攻防世界 MISC 新手练习区 writeup 007-012

原创

ChaoYue_miku 于 2021-09-04 23:53:15 发布 484 收藏 2

分类专栏： CTF # 攻防世界 # Misc 文章标签： python

本文链接：https://blog.csdn.net/ChaoYue_miku/article/details/120107907

CTF 同时被 3 个专栏收录

127 篇文章 5 订阅

订阅专栏

攻防世界

6 篇文章 0 订阅

订阅专栏

Misc

2 篇文章 0 订阅

订阅专栏

攻防世界 MISC 新手练习区 题目解答

文章目录

007 gif

难度系数： 4.0

题目来源： 暂无

题目描述： 菜狗截获了一张菜鸡发给菜猫的动态图，却发现另有玄机

0x01 下载附件,打开后发现104张jpg图片文件

图片只有两种，一种黑色方块，一种白色方块，而且104张是8的倍数，猜想可能是二进制编码

0x02 编写python解码脚本

将图片转换为二进制01字符串，然后再以一个字节（即8bit）为长度分割后转换为字符串

注意要把104张图片放到python脚本的同级目录中

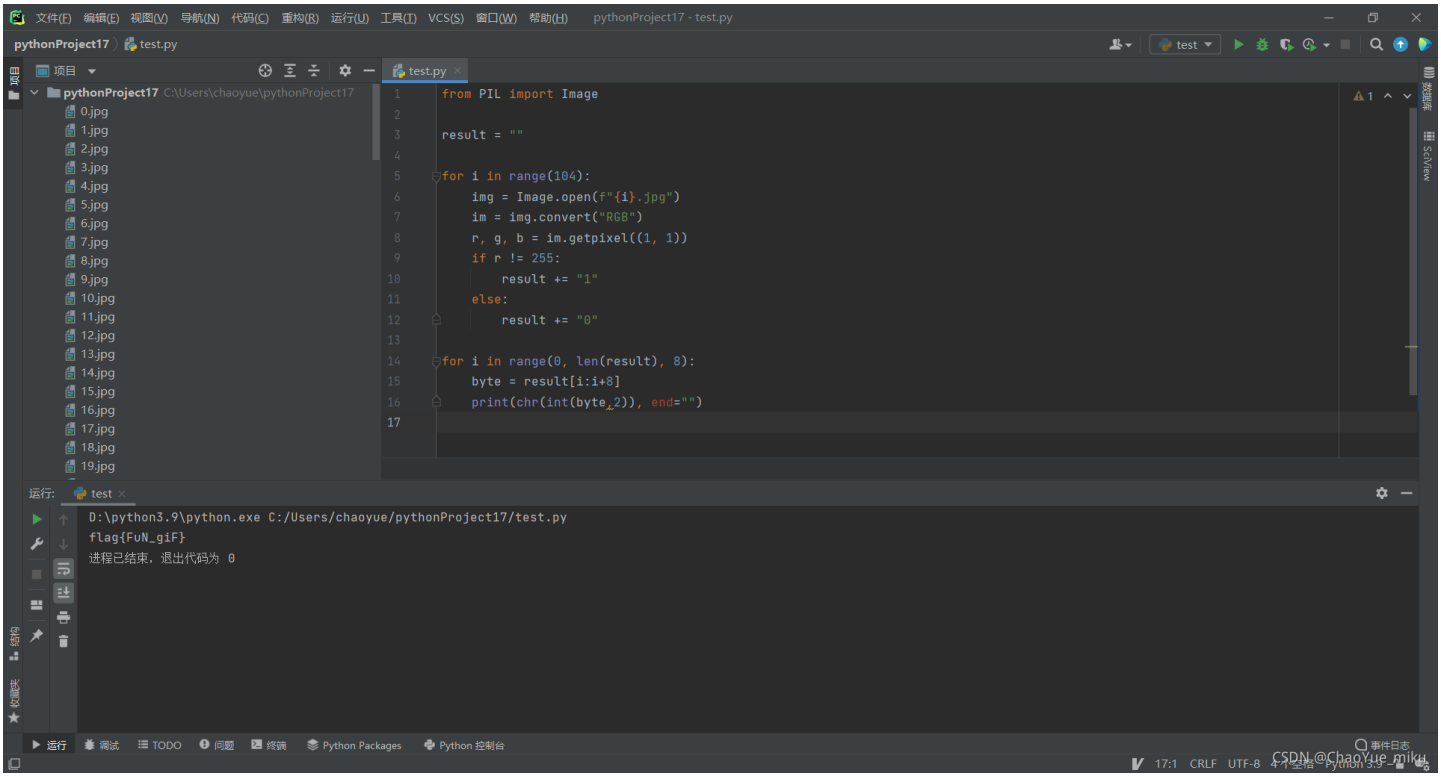
```
from PIL import Image

result = ""

for i in range(104):
    img = Image.open(f"{i}.jpg")
    im = img.convert("RGB")
    r, g, b = im.getpixel((1, 1))
    if r != 255:
        result += "1"
    else:
        result += "0"

for i in range(0, len(result), 8):
    byte = result[i:i+8]
    print(chr(int(byte,2)), end="")
```

运行结果如下：



0x03 得到flag： flag{FuN_giF}

008 掀桌子

难度系数：4.0

题目来源： DDCTF2018

题目描述： 菜狗截获了一份报文如下

c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfaebe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2，生气地掀翻了桌子(ノ °□°）ノ ︵ ┻━┻

0x01 该题目没有附件，只有题目描述中的报文

观察后发现，字符串均由0-9已经a-e组成，是十六进制字符串。

可以将他们每两个字符划为一组，转换为十进制，然后减去128，使其落入Ascii码0-127的范围内，再转换成字符串。

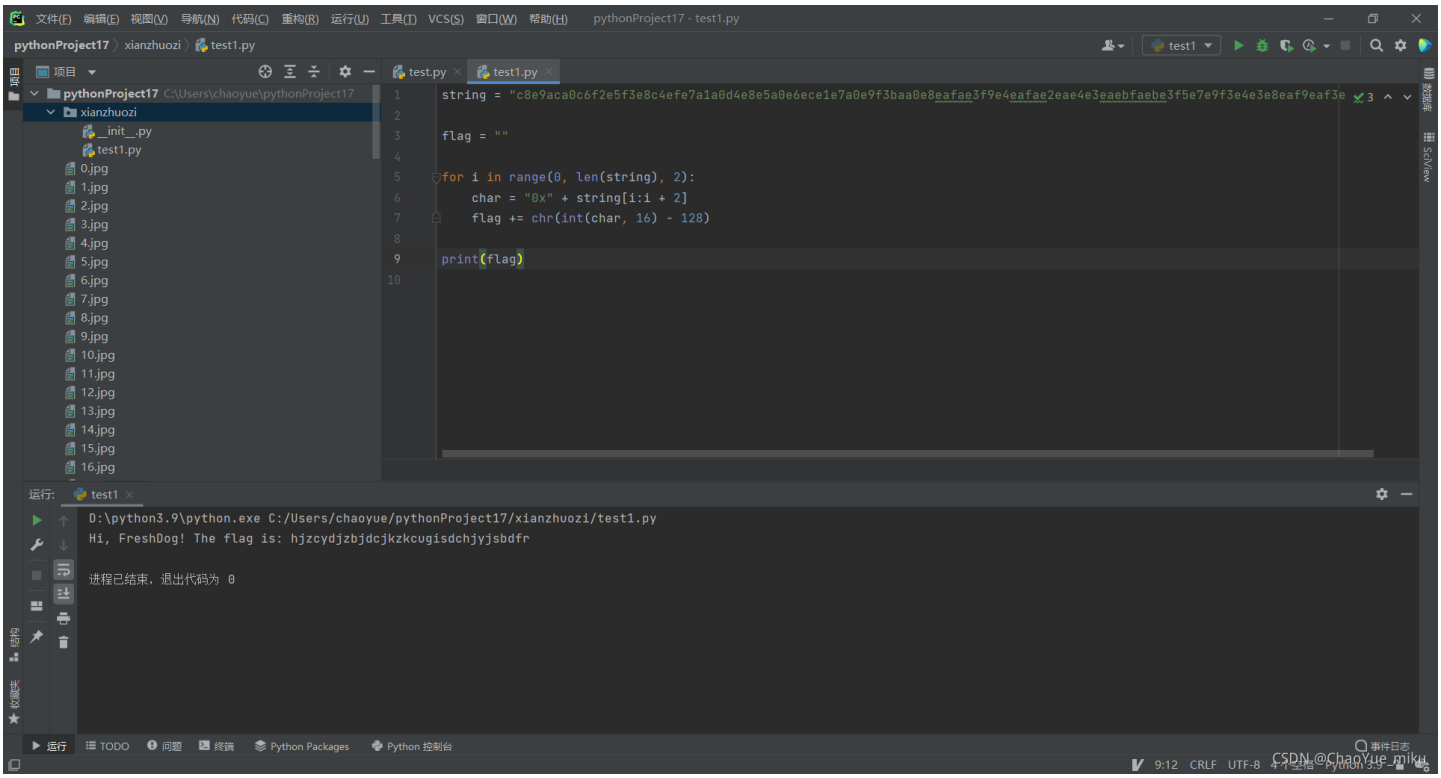
0x02 编写python解码脚本

```
string = "c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfaebe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2"

flag = ""

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

print(flag)
```



输出结果如下：

```
Hi, FreshDog! The flag is: hjzcydjzbjdcjkzkcugisdchjyjsbdfr
```

0x03 得到flag： **flag{hjzcydjzbjdcjkzkcugisdchjyjsbdfr}**

009 ext3

难度系数： 5.0
题目来源： bugku
题目描述： 今天是菜狗的生日，他收到了一个linux系统光盘

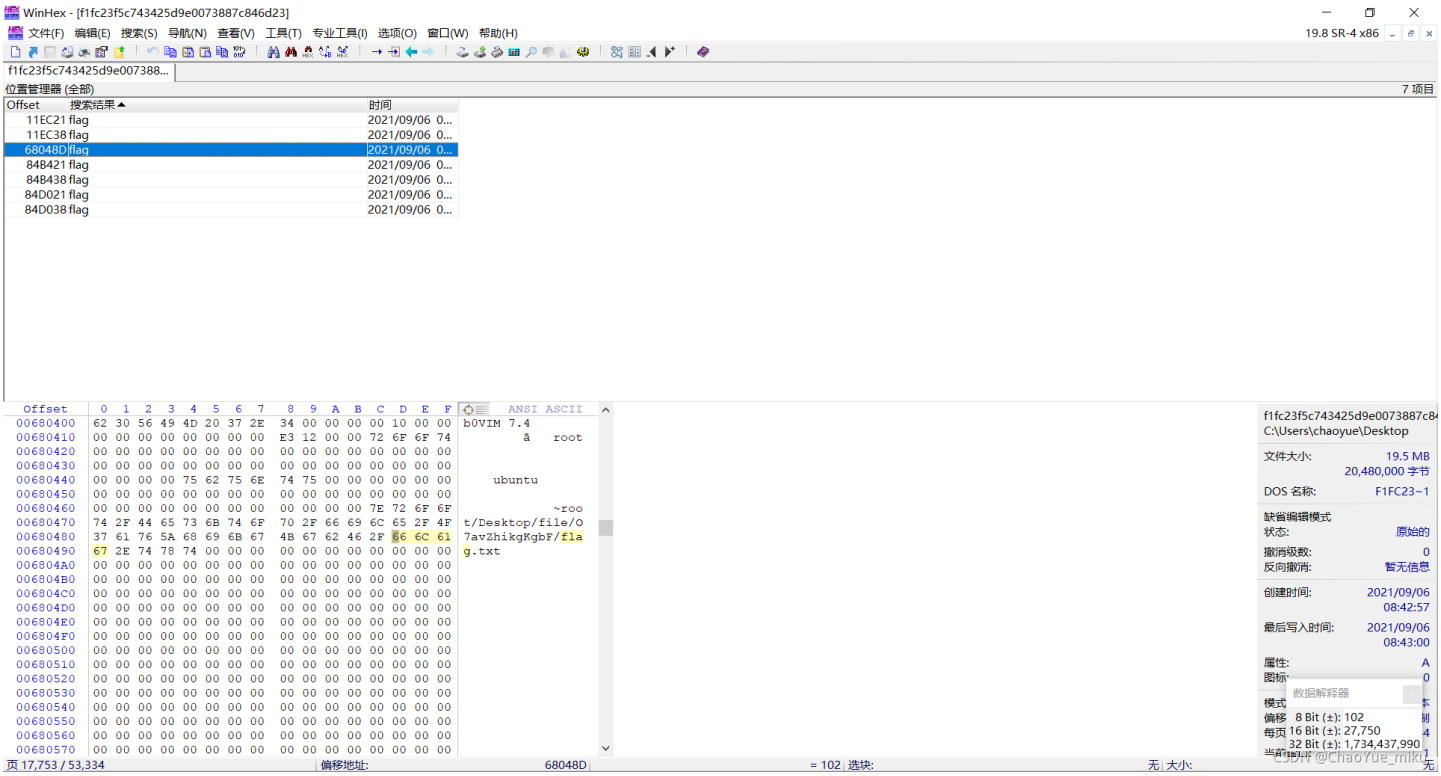
关于题目中的ext3：

`EXT3是第三代扩展文件系统（英语：Third extended filesystem，缩写为ext3），是一个日志文件系统，常用于Linux操作系统。`

法一：

0x01 下载附件,在win10中无法直接打开，使用winhex打开文件

查找带有flag的字符串

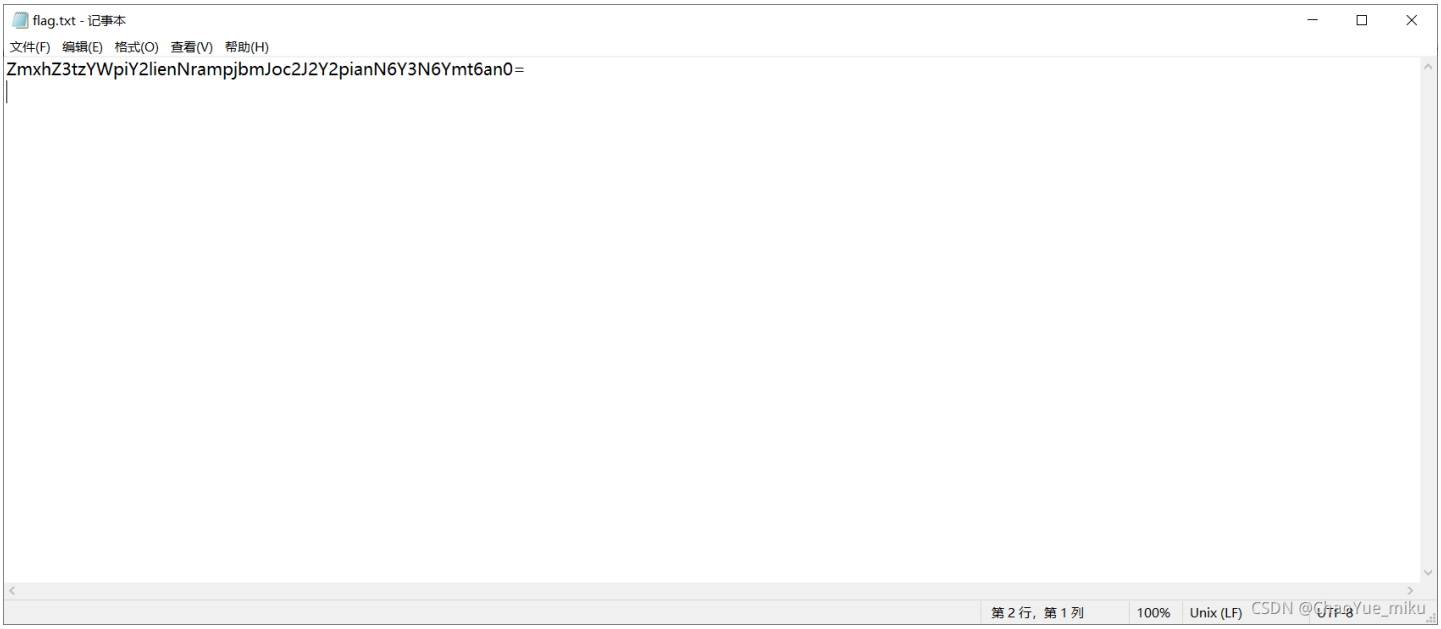


发现了flag.txt文件，以及该文件的路径

`~root/Desktop/file/O7avZhikgKgbF/flag.txt`

0x02 使用360压缩打开文件

根据刚才的路径打开flag.txt文件



`ZmxhZ3tzYWpiY2lienNrampjbmJoc2J2Y2pianN6Y3N6Ymt6an0=`

是一串base64加密字符串

0x03 进行base64解密

0x04 得到flag： flag{sajbcibzskjjcnbhsbvcjbjszcszbkzj}

法二：

0x01 下载附件,在Kali Linux系统下挂载文件

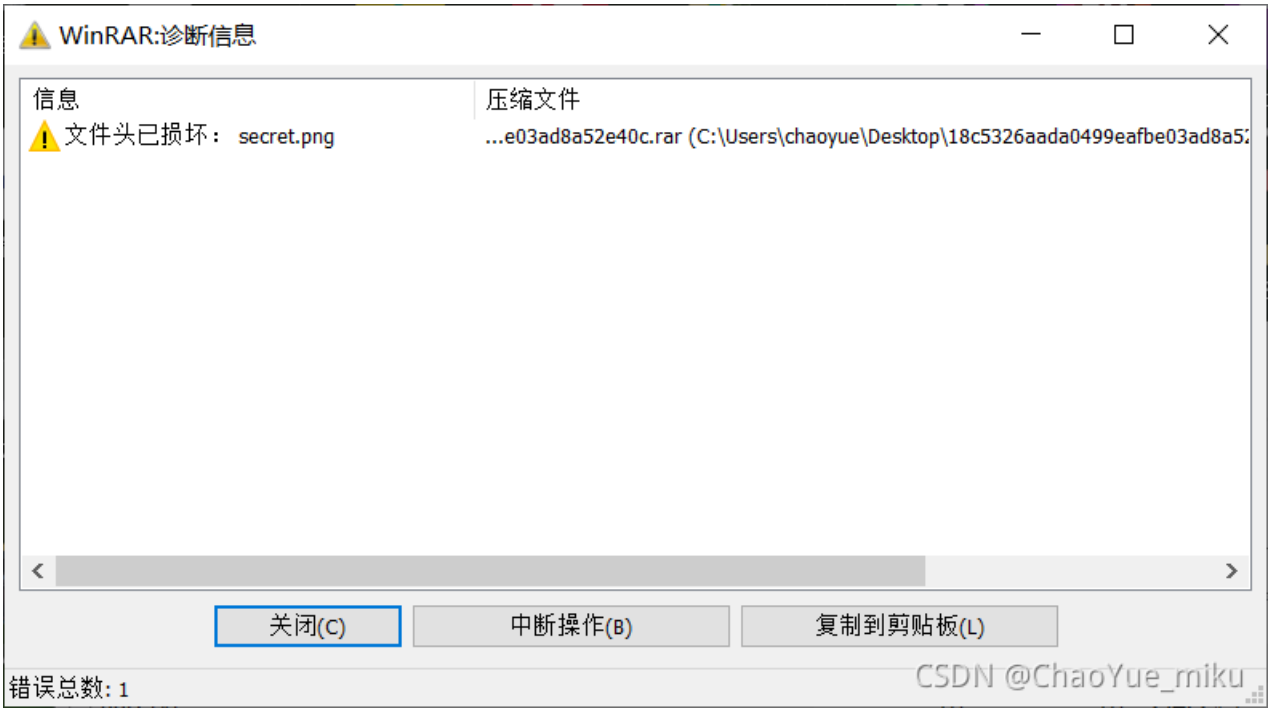
0x02 查找有关flag的文件，发现后直接base64解密得出flag

0x03 得到flag： **flag{sajbcibzskjjcnbhsbvcjbjszcszbkzj}**

010 SimpleRAR

难度系数：5.0
题目来源： 08067CTF
题目描述： 菜狗最近学会了拼图，这是他刚拼好的，可是却搞错了一块(ps:双图层)

0x01 下载附件,是一个rar压缩包文件，打开时提示文件头已损坏



0x02 使用010 Editor打开文件，修复文件头

Tips：

每一个块都是由以下域开始的：【译者注：即每一个块的头部都是由以下域（可称之为头域）组成的】

HEAD_CRC 2 bytes CRC of total block or block part

整个块或者块某个部分的CRC（根据块类型而有不同）

HEAD_TYPE 1 byte Block type

块类型【译者注：也可以理解为块头部类型，因为不同的块对应不同的块头部。后文也经常混淆这两种概念。】

已经声明过的块类型包括：

HEAD_TYPE=0x72 marker block【译者注：有些文献里也称之为MARK_HEAD】

标志块【译者注：一个固定为0x52 61 72 21 1A 07 00的7字节序列】

HEAD_TYPE=0x73 archive header【译者注：有些文献里也称之为MAIN_HEAD】

归档头部块

HEAD_TYPE=0x74 file header【译者注：有些文献里也称之为FILE_HEAD】

文件块【译者注：直译为文件头部，但是此处的类型应该指的是整个块的类型，而非块头部结构的类型，因此感觉称之为文件块更合适。】

HEAD_TYPE=0x75 old style comment header

老风格的 注释块【译者注：直译为注释头部，基于和文件块一样的原因，感觉称之为注释块更合适】

HEAD_TYPE=0x76 old style authenticity information

老风格的 授权信息块/用户身份信息块

HEAD_TYPE=0x77 old style subblock

老风格的 子块

HEAD_TYPE=0x78 old style recovery record

老风格的 恢复记录块

HEAD_TYPE=0x79 old style authenticity information

老风格的 授权信息块/用户身份信息块

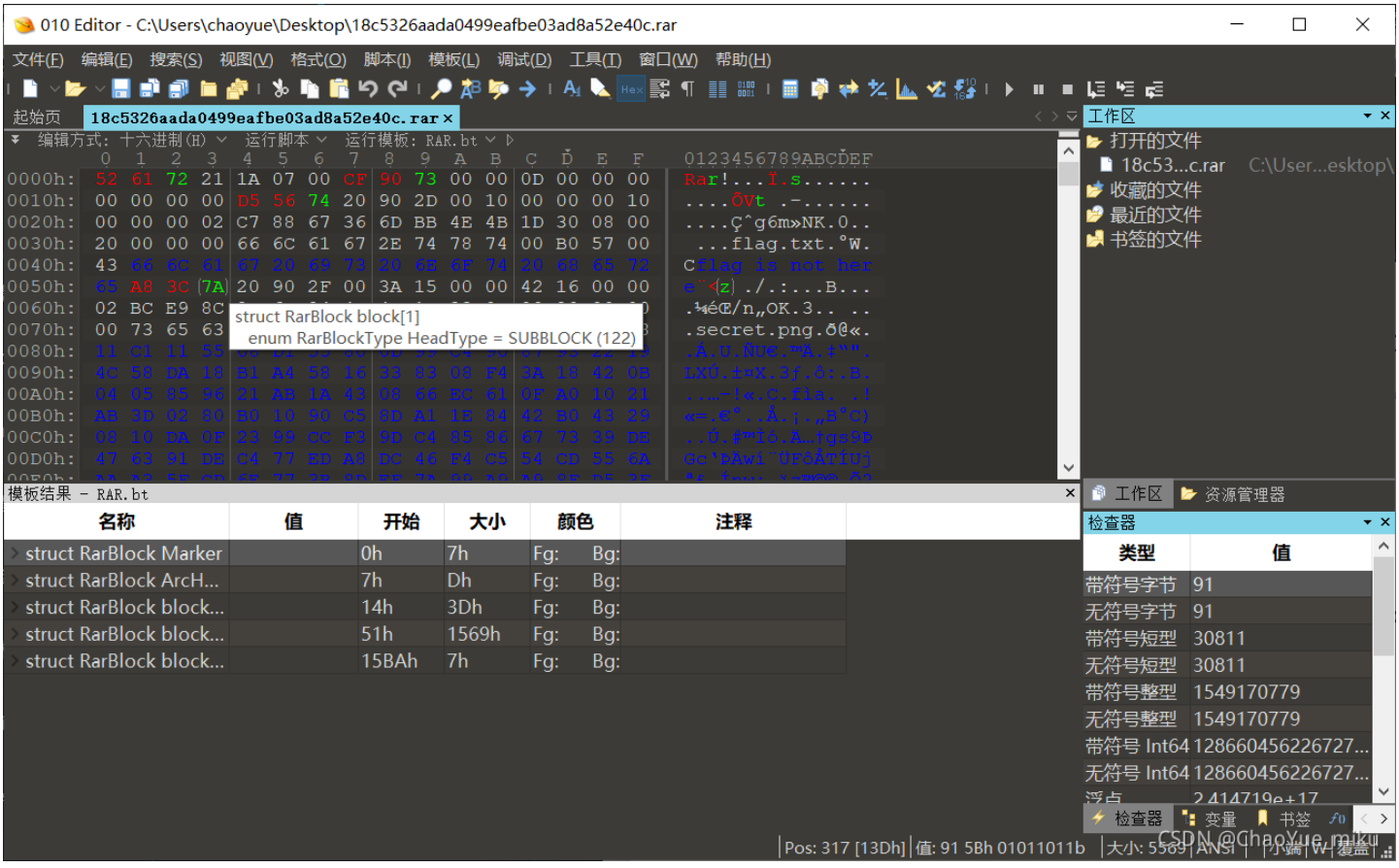
HEAD_TYPE=0x7a subblock

子块

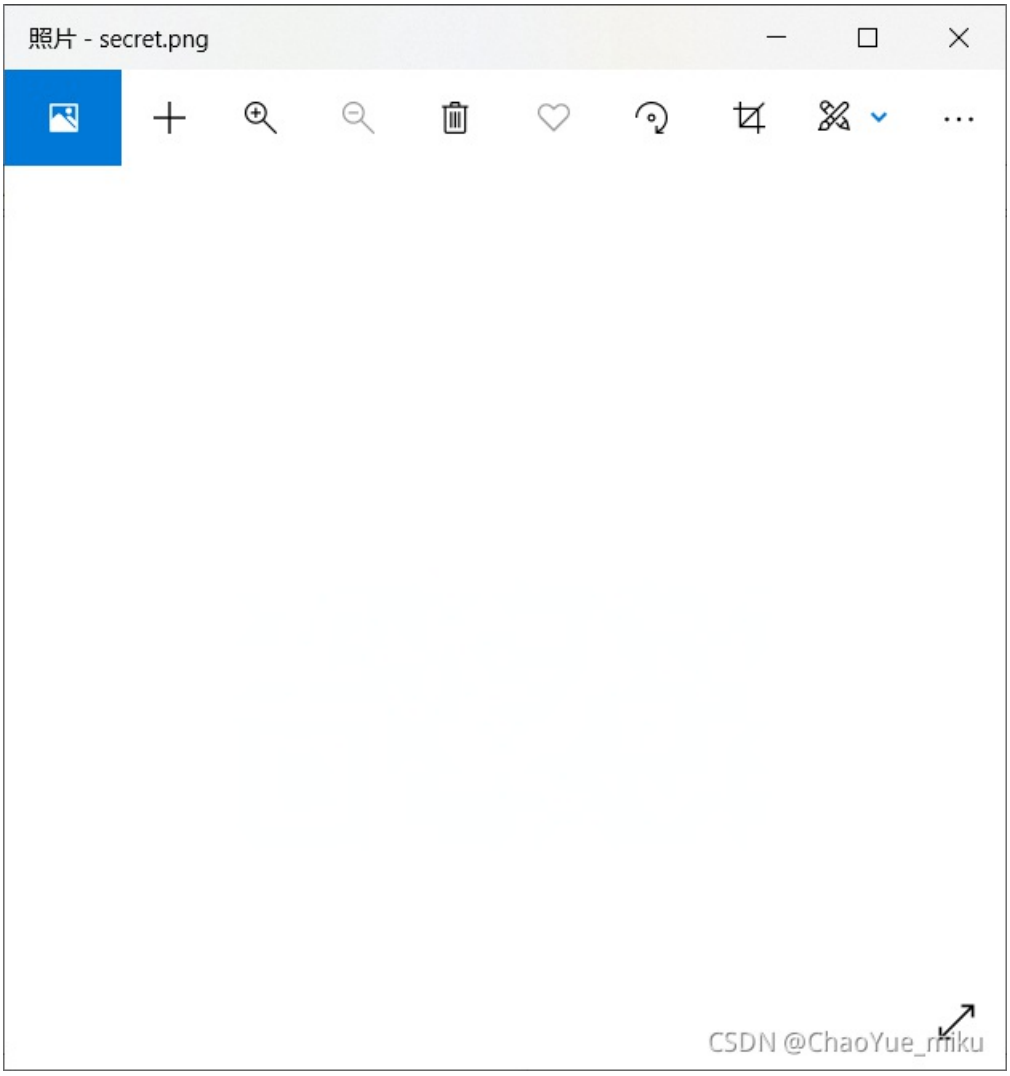
HEAD_TYPE=0x7b end block

结束块【译者注：一个固定为0xC4 3D 7B 00 40 07 00的7字节序列】

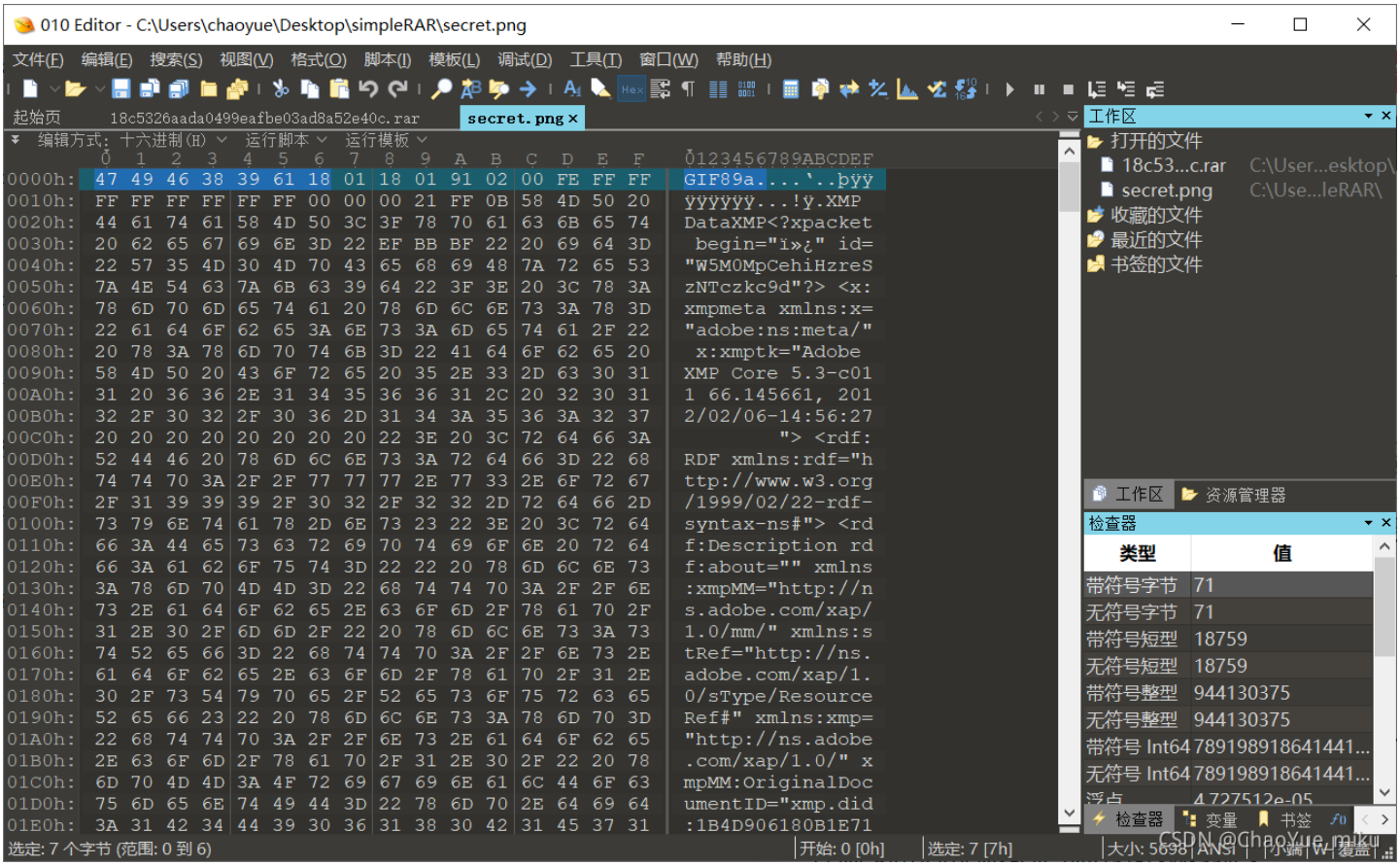
将 `A8 3C 7A` 改为 `A8 3C 74`



0x03 修改后成功打开压缩包，发现serect.png文件


照片 - secret.png
CSDN @ChaoYue_miku

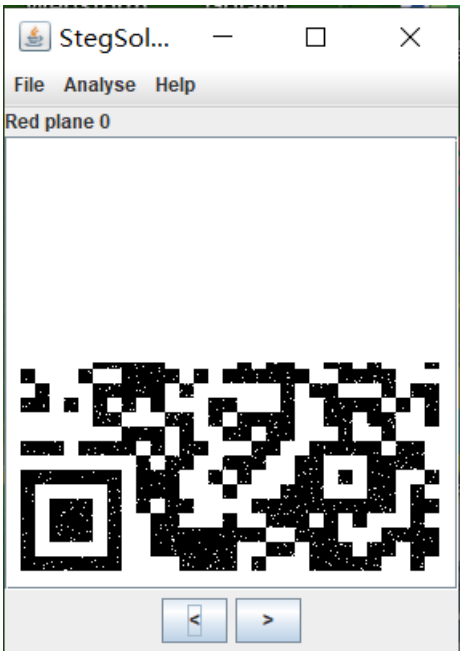
打开后一片空白，放到010 Editor（WinHex也可以）中查看一下



发现了gif文件头

0x04 将图片后缀名改为.gif，用StegoSolve打开

查看不同通道，发现了一张二维码的下半部分

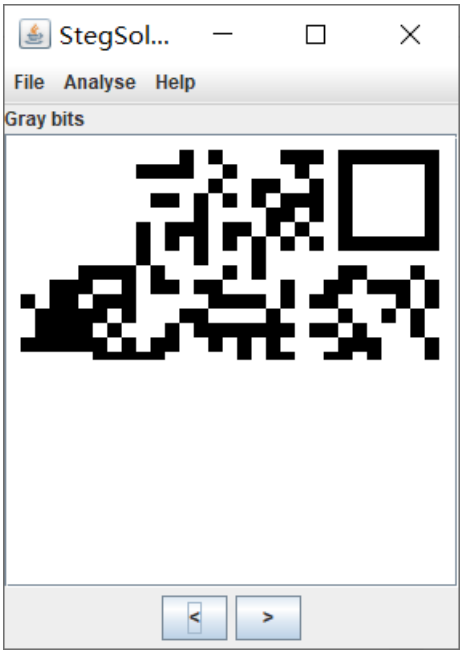


根据题目描述，应该还有一个图层

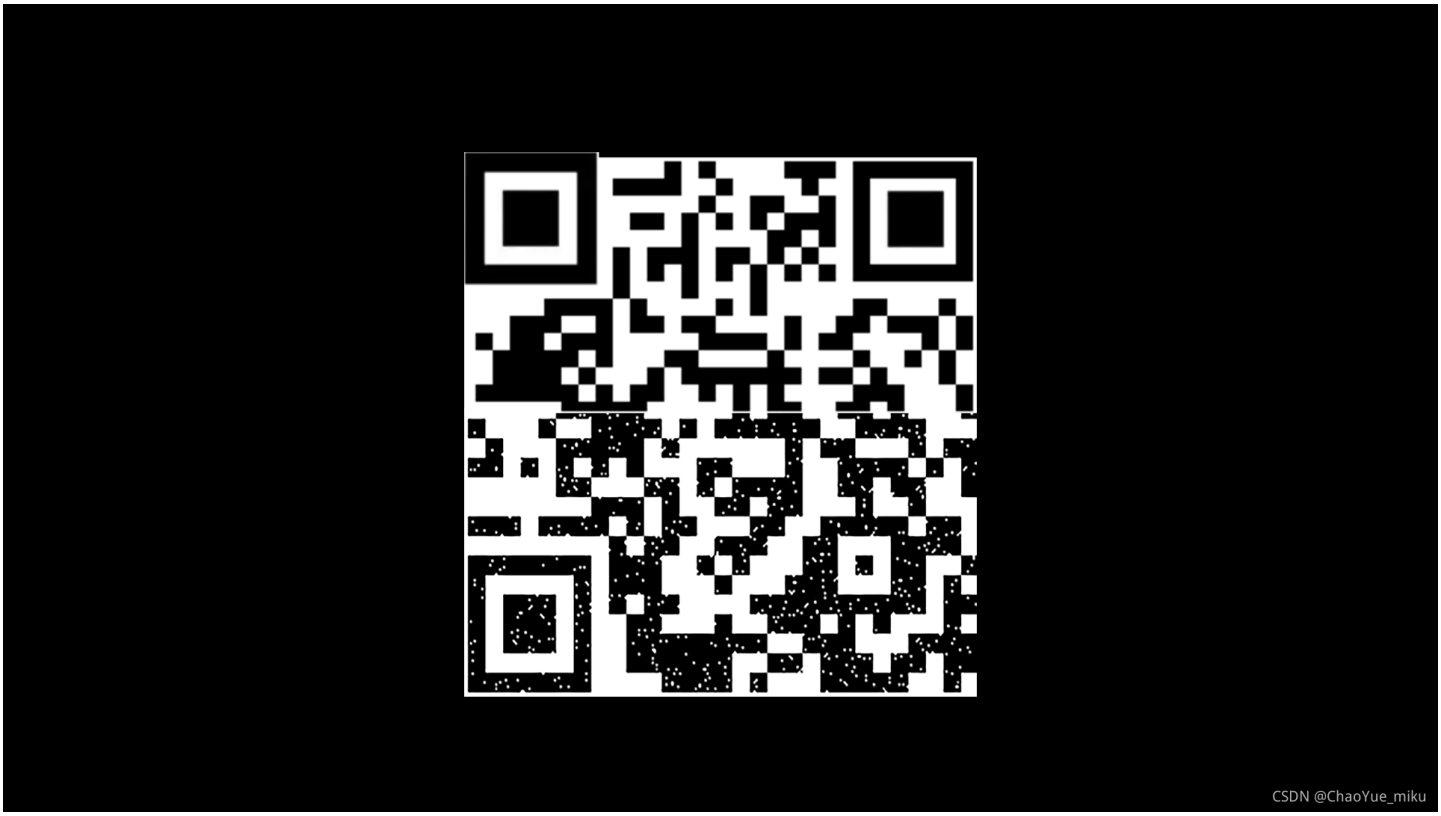
0x05 使用PS将另一个图层分离出来，用StegoSolve打开

`

查看不同通道，得到了二维码的上半部分



用PS将两个部分拼接为一个完整的二维码，然后补充上二维码定位符



0x06 使用CQR扫描二维码

0x07 得到flag： **flag{yanji4n_bu_we1shi}**

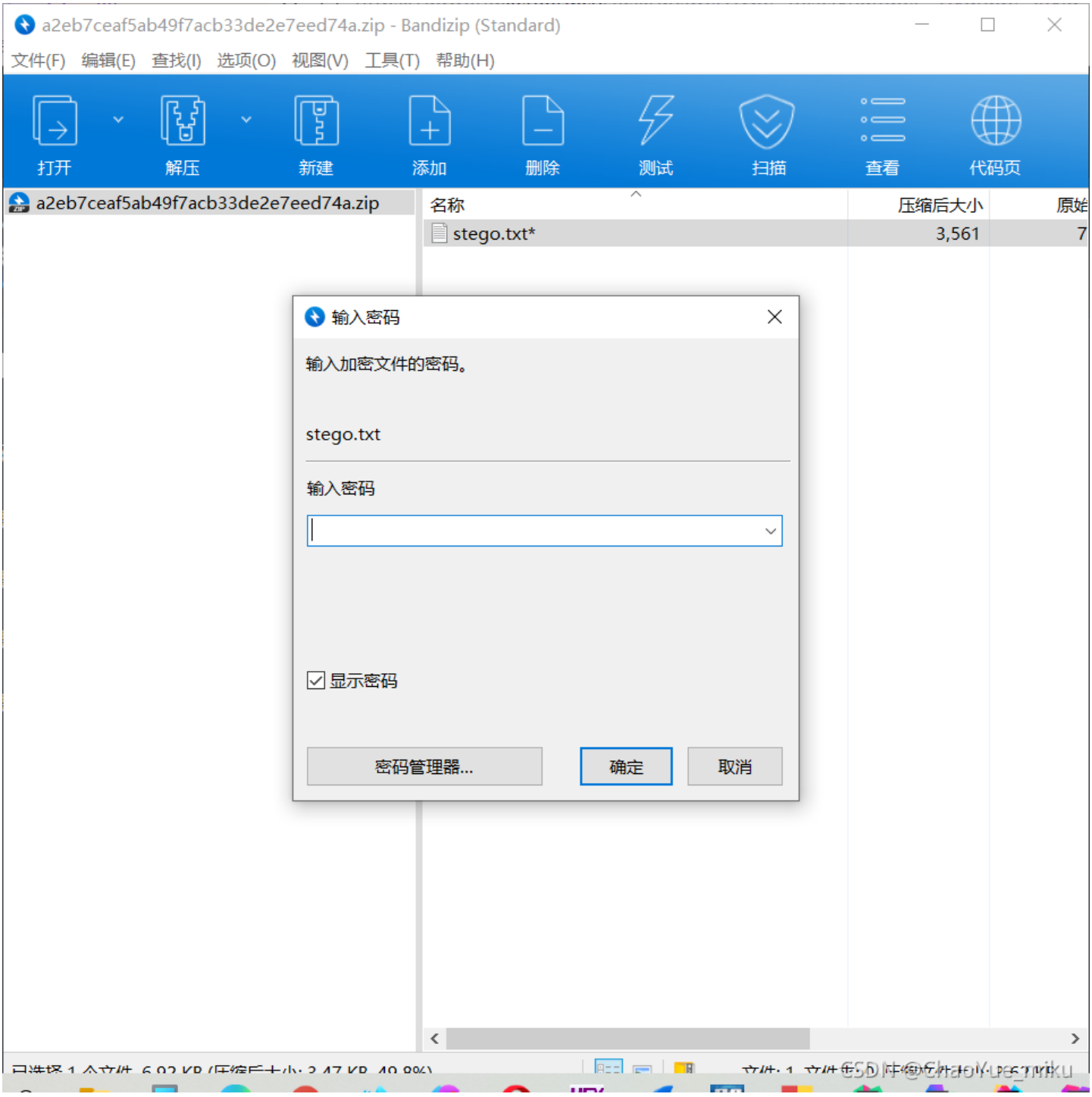
011 base64stego

难度系数： 5.0

题目来源： olympicCTF

题目描述： 菜狗经过几天的学习，终于发现了如来十三掌最后一步的精髓

0x01 下载附件并打开，在zip压缩包中有一个加密的txt文件



应该是伪加密，可以修改文件头，也能用360压缩直接打开

0x02 使用360压缩打开，查看文件内容

是许多行base64编码

```
U3RlZ2Fub2dyYXBoeSBpcyB0aGUgYXJ0IGFuZCBzY2llbmNlIG9m
IHdyaXRpbmcgaGlkZGVuIG1lc3NhZ2VzIGluIHN1Y2ggYSB3YXkgdGhhdCBubyBvbmV=
LCBhcGFydCBmcm9tIHRoZSBzZW5kZXIgYW5kIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGU=
Y3RzIHRoZSBleGlzdGVuY2Ugb2YgdGhlIG1lc3M=
YWdlLCBhIGZvcm0gb2Ygc2VjdXJpdHkgdGhyb3VnaCBvYnNjdXJpdHkuIFS=
aGUgd29yZCBzdGVnYW5vZ3JhcGh5IGlzIG9mIEdyZWVrIG9yaWdpbiBhbmQgbWVhbnMgImNvbmNlYW==
bGVkIHdyaXRpbmciIGZyb20gdGhlIEdyZWVrIHdvcmRzIHN0ZWdhbm9zIG1lYW5pbmcgImNv
dmVyZWQgb3IgcHJvdGVjdGVkIiwgYW5kIGdyYXBoZWluIG1lYW5pbmcgInRvIHc=
cml0ZSIuIFRoZSBmaXJzdCByZWNvcmRlZCB1c2Ugb2YgdGhlIHRlcm0gd2FzIGluIDE0OTkgYnkgSm9o
YW5uZXMgVHJpdGhlbWl1cyBpbiBoaXMgU3RlZ2Fub2dyYXBoaWEsIGEgdHJlYV==
dGlzZSBvbiBjcnlwdG9ncmFwaHkgYW5kIHN0ZWdhbm9ncmFwaHkgZGlzZ8==
dWlzZWQgYXMgYSBib29rIG9uIG1hZ2ljLiBHZW5lcmFsbHksIG1lc3P=
YWdlcyB3aWxsIGFwcGVhciB0byBiZSBzb21ldGhpbmcgZWxzZTogaW1hZ2VzLCBhcnRp
Y2xlcywgc2hvcHBpbmcgbGlzdHMsIG9yIHNvbWUgb3R=
aGVyIGNvdmVydGV4dCBhbmQsIGNsYXNzaWNhbGx5LCB0aGUgaGlkZGVuIG1lc3NhZ2UgbWF5IGJlIGluIGludmm=
c2libGUgaW5rIGJldHdlZW4gdGhlIHZpc2libGUgbGluZXMgb2YgYSBwcml2YXRlIGxldHRlci4NCg0KVGhl
IGFkdmFudGFnZSBvZiBzdGVnYW5vZ3JhcGh5LCBvdmVyIGNy
eXB0b2dyYXBoeSBhbG9uZSwgaXMgdGhhdCBtZXNzYWdlcyBkbyBub3QgYXR0cmFjdCBhdHRlbnRpb25=
IHRvIHRoZW1zZWx2ZXMuIFBsYWlubHkgdmlzaWJsZSBlbmNyeXB0ZWQgbWVzc2FnZXOXbm8gbWF0dGVyIF==
aG93IHVuYnJlYWthYmxll3dpbGwgYXJvdXNlIHN=
dXNwaWNpb24slGFuZCBtYXkgaW4gdGhlbXNlbHZlcyBiZSBpbmNyaW1pbmF0aW5nIP==
aW4gY291bnRyaWVzIHdoZXJlIGVuY3J5cHRpb24gaXMgaWxsZWdhbC4gVGhlcmVmb3JlLH==
IHdoZXJlYXMgY3J5cHRvZ3JhcGh5IHByb3RlY3RzIHRoZSBjb250ZW50cyBvZj==
IGEgbWVzc2FnZSwgc3RlZ2Fub2dyYXBoeSBjYW4gYmUgc2FpZCB0byBwcm90ZWN0IGJ=
b3RoIG1lc3NhZ2VzIGFuZCBjb21tdW5pY2F0aW5nIHBhcnRpZXMuDQoNClN0ZWdhbm9ncmFwaHkgaW5jbHW=
ZGVzIHRoZSBjb25jZWFsbWVudCBvZiBpbmZvcm1hdGlvbiB3aXRoaW4gY29t
cHV0ZXIgZmlsZXMuIEluIGRpZ2l0YWwgc3RlZ2Fub2dyYXBoeSwgZWxlY3Ryb25pYyBjb21tdW5pY2F0aW9u
cyBtYXkgaW5jbHVkZSBzdGVnYW5vZ3JhcGhpYyBjb2Rpbmcgaw5zaZ==
ZGUgb2YgYSB0cmFuc3BvcnQgbGF5ZXIsIHN1Y2ggYXMgYSBkb2N1bWVudCBmaWxlLCBpbWFnZSBmaWx=
ZSwgcHJvZ3JhbSBvciBwcm90b2NvbC4gTWVkaWEg
ZmlsZXMgYXJlIGlkZWFsIGZvciBhbmZvcm1hdGlvbiB3aXRoaW4gY29t
cHV0ZXIgZmlsZXMuIEluIGRpZ2l0YWwgc3RlZ2Fub2dyYXBoeSwgZWxlY3Ryb25pYyBjb21tdW5pY2F0aW9u
cyBtYXkgaW5jbHVkZSBzdGVnYW5vZ3JhcGhpYyBjb2Rpbmcgaw5zaZ==
ZGUgb2YgYSB0cmFuc3BvcnQgbGF5ZXIsIHN1Y2ggYXMgYSBkb2N1bWVudCBmaWxlLCBpbWFnZSBmaWx=
ZSwgcHJvZ3JhbSBvciBwcm90b2NvbC4gTWVkaWEg
ZmlsZXMgYXJlIGlkZWFsIGZvciBhbmZvcm1hdGlvbiB3aXRoaW4gY29t
biBiZWNhdXNlIG9mIHRoZWlyIGxhcmdlIHNpemUuIEFzIB==
YSBzaW1wbGUgZXhhbXBsZSwgYSBzZW5kZXIgbWlnaHQgc3RhcnQgd2l0aCBh
biBpbm5vY3VvdXMgaW1hZ2UgZmlsZSBhbmQgYWRqdXN0IHRoZSBjb2xvciBiBvZiBiBldmVyeSAxMDB0aCBwaXhlbCD=
dG8gY29ycmVzcG9uZCB0byBhIGxldHRlciBpbiB0aGUgYWxwaGFiZXQsIGF=
IGNoYW5nZSBzbyBzdWJ0bGUgdGhhdCBzb211b25lIG5vdCBzcGVjaWZpY2FsbHkgbG9va2luZyBm
b3IgaXQgaXMgdW5saWtlbHkgdG8gdGhpbm90aWNlIGl0Lg0KDQpuGU=
```

IGZpcnN0IHJlY29yZGVkIHVzZXMgb2Ygc3RlZ2Fub2dyYXBoeSBjYW4gYmUgdHJ=
YWNlZCBiYWNrIHRvIDQ0MCBCQyB3aGVuIEhlcm9kb3R1cyBtZW50aW9ucyB0d28gZXhhbXBsZXMgb+==
ZiBzdGVnYW5vZ3JhcGh5IGluIFRoZSBIaXN0b3JpZXMgb2Yg
SGVyb2RvdHVzLiBEZW1hcmF0dXMgc2VudCBhIHdhcm5pbmcgYWJvdXQgYSB=
Zm9ydGhjb21pbmcgYXR0YWNrIHRvIEdyZWVjZSBieSB3
cml0aW5nIGl0IGRpcmVjdGx5IG9uIHRoZSB3b29kZW4gYmFja2luZyBvZiBhIHdheCB0YWJsZXQgYmVm
b3JlIGFwcGx5aW5nIGl0cyBiZWVzd2F4IHN1cmZhY2UuIFdheCB0YWJsZXRzIHdlcmUgaW4gY29tbW9uIHVzZV==
IHRoZW4gYXMgcmV1c2FibGUgd3JpdGluZyBzdXJmYWNlcywgc29tZXRpbWX=
cyB1c2VkIGZvciBzaG9ydGhhbmQuIEFub3RoZXIgYW5jaWVudCBleGFtcGxlIGlzIHRoYXQgb9==
ZiBIaXN0aWFldXMsIHdobyBzaGF2ZWQgdGhlIGhlYWQgb2YgaGlzIG1vc3QgdHJ1c3RlZCBz
bGF2ZSBhbmQgdGF0dG9vZWQgYSBtZXNzYWdlIG9uIGl0LiBBZnRlciBoaXMgaGFpciBoYWQgZ5==
cm93biB0aGUgbWVzc2FnZSB3YXMgaGlkZGVuLiBUaGUgcHVycG9zZSB3YXMgdG+=
IGluc3RpZ2F0ZSBhIHJldm9sdCBhZ2FpbnN0IHRoZSBQZXJzaWFucy4NCg0KU3RlZ2Fub2dyYXBoeSBoYXMgYm==
ZWVuIHdpZGVseSB1c2VkLCBpbmNsdWRpbmcgaW4gcmVjYW50IGhpc3RvcmljYWwgdGltZXMgYW5kIHT=
aGUgcHJlc2VudCBkYXkuIFBvc3NpYmxlIHBlcm11dGF0aW9ucyBhcmUgZW5kbGVzcyBhbmT=
IGtub3duIGV4YW1wbGVzIGluY2x1ZGU6DQoqIEhpZGRlbiBtZXNzYWdlcyB3aXRoaW4gd2F4IHRh
YmxldHM6IGluIGFuY2llbnQgR3JlZWNlLCBwZW9wbGUgd3JvdGUgbWV=
c3NhZ2VzIG9uIHRoZSB3b29kLCB0aGVuIGNvdmVyZWQgaXQgd2l0aCB3YXggdXBvbiB3aGljaCBhbiBpbm5vY2Vu
dCBjb3ZlcmluZyBtZXNzYWdlIHdhcyB3cml0dGVu
Lg0KKiBIaWRkZW4gbWVzc2FnZXMgb24gbWVzc2VuZ2VyJ3MgYm9keTogYWxzbyB1c2VkIGluIGFuY2llbt==
dCBHcmVlY2UuIEhlcm9kb3R1cyB0ZWxscyB0aGUgc3Rvcnkgb1==
ZiBhIG1lc3NhZ2UgdGF0dG9vZWQgb24gYSBzbGF2ZSdzIHNoYXZlZCBoaWFkLCBoaWRkZW4gYnkgdGhl
IGdyb3d0aCBvZiBoaXMgaGFpciwgYW5kIGV4cG9zZVQgaXBvbiB3aGljaCBhbiBpbm5vY2Vu
dCBjb3ZlcmluZyBtZXNzYWdlIHdhcyB3cml0dGVu
Lg0KKiBiIaWRkZW4gbWVzc2FnZXMgb24gbWVzc2VuZ2VyJ3MgYm9keTogYWxzbyB1c2VkIGluIGFuY2llbt==
dCBHcmVlY2UuIEhlcm9kb3R1cyB0ZWxscyB0aGUgc3Rvcnkgb1==
ZiBhIG1lc3NhZ2UgdGF0dG9vZWQgb24gYSBzbGF2ZSdzIHNoYXZlZCBoaWFkLCBoaWRkZW4gYnkgdGhl
IGdyb3d0aCBvZiBoaXMgaGFpciwgYW5kIGV4cG9zZVQgaXBvbiB3aGljaCBhbiBpbm5vY2Vu
dCBjb3ZlcmluZyBtZXNzYWdlIHdhcyB3cml0dGVu
aXMgbWV0aG9kIGhhcyBvYnZpb3VzIGRyYXdiYWNrcyz=
IHN1Y2ggYXMgZGVsYXllZCB0cmFuc21pc3Npb24gd2hpbGUgd2FpdGluZyBmIGdGhlIHP=
bGF2ZSdzIGhhaXIgdG8gZ3JvdywgYW5kIHRoZSByZXN0cmljdGlvbnMgb3==
biB0aGUgbnVtYmVyIGFuZCBzaXplIG9mIG1lc3M=
YWdlcyB0aGF0IGNhbiBiZSBlbmNvZGVkIG9uIG9uZSBwZXJzb24=
J3Mgc2NhbHAuDQoqIEluIFdXSUksIHRoZSBGcmVuY2ggUmVzaXN0YW5jZSBzZW50IHNvbWUgbWVzc2FnZXMgd2==
cml0dGVuIG9uIHRoZSBiYWNrcyBvZiBjb3VyaWVycyD=
dXNpbmcgaW52aXNpYmxlIGluay4NCiogSGlkZGVuIG1lc3NhZ2VzIG9uIHBhcGVyIHdy
aXR0ZW4gaW4gc2VjcmV0IGlua3MsIHVuZGVyIG90aGVyIG1lc3NhZ2Vz
IG9yIG9uIHRoZSBibGFuayBwYXJ0cyBvZiBvdGhlct==
IG1lc3NhZ2VzLg0KKiBNZXNzYWdlcyB3cml0dGVuIGluIE1vcnNlIGNvZGUgb24ga25pdHRpbmcgeWFybiBhbmQg
dGhlbiBrbml0dGVkIGludG8gYSBwaWVjZSBvZiBjbG90aGluZyB3b3K=
biBieSBhIGNvdXJpZXIuDQoqIE1lc3NhZ2VzIHdyaXR0ZW4gb24gdGhlIGJhY2sgb5==
ZiBwb3N0YWdlIHN0YW1wcy4NCiogRHVyaW5nIFdZCBhZnRlcm==
IFdvcmxkIFdhciBJSSwgZXNwaW9uYWdlIGFnZW50cyB1c2VkIHBob3RvZ3JhcGhpY2FsbHkgcO==
cm9kdWNlZCBtaWNyb2RvdHMgdG8gc2VuZCBpbmZvcm1hdGlvbiBiYWNrIGFuZH==
IGZvcnRoLiBNaWNyb2RvdHMgd2VyZSB0eXBpY2FsbHkg
bWludXRlLCBhcHByb3hpbWF0ZWx5IGxlc3MgdGhhbiB0aGUgc2l6ZSBvZiB0aGUgcGVyaW9kIHByb2R=
dWNlZCBieSBhIHR5cGV3cml0dGVuIGluIFdXSUkgbWljcm9kb3RzIG5lZWRlZCB0byByBiZSBlbWJlZGRlZB==
IGluIHRoZSBwYXBlciBhbmQgY292ZXJlZCB3aXRoIGFuIGFkaGVzaXZlIChzdWNoIGFzIGNvbGxvZGlvbikuIFR=
aGlzIHdhcyByZWZsZWN0aXZlIGFuZCB0aHVzIGRldGVjdGFibGUg
Ynkgdmlld2luZyBhZ2FpbnN0IGdsYW5jaW5nZ2h0LiBBbHRlcm5hdGl2ZSB0ZWNobmlxdWVzIGluY2x1ZGVk
IGluc2VydGluZyBtaWNyb2RvdHMgaW50byBzbGl0cyBjdXQgaW50byB0aGUgZWRnZSBvZiBvdGhlciBkb2xs
IHBvc3QgY2FyZHMuDQoqIER1cmluZyBXb3JsZCBXYXIgSUksIGEgc3B5IGZvciciB=
SmFwYW4gaW4gTmV3IFlvcmsgQ2l0eSwgVmVsdmFsZWW=
IERpY2tpbnNvbiwgc2VudCBpbmZvcm1hdGlvbiB0byBhY2NvbW1vZGF0aW9=
biBhZGRyZXNzZXMgaW4gbmV1dHJhbCBTb3V0aCBBbWVyaWO=
YS4gU2hlIHdhcyBhIGRlYWxlciBpbiBkb2xscywgYW5kIG==
aGVyIGxldHRlcnMgZGlzY3Vzc2VkIGhvdyBtYW55IG9mIHRoaXMgb3IgdGhhdCBkb2xs
IHRvIHNoaXAuIFRoZSBzdGVnb3RleHQgd2FzIHRoZSBkb2xsIG9yZGVycywgb
IGNvbmNlYWxlZCAicGxhaW50ZXh0iiwgYW5kIG==
ZGVkIGFuZCBnYXZlIGluZm9ybWF0aW9uIGFib3V0IHNoaXAgbW92ZW1lbnRzLF==
IGV0Yy4gSGVyIGNhc2UgYmVjYW1lIHNvbWV3aGF0IGZh
bW91cyBhbmQgc2vuZCBpbmZvcm1hdGlvbiB0byBhY2NvbW1vZGF0aW9=
biBhZGRyZXNzZXMgaW4gbmV1dHJhbCBTb3V0aCBBbWVyaWO=
YS4gU2hlIHdhcyBhIGRlYWxlciBpbiBkb2xscywgYW5kIG==
aGVyIGxldHRlcnMgZGlzY3Vzc2VkIGhvdyBtYW55IG9mIHRoaXMgb3IgdGhhdCBkb2xs
IHRvIHNoaXAuIFRoZSBzdGVnb3RleHQgd2FzIHRoZSBkb2xsIG9yZGVycywgb
IGNvbmNlYWxlZCAicGxhaW50ZXh0iiwgYW5kIG==
ZGVkIGFuZCBnYXZlIGluZm9ybWF0aW9uIGFib3V0IHNoaXAgbW92ZW1lbnRzLF==
IGV0Yy4gSGVyIGNhc2UgYmVjYW1lIHNvbWV3aGF0IGZh
bW91cyBhbmQgc2hlIGJlY2FtZSBrbm93biBhcyB=

```
IERvbGwgV29tYW4uDQoqIENvbGQgV2FyIGNvdW50
ZXItcHJvcGFnYW5kYS4gSW4gMTk2OCwgY3JldyBtZW1iZW==
cnMgb2YgdGhlIFVTUyBQdWVibG8gKEFHRVItMikgaW50ZWxsaWdlbmNlIHNoaXAgaGVsZCBhcyBwcm==
aXNvbmVycyBieSBOb3J0aCBLb3JlYSwgY29tbXVuaWNhdGVkIGluIHNpZ25=
IGxhbmd1YWdlIGR1cmluZyBzdGFnZWQgcGhvdG8gb3Bwb3J0
dW5pdGllcywgaW5mb3JtaW5nIHRoZSBVbml0ZWQgU3RhdGVzIHRoZXkg
d2VyZSBub3QgZGVmZWN0b3JzIGJ1dCByYXRoZXIgd2VyZSBiZWluZyBoZWxkIGNh
cHRpdmUgYnkgdGhlIE5vcnRoIEtvcmVhbnMuIEluIG90aGVyIHBob3Rv
cyBwcmVzZW50ZWQgdG8gdGhlIFVTLCBjcmV3IG1lbWJlcnMgZ2F2ZSAidGhlIGZpbmdlciIgdG8g
dGhlIHVuc3VzcGVjdGluZyBOb3J0aCBLb3JlYW5zLCBpbiBhbiBhdHRlbXB0IHRvIE==
ZGlzY3JlZGl0IHBob3RvcyB0aGF0IHNob3dlZCB0aGVtIHNtaQ==
bGluZyBhbmQgY29tZm9ydGFibGUuDQoNCi0tDQpodHRwOi8vZW4ud2lraXBlZGlhLm9yZw==
L3dpa2kvU3RlZ2Fub2dyYXBoeQ0K
```

根据题目可知，文本内容是base64隐写

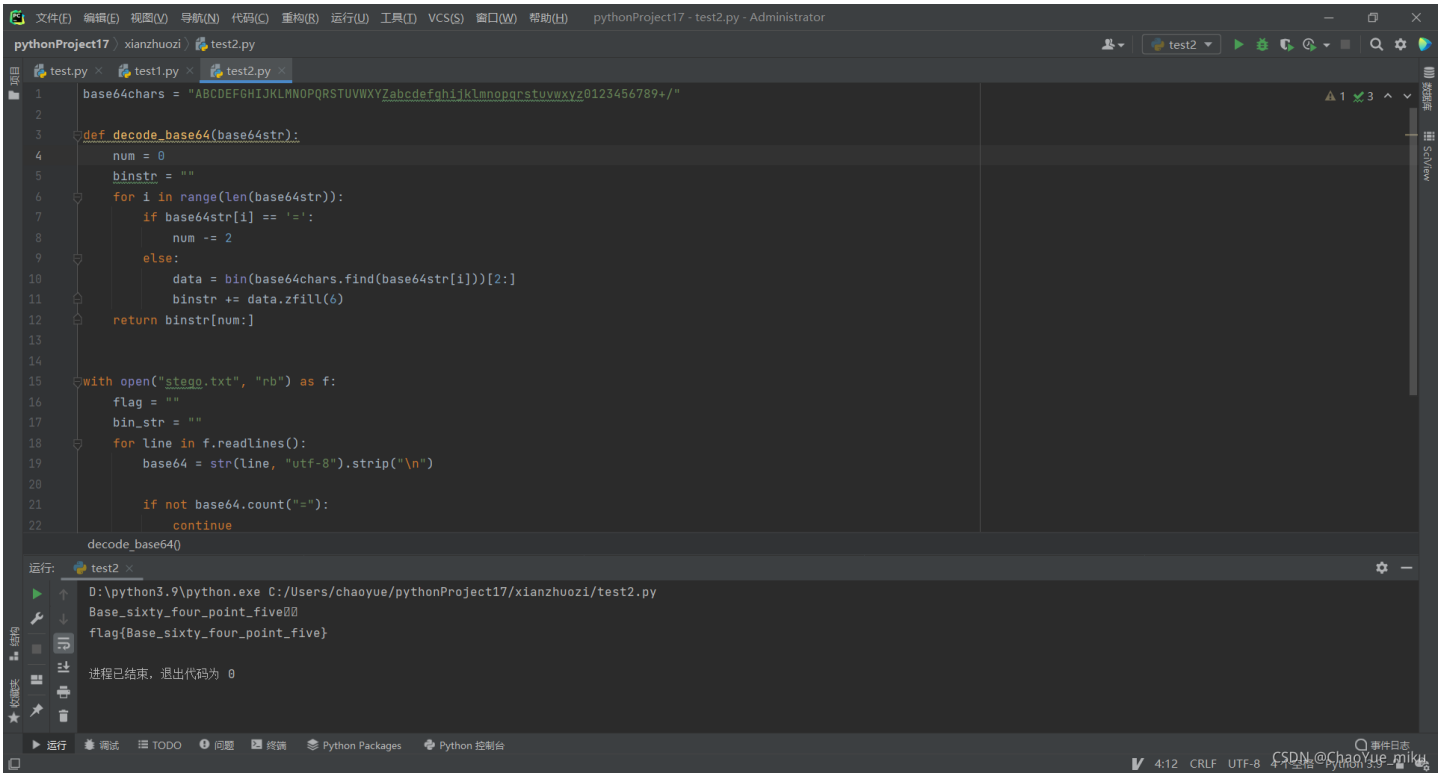
0x03 编写解密脚本

```python
base64chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"

def decode_base64(base64str):
    num = 0
    binstr = ""
    for i in range(len(base64str)):
        if base64str[i] == '=':
            num -= 2
        else:
            data = bin(base64chars.find(base64str[i]))[2:]
            binstr += data.zfill(6)
    return binstr[num:]


with open("stego.txt", "rb") as f:
    flag = ""
    bin_str = ""
    for line in f.readlines():
        base64 = str(line, "utf-8").strip("\n")

        if not base64.count("="):
            continue
        bin_str += decode_base64(base64)
    for j in range(0, len(bin_str), 8):
        flag += chr(int(bin_str[j:j + 8], 2))
    print("flag{{{}}}".format(flag.strip(b"\0x00".decode())))
```

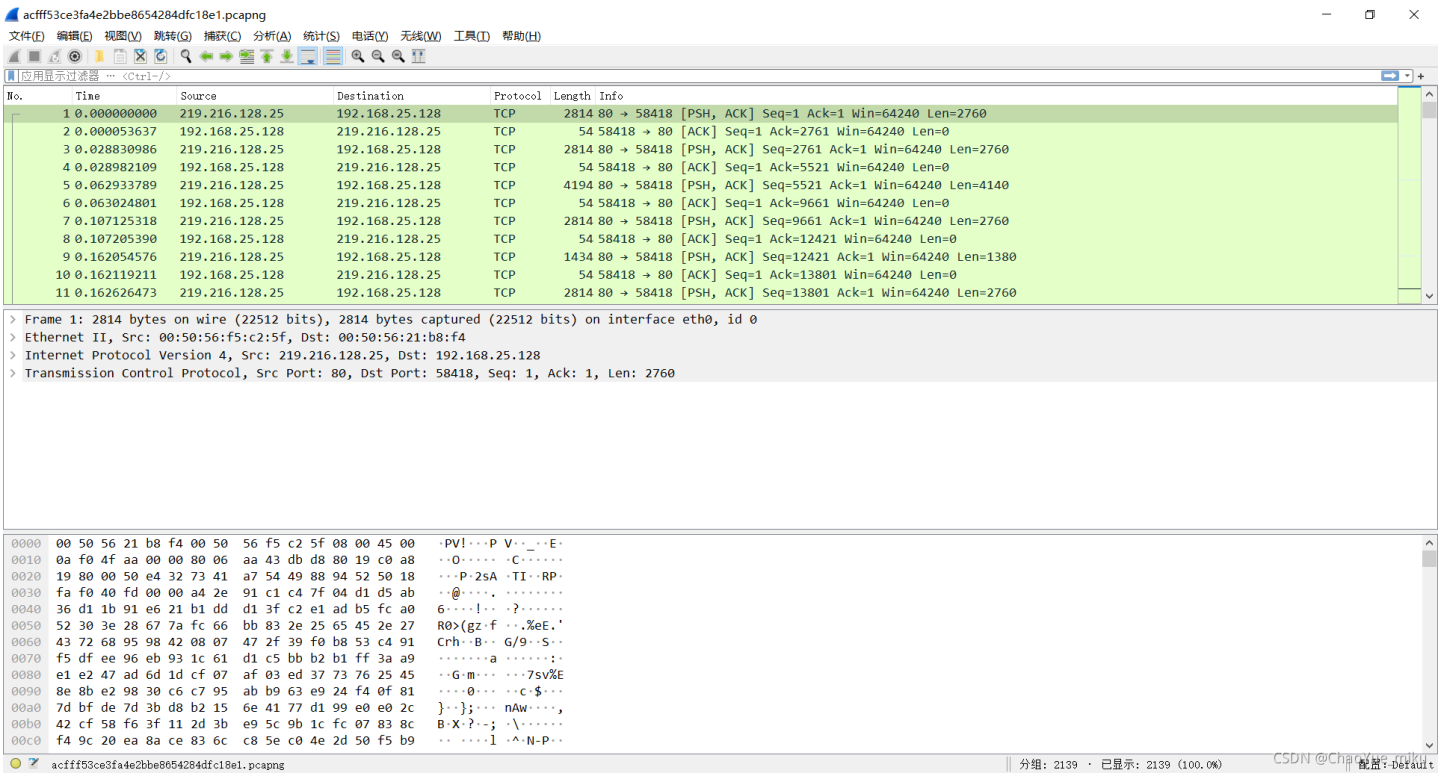


0x04 得到flag： flag{Base_sixty_four_point_five}

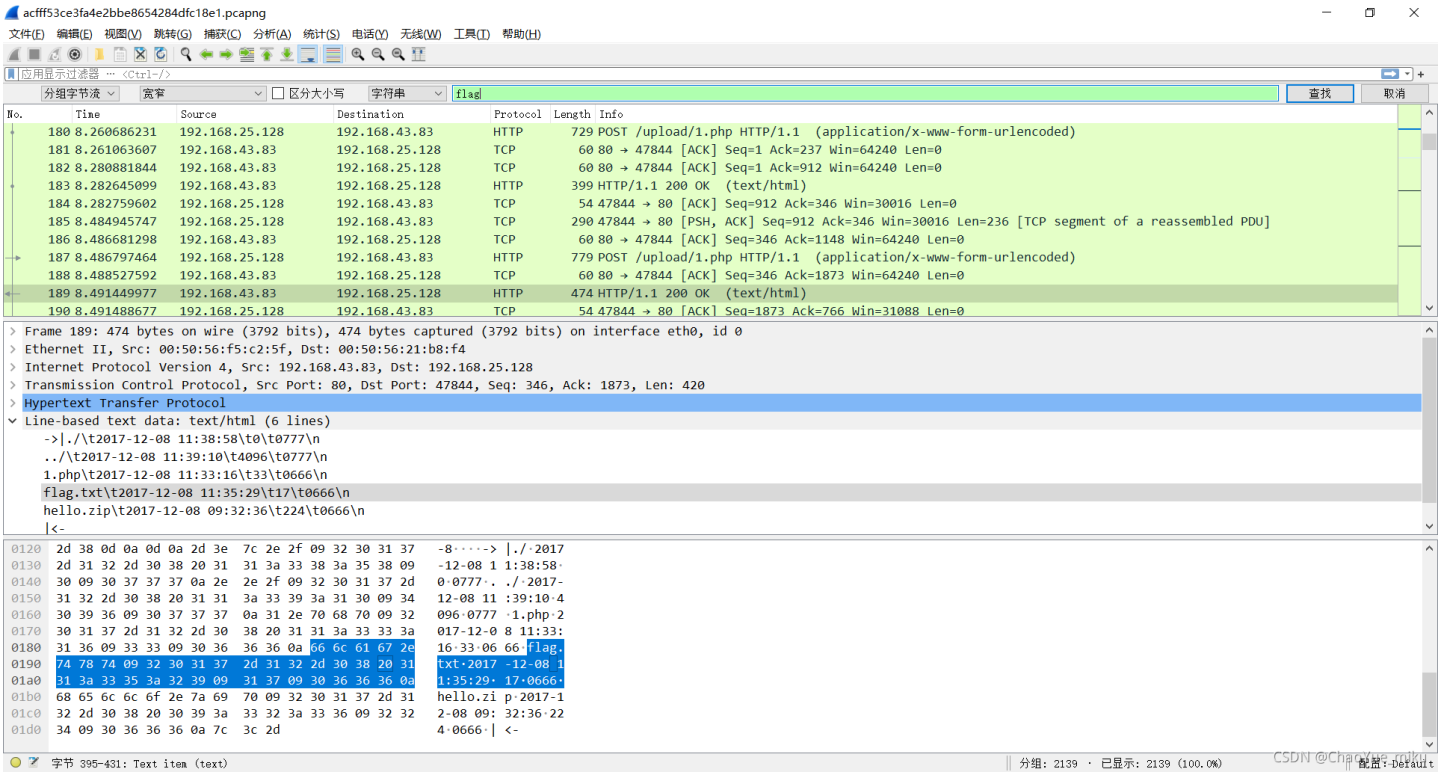
012 功夫再高也怕菜刀

难度系数： 6.0
题目来源： 安恒杯
题目描述： 菜狗决定用菜刀和菜鸡决一死战

0x01 下载附件，打开后发现是一个流量包



直接搜索字符串`flag`，发现存在一个`flag.txt`的文本文档和一个`hello.zip`的压缩包



0x02 使用foremost分离文件

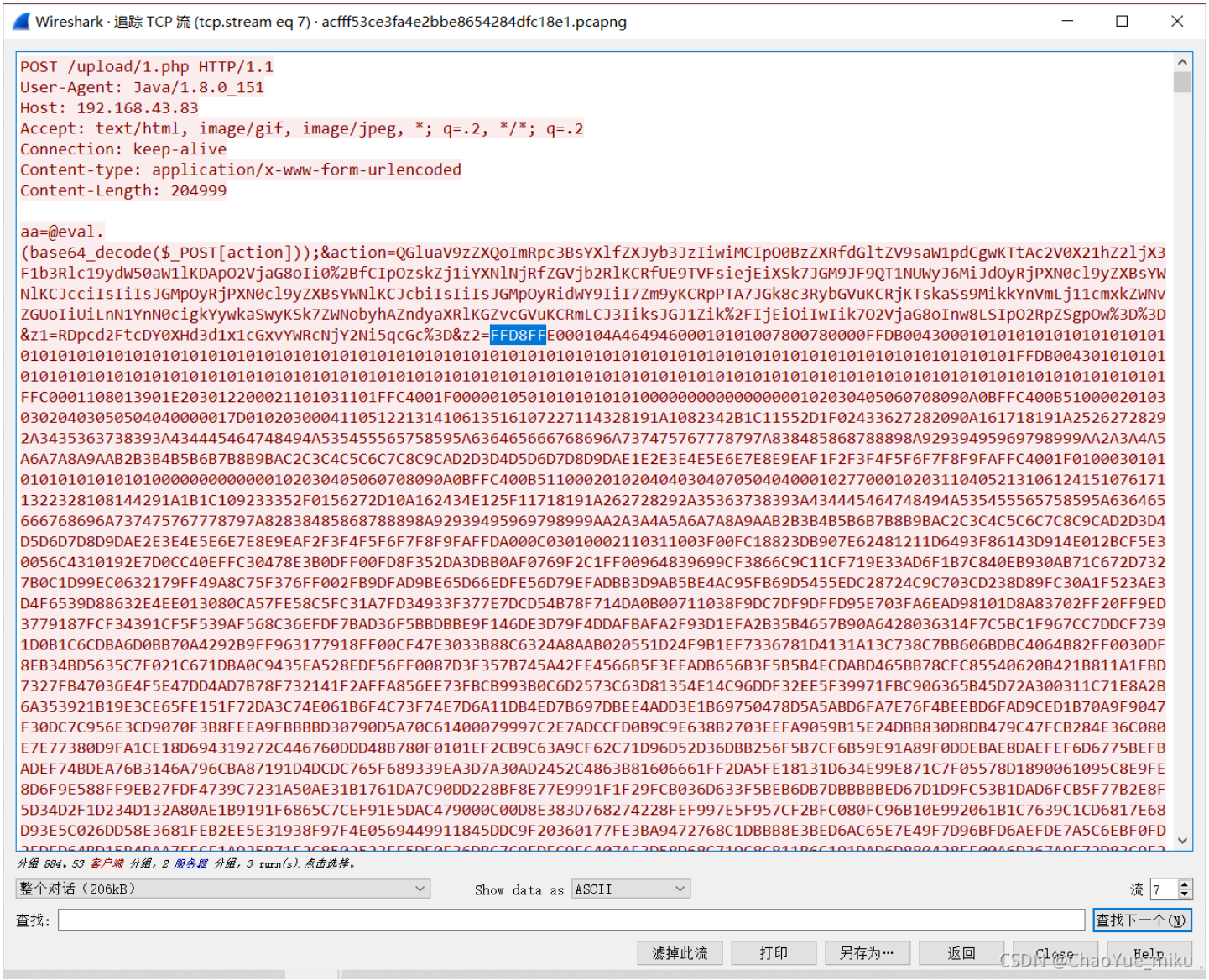
![在这里插入图片描述](https://img-blog.csdnimg.cn/7ec08916b92f460caba190d8f16b2a6d.png?x-oss-process=image/watermark,type_ZHJvaWRzYW5zZmFsbGJhY2s,shadow_50,text_Q1NETiBAQ2hhb1l1ZV9taWt1,size_20,color_FFFFFF,t_70,g_se,x_16) 分离出了压缩文件，其中打开`flag.txt`需要密码

0x03 在流量包中寻找密码

根据题目提示，有人通过`中国菜刀`上传了一些文件

对字符串搜索的第一个结果追踪TCP流，果然发现了文件上传的内容

`FFD8FFFE`开始，`FFD9`结束，很明显是jpg图片



![在这里插入图片描述](https://img-blog.csdnimg.cn/12b95d08ad4c479b83054daec2020f58.png?x-oss-process=image/watermark,type_ZHJvaWRzYW5zZmFsbGJhY2s,shadow_50,text_Q1NETiBAQ2hhb1l1ZV9taWt1,size_20,color_FFFFFF,t_70,g_se,x_16

0x04 使用010 Editor导入十六进制部分，保存后打开查看图片

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-9346Ls2f-1631116039363)(/upload/2021/09/%E5%9B%BE%E7%89%87-75cebb46c46c4ece8082d15750c4742e.png)]

得到解压密码：

`Th1s_1s_p4sswd_!!!`

0x05 输入解压密码打开flag.txt

0x06 得到flag： flag{3OpWdJ-JP6FzK-koCMAK-VkfWBq-75Un2z}