

BUUCTF MISC刷题笔记(一)

原创

[z.volcano](#) 于 2021-05-03 16:32:30 发布 2002 收藏 4

分类专栏: [# buuoj # 刷题](#)

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/weixin_45696568/article/details/116330991

版权



[buuoj](#) 同时被 2 个专栏收录

7 篇文章 1 订阅

订阅专栏



刷题

8 篇文章 0 订阅

订阅专栏

BUUOJ

Misc

[喵喵喵](#)

[弱口令](#)

[\[安淘杯 2019\]easy misc](#)

[\[XMAN2018排位赛\]通行证](#)

[蜘蛛侠呀](#)

[\[RCTF2019\]draw](#)

[\[MRCTF2020>Hello_misc](#)

[\[MRCTF2020\]Unravel!!](#)

[\[BSidesSF2019\]zippy](#)

[\[UTCTF2020\]basic-forensics](#)

[粽子的来历](#)

Misc

[喵喵喵](#)

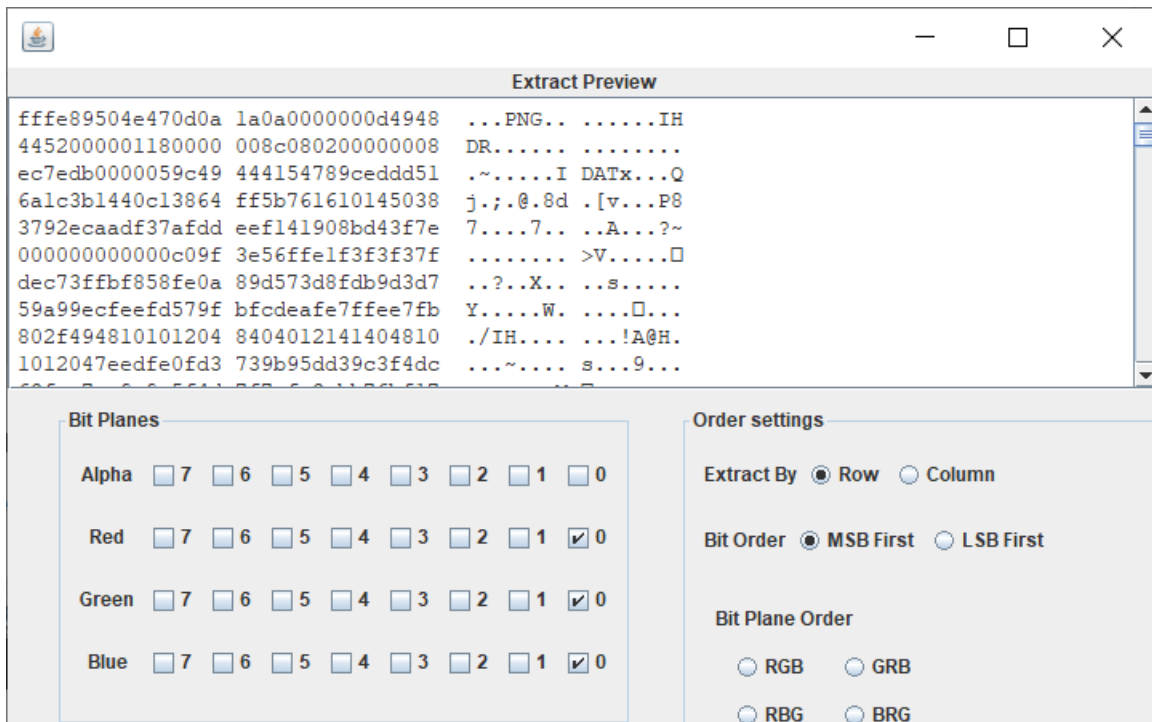


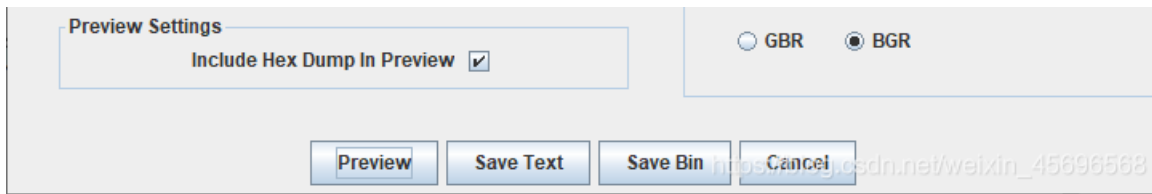
下载的得到一个图片

首先右键查看属性、用winhex打开未发现隐藏的信息，然后用StegSlove打开，在最低位通道发现，很可能是LSB隐写



选项如下时，发现隐写了一个png文件





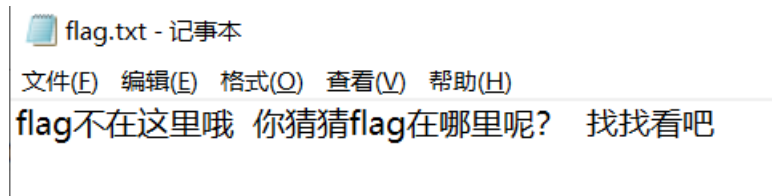
导出之后发现文件头多了一些东西，删去后发现有半个二维码

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI
FF	FE	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	ÿþPNG
44	52	00	00	01	18	00	00	00	8C	08	02	00	00	00	08	DP

分辨率是280 * 140，所以改成280 * 280，png图片的宽高修改还是比较方便的，把右边的数据改成和左边一致

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PN
00000016	00	00	01	18	00	00	00	8C	08	02	00	00	00	08	EC	7E	
00000032	DB	00	00	05	9C	49	44	41	54	78	9C	ED	DD	51	6A	1C	Û

发现这个二维码的颜色不对劲，把黑白反色(借助StegSlove)后扫码得到一个网盘地址: <https://pan.baidu.com/s/1pLT2J4f>，下



载之后得到flag.rar，里面有一个flag.txt，打开即被调戏

这个题是比赛原题，当时有一个hint是 **NTFS**，buuoj中没有给出

扫描发现flag.txt中隐藏了一个pyc文件，导出，放到pyc在线反编译网站

<input checked="" type="checkbox"/>	C:\Users\17422\Desktop\flag.txt:flag.pyc	flag.pyc	755	1
-------------------------------------	--	----------	-----	---

得到一个加密脚本

```
#!/usr/bin/env python
# visit http://tool.lu/pyc/ for more information
import base64

def encode():
    flag = '*****'
    ciphertext = []
    for i in range(len(flag)):
        s = chr(i ^ ord(flag[i]))
        if i % 2 == 0:
            s = ord(s) + 10
        else:
            s = ord(s) - 10
        ciphertext.append(str(s))

    return ciphertext[::-1]

ciphertext = [
    '96',
    '65',
    '93',
    '123',
    '91',
    '97',
    '22',
    '93',
    '70',
    '102',
    '94',
    '132',
    '46',
    '112',
    '64',
    '97',
    '88',
    '80',
    '82',
    '137',
    '90',
    '109',
    '99',
    '112']
```

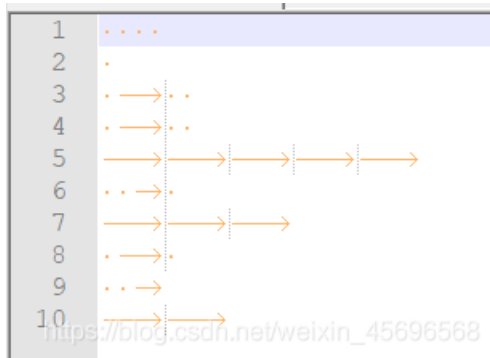
分析一下写个解密脚本就跑出flag了

弱口令

压缩包是加密的，注释信息处有线索



复制到记事本里，用notepad++打开，猜想是摩斯密码



把 tab 换成 - 空格 换成 .

得到 - - - - - - - - - - - - - - - ,在线解密得到压缩包密码 HELLOFORUM

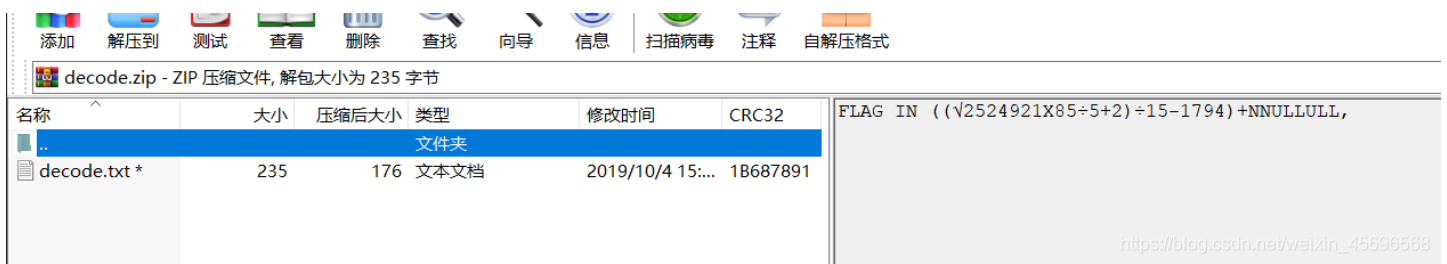
压缩包里有一个图片，用Stegsolve打开发现lsb隐写痕迹，Data Extract功能没有结果，结合题目描述和题目名，可以推断是 以弱口令为密码的lsb隐写

```
D:\lsb隐写>python2 lsb.py extract C:\Users\17422\Desktop\女神.png flag.txt 123456
[+] Image size: 500x500 pixels.
[+] Written extracted data to flag.txt.
```

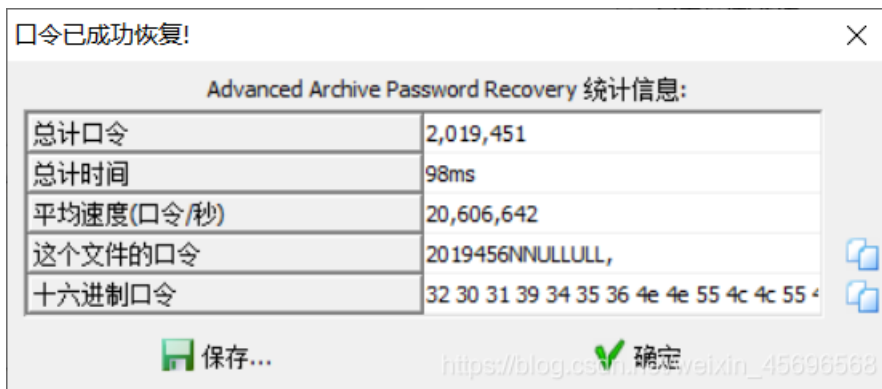
得到flag

[安洵杯 2019]easy misc

下载附件得到文件夹read、加密的压缩包和一张图片，先看压缩包



给了解压密码的提示，那个算式算出来结果是7，所以提示: FLAG IN 7+NNULLULL,，开始以为IN后面的就是密码，结果不是，几经测试后发现密码是七个数字+ NNULLULL, 的形式，于是掩码爆破



打开得到的 decode.txt 给了解密的规则:

```
a = dIW
b = sSD
c = adE
d = jVf
e = QW8
f = SA=
g = jBt
h = 5RE
i = tRQ
j = SPA
k = 8DS
l = XIE
m = S8S
n = MkF
o = T9p
p = PS5
q = E/S
r = -sd
s = SQW
t = obW
u = /WS
v = SD9
w = cw=
x = ASD
y = FTa
z = AE7
```

再结合文件夹给出的大量文本，可以知道考察的是 **字频隐写**。给出的图片，使用foremost分离出两张看似一模一样的图片，很明显是盲水印，解出信息提示 **字频隐写在11.txt** (我的py2出了点问题不能演示了)。

再加上文件夹中有一个hint.txt

```
hint:取前16个字符
```

所以分析11.txt中最 **高频** 的 **前十六个字母**，附上脚本

```
f=open("11.txt","r")
d={}
s=""
for line in f.readlines():
    line=line.lower().replace(" ","")
    for i in line:
        if 'a'<=i<='z':
            d[i]=d.get(i,0)+1
d1=sorted(d.items(), key = lambda kv:(kv[1], kv[0]),reverse=True)
for j in range(16):
    print(d1[j][0],end="")
f.close()
```

得到这十六个字母 **etaonrhsidluygw**,按照规则替换得到 **QW8obWdIWT9pMkFSQWtRQjVfXiE/WSFTajBtcw==**，base64解密得到 **Ao(mgHY?i2ARAKQB5_?!?Y!Sj0ms**，base85解密得到 **flag{have_a_good_day1}**

[XMAN2018排位赛]通行证

下载得到一个txt，内容是

```
a2FuYmJyZ2doamx7emJfX19ffXZ0bGFsbg==
```

这里用CyberChef，先base64解密一次，然后再栅栏加密一次(key=7)

The screenshot shows the CyberChef web interface. On the left, under the 'Recipe' section, there are two operations: 'From Base64' and 'Rail Fence Cipher Encode'. The 'From Base64' operation has 'Alphabet' set to 'A-Za-z0-9+/' and 'Remove non-alphabet chars' checked. The 'Rail Fence Cipher Encode' operation has 'Key' set to '7' and 'Offset' set to '0'. On the right, the 'Input' field contains the base64 string 'a2FuYmJyZ2doamx7emJfX19ffXZ0bGFsbg=='. The 'Output' field at the bottom shows the result: 'kzna{blnl_abj_lbh_trg_vg}'.

最后就是凯撒解码，把前面的xman换成flag即可

AmanCTF - 凯撒(Caesar)加密/解密

在线凯撒(Caesar)加密/解密

kzna{bInI_abj_lbh_trg_vg}

偏移量

加密

解密

枚举

bqer{scec_rsa_csy_kix_mx}
[rdbd_qrz_brx_jhw_lw}
qaca_pqy_aqw_igv_kv}
ynbo{pzbz_opx_zpv_hfu_ju}
xman{oyay_now_you_get_it}
wlzm{nxzx_mnv_xnt_fds_hs}
vkyl{mwyw_lmu_wms_ecr_gr}
ujxk{lvxv_klt_vlr_dbq_fq}
tiwj{kuwu_jks_ukq_cap_ep}

蜘蛛侠呀

得到一个流量包，用 **wireshark** 打开，没发现什么

用 **tshark** 命令看一下

```
tshark -r out.pcap -T fields -e data > 1.txt
```


打开1.txt, 发现大量十六进制数值, 且有重复的

```
242453544152542424494e666148346139614a6f776771334848774159354874724e4466594c555a
242453544152542424494e666148346139614a6f776771334848774159354874724e4466594c555a
242453544152542424494e666148346139614a6f776771334848774159354874724e4466594c555a
242453544152542424494e666148346139614a6f776771334848774159354874724e4466594c555a
24245354415254242459567959734975447170382b58546270506664695444786b6151386864345f
24245354415254242459567959734975447170382b58546270506664695444786b6151386864345f
24245354415254242459567959734975447170382b58546270506664695444786b6151386864345f
24245354415254242459567959734975447170382b58546270506664695444786b6151386864345f
242453544152542424564e5a4c35426a2f30327341514a6868474672466d42574d42486361584e5
242453544152542424564e5a4c35426a2f30327341514a6868474672466d42574d42486361584e5
242453544152542424564e5a4c35426a2f30327341514a6868474672466d42574d42486361584e5
24245354415254242464664673722f485a6669547a5864566a65317451465937794a6f687a52322i
24245354415254242464664673722f485a6669547a5864566a65317451465937794a6f687a52322i
24245354415254242464664673722f485a6669547a5864566a65317451465937794a6f687a52322i
242453544152542424446c51686a7a4b376a534735647749327557586f632f71506c55305341744f
242453544152542424446c51686a7a4b376a534735647749327557586f632f71506c55305341744f
242453544152542424446c51686a7a4b376a534735647749327557586f632f71506c55305341744f
242453544152542424446c51686a7a4b376a534735647749327557586f632f71506c55305341744f
24245354415254242458315464472f736477492f6631554e7a6476554b63737a557879533362796
24245354415254242458315464472f736477492f6631554e7a6476554b63737a557879533362796
24245354415254242458315464472f736477492f6631554e7a6476554b63737a557879533362796
242453544152542424727356492b78773776474277517674774b764a6b666e32413051775143564;
```

写脚本去重, 再把每一行的十六进制转字符。

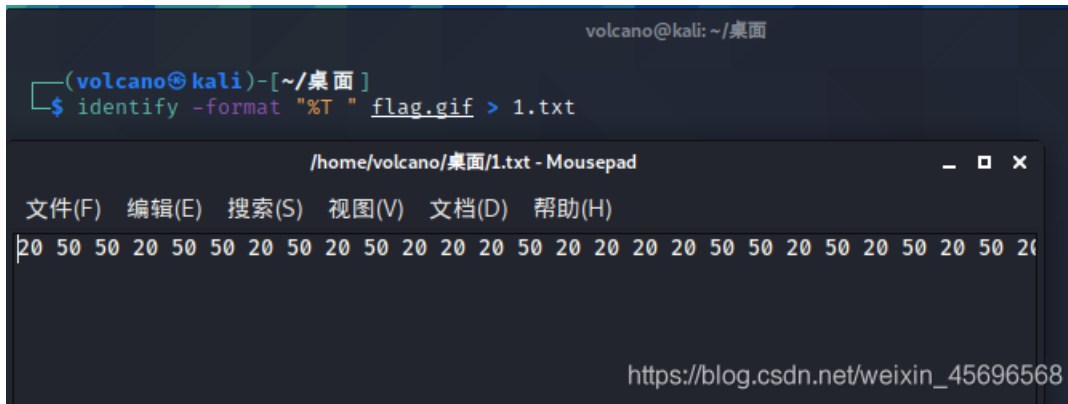
```
3.txt - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)
$$$START$$$ + 5srB91Ram3cONSZmjif9aXY9K1UCR/re0qvhPy9cDDkRZOt6BvvtsnaYwbb0LI
$$$START$$$ + AWwJ/ekSjMgOz + vh80Uqs3Fn3WM9PQqW5vap2ZvFNJZKsQv0xXD21xT+rPKs2
$$$START$$$ + CJWqZqpQMgbzM + RFG4zUM382ILDR6tild2Qrb + CxEFIK + KhQ7uXMA55qB7aT4n
$$$START$$$ + EWNtcWkEG6Z33hiPs5NcubPyVL9N6PwnhT/cU818pfCUSO/LbrgLUZkf8yqM7B
$$$START$$$ + Y1juVt + waLPTv3rrmT + MK/BPhVyXicM398lm1c4cz7ljpWF2JFEn8L4QQ1FK2
$$$START$$$ + ZCkzP8mmw27xs2N + BmskRrHELIxyH7qqyTdnIRXMuWoZC3zymN + fxU3 + y0ZhEd
$$$START$$$ + hklkPJt0jSmuglhFpY8CH5LACblgS2DjMPjGwuqpn01iA9A3V/cpSkglDnJ1WH
$$$START$$$ + qndQh1j + 2zjPzBKkXYv9ArN8aFlgHCcEE65lo7QTnE7lx9GxKC5wqLUj0MDpQ9
$$$START$$$ + yP30lJPwpBgv0Mb + 6GxG8oamr//KUW/O818uDuHfZeW6v3v + UCeyuSpUYujbZG
$$$START$$$ + /6Vlf57JROpQbl3rSsvbjEPfyHB5NxtbEVsQrqmHUO/f3V0TMfb5wZ9j3ZYEh26
$$$START$$$ + /S949Puf59eMXgcL1PsnVsdKC6d2LnDOj7f/vxZm/tLkcaXKynOah7QEgNIe/kL
$$$START$$$ + /laalKCvdmoTaH3UZe1gh5veP4mHOMYMBwX8zyWvV8ix3dieOXNEDX1y + FxaNul
$$$START$$$ + /pJOs81PbpbLnI/h85W7uYYZFRhdhHeHhV7My3AVspz6lYMenXcTdTjFgJqrYJ
$$$START$$$ + /tj8JdxNsXK1buWJ5tdayeej9tq706Kmxr/t1vQ7KzBD685xwdxrSnDjH0KbbI
$$$START$$$ + 02VFfwLtiy5mNcAVdmLsjGglBiQfgJ0el/ObU + 82Cafiwq + JUii3GNRru3lKk8E
$$$START$$$ + 0DNzMGfQ7C + 3aXfgvi/8S//QZ7/89YEGU7Qg7uvac2m2A0hsjTvR5OY/J7Htb0K
$$$START$$$ + 0MnrntRB6sg2kKaME4Kr4jhMJFniCAFGPC9Do0EuRcmfgOm64KiCaaqNPYLb07U
$$$START$$$ + 0MylsG7qdDjfsPIDwxP + buqMonlrbkY/y2BCLXA3olMtrsL/GDRh2D1luVbf5pu
$$$START$$$ + 0RPbKdEI25xODNJpEA3WBlq4DUQ311bMsgs162S9BtXdQplJo + qn6MFIsbeJ7rf
$$$START$$$ + 0TbskVicyyP5fef9n368v2pQwPPWO3SL + egf5bTlAS7kMqYzQLwf4Vg089LtpHb
$$$START$$$ + 0X9o7RNiP3F2biewPuP6lZg/Hkhf4BBfb/27kQDd8/GqqWxTXw/b3AfnHrQ/kuf
$$$START$$$ + 0sMjhm6g3u26n49QuhCQDPSAv + OSaV5oulMZ5oECtaPCUpt4dzZPHcTxrIVzjkw
第 5 行, 第 57 列 100% Unix (LF) UTF-8
```

写脚本去除多余字符, 剩下的像是base64编码, 写脚本

```
import base64

f = open("3.txt", "rb").read()
f1 = open("x", "wb")
f1.write(base64.b64decode(f))
```

得到的是一个zip文件，打开得到一个gif，这里是 [gif帧数间隔隐写](#)



把20换成0、50换成1，得到 `011011010100010000110101010111110011000101110100`

转为十六进制后再转字符，然后md5加密，最后得到 `flag{f0f1003afe4ae8ce4aa8e8487a8ab3b6}`

[RCTF2019]draw

给了一些看不太懂的东西，结合题目名，应该是要画出来

```
cs pu lt 90 fd 500 rt 90 pd fd 100 rt 90 repeat 18[fd 5 rt 10] lt 135 fd 50 lt 135 pu bk 100 pd setcolor pick [ red orange yellow green blue violet ] repeat 18[fd 5 rt 10] rt 90 fd 60 rt 90 bk 30 rt 90 fd 60 pu lt 90 fd 100 pd rt 90 fd 50 bk 50 setcolor pick [ red orange yellow green blue violet ] lt 90 fd 50 rt 90 fd 50 pu fd 50 pd fd 25 bk 50 fd 25 rt 90 fd 50 pu setcolor pick [ red orange yellow green blue violet ] fd 100 rt 90 fd 30 rt 45 pd fd 50 bk 50 rt 90 fd 50 bk 100 fd 50 rt 45 pu fd 50 lt 90 pd fd 50 bk 50 rt 90 setcolor pick [ red orange yellow green blue violet ] fd 50 pu lt 90 fd 100 pd fd 50 rt 90 fd 25 bk 25 lt 90 bk 25 rt 90 fd 25 setcolor pick [ red orange yellow green blue violet ] pu fd 25 lt 90 bk 30 pd rt 90 fd 25 pu fd 25 lt 90 pd fd 50 bk 25 rt 90 fd 25 lt 90 fd 25 bk 50 pu bk 100 lt 90 setcolor pick [ red orange yellow green blue violet ] fd 100 pd rt 90 arc 360 20 pu rt 90 fd 50 pd arc 360 15 pu fd 15 setcolor pick [ red orange yellow green blue violet ] lt 90 pd bk 50 lt 90 fd 25 pu home bk 100 lt 90 fd 100 pd arc 360 20 pu home
```

后面百度了一波，得知这个是 [logo语言](#)，可以参考这个贴子

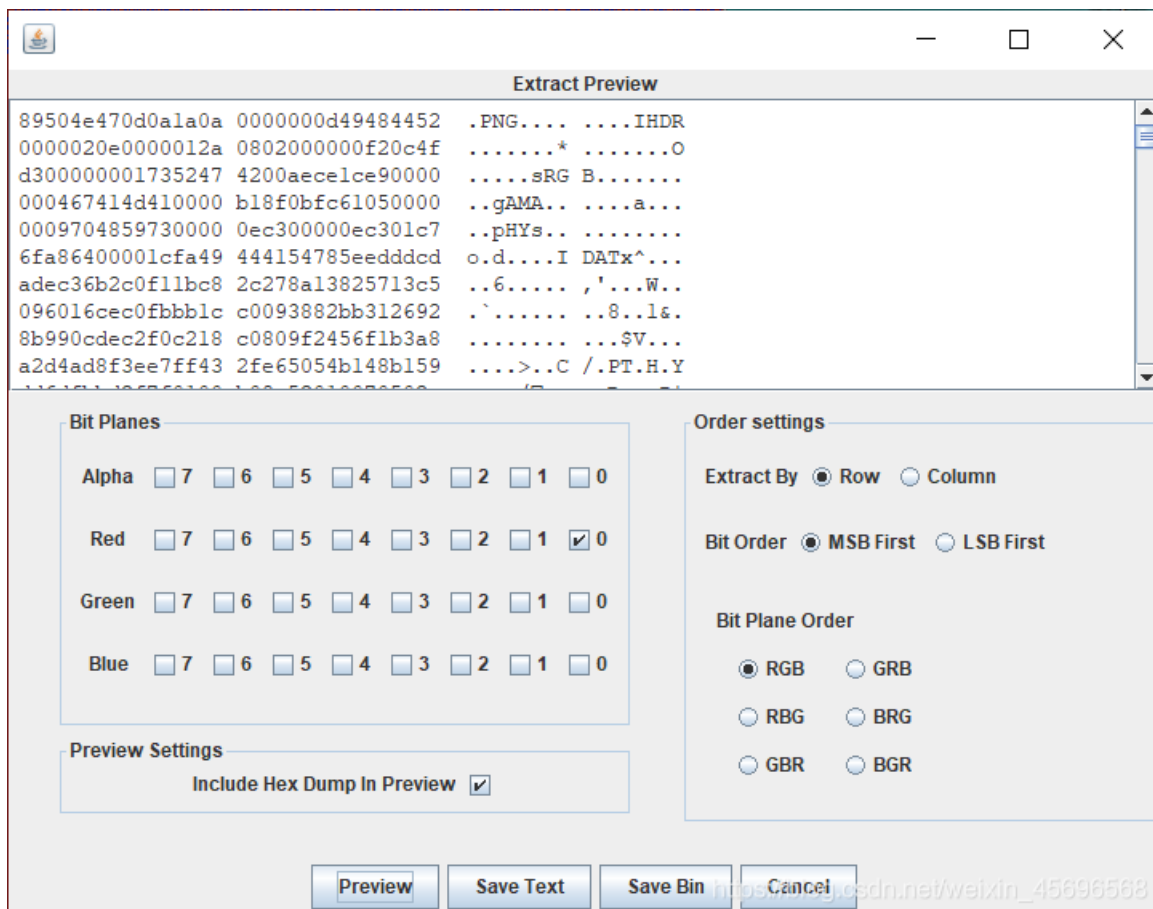
直接到这个[在线网站](#)运行即可



所以 `flag{RCTF_HeyLogo}`

[MRCTF2020>Hello_misc

用Stegsolve打开图片，在红色通道隐写了东西



save bin保存为png图片，给了zip压缩包的密码 `!@#%*67*()-+$!\@#\%67*()-+$`，题目给的是rar，所以还需要找出这个zip

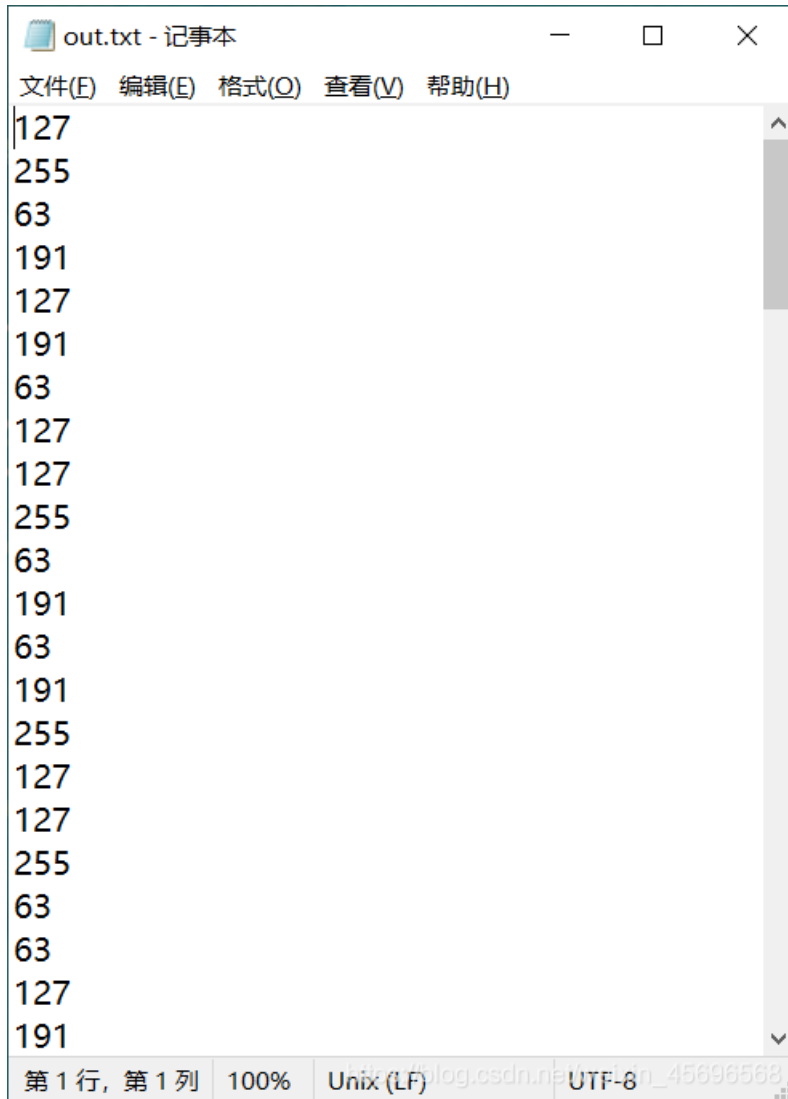
Maybe you should try to separate the files!

And I will give u zip-passwd:

!@#\$%67*()-+

https://blog.csdn.net/weixin_45696568

直接foremost，提取出zip，用密码解密，得到的out.txt中由几种数字组成



```
out.txt - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)
127
255
63
191
127
191
63
127
127
255
63
191
63
191
255
127
127
255
63
63
127
191
第 1 行, 第 1 列 100% Unix (LF) UTF-8
```

把这些数字分别转为八位的二进制，发现后面六位都是1，只有前两位有区别，写脚本把所有组的前两位连起来，再转字符。

```
f = open("out.txt", "r")
flag=y=x=""
for line in f.readlines():
    x = bin(int(line))[2:].zfill(8)
    y += x[:2]
f.close()
for i in range(len(y)//8):
    flag += chr(int(y[i*8:(i+1)*8],2))
print(flag)
```

得到rar的密码 `rar-passwd:0ac1fe6b77be5dbe`，得到的新zip很明显是一个doc文件，改后缀为doc

最下面有一些透明字，把颜色改深就可以看到

```
␣
␣
␣
␣
␣
␣
MTEwMTEwMTEwMTEwMTEwMDEwMTEwMTEwMTEwMTEwMTEwMTEwMTEwMTEwMTEwMTA
xMTEwMDAwMDAxMTEwMTEwMTEwMDAxMTAx␣
MTEwMTEwMTEwMDAxMTAxMDEwMTEwMTEwMTEwMTEwMTEwMTEwMTEwMTEwMTEwMTE
xMTAxMTEwMTEwMTEwMTEwMTEwMTEwMDEw␣
MTEwMDAwMTAxMTEwMTEwMDEwMTEwMTEwMTEwMTAwMDEwMTEwMTEwMTEwMDAxMDAxM
TAxMTEwMDAwMDEwMTEwMDAwMDEwMTEwMTEw␣
MTEwMTEwMTAwMDAxMTEwMDEwMTEwMTEwMTEwMDEwMTAxMTEwMTEwMTEwMTEwMTEwM
TAxMTEwMTEwMTAwMTEwMTEwMTEwMTEw␣
MTEwMTEwMTAxMTEwMTEwMDEwMTEwMTEwMTAxMDEwMTAxMTEwMTEwMTEwMTEwMTEwMT
AxMTAxMTEwMTAwMTEwMTEwMTEwMTEw␣
MTEwMTEwMTAwMDAxMTAwMDAwMTEwMDAwMDAxMTAwMDEwMTAwMDAwMTEwMTEwMTEw
TEwMTAxMTEwMDAwMDAxMTEwMDAwMDEwMTEwMTEw␣
```

https://blog.csdn.net/weixin_45696568

得到的这些像是base64编码，分别解码得到一串二进制

The screenshot shows an online Base64 decoder. The 'Input' field contains the Base64 string from the previous image. The 'Output' field shows the decoded binary data. The tool interface includes a 'Recipe' section with settings for alphabet and character removal.

这里不能这样看，把结果平均分为六行，再把1替换成空格就能看到flag

```
1  ..0..0.....00..0.....0..000000.....00..0..
2  ..0..0..000..0..0.....0.....0.....0.....0..00..
3  ..0000..0..0.....0.....000.....00..00..0..00000.....0000.....
4  ..0..0..0000.....0.....0.....0.....0.....0.....00..0.....
5  ..0..0..0.....0.....0.....0.....0.....0.....00..0.....
6  ..0..0..0000.....00000.....000000.....000.....00000.....0000.....
```

flag{He1Lo_mi5c~}

[MRCTF2020]Unravel!!

拿到三个文件

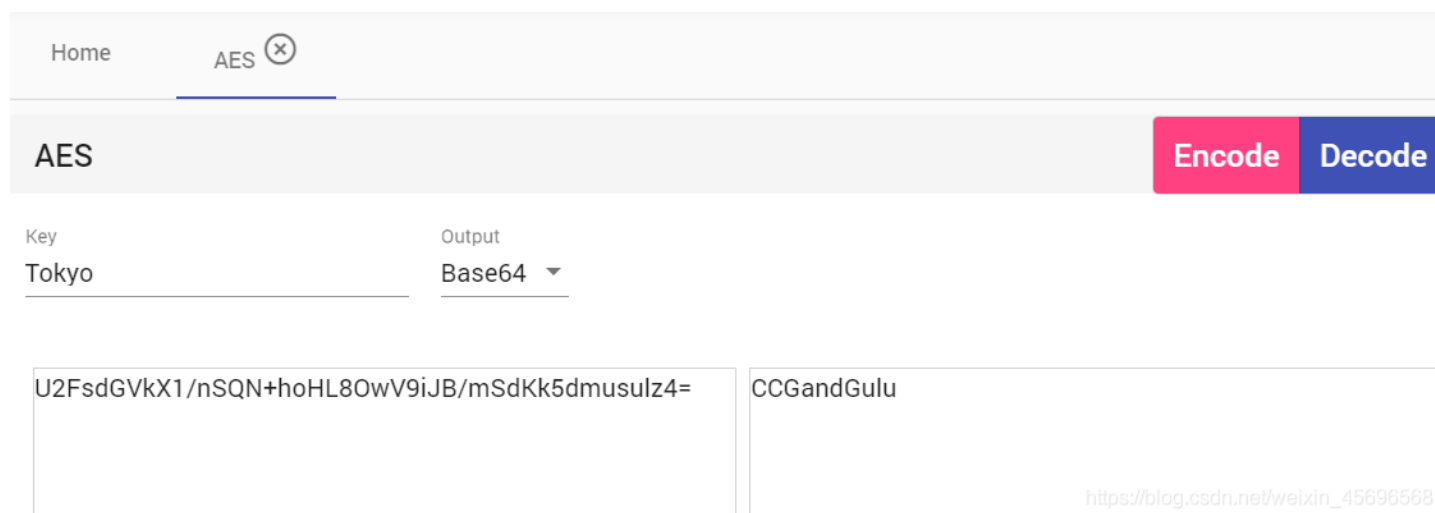
			文件夹			
win-win.zip	4,957,332	4,957,332	WinRAR ZIP 压缩...	2020/3/12 21:...	7424C34C	
JM.png	407,605	398,554	PNG 文件	2020/3/12 21:...	361DEAAE	
Look_at_the_file_ending.wav	6,078,590	3,713,450	WAV 文件	2020/3/15 16:...	99277858	

先看一下wav，文件名提示让看一下文件的尾部，所以用010打开看一下

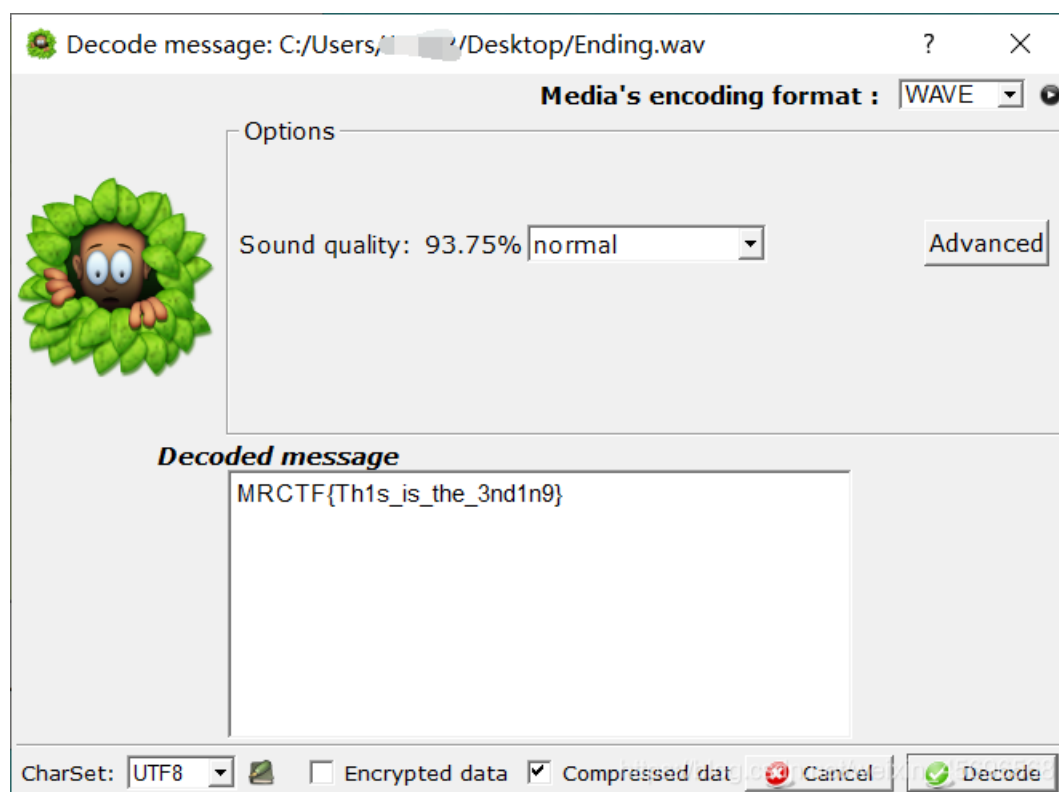
```
5C:C020h: B7 FC E9 FC 9B FC F5 FC 8D FC FD FC 74 FC F1 FC ·ueu>uou.uyutunu
5C:C030h: 4E FC E2 FC 2B FC E2 FC 10 FC D6 FC 03 FC B0 FC Nûâü+ûâü.üÖü.ü°ü
5C:C040h: 0C FC 84 FC 25 FC 54 FC 3C FC 24 FC 44 FC 6B 65 .ü,,ü%üTü<ü$üDüke
5C:C050h: 79 3D 55 32 46 73 64 47 56 6B 58 31 2F 6E 53 51 y=U2FsdGVkX1/nSQ
5C:C060h: 4E 2B 68 6F 48 4C 38 4F 77 56 39 69 4A 42 2F 6D N+hoHL8OwV9iJB/m
5C:C070h: 53 64 4B 6B 35 64 6D 75 73 75 6C 7A 34 3D SdKk5dmusulz4=
```

U2F开头，很明显是AES了，需要密钥，在这个文件里找了一下没找见

再去看一下那个图片，直接binwalk一把梭，拿到一个图片，内容就是前面需要的秘钥：Tokyo
解得key=CCGandGulu

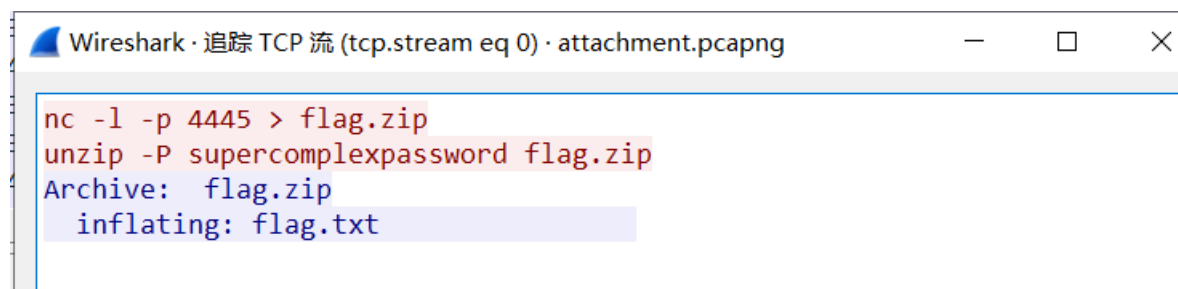


拿得到的key去解压压缩包，得到一个wav文件，这里是比较坑的，没找到提示，尝试了一遍发现要用 SilentEye



[BSidesSF2019]zippy

流量包，wireshark打开分析，追踪TCP流时发现线索



```
Wireshark · 追踪 TCP 流 (tcp.stream eq 0) · attachment.pcapng
nc -l -p 4445 > flag.zip
unzip -P supercomplexpassword flag.zip
Archive:  flag.zip
  inflating: flag.txt
```

直接binwalk分离出压缩包，密码是： `supercomplexpassword`，解压拿到flag

[UTCTF2020]basic-forensics

下载的jpg打不开，拖进winhex里看一下，发现里面都是明文

所以用notepad打开，ctrl+f搜索一下 **flag**、**key**等关键字

然后找到了 `utflag{fil3_ext3nsi0ns_4r3nt_r341}`

粽子的来历

曹操的私生子曹小明因为爸爸活着的时候得罪太多人，怕死后被抄家，所以把财富保存在一个谁也不知道的地方。曹小明比较喜欢屈原，于是把地点藏在他的诗中。三千年后，小明破译了这个密码，然而却因为担心世界因此掀起战争又亲手封印了这个财富并仿造当年曹小明设下四个可疑文件，找到小明喜欢的DBAPP标记，重现战国辉煌。(答案为正确值(不包括数字之间的空格)的小写32位md5值) 注意：得到的 flag 请包上 flag{} 提交

鬼脑洞题...

乍一看，四个文件的内容好像是一样的

帝高阳之苗裔兮，朕皇考曰伯庸。↵
摄提贞于孟陬兮，惟庚寅吾以降。↵
皇览揆余初度兮，肇锡余以嘉名：↵
名余曰正则兮，字余曰灵均。↵
纷吾既有此内美兮，又重之以修能。↵
扈江离与辟芷兮，纫秋兰以为佩。↵
汨余若将不及兮，恐年岁之不吾与。↵
朝搴阰之木兰兮，夕揽洲之宿莽。↵
日月忽其不淹兮，春与秋其代序。↵
唯草木之零落兮，恐美人之迟暮。↵
不抚壮而弃秽兮，何不改乎此度？↵
乘骐骥以驰骋兮，来吾道夫先路！↵

↵

https://blog.csdn.net/weixin_45696568

但是仔细对比会发现 **行距不一样**

帝高阳之苗裔兮，朕皇考曰伯庸。↵
摄提贞于孟陬兮，惟庚寅吾以降。↵
皇览揆余初度兮，肇锡余以嘉名：↵
名余曰正则兮，字余曰灵均。↵
纷吾既有此内美兮，又重之以修能。↵
扈江离与辟芷兮，纫秋兰以为佩。↵
汨余若将不及兮，恐年岁之不吾与。↵
朝搴阰之木兰兮，夕揽洲之宿莽。↵
日月忽其不淹兮，春与秋其代序。↵
唯草木之零落兮，恐美人之迟暮。↵
不抚壮而弃秽兮，何不改乎此度？↵
乘骐骥以驰骋兮，来吾道夫先路！↵

↵

帝高阳之苗裔兮，朕皇考曰伯庸。↵
摄提贞于孟陬兮，惟庚寅吾以降。↵
皇览揆余初度兮，肇锡余以嘉名：↵
名余曰正则兮，字余曰灵均。↵
纷吾既有此内美兮，又重之以修能。↵
扈江离与辟芷兮，纫秋兰以为佩。↵
汨余若将不及兮，恐年岁之不吾与。↵
朝搴阰之木兰兮，夕揽洲之宿莽。↵
日月忽其不淹兮，春与秋其代序。↵
唯草木之零落兮，恐美人之迟暮。↵
不抚壮而弃秽兮，何不改乎此度？↵
乘骐骥以驰骋兮，来吾道夫先路！↵

https://blog.csdn.net/weixin_45696568

把每一行右键→段落，查看行距，把**1.5倍行距**换成**1**、**单倍行距**换成**0**

ABCD对应的分别是

- A 100111100010
- B 100100100001
- C 100100100001
- D 010100100001

挨个测试，最后正确的是C对应的二进制值转md5

`flag{d473ee3def34bd022f8e5233036b3345}`