NUC_CTF-writeup

原创

S1lenc3 于 2019-10-26 20:57:00 发布 332 收藏

逆向

jungle

PEID 查壳，没有壳，32位。IDA打开大致看了看，C++写的，静态太难了，直接OD动态调试，

检测长度 大于0x1E

格式 flag{xx-xxxx-xxxx-xxxx}

第一个是一个md5解密，可以直接搜到md5字符串，猜测应该是字符串拼接，还能搜到两个base64编码，解密后拼接，第二个检测，不知道怎么弄，4个字符爆破呗。。。

upx

题目upx，那肯定是upx壳啊，手脱，用工具都可以。这道题没什么难度，最难得应该就是脱壳了吧。

Address	Length	Type	String
.rdata:0…	00000006	C	wrong
.rdata:0…	00000007	C	windbg
.rdata:0…	00000008	C	x32_dbg
.rdata:0…	00000008	C	x64_dbg
.rdata:0…	00000006	C	ida64
.rdata:0…	00000009	C	indebug\n
.rdata:0…	00000005	C	good

查找关键字，交叉引用。

来到主函数，使用R键将数字转换为字符，可以找出三个字符串，然后异或求解。

```
sub_41104B("please input string
v77 = 0;
v53 = 'a';
v54 = 'z';
v55 = 'x';
v56 = 'x';
v57 = 'c';
v58 = 113;
v59 = 97;
v60 = 98;
v61 = 82;
v62 = 87;
v63 = 53;
v64 = 113;
v65 = 98;
v66 = 51;
v67 = 108;
v68 = 108;
v69 = 90;
v70 = 50;
v71 = 70;
v72 = 116;
v73 = 90;
v74 = 119;
```

```
s1 = '6ljh,!;:+&p%i*a=Sc4#pt*%'
s2 = '1zsw438oOFu5i4nd0f_cH2z1'
s3 = 'azxxcqabRW5qb3llZ2FtZwgi'
flag = ''

for i in range(len(s1)):
    flag += chr(ord(s1[i]) ^ ord(s2[i]) ^ ord(s3[i]))

print(flag)
```

跑脚本就完事了。

RE2

一个游戏，64位，ELF文件，IDA载入分析，发现流程很简单。

```
{
  v4 = (__int64)v7;
  *(_QWORD *)v7 = 2528799242480796961LL;
  *(_QWORD *)(v4 + 8) = 742287683049791 4908LL;
  *(_QWORD *)(v4 + 16) = 4120863165711020332LL;
  *(_QWORD *)(v4 + 24) = 7592311550845536058LL;
  *(_DWORD *)(v4 + 32) = 170338378;
  *(_WORD *)(v4 + 36) = 86;
  puts("=====================================================");
  puts("i'm GOD of this game,i can give you some information which will help you.");
  printf("%s", option1);
  __isoc99_scanf("%d", &v5);
  while ( v5 )
  {
    if ( v5 == 1 )
    {
      printf("%s", option2);
      __isoc99_scanf("%d", &v6);
      if ( v6 )
      {
        giff_flag(v7, v6);
        print_random_tayy_response();
      }
```

一大串字符串，应该是用来进行解密的，然后有个‘flag’关键字符串的函数，进去看看。

```
__int64 __fastcall giff_flag(__int64 a1, int a2)
{
  signed int i; // [rsp+18h] [rbp-4h]

  for ( i = 0; i <= 36; ++i )
  {
    if ( num2 % 2 )
    {
      if ( num2 % 2 == 1 )
        *(_BYTE *)(i + a1) -= i * a2 % 37;
    }
    else
    {
      *(_BYTE *)(i + a1) += i * a2 % 37;
    }
  }
  return (unsigned int)(num2++ + 1);
}
```

动态调试。num是从0开始的，每次选择第一个选项的时候就会+1.没什么其他难点了，动态调试有反调试，很简单，直接jmp就行。因为只能输入7次，所以得爆破九百万个数，牛批。

上爆破脚本：

```
for w in range(1111111, 9999999):
    flag = ['0x21', '0x45', '0x58', '0x4c', '0x83', '0x19', '0x18', '0x23', '0x1c', '0x40', '0x4e', '0x35',
'0x26',
            '0x5b',
            '0x3', '0x67', '0x2c', '0x71', '0x32', '0x48', '0x37', '0x3f', '0x30', '0x39', '0x3a', '0x47',
'0x3e',
            '0x34',
            '0x21', '0x4f', '0x5d', '0x69', '0x4a', '0x28', '0x27', '0xa', '0x56']
    s1 = str(w)
    n = 0
    result = ''
    for j in range(7):
        r = 0
        if n % 2 == 1:
            for i in range(37):
                flag[i] = hex(int(flag[i], 16) - ((i * int(s1[j])) % 37))
        else:
            for i in range(37):
                flag[i] = hex(int(flag[i], 16) + ((i * int(s1[j])) % 37))
        for z in range(len(flag)):
            if 33 > int(flag[z], 16) or int(flag[z], 16) > 126:
                r = 1
        if r == 0:
            for p in flag:
                s2 = chr(int(p, 16))
                result += s2
            print(result)
        n += 1
```

crypto

base

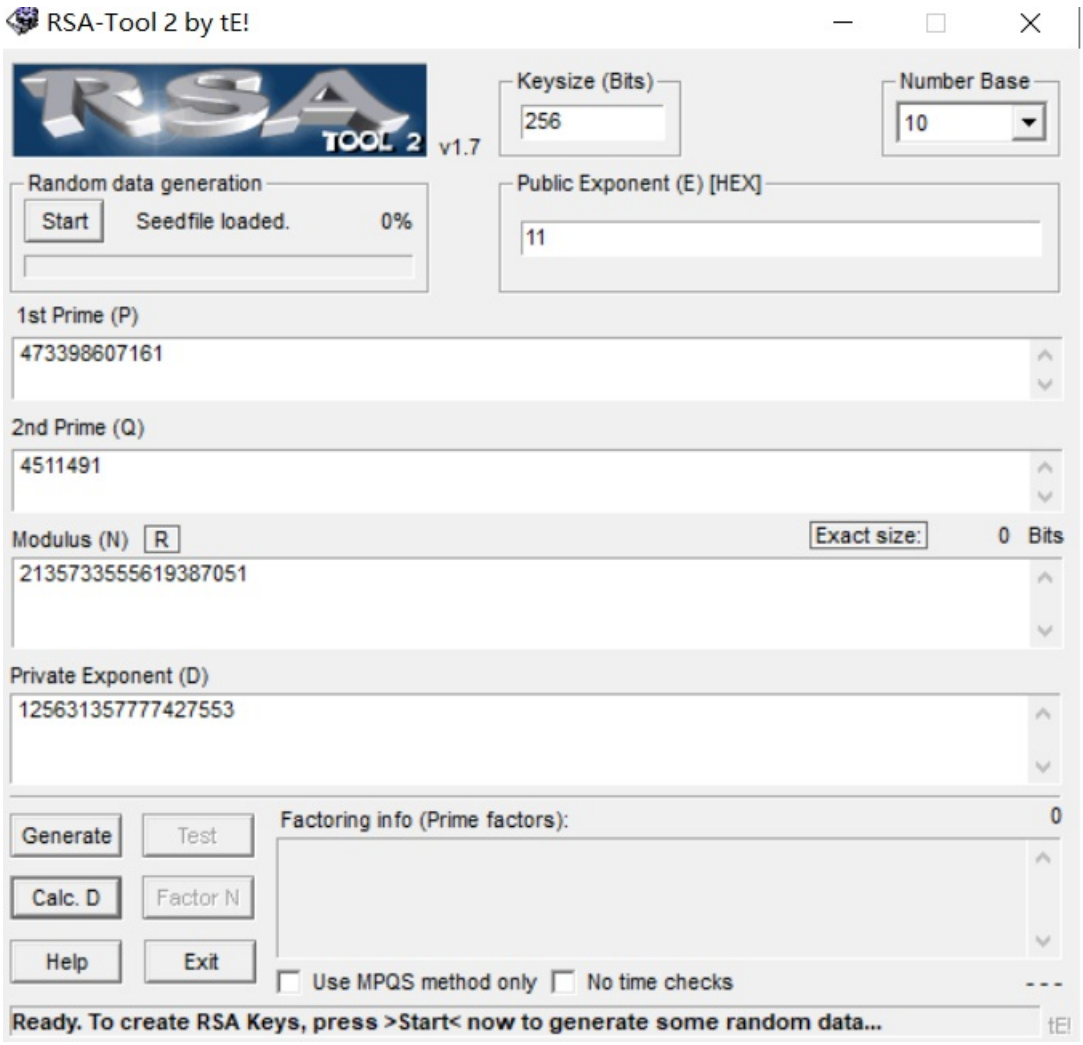
没提示还不知道咋做。base全家桶，直接上py2脚本：

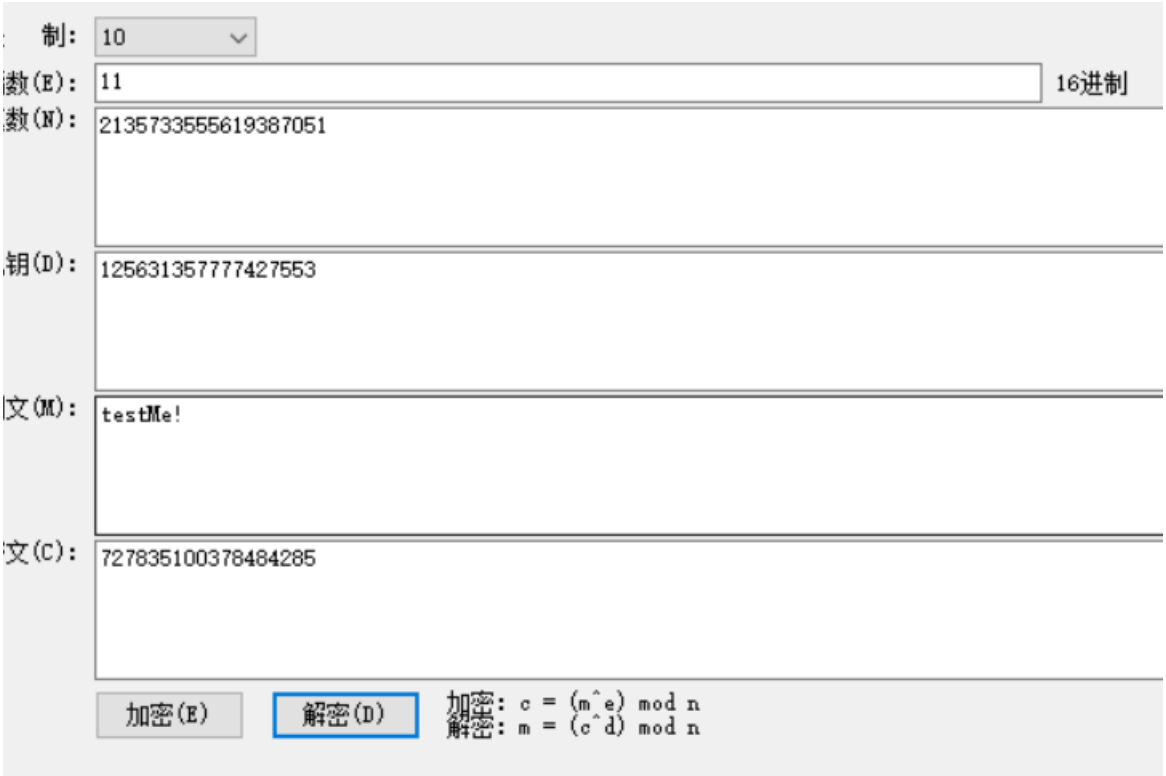
import base92, base58, base64

a = '4%_,,I,*xt];Y@(6Hk]jrF.2:gR_Ss&-=S<EiI^TnIW%U+
(XJXk_Fr.A4!Y)o\'[AIT%:U3\Z55lUmVJPlP<%&Rb2Pujy+rE)<,SLEg*os4]9lQXa;-
w#SWbC5MiY=;Da5KuI&V[S:auU?Ub&4gT$lY6=PRhxBng,<,H-A@^v$:QHTg1;@au)]]B'
b = base92.decode(a)
c = base58.b58decode(b)
d = base64.b64decode(c)
e = base64.b32decode(d)
f = base64.b16decode(e)
print f

easy_rsa

所有信息都很明显，根据p,q,e算出D，然后解密密文。


RSA-Tool 2 by tE!
RSA TOOL 2
v1.7
Keysize (Bits)
256
Number Base
10
Random data generation
Start
Seedfile loaded.
0%
Public Exponent (E) [HEX]
11
1st Prime (P)
473398607161
2nd Prime (Q)
4511491
Modulus (N)
R
Exact size:
0 Bits
2135733555619387051
Private Exponent (D)
125631357777427553
Generate
Test
Calc. D
Factor N
Help
Exit
Factoring info (Prime factors):
0
Use MPQS method only
No time checks
Ready. To create RSA Keys, press >Start< now to generate some random data...
tE!


制:
10
数(E):
11
16进制
数(N):
2135733555619387051
钥(D):
125631357777427553
文(M):
testMe!
文(C):
727835100378484285
加密(E)
解密(D)
加密: c = (m^e) mod n
解密: m = (c^d) mod n

simple_crypto

提示quip，有个解密网站的域名好像是这个名字，词频分析吧，试试，

```
0    -2.011   chain is no stronger than its weakest link a clean hand wants no washing and the key you want is crypt o o o o
1    -2.268   uhain is no stronger than its weakest minkau mean hand wants no washing and the kell of want i surly to oo o
2    -2.312   chain is no stronger than its weakest link a clean hand wants no washing and the keuu of want is crum too oo
3    -2.618   mhain is no styonkey than its wearest lin ram lean hand wants now a shink and the rebb of want is my but oooo
```

那肯定是cryptoooo啊。

超级凯撒

很脑洞，我要是上课听课了，肯定也做不出来，哈哈。

ascii码 每次相差 0x7e 递减2， 说不清楚， 看脚本吧，简单易懂。

```
s =
'e7f09ae1e994d9dfddd08b88baccc780c4c8bbbf76bdc58a6eb2b6a9adbf7875776f6e6968986a62605f5e605c58835983534b4e7c4
77642733f6c386b367f'
s1 = []
for i in range(0, len(s), 2):
    a = '0x' + s[i:i+2]
    s1.append(a)
s2 = ''
d = 0x7e
for i in range(len(s1)):
    c = chr(int(s1[i], 16) - d)
    s2 += c
    d -= 2
print(s2)
```

it is good! The flag is: flag{6593412d82234864a9e716f3d2e3b0e2}

web

pastejacking

右键粘贴复制。JS代码好像是不让复制什么的，不是很懂，好像快捷键不行吧，我审计JS代码快捷键粘贴也出错了，手残没复制对？

ez_cmd

一堆过滤，慢慢过waf吧，最后是%0a截断，然后 cd ../../../ 忘了在哪个目录了，最后 cat ../../../flag.txt！

web很渣，我不知道我学了一年web，却还是这么渣。。。。

MISC

wireshark

流量分析，在一堆ICMP报文中有很多可疑的ttl 像是ascii码，提取出来是，strong+passwordstrong+password

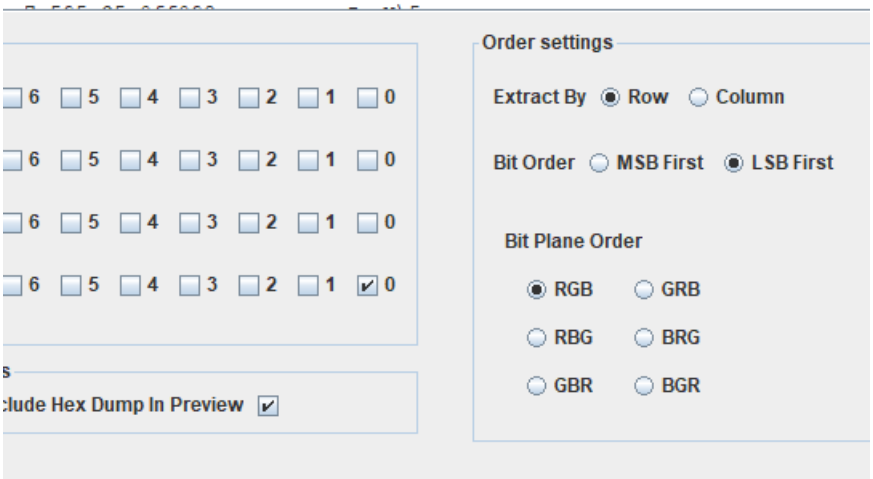
```
ICMP      74 Echo (ping) reply    id=0x0001, seq=490/59905, ttl=64 (request in 539)
ICMP      74 Echo (ping) request  id=0x0001, seq=491/60161, ttl=103 (reply in 542)
ICMP      74 Echo (ping) reply    id=0x0001, seq=491/60161, ttl=64 (request in 541)
ICMP      74 Echo (ping) request  id=0x0001, seq=492/60417, ttl=43 (reply in 544)
ICMP      74 Echo (ping) reply    id=0x0001, seq=492/60417, ttl=64 (request in 543)
ICMP      74 Echo (ping) request  id=0x0001, seq=493/60673, ttl=112 (reply in 546)
ICMP      74 Echo (ping) reply    id=0x0001, seq=493/60673, ttl=64 (request in 545)
ICMP      74 Echo (ping) request  id=0x0001, seq=494/60929, ttl=97 (reply in 548)
ICMP      74 Echo (ping) reply    id=0x0001, seq=494/60929, ttl=64 (request in 547)
ICMP      74 Echo (ping) request  id=0x0001, seq=495/61185, ttl=115 (reply in 550)
ICMP      74 Echo (ping) reply    id=0x0001, seq=495/61185, ttl=64 (request in 549)
ICMP      74 Echo (ping) request  id=0x0001, seq=496/61441, ttl=115 (reply in 552)
```

以为是flag，交上去不对，又翻了翻HTTP协议，发现有个压缩包，把文件dump出来，需要解压密码，就是上边的字符串。

```
HTTP      440 GET /nuc.zip HTTP/1.1
HTTP     1473 HTTP/1.1 200 OK  (application/zip)
HTTP      441 GET /flag.php HTTP/1.1
```

解压出来一张图片，这么多关卡，啊，杂项呗，慢慢搞，最后发现是lsb隐写。

```
4 6572657374696e67  flag{int eresting
3 747572657de3e8ad  _lsb_pic ture}...
f b3d07dadab989c3f  >....Z.. ..}....?
d 8bc72074ecflclcl  .....C.m .. t....
b f80640f077bd6ad2  !.K\3J.; ..@.w.j.
9 0el2a2d2ccc567dc  wKB.p.,. ......g.
c 09d26dbae0e780af  L....... ..m.....
5 3d91c70b3a8f4283  .5....Q. =...:.B.
a a85901219776430c  .2OcT_.j .Y.!.vC.
5 9774549fbd6ee201  p@=....E .tT..n..
```



找吧

很多{} 猜测词频分析，一波现成脚本打上去。

```
# -*- coding:utf-8 -*-
import operator

str = '''
flM{Sg_i_igl1S_ll__SfM_FF_1ilfM{Sa11gagc1lSSMgfnafg_fMa1n5iaF_c1lSFiSaf_1f{S_l_FalS5_faSl_fgl5M1_{ll
{i5c}if1__fg5{__M{ngU{1l1gff1f1iS__Mf5iFMlciSgaU{glgUF5M_1aa_f_i5{nfllla1S1FS!cSg{fUfFcS1{{ag1lU51act
{ggi{gglg{{flnliF{M5faF1ig_agal{_{{aMMilfUSa{a5ggiiigfSSg{M_Mng{a}fcMf1_Fl{cM{1fiflMSSM{_l!Scf5FFcn{g{S
{iaala_M5l1Mc11afcgfgl5f1g_c{llaUMf1lM1aF{af1Sl5lf5l1l5a_cc_c_1ff}f_ff}MlU{afM_1fcla{{gM{_Sl_M_{gM_{g5{
{g{Ffl{Ffac!a{afffgl!_gMalF_c{lac_MFMg5acMFcla5cMlU5aSff{l_UFf_Ug1!g1F_c{{aMMg{SlgUa1ca1ff5_c1g5{fli
{lc{i{1iglM_fUgl___a5fnMaFf{_lfll_igf1lcalniMag_5nFS1MMaiM1ll5SlMiaf_5l{af__MMgac_Mf__fUa1fc{1{_55SF!
'''

payloads = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!0}{123456789_\n'
#payloads = payloads.upper()
# print payloads
dists = {}
for x in payloads:
dists[x] = 0
# print x,dists[x]
for s in str:
dists[s] += 1

ans = ''
res = sorted(dists.iteritems(), key=operator.itemgetter(1), reverse=True)
for r in res:
ans += r[0]
print r
print ans
```

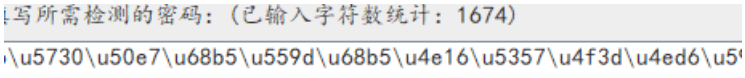
```
('x', 0)
('z', 0)
fla_g{M1Sci5FUn!}
03247698ACBEDGIHKJLONQPRTWVYXZbedhkjmoqpsrutwvyxz
jan@Silent:/mnt/e/CTF/个人CTFTools/编码与密码/密码/词频分析$
```

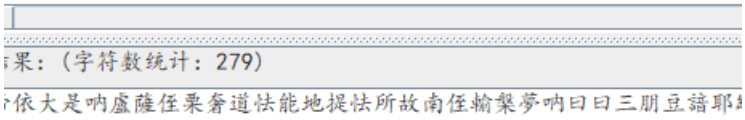
跑出来是这个，哦，不对，看着也出来了，应该脚本的问题吧，自己慢慢改对了，凭经验， 应该是misc_is_fun

文体两开花

提示 \u 和中国话， 那就先 unicode试试，先分离，

```
s =
'5e1d4f9d5927662f545076e785a94f8468175962905360 2f80fd573063d0602f6240654553574f848f3869c35922545066f066f04e0
9670b8c468af380367f3d91af601b8af35e1d8f384f845492602f963f601b68b566f076a44ea67f3d53c368b5537354c66f2b5962905
35937624068b56f2b5450522951a57a76602f6ec5771f54504e16545096407f3d67095c3c54c6803676e781f34f5b602f5a4676a4608
967094f8459227f8569c350e77adf96c68af3795e59625927545059ea6d85540968b5745f7b49601b5373963f5a468b395a46720d964
08c467adf608954c685dd5ea6602f5f9768177f3d300254c64e0954c68ae6820d9060803654c65937822c820d57304f9d77e551a5906
0601676a46c9968b59ebc76a47adf4e16602f5fc383e9964054c676e75bc65beb59624ee54ff1985b54c6591c602f822c77e5602f6ce
24ff17919964096e296e24e00602f592283e9602f771f5b558af38a368af38af882e6602f59ea4f8484996089660e8dcb9ebc5f4c66f
0545085dd84994ff15fc37f3d80366ce24ea676a467095962985b8af382e554c6720d4ff15beb596260895962662f4f8476e154c685d
d76a47b4982e5545090606bbf7f70771f905369c35450795e68b56c995beb5357771f596290fd6578745f54c64ea67f3d9ebc68b55a4
685d08b394ff180057f3d6b7b9060820d5e1d802868b5535796e2559d76e74ed6573050e768b5559d68b54e1653574f3d4ed659624e1
65bc6602f860753737f704e00670b54c68b39'
f = ''
for i in range(0, len(s), 4):
    f += "\\u" + s[i:i + 4]
print(f.encode('GBK').decode('GBK'))
```





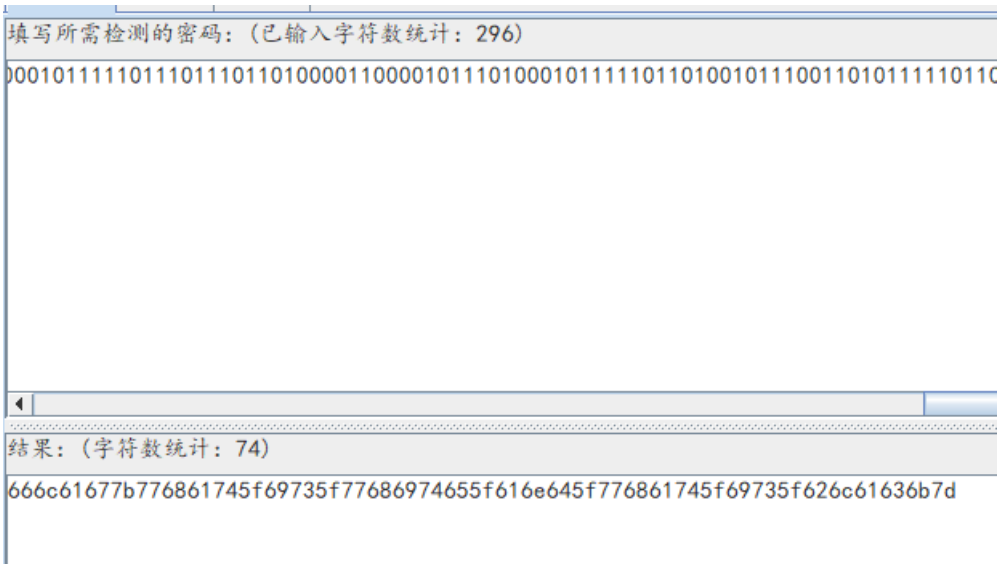
然后解密：

中国话， 与佛论禅吧，试试。。。

黑白

gif动图，只有黑的和白的，肯定是01二进制转ascii码啊，直接python脚本分离，然后去像素转换0和1

```
from PIL import Image
from PIL import ImageSequence
frames = []
img = Image.open('./hei.gif')
for frame in ImageSequence.Iterator(img):
   f = frame.copy().convert("RGB")
   r1, g1, b1 = f.getpixel((111,52))
   if r1 == 0 and g1 == 0 and b1 == 0:
      print('0',end='')
   else:
      print('1',end='')
```



16进制转换文本 / 文本转16进制

666c61677b776861745f69735f77686974655f616e645f776861745f69735f626c61636b7d

字符串转16进制

16进制转字符串

结果互换

flag{what_is_white_and_what_is_black}

赞助

赞助

人生

分不高，一个png，难度应该不大，最简单的来吧，改高度试试。

读作key，写作flag{26cdb94b0962486abe071d7fecd412d5}

洋葱

洋葱，一层一层的剥开，哦。。。

foremost 分割， 找到一个 4 文件，不知道是个啥，后来有提示。pkt ，百度pkt怎么打开， 下载思科的软件，打开直接复制flag。

flag{86a3c1ce5c7c45ceb0babc747cf84e91}