

# CTF学习之路-攻防世界：MISC，入门篇

原创

[屌丝小帅的逆袭](#) 于 2020-11-26 11:24:17 发布 1592 收藏 25

分类专栏：[屌丝小帅的CTF学习过程](#) 文章标签：[安全](#) [经验分享](#) [恰饭](#)

版权声明：本文为博主原创文章，遵循[CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/xuhc25/article/details/110167384>

版权



[屌丝小帅的CTF学习过程](#) 专栏收录该内容

4 篇文章 0 订阅

订阅专栏

## 目录

- 一、感谢和安利
- 二、MISC-新手练习篇Writeup
  - 1、this\_is\_flag
  - 2、pdf
  - 3、如来十三掌
  - 4、give\_you\_flag
  - 5、坚持60S
  - 6、gif
  - 7、掀桌子
  - 8、stegano
  - 9、SimpleRAR(比较麻烦)
  - 10、base64stego
  - 11、ext3
  - 12、功夫再高也怕菜刀

## 一、感谢和安利

我是从攻防世界入门的，感谢攻防世界的免费靶场

靶场地址：攻防世界


<https://adworld.xctf.org.cn/>



## 二、MISC-新手练习篇Writeup

## 1、this\_is\_flag

返回 本题用时: 48秒

this\_is\_flag  76 最佳Writeup由王兆敏提供

WP  建议

难度系数:    2.0

题目来源: 暂无

题目描述: Most flags are in the form flag{xxx}, for example:flag{th1s\_!s\_a\_d4m0\_4!a9}

题目场景: 暂无

题目附件: 暂无

题目已答对

分享wp点赞赚金币哦  
马上去写  
<https://blog.csdn.net/xuhc25>

WP:

明显，入门用的，了解flag格式，直接提交题目例子的flag就可以了

flag{th1s\_!s\_a\_d4m0\_4!a9}

## 2、pdf

返回 本题用时: 2时8分12秒

pdf  71 最佳Writeup由S\_O\_L\_R提供

难度系数:     3.0

题目来源: [csaw](#)

题目描述: 菜猫给了菜狗一张图，说图下面什么都没有

题目场景: 暂无

题目附件: [附件1](#)

<https://blog.csdn.net/xuhc25>

WP:

题目给的一个PDF文件，使用PDF编辑器打开（我用的是万兴PDF编辑器）



<https://blog.csdn.net/xuhc25>

移出来看到就是flag了



flag{security\_through\_obscurity}

<https://blog.csdn.net/xuhc25>

### 3、如来十三掌





本题用时: 1天1时28分34秒

# 如来十三掌

👍 83

最佳Writeup由 [flag{not\\_here}](#) · 渣渣禹提供

难度系数: 3.0

题目来源: 暂无

题目描述: 菜狗为了打败菜猫, 学了一套如来十三掌。

题目场景: 暂无

题目附件: [附件1](#)

<https://blog.csdn.net/xuhc25>

附件是一份文档:

夜哆悉諳多苦奢陀奢諦冥神哆盧穆幡三侄三即諸諳即冥迦冥隸數顛耶迦奢若吉怯陀諳怖奢智侄諸若奢數菩奢集遠俱老竟寫明奢若梵等盧幡豆蒙密離怯婆幡礙他哆提哆多鉢以南哆心曰姪罰蒙呐神。舍切真怯勝呐得俱沙罰娑是怯遠得呐數罰輸哆遠薩得槃漫夢盧幡亦醢呐娑幡瑟輸諳尼摩罰薩冥大倒參夢侄阿心罰等奢大度地冥殿幡沙蘇輸奢恐豆侄得罰提哆伽諳沙楞鉢三死怯摩大蘇者數一遮

文档结尾

<https://blog.csdn.net/xuhc25>

一段经文, 那不就是与佛论禅加密?

<http://www.keyfc.net/bbs/tools/tudoucode.aspx>

记得加“佛曰”前缀

## 与佛论禅

MzkuM3gvMUAwnzuvn3cgozM1MTuvqzAenJchMUAeqzWenzEmLJW9

听佛说宇宙的真谛

参悟佛所言的真意

普度众生

人无善恶, 善恶存乎尔心

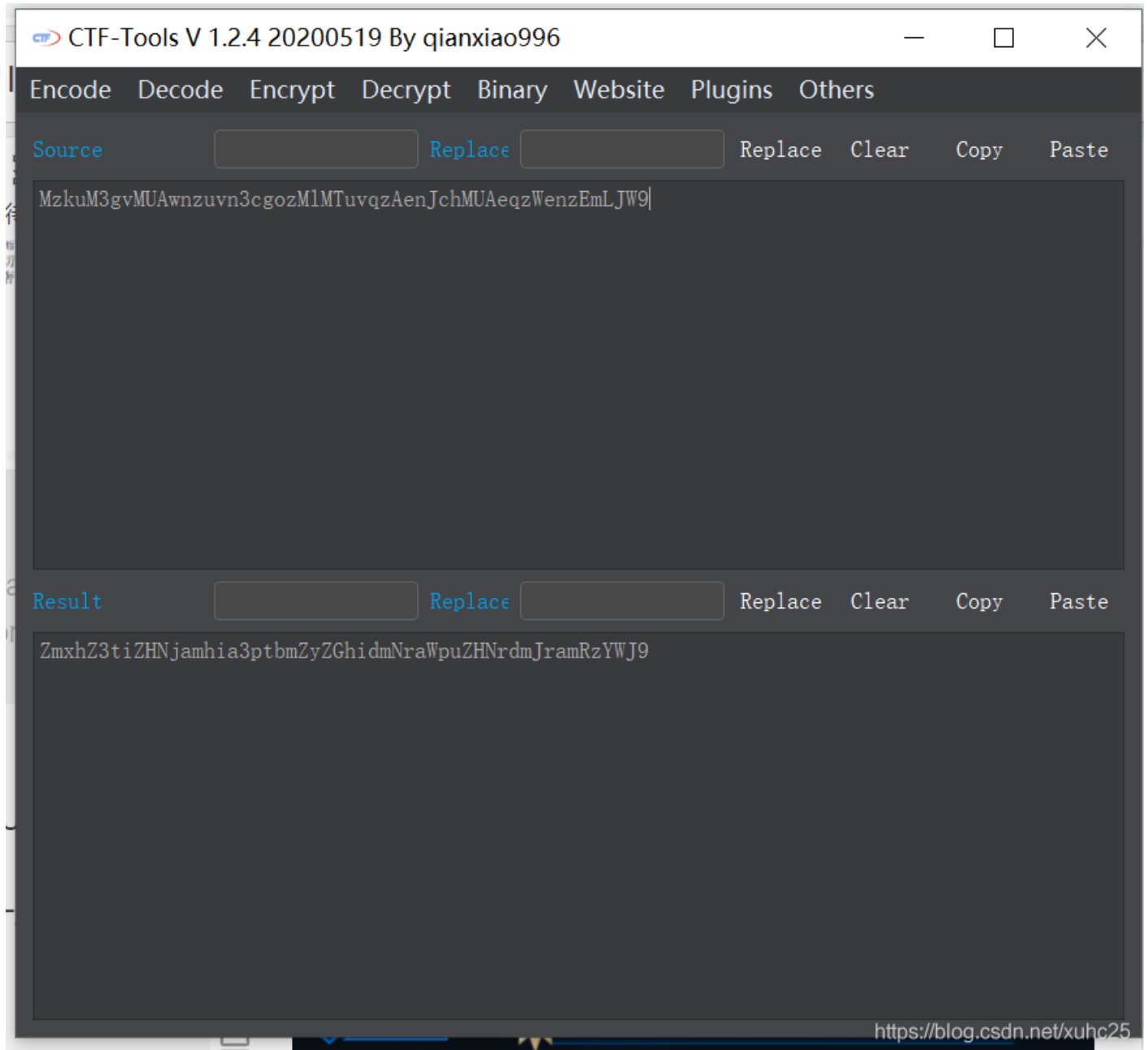
佛曰: 夜哆悉諳多苦奢陀奢諦冥神哆盧穆幡三侄三即諸諳即冥迦冥隸數顛耶迦奢若吉怯陀諳怖奢智侄諸若奢數菩奢集遠俱老竟寫明奢若梵等盧幡豆蒙密離怯婆幡礙他哆提哆多鉢以南哆心曰姪罰蒙呐神。舍切真怯勝呐得俱沙罰娑是怯遠得呐數罰輸哆遠薩得槃漫夢盧幡亦醢呐娑幡瑟輸諳尼摩罰薩冥大倒參夢侄阿心罰等奢大度地冥殿幡沙蘇輸奢恐豆侄得罰提哆伽諳沙楞鉢三死怯摩大蘇者數一遮

得到一串字符:

MzkuM3gvMUAwnzuvn3cgozMlMTuvqzAenJchMUAeqzWenzEmLJW9

看着这段字符, 百思不得其解, 最后题目: “如来十三掌”, 十三?? 试试rot13解码, 得到一串字符串:

ZmxhZ3tiZHNjamhia3ptbmZyZGhidmNraWpuZHNrdmJramRzYWJ9



最后丢到解密工具里一键解码, 得到flag:

flag{bdscjhbkmnfrdhbvckijndskvbkjdsab}



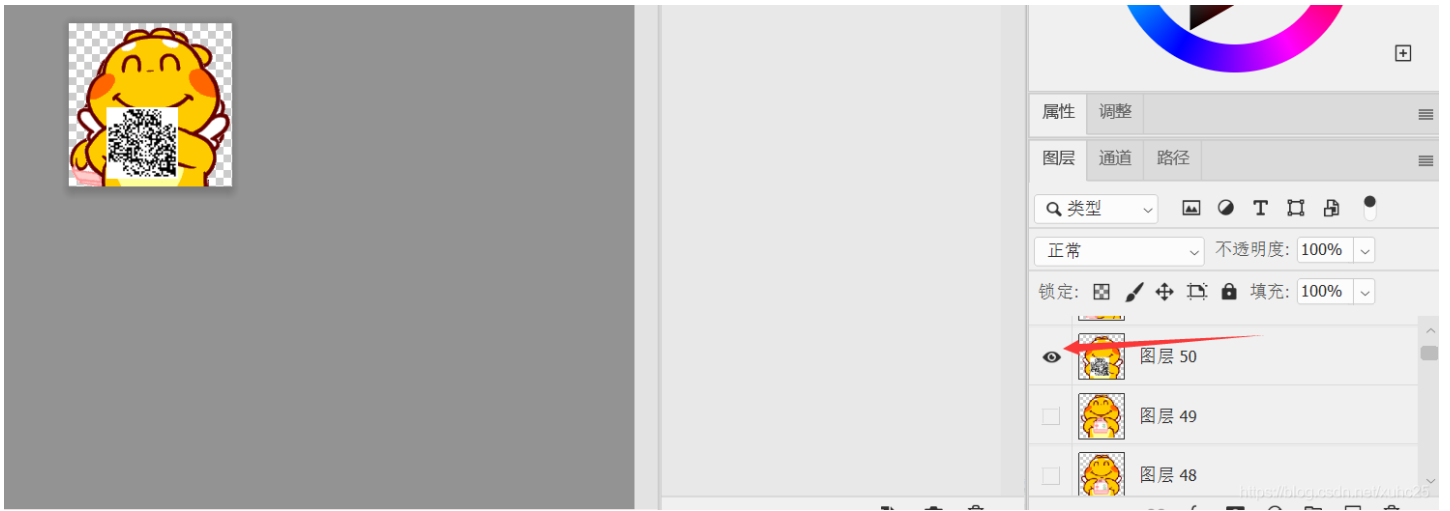


#### 4、give\_you\_flag

WP: 附件是个GIF图片, 看到闪的飞快, 最后有一帧黑乎乎的, 那就是可疑的地方了, 思路肯定是工具逐帧分解GIF

可以用“GIF动画帧提取器”, 或者PS分解出来。

PS打开: 找到可疑图层, 打开图层显示, 就可以看到可疑的那一帧。



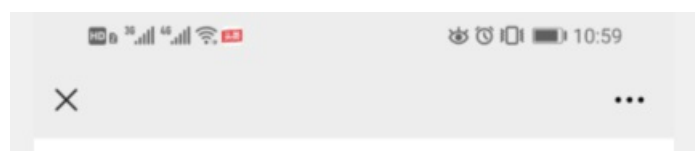
GIF动画帧提取器打开，找到可疑帧



小恐龙手上拿着的是二维码，但是少了三个定位点，要PS回去。



扫描二维码得到flag

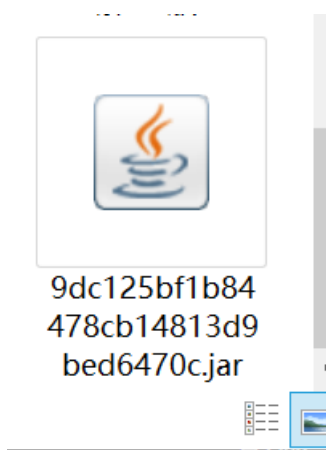


flag{e7d478cf6b915f50ab1277f78502a2c5}

## 5、坚持60S

The screenshot shows the 'World of Attack&Defense' (攻守世界) CTF platform interface. At the top, there are navigation tabs for '答题' (Solve), '竞赛' (Contest), '排行榜' (Ranking), and '队伍' (Team). Below the navigation, there is a '返回' (Return) button and a star icon. The challenge title '坚持60s' is displayed, along with a '12' likes badge and a '最佳Writeup由不要让我起名提供' (Best Writeup by 不要让我起名提供) badge. The difficulty coefficient is shown as 4.0 stars. The source is '08067CTF'. The description is '菜狗发现最近菜猫不爱理他，反而迷上了菜鸡' (A dog found that a cat doesn't like him anymore, but he's obsessed with a chicken). The scene is '暂无' (None). There is one attachment named '附件1'. The URL 'https://blog.csdn.net/xuhc25' is visible in the bottom right corner.

下载下来是一个jar包



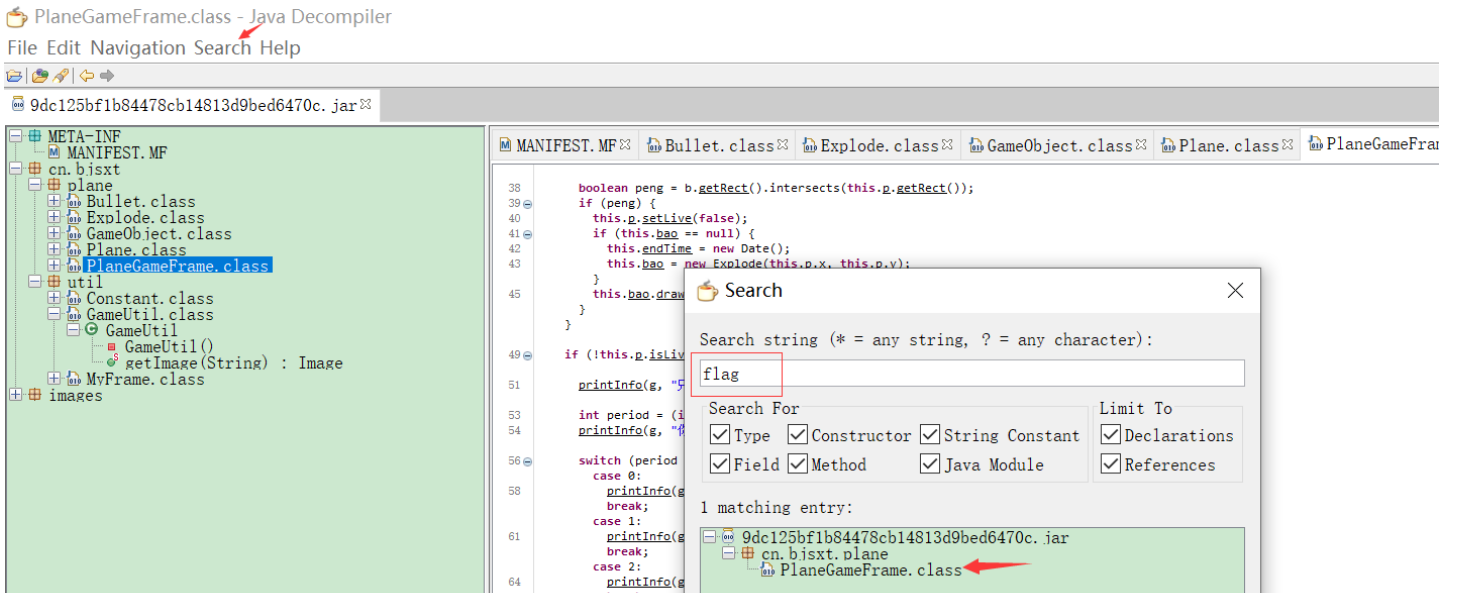
运行起来是一个游戏，没啥有用的信息

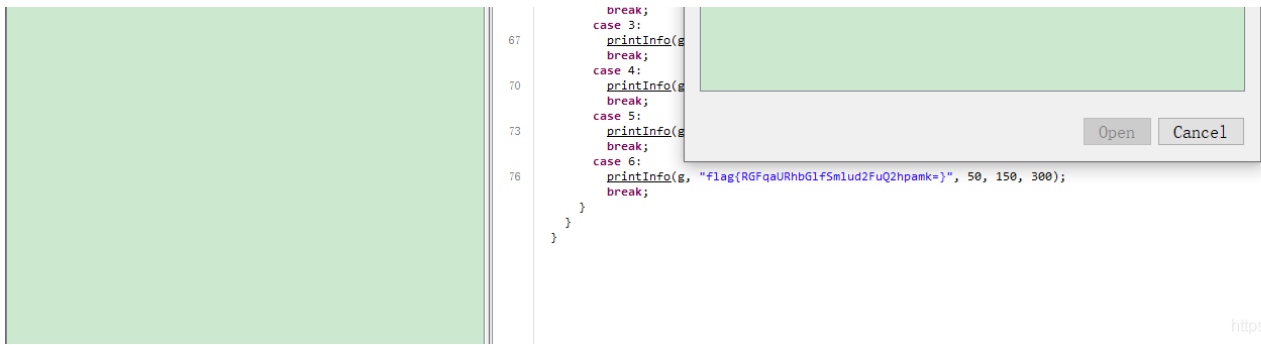






使用gd-gui工具反编译看看  
查找关键字flag



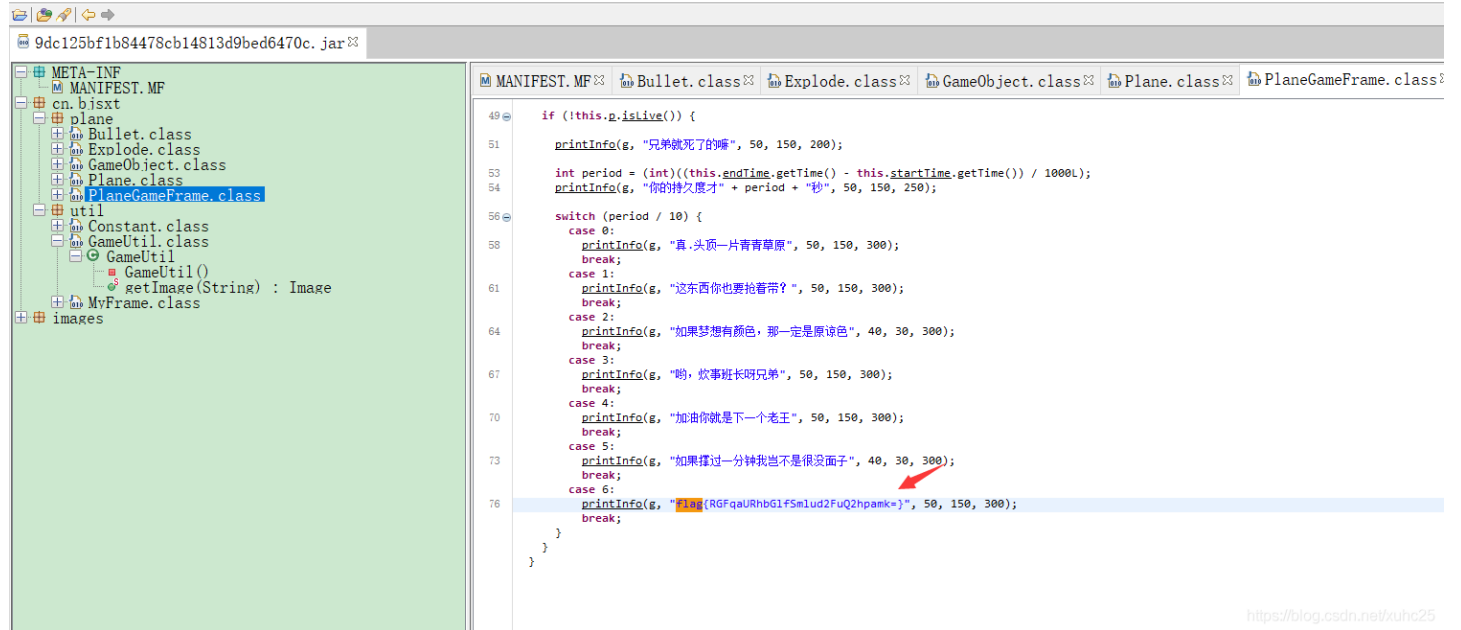


<https://blog.csdn.net/xuhc25>

看到还真能找到

PlaneGameFrame.class - Java Decompiler

File Edit Navigation Search Help




<https://blog.csdn.net/xuhc25>

flag{RGFqaURhbG1fSmlud2FuQ2hpamk=}

拿去提交, 发现是不对的

flag{RGFqaURhbGlfSmlud2FuQ2hpamk=}看着像base64加密，拿去解密。

 [随波逐流]CTF编码工具 V1.0 20201022

Base加解密 字符加解密 字符编码转换 已知key解密 进制转换 其他工

需要解密的文本 ↓

密钥(key) :

RGFqaURhbGlfSmlud2FuQ2hpamk=

解密结果 ↓

一键解码:

结果

base64解码: DajiDali\_JinwanChiji

base32解码:

base16解码:

<https://blog.csdn.net/xuhc25>


得到flag

flag{DajiDali\_JinwanChiji}

## 6、gif

 返回

 本题用时: 1天1时24分12秒

gif  40 最佳Writeup由不要让我起名提供

难度系数:  4.0

题目来源: 暂无

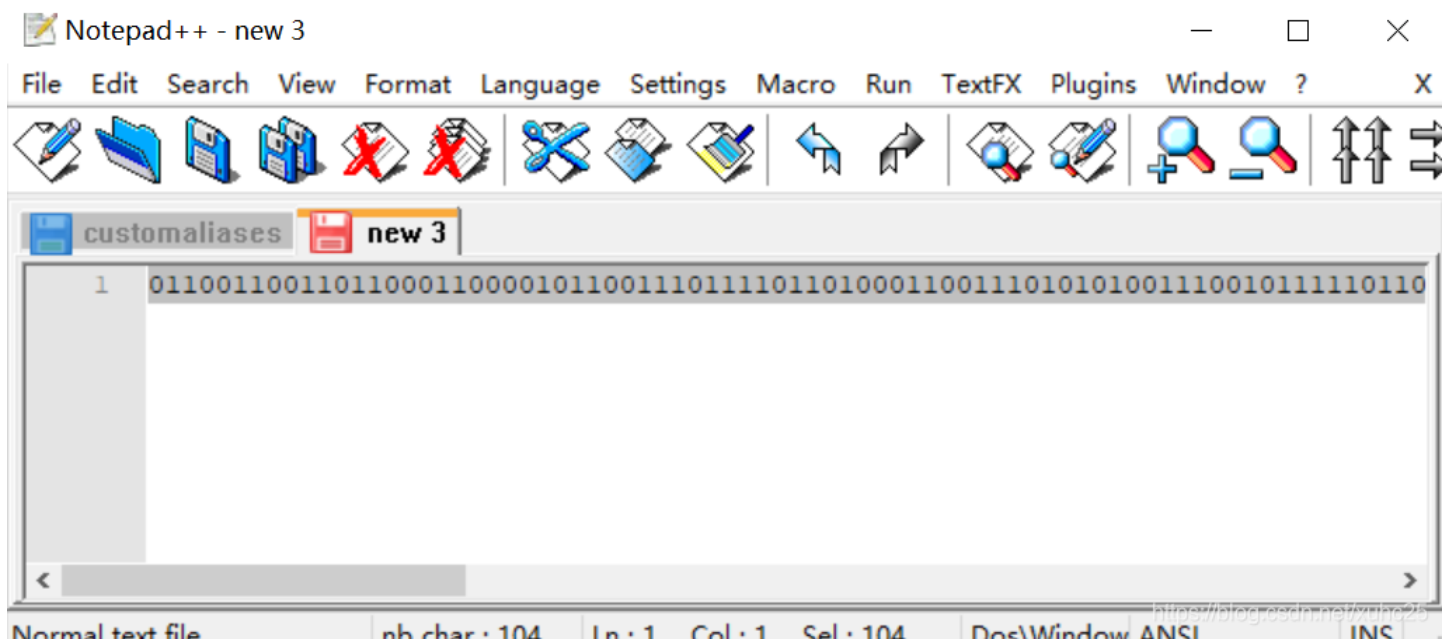
题目描述: 菜狗截获了一张菜鸡发给菜猫的动态图，却发现另有玄机

题目场景： 暂无

题目附件： 附件1

<https://blog.csdn.net/xuhc25>

下载下来，发现是个压缩包，打开看看，发现都是黑、白两色块。脑袋里啪的想到：白：0，黑：1，为啥不是白：1，黑：0？做久了开发就知道，0是白黑是1，1都是反派的那个。



一番比对后得到一串数字：

0110011001101100011000010110011101111011010001100111010101001110010111110110011101101001010001100111101

然后发现总数是104个，这要么是2、4、8一组，计算机中以字节为单位存储和解释信息,规定一个字节由八个二进制位构成，那么8个一组没跑了。

8个一组得到：

```
01100110 01101100 01100001 01100111 01111011 01000110 01110101 01001110 01011111 01100111 01101001
01000110 01111101
```

二进制转字符得到flag（用的随波逐流提供的解密工具）：flag{FuN\_giF}



\*\*

## 7、掀桌子

World of Attack&Defense

答题 竞赛 排行榜 队伍 商城

返回 本题用时: 1天2时22分50秒

掀桌子 👍 105 最佳Writeup由flag{not\_here} • 渣渣禹提供 WP 建议

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c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfabe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2, 生气地掀翻了桌子(°□°) ㄟ( ㄟ ㄟ

题目场景: 暂无

题目附件: 暂无

<https://blog.csdn.net/xuhc25>

这题没啥附件，只有一串字符，看字符是0-9，a-f组成的，就知道这是16进制字符串咯，那么2个16进制可以代表一个字符串，就用python进行转换下。

```
string = "c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfabe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2"
for i in range(0,len(string),2):
    s = '0x'+string[i]+string[i+1]
    i = int(s,16)
    print(str(i))
```

发现结果字符块都大于128，ASCII码范围是0-127

```
string = "c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfabe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2"
flag = ''
for i in range(0,len(string),2):
    s = '0x'+string[i]+string[i+1]
    i = int(s,16)
    print(str(i))
```

Search in Files Search Stack Data Debug I/O Exceptions Debug Console Watch Modules Python Shell Bookmarks Breakpoints

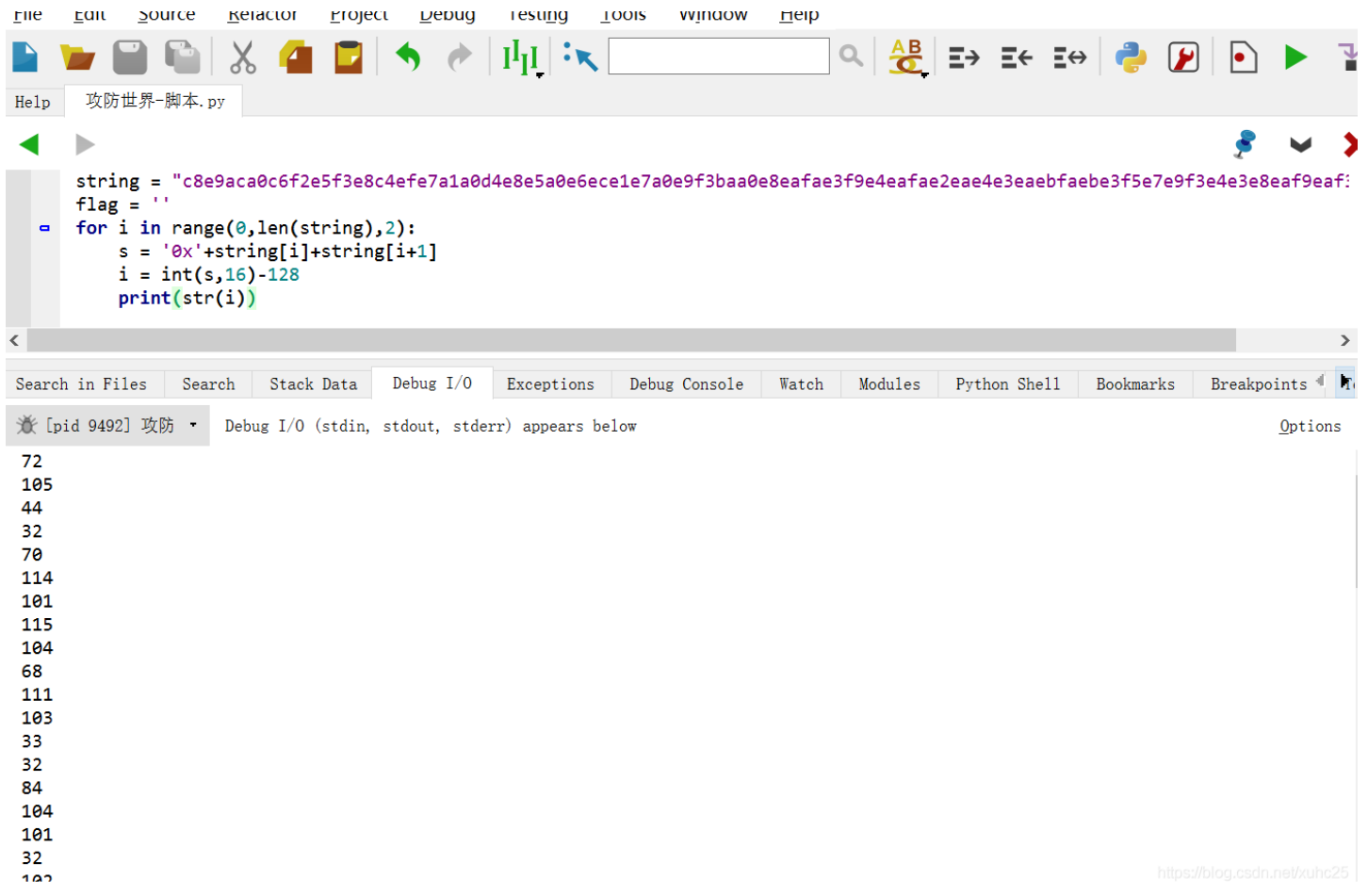
[pid 11068] 攻 Debug I/O (stdin, stdout, stderr) appears below Options

```
200
233
172
160
198
242
229
243
232
196
239
231
161
160
212
232
229
160
222
```

<https://blog.csdn.net/xuhc25>



怎么办呢？上面结果再减下128



```
string = "c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfabe3f5e7e9f3e4e3e8eaf9eaf9eaf3e2e4e6f2"
flag = ''
for i in range(0,len(string),2):
    s = '0x'+string[i]+string[i+1]
    i = int(s,16)-128
    print(str(i))
```

Debug I/O (stdin, stdout, stderr) appears below

```
72
105
44
32
70
114
101
115
104
68
111
103
33
32
84
104
101
32
100
```

<https://blog.csdn.net/xuhc25>

这就对了。跟着感觉走没毛病，把进制转换成字符。



```
string = "c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfabe3f5e7e9f3e4e3e8eaf9eaf9eaf3e2e4e6f2"
flag = ''
for i in range(0,len(string),2):
    s = '0x'+string[i]+string[i+1]
    i = int(s,16)-128
    flag+=chr(i)#chr是转换，flag+=的意思是拼接字符
print(flag)
```

Debug process terminated

Hi, FreshDog! The flag is: hjzcydjzbdckzkcgisdchjyjsbdfn

<https://blog.csdn.net/xuhc25>

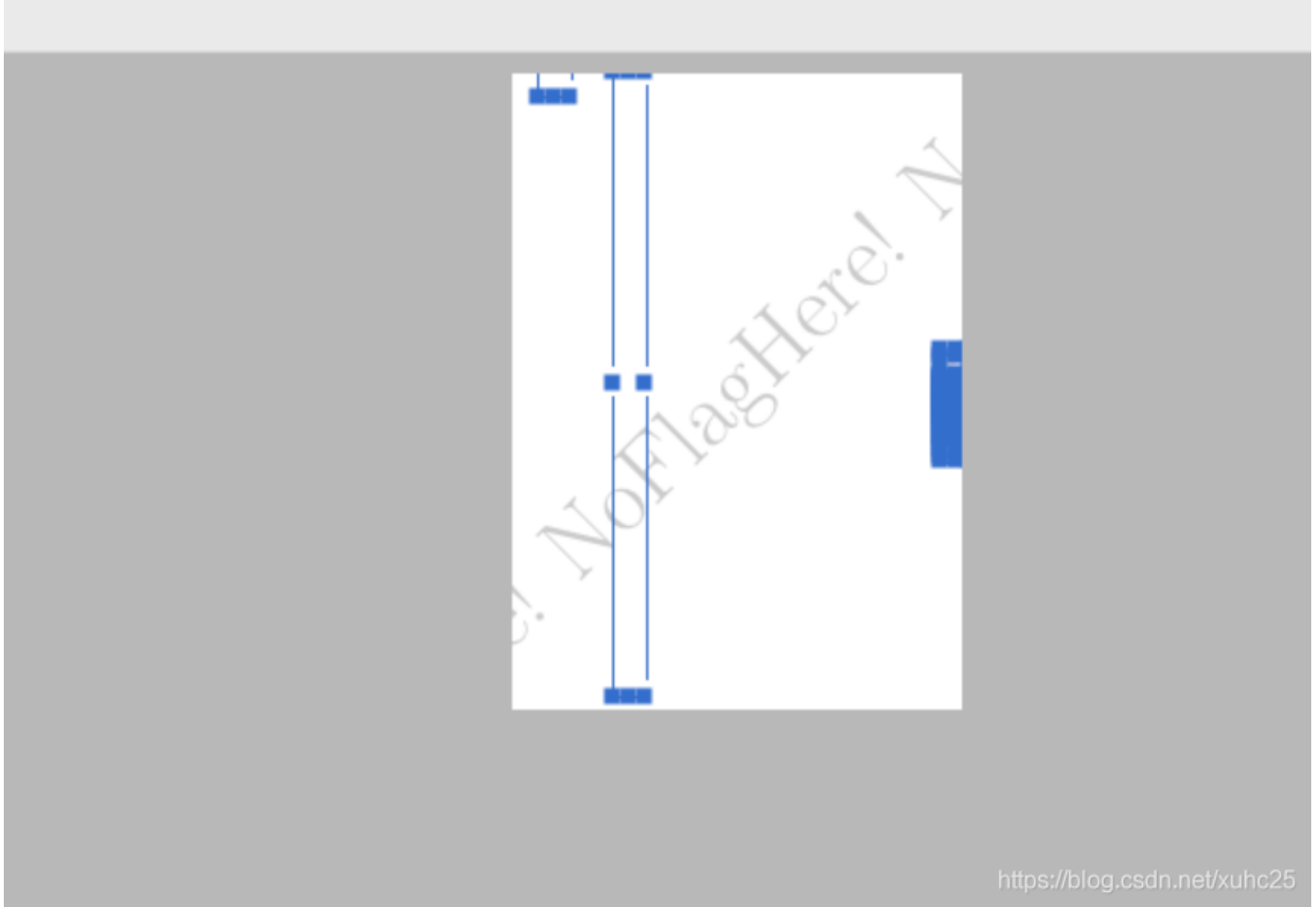
```
string = "c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfabe3f5e7e9f3e4e3e8eaf9eaf9eaf3e2e4e6f2"
flag = ''
for i in range(0,len(string),2):
    s = '0x'+string[i]+string[i+1]
    i = int(s,16)-128
    flag+=chr(i)
print(flag)
```



a felis quis, blandit porta ipsum. Donec sed nibh egestas, tristique mauris eu, rutrum justo. Nulla facilisi. Duis gravida semper dui laoreet vulputate. Aenean quis tempor orci. Cras placerat lectus nulla, eu bibendum metus interdum in. Cras placerat lectus nulla, eu bibendum metus interdum in. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras faucibus odio ut metus vulputate, id laoreet magna volutpat. Integer nec enim vel arcu porttitor egestas. Vestibulum suscipit lorem sed sem faucibus rutrum. Nunc diam orci, convallis vitae auctor vehicula, interdum ut mi. Maecenas nec urna at dolor mattis dictum sit amet at orci. Mauris condimentum adipiscing erat nec feugiat. Curabitur scelerisque varius ligula, faucibus adipiscing dui. Duis eget ullamcorper arcu. In facilisis et tortor commodo aliquam. Nulla feugiat, sem eu molestie bibendum, leo nisi porttitor massa, id accumsan sapien libero id tellus. In enim lacus, sollicitudin a felis quis blandit porta ipsum. Donec sed nibh egestas, tristique mauris eu, rutrum justo. Nulla facilisi. Duis gravida semper

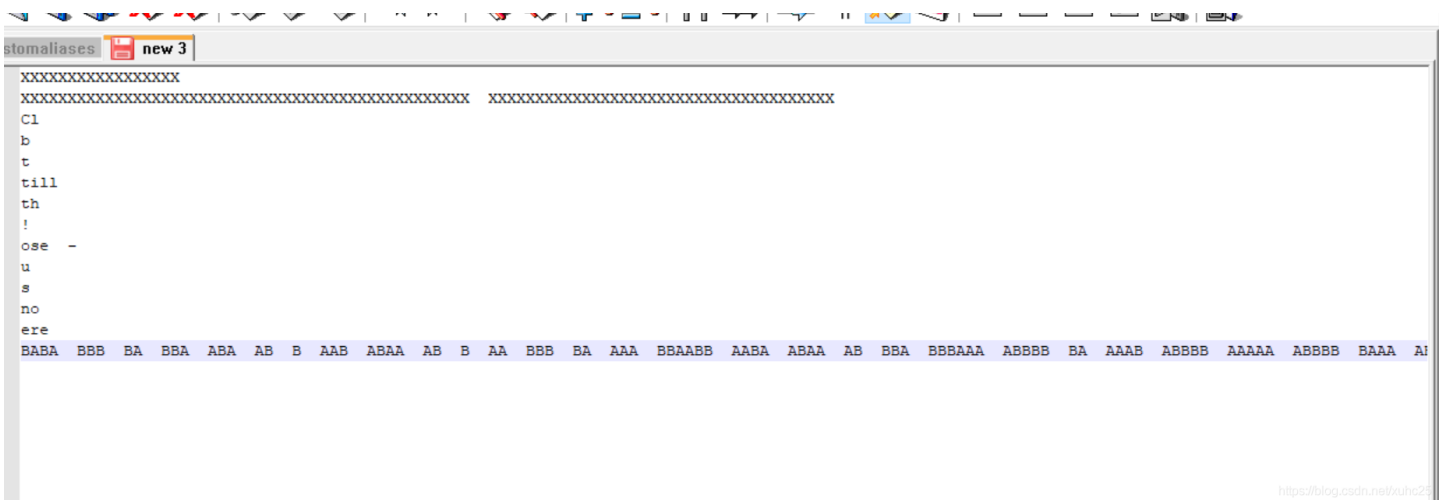
https://blog.csdn.net/xuhc25

PDF编辑器，删掉挡在那对英文，看到还剩下几个文本框。



https://blog.csdn.net/xuhc25

复制文本出来，贴在记事本上

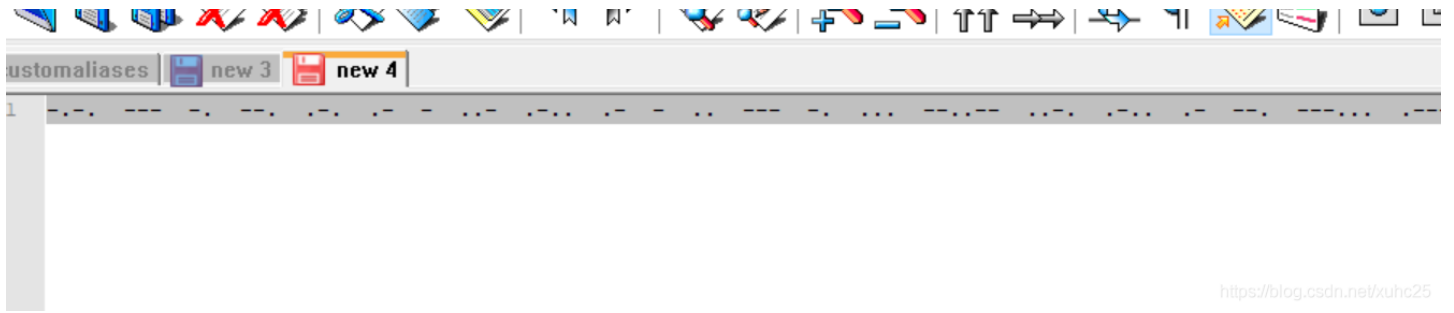


https://blog.csdn.net/xuhc25

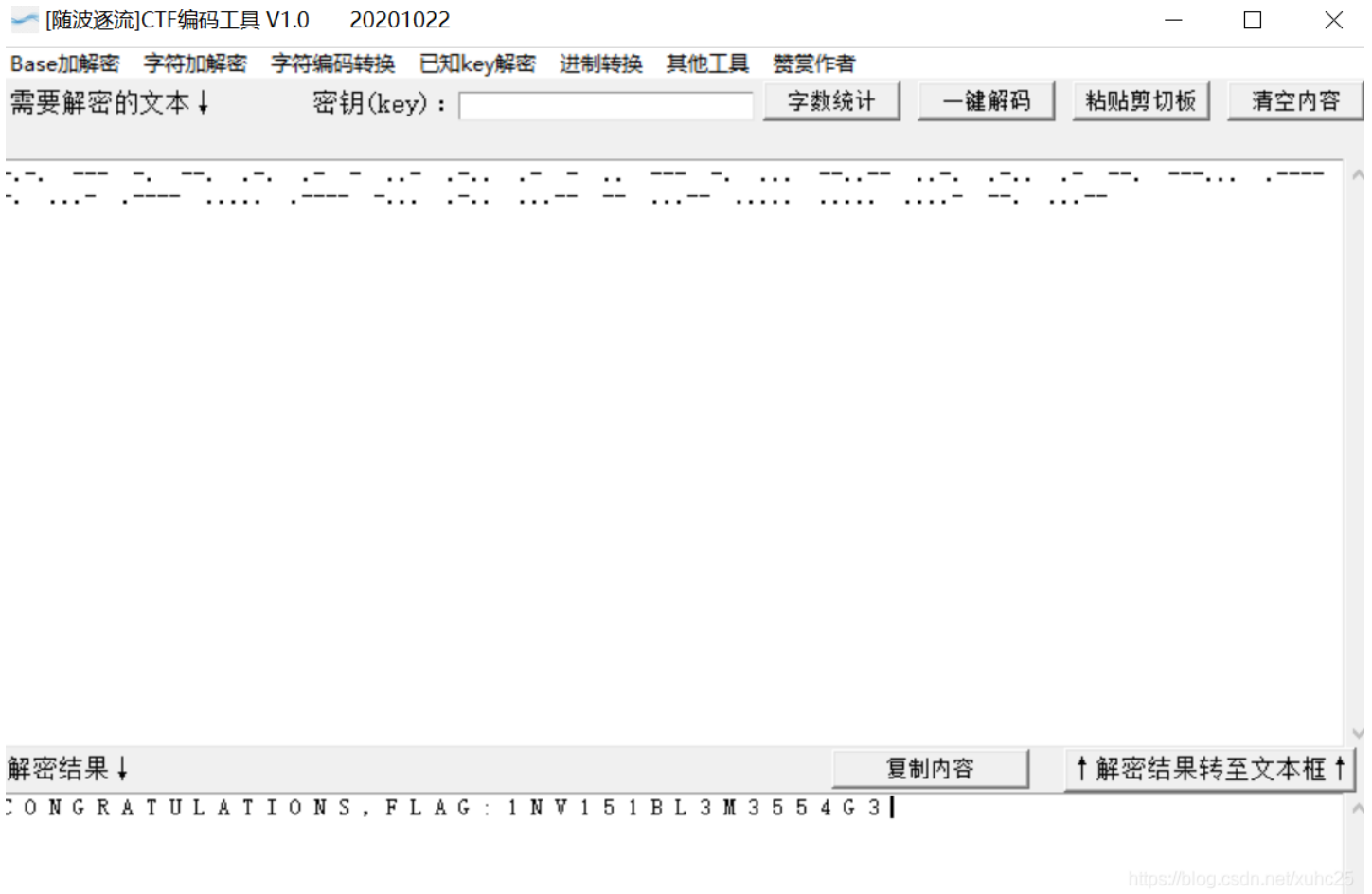
很明显，ABAB那串就是密文了。复制出来

```
BABA BBB BA BBA ABA AB B AAB ABAA AB B AA BBB BA AAA BBAABB AABA ABAA AB BBA BBBAAA AB BBB BA AAAB AB BBB AAAAA AB BBB BAAA ABAA AAAB BB AAAB AAAAA AAAAA AAAAB BBA AAAB
```

试下转换为摩斯密码，记事本替换下A和B，A替换为.，B替换为-。  
到如下结果。



放到解密工具上解密



去掉空格，得到一串字符

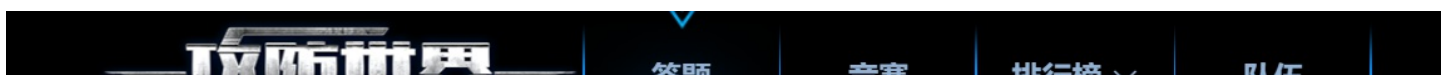
CONGRATULATIONS,FLAG:1NV151BL3M3554G3

题目说flag是小写，转换为小写，得到flag。

congratulations,flag:1nv151bl3m3554g3

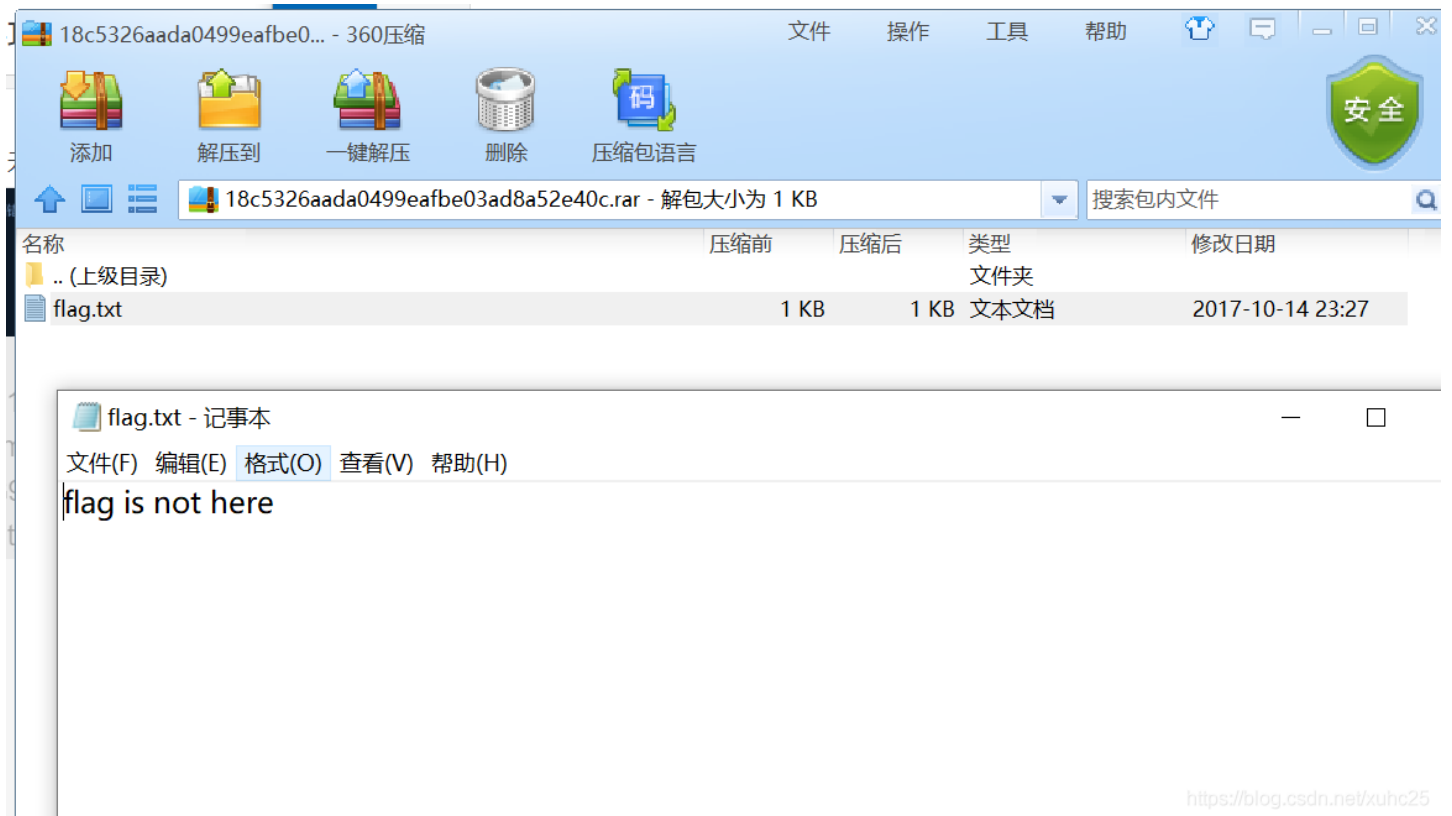
那么flag就是flag{1nv151bl3m3554g3}

## 9、SimpleRAR(比较麻烦)





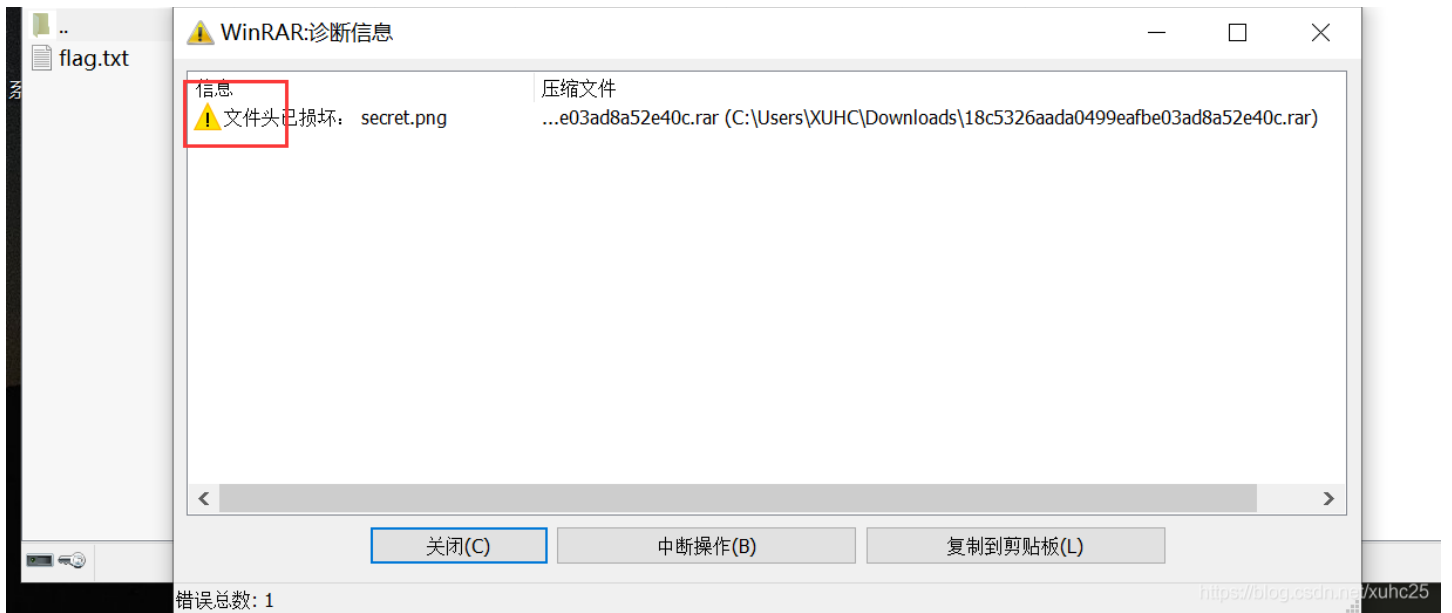
下载是1个压缩包，360解压



当然是用rar啦，360压缩不会提示错误的，winrar才会。







文件头损坏，那就不用winhex打开修复。

首先了解下rar块头的知识：

## 压缩文件头块

第二块为压缩文件头(MAIN\_HEAD)，和标记块一样

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	UTF-8
00000000	52	61	72	21	1A	07	00	CF	90	73	00	00	0D	00	00	00	Rar!���6 s�� ��
00000016	00	00	00	00	D5	56	74	20	90	2D	00	10	00	00	00	10	���� � ����
00000032	00	00	00	02	C7	88	67	35	6D	EB	4E	4B	1D	30	08	00	����j g6m� ���
00000048	20	00	00	00	66	6C	61	67	2E	74	78	74	00	B0	57	00	���flag.txt��
00000064	43	66	6C	61	67	20	69	73	20	6E	6F	74	20	68	65	72	flag is not her
00000080	65	A8	3C	74	20	90	2E	00	3A	15	00	00	42	16	00	00	e� � ���B���
00000096	02	BC	E9	8C	2F	6E	84	4F	4B	1D	33	0A	00	20	00	00	�� n� 3 � ��
00000112	00	73	65	63	72	65	74	2E	70	6E	67	00	F0	40	AB	18	�secret.png��
00000128	11	C1	11	55	08	D1	55	80	0D	99	C4	90	87	93	22	19	�� � � �

标记块

压缩文件头  
CRC校验值

头类型

位标记

文件头大小

保留字节1

保留字节2

[https://blog.csdn.net/Glaming\\_D](https://blog.csdn.net/Glaming_D)

这里的头类型是0x73表示压缩文件头块，位标记为0x0000 没有位被置为1，如果块头被加密，则位标记应该为：0x8000，文件头大小为0x0D00，由上图可以看出这个压缩文件头块占13个字节。

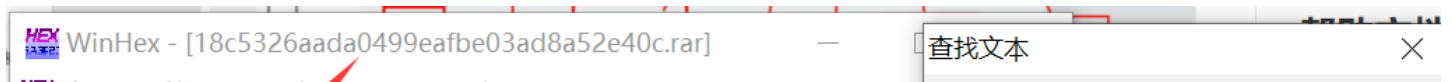
<https://blog.csdn.net/xuhc25>

在这里怎么定位坏的块呢？

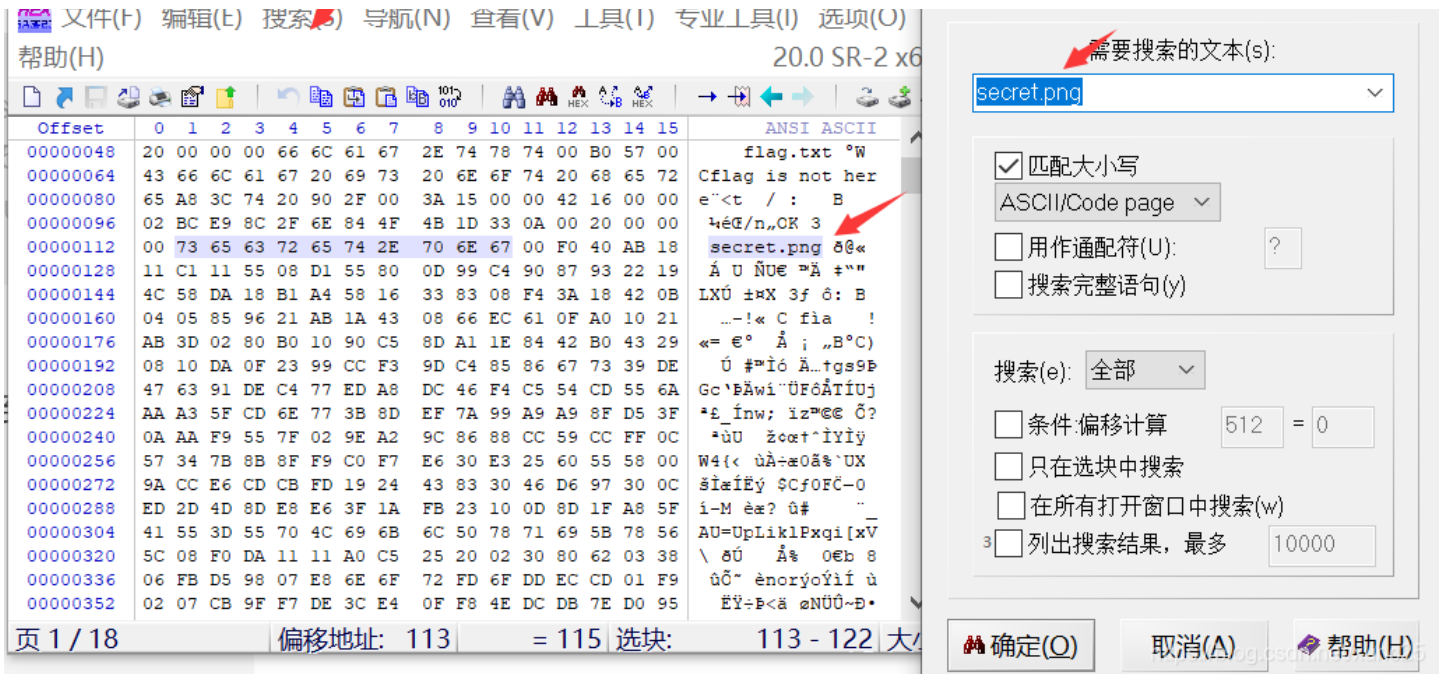
1、定位文件位置

rar提示secret.png文件头已损坏，winhex先定位到文件所在。

通过搜索文本，可以定位到文件所在了。







题目用到的是这两个知识点

## 文件头块

在网上找了图片资源，然后我再举个例子，由于现在rar版本不断更新，图片资源有点老了，部分内容现在可能遇不到。

HEAD\_CRC      2 字节      从 HEAD\_TYPE 到 FILEATTR 的 CRC 结构和文件名

HEAD\_TYPE      1 字节      头类型: 0x74

HEAD\_FLAGS      2 字节      位标记:

0x01 - 文件在前一卷中继续

0x02 - 文件在后一卷中继续

0x04 - 文件使用密码加密

0x08 - 文件注释存在

RAR 3.x 使用分开的注释块，不设置这个标记。

0x10 - 前一文件信息被使用(固实标记)

(对于 RAR 2.0 和以后版本)

765 位(对于 RAR 2.0 和以后版本)

[https://blog.csdn.net/qq\\_34374601/article/details/104000000](https://blog.csdn.net/qq_34374601/article/details/104000000)

附上常见的块类型(HEAD\_TYPE)如下:

标记块: HEAD\_TYPE=0x72  
 压缩文件头: HEAD\_TYPE=0x73  
 文件头: HEAD\_TYPE=0x74  
 旧风格的注释头: HEAD\_TYPE=0x75  
 旧风格的用户身份信息: HEAD\_TYPE=0x76  
 旧风格的子块: HEAD\_TYPE=0x77  
 旧风格的恢复记录: HEAD\_TYPE=0x78  
 旧风格的用户身份信息: HEAD\_TYPE=0x79  
 子块: HEAD\_TYPE=0x7A  
 最后的结束块: HEAD\_TYPE=0x7B

## 2、找到块头

上一个文件结束，是 flag is not here，就是那个迷惑的txt文件内容结尾啦。那么跟着的就是下个文件，图片文件的开头。看下图，从上面知识，知道第三个字节开始，就是文件块头咯，那么我们要改的就是这个字节。提示文件头损坏，那就给他改成对应的文件头: HEAD\_TYPE=0x74

**文件头块**

在网上找了图片资源，然后我再举个例子。由于现在rar版本不断更新，图片资源有点老了，部分内容现在可能看不到。

HEAD_CRC	2 字节	从 HEAD_TYPE 到 FILEATTR 的 CRC 结构和文件名
HEAD_TYPE	1 字节	头类型: 0x74
HEAD_FLAGS	2 字节	位标记:

- 0x01 - 文件在前一卷中继续
- 0x02 - 文件在后一卷中继续
- 0x04 - 文件使用密码加密
- 0x08 - 文件注释存在
- RAR 3.x 使用分开的注释块，不设置这个标记。
- 0x10 - 前一文件信息被使用(固实标记)  
(对于 RAR 2.0 和以后版本)

765 位(对于 RAR 2.0 和以后版本)

Hex view data (offset 00000064): 43 66 6C 61 67 20 69 73 20 6E 6F 74 20 68 65 72

改完后保存，如下图

WinHex - [18c5326aada0499eafbe03ad8a52e40c .rar]

文件(F) 编辑(E) 搜索(S) 导航(N) 查看(V) 工具(T) 专业工具(I) 选项(O) 窗口(W) 帮助(H)

20.0 SR-2 x64

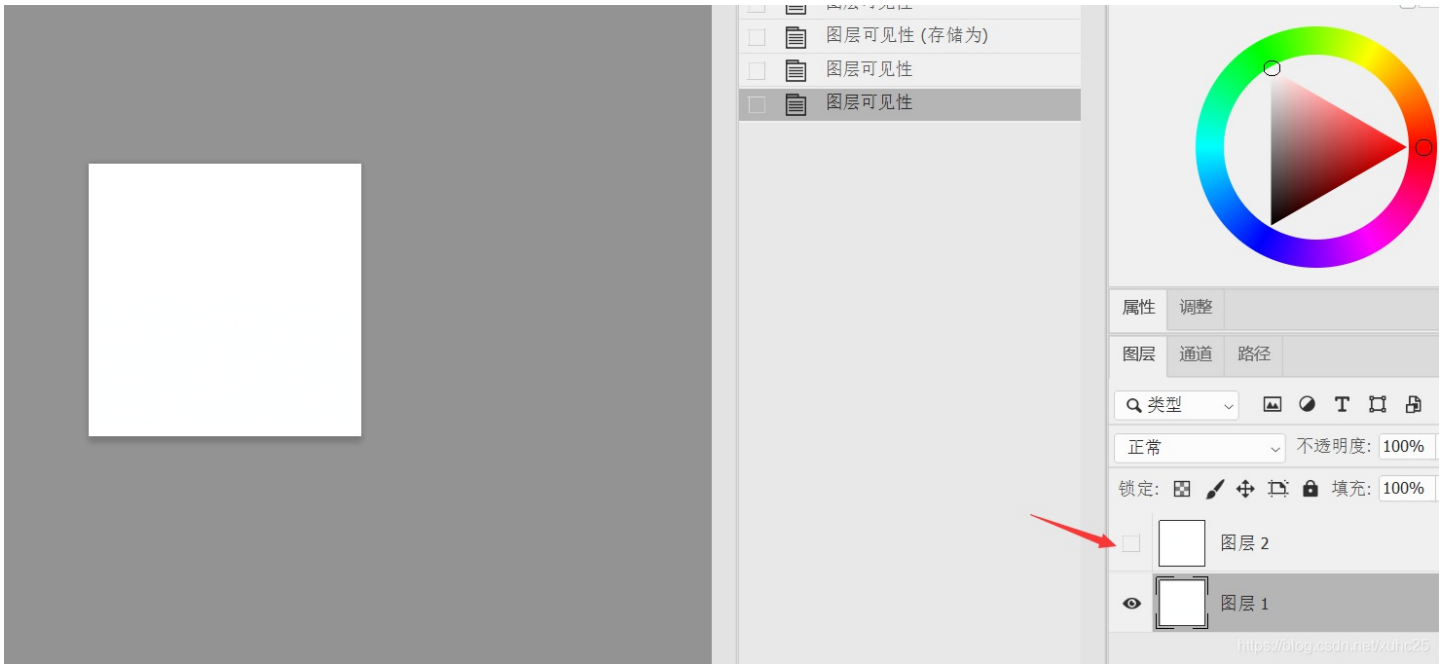
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
00000000	52	61	72	21	1A	07	00	CF	90	73	00	00	0D	00	00	00	Rar! i s
00000016	00	00	00	00	D5	56	74	20	90	2D	00	10	00	00	00	10	0vt -
00000032	00	00	00	02	C7	88	67	36	6D	BB	4E	4B	1D	30	08	00	Ç^g6m»NK 0
00000048	20	00	00	00	66	6C	61	67	2E	74	78	74	00	B0	57	00	flag.txt °W
00000064	43	66	6C	61	67	20	69	73	20	6E	6F	74	20	68	65	72	Cflag is not here
00000080	65	A8	3C	74	20	90	2F	00	3A	15	00	00	42	16	00	00	e<t / : B
00000096	02	BC	E9	8C	2F	6E	84	4F	4B	1D	33	0A	00	20	00	00	4éG/n„OK 3
00000112	00	73	65	63	72	65	74	2E	70	6E	67	00	F0	40	AB	18	secret.png δ@«
00000128	11	C1	11	55	08	D1	55	80	0D	99	C4	90	87	93	22	19	Á U ÑUE ¢Ä +""
00000144	4C	58	DA	18	B1	A4	58	16	33	83	08	F4	3A	18	42	0B	LXÜ ±xX 3f ô: B
00000160	04	05	85	96	21	AB	1A	43	08	66	EC	61	0F	A0	10	21	...!« C fia !
00000176	AB	3D	02	80	B0	10	90	C5	8D	A1	1E	84	42	B0	43	29	«= €° Ä ; „B°C)
00000192	08	10	DA	0F	23	99	CC	F3	9D	C4	85	86	67	73	39	DE	Ú #°Ió Ä..tgs9P
00000208	47	63	91	DE	C4	77	ED	A8	DC	46	F4	C5	54	CD	55	6A	Gc`PÄwi`ÜFóÄTÍUj
00000224	AA	A3	5F	CD	6E	77	3B	8D	EF	7A	99	A9	A9	8F	D5	3F	*É ínW; iz™@E Ó?
00000240	0A	AA	F9	55	7F	02	9E	A2	9C	86	88	CC	59	CC	FF	0C	*Ü zœt`iYiÿ
00000256	57	34	7B	8B	8F	F9	C0	F7	E6	30	E3	25	60	55	58	00	W4{< ùÄ=α0Ä\$`UX
00000272	9A	CC	E6	CD	CB	FD	19	24	43	83	30	46	D6	97	30	0C	šIäIÿ \$CfOFÇ-0
00000288	ED	2D	4D	8D	E8	E6	3F	1A	FB	23	10	0D	8D	1F	A8	5F	i-M èæ? ú#
00000304	41	55	3D	55	70	4C	69	6B	6C	50	78	71	69	5B	78	56	AU=ÜpLlklPxqi[xV

页 1 / 18 偏移地址: 83 = 116 选块: 无 大小: 无

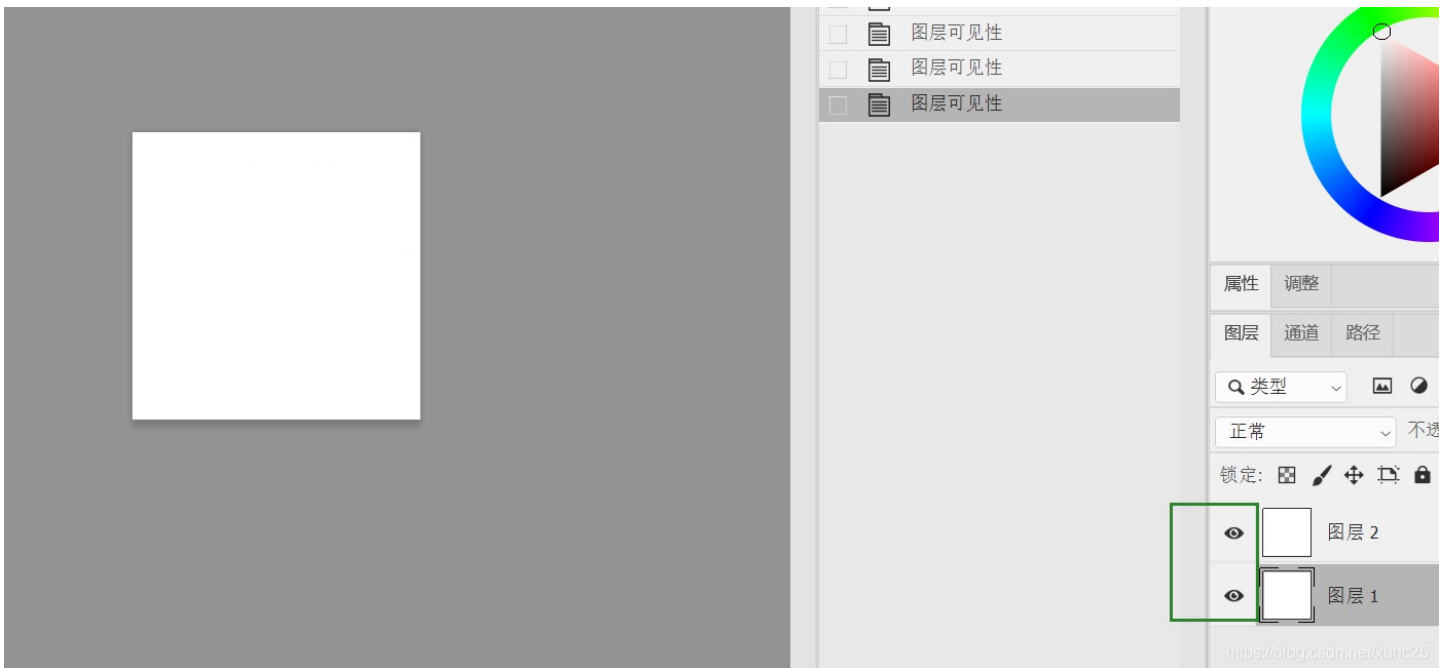




题目说是分层图片，ps打开，发现有图层是隐藏的。



开启图层再另存为。



用stegsolve打开，得到半截二维码。



两半二维码拼合



还是缺定位点没有，ps上去，得到完整的二维码。



扫码得到flag

flag{yanji4n\_bu\_we1shi}

## 10、base64stego

返回 本题用时: 26分57秒

**base64stego** 👍 127 最佳Writeup由CTFshow • zEr0\_0提供

难度系数: ★★★★★ 5.0

题目来源: olympicCTF

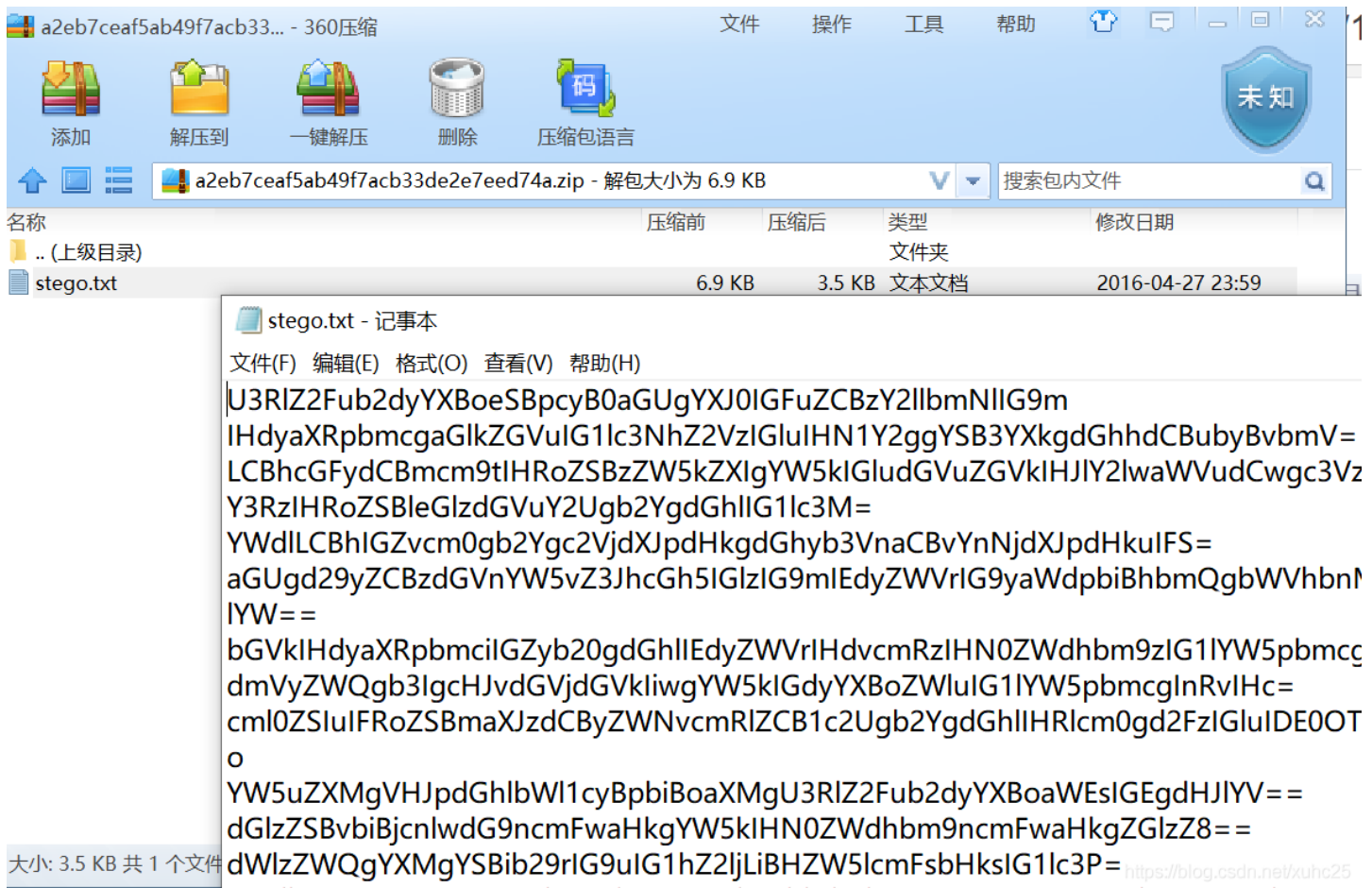
题目描述: 菜狗经过几天的学习, 终于发现了如来十三掌最后一步的精髓

题目场景: 暂无

题目附件: 附件1

<https://blog.csdn.net/xuhc25>

下载完提示有密码, 但是这是伪加密, 360压缩可以无视。  
360压缩打开压缩包是这么一大串东西



base64解密



```

98 ZXItcHJvcGFnYW5kYS4gSW4gMTk2OCwgY3JldyBtZWliZW==
99 cnMgb2YgdGhlIFVtUyBqdWVibG8gKEFHRVItMikgaW50ZWxsaWdlbmNlIHNoaXAgAGVsZ
100 aXNvbmVycyBieSB0b3J0aCBLb3JlYSwgY29tbXVuaWNhdGVkIGluIHNPZ25=
101 IGxhbmdlYWdlIGR1cm1uZyBzdGFuZG9gcGhvdG8gb3Bwb3JO
102 dW5pdG11cywgaW5mb3JtaW5nIHRoZSBVbml0ZWQgU3RhdGVzIHRoZXXkg
103 d2VyZSBub3QgZGVmZWNo3JzIGJldCBYXXRoZXIgd2VyZSBiZWluZyBoZWxkIGNh
104 cHRpdmluYnkgdGh1IE5vcnRoIEtvcmluIEluIG90aGVyIHBob3Rv
105 cyBwcmVzZW50ZWQgdG8gdGhlIFVtLCBjcmV3IG11bWJlcnMgZ2F2ZSAidGh1IGZpbmdlc
106 dGh1IHVuc3VzcGVjdGluZyB0b3J0aCBLb3JlYW5zLCBpbihhbiBhdHR1bXB0IHRvIE==
107 ZG1zY3JlZG10IHBob3RvcyB0aGF0IHNo3d1ZCB0aGVtIHNaQ==
108 bGluZyBhbmQyY29tZm9ydGFibGUuUHQeNCi0tDQpodHRwO18vZW4ud2lraXB1ZG1hLm9yZ
109 L3dpa2kvU3R1Z2Fub2dyYXBoeQ0k
110

```

```

1 Steganography is the art and science of writing hidden messages in
such a way that no one, apart from the sender and intended
recipient, suspects the existence of the message, a form of security
through obscurity. The word steganography is of Greek origin and
means "concealed writing" from the Greek words steganos meaning
"covered or protected", and graphein meaning "to write". The first
recorded use of the term was in 1499 by Johannes Trithemius in his
Steganographia, a treatise on cryptography and steganography
disguised as a book on magic. Generally, messages will appear to be
something else: images, articles, shopping lists, or some other
covert text and, classically, the hidden message may be in invisible
ink between the visible lines of a private letter.
2
3 The advantage of steganography over cryptography alone is that

```

就一段隐写技术的说明，没啥用。

题目中说base64解密，最后一步精髓（最后一步是精髓，不就是提示你取行尾的密文），所以提示很明显啦，base64隐写。这篇文章说的很好啦，base64隐写原理。

<https://blog.csdn.net/xnighmare/article/details/103774379>

然后解题思路是，边解密边把隐写的密文拿到再解密。

依次读取每行，从中提取出隐写位。

如果最后没有‘=’，说明没有隐写位，跳过。

如果最后是一个‘=’，说明有两位隐写位，将倒数第二个字符转化为对应的二进制索引，然后取后两位。

如果最后是两个‘=’，说明有四位隐写位，将倒数第三个字符转化为对应的二进制索引，然后取后四位。

将每行提取出的隐写位依次连接起来，每8位为一组转换为ASCII字符，最后不足8位的丢弃。

贴上大佬的代码：

```

# -*- coding: cp936 -*-
import base64
b64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/' #base64字典库
with open(r'C:\Users\XUHC\Documents\代码段\stego.txt', 'rb') as f: #读取文件内容
    bin_str = ''
    for line in f.readlines():
        stegb64 = str(line, "utf-8").strip("\n")
        rowb64 = str(base64.b64encode(base64.b64decode(stegb64)), "utf-8").strip("\n")
        offset = abs(b64chars.index(stegb64.replace('=', ''))[-1]) - b64chars.index(rowb64.replace('=', ''))[-1])
        equalnum = stegb64.count('=') #no equalnum no offset
        if equalnum:
            bin_str += bin(offset)[2:].zfill(equalnum * 2)
    print(''.join([chr(int(bin_str[i:i + 8], 2)) for i in range(0, len(bin_str), 8)])) #8 位一组

```

```

3 b64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/' #base
4 with open(r'C:\Users\XUHC\Documents\代码段\stego.txt', 'rb') as f: #读取文件内容
5     bin_str = ''
6     for line in f.readlines():
7         stegb64 = str(line, "utf-8").strip("\n")
8         rowb64 = str(base64.b64encode(base64.b64decode(stegb64)), "utf-8").strip()
9         offset = abs(b64chars.index(stegb64.replace('=','')[-1]) - b64chars.index(ro
10        equalnum = stegb64.count('=') #no equalnum no offset
11        if equalnum:
12            bin_str += bin(offset)[2:].zfill(equalnum * 2)
13        print(''.join([chr(int(bin_str[i:i + 8], 2)) for i in range(0, len(bin_str), 8)

```

with open(r'C:\Users\XUHC\Docum... > for line in f.readlines())

un: main x

C:\Users\XUHC\AppData\Local\Programs\Python\Python39\python.exe C:/Users/XUHC/AppData/Local/Temp/攻防世界-脚本1.py/main.py

Base\_sixty\_four\_point\_five

Process finished with exit code 0

<https://blog.csdn.net/xuhc25>

得到flag

Base\_sixty\_four\_point\_five

## 11、ext3

World of Attack&Defense

答题 竞赛 排行榜

返回 本题用时: 10分5秒

ext3 116 最佳Writeup由hackcat提供

难度系数: ★★★★★★ 6.0

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

题目场景: 暂无

题目附件: 附件1

<https://blog.csdn.net/xuhc25>

提示很明显: ext3文件系统, Linux环境。

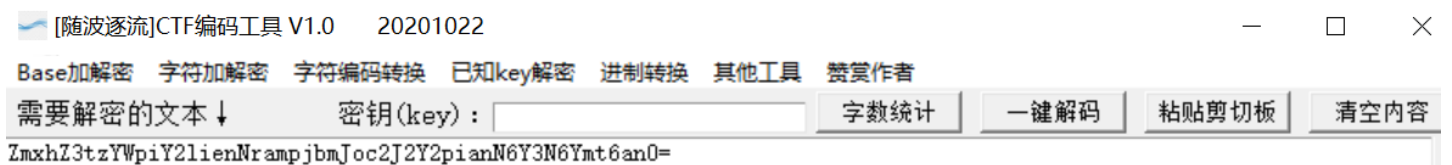
直接mount试试看, 然后再进入到目录, 试着查找flag文件, 没想到还真有, 查看flag.txt, 得到密文:

ZmxhZ3tzYWpiY2lienNrampjbmJoc2J2Y2pianN6Y3N6Ymt6an0=

```
文件(F) 动作(A) 编辑(E) 查看(V) 帮助(H)
(root@kali)~[~/桌面]
# mount f1fc23f5c743425d9e0073887c846d23 /mnt
(root@kali)~[~/桌面]
# cd mnt
cd: 没有那个文件或目录: mnt
(root@kali)~[~/桌面]
# cd /mnt
1 x
(root@kali)~/mnt]
# ls
02CdWGSxGPX.bin  8A2MFawD4  ix1EMRHRpIc2  n  r
0GY1l  8DQFirm0D  j6uLMX  NgzQPW  Raf3SYj
0h3a5  8HhWfV9nK1  jE  Nv  rhZE1LZ6g
0l  8nwg  jj  o  Ruc9
0qsd  8RxQG4bvd  KxEQM  07avZhikgKgbF  RZT0Gd
0wDq5  FinD  LG6F  o8  scripts
0Xs  fm  Lh  00o0s  sdb.cramfs
1  g  LLC6Z0zrgy.bin  orcA  sn
2X  gtj  L00J8  oSx2p  SPaK8l2sYN
3  h  lost+found  OT  SrZznhsAJ
3J  H  LvuGM  poiuy7Xdb  t
44aAm  H2Zj8FNbu  lWIRfzP  px6u  T
4A  hdi7  m  Q  TFGV0SwYd.txt
6JR3  hYuPvID  m9V0lIaElz  qkCN8
6wUaZE1vbsW  i  MiU  QmUY1d
7H7geLLS5  imgLDpt4BY  Mnuc  QQY3sF63w
(root@kali)~/mnt]
# find -name flag
(root@kali)~/mnt]
# find -name flag.*
./07avZhikgKgbF/flag.txt
(root@kali)~/mnt]
# cat ./07avZhikgKgbF/flag.txt
ZmxhZ3tzYWpiY2lienNrampjbmJoc2J2Y2pianN6Y3N6Ymt6an0=
(root@kali)~/mnt]
#
```

<https://blog.csdn.net/xuhc25>

解密后得到flag



解密结果 ↓ 复制内容 ↑ 解密结果转至文本框 ↑

一键解码: 结果  
 base64解码: `flag{sajbcibzskjjcnbhsbvcjbjzscszbkj}`  
 base32解码:

<https://blog.csdn.net/xuhc25>

flag{sajbcibzskjjcnbhsbvcjbjzscszbkj}

## 12、功夫再高也怕菜刀

← 返回
本题用时: 6时4分49秒

### 功夫再高也怕菜刀

👍 37
最佳Writeup由 **B301** • dals提供

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

题目场景: 暂无

题目附件: 附件1

360安全大脑提醒您

i 您正在检测的文件包含加密压缩包，请输入解压密码进行安全检测。

**acfff53ce3fa4e2bbe865...**

显示密码

确定
取消

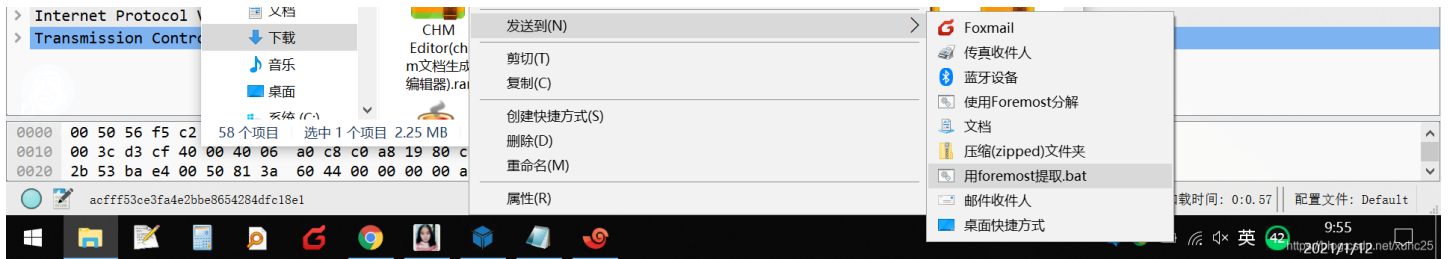
<https://blog.csdn.net/xuhc25>

附件后缀很奇怪：acfff53ce3fa4e2bbe8654284dfc18e1.pcapng  
 pcap是wreshark流量的抓包，png是图片，从这两方面下手了。

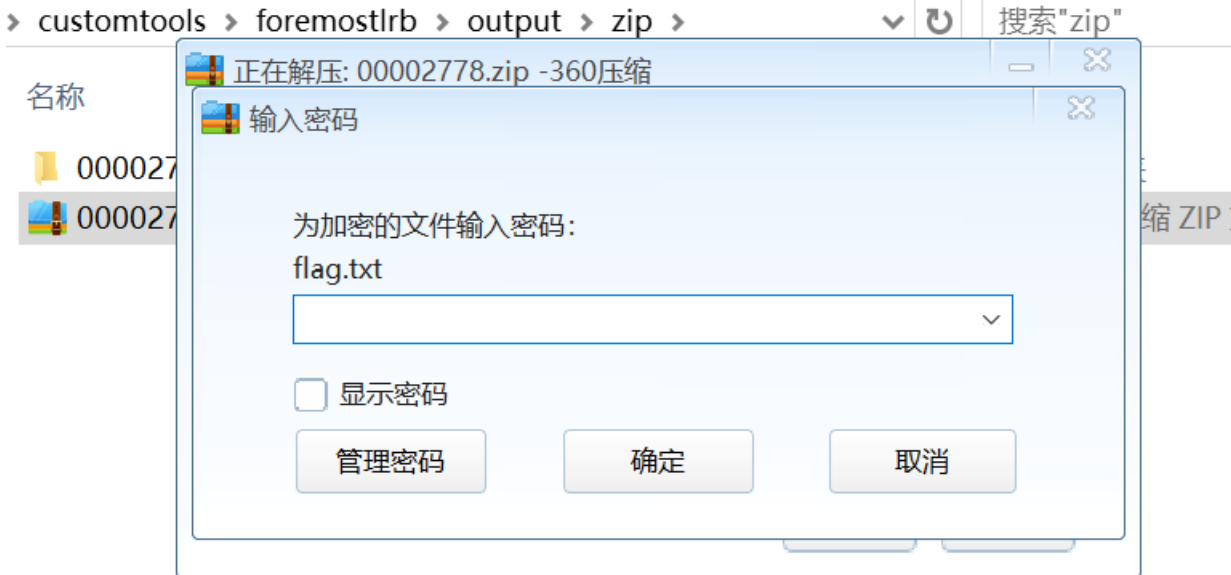
1、万变不离其宗，先分解提取。

foremost 提取

The screenshot shows a Windows desktop environment. In the foreground, a file explorer window is open, displaying a folder named 'acfff53ce3fa4e2bbe8654284dfc18e1.zip'. A context menu is open over this file, showing various actions such as '添加到压缩文件(A)...', '添加到 "acfff53ce3fa4e2bbe8654284dfc18e1.zip" (T)', and '其他压缩命令'. In the background, a Wireshark window is visible, showing a list of network traffic packets. The details pane on the right shows a TCP segment with 'MSS=1460', 'SACK\_PERM=1', and 'TSval=32852...'. The interface is in Chinese.



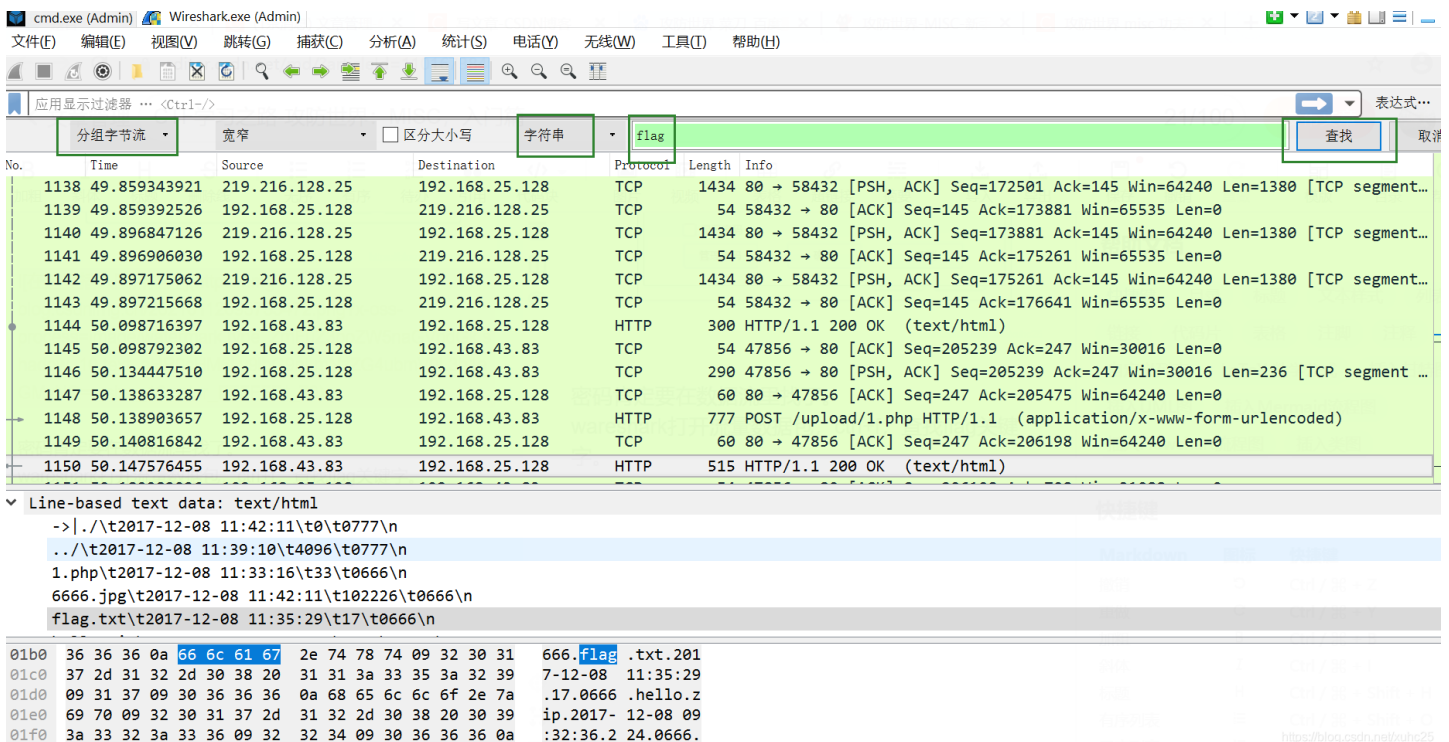
得到压缩包，解压要密码，确认不是伪加密后，进行下一步。



<https://blog.csdn.net/xuhc25>

密码估计要在数据流里找了。

Wireshark打开流量数据包，ctrl+f，查找flag关键字。



根据题目提示，或许会跟png有关，找了一圈没有，发现有个6666.jpg文件，追踪流-tcp数据流，发现可疑的东西







```
0D8FFF0040155A4EABFF0060C97F9BD76D35B79E9B2F25D7D34F92D526A5C7525A6D7D2FF8297FEDD66EDAB7292B36B9684
84301B49954C876E724DEDD9FBCE47FCF08893C93CFD58E2AB92436EFDEA993F78464FDB2EB90B12918630C59C1C7CA474C
6E515607DEB3FF00AF5B8FE52D578BFE61FF00F5CE7AECA6AED5FB2FC7D9FF00F27AF751D7E295F9A52B5FADBBF9738FBFD
CF95FFBB1E5AD211F3EFF0098061F68653CCD3E014B58881FEAD38DFB41C633D90567C843799E63908306F1D78C95FF0055
650FAFDD01B1D31CF09CDE1F76C3FEBACDFF00A10ACDFF0097783FEBFDFF00F65AEDA6B48F5BD9EBE6A0FE7F1EADFEF759
B6B96A371BABBDD2EDF69EAB7B7C1A745EE696824EACEBBE40F2E12487E72E3836303C525BF911631B65B88679619003931
4AF1B0C33815E56EC7F74DE5E71938B4B53CE31FC53CA08E4E09DDD8BE04D27FAA97FEC27FD64AA97BFF002FDFF5F917FEC
D5DD4A0AEDD97335772B6AF953B27FF0080E9D9C9B56B47979E527CA936DA8D9455F45774AF65AA4AF2D5755149B779B954
9189DA41F2C843E483D2D2D8925E67FF00A6D367823E63D41F996B3252C4A6C014E18DB8603F7518CF9B792F6F30F3B4B1C
82011C0506FDE74BEFAC1FCAB3EE7ADFF00FD7083F9C75D0FDD83B74D7D6DCDE565FC3EDBB5FC91B73D2F7A4BCF937D6DCF
EC97CECAADB5DD295FE393322E24C6C1183925BECCAC4F5E7CDBC9BD48DADB41C018E32139C694801447F38DCDE486FF009
6F3747B997B08D3076EEF61D9C56A5DFDFB8FFB0745FF00B4EB1E6EABFF0060E7FE42BCCAD75A76F2B6B74AE97CB4E8BDD4
EE9352F630E95A3E76FC545FFEDD2DBCADAB392E4CD99B0A31FBD05FE4CF2D79723EF48D93930C648EBC608CE0935893BE7
71399079983839FB4DD374553FF003CA33D80E79EEC00D49BEF41FF005E32FF0027ACA3D6D3FEB85CFF00296BC6ACF5F9A5
F7FB3FFE4FE6959FC53E6F770E96AFB5FCB6E67D366DC3A689B6D2D236C8BA9000DB89705BF798CE679C1F96043FF3CE3EA
5BA7191FC158B70E4F98643BB0409D971F3B7FCB2B58F03855C7CDB41039FEEA83A72BDFB3FF7DFF0066AC83D2D3FEBEA4
FF00D0C578F8896CBBC53FBD53767DFF0089D775CDFCEDEEE192BFA36BE69DAF5F0E96DBDB690465CEF8326FE08C7DA1
97A20E365B47D831C00DE9DF201CE3DCB677866DADB71291FF2C61E8B02FABBF1B8638DD83CEEAD297FD5A7FD7F37F4AC8B
8E92FF00D7E8FE46BC5AD276BFCFEFF7B57F87CE4FAAB7BD865771E9ADB2D251FD36DBE156693E6C9BA90E580C21D9866F
F009F780F1B467A4926E00FF176E0B3118731C7CD8D876128A738B780F2646FFA6B2678C1E49E30480356FEEDCFF00D7D2
7F37ACAD47A5FF00FBF07F235E3577A7AB7F85FF001F775B37756B452F6B0DAB827BC9C75F5F651DB6D39FEE56FB526F0E
E1CB30D9F28018C00E3F751F57B8931FF002D1B191E99E32319C999F681B7A90C6156C60609DD73276E31F2E7B8183F2F37
EE7EFDCFFD7383F9C758F77D65FF00AF78FF00F408EBC6C4DD4ADE4F5F4EDBD97BBA2E9A7F2A3E830EB45E89F7DD41EBD5F
C7AF7F7BAC9B59E658327293B9CF2E1BEF7FB5FF02EBF8D15763FF571FF00B8BFFA08A2B86DFE1FFC05797F93FBFEFE9E65
DA5FF81BFF002F5FE968FFD9HTTP/1.1 200 OK
Date: Fri, 08 Dec 2017 11:42:07 GMT
Server: Apache/2.4.23 (Win64) PHP/5.6.25
X-Powered-By: PHP/5.6.25
Content-Length: 7
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```

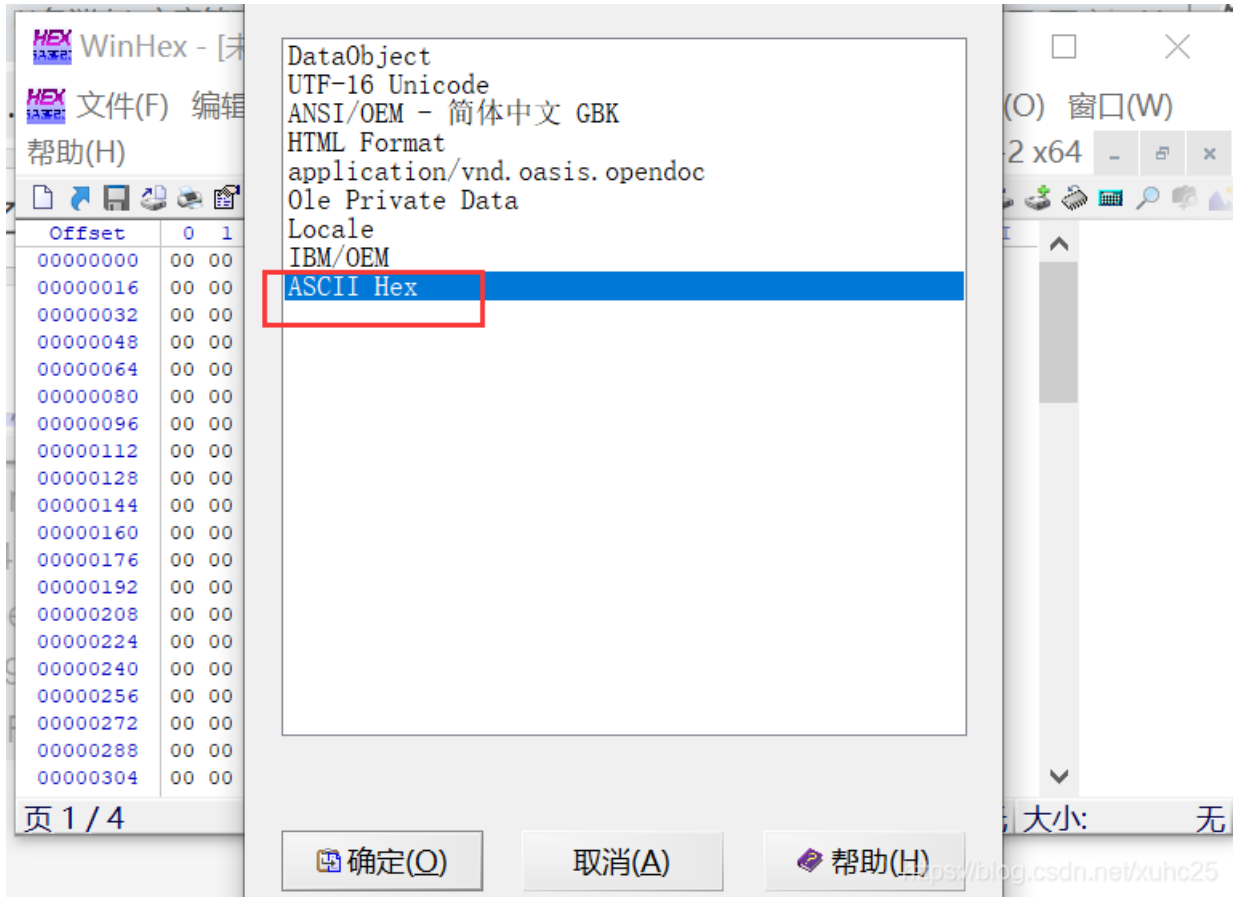
分组 1150, 55 客户端 分组, 145 服务器 分组, 71 turn(s). 点击选择。

winhex生成图片

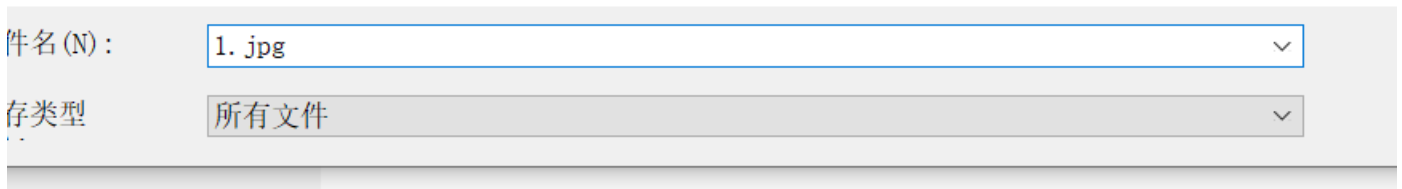
从FFD8FF开头，FFD9结尾复制，打开winhex  
点击文件，新建，大小设置1111



粘贴, 提示编码选择, 选ASCII Hex



另存为, 加后缀.jpg



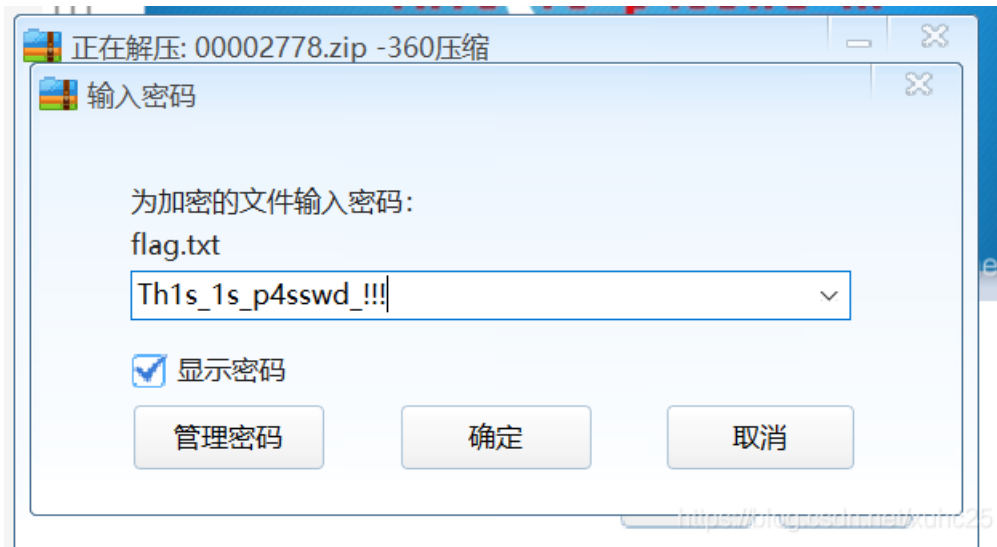
得到一张图, 估计是解压密码了






Th1s\_1s\_p4sswd\_!!!

拿去解压，得到flag.txt文件



打开flag.txt，得到flag

flag{3OpWdJ-JP6FzK-koCMAK-VkfWBq-75Un2z}

 flag.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

flag{3OpWdJ-JP6FzK-koCMAK-VkfWBq-75Un2z}

至此，新手的misc模块题目全部完成了。