




# 攻防世界杂项(misc)--新手练习区(详解十二道题完结, 附件做题过程中使用到的各种工具和网站)

原创

DBINGSEC  于 2021-10-07 15:57:50 发布  1066  收藏 6

分类专栏: [CTF合集](#) 文章标签: [其他](#) [经验分享](#)

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/qg\\_56607768/article/details/120636729](https://blog.csdn.net/qg_56607768/article/details/120636729)

版权



[CTF合集](#) 专栏收录该内容

6 篇文章 1 订阅

订阅专栏

## 攻防世界杂项(misc)--新手练习区 (详解)

### 第一题: [this\\_is\\_flag](#)

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

根据题目描述确定flag为: flag{th1s\_!s\_a\_d4m0\_4!a9}

### 第二题: [pdf](#)

题目描述: 菜猫给了菜狗一张图, 说图下面什么都没有

得到的文件为一个pdf图片：



根据题目描述：图片下面什么都没有，那么图片猜测flag就这图片下面，于是用ctrl+a全选粘到文本文档中，果然flag出现了：  
flag{security\_through\_obscurity}

### 第三题：如来十三掌

题目描述：菜狗为了打败菜猫，学了一套如来十三掌。

#### 如来十三掌

👍 172

最佳Writeup由flag{not\_here} · 渣渣再提供

WP

建议

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

打开文件：

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苦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訥神。舍切真怯勝訥得俱沙罰娑是怯遠得訥數罰輸哆遠薩得槃漫夢盧幡亦醯訥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

根据题目描述：菜狗为了打败菜猫，学了一套如来十三掌。然后依照文件中的内容，那么猜测是与佛论禅密码

[与佛论禅密码加解密网站](#)

## AmanCTF - 与佛论禅密码

在线与佛论禅加密/解密

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苦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訥神。舍切真怯勝訥得俱沙罰娑是怯遠得訥數罰輸哆遠薩得槃漫夢盧幡亦醯訥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

加密

解密

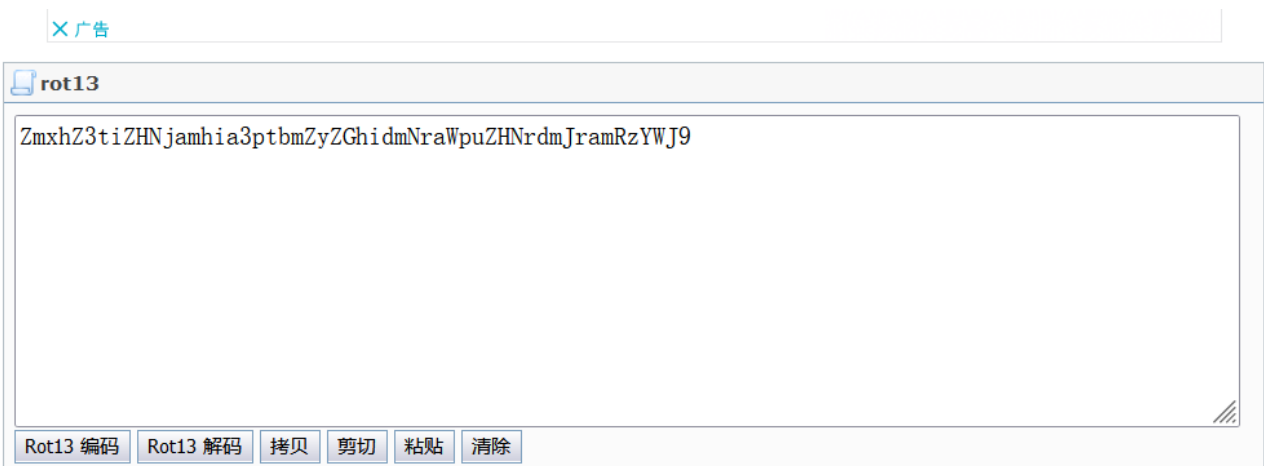
MzkuM3gvMUAwnzuvn3cgozMIMTuvqzAenJchMUAeqzWenzEmLJW9

CSDN @seven-BING

MzkuM3gvMUAwnzuvn3cgozMIMTuvqzAenJchMUAeqzWenzEmLJW9

得到后感觉还不是flag,首先想到base64解密，但是不对，后来想到如来十三掌，那么猜测可能是rot13加密

[rot13加解密网站](#)



CSDN @seven-BING

ZmxhZ3tiZHNjamhia3ptbmZyZGhidmNraWpuZHNrdmJramRzYWJ9

然后我进行base64解密得到flag: flag{bdscjhbkmznmfrdhubvckijndskvbkjdsab}

[base64加解密网站](#)

## AmanCTF - BASE64编码解码

在线BASE64编码解码

```
ZmxhZ3tiZHNjamhia3ptbmZyZGhidmNraWpuZHNrdmJramRzYWJ9
```

加密

解密

```
flag{bdscjhbkmfrdhbvckijndskvbkjdsab}
```

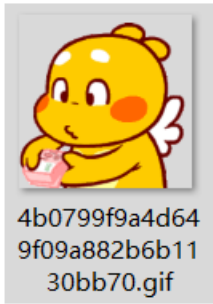
CSDN @seven-BING

## 第四题：give\_you\_flag

题目描述：菜狗找到了文件中的彩蛋很开心，给菜猫发了个表情包

The screenshot shows a CTF challenge interface with a dark theme. At the top, there is a navigation bar with a '返回' (Return) button, a star icon, and a timer showing '本题用时: 3分13秒'. On the right side of the navigation bar is a gold medal icon. Below the navigation bar, the challenge title 'give\_you\_flag' is displayed, followed by a thumbs-up icon and the number '107', and a badge indicating '最佳Writeup由testtestzrs提供'. To the right of the title are two buttons: 'WP' and '建议'. Below the title, the '难度系数' (Difficulty Coefficient) is shown as '4.0' with five stars. The '题目来源' (Source) is '暂无'. The '题目描述' (Description) is '菜狗找到了文件中的彩蛋很开心，给菜猫发了个表情包'. The '题目场景' (Scenario) is '暂无'. The '题目附件' (Attachments) section shows a button for '附件1'. At the bottom right of the challenge area is a right-pointing arrow. The footer of the page reads 'CSDN @seven-BING'.

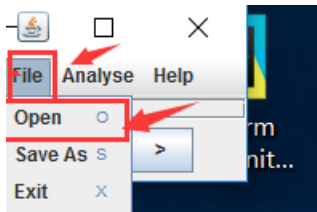
打开文件是一个gif动态图：



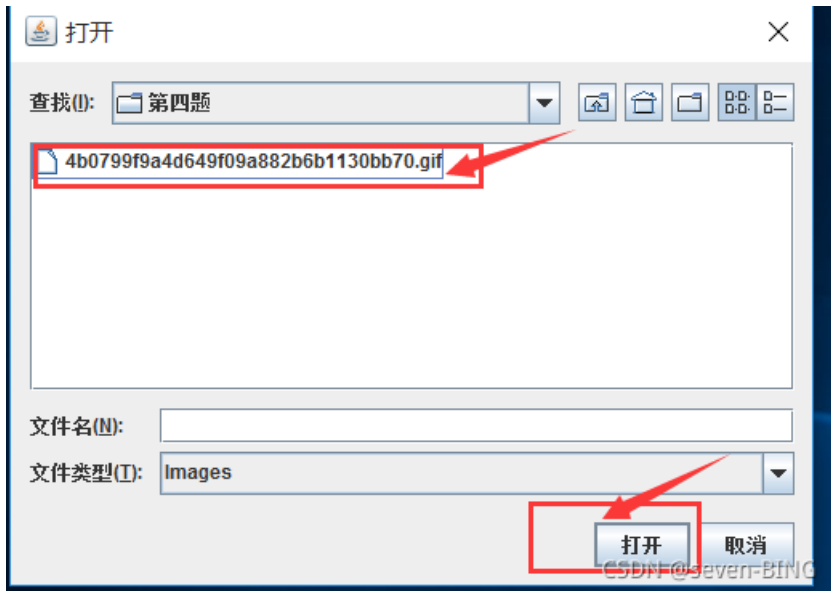
于是我用stegsolve工具进行动态图片分离

工具链接:[stegsolve工具](#)

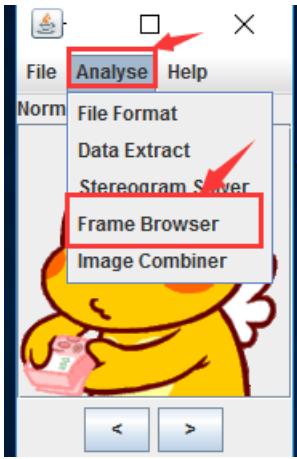
1.首先我点击file



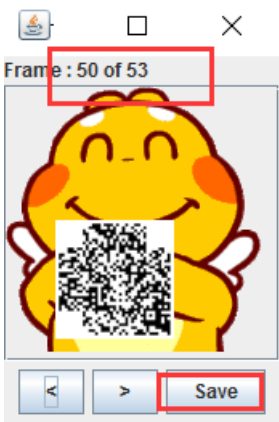
2.点击下好的附件，打开



3.点击分析



4.翻到第五十张的时候出现二维码，点击保存



看到是一个缺少定位符的二维码,在网上找一个定位点图,用3d画图修改图片即可。得到以下图片



CSDN @seVen-BING

使用CQR二维码扫描工具得到flag值： flag{e7d478cf6b915f50ab1277f78502a2c5}



工具链接:[CQR二维码扫描工具](#)

## 第五题: stegano

题目描述: 菜狗收到了图后很开心, 玩起了pdf 提交格式为flag{xxx}, 解密字符需小写



stegano

👍 478 最佳Writeup由LK-TEAM • 来自南方的羊提供

WP

建议

难度系数：★★★★ 4.0

题目来源：CONFidence-DS-CTF-Teaser

题目描述：菜狗收到了图后很开心，玩起了pdf 提交格式为flag{xxx}，解密字符需小写

题目场景：暂无

题目附件：附件1

CSDN @seven-BING

打开是一个：pdf文件，并提示flag不在此里



然后我ctrl+a全选文章，复制到文本文档中，发现一串AB。

Vestibulum suscipit lorem sed sem faucibus rutrum. Nunc diam orci, convallis vitae auctor vehicula, Maecenas nec urna at dolor mattis dictum sit amet at orci. Mauris condimentum adipiscing erat nec f scelerisque varius ligula, iaculis adipiscing dui. Duis eget ullamcorper arcu. In facilisis et tort Your flag is not here lol estie bibendum, leo nisi porttitor massa, id accumsan sapien libero id tel sollicitudin a felis quis, blandit porta ipsum. Donec sed nibh egestas, tristique mauris eu, rutrum Duis gravida semper dui laoreet vulputate. Aenean quis tempor orci. Cras placerat lectus nulla, eu interdum in. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras faucibus odio ut metus vu magna volutpat. Integer nec enim vel arcu porttitor egestas. Vestibulum suscipit lorem sed sem fauc diam orci, convallis vitae auctor vehicula, interdum ut mi. Maecenas nec urna at dolor mattis dictu Mauris condimentum adipiscing erat nec feugiat. Curabitur scelerisque varius ligula, iaculis adipis ullamcorper arcu. In facilisis et tortor commodo aliquam. Nulla feugiat, sem eu molestie bibendum, massa, id accumsan sapien libero id tellus. In enim lacus, sollicitudin a felis quis, blandit porta egestas, tristique mauris eu, rutrum justo. Nulla facilisi. Duis gravida semper dui laoreet vulputa orci. Cras placerat lectus nulla, eu bibendum metus interdum in. Lorem ipsum dolor sit amet, consect elit. Cras faucibus odio ut metus vulputate, id laoreet magna volutpat. Integer nec enim vel arcu p Vestibulum suscipit lorem sed sem faucibus rutrum. Nunc diam orci, convallis vitae auctor vehicula, Maecenas nec urna at dolor mattis dictum sit amet at orci. Mauris condimentum adipiscing erat nec f scelerisque varius ligula, iaculis adipiscing dui. Duis eget ullamcorper arcu. In facilisis et tort feugiat, sem eu molestie bibendum, leo nisi porttitor massa, id accumsan sapien libero id tellus. I a felis quis, blandit porta ipsum. Donec sed nibh egestas, tristique mauris eu, rutrum justo. Nulla semper dui laoreet vulputate. Aenean quis tempor orci. Cras placerat lectus nulla, eu bibendum metu placerat lectus nulla, eu bibendum metus interdum in. Lorem ipsum dolor sit amet, consectetur adipis faucibus odio ut metus vulputate, id laoreet magna volutpat. Integer nec enim vel arcu porttitor eg suscipit lorem sed sem faucibus rutrum. Nunc diam orci, convallis vitae auctor vehicula, interdum u urna at dolor mattis dictum sit amet at orci. Mauris condimentum adipiscing erat nec feugiat. Curab varius ligula, iaculis adipiscing dui. Duis eget ullamcorper arcu. In facilisis et tortor commodo a sem eu molestie bibendum, leo nisi porttitor massa, id accumsan sapien libero id tellus. In enim la quis, blandit porta ipsum. Donec sed nibh egestas, tristique mauris eu, rutrum justo. Nulla facilis dui laoreet vulputate. Aenean quis tempor orci. Cras placerat lectus nulla, eu bibendum metus inter

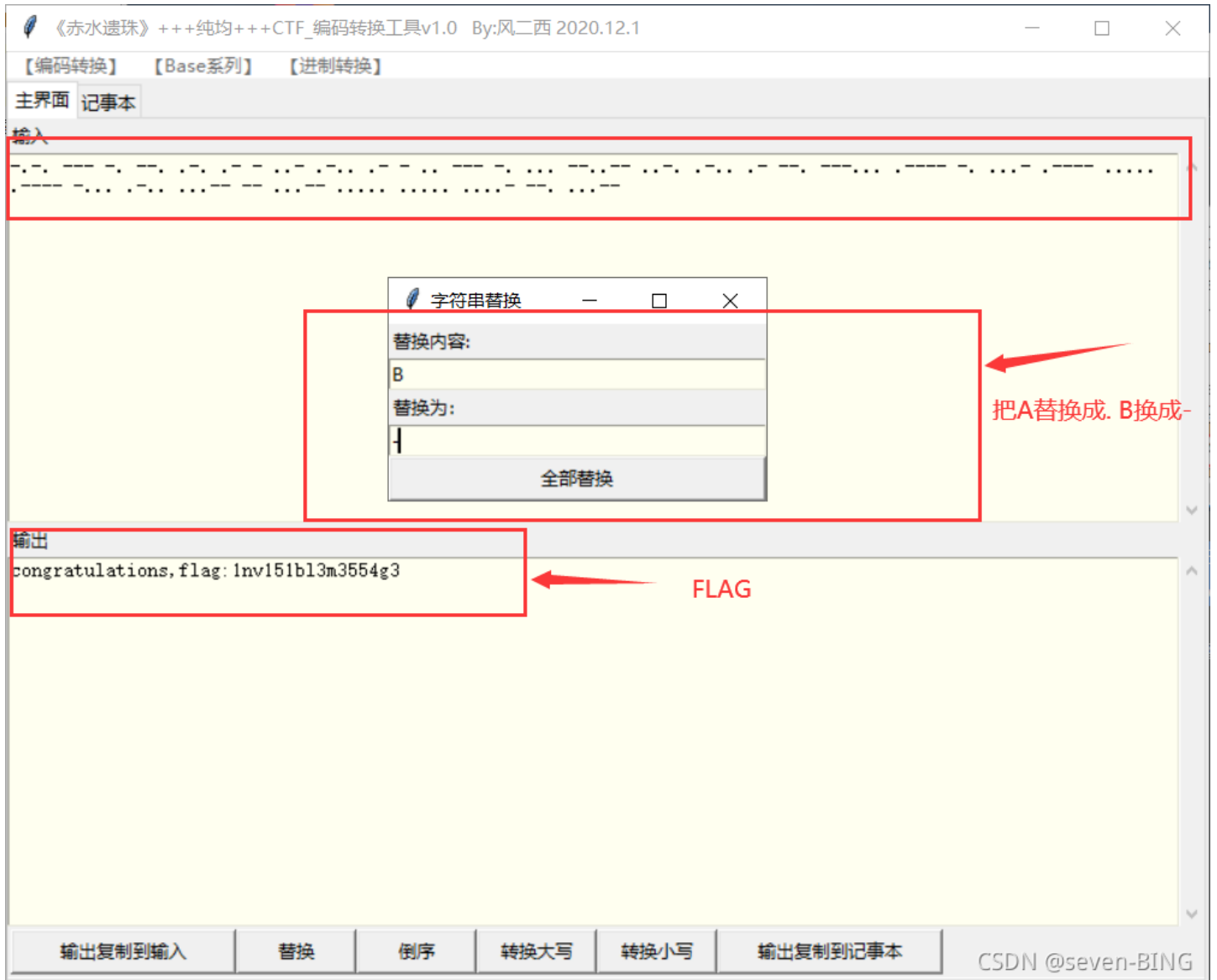
BABA BBB BA BBA ABA AB B AAB ABAA AB B AA BBB BA AAA BBAABB AABA ABAA AB BBA BBBAAA AB BBB BA AAAB A

Close - but still not here !

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  
 AAAB AB BBB AAAA AB BBB BAAA ABAA AAABB BB AAABB AAAA AAAA AAAAB BBA AAABB

根据密文猜测是摩斯密码，这里使用到B站一位UP主（风二西）的工具（CTF编码转化工具）：

根据密文想到A替换成摩斯的. B替换成摩斯-



得到的flag: flag{1nv151b13m3554g3}

## 第六题：坚持60s

题目描述：菜狗发现最近菜猫不爱理他，反而迷上了菜鸡

← 返回  本题用时: 2分38秒

坚持60s  20 最佳Writeup由不要让我起名提供 WP 建议

难度系数:  4.0

题目来源: 08067CTF

题目描述: 菜狗发现最近菜猫不爱理他, 反而迷上了菜鸡

题目场景: 暂无

题目附件: 附件1

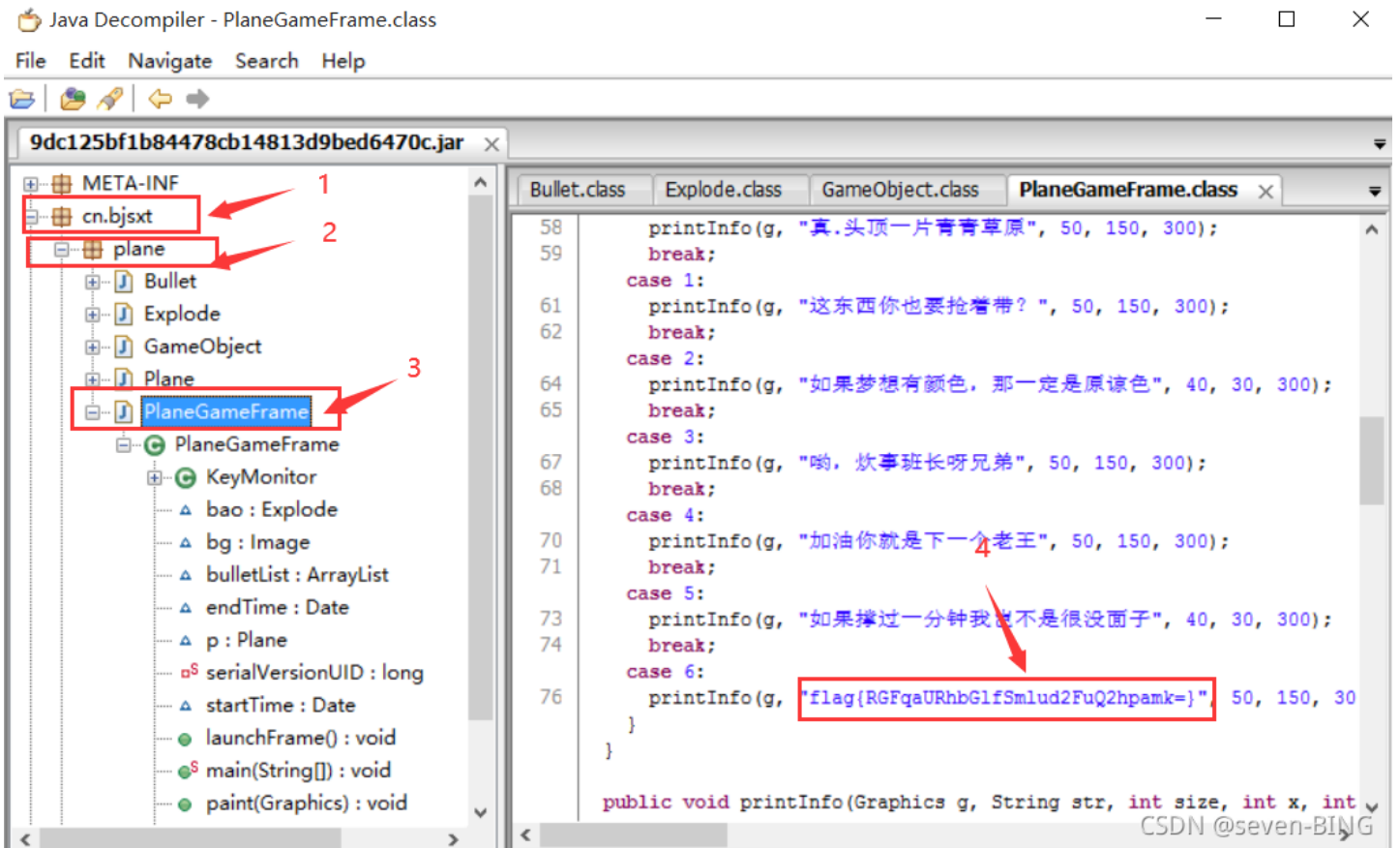
CSDN @seven-BING

打开文件是一个后缀为: jar



方法一: 坚持60s即可得到flag

方法二: 使用逆向简单工具jd-gui [工具链接](#)



flag{RGFqaURhbGlfSmlud2FuQ2hpamk=}

通过尝试上传flag后发现不对, 那么一看加密方式为base64进行解密, 在线加解密网

站: <https://www.qqxiuzi.cn/bianma/base64.htm>

设计 | 头像 | 同 | 有刷 | 反 | 观 | 涂 | 无 | 资 | 下 | 载 | 仅 | 由 | 参 | 考 | 资 | 料 | !

基恩士 打开 >

RGFqaURhbGlfSmlud2FuQ2hpamk=

清空
加密
解密
 解密为UTF-8字节流

DajiDali\_JinwanChiji

复制

最后flag值为: flag{DajiDali\_JinwanChiji}

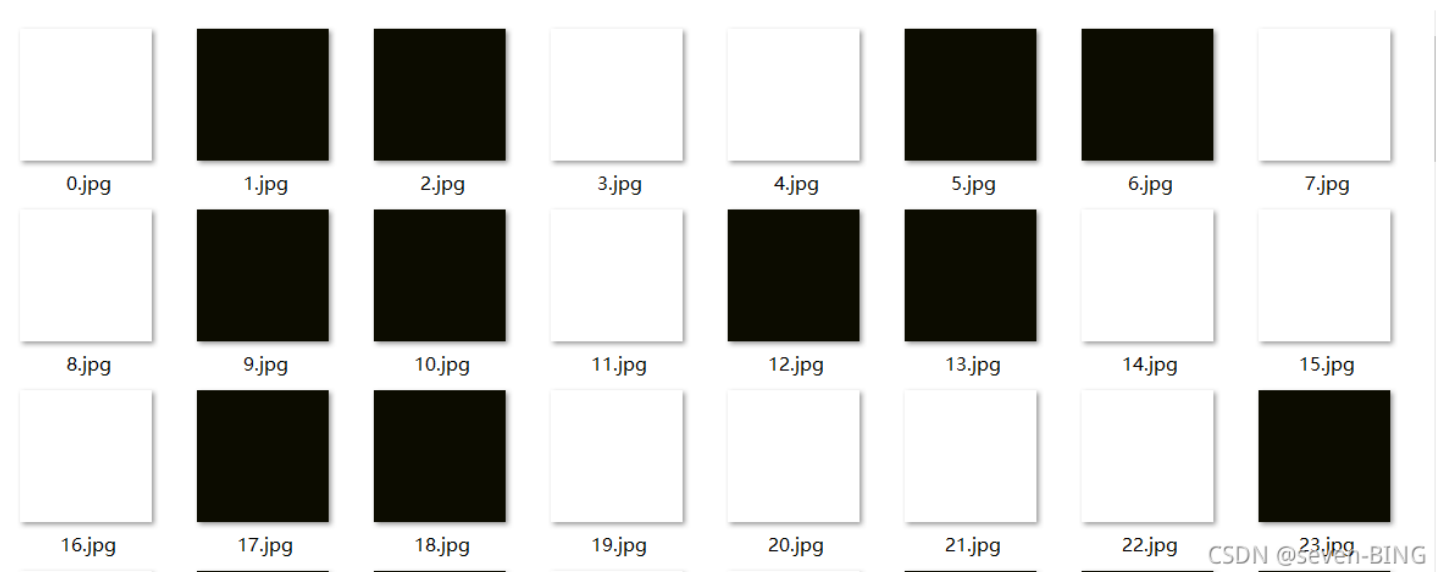
## 第七题：gif

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



The screenshot shows a CSDN problem page for a challenge titled "gif". At the top, there is a navigation bar with a "返回" (Return) button and a timer showing "本题用时: 1分4秒". The problem title "gif" is followed by a thumbs-up icon and the number "59", and a badge that says "最佳Writeup由不要让我起名提供". There are two buttons: "WP" and "建议". Below the title, the "难度系数" (Difficulty Coefficient) is shown as "★★★★ 4.0". The "题目来源" (Source) is "暂无" (None). The "题目描述" (Description) is "菜狗截获了一张菜鸡发给菜猫的动态图，却发现另有玄机". The "题目场景" (Scenario) is "暂无" (None). The "题目附件" (Attachments) section shows a button for "附件1". In the bottom right corner, there is a watermark "CSDN @seven-BING".

解压后是一组图片：



打开文件出现多个黑白，让人联想到二进制，白色图片代表0，黑色图片代表1。01100110前八位二进制换算后为f证明思路正确。

方法一：手动将图片转化为01二进制，转化为

0110011001101100011000010110011101111011010001100111010101001110010111110110011101101001010001100111101

方法二：编写python脚本：

```

from PIL import Image
result=""
for num,i in enumerate(range(104)):
    img=Image.open(f"G:/Desktop/dbbc971bf4da461fb8939ed8fc9c4c9d/gif/{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="")

```

```

1  from PIL import Image
2  result=""
3  for num,i in enumerate(range(104)):
4      img=Image.open(f"G:/Desktop/dbbc971bf4da461fb8939ed8fc9c4c9d/gif/{i}.jpg")
5      im=img.convert("RGB")
6      r,g,b=im.getpixel((1,1))
7      if r!=255:
8          result+="1"
9      else:
10         result+="0"
11  for i in range(0,len(result),8):
12      byte=result[i:i+8]
13      print(chr(int(byte,2)),end="")

```

for num,i in enumerate(10...

Run: test x

D:\python3.9.4\python3.9.exe D:/python3.8.0/python代码练习/练习/test.py  
flag{FuN\_giF}  
进程已结束,退出代码0

CSDN @seven-BING

最后flag是: flag{FuN\_giF}

## 第八题: 掀桌子

题目描述: 菜狗截获了一份报文如下

c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaeabfaebe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2, 生气地掀翻了桌子(ノ°°)ノ ㄣ ㄣ ㄣ

### 掀桌子

👍 176 最佳Writeup由flag{not\_here} · 渣渣再提供

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c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaeabfaebe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2, 生气地掀翻了桌子(ノ°°)ノ ㄣ ㄣ ㄣ

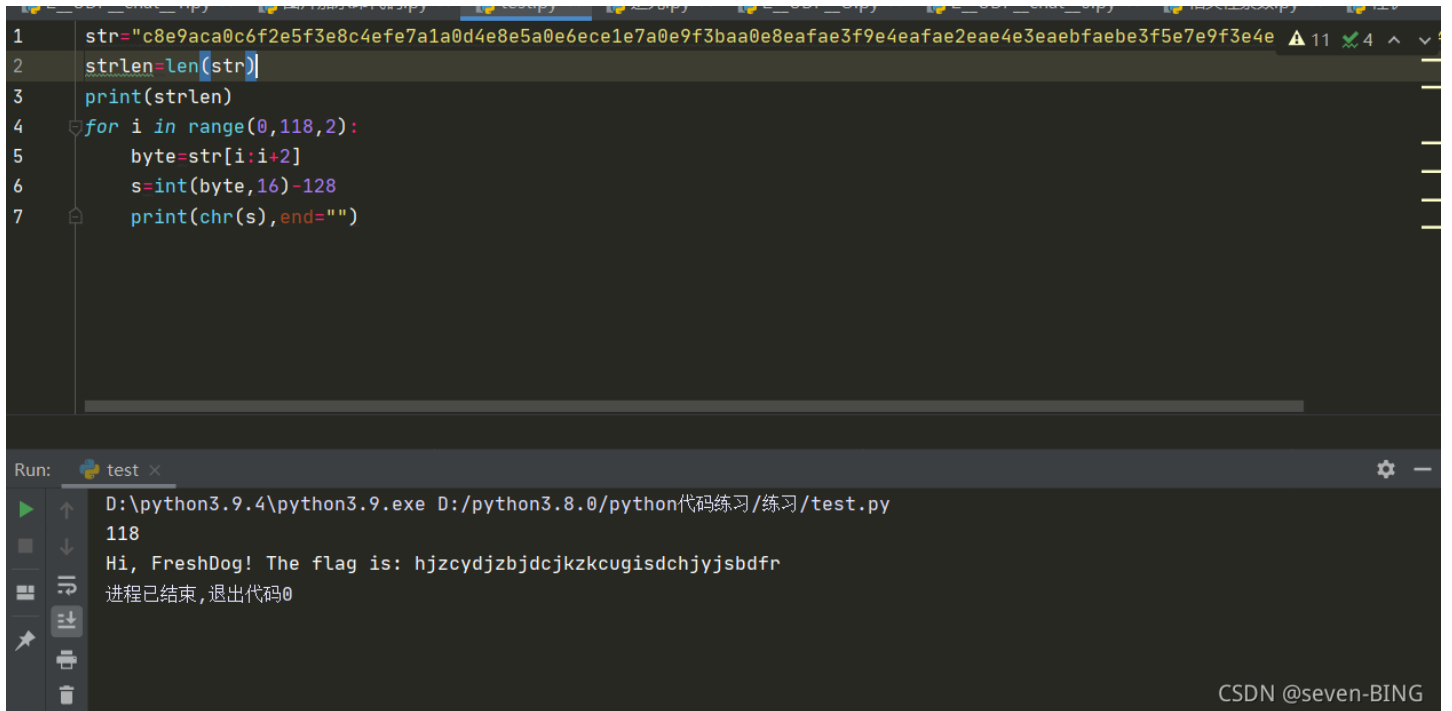
题目场景: 暂无

题目附件: 暂无

CSDN @seven-BING

看题给字符串比较像十六进制, 两个一组, 转化为十进制, 减去128, 再转字符串得到flag, 有一点投机取巧。

```
str="c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfabe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2"
strlen=len(str)
print(strlen)
for i in range(0,118,2):
    byte=str[i:i+2]
    s=int(byte,16)-128
    print(chr(s),end="")
```



```
1 str="c8e9aca0c6f2e5f3e8c4efe7a1a0d4e8e5a0e6ece1e7a0e9f3baa0e8eafae3f9e4eafae2eae4e3eaebfabe3f5e7e9f3e4e3e8eaf9eaf3e2e4e6f2"
2 strlen=len(str)
3 print(strlen)
4 for i in range(0,118,2):
5     byte=str[i:i+2]
6     s=int(byte,16)-128
7     print(chr(s),end="")
```

Run: test x

D:\python3.9.4\python3.9.exe D:/python3.8.0/python代码练习/练习/test.py

118

Hi, FreshDog! The flag is: hjzcydjzbdcjzkzcugisdchjysbdf

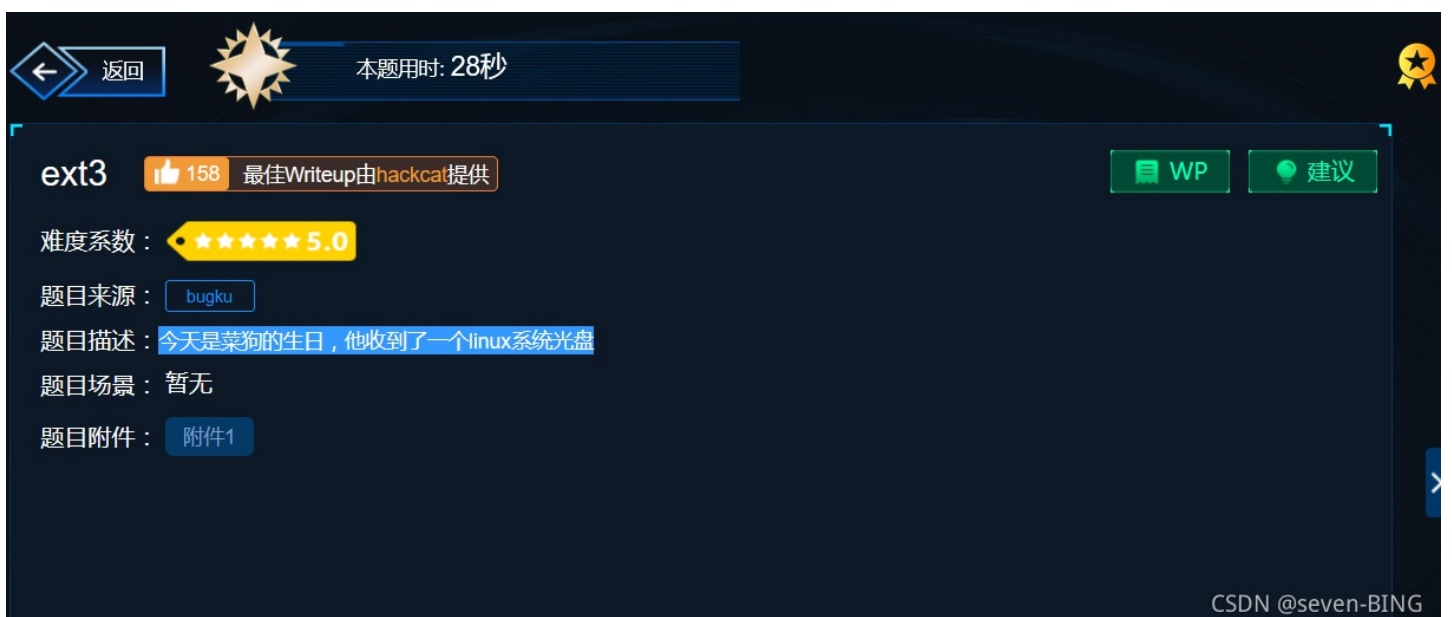
进程已结束,退出代码0

CSDN @seven-BING

最后flag为: flag{hjzcydjzbdcjzkzcugisdchjysbdf}

## 第九题: ext3

题目描述: 今天是菜狗的生日, 他收到了一个linux系统光盘



返回 本题用时: 28秒

ext3 158 最佳Writeup由hackcat提供 WP 建议

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

题目场景: 暂无

题目附件: 附件1

CSDN @seven-BING

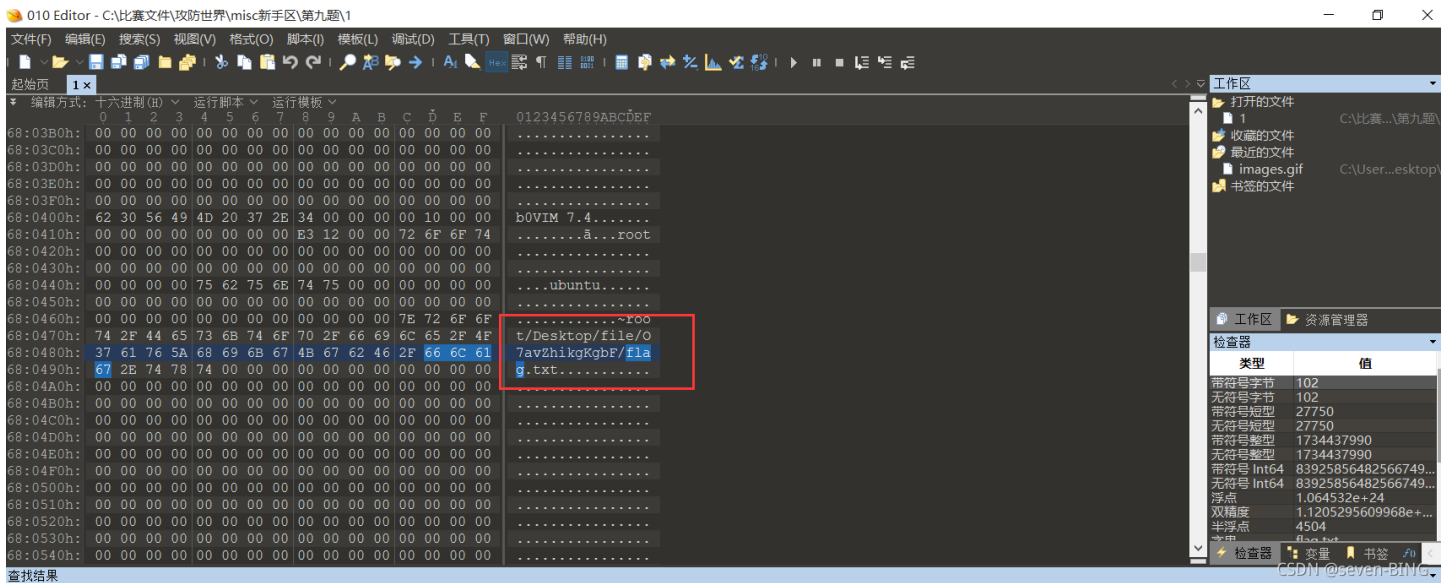


得到的文件是：

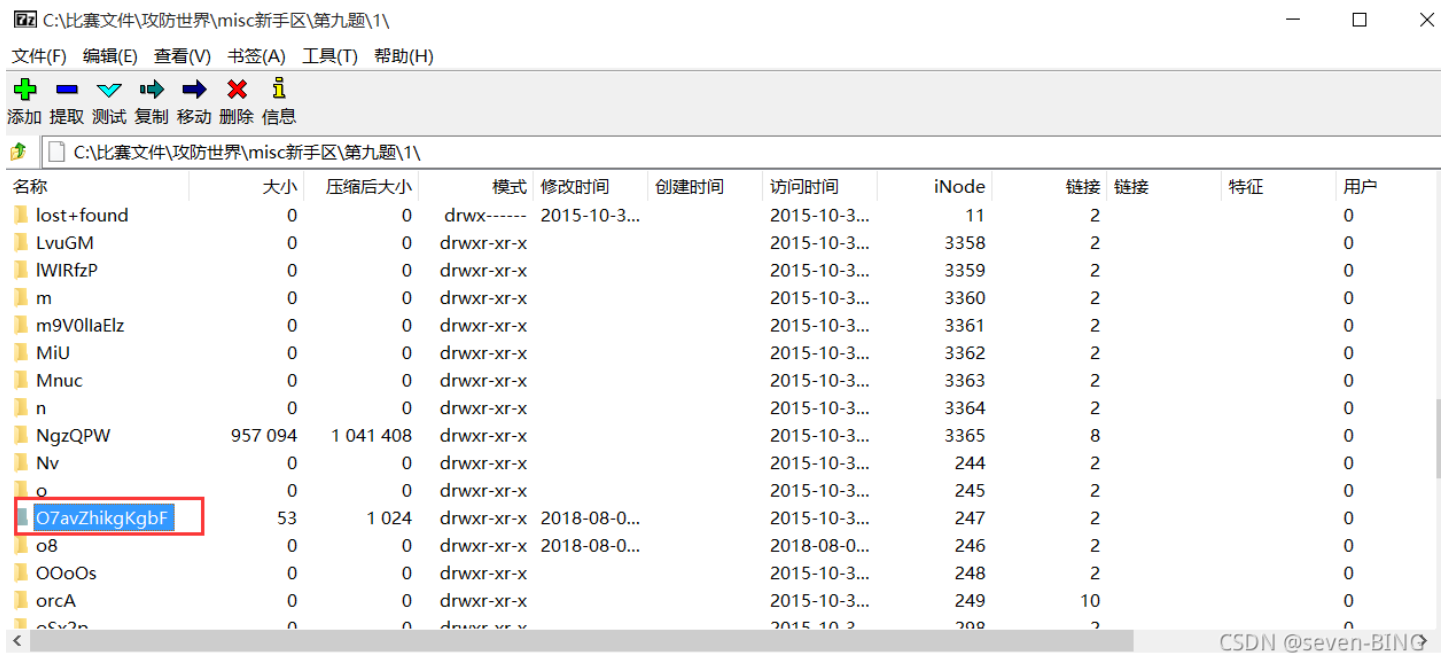


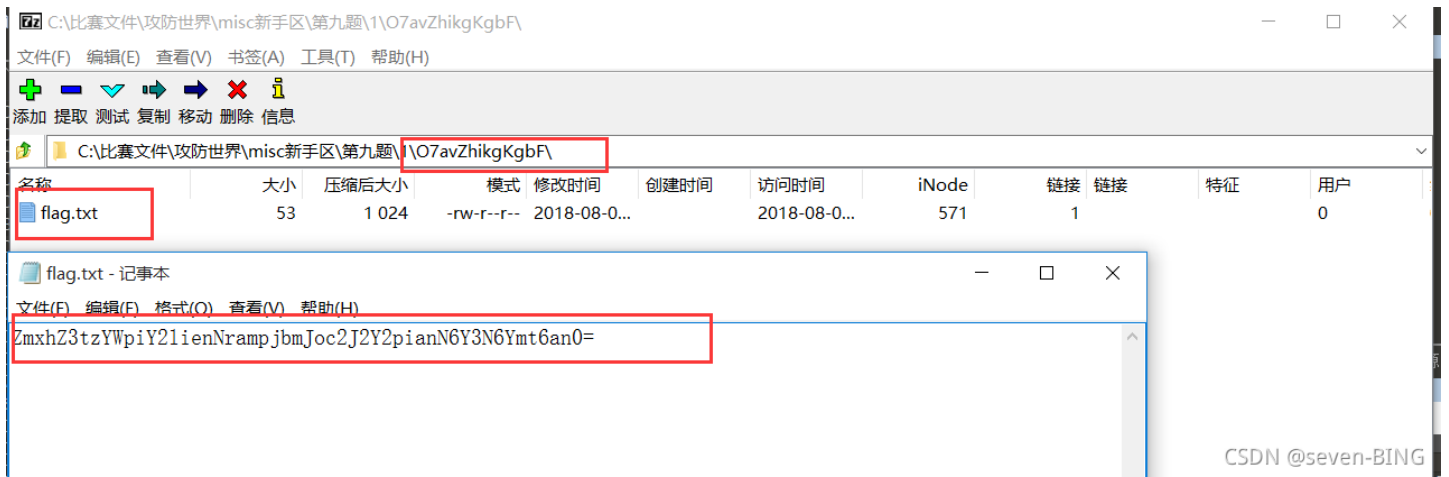
方法一：首先放到010editor查看，搜索字符串flag,发现了如图的

010editor工具链接：



然后我用7z工具打开，发现了一些文件夹，其中就有我们010editor看到的文件夹，点击查看，flag





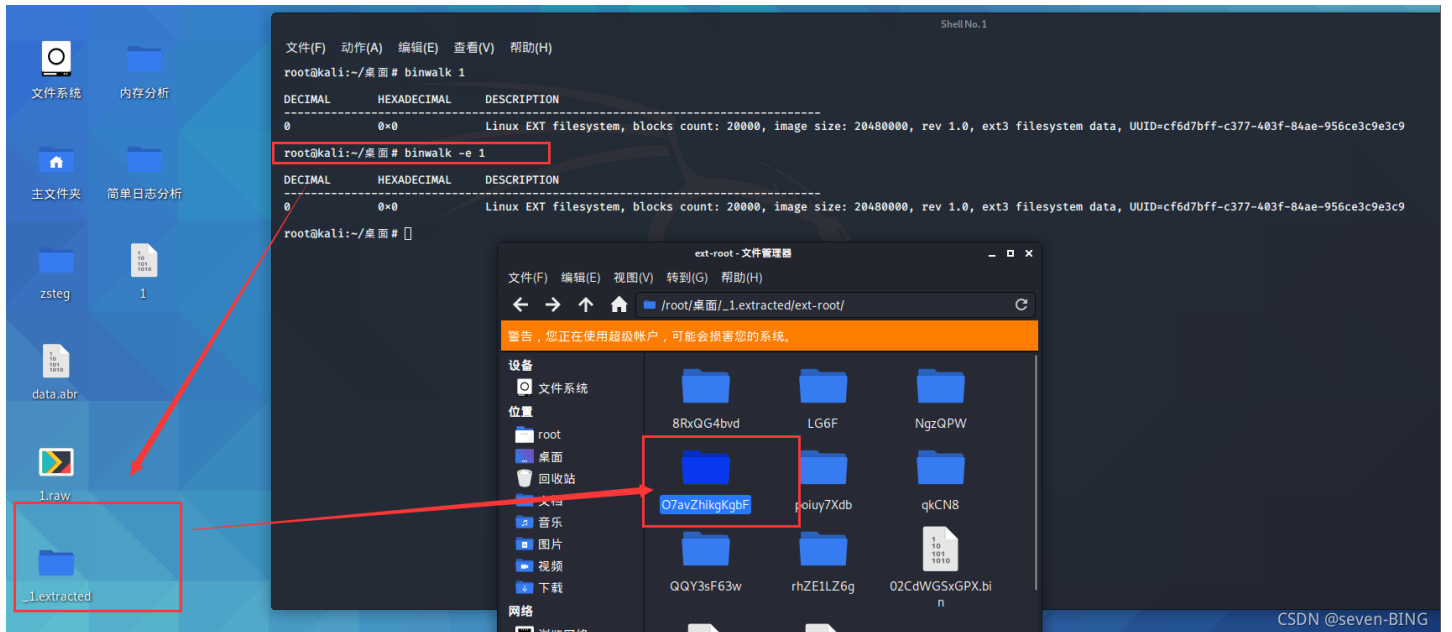
ZmxhZ3tzYWpiY2lienNrampjbmJoc2J2Y2pianN6Y3N6Ymt6an0=

看到这个就是用base64加密的: [base64在线加解密网站](#)



最后得到的flag: flag{sajbcibzskjcnbhsbvcjbjszcszbkzj}

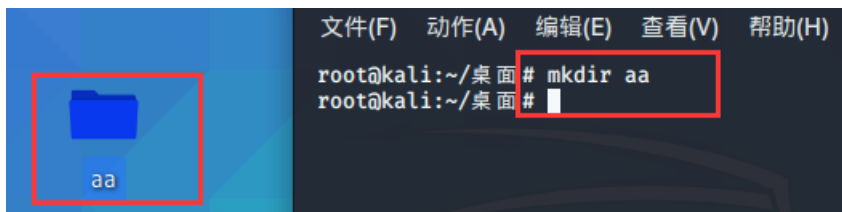
方法二: 用kali中binwalk -e 分离出文件, 即可得到flag



方法三：挂载到kali下

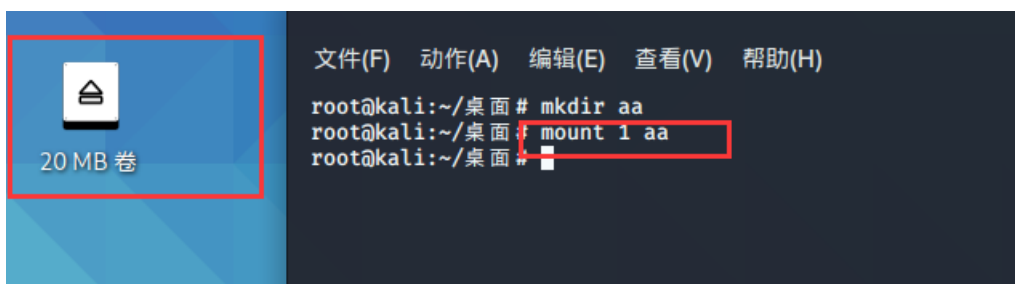
1.首先创建一个文件夹

命令：mkdir aa



2.挂载到刚刚创建的文件夹下面

命令：mount 1 aa



3.进入到aa 文件夹下 cd aa ,并查看有哪些文件 ls

```
文件(F) 动作(A) 编辑(E) 查看(V) 帮助(H)
root@kali:~/桌面 # mkdir aa
root@kali:~/桌面 # mount 1 aa
root@kali:~/桌面 # cd aa
root@kali:~/桌面/aa# ls
02CdWGSxGPX.bin 0wDq5 3J 7H7geLLS5 8RxQG4bvd h i jj L00J8 m9V0lIaElz Nv orcA Q
0GY1l 0Xs 44aAm 8A2MFawD4 FinD H imgLDpt4BY KxEQM lost+found MiU o oSx2p qkCN8
0h3a5 1 4A 8DQFirm0D fm H2Zj8FNbu ix1EMRHRpIc2 LG6F LvuGM Mnuc 07avZhikgKgbF OT QmUY1d
0l 2X 6JR3 8HhWfV9nK1 g hdi7 j6uLMX Lh LWIRfzP n o8 poiuy7Xdb QQY3sF6
0qsd 3 6wUaZE1vbsW 8nwg gtj hYuPvID jE LLC6Z0zrgy.bin m NgzQPW 00o0s px6u r
root@kali:~/桌面/aa#
```

进入文件夹  
查看内容

4. 查看文件中是否存在flag文件，用 find -name flag\*命令

```
文件(F) 动作(A) 编辑(E) 查看(V) 帮助(H)
root@kali:~/桌面 # mkdir aa
root@kali:~/桌面 # mount 1 aa
root@kali:~/桌面 # cd aa
root@kali:~/桌面/aa# ls
02CdWGSxGPX.bin 0wDq5 3J 7H7geLLS5 8RxQG4bvd h i jj L00J8 m9V0lIaElz Nv orcA Q
0GY1l 0Xs 44aAm 8A2MFawD4 FinD H imgLDpt4BY KxEQM lost+found MiU o oSx2p qkCN8
0h3a5 1 4A 8DQFirm0D fm H2Zj8FNbu ix1EMRHRpIc2 LG6F LvuGM Mnuc 07avZhikgKgbF OT QmUY1d
0l 2X 6JR3 8HhWfV9nK1 g hdi7 j6uLMX Lh LWIRfzP n o8 poiuy7Xdb QQY3sF6
0qsd 3 6wUaZE1vbsW 8nwg gtj hYuPvID jE LLC6Z0zrgy.bin m NgzQPW 00o0s px6u r
root@kali:~/桌面/aa# find -name flag*
./07avZhikgKgbF/flag.txt
root@kali:~/桌面/aa#
```

5. 使用cat 查看里面的内容

```
Shell No.1
文件(F) 动作(A) 编辑(E) 查看(V) 帮助(H)
root@kali:~/桌面 # mkdir aa
root@kali:~/桌面 # mount 1 aa
root@kali:~/桌面 # cd aa
root@kali:~/桌面/aa# ls
02CdWGSxGPX.bin 0wDq5 3J 7H7geLLS5 8RxQG4bvd h i jj L00J8 m9V0lIaElz Nv orcA Q
0GY1l 0Xs 44aAm 8A2MFawD4 FinD H imgLDpt4BY KxEQM lost+found MiU o oSx2p qkCN8
0h3a5 1 4A 8DQFirm0D fm H2Zj8FNbu ix1EMRHRpIc2 LG6F LvuGM Mnuc 07avZhikgKgbF OT QmUY1d
0l 2X 6JR3 8HhWfV9nK1 g hdi7 j6uLMX Lh LWIRfzP n o8 poiuy7Xdb QQY3sF6
0qsd 3 6wUaZE1vbsW 8nwg gtj hYuPvID jE LLC6Z0zrgy.bin m NgzQPW 00o0s px6u r
root@kali:~/桌面/aa# find -name flag*
./07avZhikgKgbF/flag.txt
root@kali:~/桌面/aa# cat ./07avZhikgKgbF/flag.txt
ZmxhZ3tzYWpiY2lienNrampjbmJoc2J2Y2pianN6Y3N6Ymt6an0=
root@kali:~/桌面/aa#
```

ZmxhZ3tzYWpiY2lienNrampjbmJoc2J2Y2pianN6Y3N6Ymt6an0=

使用base64解密即可得到flag flag{sjabcibzskjcnbhsbvcjbjsczszbkzj}

## 第十题：SimpleRAR

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)

难度系数：★★★★★ 5.0

题目来源：0806CTF

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:双层层)

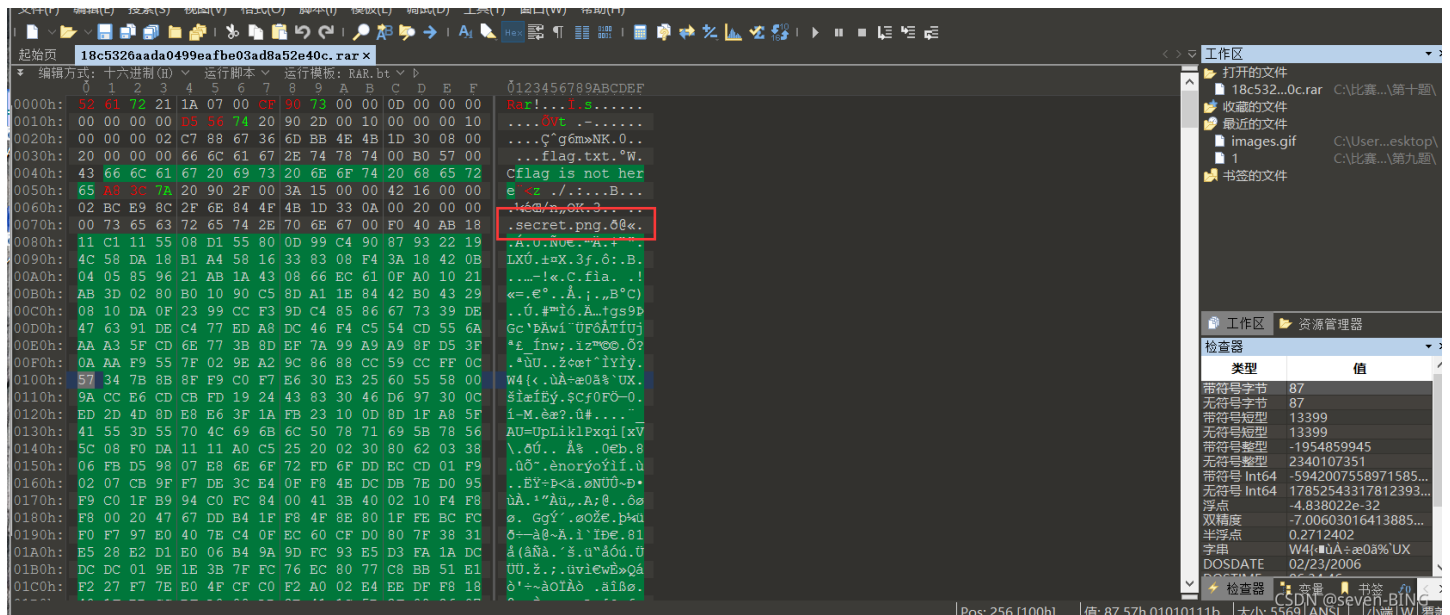
题目场景：暂无

题目附件：附件1

文件是一个压缩包

1.首先我解压，提示压缩包损坏，我在用7z打开，里面有一个flag.txt文件，但是打开说flag 不在这

2.再用010editor打开，看见出一些端倪 010editor工具链接



Header CRC mismatch in Block #4

打开并提示：

压缩包被修改的。

rar 每个块的开头基本知识：(看到人家的writeup)

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】

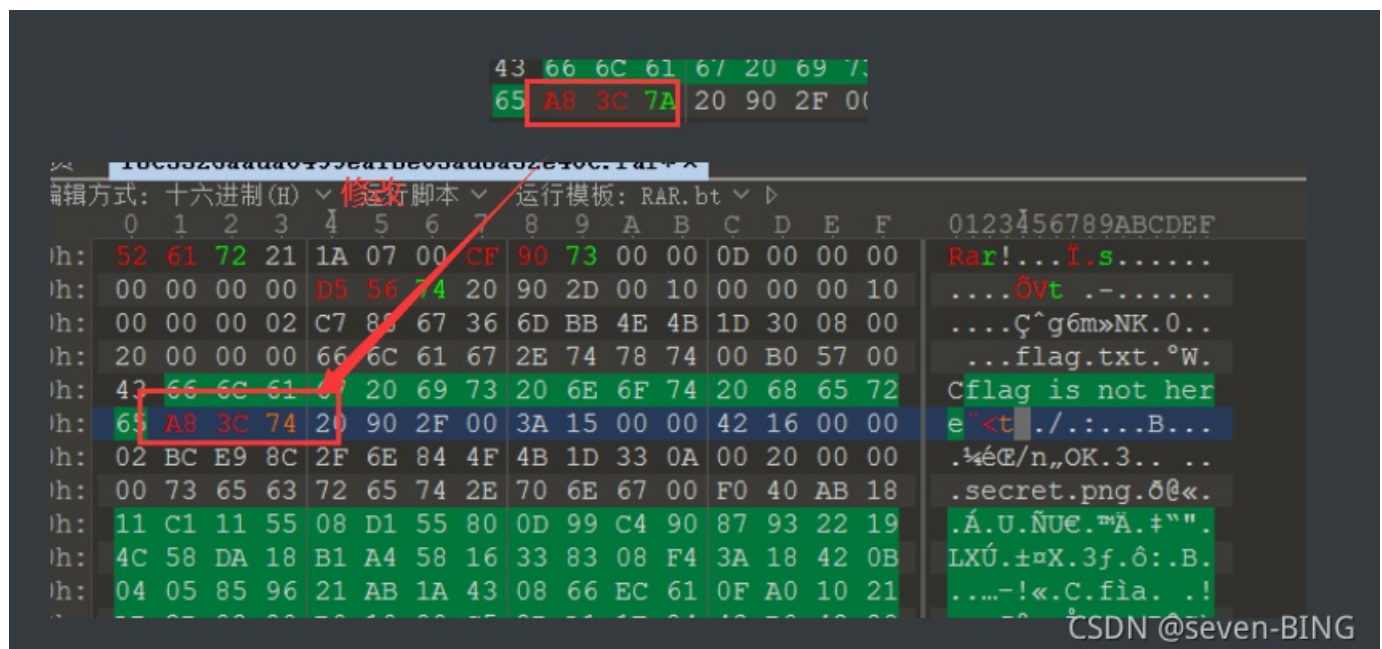
HEAD\_CRC 2 bytes CRC of total block or block part  
整个块或者块某个部分的CRC（根据块类型而有不同）  
HEAD\_TYPE 1 byte Block type  
块类型【译者注：也可以理解为块头部类型，因为不同的块对应不同的块头部。后文也经常混淆这两种概念。】

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】

CSDN @seven-BING

010editor 查看压缩包，rar 文件块的开头是 A8 3C 74 我们需要文件块而不是子块，于是更改 A8 3C 7A 为 A8 3C 74



解压出来：



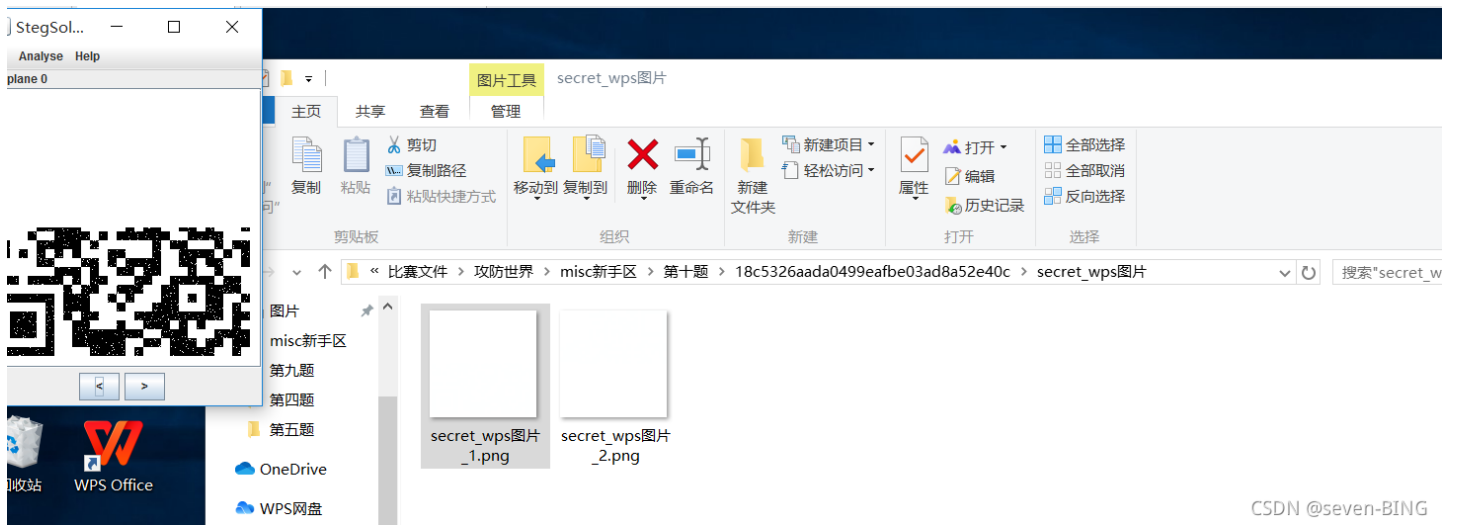
flag.txt



secret.png

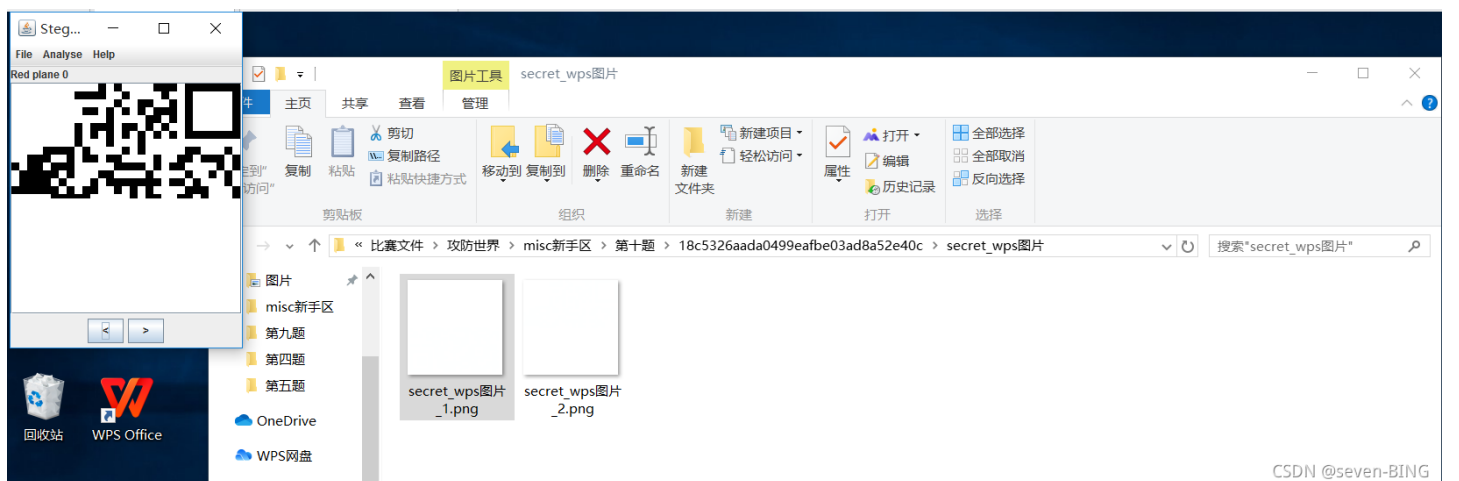
打开图片有两张白色的图片，首先我看了一下文件的属性，发现没有任何东西，我先分别保存两张图片，于是我分别放到 stegsolve 中，查看。[stegsolve 工具链接](#)

通过第一张图片得到半张二维码



CSDN @seven-BING

同样进行第二张图片分析



CSDN @seven-BING

于是3D将两张图片组合拼接，并把定位点拼接上，得到图片



使用CQR二维码扫描工具得到flag值：flag{yanji4n\_bu\_we1shi} [CQR二维码工具链接](#)





## 第十一题: base64stego

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

← 返回 本题用时: 1分1秒

---

**base64stego** 👍 194 最佳Writeup由CTFshow • zEr0\_0提供 WP 建议

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

CSDN @seven-BING

得到的文件是一个压缩包:我尝试解压,发现需要密码,于是我用7z打开,发现里面有一个文件,且里面的内容能够打开,说明该压缩包是一个伪加密。7z工具链接

C:\比赛文件\攻防世界\misc新手区\第十一题\ae2eb7ceaf5ab49f7acb33de2e7eed74a.zip\

名称	大小	压缩后大小	修改时间	创建时间	访问时间	属性	加密	注释	CRC	算法	特征
stego.txt	7 093	3 561	2016-04-2...				-		4B7D32FE	Deflate	Local

```

U3R1Z2Fub2dyYXB0eSBpcyB0aGUgYXJ0IGFuZCBzY211bmN1IG9mIHdyaXRpbmcgaG1kZGVuIG1lc3NhZ2VzIG1uIHN1Y2ggYSB
G1uZXMgb2YgYSBwcm12YXR1IGxldHRlc i4NCg0KVGH1IGFkdmFudGFnZSBvZiBzdGVnYW5vZ3JhcGh5LCBvdmVvIGNyeXB0b2dy
JhcGhpYyB0cmFuc2lpc3Npb+==biBiZWVhdXN1IG9mIHRoZW1yIGxhcml1IHNpemUuIEFzIB==YSBzaW1wbGUgZlhbXhBzZSwgY
=ZiBIaXN0aWFlc2VzIG9mIHRoZW1yIGxhcml1IHNpemUuIEFzIB==YSBzaW1wbGUgZlhbXhBzZSwgYS
LiBUaGUgbWVzc2FnZSBhbGx1Z2VkbHkgY2Fycml1ZCBhIHdhdhcm5pbmcgdG8gR3JlZWN1IGFib5==dXQgUGVyc2lhb1BpbmZhc2l
cGhpY2FsbHkgc0==cm9kdWN1ZCBtaWVyb2RvdHMgdG8gc2VuZCBpbmZvcmlhdGlvbiBiYWVnIGF1ZD==IGZvcnRoLiBNaWVyb2R
aW50ZXh0IiB3YXMaXRzZWxmIGVvY2+=ZGVkIGFuZCBnYXZ1IGluZm9ybWV0aW9uIGFib3V0IHNoaXAgaW92ZW11bnRzLF==IGV
  
```

CSDN @seven-BING

解决伪加密两种方法:

第一种直接用7z工具打开

第二种:用010editor打开,修改其密码位

zip的知识补充:

### 压缩源文件数据区

```
50 4B 03 04 //数据区，压缩包的文件头
14 00 // 解压文件所需的pkware版本
00 00 // 全局方式位标记（有无加密）
08 00 // 压缩方式
```

### 压缩源文件目录区

```
50 4B 01 02 // 目录区，目录中文件的文件头标记
3F 00 // 压缩使用的pkware版本
14 00 // 解压文件所需的pkware版本
00 00 // 全局方式位标记（有无加密）
08 00 // 压缩方式
```

### 压缩文件目录结束标志

```
50 4B 05 06 // 目录结束标志
```

## 常见压缩包格式分析

1.zip																ANSI ASCII		
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
00000000	50	4B	03	04	14	00	00	00	08	00	E8	51	2D	4C	3D	51	PK	èQ-L=Q
00000010	6B	4D	05	00	00	00	03	00	00	00	05	00	00	00	31	2E	kM	1.
00000020	74	78	74	33	34	34	04	00	50	4B	01	02	1F	00	14	00	txt344	PK
00000030	00	00	08	00	E8	51	2D	4C	3D	51	6B	4D	05	00	00	00		èQ-L=QkM
00000040	03	00	00	00	05	00	24	00	00	00	00	00	00	00	20	00		\$

无加密

1.zip																ANSI ASCII		
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
00000000	50	4B	03	04	14	00	00	00	08	00	E8	51	2D	4C	3D	51	PK	èQ-L=Q
00000010	6B	4D	05	00	00	00	03	00	00	00	05	00	00	00	31	2E	kM	1.
00000020	74	78	74	33	34	34	04	00	50	4B	01	02	1F	00	14	00	txt344	PK
00000030	09	00	08	00	E8	51	2D	4C	3D	51	6B	4D	05	00	00	00		èQ-L=QkM
00000040	03	00	00	00	05	00	24	00	00	00	00	00	00	00	20	00		\$

伪加密

1.zip																ANSI ASCII		
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
00000000	50	4B	03	04	14	00	09	00	63	00	E8	51	2D	4C	00	00	PK	c èQ-L
00000010	00	00	21	00	00	00	03	00	00	00	05	00	0B	00	31	2E	!	1.
00000020	74	78	74	01	99	07	00	02	00	41	45	03	08	00	E5	4F	txt "	AE áO
00000030	6E	3A	0E	2D	22	F2	D4	78	67	E8	CA	A4	0C	12	DB	00	n: -"òôxgèÈm	Ù
00000040	7E	55	17	EE	EB	53	47	FD	17	F7	DA	7C	49	9D	57	50	~U ièSGý	-Ù I WP
00000050	4B	07	08	00	00	00	00	21	00	00	00	03	00	00	00	50	K	! P
00000060	4B	01	02	1F	00	14	00	09	00	63	00	E8	51	2D	4C	00	K	c èQ-L

真加密

# 常见文件的文 识

0:11:15

jpeg (jpg)	FF D8 FF
png	89 50 4E 47
bmp	42 4D 36 5D
gif	47 49 46 38
zip	50 4B 03 04
rar	52 61 72 21
wav	57 41 56 45

CSDN @seven-BING

Hex editor window: a2eb7ceaf5ab49f7acb33de2e7eed74a.zip

编辑方式: 十六进制 (H) | 运行脚本 | 运行模板: ZIP.bt

00h:	50 4B 03 04	14 03 00 00	08 00 68 BF	9B 48 FE 32	PK.....h;
10h:	7D 4B E9 0D	00 00 35 1B	00 00 09 00	00 00 73 74	}Ké...µ....
20h:	65 67 6F 2E	74 78 74 7D	59 C9 76 E2	48 10 B0 EF	不一样, 请仔细加密
30h:	57 E6 22 24	33 CF 1C 38	8C 68 83 C4	18 7A 00 A3	Wæ"§\$İ.8ChfÄ
40h:	ED A6 C5 0F	01 12 30 0D	08 C4 D7 4F	44 56 09 68	í!À 0 Äxc
36 7A C7 00	3A B1 B6 F5	21 AD F0 CC	FC DB F8 0F	6zÇ.:±¶ó/-èiUø.	
50 4B 01 02	3F 03 14 03	09 00 08 00	58 BF 9B 48	PK..?.....h;>H	
FE 32 7D 4B	E9 0D 00 00	B5 1B 00 00	09 00 24 00	p2}Ké...µ....\$.	
00 00 00 00	00 00 20 80	ED 81 00 00	00 00 73 74	..... eí.....st	
65 67 6F 2E	74 78 74 0A	00 20 00 00	00 00 00 01	ego.txt.. .....	
00 18 00 80	0B 49 BF 9D	A0 D1 01 80	A7 42 38 B7	...€.I¿. Ñ.€SB8.	
2F D4 01 00	11 AA 37 B7	2F D4 01 50	4B 05 06 00	/ô...ª7-/â PK	
00 00 00 01	00 01 00 5B	00 00 00 10	0E 00 00 00	.....	

CSDN @seven-BING

00000DE0	14 2D DE 6A EC B9 36 4F 18 ED EC 71 DA E5 FB FA	-þji'60 iiqUâúú
00000DF0	B5 8E 01 5B 68 F9 8F 24 74 78 50 F1 8E E7 E3 0B	µl [hù \$txPñlçã
00000E00	36 7A C7 00 3A B1 B6 F5 2F AD E8 CC FC DB F8 0F	6zÇ :±¶ö/-èlùÛø
00000E10	50 4B 01 02 3F 03 14 03 00 00 08 00 68 BF 9B 48	PK ? h¿IH
00000E20	FE 32 7D 4B E9 0D 00 00 B5 1B 00 00 09 00 24 00	þ2}Ké µ \$
00000E30	00 00 00 00 00 00 20 80 ED 81 00 00 00 00 73 74	i st
00000E40	65 67 6F 2E 74 78 74 0A 00 20 00 00 00 00 00 01	ego.txt
00000E50	00 18 00 80 0B 49 BF 9D A0 D1 01 80 A7 42 38 B7	I I¿ Ñ  \$B8·
00000E60	2F D4 01 00 11 AA 37 B7 2F D4 01 50 4B 05 06 00	/Ô a7·/Ô PK
00000E70	00 00 00 01 00 01 00 5B 00 00 00 10 0E 00 00 00	[

名称	修改日期	类型	大小
stego.txt	2016/4/27 23:59	文本文档	7 KB

解压出来。

解压后发现是一堆base64加密的字符串，看题目来源是olympicCTF，这是俄罗斯2014年有道 misc 题是关于 Base64的隐写题，那我们直接写解码，这里的思路是先循环解密base64字符串，提取出可以隐写的最后2-4位，再拼接最后转回ascii码flag就出来了，下面是python2脚本

```
#coding=utf-8
def get_base64_diff_value(s1, s2):
    base64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    res = 0
    for i in xrange(len(s2)):
        if s1[i] != s2[i]:
            return abs(base64chars.index(s1[i]) - base64chars.index(s2[i]))
    return res

def solve_stego():
    with open('G:/Desktop/1.txt', 'rb') as f:
        file_lines = f.readlines()
        bin_str = ''
        for line in file_lines:
            steg_line = line.replace('\n', '')
            norm_line = line.replace('\n', '').decode('base64').encode('base64').replace('\n', '')
            diff = get_base64_diff_value(steg_line, norm_line)
            print diff
            pads_num = steg_line.count('=')
            if diff:
                bin_str += bin(diff)[2:].zfill(pads_num * 2)
            else:
                bin_str += '0' * pads_num * 2
            print goflag(bin_str)

def goflag(bin_str):
    res_str = ''
    for i in xrange(0, len(bin_str), 8):
        res_str += chr(int(bin_str[i:i + 8], 2))
    return res_str

if __name__ == '__main__':
    solve_stego()
```

```
3 base64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
4 res = 0
5 for i in xrange(len(s2)):
6     if s1[i] != s2[i]:
7         return abs(base64chars.index(s1[i]) - base64chars.index(s2[i]))
8     return res
9
10 def solve_stego():
11     with open('G:/Desktop/1.txt', 'rb') as f:
```

Run: test x

```
0
Base_sixty_four_point_fiv
4
Base_sixty_four_point_five
0
Base_sixty_four_point_five
0
Base_sixty_four_point_five
0
Base_sixty_four_point_five
```

CSDN @seven-BING

最后得到flag为: flag{Base\_sixty\_four\_point\_five }

## 第十二题：功夫再高也怕菜刀

题目描述：菜狗决定用菜刀和菜鸡决一死战

返回 本题用时: 32秒

### 功夫再高也怕菜刀

👍 53 最佳Writeup由B301 • dals提供 WP 建议

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

CSDN @seven-BING

打开文件是一个流量包、

acfff53ce3fa4e2bbe8654284dfc18e1... 2021/10/7 14:15 Wireshark captu... 2,310 KB

提示说菜刀，那么我们就筛选信息。

acfff53ce3fa4e2bbe8654284dfc18e1.pcapng

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

http

分组字节流 英语 区分大小写 字符串 flag 查找 取消

No.	Time	Source	Destination	Protocol	Length	Info
184	8.282759602	192.168.25.128	192.168.43.83	TCP	54	47844 → 80 [ACK] Seq=912 Ack=346 Win=30016 Len=0
185	8.484945747	192.168.25.128	192.168.43.83	TCP	290	47844 → 80 [PSH, ACK] Seq=912 Ack=346 Win=30016 Len=236 [TCP segment of a reassembled
186	8.486681298	192.168.43.83	192.168.25.128	TCP	60	80 → 47844 [ACK] Seq=346 Ack=1148 Win=64240 Len=0
187	8.486797464	192.168.25.128	192.168.43.83	HTTP	779	POST /upload/1.php HTTP/1.1 (application/x-www-form-urlencoded)
188	8.488527592	192.168.43.83	192.168.25.128	TCP	60	80 → 47844 [ACK] Seq=346 Ack=1873 Win=64240 Len=0
189	8.491449977	192.168.43.83	192.168.25.128	HTTP	474	HTTP/1.1 200 OK (text/html)
190	8.491488677	192.168.25.128	192.168.43.83	TCP	54	47844 → 80 [ACK] Seq=1873 Ack=766 Win=31088 Len=0

> Frame 189: 474 bytes on wire (3792 bits), 474 bytes captured (3792 bits) on interface 0  
 > Ethernet II, Src: Vmware\_f5:c2:5f (00:50:56:f5:c2:5f), Dst: Vmware\_21:b8:f4 (00:50:56:21:b8:f4)  
 > Internet Protocol Version 4, Src: 192.168.43.83, Dst: 192.168.25.128  
 > Transmission Control Protocol, Src Port: 80, Dst Port: 47844, Seq: 346, Ack: 1873, Len: 420  
 > Hypertext Transfer Protocol

Line-based text data: text/html (6 lines)

```

->|./\t2017-12-08 11:38:58\t0\t0777\n
..|\t2017-12-08 11:39:10\t4096\t0777\n
1.php\t2017-12-08 11:33:16\t33\t0666\n
flag.txt\t2017-12-08 11:35:29\t17\t0666\n
hello.zip\t2017-12-08 09:32:36\t224\t0666\n
|<-
  
```

0180	31 36 09 33 33 09 30 36	36 36 0a 66 6c 61 67 2e	16-33-06 66-flag.
0190	74 78 74 09 32 30 31 37	2d 31 32 2d 30 38 20 31	txt-2017 -12-08 1
01a0	31 3a 33 35 3a 32 39 09	31 37 09 30 36 36 0a	1:35:29- 17-0666-
01b0	68 65 6c 6c 6f 2e 7a 69	70 09 32 30 31 37 2d 31	hello.zip-2017-1

CSDN @seven-BING

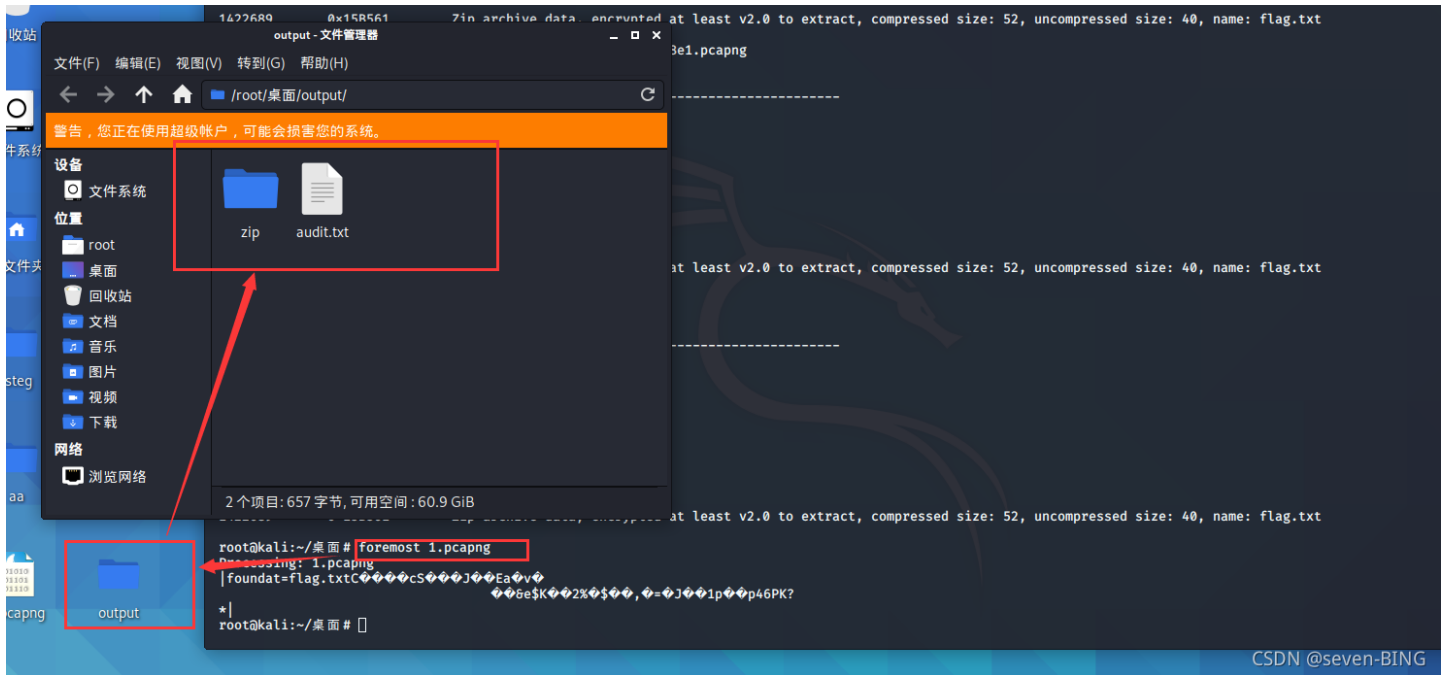
提示里面包含一个flag.txt文件，于是再kali里面先用binwalk查看一下，也可以用windows中binwalk工具链接

```

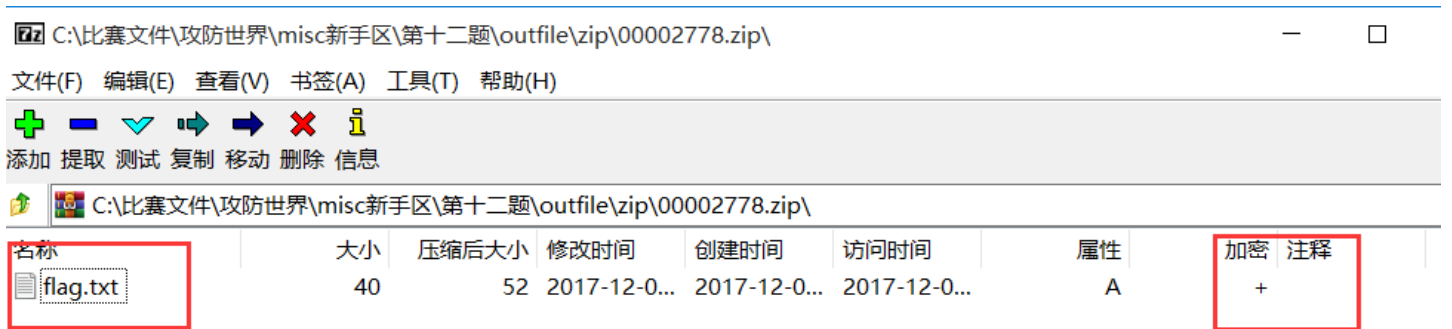
422689 0x15B561 Zip archive data, encrypted at least v2.0 to extract, compressed size: 52, uncompressed size: 40, name: flag.txt
root@kali:~/桌面# binwalk acfff53ce3fa4e2bbe8654284dfc18e1.pcapng
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
363085      0xA1E2D      xz compressed data
364045      0xA21ED      xz compressed data
312025      0xC63F9      xz compressed data
314001      0xC6BB1      xz compressed data
3238637     0x12E66D     xz compressed data
3240937     0x12EF69     xz compressed data
3391563     0x153BCB     xz compressed data
3393067     0x1541AB     xz compressed data
3406647     0x1576B7     xz compressed data
3412887     0x158F17     xz compressed data
3422689     0x15B561     Zip archive data, encrypted at least v2.0 to extract, compressed size: 52, uncompressed size: 40, name: flag.txt
root@kali:~/桌面#
  
```

CSDN @seven-BING

说明存在flag.txt,使用kali 中foremost进行分离。也可以用windows中foremost工具链接



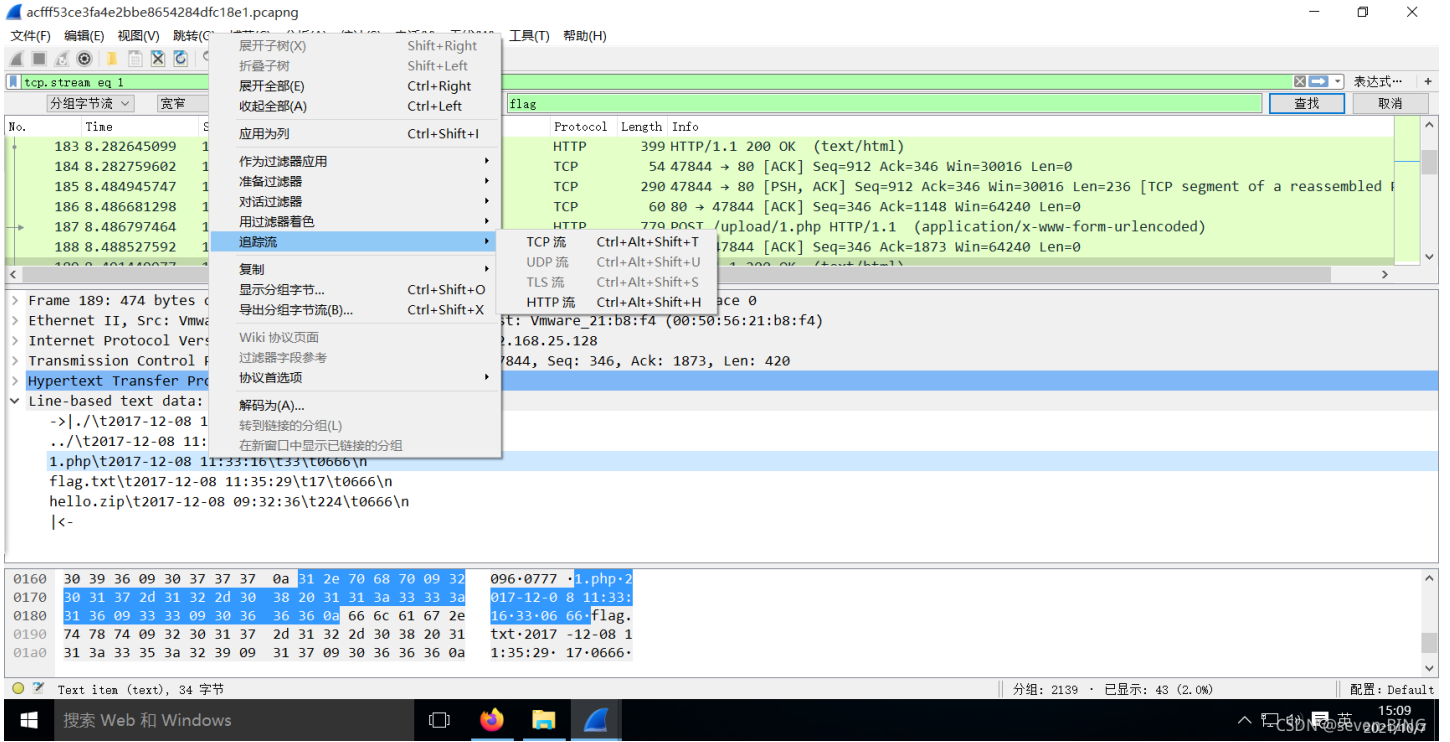
用7z工具打开, 需要密码:



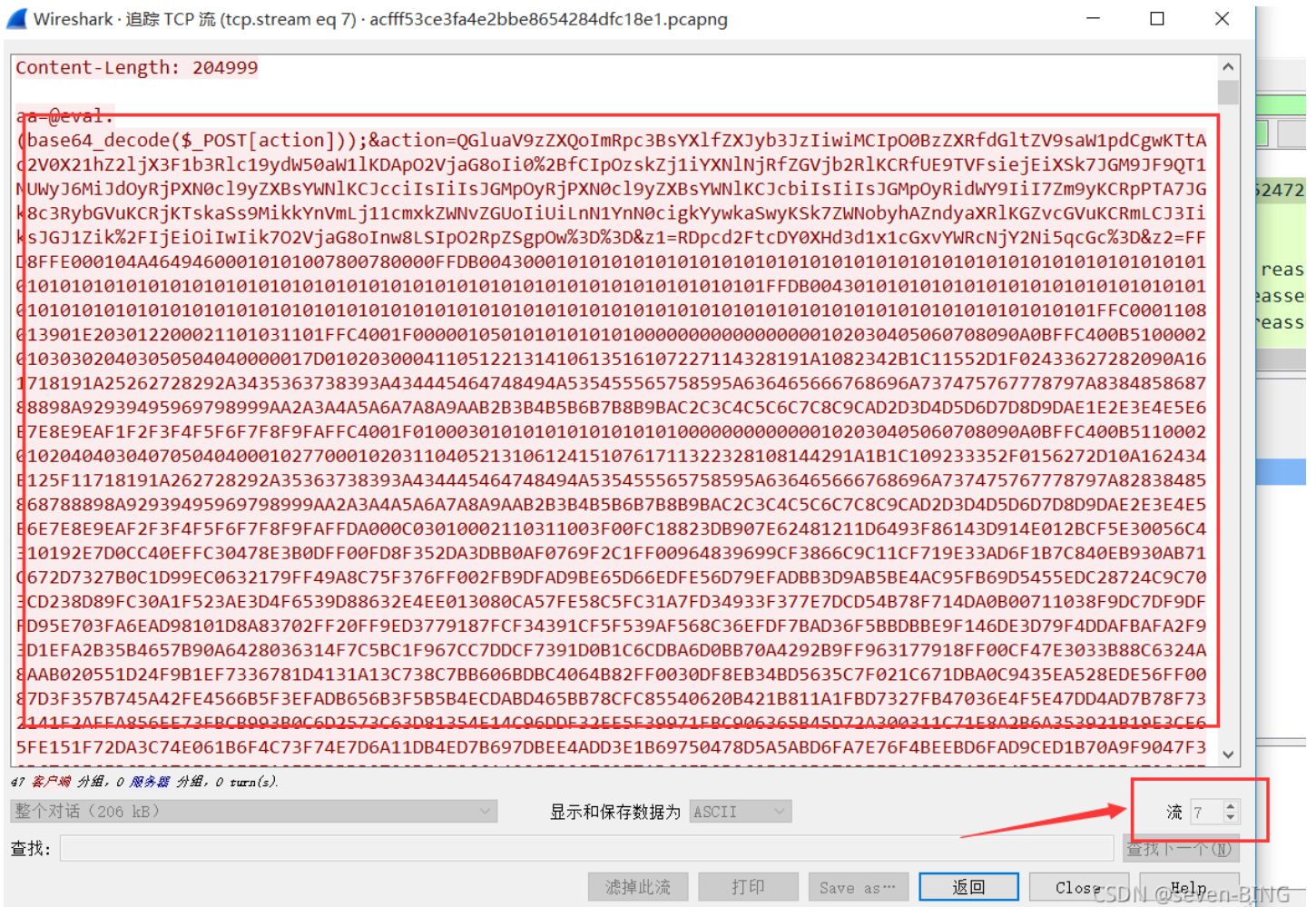
于是, 转头又分析流量包, 发现刚刚找flag时候出现的6666.jpg有点可疑。

右击, 点击追踪流, 点击tcp流

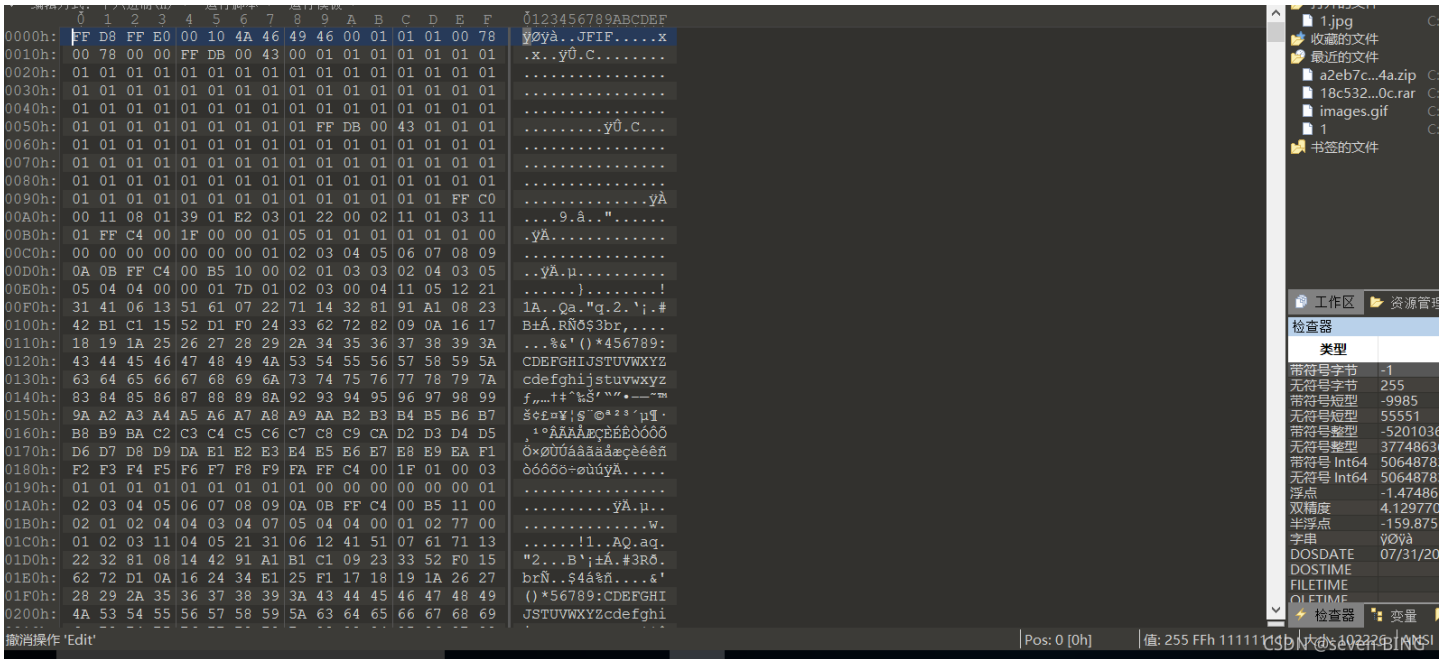
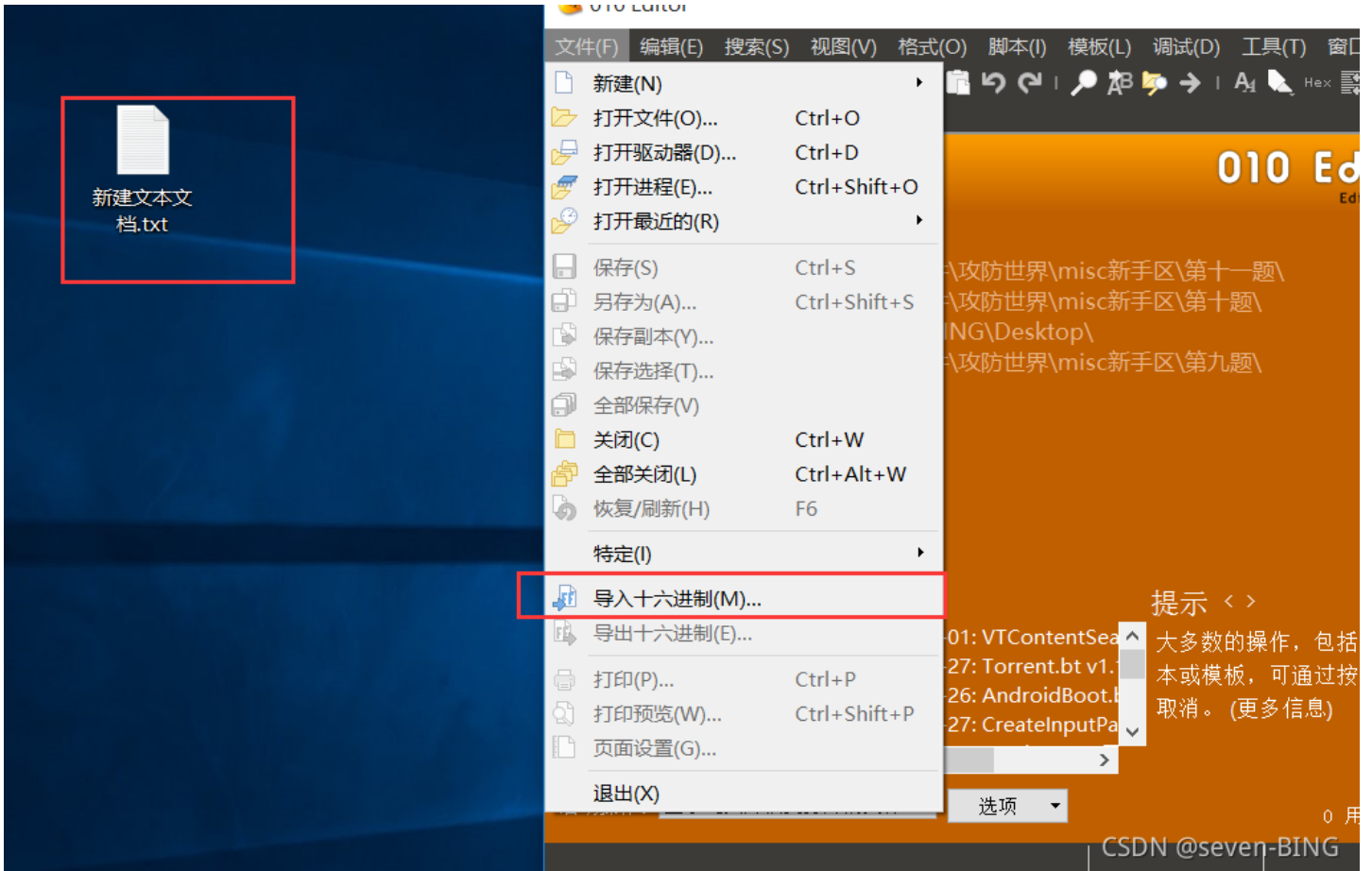




当流到第七个的时候，出现了可疑的数据流。







点击另存为1.jpg

图片就出来



CSDN @seven-BING

第一次猜测这个是flag，但是输入进去不对，那么猜测就是压缩包密码：Th1s\_1s\_p4sswd\_!!!

SO

你不爱我了吗?



你说啥



点赞  
是最好的鼓励

请大家多多关注我哟



CUTE