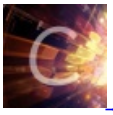
【Writeup】2017陕西网络空间安全技术大赛CSTC misc部分

原创

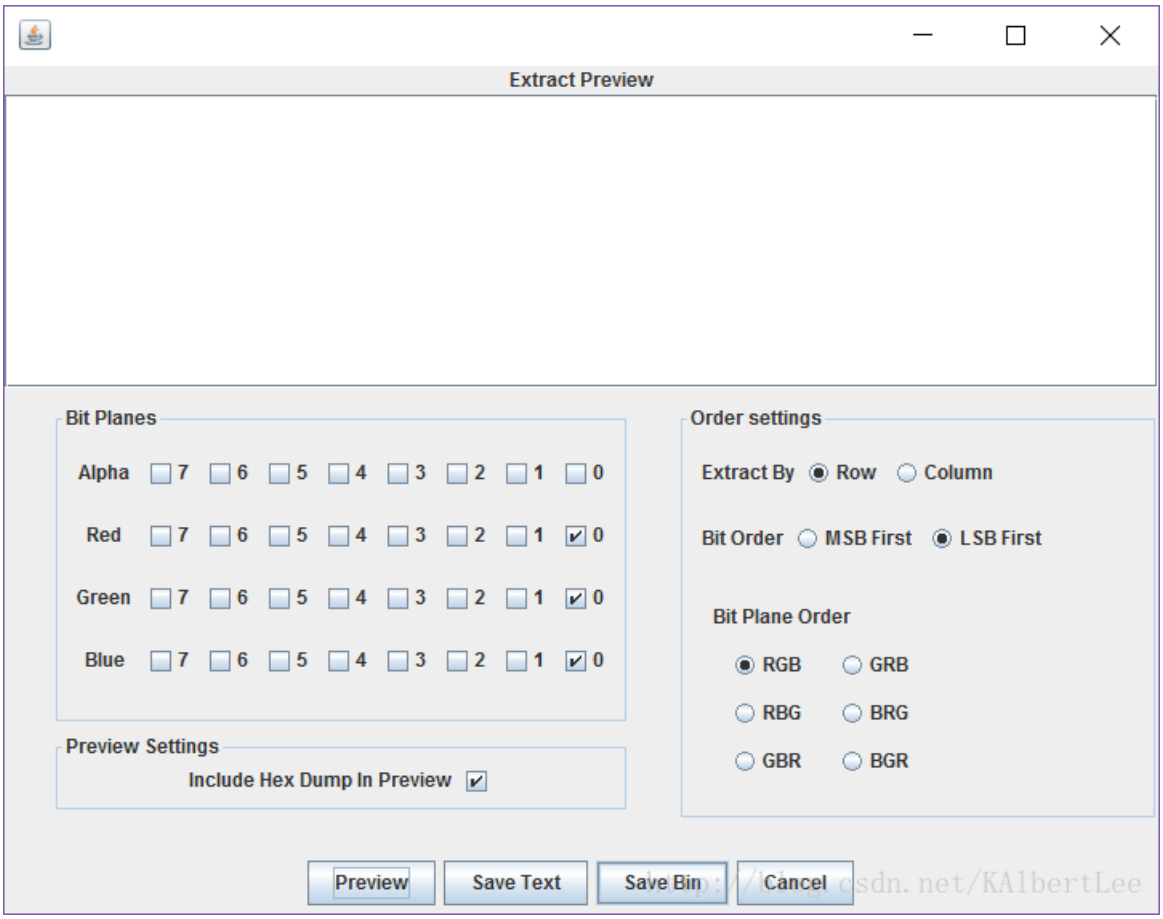
KAlbertLee 于 2017-04-17 19:51:46 发布 2557 收藏 1
分类专栏： CTF 文章标签： CTF 攻防 安全

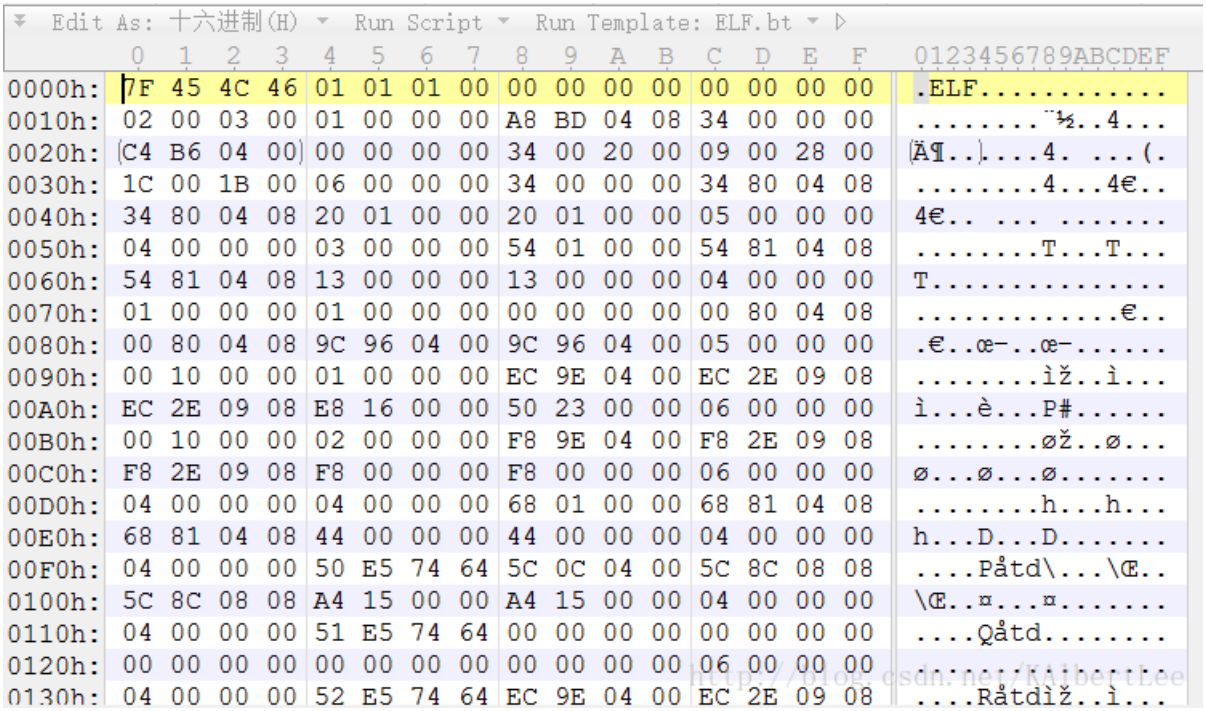


Misc 一维码

扫描一维码得到keyword:hydan

对一维码使用Stegsolve LSB隐写提取得到一个ELF文件





网上找hydan得到这个信息隐藏工具

Hydan: Information Hiding in Program Binaries - crazyboy.com
www.crazyboy.com/hydan/ ▾ 翻译此页
2003年3月15日 - Hydan [hI-dn]: Old english, to hide or conceal. Intro: Hydan steganographically conceals a message into an application. It exploits redundancy ...

安装好，然后执行

./hydan-decode flagrgb0.elf

密码为hydan

得到flag

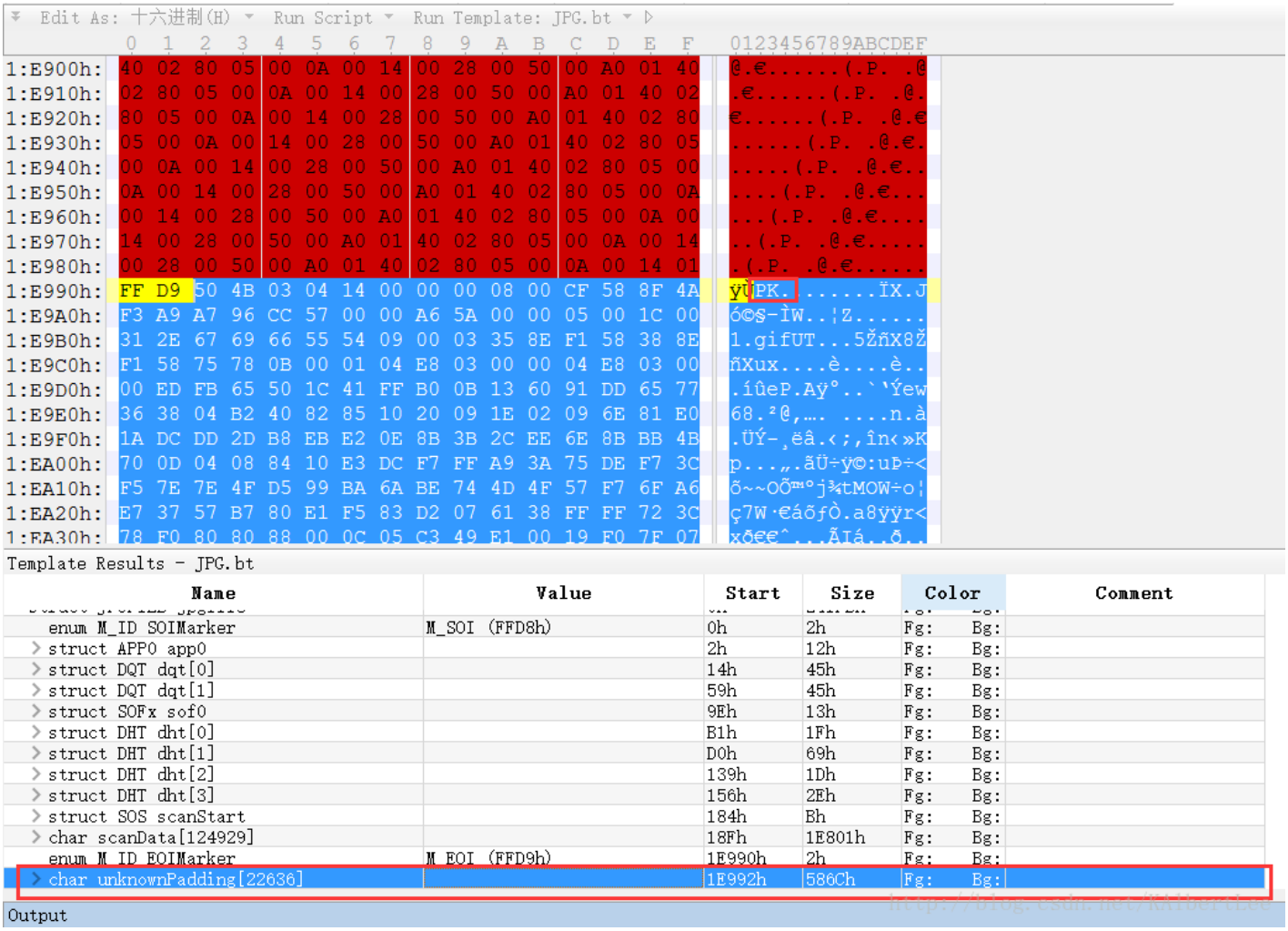
Misc 种棵树吧

解压之后有两个图片

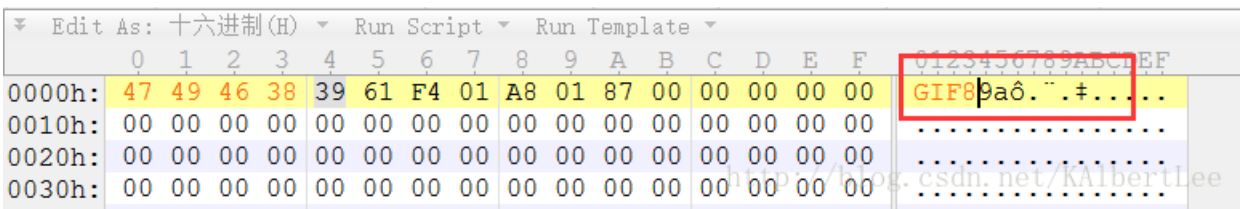
1111.jpg

2222.jpg

1111.jpg查看二进制后可见有一个压缩包，提取出来



提取得到的gif是损坏的，需要补齐头部“GIF8”



打开1.gif动图，单幅截下来，得到

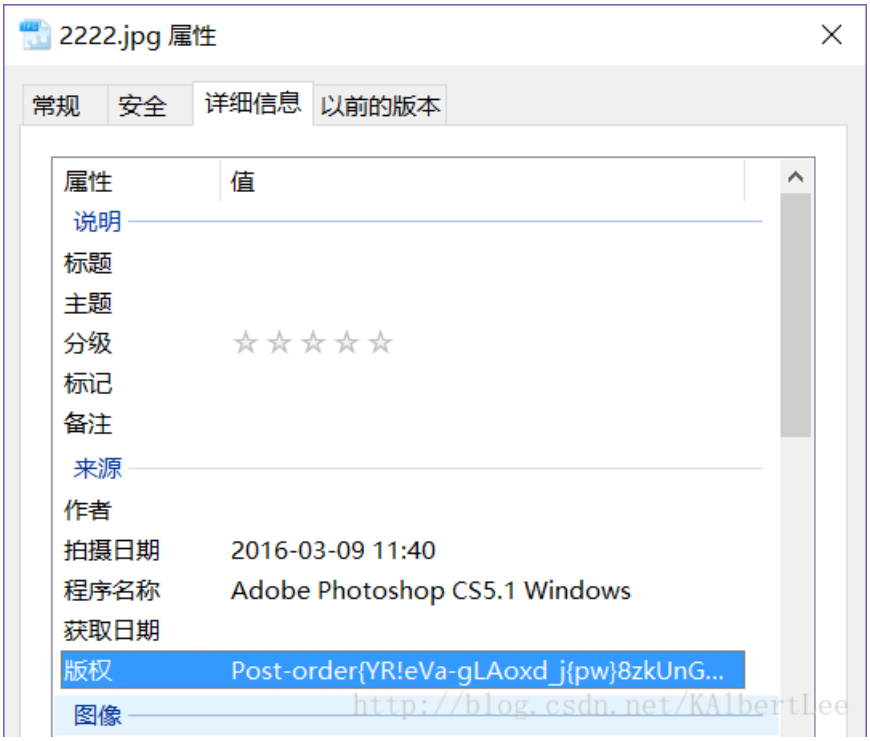

gif5.net
In-order{RY!heHVaL-goA
http://blog.csdn.net/KAlbertLee


gif5.net
!heHVaL-goAI{dxj_GpnUw8}
http://blog.csdn.net/KAlbertLee


gif5.net
nUw8}kzu*Er:s56fFl2i}
http://blog.csdn.net/KAlbertLee

In-order{RY!heHVaL-goAl{dxj_GpnUw8}kzu*Er:s56fFl2i}

2222.jpg查看其JPG图片属性，得到



Post-order{YR!eVa-gLAoxd_j{pw}8zkUnGulHh:r65f2lFsEi*}

则这两个字符串代表着中序遍历和后序遍历，根据题目意思，我们可以建出一棵树

In-order{RY!heHVaL-goAl{dxj_GpnUw8}kzu*Er:s56fFl2i}

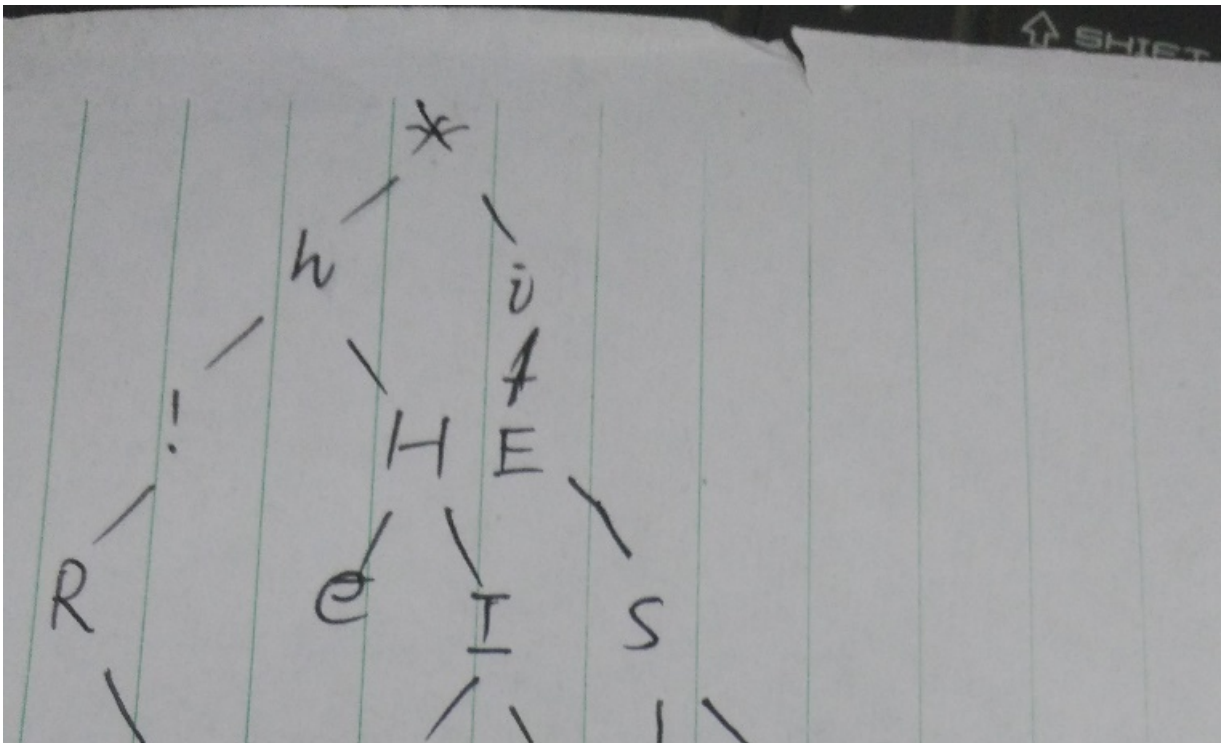
Post-order{YR!eVa-gLAoxd_j{pw}8zkUnGulHh:r65f2lFsEi*}

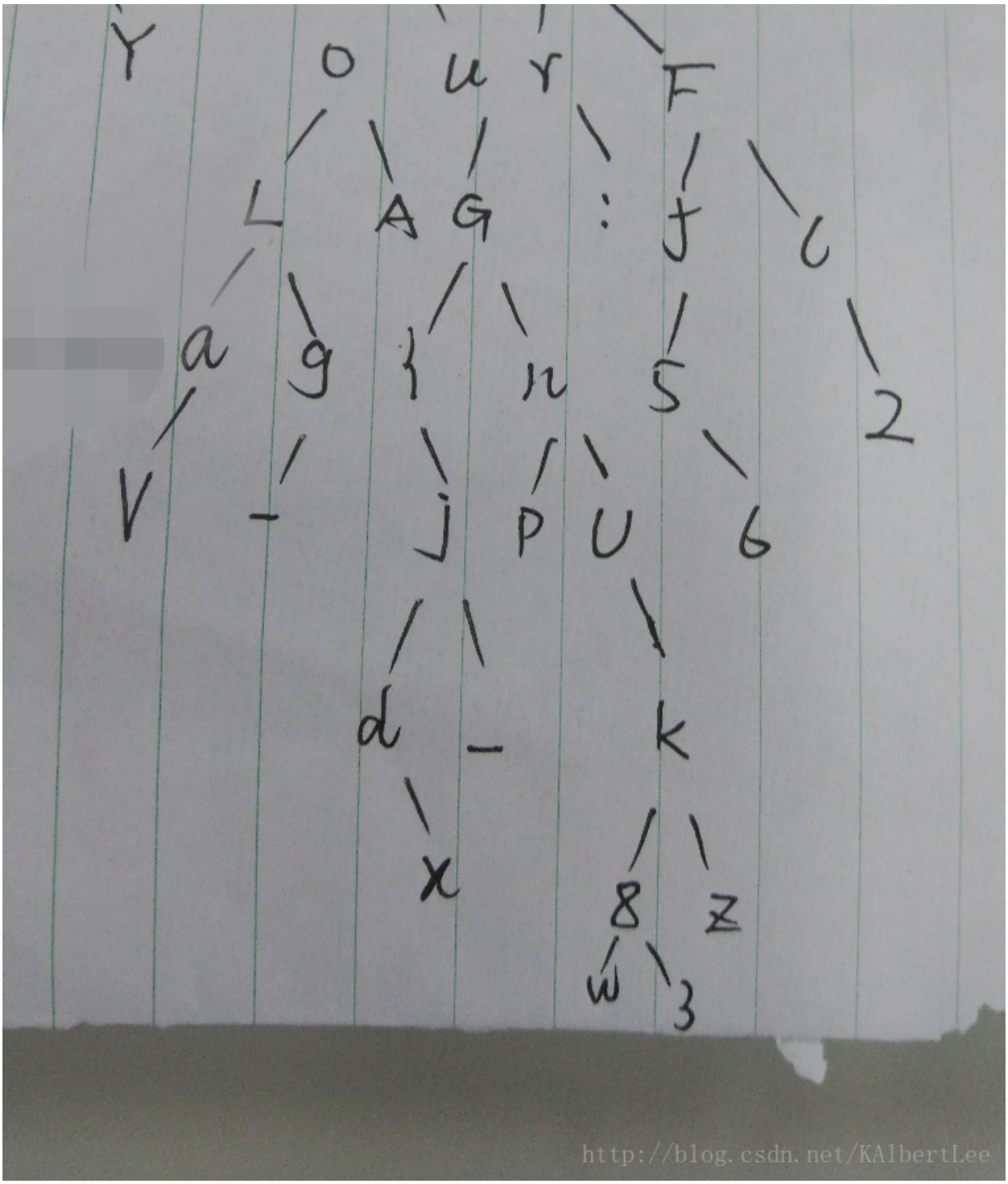
拿个脚本跑一下得到树结构

脚本来源：http://blog.csdn.net/hinyunsin/article/details/6316185

```
Build from preorder & inorder
Preorder: *h!RYHeIoLaVg-AuG{jdx_npUk8w}ziEsr:Ff56l2
Inorder: RY!heHVaL-goAI{dxj_GpnUw8}kzu*Er:s56fFl2i
Postorder: YR!eVa-gLAoxd_j{pw}8zkUnGuIHh:r65f2lFsEi*
The BTree is (* means no such a node):
*
--h
  --!
    --R
      --*
      --Y
        --*
        --*
    --*
  --H
    --e
      --*
      --*
    --I
      --o
        --L
          --a
            --V
              --*
```


以下省略部分，得到树结构：

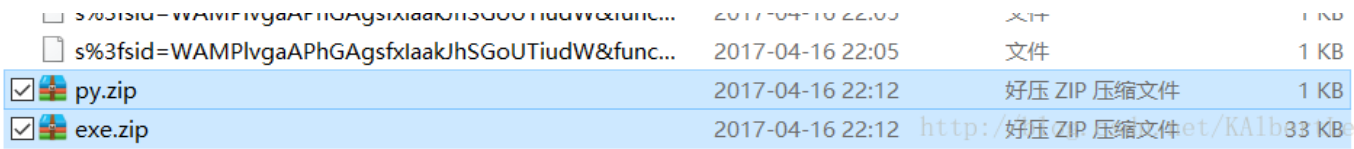




从上到下，从左到右读出字母：hi!HEReIsYourFLAG:flag{n52V-jpU6d_kx8zw}

Misc 乾坤

从文件HTTP流导出两个文件，里面包含py脚本和一个exe



在压缩包中exe结尾有一段类似于base64的字符串

```
L:5610h: 56 6C 7A 52 68 68 6B 53 58 5A 5F 56 30 4A 54 56   VlzRhhkSXZ_V0JTV
L:5620h: 5F 77 47 62 53 78 6D 53 46 4E 47 64 61 5A 6C 56   _wGbSxmSFNGdaZlV
L:5630h: 33 5A 6C 56 55 39 47 61 48 4A 2A 56 2A 74 32 55   3ZlVU9GaHJ*V*t2U
L:5640h: 35 6C 6C 52 6C 4A 45 63 74 5A 56 56 77 56 56 54   5llRlJEctZVVwVVT
L:5650h: 6F 68 47 62 55 56 6B 53 79 59 46 65 42 70 6D 56   ohGbUVkSyYFeBpmV
L:5660h: 55 70 55 4D 68 68 6C 54 45 4A 32 56 78 41 6A 56   UpUMhhlTEJ2VxAjV
L:5670h: 54 4A 5F 61 58 4E 44 62 2A 5F 30 56 53 52 6B 2A   TJ_aXNDb*_0VSRk*
L:5680h: 33 6C 56 4D 2A 52 44 5A 77 59 5F 56 34 64 30 56   3lVM*RDZwY_V4d0V
L:5690h: 2A 78 47 2A 56 6C 58 55 79 51 32 64 77 30 6D 56   *xG*VlXUyQ2dw0mV
L:56A0h:
```


而py脚本是一个encode

写一个脚本进行解密decode

```
from base64 import b64decode

flag = #‘from exe end’太多了就不贴了
#print flag
flag =flag.replace("_","1")
flag = flag.replace("*","W")
flag = list(flag)
flag.reverse()
flag = "".join(flag)
for i in range(0,25):
       flag=b64decode(flag)

print flag
flag：flag{n1_hEn_baNg_0}
```

Misc 轨迹

有一个记录USB的数据包

百度找到这篇文章http://bobao.360.cn/learning/detail/3351.html

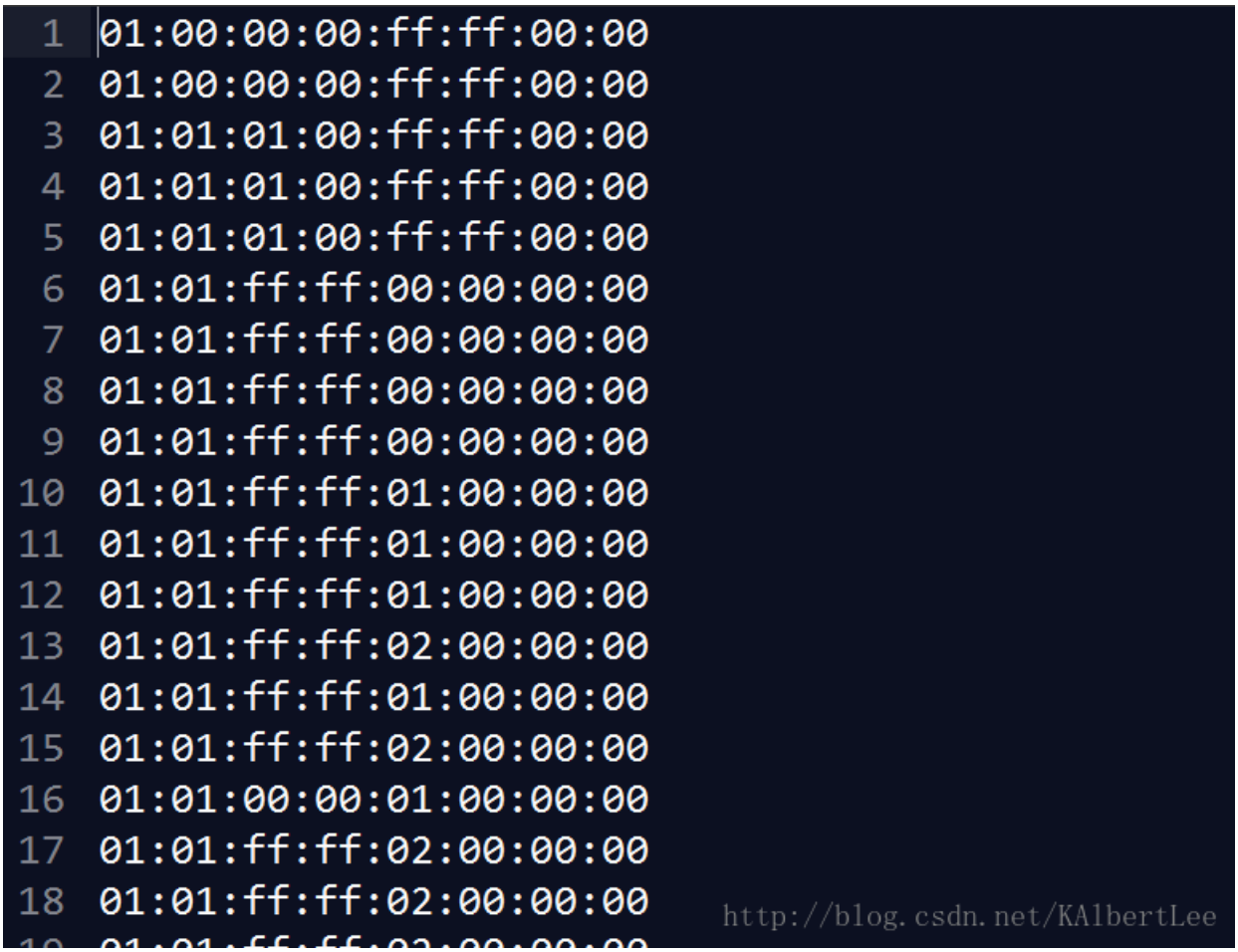
数据包里面传输了鼠标的位移数据，改变Letfover Capture Data

```
> Frame 746: 35 bytes on wire (280 bits), 35 bytes captured (280 bits)
> USB URB
  Leftover Capture Data: 01000200feff0000
```


从数据包截取出这个字段：

```
tshark.exe -r trace.io.pcap -T fields -eusb.capdata > usbdata.txt
```

得到如下：



然后根据上面那篇文章写一个脚本：

其中

int(line[1],16) 1是按下左键，2是按下右键，0没按

int(line[2],16) int(line[3],16)　第3位:0x0正0xff负->第2位:正右负左

int(line[4],16) int(line[5],16)　第5位:0x0正0xff负->第4位:正下负上

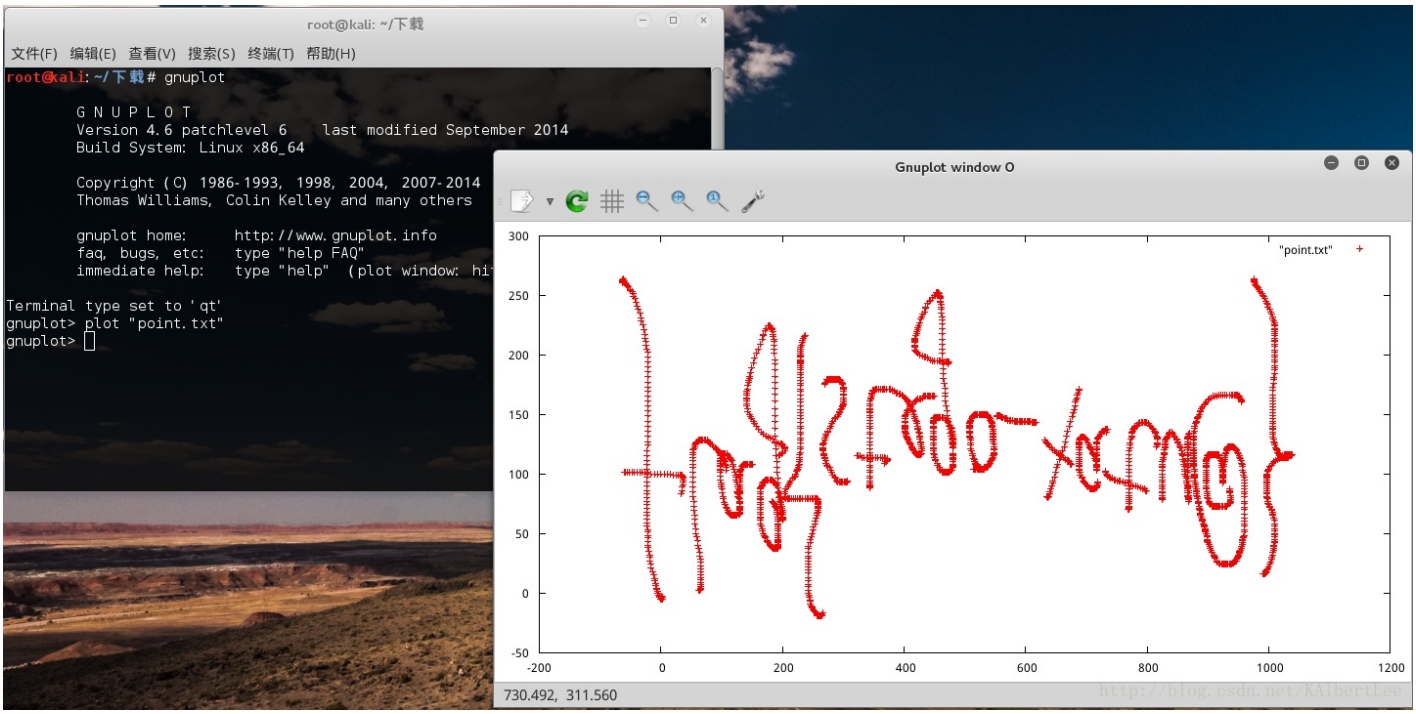
```
f =open("point.txt","w")

nums = []
data = open('usbdata.txt','r')
posx = 0
posy = 0
line = data.readline()
while line != "":
   line = line[:-1].split(":")
    x= int(line[2],16)
    y= int(line[4],16)
   if int(line[3],16) == 0xff :
       x -= 0x100
   if int(line[5],16) == 0xff :
       y -= 0x100
   posx += x
   posy += y
   btn_flag = int(line[1],16)  # 1for left , 2 for right , 0 for nothing
   if btn_flag == 1 :
       print >>f,posx , posy
   line = data.readline()
f.close()
```

于是得到鼠标的位移产生的点：

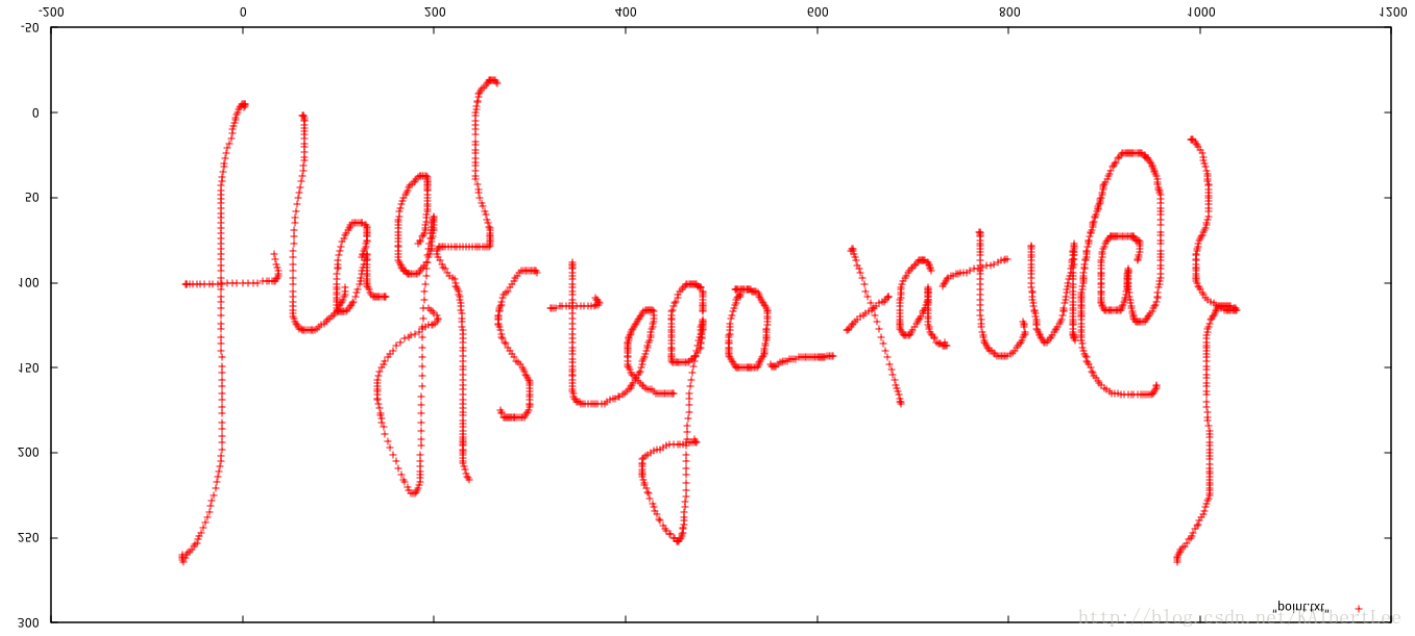
```
247  107 72
248  106 74
249  104 76
250  103 79
251  102 81
252  101 84
253  100 87
254  99 89
255  99 91
256  99 92
257  98 95
258  98 98
259  98 101
```


然后使用kali下的gunuplot进行作图：

plot <inputfile>



画反了，反转一下



flag{stego_xatu@}

Misc 我们的秘密（后面补的题）

本来在赛场内没做出来，不知道暴力保平安这个道理，赛后得到大佬提醒得到做法

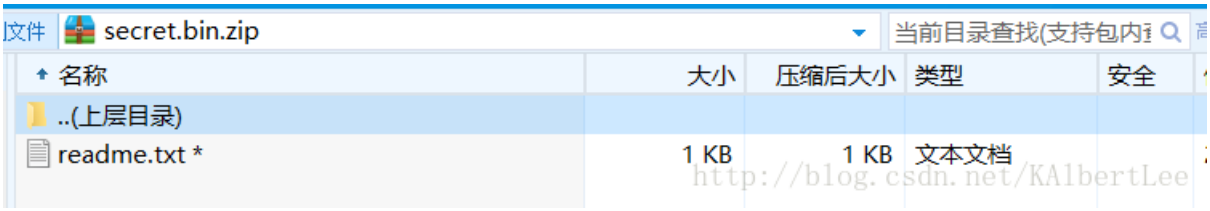
首先是一个文件，拖到hex查看器内发现是个压缩包，并且里面发现三个文件

```
0000h: 50 4B 03 04 14 00 09 00 08 00 8F 72 8B 4A 7A F0  PK.........r‹Jzð
0010h: E7 AE B9 00 00 00 CD 00 00 00 0A 00 00 00 72 65  ç®¹...Í.......re
0020h: 61 64 6D 65 2E 74 78 74 6B 44 23 D3 A2 93 EA 96  adme.txtkD#Ó¢"ê–
0030h: 62 72 40 23 B7 F5 3C 2E 71 15 7F 30 2F AB BE 33  br@#·õ<.q..0/«¾3
0040h: 8E 7C 46 94 8E 02 8B 4D EF 76 47 7E DF 1A BB D0  Ž|F"Ž.‹MïvG~ß.»Ð
0050h: 2C B7 7F 0C 69 63 F3 07 44 EA A6 C4 81 B4 C5 9A  ,·..icó.Dê¦Ä.´Åš
0060h: DF 69 0C B0 DF A7 35 79 7C 1E A1 A2 C3 EB AA E5  ßi.°ß§5y|.¡¢Ãëªå
0070h: 54 E9 51 68 7B 3A 57 6E 79 D1 73 6C E5 20 AF 00  TéQh{:WnyÑslå ¯.
0080h: CA 4A 29 D2 7A 0C 3C D2 4E 93 FB 22 34 60 D0 0E  ÊJ)Òz.<ÒN"û"4`Ð.
0090h: 25 E1 60 23 CC A0 EA A8 21 CF 67 76 28 A7 02 67  %á`#Ì ê¨!Ïgv(§.g
00A0h: 9B A1 C2 22 8C 05 24 8E 63 C3 C2 10 CF 6E EC 30  ›¡Â"Œ.$ŽcÃÂ.Ïnì0
00B0h: 75 F1 AC FE 4C 36 64 8F F2 D2 F1 66 CC 5C 80 E8  uñ¬þL6d.òÒñfÌ\€è
00C0h: C7 E2 BA 40 E7 3E AC 6E 9E 11 98 5F DC D6 25 91  Çâº@ç>¬nž.˜_ÜÖ%'
00D0h: A0 10 A5 17 82 DB 37 8D 1C 32 04 7F 1C A2 27 B5  .¥.‚Û7..2...¢'µ
00E0h: B2 50 4B 07 08 7A F0 E7 AE B9 00 00 00 CD 00 00  ²PK..zðç®¹...Í..
00F0h: 00 50 4B 03 04 14 00 09 00 08 00 11 71 8B 4A 64  .PK.........q‹Jd
0100h: AF E0 E7 3D 0E 08 00 77 1A 08 00 0D 00 00 00 61  ¯àç=...w.......a
0110h: 63 74 6F 72 73 68 6F 77 2E 6D 70 34 3B FA DD AB  ctorshow.mp4;úÝ«
0120h: AF 8B 56 03 29 9A 06 A9 3A 46 4E AF 08 20 80 C4  ¯‹V.)š.©:FN¯. €Ä
0130h: 28 F4 72 88 A9 63 76 CD 3C 79 DE F3 CD C4 A1 E3  (ôrˆ©cvÍ<yÞóÍÄ¡ã
```

Template Results - ZIP.bt

Name	Value	Start	Size	Color
struct ZIPFILERECORD record[0]	readme.txt	0h	E1h	Fg: Bg:
struct ZIPDATADESCR dataDescr[0]		E1h	10h	Fg: Bg:
struct ZIPFILERECORD record[1]	actorshow.mp4	F1h	80E68h	Fg: Bg:
struct ZIPDATADESCR dataDescr[1]		80F59h	10h	Fg: Bg:
struct ZIPFILERECORD record[2]	cool.wav	80F69h	E927h	Fg: Bg:
struct ZIPDATADESCR dataDescr[2]		8F890h	10h	Fg: Bg:
struct ZIPDIRENTRY dirEntry[0]	readme.txt	8F8A0h	5Ch	Fg: Bg:
struct ZIPDIRENTRY dirEntry[1]	actorshow.mp4	8F8FCh	5Fh	Fg: Bg:
struct ZIPDIRENTRY dirEntry[2]	cool.wav	8F95Bh	5Ah	Fg: Bg:
struct ZIPENDLOCATOR endLocator[0]		8F9B5h	16h	Fg: Bg:
struct ZIPFILERECORD record[3]	readme.txt	8F9CBh	D5h	Fg: Bg:
struct ZIPDIRENTRY dirEntry[3]	readme.txt	8FAA0h	5Ch	Fg: Bg:
struct ZIPENDLOCATOR endLocator[1]		8FAFCh	16h	Fg: Bg:

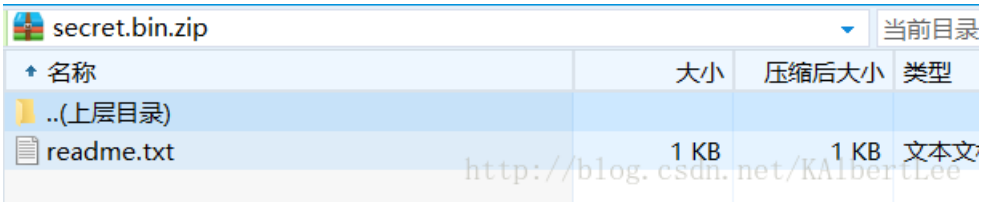
然而以压缩包形式打开却只有一个文件。。。并且加密了

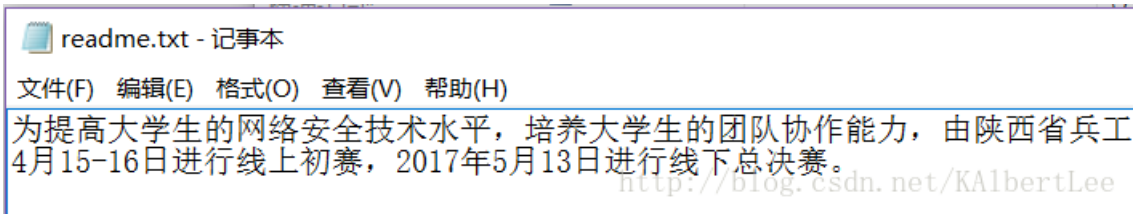


尝试使用ZipCenOp.jar解密发现可以，证明是伪加密

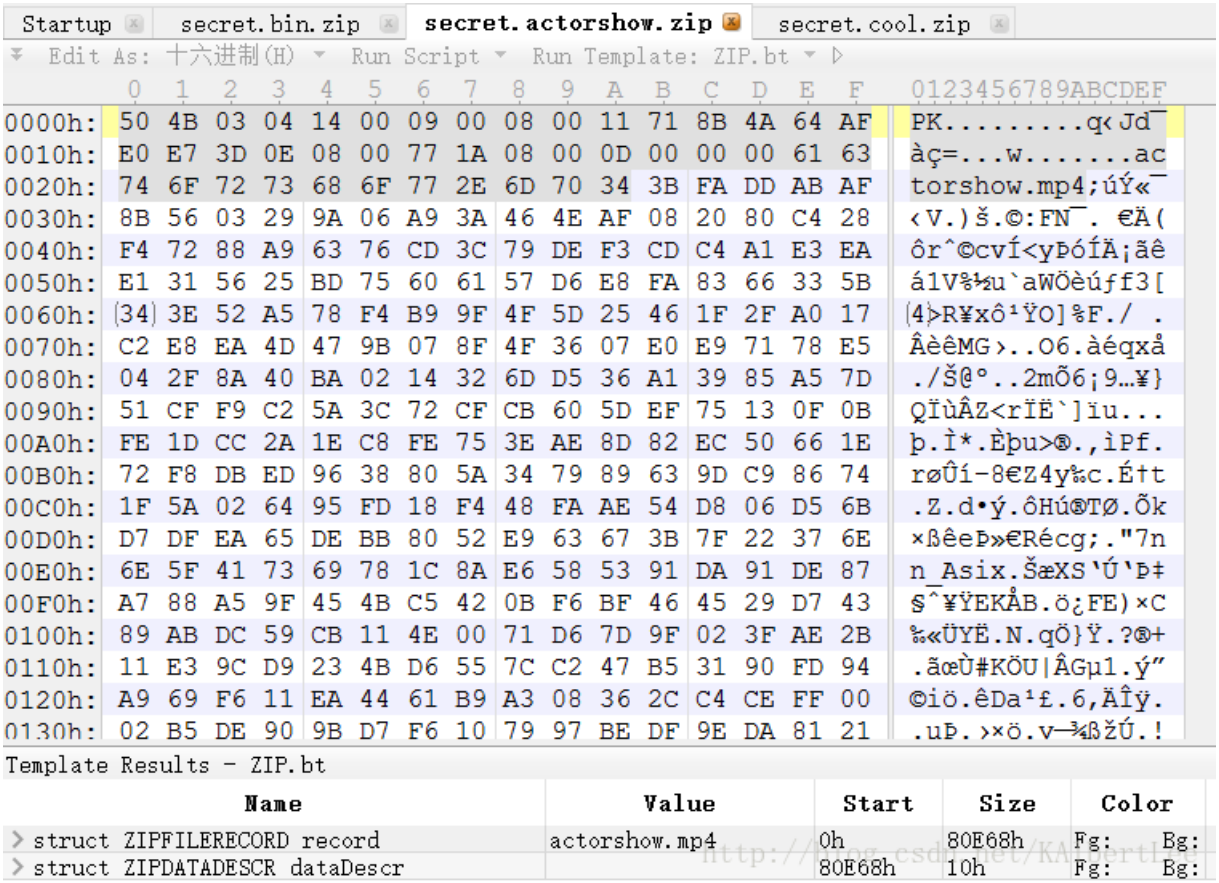
```
E:\学习资料\CTF\CTF工具\解密>java -jar ZipCenOp.jar r secret.bin.zip
success 4 flag(s) found
E:\学习资料\CTF\CTF工具\解密>
```


打开之后却发现这个txt。。也没什么用





之后只能是把另外两个文件抠出来



Startup | secret.bin.zip | secret.actorshow.zip | **secret.cool.zip**

Edit As: 十六进制(H) | Run Script | Run Template: ZIP.bt

```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 50 4B 03 04 14 00 09 00 08 00 BB 70 8B 4A A8 B4  PK........»p‹J¨´
0010h: D8 F9 01 E9 00 00 0E 49 03 00 08 00 00 00 63 6F  Øù.é...I......co
0020h: 6F 6C 2E 77 61 76 07 1E 56 05 B5 B4 A2 25 80 3F  ol.wav..V.µ´¢%€?
0030h: 8A 83 27 AD 00 4B 87 E7 73 BA A6 82 CD 86 1A DF  Šƒ'-.K‡çs°¦‚Í†.ß
0040h: A6 DB 02 76 EC 7E 37 EE 69 D9 41 AC 75 4B C2 F2  ¦Û.vì~7îiÙA¬uKÂò
0050h: FD 9A D4 1A BC 8A 12 E0 63 AE 90 3B 4D 6D 05 0F  ýšÔ.¼Š.àc®.;Mm..
0060h: 93 DE 5D 73 35 C8 22 BC AF 73 EE BC BD B2 FD 46  "Þ]s5È"¼¯sî¼½²ýF
0070h: C3 89 C0 D1 EB 43 3F 69 DE 74 65 18 48 39 26 AA  Ã‰ÀÑëC?iÞte.H9&ª
0080h: 97 03 5F 2A 98 EA FF 05 80 A0 5F 91 6C C6 AE 36  —._*˜êÿ.€ _'lÆ®6
0090h: C1 F3 42 06 A2 A5 32 5F FF C9 C7 B0 F3 32 F3 FE  ÁóB.¢¥2_ÿÉÇ°ó2óþ
00A0h: D1 B4 7A C1 05 31 F8 07 25 4D BC C7 80 1B 11 87  Ñ´zÁ.1ø.%M¼Ç€..‡
00B0h: 31 99 33 58 01 6E 8F 51 0D BD E2 05 D9 06 E4 46  1™3X.n.Q.½â.Ù.äF
00C0h: 01 ED D7 8C 7E A7 C9 81 BC 17 79 FD 52 E1 11 1F  .í×Œ~§É.¼.yýRá..
00D0h: 1A 0B 81 03 78 0A 32 84 75 6F 8E B6 19 BA C1 4C  ....x.2„uoŽ¶.°ÁL
00E0h: A5 F5 85 D6 43 15 86 C4 90 8F 2A 46 B4 3D B8 B8  ¥õ…ÖC.†Ä..*F´=¸¸
00F0h: 29 44 4E E7 BF A4 12 1B 7B 5B F7 CF C3 8C 7B 9A  )DNç¿¤..{[÷ÏÃŒ{š
0100h: 15 E4 25 D2 01 EC 38 72 17 7A 6F 74 13 27 0B 8A  .ä%Ò.ì8r.zot.'.Š
0110h: E4 EA 80 21 62 A1 BC 01 D4 3F 69 D8 B2 84 86 79  äê€!b¡¼.Ô?iØ²„†y
0120h: 5C 5F 74 93 E7 F9 A7 93 6C 26 16 34 E7 BC 32 F6  \_t"çù§"l&.4ç¼2ö
0130h: A0 D8 54 A8 AD BC 2E A3 01 48 96 26 88 B8 AD 96   ØT¨-¼.£.H–&ˆ ¸–
```

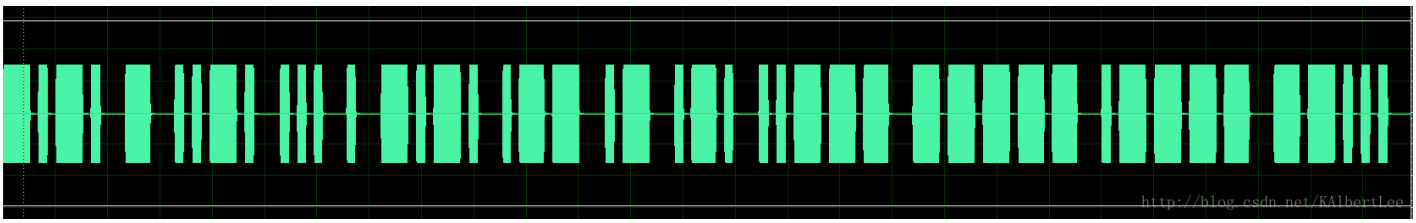
Template Results - ZIP.bt

Name	Value	Start	Size	Color
struct ZIPFILERECORD record	cool.wav	0h	E927h	Fg: Bg:
struct ZIPDATADESCR dataDescr		E927h	10h	Fg: Bg:

发现这两个压缩包也同样需要解密。。。并且ZipCenOp.jar也无法解开，可能是真加密

（经大佬提点）对cool.zip这个压缩包进行口令爆破，额，爆破过程就省略了

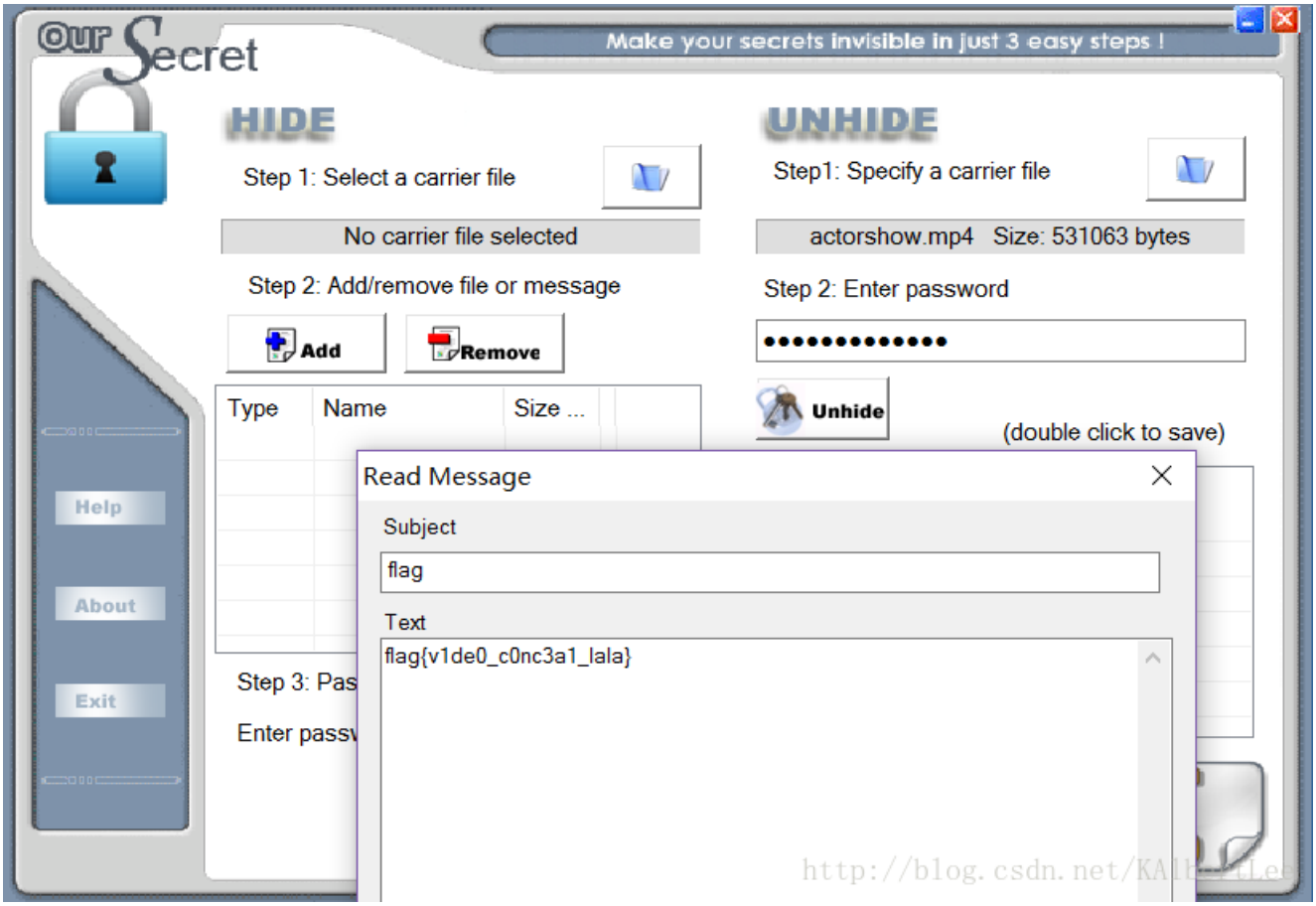
得到的是3xatu2o17，解开得到cool.wav



是摩斯电码，解密得到CTFSECWAR2017

然后以3xatu2o17作为密码解密actorshow.zip，得到一个《人民的名义》小视频（出题人最近很上瘾啊）

并不能发现有什么异常，然后（经大佬提点）使用oursecret（我们的秘密）一个信息隐藏软件进行解密，解密密码用上一步的CTFSECWAR2017



得到flag{v1de0_c0nc3a1_lala}

Misc 什么玩意（后面补的题）

得到两个文件：whatisit和whatisthat，首先先看whatisthat，里面有明文，百度谷歌各种方式，查到这是个Bluetooth握手包，需要在文件名补上.cvs表单格式

	Protocol	LMP	XiAn		
2	Index	Slave/Mas	Type	Descriptio	Payload Data
3	5	Master	DM1	LMP_versi	4A 02 0A 00 DB 04
4	12	Slave	DM1	LMP_versi	4C 01 01 00 2C 02
5	19	Master	DM1	LMP_featu	4E BF FE 0F 00 18 18 00 00
6	26	Slave	DM1	LMP_featu	50 BF 28 21 00 00 00 00 00
7	33	Master	DM1	LMP_host_	66
8	62	Slave	DM1	LMP_acce	06 33
9	66	Slave	DM1	LMP_setu	63
10	69	Master	DM1	LMP_setu	62
11	73	Master	DM1	LMP_max_	5A 05
12	75	Master	DM1	LMP_max_	5C 05
13	120	Slave	DM1	LMP_acce	06 2E
14	127	Master	DM1	LMP_clkof	0A
15	134	Slave	DM1	LMP_clkof	0C 86 44
16	137	Master	DM1	LMP_nam	02 00
17	146	Slave	DM1	LMP_nam	04 00 0A 4E 6F 70 68 72 65 74 65 74 65 00 00 00 00
18	152	Slave	DM1	LMP_nam	04 00 0A 4E 6F 70 68 72 65 74 65 74 65 00 00 00 00
19	153	Master	DM1	LMP_featu	4E BF FE 0F 00 18 18 00 00
20	160	Slave	DM1	LMP_featu	50 BF 28 21 00 00 00 00 00
21	187	Master	DM1	LMP_versi	4A 02 0A 00 DB 04
22	194	Slave	DM1	LMP_versi	4C 01 01 00 2C 02
23	5933	Master	DM1	LMP_in_ra	10 EC 50 3F 96 EF 26 97 7E 4E DE 35 10 9D 6A 91 68
24	6004	Slave	DM1	LMP_nam	03 00
25	6011	Master	DM1	LMP_nam	05 00 05 41 75 64 69 74 00 00 00 00 00 00 00 00 00
26	13536	Slave	DM1	LMP_acce	06 08
27	13561	Master	DM1	LMP_com	12 76 4F DA 77 B7 EE 88 9A 6C 11 D0 CA 08 83 73 CD
28	13568	Slave	DM1	LMP_com	12 FF 80 DF F2 CD 72 83 76 83 A4 9C C9 A7 F1 C3 BB

然后再寻找Bluetooth破解工具，在github找到这么一个东西：BTcrack

简单学习一下使用方法，然后可以进行破解：

```
root@kali:~/下载/btcrack-master# ./btcrack 10 00:11:9F:C4:F3:AE 00:60:57:1A:6B:F
1 whatisthat.csv
Link Key:  f7:e6:e3:2c:1d:2a:0b:5f:c2:4c:41:fa:b5:30:8c:b7
Pin:  9955
Pins/Sec:  12296
```

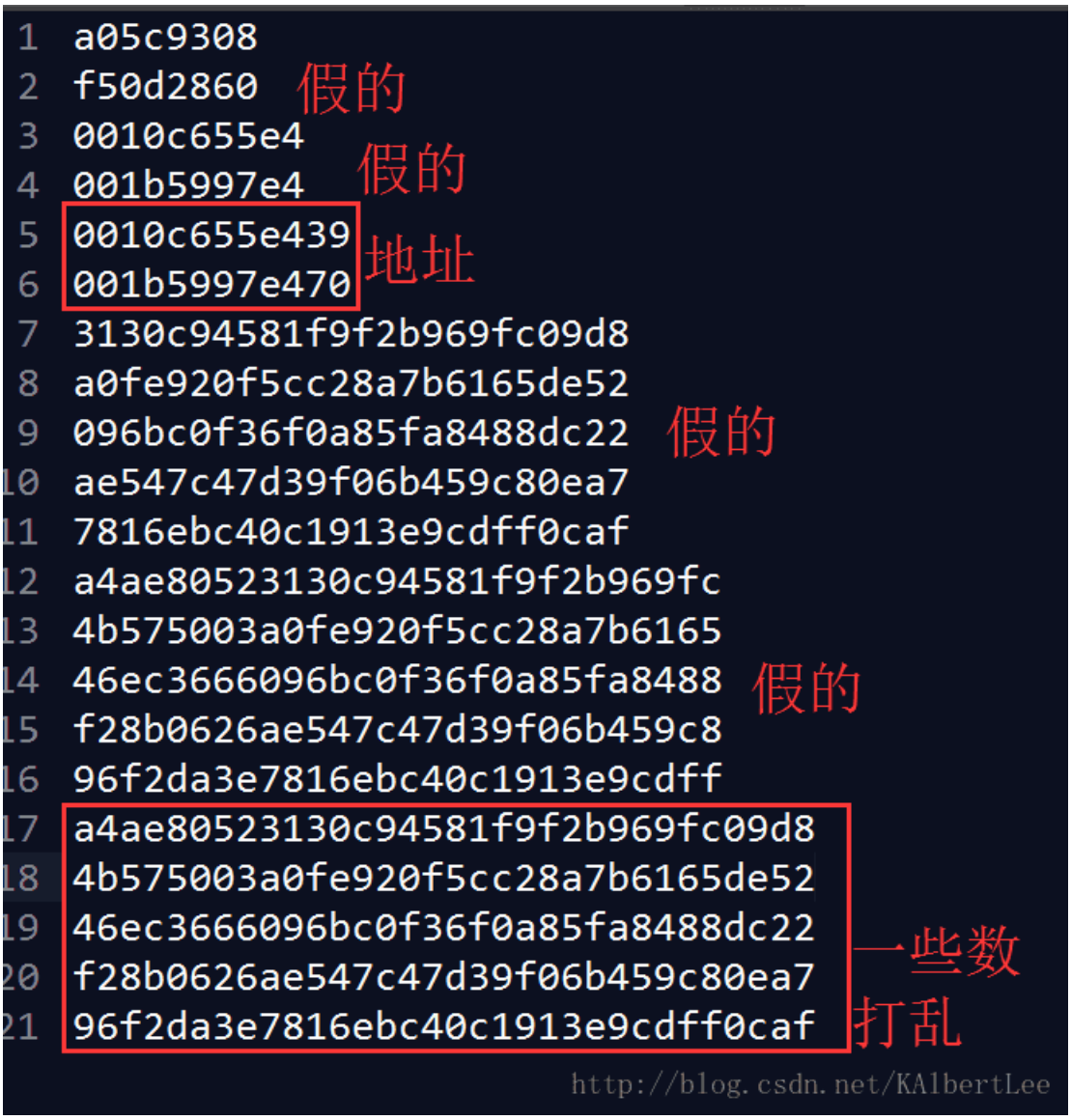

尝试以这里的Link Key: f7:e6:e3:2c:1d:2a:0b:5f:c2:4c:41:fa:b5:30:8c:b7作为flag提交，可是不对

可能是whatisit里面才包含key

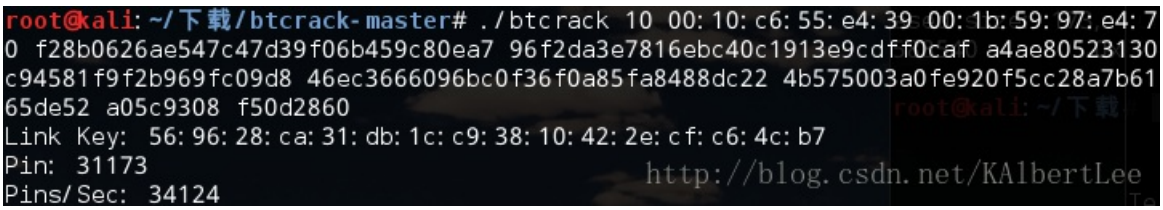
然后得采用第二种格式的输入才行，于是在whatisit里面找对应的参数

```
root@kali:~/下载/btcrack-master# ./btcrack ?
./btcrack <#threads> <master addr> <slave addr> <filename.csv>
./btcrack <#threads> <master addr> <slave addr> <in_rand> <comb_master> <comb_sl
ave> <au_rand_m> <au_rand_s> <sres_m> <sres_s>
root@kali:~/下载/btcrack-master#
```


经过近百次的尝试，终于找到对应的参数输入



就是这个



拿到Link Key: 56:96:28:ca:31:db:1c:c9:38:10:42:2e:cf:c6:4c:b7

flag{569628ca31db1cc93810422ecfc64cb7}

至此 Misc .AK .END