文件>..." 更新要提交的内容）
  （使用 "git checkout -- <文件>..." 丢弃工作区的改动）

        修改：      flag.txt

root@kali:~/Downloads/misc higher/25/ctf# python s.py
NCN4dd992213ae6b76f27d7340f0dde1222888df4d3
root@kali:~/Downloads/misc higher/25/ctf#
```

26.3-1

下载下来是rar压缩包，不过文件没有文件后缀名，binwalk查看得rar，添加.rar文件后缀名，提取压缩包出来时一个流量包，用wireshark打开，搜索flag关键字，追踪数据流，在第6个流可以看到flag.rar,base64编码



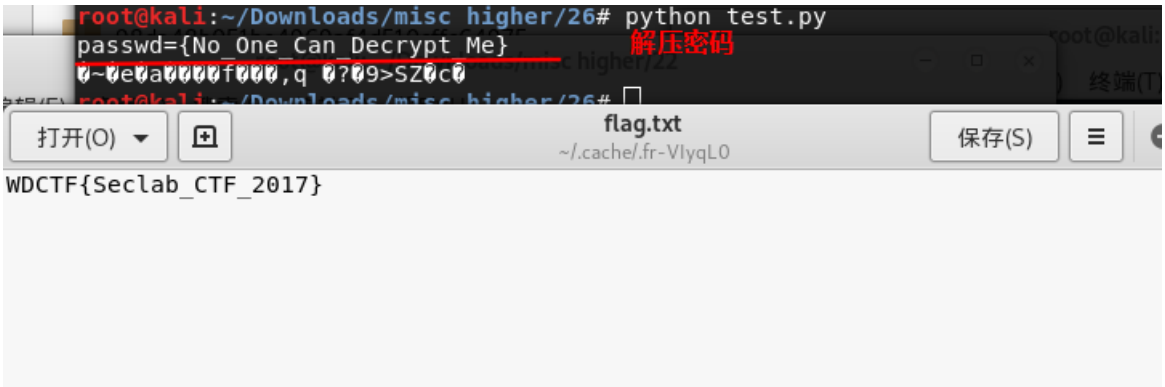
导出对象，选择http，flag.rar文件，解压需要密码，可以从数据流中可以看解密脚本多半时压缩包密码



提取出来整合修改下

```
# coding:utf-8
from Crypto import Random
from Crypto.Cipher import AES
import sys
import base64
IV = 'QWERTYUIOPASDFGH'
def decrypt(encrypted):

  aes = AES.new(IV, AES.MODE_CBC, IV)
  return aes.decrypt(encrypted)



def encrypt(message):

  length = 16

  count = len(message)

  padding = length - (count % length)

  message = message + '\0' * padding

  aes = AES.new(IV, AES.MODE_CBC, IV)

  return aes.encrypt(message)
str = '19aaFYsQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo='
example = decrypt(base64.b64decode(str))
print example
print decrypt(example)
```

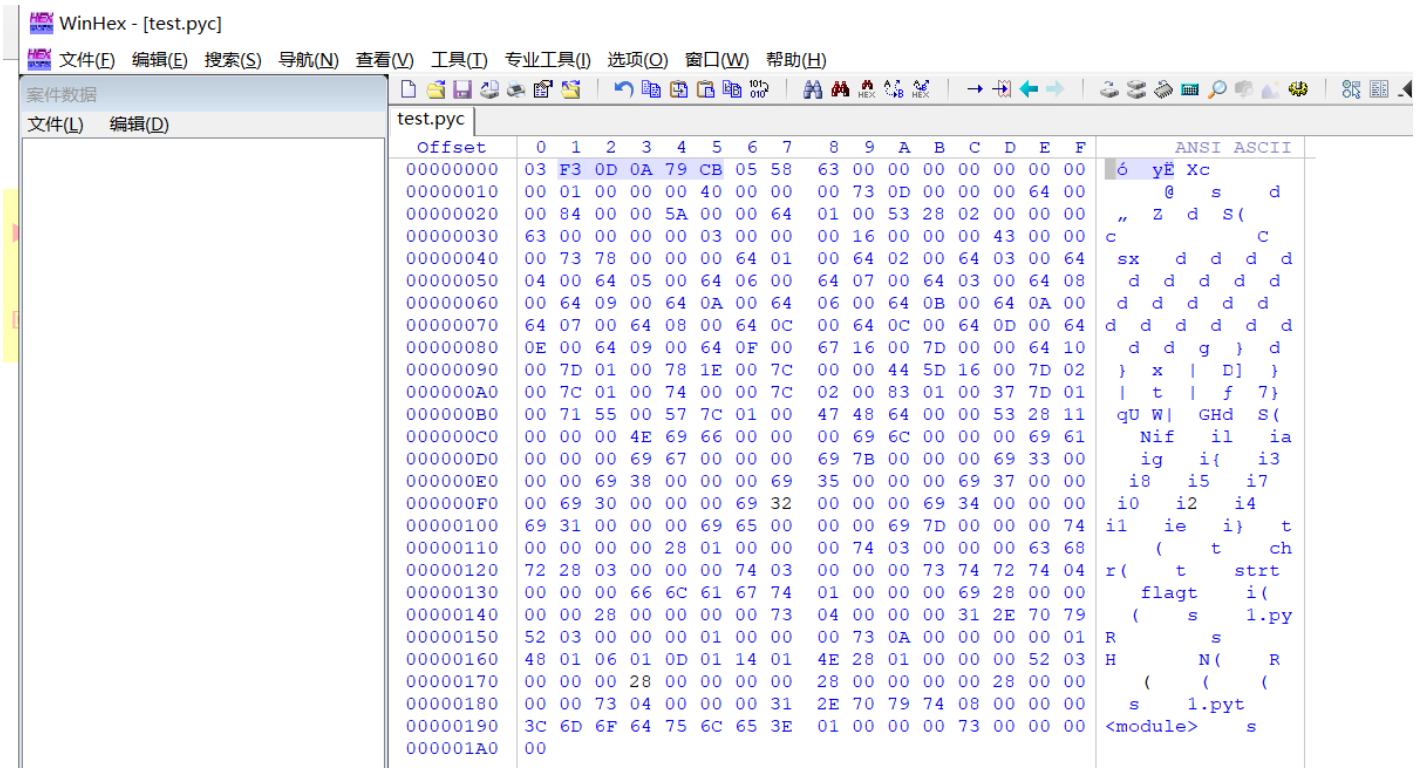
脚本运行一下，打开flag.txt即可看到flag



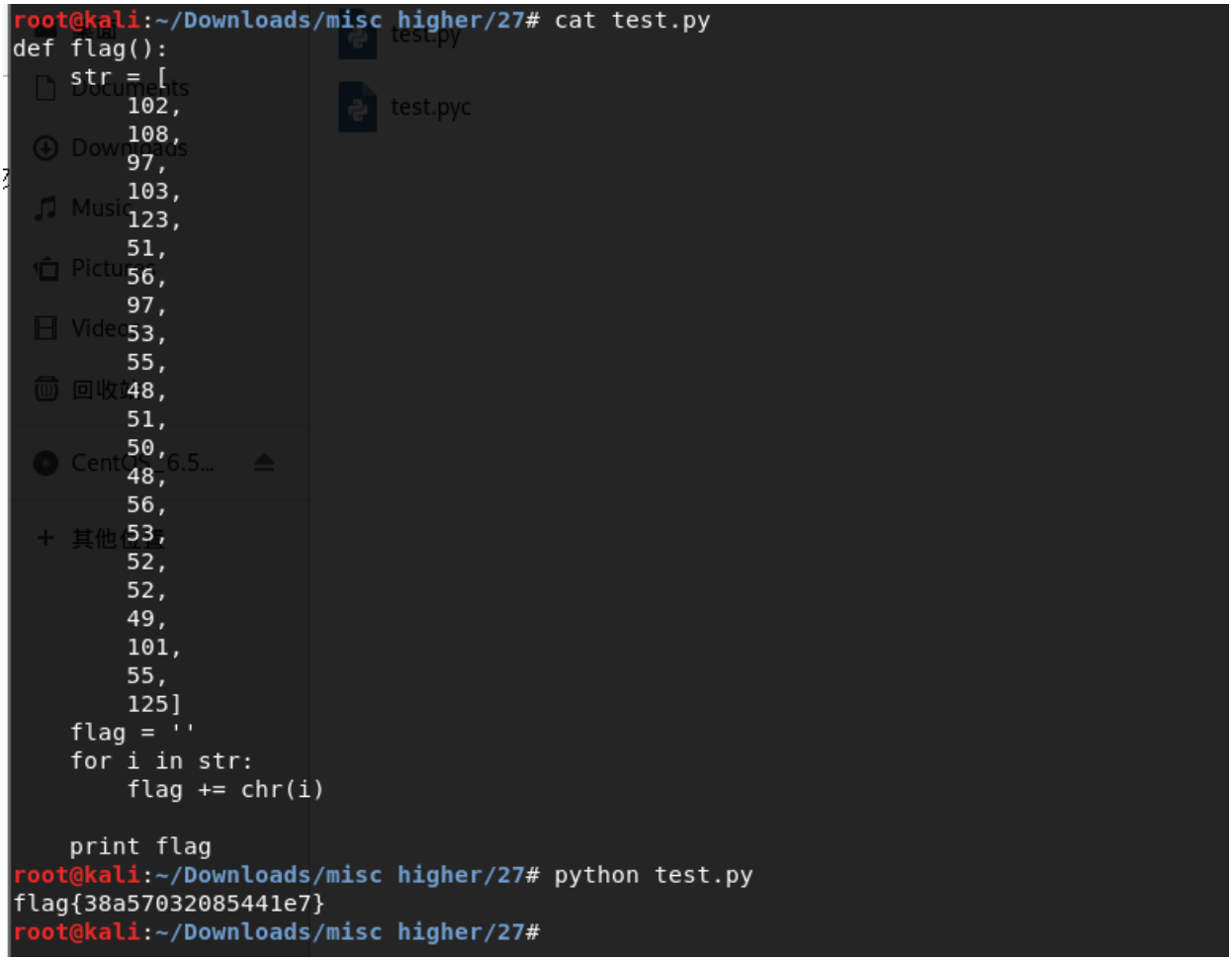
27.适合作为桌面

下载压缩包解压是一张图片，放stegsolve弄一下，可以看到一个二维码，扫一下可以看到一串数字

这是一串十六进制的数字，在winhex里创建一份文件，把这些数字写进去，保存为.pyc

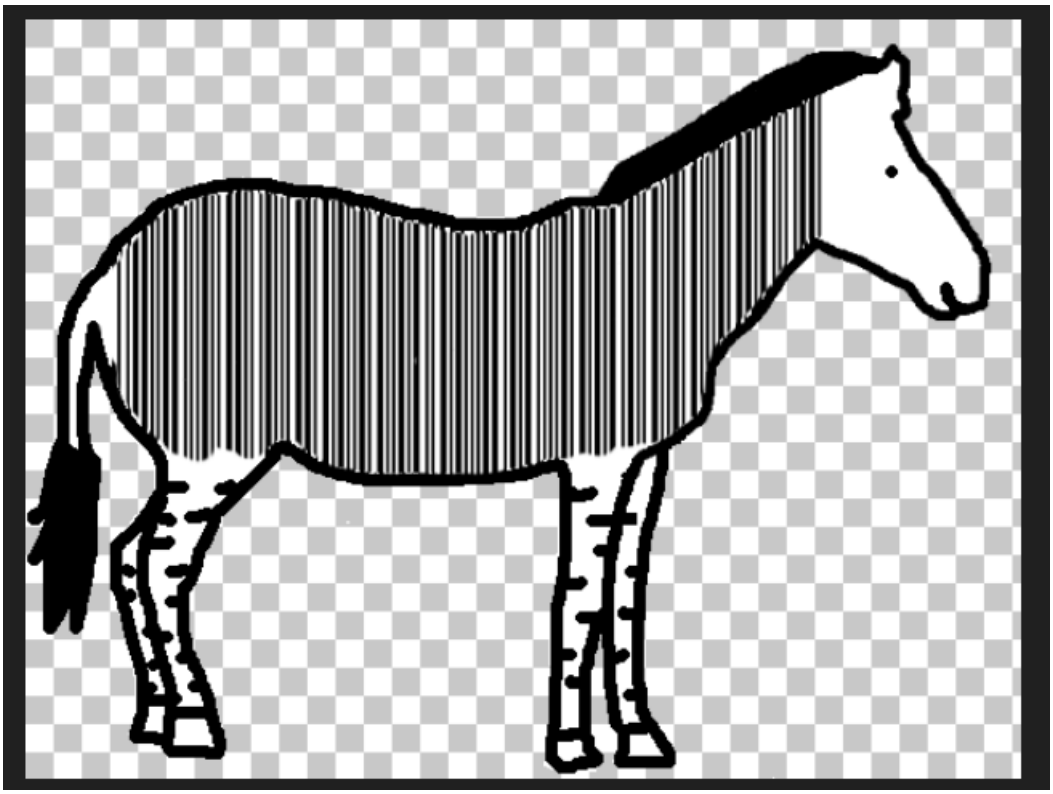


到https://tool.lu/pyc/这个网站解密pyc文件，解密出来python脚本跑一下即为flag

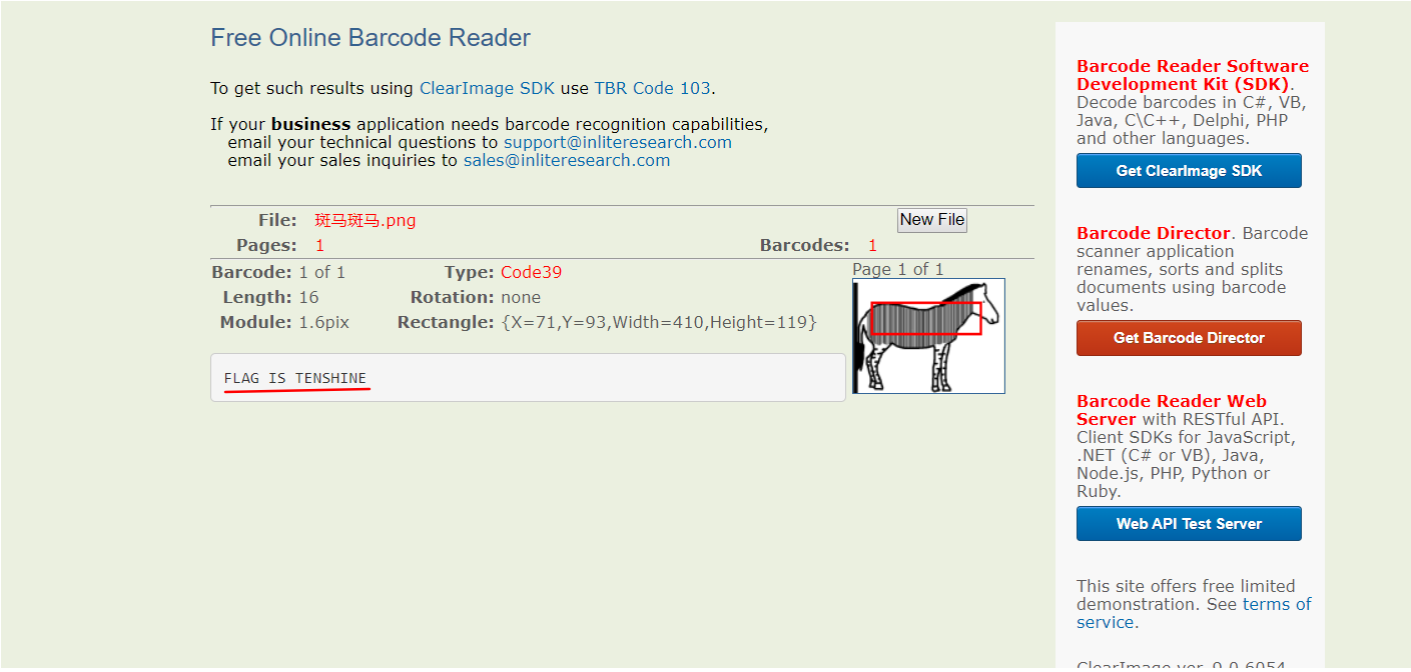


28.Banmabanma

下载下来是一张斑马的图片，猜测是条形码

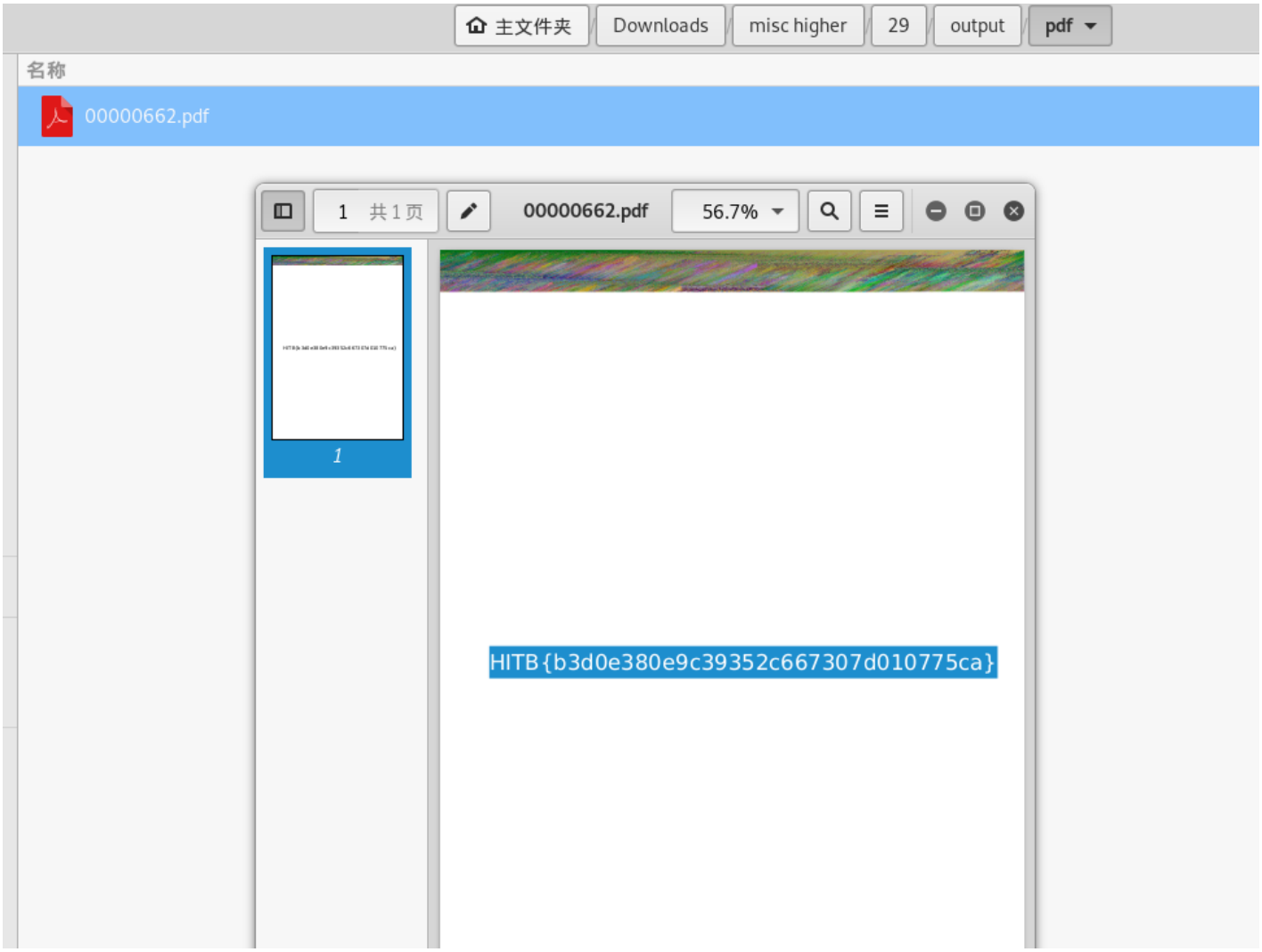


到https://online-barcode-reader.inliteresearch.com/解码一下，都不用ps一下了，nb



29.simple_transfer

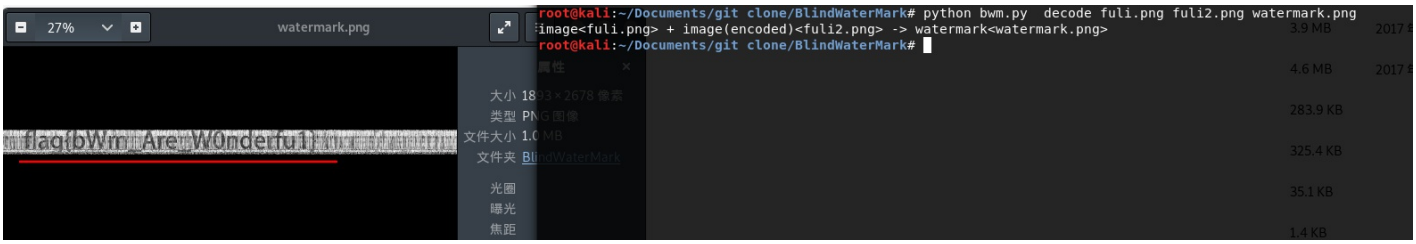
下载下来是个流量包，binwalk有点东西，foremost分离出来个pdf，打开pdf即可看到flag



30.warmup

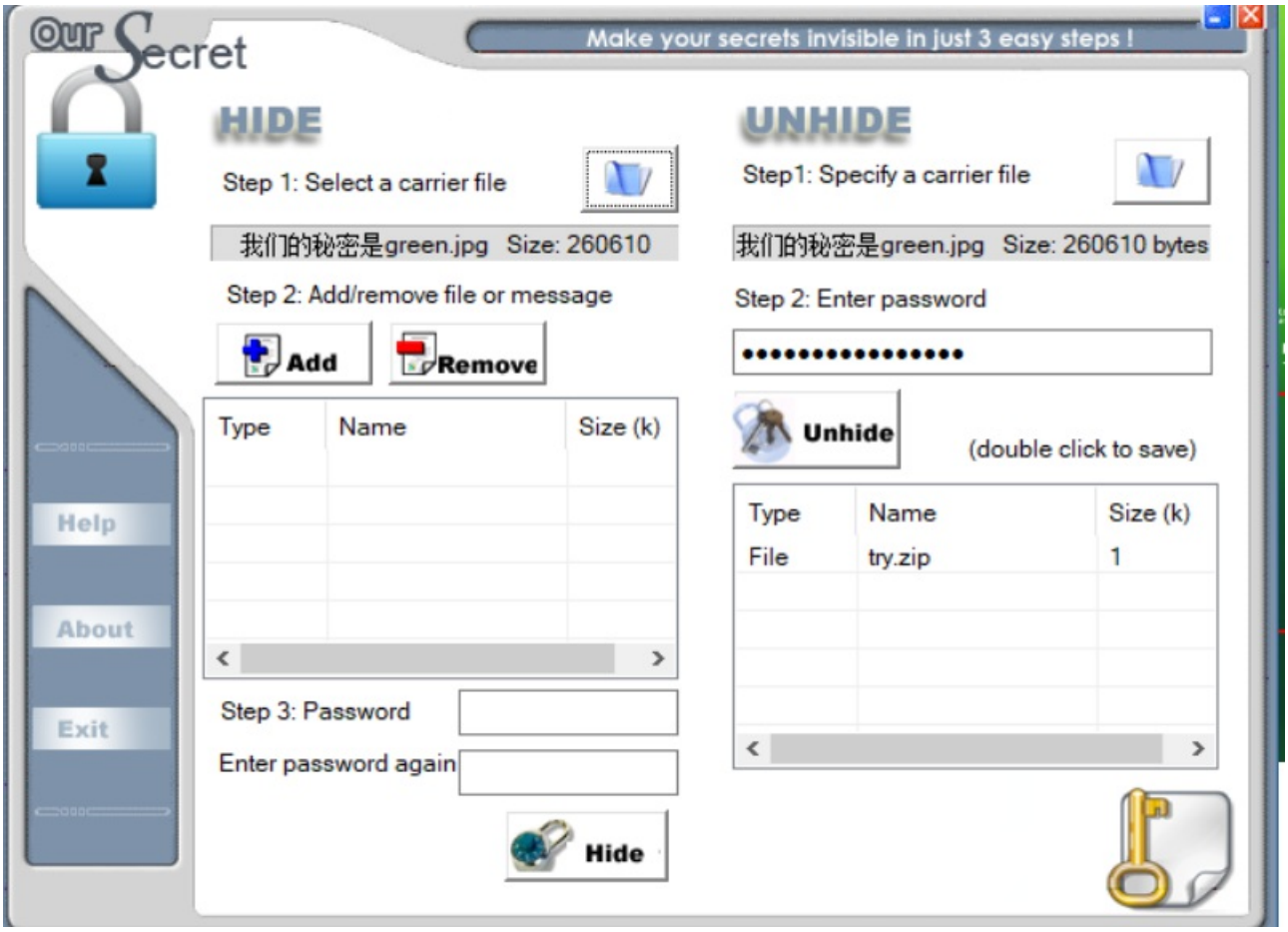
下载下来是一张图片和一个加密的zip包，加密的zip也有一个open_forum.png的文件，把open_forum.png压缩用ARCHPR明文进行攻击，破解出加密的zip包，这里要注意压缩软件要使用winrar

解压出来的是两张一样的图，猜测是盲水印，这里使用https://github.com/chishaxie/BlindWaterMark解盲水印的脚本跑出水印的图片即为flag



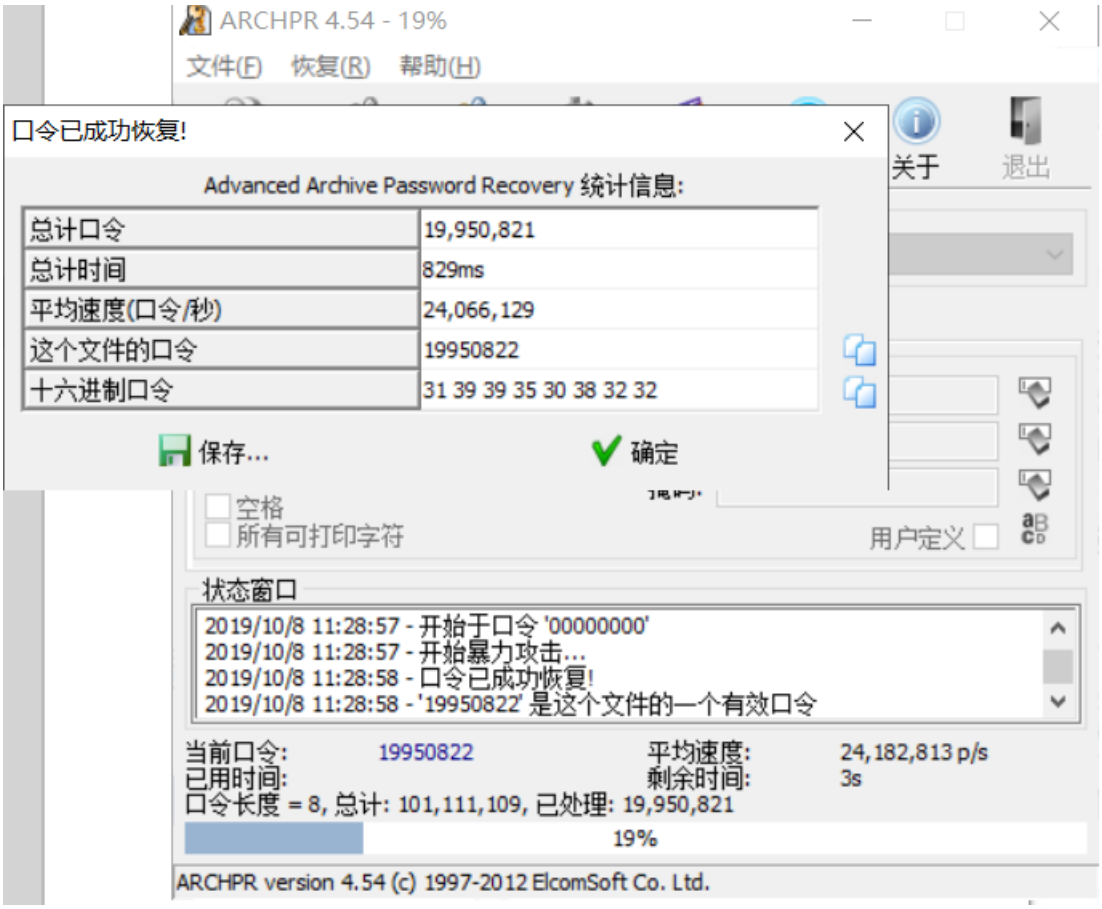
31.我们的秘密是绿色的

下载下来是张日历图片，这里根据题目名字提示，我们要用到Our secret文件隐藏加密软件，密码即是日历图中绿色数字0405111218192526，

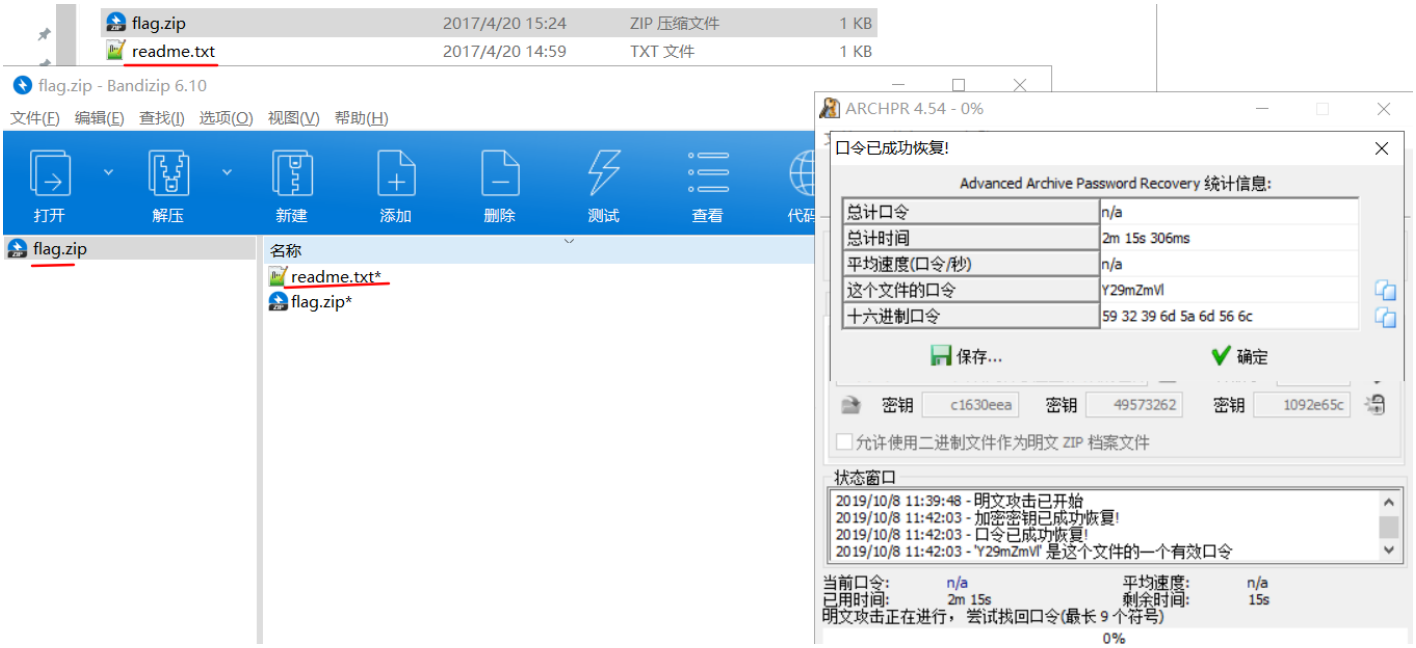


可以得到try压缩包，用记事本打开可以看到有生日提示，那就用生日当作密码进行暴力破解，密码最长8位

```
ETX EOT
I SOH BS NUL NUL ACK {攜t3派\ STX NUL NUL P STX NUL NUL BS NUL NUL NUL flag.zip EOT 鮀q搓拶?竝2u錴Mq US o獻e塹逡鶸鰩   豕+?      pd%昨X?塝?□
?BS?P   ?J   耒峹肒僔驥B EM FS |WD8 VT 0缃N?嘆 NAK銹 SO DC1 ?Z綢紡广~Y%渦礪b峈 FS 痹2Q?□
S+DC1 DC3 GS F\m妍ga疣?欖?□(?3 STX a-   凭繵敪J綵*辭#B?N+`匪↓FS 0Hj^   ?,?_鐌r:\鏍?)y袤袑`V
捎BP?m   }扛滒?3扰.嗩?囂芅{!渟#躋攫 ETB N}F橤?层X?笏)溗 SI CAN $#搗O!
S蒙/7糔蕭I?h   ?韜泜   甬 ETX 萣骯|岈奸~s3?趙綬鐉P>20?>R閤 ETX 榑 袅粻 EM 堿饲藋碎V
.?NAK 腰<DC2   五魅?EOT E2^K\ DC3 M   €A馄`[hMk?媮 BS ?&黎 DC4 Q EM 塹?塕?ETX?餐馹Q恵 SYN 榘
Z槻鯽眐F繆?測5柵{,淄瀅吉迦9S仮Ju邧浹S   ??萨^I-牮eS挏?.
尚   ?倫叔 BS f   C駋咡e削嫢焰薹i?鏌Zlby貶 RS 傒S夫} STX DLE )拯D   落?埳 SYN 筵 筆?幹 CAN 2Z0跘官鎚g?e镠)R ETX
嵒C EM 楛8? SUB V:APK ETX EOT
I SOH BS NUL NUL qw攜慥Y簃 NUL NUL NUL a NUL NUL NUL
I NUL NUL readme.txt膊jK ENQ螟   埆牡 RS 鄿?□

D k   !?檍   GS /▯   DC4 R;嘲夾鶣喊? NAK 捎 FS ?Xt? US &E-C遑   ?苘翴`A6柟▯gf暖 VT   dde= ENQ 閏?Z7-lZ<aOP SOH STX ? NUL
I SOH BS NUL NUL ACK {攜t3派\ STX NUL NUL P STX NUL NUL BS NUL ? NUL NUL NUL NUL NUL NUL NUL   NUL NUL NUL NUL NUL NUL NUL flag.zip
I   NUL NUL NUL NUL NUL SOH NUL CAN NUL Z赽 SUB Ч? 鵊?ч? 鵊?ч SOH STX ? NUL
I SOH BS NUL NUL qw攜慥Y簃 NUL NUL NUL a NUL NUL NUL
I $ NUL NUL NUL NUL NUL NUL NUL   NUL NUL NUL ? NUL NUL readme.txt
I   NUL NUL NUL NUL NUL SOH NUL CAN NUL ?應 9 ? STX 亙挂 SOH o/ STX 亙挂 SOH PK ENQ ACK NUL NUL NUL NUL STX NUL STX NUL ¶ NUL NUL ETB ETX NUL NUL GS NUL 你知道coffee的生日是多少么~~~
```



有破解的密码解压还是一个带加密的zip包，不过zip也有一个readme.txt，明文攻击走起，得到密码Y29mZmVl



解压后又一个加密压缩包，爆破无解，用winhex打开看看，看到伪加密了，把01改成00，解压就不需要密码了

test.pyc | flag.zip

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
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													

解压后得到一个flag.txt文本，内容为qddpqwnpcplen%prqwn{zz*d@gq}，猜测栅栏密码，再凯撒密码

Crypto | Image | UnZip

填写所需检测的密码：(已输入字符数统计：30)

qddpqwnpcplen%prqwn_{_zz*d@gq}

结果：(字符数统计：234)

得到因数(排除1和字符串长度)：
2 3 5 6 10 15

第1栏：qdqnclnpqn{z*@qdpwppe%rw__zdg}
第2栏：qpnpnrn_*gdqpl%q_zdqdwcepw{z@}
第3栏：qwlr{ddneq_@dpnwzgpc%nzqqpp_*}
第4栏：qnnn*dp%_ddcp{@ppr_gqlqzqwewz}
第5栏：ql{de_dnzp%zqp*wrdnq@pwgcnqp_}
第6栏：qrdqdwpnq_w{n_pzczp*lde@ng%qp}

Crypto | Image | UnZip

填写所需检测的密码：（已输入字符数统计：30）

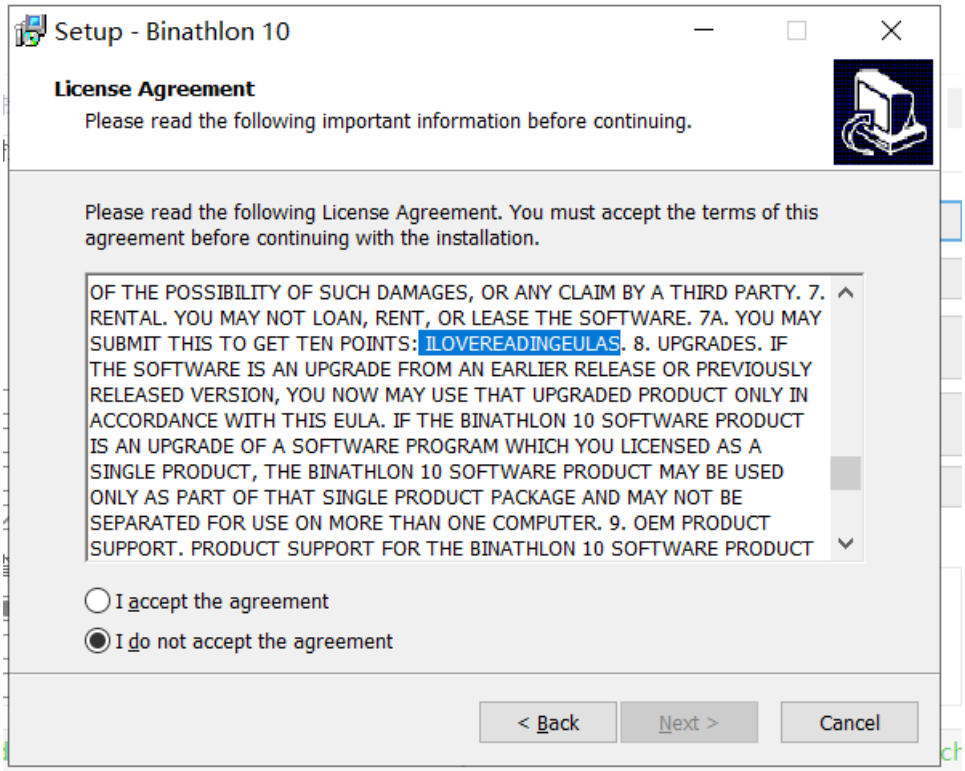
```
qwlr{ddneq_@dpnwzgpc%nzqqpp_*}
```

结果：（字符数统计：780）

```
rxms{eeofr_@eqoxahqd%oarrqq_*}
synt{ffpgs_@frpybire%pbssrr_*}
tzou{ggqht_@gsqzcjsf%qcttss_*}
uapv{hhriu_@htradktg%rduutt_*}
vbqw{iisjv_@iusbeluh%sevvuu_*}
wcrx{jjtkw_@jvtcfmvi%tfwwvv_*}
xdsy{kkulx_@kwudgnwj%ugxxww_*}
yetz{llvmy_@lxvehoxk%vhyyxx_*}
zfua{mmwnz_@mywfipyl%wizzyy_*}
agvb{nnxoa_@nzxgjqzm%xjaazz_*}
bhwc{ooypb_@oayhkran%ykbbaa_*}
cixd{ppzqc_@pbzilsbo%zlccbb_*}
djye{qqard_@qcajmtcp%amddcc_*}
ekzf{rrbse_@rdbknudq%bneedd_*}
flag{ssctf_@seclover%coffee_*}
gmbh{ttdug_@tfdmpwfs%dpggff_*}
hnci{uuevh_@ugenqxgt%eqhhgg_*}
iodj{vvfwi_@vhforyhu%friihh_*}
jpek{wwgxj_@wigpsziv%gsjjii_*}
kqfl{xxhyk_@xjhqtajw%htkkjj_*}
lrgm{yyizl_@ykirubkx%iullkk_*}
mshn{zzjam_@zljsvclv%jvmmll_*}
```

32.Just-No-One

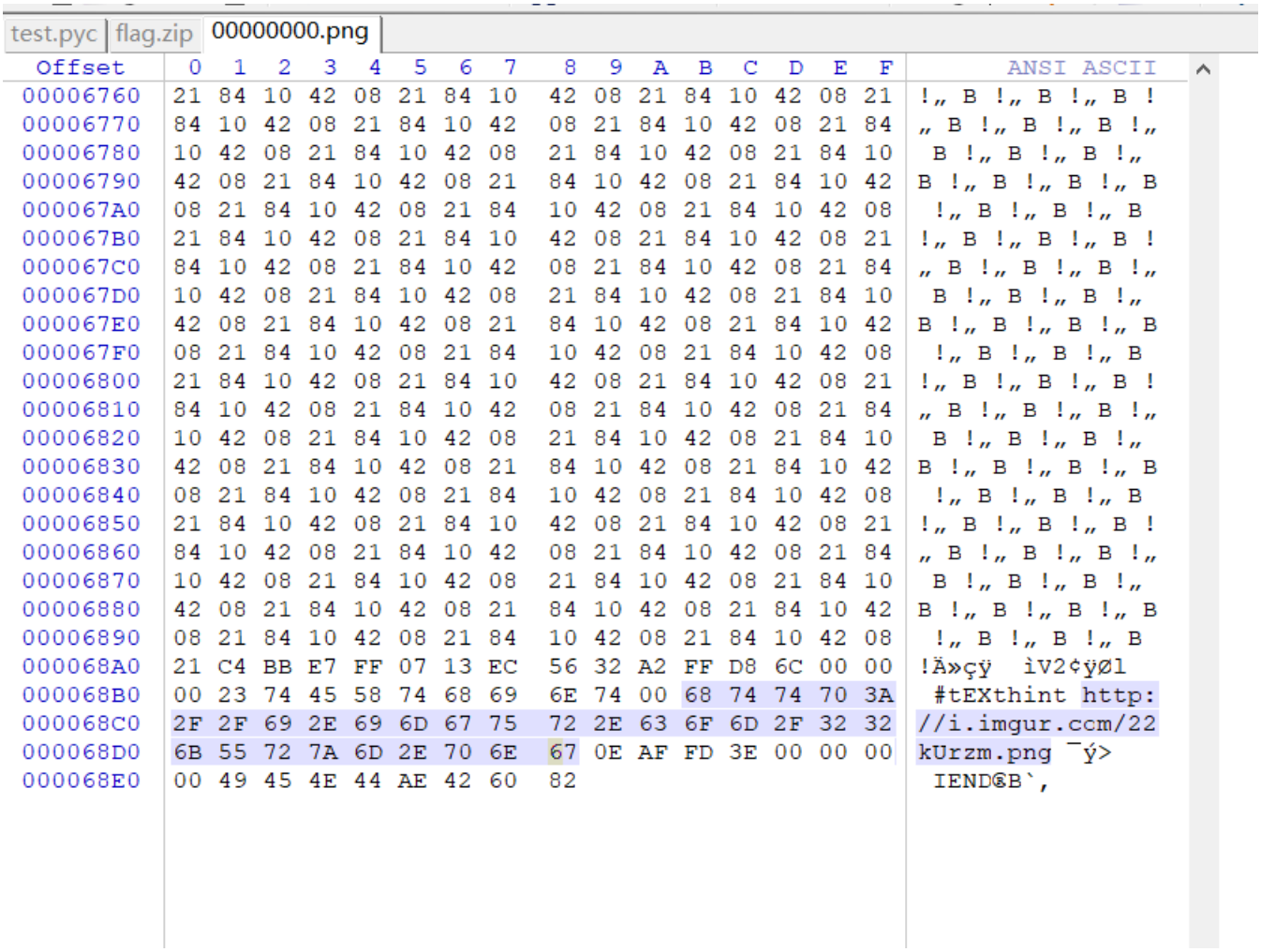
下载下来后是一个.exe程序，安装后提醒flag值再安装许可协议，所以找吧，flag即为ILOVEREADINGEULAS



33.Erik-Baleog-and-Olaf

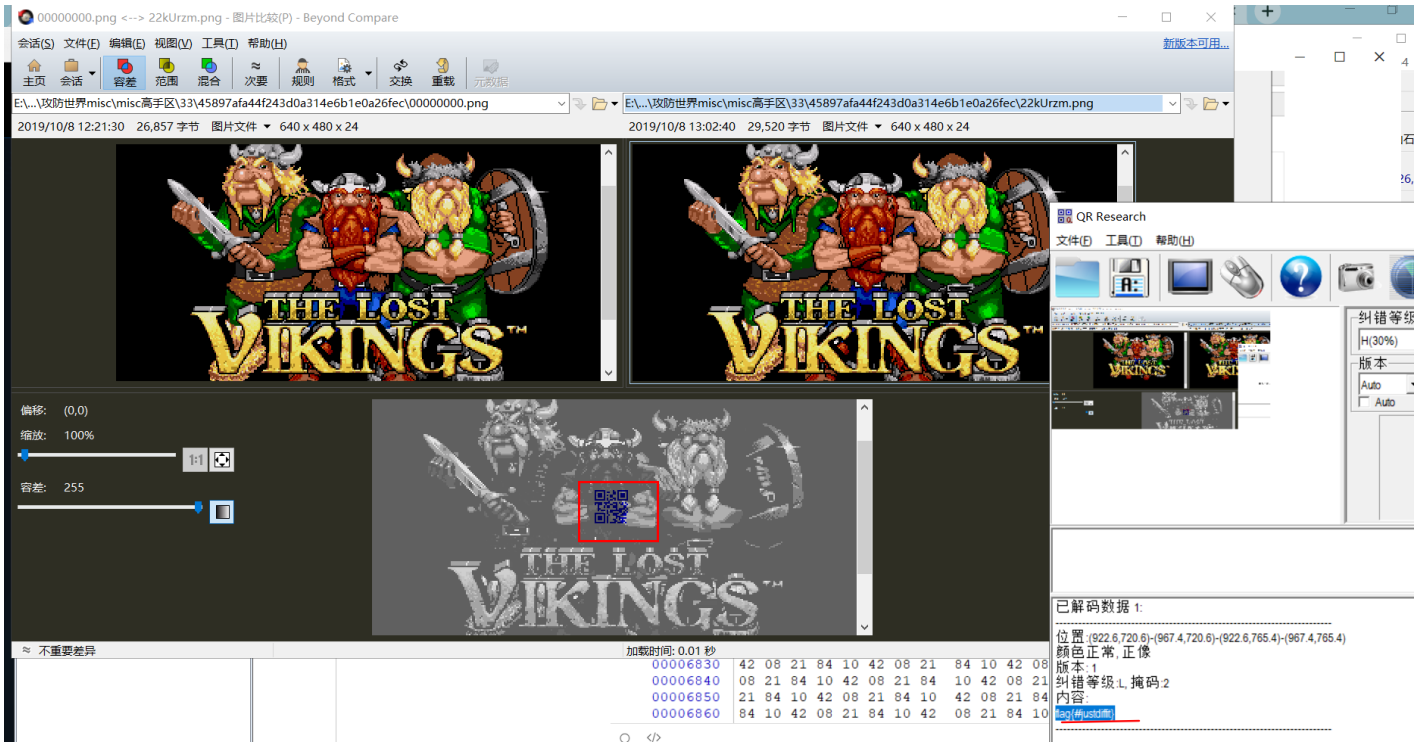
下载下来是一张图片，用winhex打开看看，可以看到一个图片网址，http://i.imgur.com/22kUrzm.png

到这个网址把图片下载下来，（需要fq）再对比两张图片，这里我用到的软件是beyond compare,可以看到二维码



test.pyc | flag.zip | 00000000.png

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00006760	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	!„ B !„ B !„ B !
00006770	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	„ B !„ B !„ B !„
00006780	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	B !„ B !„ B !„
00006790	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	B !„ B !„ B !„ B
000067A0	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	!„ B !„ B !„ B
000067B0	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	!„ B !„ B !„ B !
000067C0	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	„ B !„ B !„ B !„
000067D0	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	B !„ B !„ B !„
000067E0	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	B !„ B !„ B !„ B
000067F0	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	!„ B !„ B !„ B
00006800	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	!„ B !„ B !„ B !
00006810	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	„ B !„ B !„ B !„
00006820	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	B !„ B !„ B !„
00006830	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	B !„ B !„ B !„ B
00006840	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	!„ B !„ B !„ B
00006850	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	!„ B !„ B !„ B !
00006860	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	„ B !„ B !„ B !„
00006870	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	B !„ B !„ B !„
00006880	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	B !„ B !„ B !„ B
00006890	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	!„ B !„ B !„ B
000068A0	21	C4	BB	E7	FF	07	13	EC	56	32	A2	FF	D8	6C	00	00	!Ä»çÿ ìV2¢ÿØl
000068B0	00	23	74	45	58	74	68	69	6E	74	00	68	74	74	70	3A	#tEXthint http:
000068C0	2F	2F	69	2E	69	6D	67	75	72	2E	63	6F	6D	2F	32	32	//i.imgur.ccm/22
000068D0	6B	55	72	7A	6D	2E	70	6E	67	0E	AF	FD	3E	00	00	00	kUrzm.png ¯ý>
000068E0	00	49	45	4E	44	AE	42	60	82								IEND®B`,



34.Py-Py-Py

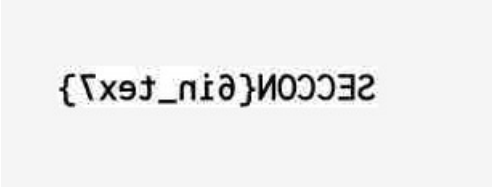
35.Reverse-it

下载下来一个不知道是什么文件，binwalk也看不出来，放进winhex看一下，可以文件末尾数字反转过过来就是jpg的文件头FFD8FF 。再结合题目名字，把所有的数字反转过来，winhex新建一个文件，保存jpg

格式

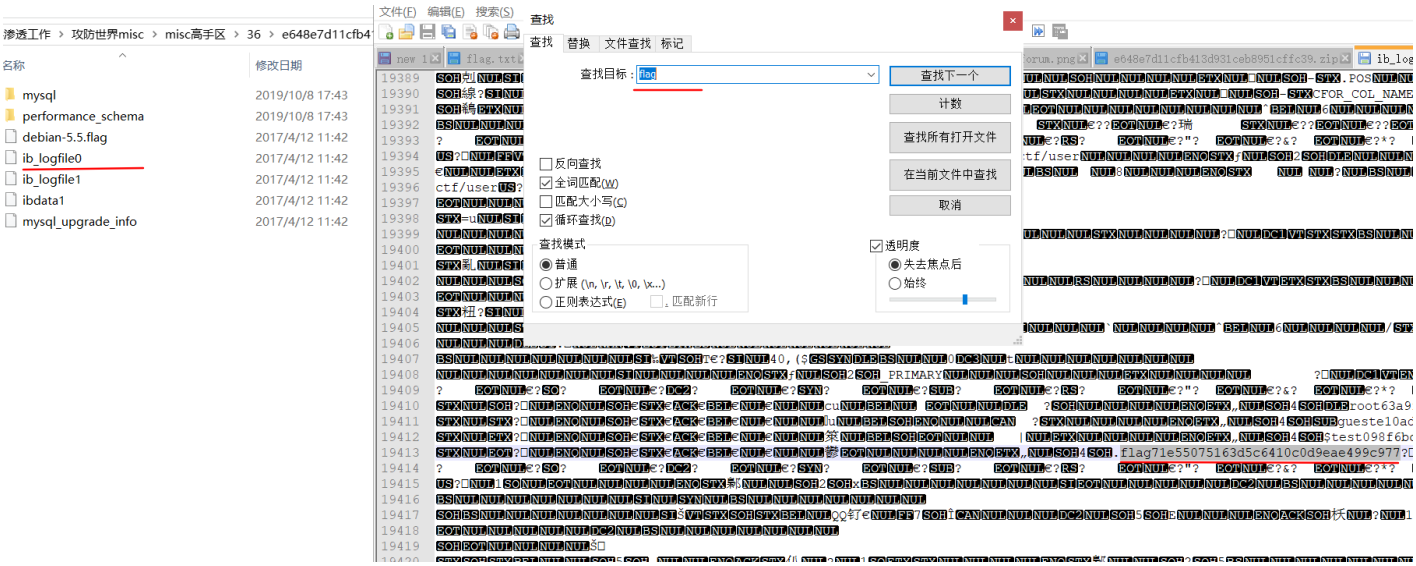
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
0001C60	D2	66	46	27	D2	23	23	F2	23	03	F2	93	93	93	13	F2	ÒfF'Ò##ò# ò""" ò
0001C70	76	27	F6	E2	33	77	E2	77	77	77	F2	F2	A3	07	47	47	v'öâ3wâwwwòò£ GG
0001C80	86	22	D3	66	46	27	A3	37	E6	C6	D6	87	02	64	44	25	†"ÓfF'£7æÆÖ‡ dD%
0001C90	A3	66	46	27	C3	02	E3	22	03	E2	43	E2	53	02	56	27	£fF'Ã ã" âCâS V'
0001CA0	F6	34	02	05	D4	85	22	D3	B6	47	07	D6	87	A3	87	02	ö4 Ô…"Ó¶G Ö‡£‡
0001CB0	22	F2	16	47	56	D6	A3	37	E6	A3	56	26	F6	46	16	22	"ò GVÖ£7æ£V&öF "
0001CC0	D3	87	A3	37	E6	C6	D6	87	02	16	47	56	D6	07	D6	87	Ó‡£7æÆÖ‡ GVÖ Ö‡
0001CD0	A3	87	C3	02	E3	F3	22	46	93	36	B6	A7	36	45	E4	A7	£‡Ã ãó"F"6¶§6Eä§
0001CE0	35	56	27	A7	84	96	86	56	34	07	D4	03	D4	53	75	22	5V'§„–†V4 Ô ÔSu"
0001CF0	D3	46	96	02	22	FB	BB	FE	22	D3	E6	96	76	56	26	02	ÓF– "û»þ"Óæ–vV&
0001D00	47	56	B6	36	16	07	87	F3	C3	00	F2	03	E2	13	F2	07	GV¶6 ‡óÃ ò â ò
0001D10	16	87	F2	D6	F6	36	E2	56	26	F6	46	16	E2	37	E6	F2	‡òÖö6âV&öF â7æò
0001D20	F2	A3	07	47	47	86	09	90	1E	FF	00	00	00	00	A1	00	ò£ GG† ÿ ¡
0001D30	00	00	10	00	00	00	40	00	30	0A	8C	00	00	00	10	00	@ 0 Œ
0001D40	00	00	40	00	20	0A	00	00	10	00	10	00	00	00	30	00	@ 0
0001D50	10	0A	30	00	00	23	03	A3	13	43	A3	03	13	02	23	23	0 # £ C£ ##
0001D60	A3	03	03	A3	13	03	03	23	64	00	14	46	F6	26	56	02	£ £ #d Fö&V
0001D70	05	86	F6	47	F6	37	86	F6	07	02	54	C6	56	D6	56	E6	†öGö7†ö TÆVÖVæ
0001D80	47	37	10	00	00	00	84	00	00	00	10	00	00	00	84	00	G7 „ „
0001D90	00	00	00	00	00	00	0A	00	00	00	10	00	00	00	40	00	@
0001DA0	96	78	C8	00	00	00	41	00	00	00	20	00	23	10	27	00	–xÈ A # '
0001DB0	00	00	91	00	00	00	20	00	13	10	00	00	20	00	10	00	‘
0001DC0	00	00	30	00	82	10	A6	00	00	00	10	00	00	00	50	00	0 ‚ ¦ P
0001DD0	B1	10	26	00	00	00	10	00	00	00	50	00	A1	10	00	00	± & P ¡
0001DE0	10	00	10	00	00	00	30	00	21	10	70	00	80	00	00	00	0 ! p €
0001DF0	A2	00	D4	D4	00	00	66	96	87	54	2D	00	1E	FF	00	00	¢ ÔÔ f–‡T– ÿ
0001E00	84	00	84	00	10	10	10	00	64	94	64	A4	01	00	0E	FF	„ „ d”d¤ ÿ
0001E10	8D	FF															ÿ

新文件图片如下，有点难看哈，拿一个镜子对着图片照，就容易看了，flag为：SECCON{6in_tex7}



36.mysql

下载出来一个压缩包，再解压后的所有文件内容搜索关键字flag，再ib_logfile0文件找到flag，

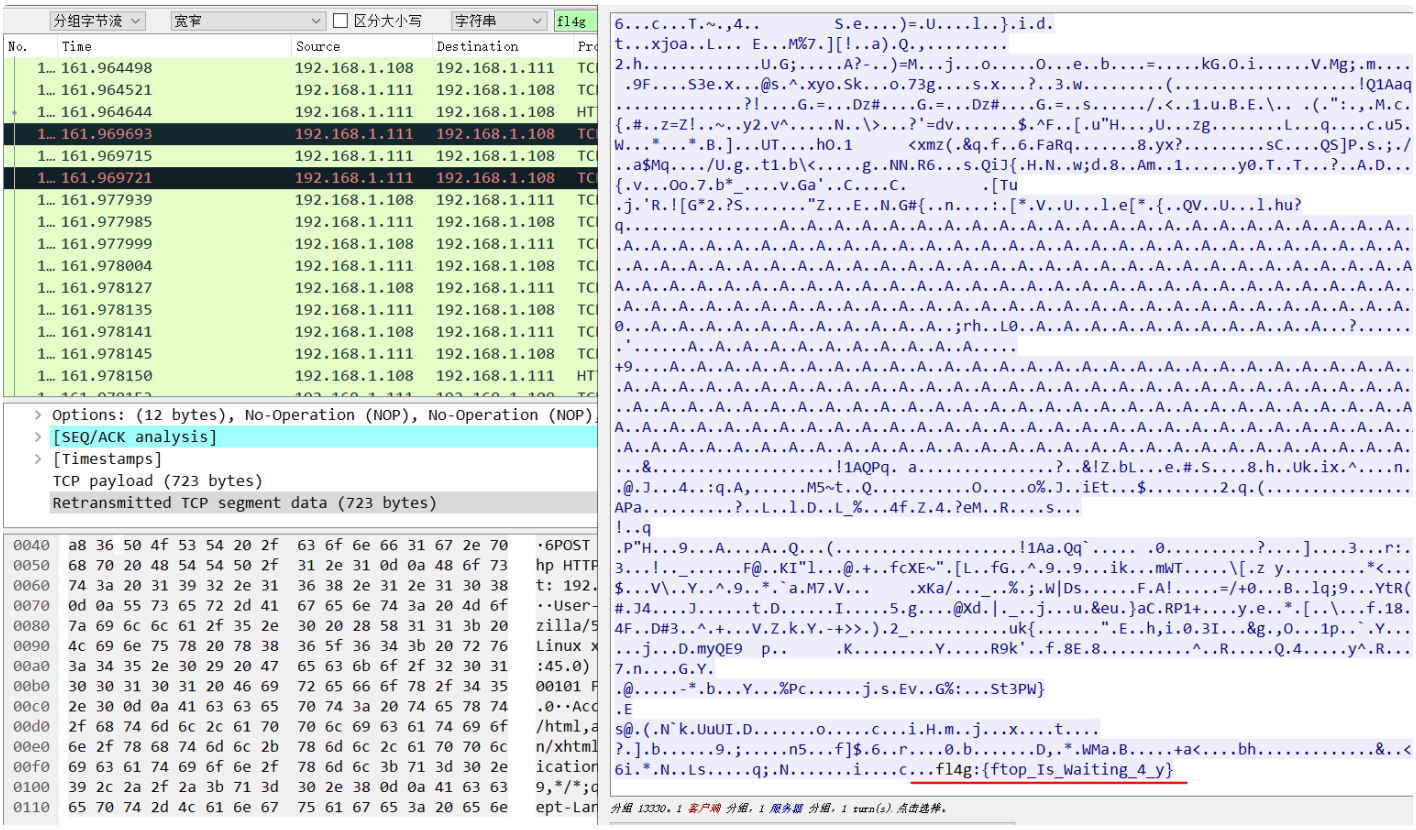


37.4433

38.4433

40.心仪的公司

下载解压时一个流量包，追踪http流可以看到flag，有点难找啊


分组字节流
宽窄
区分大小写
字符串
fl4g
No. Time Source Destination Pro
1… 161.964498 192.168.1.108 192.168.1.111 TC
1… 161.964521 192.168.1.111 192.168.1.108 TC
1… 161.964644 192.168.1.111 192.168.1.108 HT
1… 161.969693 192.168.1.111 192.168.1.108 TC
1… 161.969715 192.168.1.111 192.168.1.108 TC
1… 161.969721 192.168.1.111 192.168.1.108 TC
1… 161.977939 192.168.1.108 192.168.1.111 TC
1… 161.977985 192.168.1.111 192.168.1.108 TC
1… 161.977999 192.168.1.108 192.168.1.111 TC
1… 161.978004 192.168.1.111 192.168.1.108 TC
1… 161.978127 192.168.1.108 192.168.1.111 TC
1… 161.978135 192.168.1.111 192.168.1.108 TC
1… 161.978141 192.168.1.108 192.168.1.111 TC
1… 161.978145 192.168.1.111 192.168.1.108 TC
1… 161.978150 192.168.1.108 192.168.1.111 HT
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP),
[SEQ/ACK analysis]
[Timestamps]
TCP payload (723 bytes)
Retransmitted TCP segment data (723 bytes)
0040 a8 36 50 4f 53 54 20 2f 63 6f 6e 66 31 67 2e 70 ·6POST
0050 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 hp HTTP
0060 74 3a 20 31 39 32 2e 31 36 38 2e 31 2e 31 30 38 t: 192.
0070 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f ··User-
0080 7a 69 6c 6c 61 2f 35 2e 30 20 28 58 31 31 3b 20 zilla/5
0090 4c 69 6e 75 78 20 78 38 36 5f 36 34 3b 20 72 76 Linux x
00a0 3a 34 35 2e 30 29 20 47 65 63 6b 6f 2f 32 30 31 :45.0)
00b0 30 30 31 30 31 20 46 69 72 65 66 6f 78 2f 34 35 00101 F
00c0 2e 30 0d 0a 41 63 63 65 70 74 3a 20 74 65 78 74 .0··Acc
00d0 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f /html,a
00e0 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c n/xhtml
00f0 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 2e ication
0100 39 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d 0a 41 63 63 9,*/*;q
0110 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 65 6e ept-Lan
fl4g:{ftop_Is_Waiting_4_y}
分组 13330. 1 客户端 分组, 1 服务器 分组, 1 turn(s). 点击选择。

42.很普通的数独