2021年“绿盟杯”重庆市大学生信息安全竞赛—Light1ng战队 Writeup

原创

Le1a 于 2021-10-23 16:22:10 发布 603 收藏 5

分类专栏： CTF 文章标签： 网络安全 1024程序员节

本文链接：https://blog.csdn.net/weixin_52091458/article/details/120922444

2021年“绿盟杯”重庆市大学生信息安全竞赛—Light1ng战队Writeup

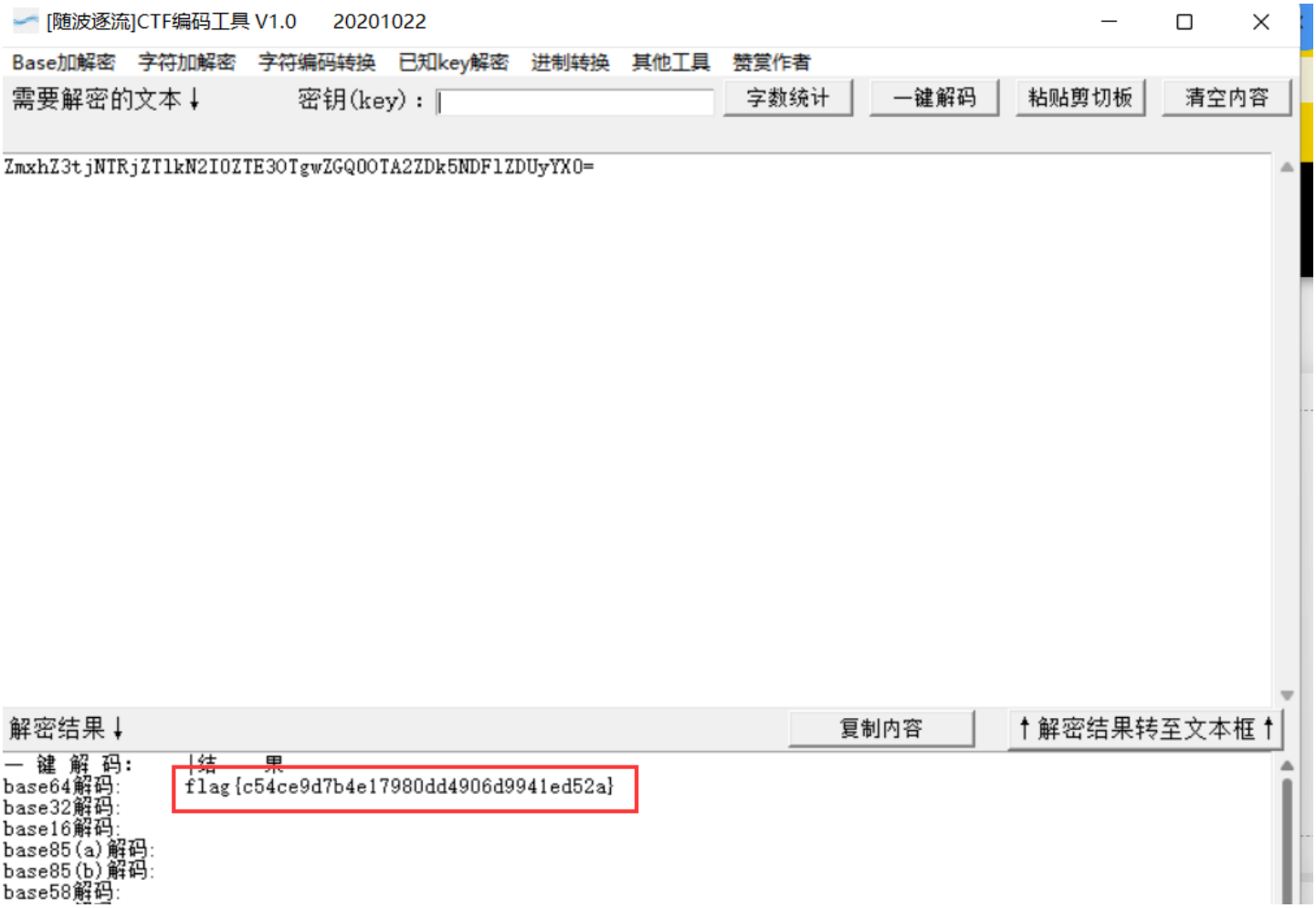
其余方向Writeup详见pdf:https://wws.lanzoui.com/iWk1ovo0eaj

密码:Le1a

Misc

Misc1:签到1

题目给了一串base64编码

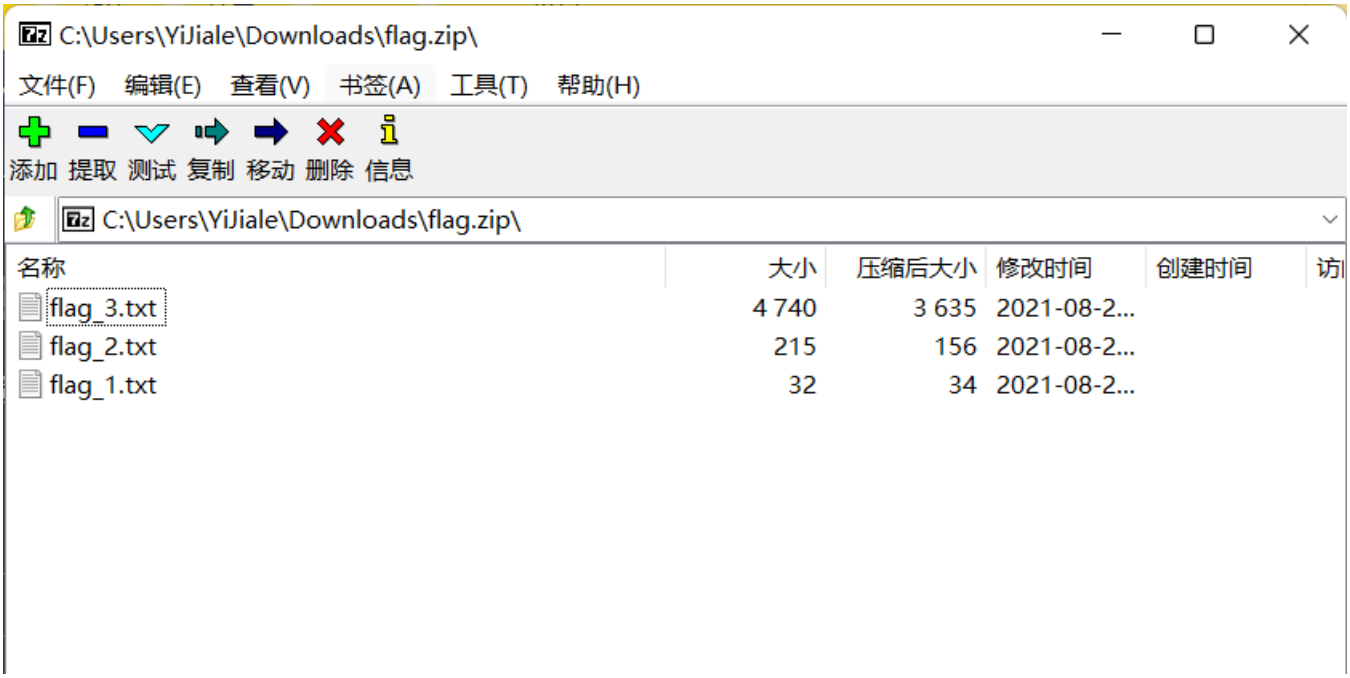


解码得到flag：

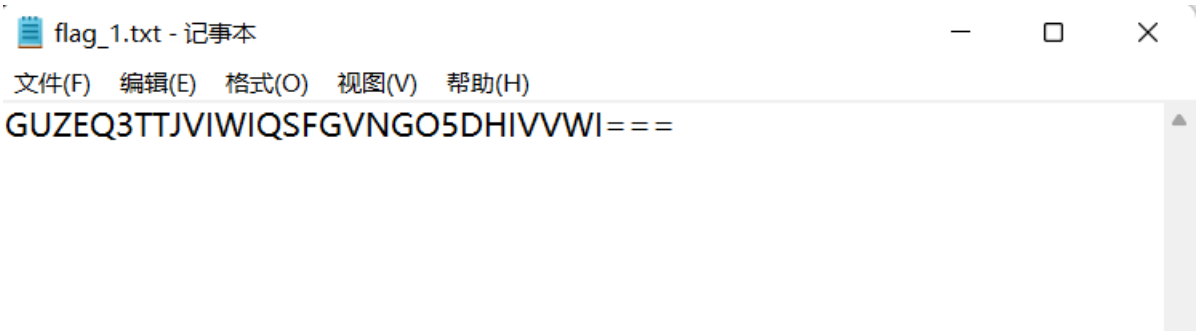
```
flag{c54ce9d7b4e17980dd4906d9941ed52a}
```

Misc2：DECODER

下载附件，打开得到3个txt，应该对应着三个部分的flag



flag_1.txt: 042f38b694



base32解码得到`52HnsMQdBE5ZgtgEkd`

再次base58解码得到`0JkOs1H8i%3A^`

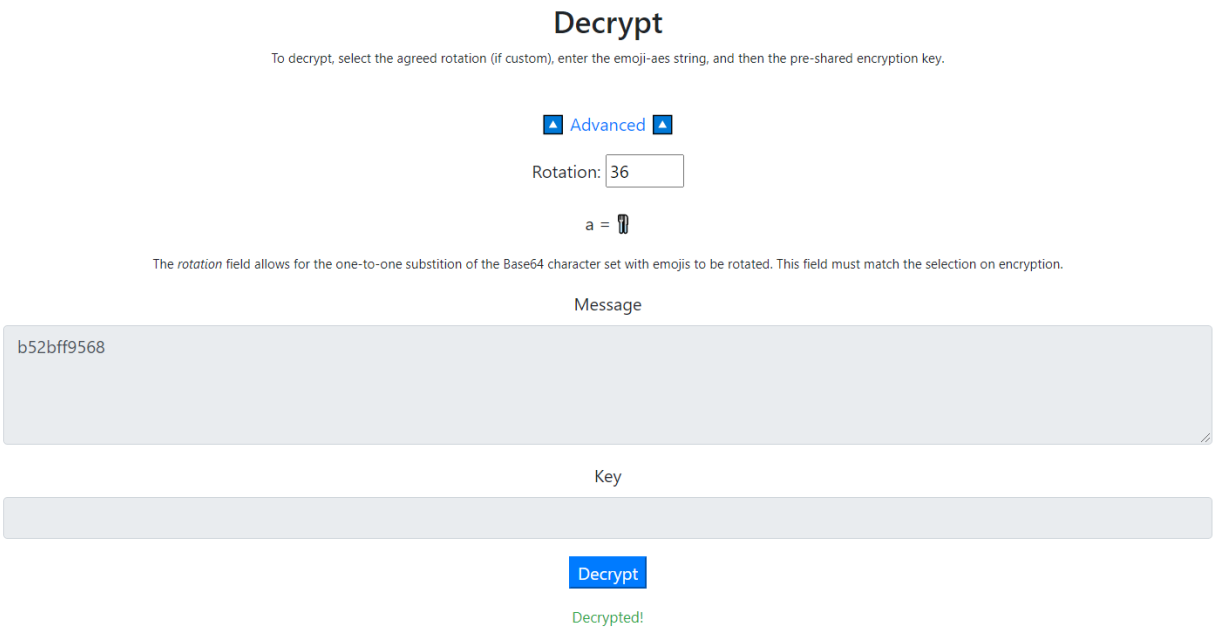
再次base85(a)解码得到`042f38b694`

`flag_2.txt:` b52bff9568

Key可以直接base100解码得到whhjno

然后进入emoji-aes解密网站 https://aghorler.github.io/emoji-aes/

直接解密不行，于是尝试进阶移位，调试到36位的时候，可以成功解密得到`b52bff9568`



flag_3.txt

打开得到一串字符，丢到随波逐流里面一键解码

发现是base91，解码得到：37f267472516

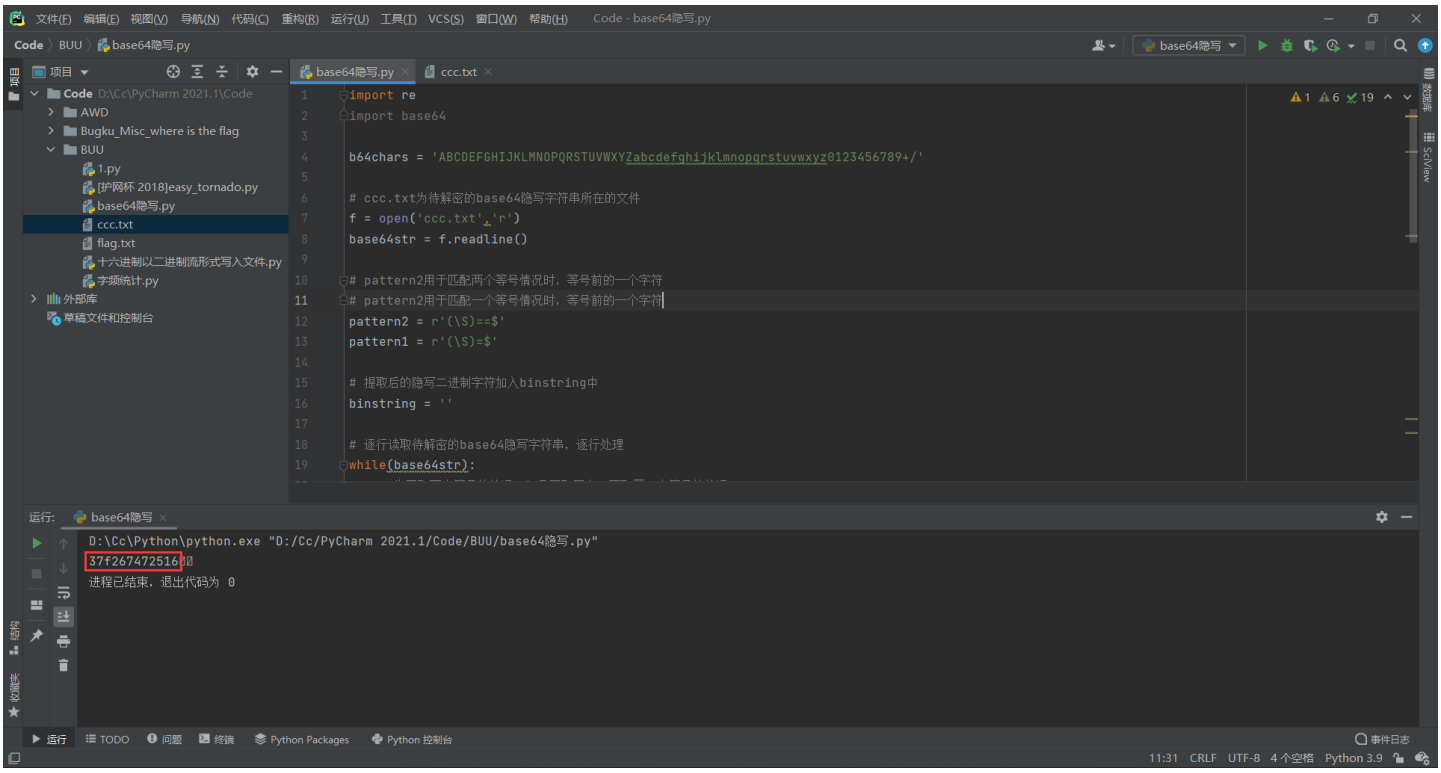
```
U3RlZ2Fub2dyYXBoeSBpcyB0aGUgYXJ0IGFuZCBzY2llbmNlIG9m
IHdyaXRpbmcgaGlkZGVuIG1lc3NhZ2VzIGluIHN1Y2ggYSB3YXkgdGhhdCBubyBvbmU=
LCBhcGFydCBmcm9tIHRoZSBzZW5kZXIgYW5kIGludGVuZGVkIHJlY2lwaWVudCwgc3VzcGX=
Y3RzIHRoZSBleGlzdGVuY2Ugb2YgdGhlIG1lc3M=
YWdlLCBhIGZvcm0gb2Ygc2VjdXJpdHkgdGhyb3VnaCBvYnNjdXJpdHkuIFT=
aGUgd29yZCBzdGVnYW5vZ3JhcGh5IGlzIG9mIEdyZWVrIG9yaWdpbiBhbmTgbWVhbnMgImNvbmNlYT==
bGVkIHdyaXRpbmciIGZyb20gdGhlIEdyZWVrIHdvcmRzIHN0ZWdhbm9zIG1lYW5pbmcgImNv
dmVyZWQgb3IgdHJvdGVjdGVkIiwgYW5kIGdyYXBoZWluIG1lYW5pbmdInRvIHd=
cml0ZSIuIFRoZSBmaXJzdCByZWNvcmRlZCB1c2Ugb2YgdGhlIHRlcm0gd2FzIGluIDE0OTkgYnkgSm9o
YW5uZXMgVHJpdGhlbWl1cyBpbiBoaXMgU3RlZ2Fub2dyYXBoaWEsIGEgdHJlYd==
dGlzZSBvbiBjcnl5dG9ncmF5aHkgYW5kIHN0ZWdhbm9ncmF5aHkgZGlzZ5==
dWlzZWQgYXOgYSBib29rIG9uIG1hZ2ljLiBHZW5lcmFsbHksIG1lc3O=
YWdlcyB3aWxsIGFwcGVhciB0byBiZSBzb21ldGhpbmcgZWxzZTogaW1hZ2VzLCBhcnRp
```

```
Y2xlcywgc2hvcHBpbmcgbGlzdHMsIG9yIHNvbWUgb3Q=
aGVyIGNvdmVydGV4dCBhbmQsIGNsYXNzaWNhbGx5LCB0aGUgaGlnZGVuIG1lc3NhZ2UgbWF5IGJlIGluIGludmn=
c2libGUgaW5rIGJldHdlZW4gdGhlIHZpc2libGUgbGluZXMgb2YgYSBwcml2YXRlIGxldHRlci4=
VGhlIGFkdmFudGFnZSBvZiBzdGVnYW5vZ3JhcGh5LCBvdmVyIGNy
eXB0b2dyYXBoeSBhbG9uZSwgaXMgdGhhdCBtZXNzYWdlcyBkbyBub3QgYXR0cmFjdCBhdHRlbnRpb26=
IHRvIHRoZW1zZWx2ZXMuIFBsYWlubHkgdmlzaWJsZSBlbmNyeXB0ZWQgbWVzc2FnZXPogbRubyBtYXR0ZXIg
aG93IHVuYnJlYWthYmxl6IG0d2lsbCBhcm91c2Ugcz==
dXNwaWNpb24sIGFuZCBtYXkgaW4gdGhlbXNlbHZlcyBiZSBpbmNyaW1pbmF0aW5nIG==
aW4gY291bnRyaWVzIHdoZXJlIGVuY3J5cHRpb24gaXMgaWxsZWdhbC4gVGhlcmVmb3JlLD==
IHdoZXJlYXMnY3J5cHRvZ3JhcGh5IHByb3RlY3RzIHRoZSBjb250ZW50cyBvZn==
IGEgbWVzc2FnZSwgc3RlZ2Fub2dyYXBoeSBjYW4gYmUgc2FpZCB0byBwcm90ZWN0IGI=
b3RoIG1lc3NhZ2VzIGFuZCBjb21tdW5pY2F0aW5nIHBhcnRpZXMu
U3RlZ2Fub2dyYXBoeSBpbmNsdWRlcyD=
dGhlIGNvbmNlYWxtZW51IG9mIGluZm9ybWF1aW9uIHdpdGhpbiBjb21=
cHV0ZXIgZmlsZXMuIEluIGRpZ2l0YWwgc3RlZ2Fub2dyYXBoeSwgZWxlY3Ryb25pYyBjb21tdW5pY2F0aW9u
cyBtYXkgaW5jbHVkZSBzdGVnYW5vZ3JhcGhpYyBjb2RpbmcgaW5zaQ==
ZGUgb2YgYSB0cmFuc3BvcnQgbGF5ZXIsIHN1Y2ggYXMgYSBkb2N1bWVudCBmaWxlLCBpbWFnZSBmaWz=
ZSwgcHJvZ3JhbSBvciBwcm90b2NvbC4gTWVkaWEg
ZmlsZXMgYXJlIGlkZWFsIGZvciBzdGVnYW5vZ3JhcGhyYyB0cmFuc21pc3Npb3==
biBiZWNhdXNlIG9mIHRoZWlyIGxhcmdlIHNpemUuIEFzID==
YSBzaW1wbGUgZXhhbXBsZSwgYSBzZW5kZXIgbWlnaHQgc3RhcnQgd2l0aCBh
biBpbm5vY3VvdXMgaW1hZ2UgZmlsZSBhbmQgYWRqdXN0IHRoZSBjb2xvciBiBldmVyeSAxMDB0aCBwaXhlbCA=
dG8gY29ycmVzcG9uZCB0byBhIGxldHRlciBpbiB0aGUgYWxwaGFiZXQsIGG=
IGNoYW5nZSBzbyBzdWJ0bGUgdGhhdCBzb21lb25lIG5vdCBzcGVjaWZpY2FsbHkgbG9va2luZyBm
b3IjaXQjaXMjdW5saWtlbHkjdG8jbm90aWNlIGl0Lj==
VGhlJGZpcnN0JHJlY29yZGVkJHVzZXMgb2Ygc3RlZ2Fub2dyYXBoeSBjYW4gYmUgdHJ=
YWNlZCBiYWNrIHRvIDQ0MCBCQyB3aGVuIEhlcm9kb3R1cyBtZW50aW9ucyB28gZXhhbXBsZXMgb0==
ZiBzdGVnYW5vZ3JhcGh5IGluIFRoZSBIaXN0b3JpZXMgb2Yg
SGVyb2RvdHVzLiBEZW1hcmF0dXMgc2VudCBhIHdhcm5pbmcgYWJvdXQgYSD=
Zm9ydGhjb21pbmcgYXR0YWNrIHRvIEdyZWVjZSBieSB3
cml0aW5nIGl0IGRpcmVjdGx5IG9uIHRoZSB3b29kZW4gYmFja2luZyBvZiBhIHdheCB0YWJsZXQgYmVm
b3JlIGFwcGx5aW5nIGl0cyBiZWVzd2F4IHN1cmZhY2UuIFdheCB0YWJsZXRzIHdlcmUgaW4gY29tbW9uIHVzZR==
IHRoZW4gYXMgcmV1c2FibGUgd3JpdGluZyBzdXJmYWNlcywgc29tZXRpbWU=
c3VzZWQgZm9yIHNob3J0aFuZC4gQW5vdGhlciBhbmNpZW50IGV4YW1wbGUgaXMgdGhhdCBv
Zkhpc3RpYWV1cywgd2hvIHNoYXZlCB0aGUgaGVhZCBieSBlW9kb3R1cyBtZW50aW9uIHZpZGVkIHP=
bGF2ZSBhbmQgdGF0dG9vZWQgYSBtZXNzYWdlIG9uIGl0LiBBZnRlciBoaXMgaGFpciBoYWQgZ2==
cm93biB0aGUgbWVzc2FnZSB3YXMgaGlkZGVuLiBUaGUgcHVycG9zZSB3YXMgdG8=
aW5zdGlnYXRlIGEgcmV2b2x0IGFnYWluc3QgdGhlIFBlcnNpYW5zLg==
U3RlZ2Fub2dyYXBoeSBoYXMgYmVlbiB3aWRlbHkgdXNlZCwg
aW5jbHVkaW5nIGluIHJlY2VudCBoaXN0b3JpY2FsIHRpbWVzIGFuZCB0
aGUgcHJlc2VudCBkYXkuIFBvc3NpYmlsIHBlcm11dGF0aW9ucyBhcmUgZW5kbGVzcyBhbmQ=
a25vd24gZXhhbXBsZXMgaW5jbHVkZTo=
SGlkZGVuIG1lc3NhZ2VzIHdpdGhpbiB3YXggdGE=
YmxldHM6IGluIGFuY2llbnQgR3JlZWNlLCBwZW9wbGUgd3JvdGUgbWU=
c3NhZ2VzIG9uIHRoZSB3b29kLCB0aGVuIGNvdmVyZWQgaXQgd2l0aCB3YXggdXBvbiB3aGljaCBhbiBpbm5vY2Vu
dCBjb3ZlcmluZyBtZXNzYWdlIHdhcyB3cml0dGVu
SGlkZGVuIG1lc3NhZ2VzIG9uIG1lc3NlbmdlcidzIGJvZHk6IGFsc28gdXNlZCBpbiBhbmNpZW4=
```

再次base64解码得到

```
Steganography is the art and science of writing hidden messages in such a way that no one
```

这么长的base64 不可能只得到这么一串英文，所以猜测base64隐写，通过python脚本得到37f267472516



脚本如下:

```
import re
import base64

b64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'

# ccc.txt为待解密的base64隐写字符串所在的文件
f = open('ccc.txt','r')
base64str = f.readline()

# pattern2用于匹配两个等号情况时，等号前的一个字符
# pattern2用于匹配一个等号情况时，等号前的一个字符
pattern2 = r'(\S)==$'
pattern1 = r'(\S)=$'

# 提取后的隐写二进制字符加入binstring中
binstring = ''

# 逐行读取待解密的base64隐写字符串，逐行处理
while(base64str):
    # 先匹配两个等号的情况，如果匹配不上，再配置一个等号的情况
    # 如果无等号，则没有隐藏，无需处理
    if re.compile(pattern2).findall(base64str):
        # mstr为等号前的一个字符，该字符为隐写二进制信息所在的字符
        mstr = re.compile(pattern2).findall(base64str)[0]
        # 确认mstr字符对应的base64二进制数，赋值给mbin
        mbin = bin(b64chars.find(mstr))
        # mbin格式如0b100，mbin[0:2]为0b
        # mbin[2:].zfill(6)为将0b后面的二进制数前面补0，使0b后面的长度为6
        mbin2 = mbin[0:2] + mbin[2:].zfill(6)
        # 两个等号情况隐写了4位二进制数，所以提取mbin2的后4bit
        # 赋值给stegobin，这就是隐藏的二进制信息
        stegobin = mbin2[-4:]
        binstring += stegobin
    elif re.compile(pattern1).findall(base64str):
        mstr = re.compile(pattern1).findall(base64str)[0]
        mbin = bin(b64chars.find(mstr))
        mbin2 = mbin[0:2] + mbin[2:].zfill(6)
        # 一个等号情况隐写了2位二进制数，所以提取mbin2的后2bit
        stegobin = mbin2[-2:]
        binstring += stegobin
    base64str = f.readline()

# stegobin将各行隐藏的二进制字符拼接在一起
# 从第0位开始，8bit、8bit处理，所以range的步进为8
for i in range(0,len(binstring),8):
    # int(xxx,2)，将二进制字符串转换为10进制的整数，再用chr()转为字符
    print(chr(int(binstring[i:i+8],2)),end='')
```

所以flag为：

```
flag{042f38b694b52bff956837f267472516}
```

Misc3：huahua

下载附件得到huahua.zip，但是打不开，于是丢到010中分析结构，发现文件头异常，正常的zip文件头是50 4B 03 04

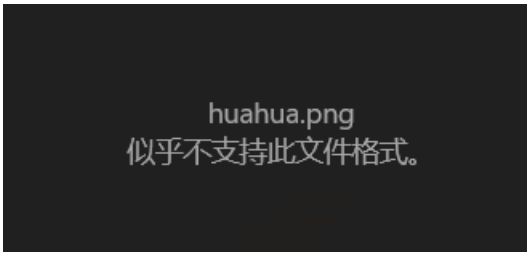
010 Editor - C:\Users\YiJiale\Desktop\huahua.zip

```
文件(F) 编辑(E) 搜索(S) 视图(V) 格式(O) 脚本(I) 模板(L) 调试(D) 工具(T) 窗口(W) 帮助(H)
huahua.zip ×
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0000h: 55 4A 13 14 14 00 00 00 08 00 64 6F B8 52 F4 B2 UJ........do¸Rô²
0010h: 9C 58 E6 C0 05 00 18 CC 05 00 0A 00 00 00 68 75 œXæÀ...Ì......hu
0020h: 61 68 75 61 2E 70 6E 67 EC BD F5 57 5C 41 D3 2D ahua.pngì½õW\AÓ-
0030h: 9C E0 12 5C 83 86 C1 75 D0 10 3C C8 00 81 E0 1E œà.\ƒ†ÁuÐ.<È..à.
0040h: 3C B8 BB 5B 70 06 CD 84 E0 16 34 48 70 87 C1 35 <¸»[p.Í„à.4Hp‡Á5
0050h: B8 0F EE EE EE F2 91 E4 79 DE F7 DE 3F E1 5B EB ¸.îîîò'äyÞ÷Þ?á[ë
0060h: FE C0 70 46 56 9F EE AE AA BD 77 75 D7 39 07 03 þÀpFVŸî®ª½wu×9..
```

010 Editor - C:\Users\YiJiale\Desktop\huahua.zip

```
文件(F) 编辑(E) 搜索(S) 视图(V) 格式(O) 脚本(I) 模板(L) 调试(D) 工具(T) 窗口(W) 帮助(H)
huahua.zip ×
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0000h: 50 4B 03 04 14 00 00 00 08 00 64 6F B8 52 F4 B2 PK........do¸Rô²
0010h: 9C 58 E6 C0 05 00 18 CC 05 00 0A 00 00 00 68 75 œXæÀ...Ì......hu
0020h: 61 68 75 61 2E 70 6E 67 EC BD F5 57 5C 41 D3 2D ahua.pngì½õW\AÓ-
0030h: 9C E0 12 5C 83 86 C1 75 D0 10 3C C8 00 81 E0 1E œà.\ƒ†ÁuÐ.<È..à.
```

保存之后，重新打开得到一张huahua.png，同样的图片也打不开，丢入010



010 Editor - C:\Users\YiJiale\Desktop\huahua.png

```
文件(F) 编辑(E) 搜索(S) 视图(V) 格式(O) 脚本(I) 模板(L) 调试(D) 工具(T) 窗口(W) 帮助(H)
起始页 huahua.png × 猫.png
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
000h: 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 00 00 02 F1 ........IHDR...ñ
010h: 00 00 01 F1 08 06 00 00 00 A2 FA 39 9A 00 00 00 ...ñ.....¢ú9š...
020h: 01 73 52 47 42 00 AE CE 1C E9 00 00 00 04 67 41 .sRGB.®Î.é....gA
030h: 4D 41 00 00 B1 8F 0B FC 61 05 00 00 00 09 70 48 MA..±..üa.....pH
040h: 59 73 00 00 12 74 00 00 12 74 01 DE 66 1F 78 00 Ys...t...t.Þf.x.
050h: 00 FF A5 49 44 41 54 78 5E EC FD 05 B8 1D 55 B6 .ÿ¥IDATx^ìý.¸.U¶
060h: B6 0D FF EF 7B BA 1B 77 77 77 02 21 40 70 4D E3 ¶.ÿï{º.www.!@pMã
070h: EE 6E 01 82 13 42 84 20 C1 DD 09 DE B8 05 B7 40 în.‚.B„ ÁÝ.Þ¸.·@
```

正常的49前面应该是89 50 4E 47 0D 0A 1A 0A 00 00 00 0D，huahua.png缺少了89 50 4E 47

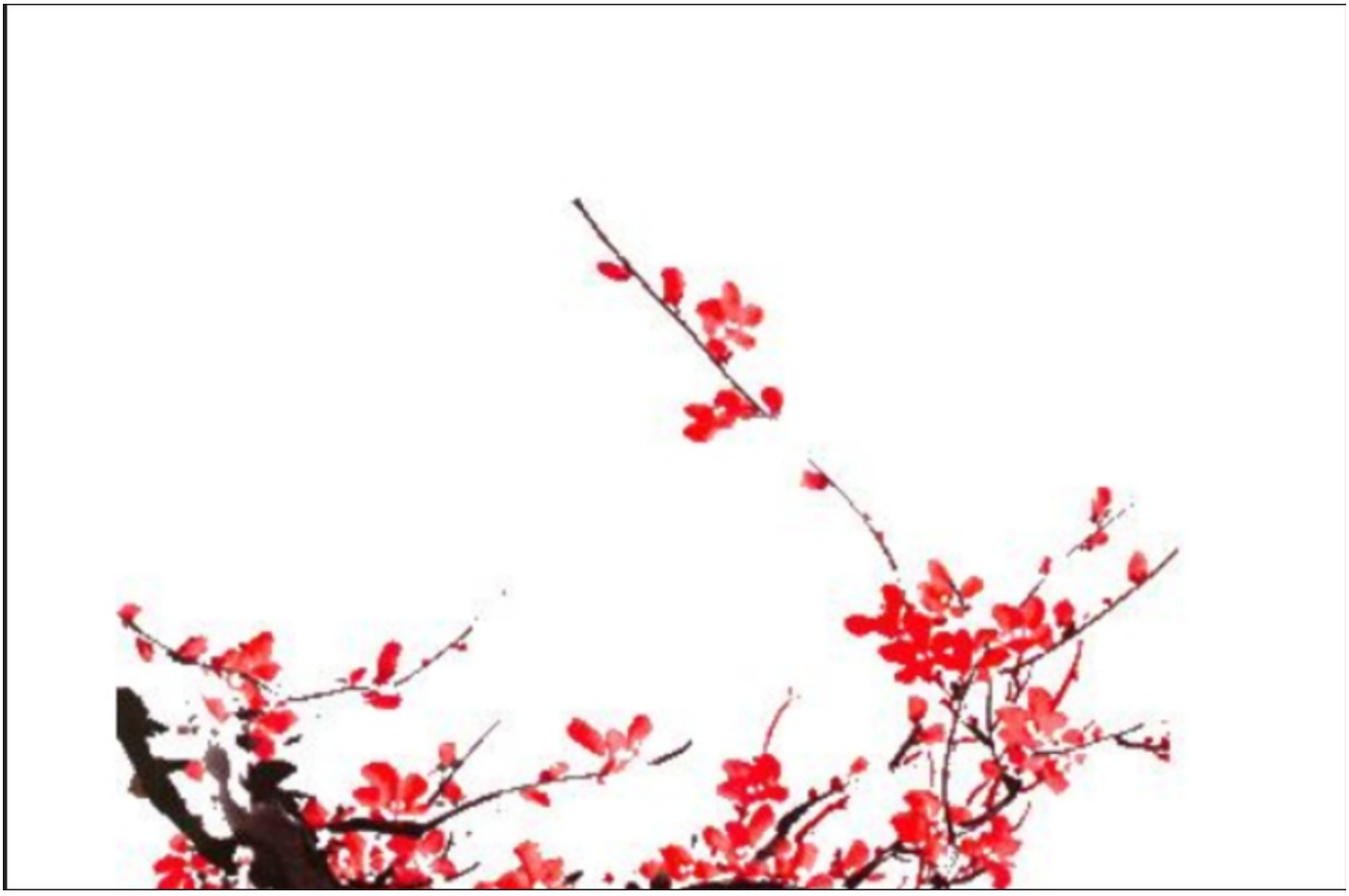
010 Editor - C:\Users\YiJiale\Pictures\站点logo\猫.png

文件(F) 编辑(E) 搜索(S) 视图(V) 格式(O) 脚本(I) 模板(L) 调试(D) 工具(T) 窗口(W) 帮助(H)

起始页 huahua.png 猫.png ×

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	‰PNG........IHDR
0010h:	00	00	00	12	00	00	00	C8	08	06	00	00	00	D2	89	BC	...ò...È.....Ò‰¼
0020h:	6F	00	00	20	00	49	44	41	54	78	5E	ED	5D	07	58	53	o.. .IDATx^í].XS
0030h:	D7	17	FF	DD	84	8D	03	08	4E	50	10	82	03	09	6E	AD	×.ÿÝ„...NP.‚..n-
0040h:	DA	D6	EE	3A	3A	ED	DF	2E	BB	B7	1D	D6	EE	A9	5D	B6	ÚÖî::íß.»·.Öî©]¶
0050h:	D5	D6	EE	6D	F7	D0	5A	AB	76	D7	3D	3A	5C	75	13	C4	ÕÖîm÷ÐZ«v×=:\u.Ä
0060h:	41	40	44	C4	01	01	04	95	95	E4	FE	BF	F3	22	0A	E4	A@DÄ...••äþ¿ó".ä
0070h:	25	79	2F	79	2F	04	CC	F9	BE	7C	0F	CD	B9	F7	9E	73	%y/y/.Ìù¾|.Í¹÷žs
0080h:	DE	FB	E5	DE	77	EF	19	0C	2A	91	21	39	76	00	34	B6	ÞûåÞwï..*'!9v.4¶
0090h:	BB	C0	D9	70	00	49	00	F6	81	F3	4D	1C	9A	4D	60	D8	»ÀÙp.I.ö.óM.šM`Ø

填上文件头后打开得到正常的图片



修改高度得到flag

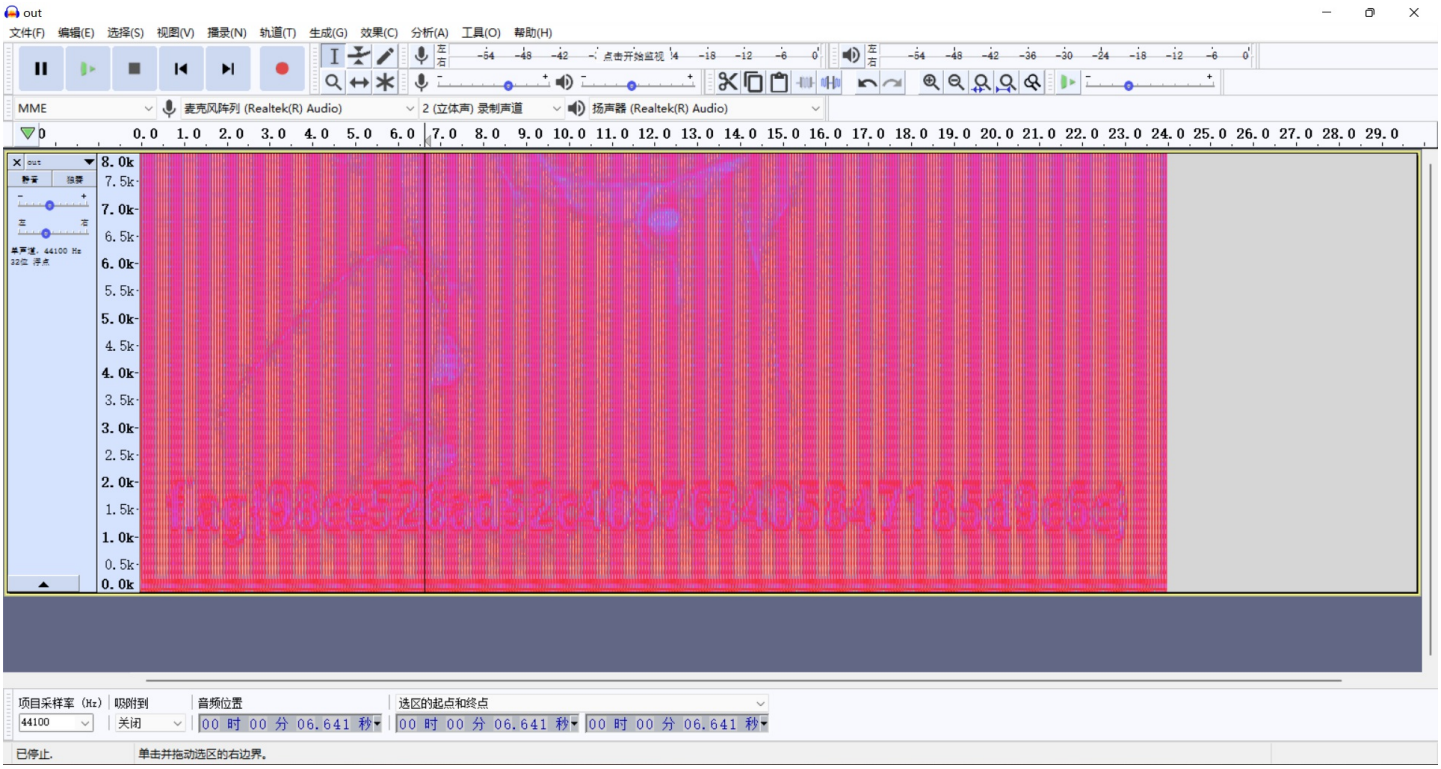
所以flag为：

```
flag{b3afc91a8fbb6cc798bdebb253b02550}
```

Misc4:NOISE

下载附件，里面有3个文件，其中out文件，丢入010分析发现是wav音频文件

将out文件加上`.wav`后缀，然后用Audacity打开，分析频谱图，得到flag

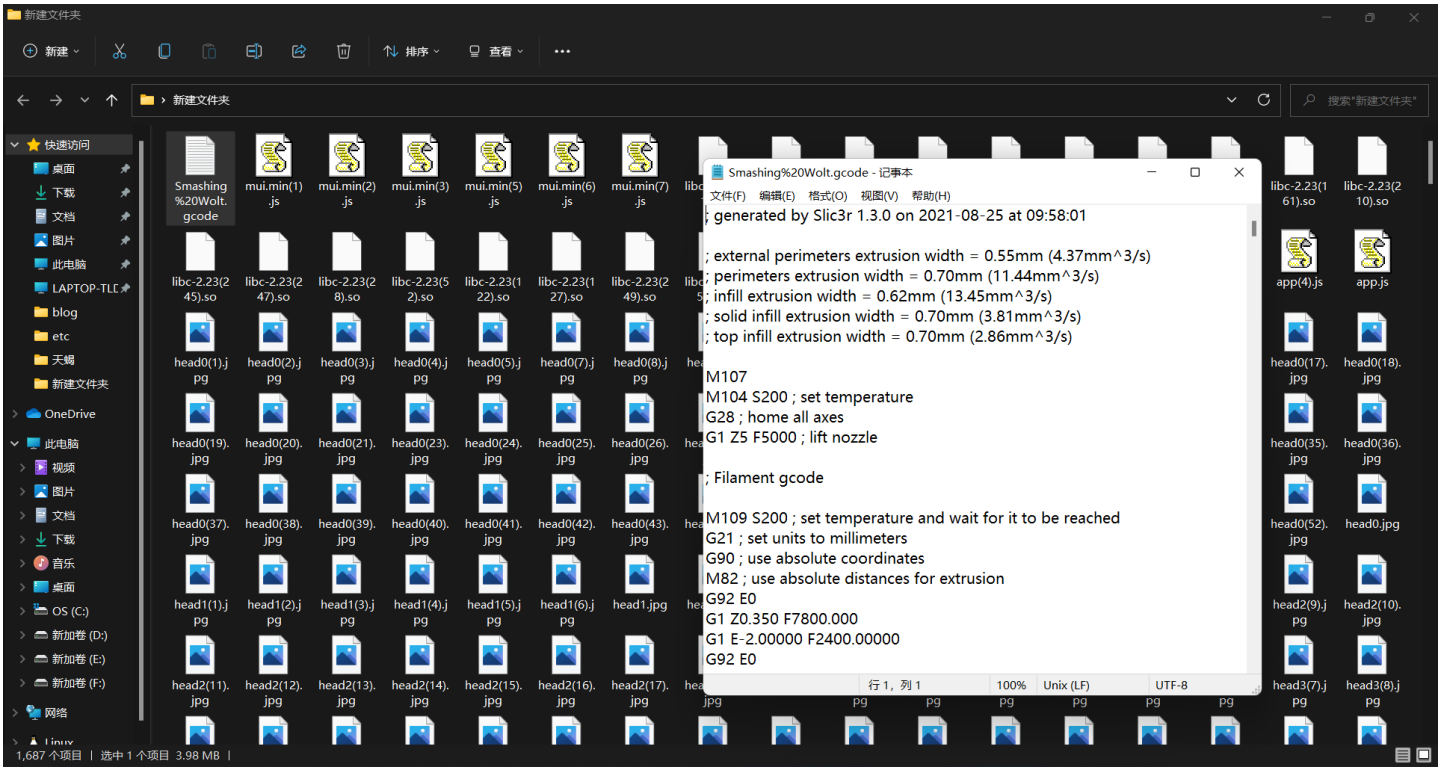


所以flag为：

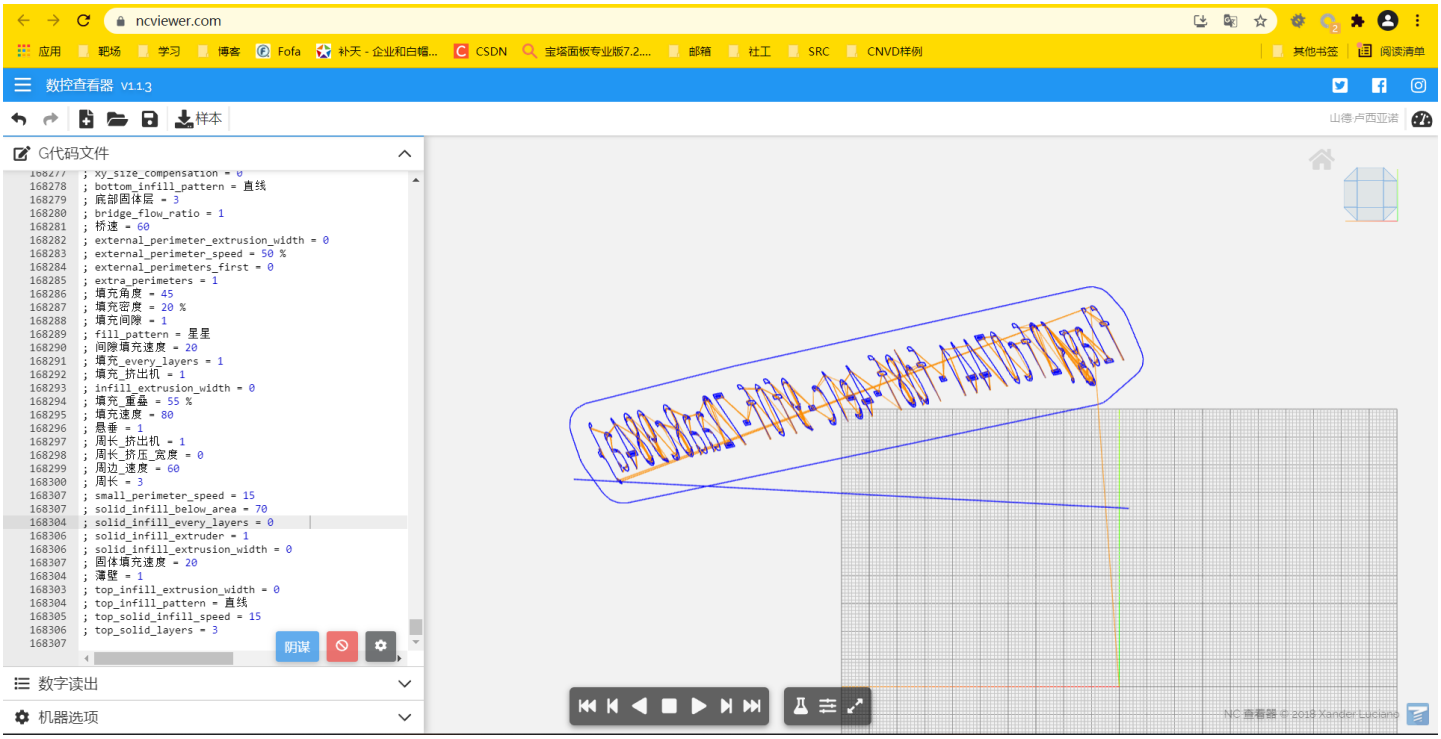
```
flag{98ce526ad52c409763405847185d9c6c}
```

Misc5:DdDdDd

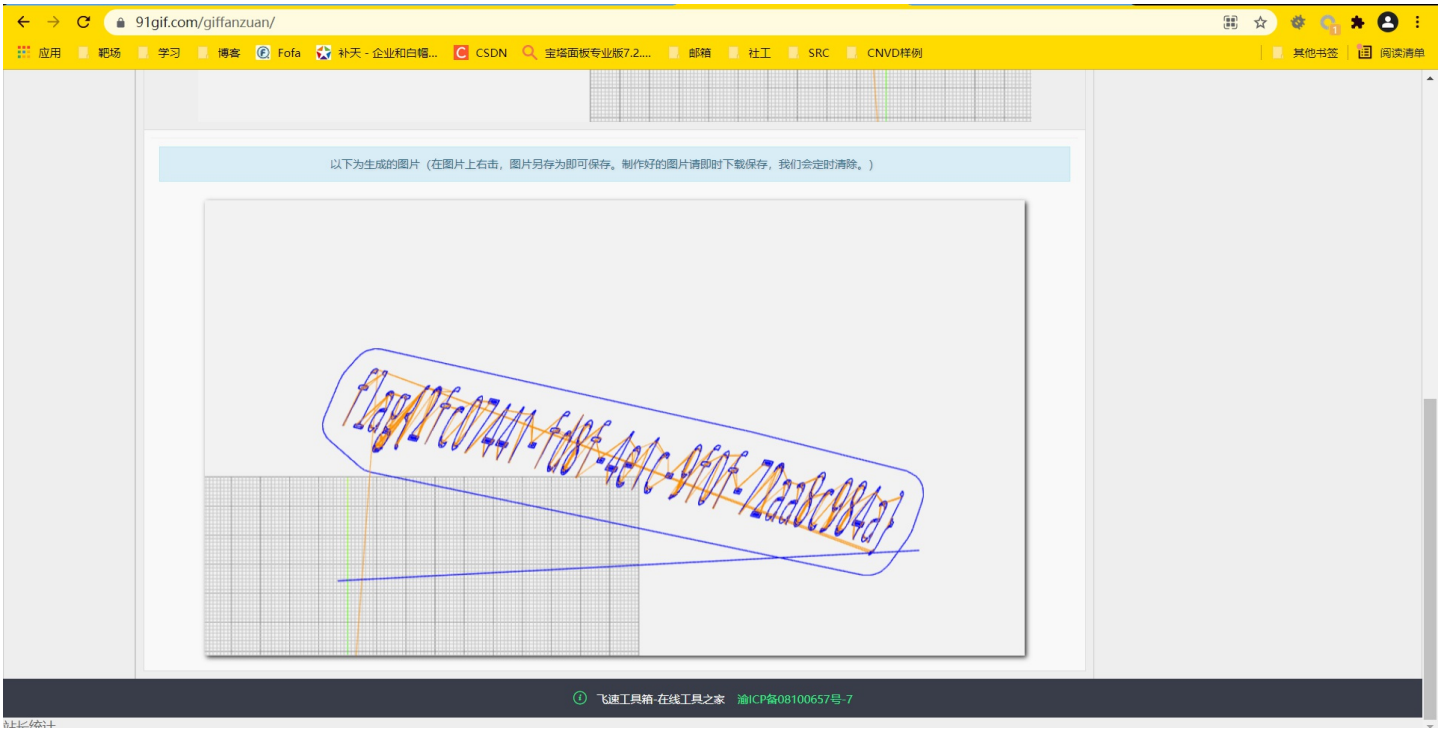
打开流量包，导出http流量，发现`Smashing%20Wolt.gcode`是G语言



使用https://ncviewer.com/网站在线编译



右上角可以调角度，得到翻转的flag，截图下来，用在线工具https://www.91gif.com/giffanzuan/ 可以镜像翻转过来



得到flag为：

```
flag{2fc07441-fd8f-4e1c-9f0f-72aa8c984a}
```

Misc6:Forensic

下载附件，用vol.exe对内存分析，直接查找带有flag字样的文件

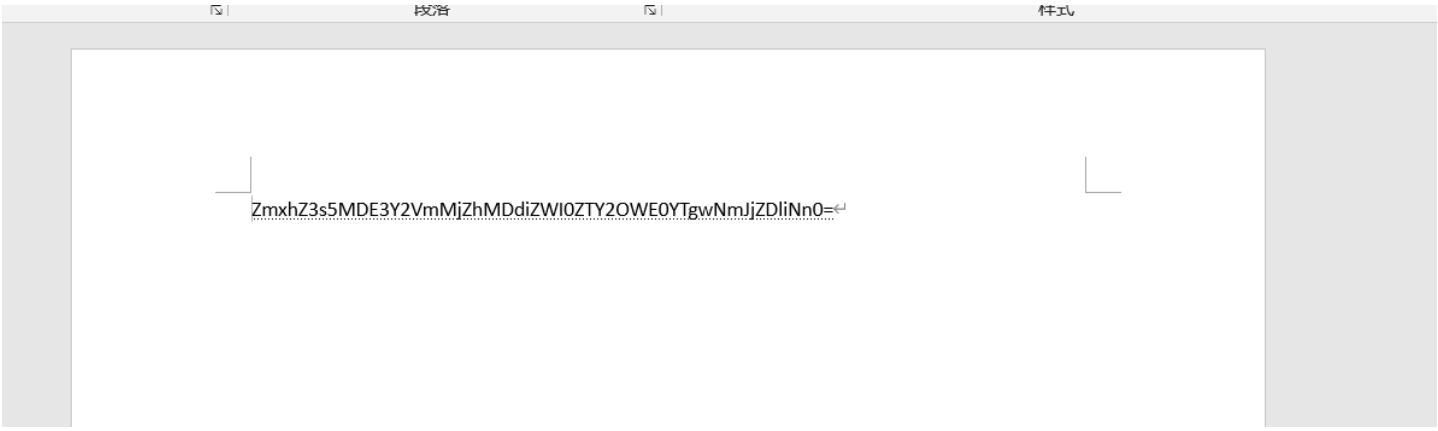
命令：

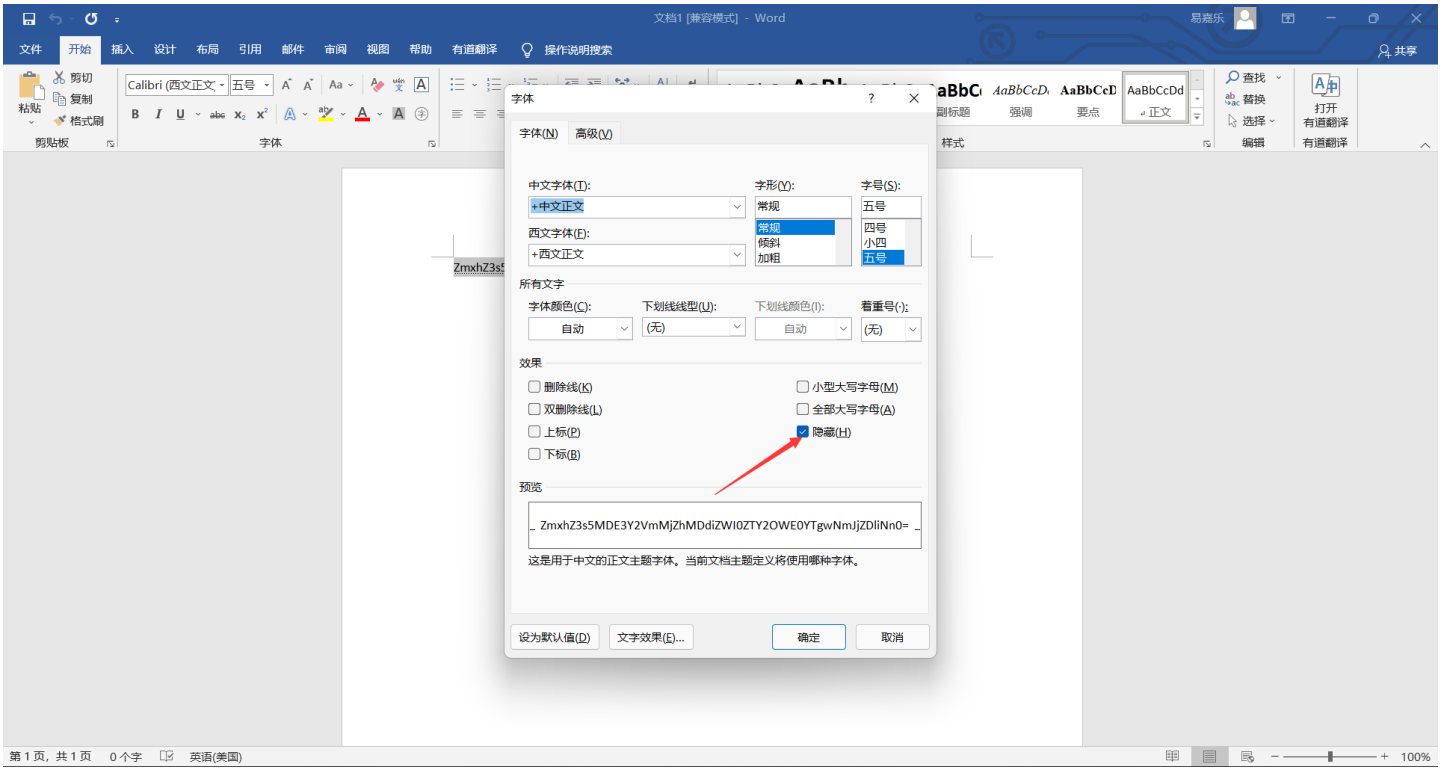
```
vol.exe -f data.raw --profile=Win7SP1x64 filescan |findstr /r "flag"
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.22000.258]
(c) Microsoft Corporation。保留所有权利。

F:\CTF2\CTF工具箱\数字取证\volatility-master\volatility-master\绿盟杯>vol.exe -f data.raw --profile=Win7SP1x64 filescan
|findstr /r "flag"
Volatility Foundation Volatility Framework 2.6
0x000000007d09b610      2      0 RW-rw- \Device\HarddiskVolume2\Users\sun\AppData\Roaming\Microsoft\Windows\Recent\flag.
docx.lnk
0x000000007d1a0d10     16      0 RW---- \Device\HarddiskVolume2\Users\sun\Desktop\flag.docx
0x000000007efbff20     15      0 R--rwd \Device\HarddiskVolume2\Users\sun\Desktop\flag.zip
0x000000007f3cb430     16      0 RW---- \Device\HarddiskVolume2\Users\sun\Desktop\flag.docx
0x000000007f8a57b0      2      0 RW-rw- \Device\HarddiskVolume2\Users\sun\AppData\Roaming\Microsoft\Windows\Recent\flag.
txt.lnk
0x000000007f9ea070      2      0 RW-rw- \Device\HarddiskVolume2\Users\sun\AppData\Roaming\Microsoft\Windows\Recent\flag.
png.lnk
0x000000007fde2f20     16      0 RW-r-- \Device\HarddiskVolume2\Users\sun\Desktop\flag.txt

F:\CTF2\CTF工具箱\数字取证\volatility-master\volatility-master\绿盟杯>
```

```
F:\CTF2\CTF工具箱\数字取证\volatility-master\volatility-master\绿盟杯>vol.exe -f data.raw --profile=Win7SP1x64 dumpfiles
 -Q 0x000000007d1a0d10 -D F:\
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x7d1a0d10   None   \Device\HarddiskVolume2\Users\sun\Desktop\flag.docx

F:\CTF2\CTF工具箱\数字取证\volatility-master\volatility-master\绿盟杯>vol.exe -f data.raw --profile=Win7SP1x64 dumpfiles
 -Q 0x000000007f3cb430 -D F:\
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x7f3cb430   None   \Device\HarddiskVolume2\Users\sun\Desktop\flag.docx

F:\CTF2\CTF工具箱\数字取证\volatility-master\volatility-master\绿盟杯>
```

一共有两个flag.docx，个flag.zip，其中一个flag.zip导不出来，一个flag.docx里面是base64，但是隐藏了的



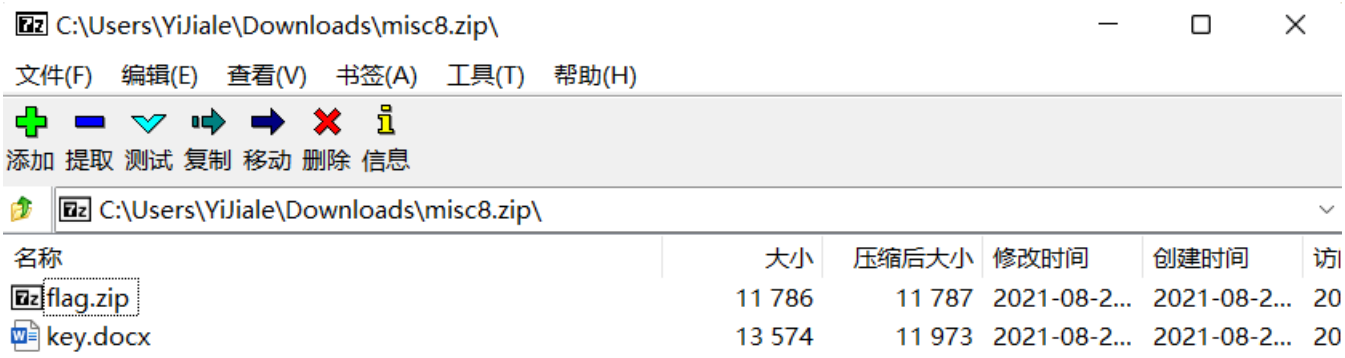


取消隐藏，解密得到flag：

```
flag{9017cef26a07beb4e669a4a806bcd9b6}
```

Misc7:隐藏的数据

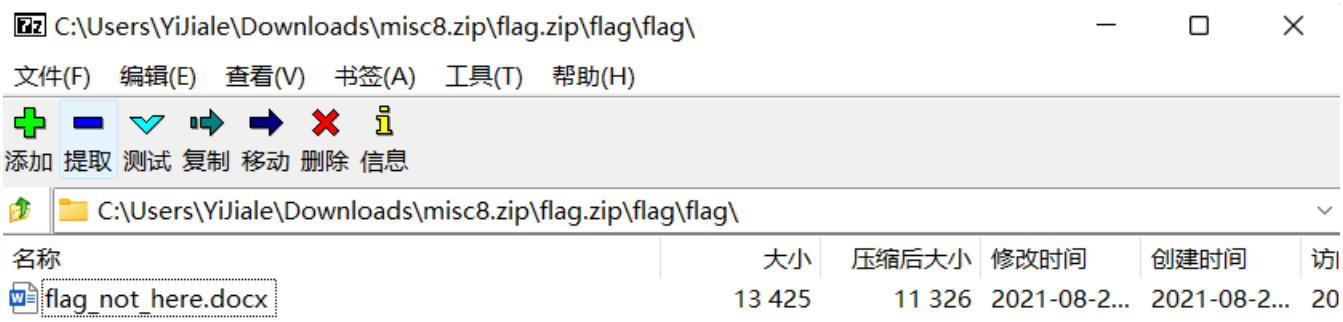
下载附件，得到一个flag.zip和一个word文档



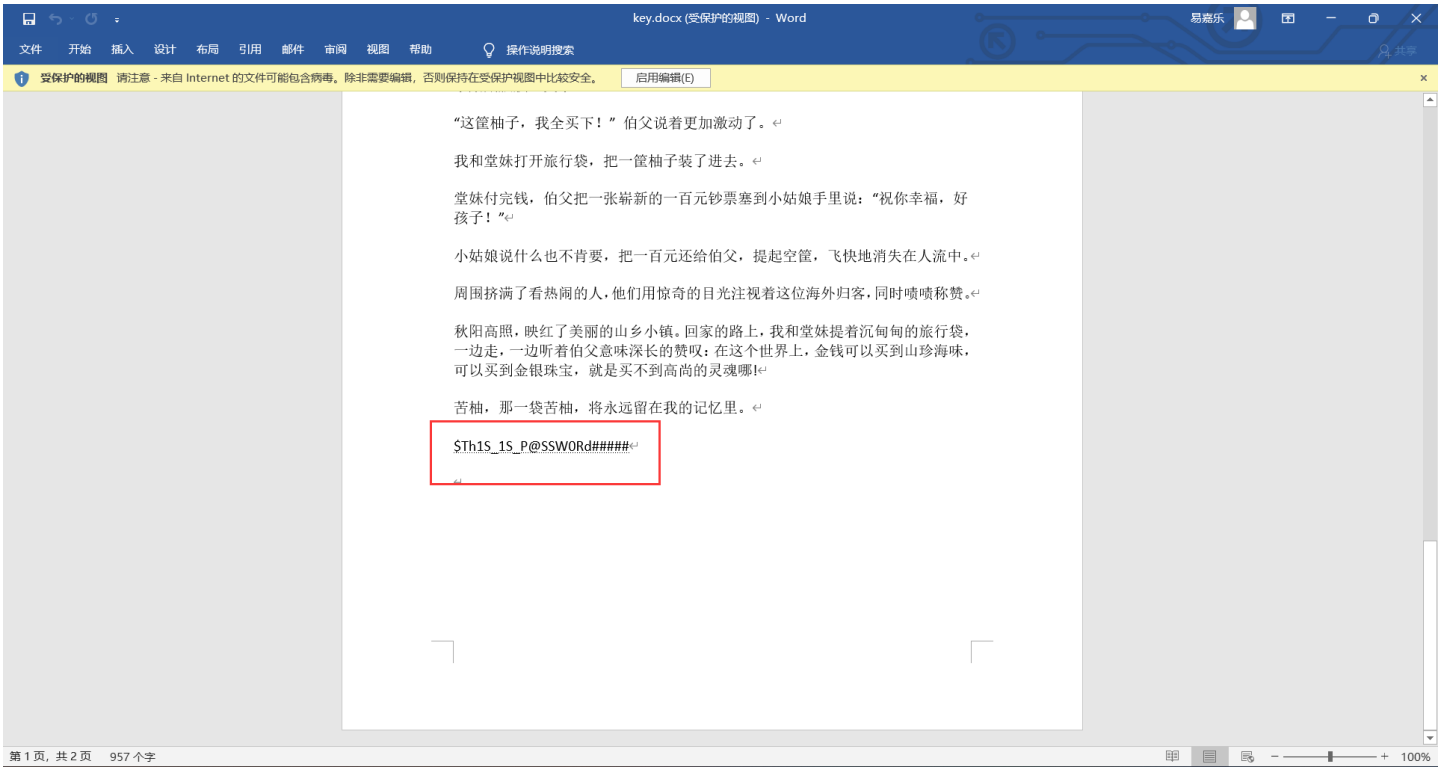
打开flag.zip，发现里面是flag文件，丢入010分析发现是伪加密的压缩包，但是7z可以无视伪加密，所以直接打开，里面又是一个flag文件

再次打开发现需要密码，爆破一下，得到密码为0546

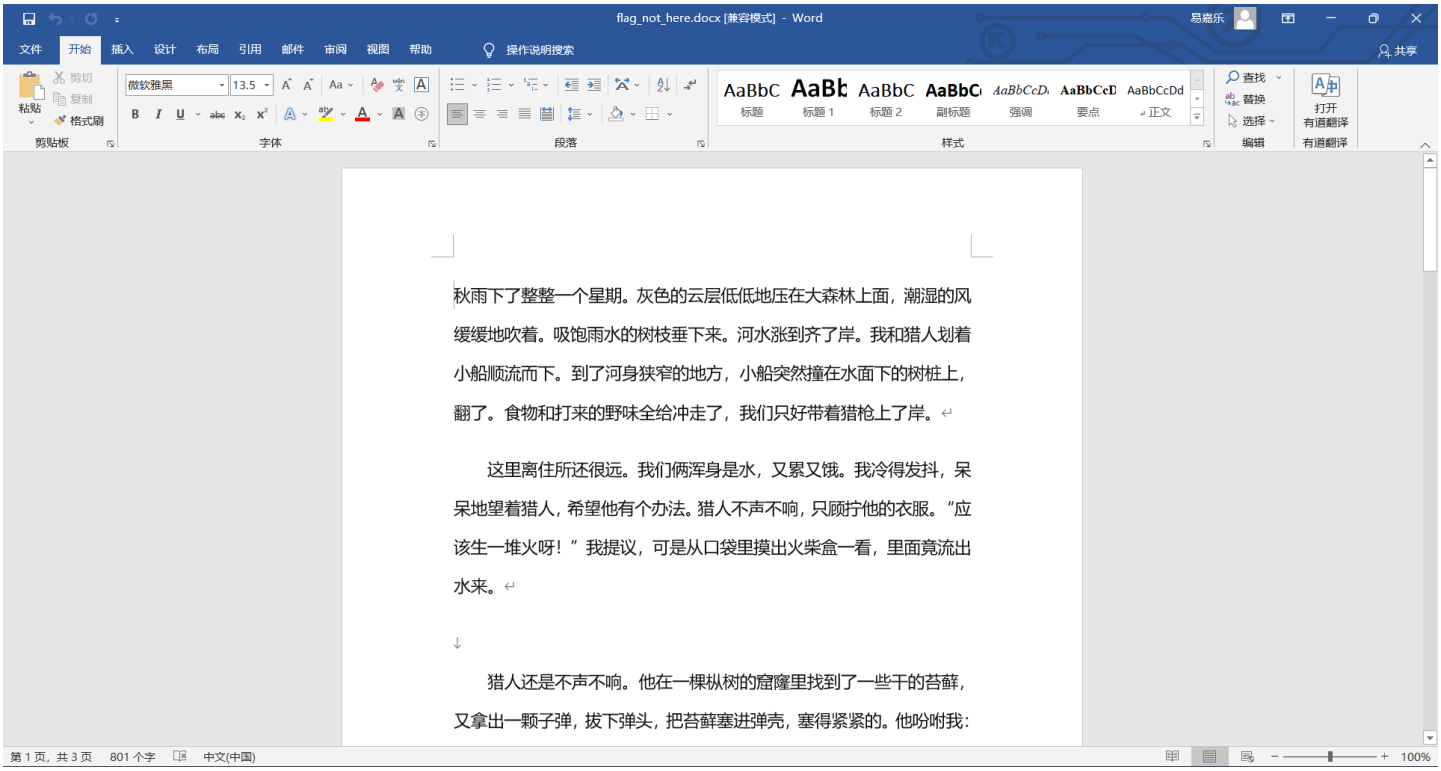
打开压缩包得到`flag_not_here.docx`，尝试打开，发现需要密码，爆破不出来，于是想到一开始的word文档还没打开过



打开里面有一个密码`$Th1S_1S_P@SSW0Rd#####`

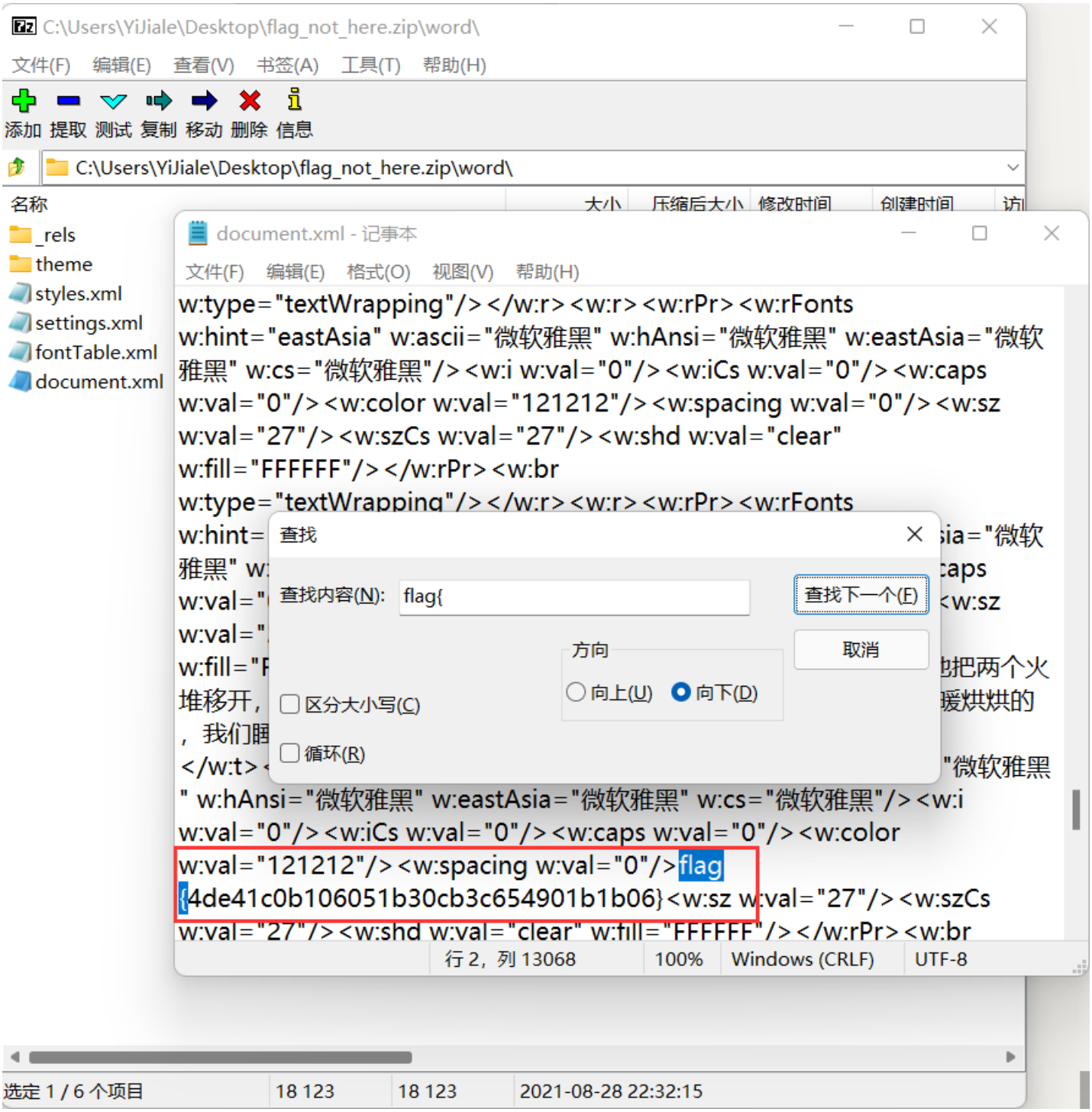


用这个密码可以打开刚刚那个`flag_not_here.docx`



里面什么东西都没有发现，于是把`.docx`改为`.zip`然后去打开压缩包 查看word文档的结构


C:\Users\YiJiale\Desktop\flag_not_here.zip\
文件(F) 编辑(E) 查看(V) 书签(A) 工具(T) 帮助(H)
添加 提取 测试 复制 移动 删除 信息
C:\Users\YiJiale\Desktop\flag_not_here.zip\
名称 大小 压缩后大小 修改时间 创建时间 访
_rels 737 247 2012-07-0... 2021-08-2... 20
word 57 302 8 265 2012-07-0... 2021-08-2... 20
docProps 1 770 946 2012-07-0... 2021-08-2... 20
customXml 881 565 2012-07-0... 2021-08-2... 20
[Content_Types].xml 1 432 350 2012-07-0... 2021-08-2... 20



在`document.xml`中找到了flag：

```
flag{4de41c0b106051b30cb3c654901b1b06}
```

Misc8:something in picture

这题是第五届强网杯Threebody原题，贴一个原题wp链接吧 https://www.sohu.com/a/472787619_121118996

flag：

```
flag{D1mEn5i0nAl_Pr061em}
```