HBCTF——WriteUp&&涨姿势（2）

原创

浅零半泣 于 2017-04-23 12:13:11 发布

分类专栏： CTF 文章标签： CTF

WriteUp

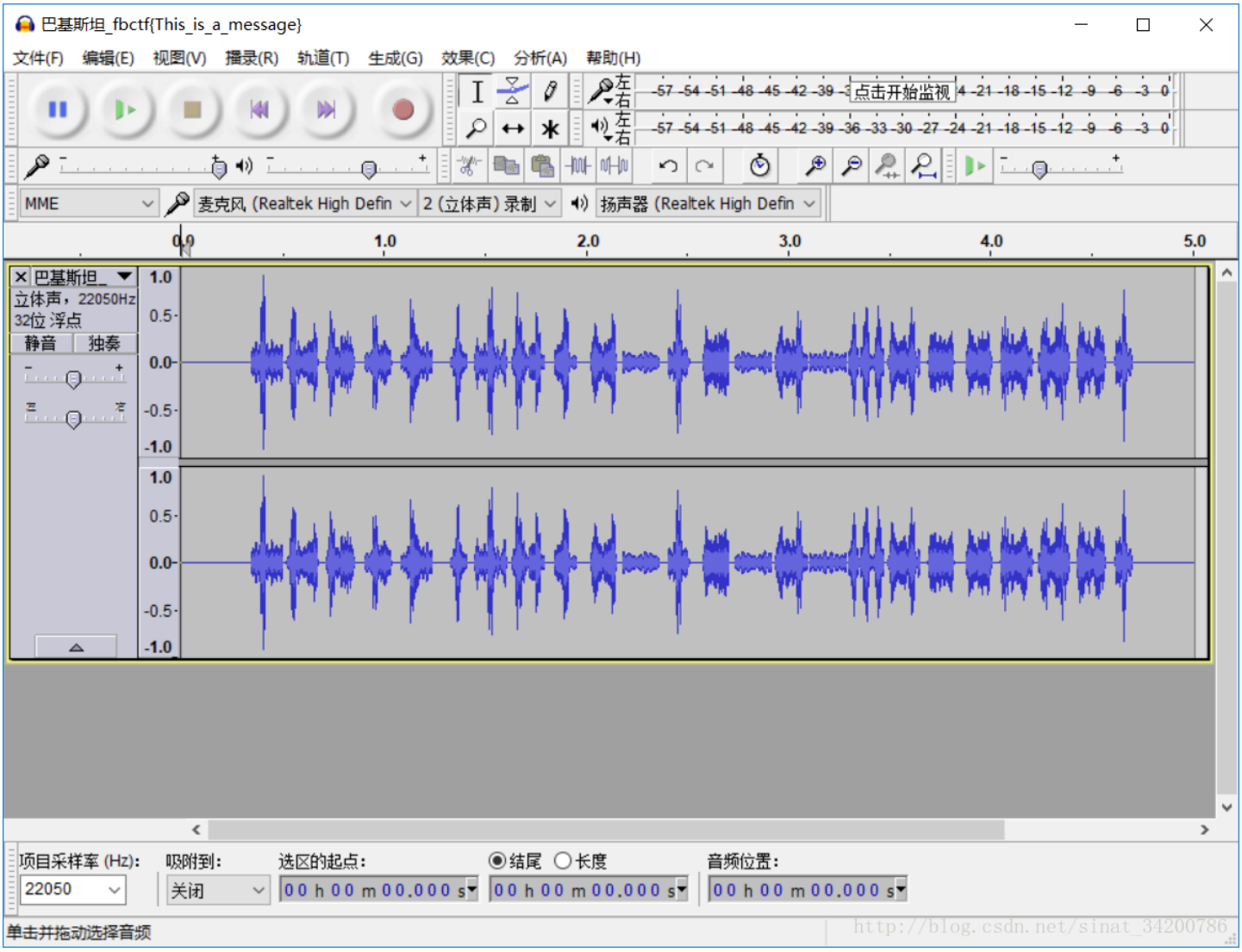
1. 巴基斯坦_听音乐
2. 塔吉克_扫一扫
3. 印尼_find_the_key
4. 安哥拉_一步一步
5. 北韩_经过多少次加密
6. 日本_hundouluo
7. 伊朗_爆破
8. 利比亚_密码多简单
9. 越南_真的加密了？
10. 埃塞俄比亚-隐写术
11. 秘鲁_源代码给你又如何

涨姿势

1. 坦桑尼亚-知己知彼

巴基斯坦_听音乐

原题

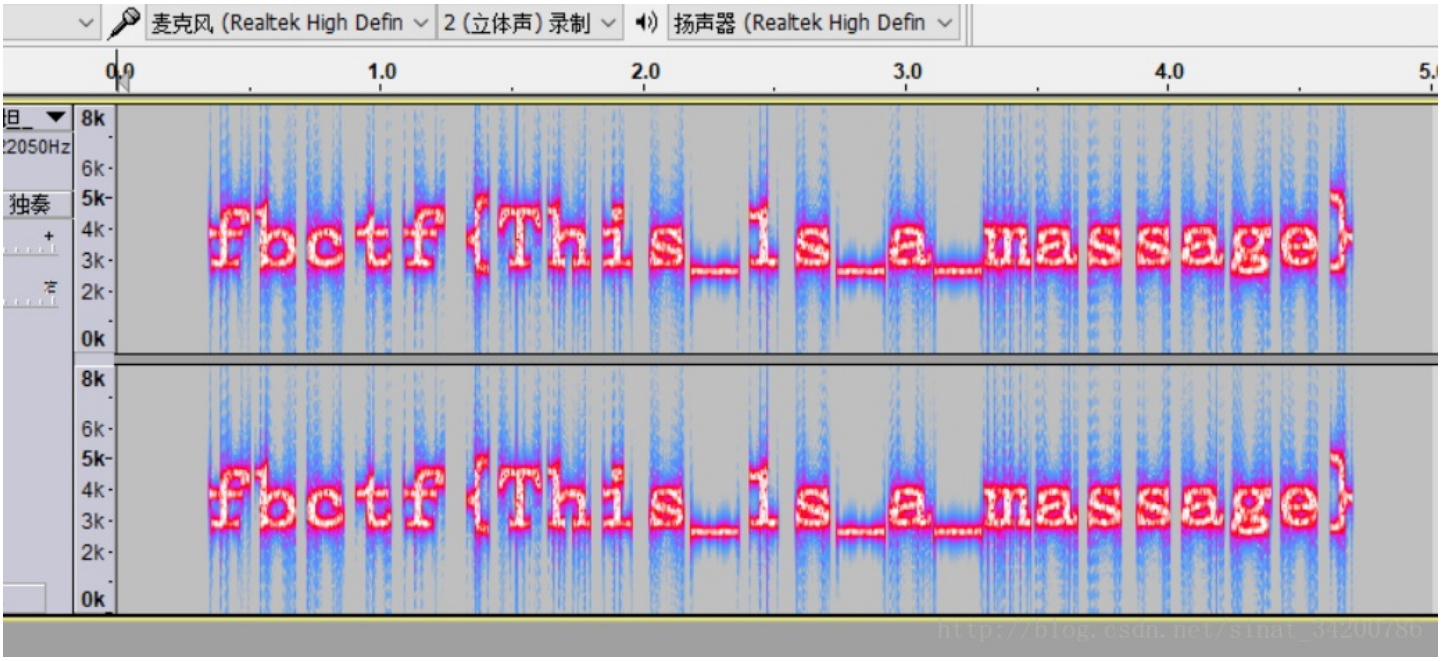


解题思路

简单音频隐写题在波形，频谱，高音等找线索

WriteUp

选择频谱图,找到答案



塔吉克_扫一扫

原题

解题思路

先扫了再说，扫完出现一串数字

```
504B0304140000080800FB1294AE64036D40F000000
0D0000000500000031 2E7478744B4B4A2E49AB2ECFCC
CB48ADA80500504B01023F00140000080800FB1294A
E64036D40F0000000D0000000500240000000000000
2000000000000000312E7478740A00200000000000010
01800EDEFA1FD816AD201B350D4E8816AD201B350D4E
8816AD201504B0506000000000100010057000000320 0
00000000
```

很明显是zip格式的数据

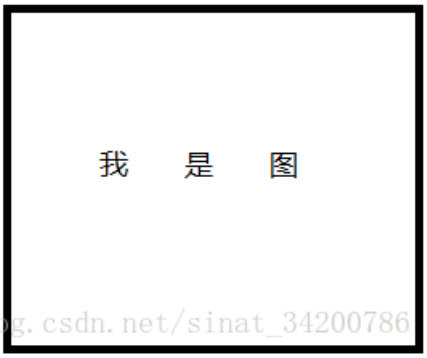
WriteUp

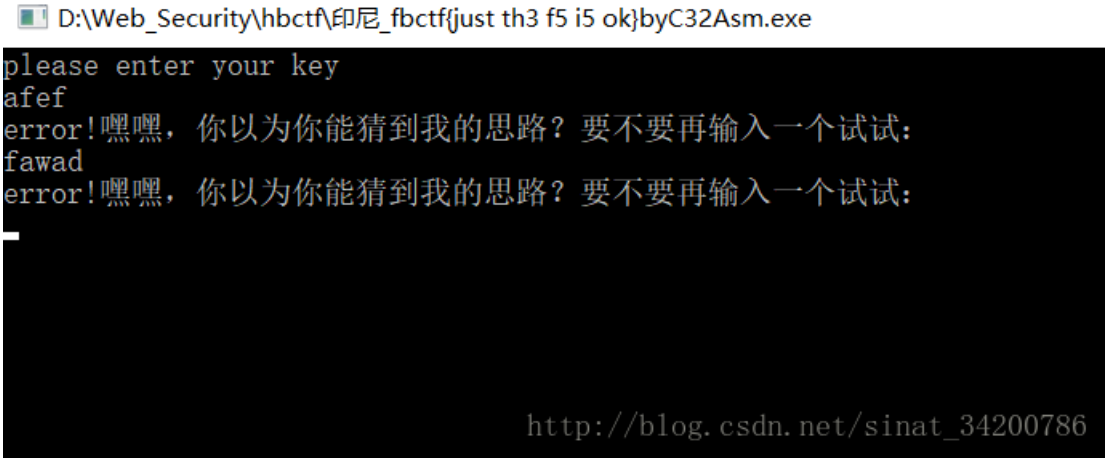
HxD新建文件，将数字串复制进去，保存

保存的文件扩展名改为 .zip，打开得flag

印尼_find_the_key

原题



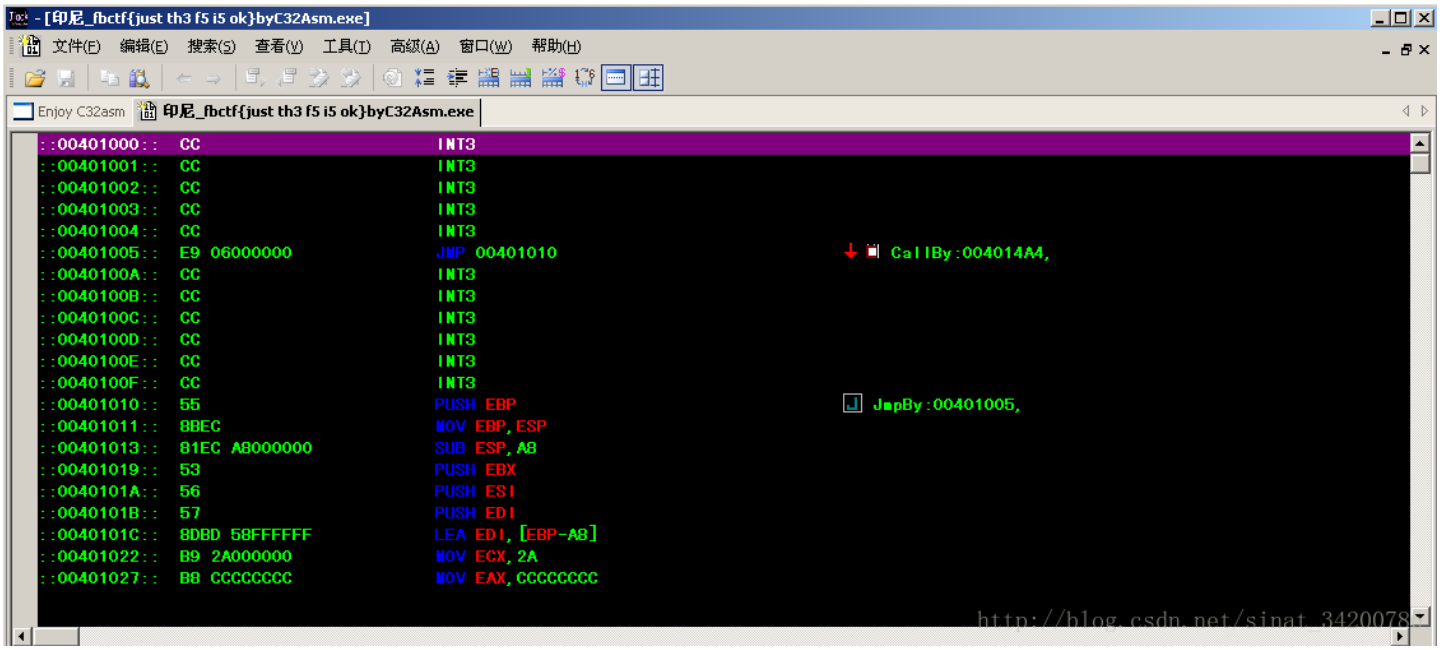


解题思路

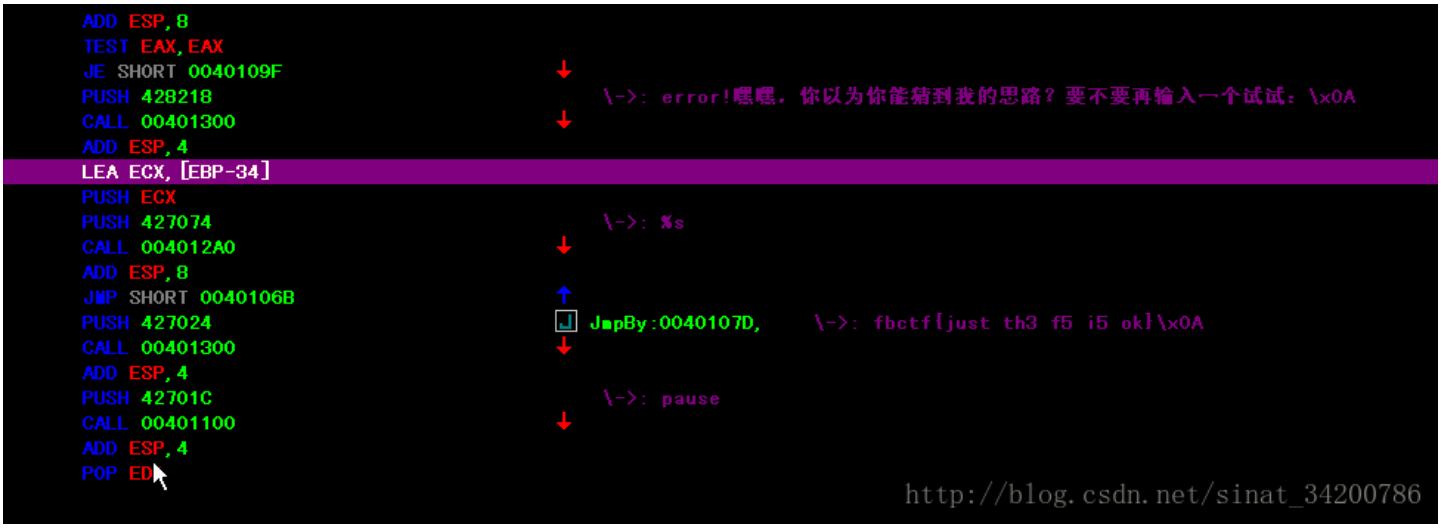
输入正确的字符串才会输出flag，在没有源代码的情况下谁知道（出题人知道）要输入什么？所以关键不在输入。想一下可以知道fla

WriteUp

将exe拖到C32Asm里面



往下找到flag即可



安哥拉_一步一步

原题

我　是　图

解题思路

整个GIF看完发现其中某帧很特别，那就一帧一帧的找到特别那帧

WriteUp

拖入PS中一帧一帧铺开,找到特别那帧即可

北韩_经过多少次加密

原题

解题思路

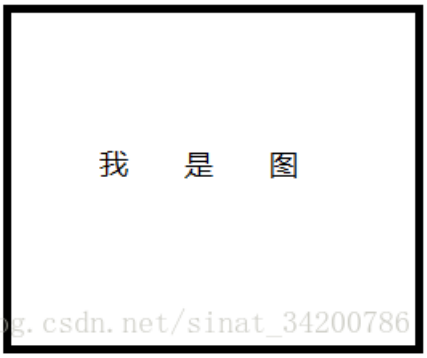
```
很明显是base64加密，递归解密
```

WriteUp

```
import base64
import re

file = open('1.txt','r')
str1 = file.read()
str1 = base64.b64decode(str1)
str2 = str(str1,'utf-8')

while re.match('fbctf',str2)==None :
    count = len(str2)%4          #此处有个小坑，python进行base64解码时要在末尾补齐 = 号
    if count!=0:
        str2 += ("="*(4-count))
    str1 = base64.b64decode(str2)
    str2 = str(str1, 'utf-8')

print(str2)
```

日本_hundouluo

原题



解题思路

```
模拟器运行以下，说不定要通关才有flag呢？
```

WriteUp

```
拖入模拟器中，你说那个是不是flag？
```

伊朗_爆破

原题

伊朗_爆破_2016.zip - ZIP 压缩文件, 解包大小为 4 字节

名称	大小	压缩后大小	类型	修改时间	CRC32
..			本地磁盘		
flag4位数字.txt *	4	16	文本文档	2016/12/22 2...	56EA988D

解题思路

```
就按照题目的意思办吧，爆破CRC32
```

WriteUp

```
import binascii
real = 0x56EA988D
for y in range(1000,9999):
    if real == (binascii.crc32(bytes(str(y),'utf-8')) & 0xffffffff):
    print(y)
print('End')
```

利比亚_密码多简单

原题

解题思路

```
这三只松鼠一看就不是好鼠，居然聚众赌博。。。
还是先HxD看下，不是FF D9结尾而且还有莫名其妙的字符串？
```

WriteUp

```
HxD发现异常，可以确定图片包含多个文件
```

```
000103C0  AA 41 8F 64 84 D2 65 85 15 2B 0F FE DD F4 0C C4  ªA.d„Òe….+.þÝô.Ä
000103D0  3A A1 AC 74 7F 2D 69 41 49 B6 33 85 C0 04 51 4F  :¡¬t.-iAI¶3…À.QO
000103E0  78 F2 87 88 13 AD 1F 9A F2 EE 8B FF A7 C0 75 50  xò‡ˆ...šòî‹ÿ§ÀuP
000103F0  4B 01 02 3F 00 14 00 01 08 08 00 ED 62 96 49 79  K..?.......íb–Iy
00010400  94 EA 6F B2 B8 00 00 E5 C3 00 00 0A 00 24 00 00  ”êo²¸..åÃ....$..
00010410  00 00 00 00 00 20 00 00 00 00 00 00 00 6D 65 69  ..... .......mei
00010420  7A 68 69 2E 6A 70 67 0A 00 20 00 00 00 00 00 01  zhi.jpg.. ......
00010430  00 18 00 DD 2A 96 23 0B 5C D2 01 C3 92 93 23 0B  ...Ý*–#.\Ò.Ã’“#.
00010440  5C D2 01 7C 8F 6F 23 0B 5C D2 01 50 4B 05 06 00  \Ò.|.o#.\Ò.PK...
00010450  00 00 00 01 00 01 00 5C 00 00 00 DA B8 00 00 00  .......\...Ú¸...
00010460  00                                               .
```


Binwalk分析

```
root@kali: ~
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)
root@kali:~# binwakl 2.jpg
bash: binwakl: 未找到命令
root@kali:~# binwalk 2.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
66635         0x1044B         End of Zip archive

root@kali:~#
```

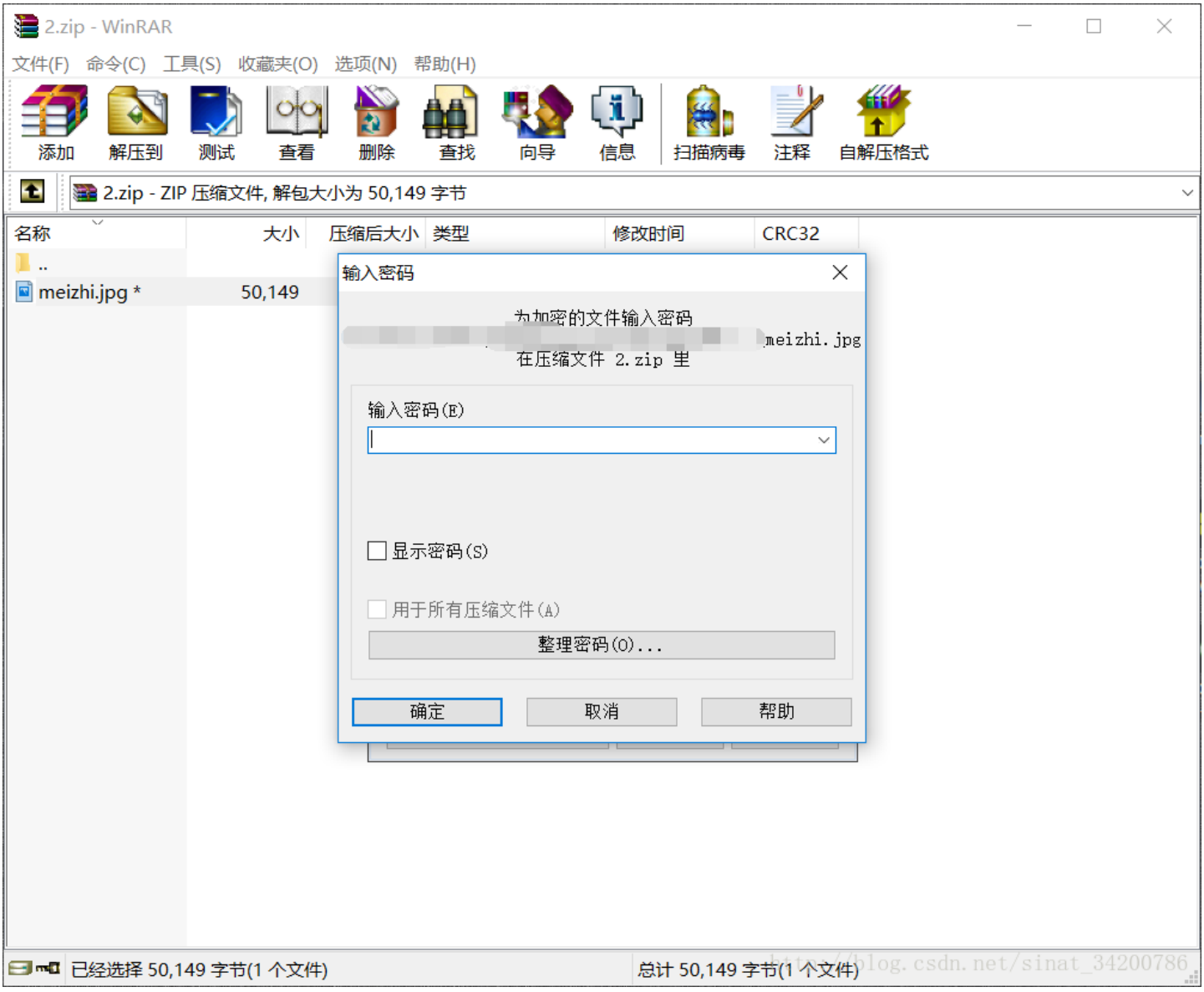

发现包含Zip文件，不过文件头缺失，补上去看看

文件头缺失很可能是50 4B的缺失，那么搜索14 00看看哪里的50 4B不见了

```
00004AC0  C8 D6 36 34 EC 28 B2 58 C9 5C 85 FB DB 93 CE 6B  ÈÖ64ì(²XÉ\…ûÛ"Îk
00004AD0  82 CD 64 24 01 95 FE 5C F6 AB 52 57 96 3D 2E 41  ‚Íd$.•þ\ö«RW–=.A
00004AE0  0B 9C 6C 29 8D AA 28 8C 9C 76 A7 44 58 B2 4E 9E  .œl).ª(Œœv§DX²Nž
00004AF0  34 93 E1 96 3B 00 DA AB D4 DE 46 30 C4 1D 36 6A  4"á–;.Ú«ÔÞF0Ä.6j
00004B00  F5 3A 11 FF D9 6B 65 79 3A 6D 69 6D 61 64 75 6F  õ:.ÿÙkey:mimaduo
00004B10  6A 69 61 6E 64 61 6E 03 04 14 00 01 08 08 00 ED  jiandan.....í
00004B20  62 96 49 79 94 EA 6F B2 B8 00 00 E5 C3 00 00 0A  b–Iy"êo²¸..åÃ...
00004B30  00 00 00 6D 65 69 7A 68 69 2E 6A 70 67 EB 9C 47  ...meizhi.jpgëœG
00004B40  55 0A 95 59 08 8E F9 68 23 EE 24 7D D7 05 A2 68  U.•Y.Žùh#î$}×.¢h
00004B50  BE BB 6A 16 4B 53 82 51 B8 02 2D D9 C6 B8 8D E1  ¾»j.KS‚Q¸.-ÙÆ¸.á
00004B60  AB 93 30 1E D7 BB BC 50 85 15 C0 52 30 9A 64 81  «"0.×»¼P….ÀR0šd.
00004B70  8E 25 B7 81 7A 64 D7 A2 B9 1F 8A EB B7 29 FD F4  Ž%·.zd×¢¹.Šë·)ýô
```


发现一个key：mimaduojiandan 并且14 00 前面的 50 4B被替换成61 6E

从被替换处往下全选，新建文件粘贴进去然后保存，改后缀为Zip，打开



密码很明显就是刚才那个key

越南_真的加密了？

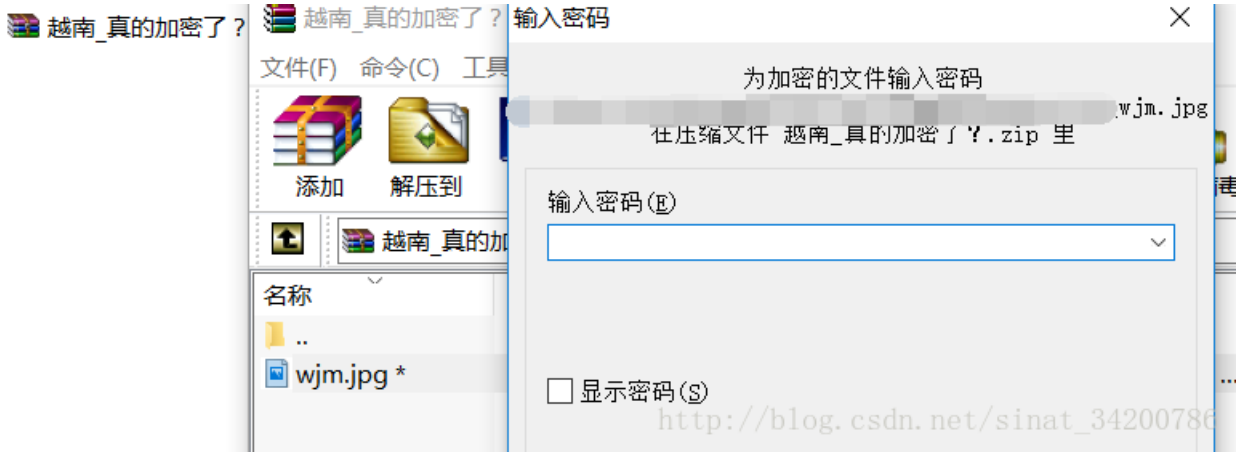
原题

解题思路

```
懒得放到HxD里面了，直接改后缀为Zip看看吧，看题目名就像是伪加密
```

WriteUp

```
改后缀后发现有jpg并且解压需要密码，直接按伪加密解决
```




```
00078270  5D 05 9A D1 90 4D 22 FB 7B BF 75 F6 58 4D 6D 37  ].šÑ.M"û{¿uöXMm7
00078280  BD 82 AC 46 BD BF 8E BD E2 AB A3 F3 D7 C2 0E B4  ½‚¬F½¿Ž½â«£ó×Â.´
00078290  14 04 81 17 E3 FF 0F 50 4B 01 02 3F 00 14 00 00  ....ãÿ.PK..?....
000782A0  08 08 00 8F 61 95 49 86 42 0F 2A 0C 5A 01 00 71  ....a•I†B.*.Z..q
000782B0  61 01 00 07 00 24 00 00 00 00 00 00 00 20 00 00  a....$....... ..
000782C0  00 00 00 00 00 77 6A 6D 2E 6A 70 67 0A 00 20 00  .....wjm.jpg.. .
000782D0  00 00 00 00 01 00 18 00 C4 D4 F3 71 40 5B D2 01  ........ÄÔóq@[Ò.
000782E0  28 6A EF 2D 40 5B D2 01 F1 6F 72 35 40 5B D2 01  (jï-@[Ò.ñor5@[Ò.
000782F0  50 4B 05 06 00 00 00 00 01 00 01 00 59 00 00 00  PK..........Y...
00078300  31 5A 01 00 00 00                               1Z....
```


提取图片出来即可

埃塞俄比亚-隐写术

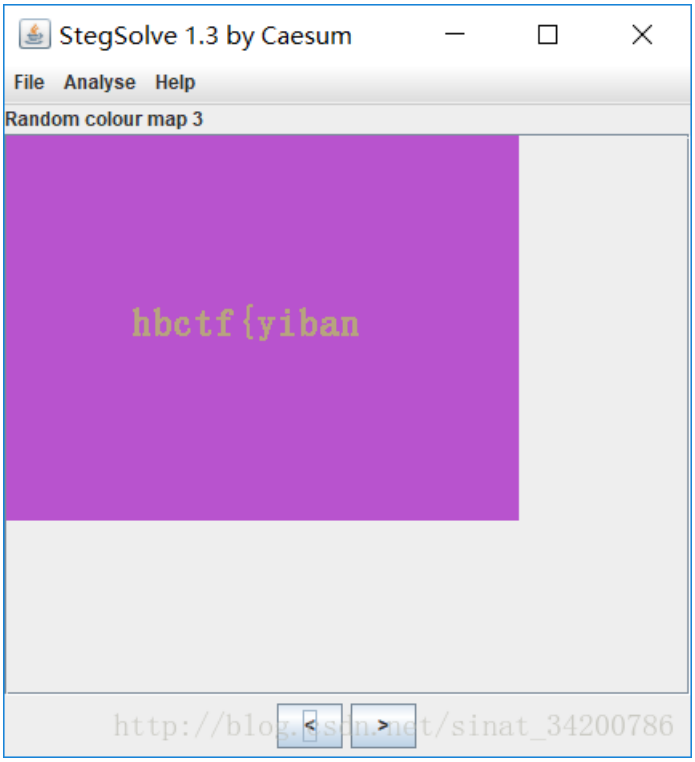
原题

解题思路

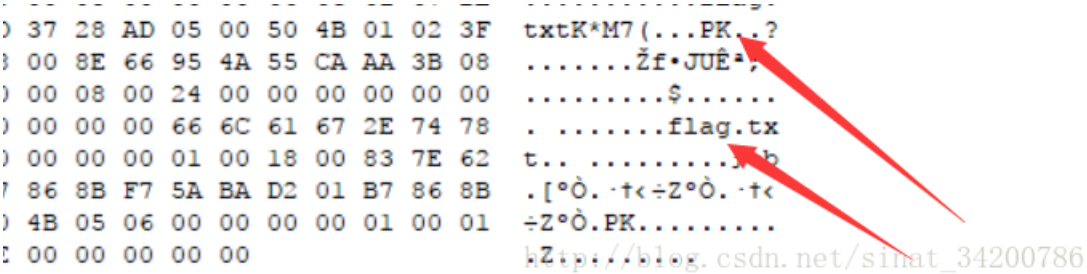
```
Stegsolve 和 HxD一顿操作打完收工
```

WirteUp

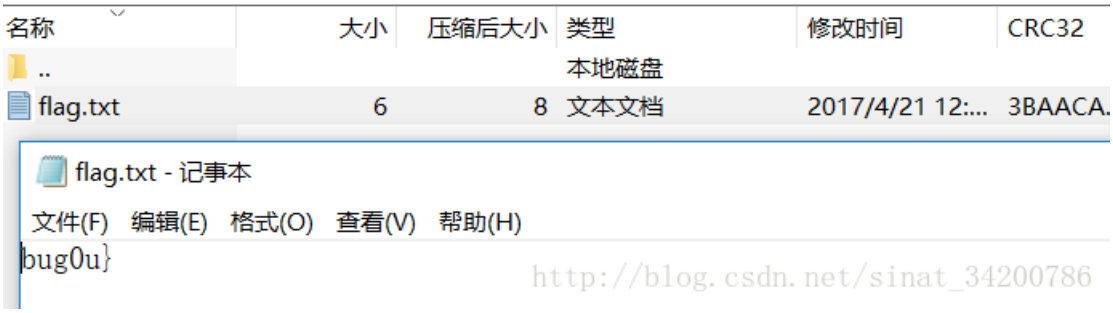
```
先用Stegsolve打开，左看看右看看(左右翻页)，抓到半个flag(半个？)
```




```
flag都提示是一半了，再仍HxD里看看好了
```

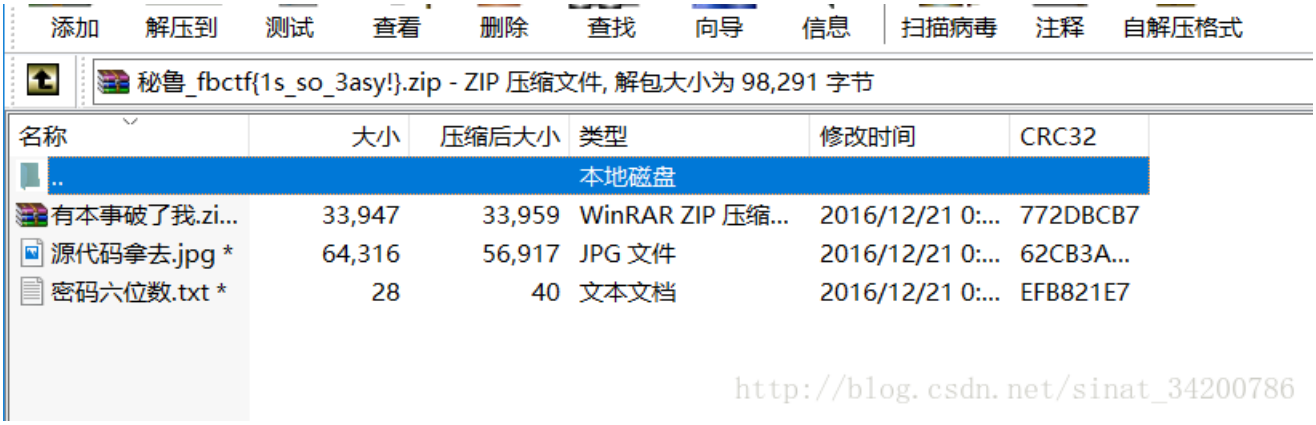


发现压缩文件的标志和flag.txt，那好，改后缀为.zip打开，拿到另一半flag



秘鲁_源代码给你又如何

原题



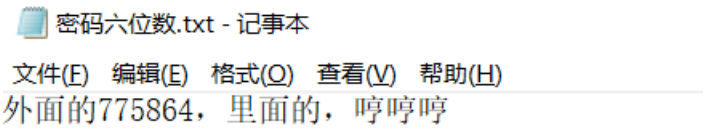
解题思路

先爆破Zip密码，拿到源代码再分析要输入的字符串（真的是这样？）

WriteUp

Ziperello爆破密码，纯数字秒破

把能看的都看看

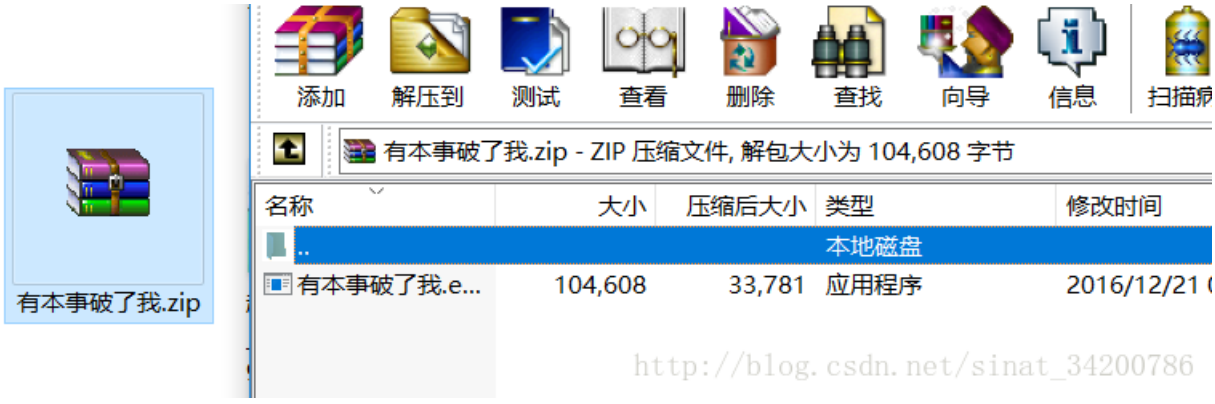


果然没有那么简单的，再看看源代码的图片

```c
#include "stdio.h"
#include "windows.h"

void challenge(char* str)
{
    char temp[9]={NULL};
    strncpy(temp,str,8);
    printf("temp=%s\n",temp);
    if(strcmp(temp,"please!@")==0)
    {
        printf("fbctf                    ");
    }
}
int main(int argc,char* argv[])
{
    char buf[8];
    int check=1;
    char buf2[80];

    strcpy(buf2,"give me flag!");
    strcpy(buf,argv[1]);
    printf("Value of Check:%d\n",check);
    printf("buf2 :%s\n",buf2);

    if(check ==65)
    {
        challenge(buf2);
    }
    else
    {
        printf("Check is not 65\n %d Program terminated!!\n",check);
    }
    return 0;
}
```


分析源代码可以知道，buf,check和buf2在堆栈中的空间是连续的。利用strcpy（）的长度漏洞可以用输入参数覆盖后面的check和

前面的分析很有道理的样子，可是我怎么拿到压缩包里的程序呢？



一开始解压出来三个文件，怎么看都是那张图片比较可疑，扔进HxD里看看

```
0000FAC0  45 14 00 51 45 14 00 51 45 14 00 51 45 14 00 51  E..QE..QE..QE..Q
0000FAD0  45 14 00 51 45 14 00 51 45 14 00 51 45 14 00 51  E..QE..QE..QE..Q
0000FAE0  45 14 00 51 45 14 00 51 45 14 00 51 45 14 00 51  E..QE..QE..QE..Q
0000FAF0  45 14 00 51 45 14 00 51 45 14 00 51 45 14 00 51  E..QE..QE..QE..Q
0000FB00  45 14 00 51 45 14 00 51 45 14 00 51 45 14 00 51  E..QE..QE..QE..Q
0000FB10  45 14 00 51 45 14 00 51 45 14 00 51 45 14 00 51  E..QE..QE..QE..Q
0000FB20  45 14 00 51 45 14 00 51 45 14 00 51 6B 65 79 3A  E..QE..QE..Qkey:
0000FB30  6A 69 75 73 68 69 6B 65 6E 6E 69 61              jiushikennia
```


果然吧，顺利拿到密码解压出程序，随便输入下看看

```
D:\Web_Security\hbctf>1.exe abcdefghA
Value of Check:65
buf2 :give me flag!
temp=give me

D:\Web_Security\hbctf>
```


可以发现确实是覆盖了check，那么这样把buf2也覆盖就ok了不是么？试试

```
D:\Web_Security\hbctf>1.exe abcdefghAplease!@
Value of Check:1701605441
buf2 :give me flag!
Check is not 65
 1701605441 Program terminated!!

D:\Web_Security\hbctf>
```

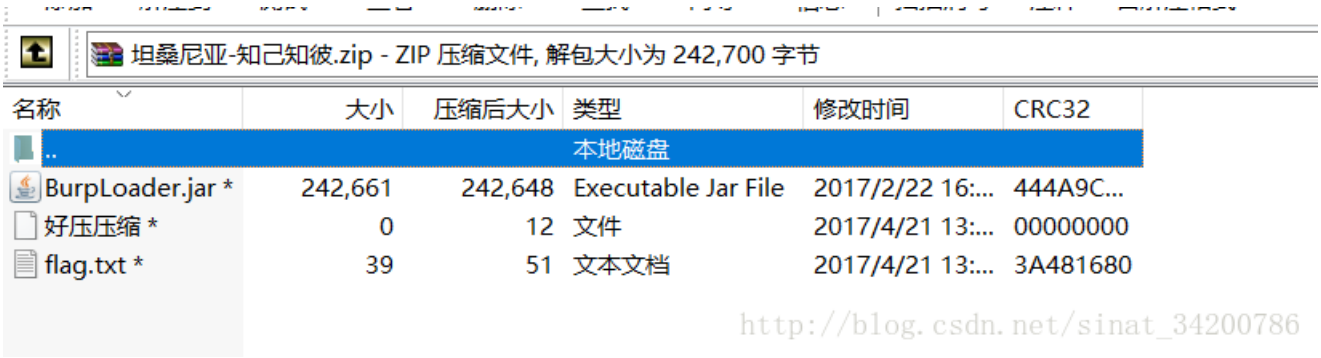

这个时候你发现问题了么？如果没有就再想想

问题在于int 是4个字节而char 是1个字节，就是说A只是占了check的最低位字节还有三个字节需要用0填充，但是你输入不了ASCII

所以老办法，扔进C32Asm查找字符串

坦桑尼亚-知己知彼

原题



解题思路

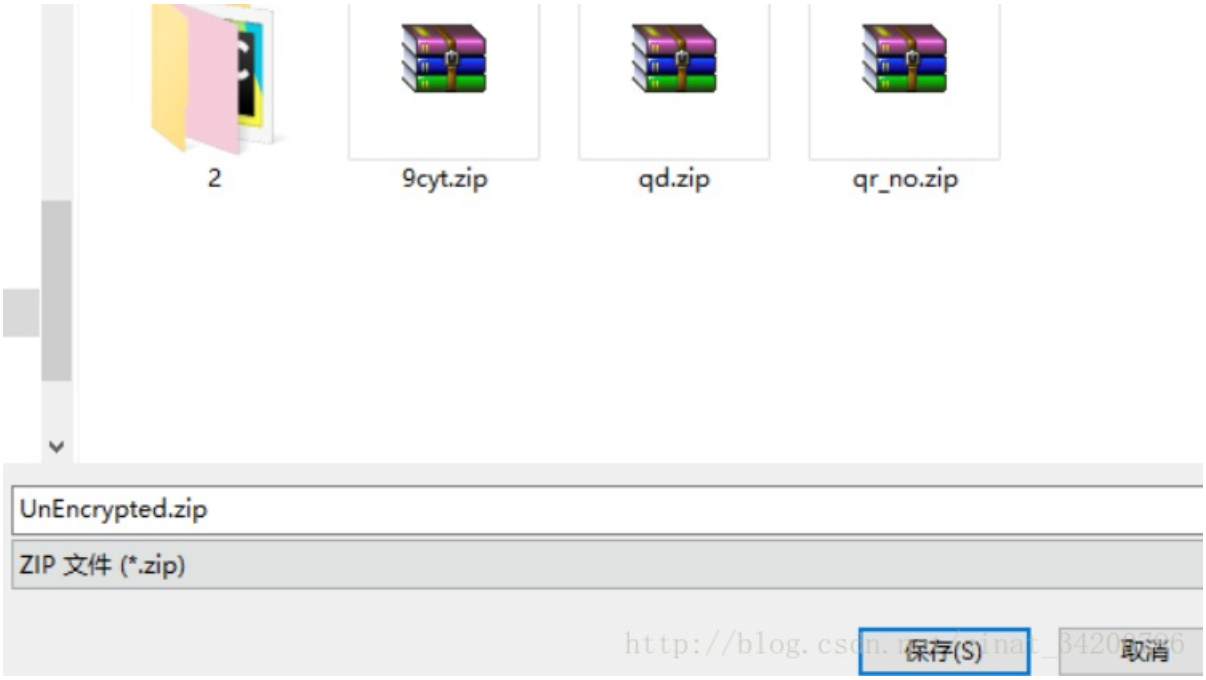
```
根据提示猜想是明文攻击
```

WriteUp

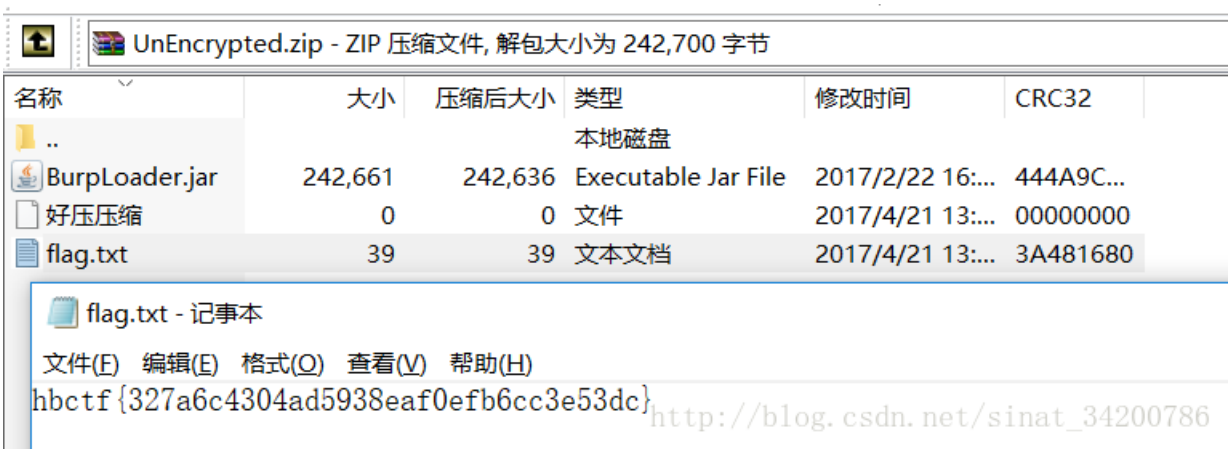
```
利用AZPR进行明文攻击,先根据压缩包内的提示将BurpLoader.jar用好压压缩。然后将原文件，好压压缩文件拖入AZPR相应位置，选
```

提示没有找到密码

不管，直接确定保存



直接打开UnEncrypted.zip内的flag.txt即可



涨姿势点

明文攻击的方式，大概的原理和应用

备注

为什么要用好压压缩才行呢？（winrar亲测不行）