HGAME 2021 Week1 - 容熙

MISC

Base全家福

题目

`R1k0RE1OWldHRTNFSU5SVkc1QkRLTlpXR1VaVENOUlRHTVlETVJCV0dVMlVNTlpVR01ZREtSUlVIQTJET01aVUdSQ0RHTVpWSVlaVEVNWlFHTVpER01KWElRPT09PT09`

经过Base64解码获得如下结果：

`GY4DMNZWGE3EINRVG5BDKNZWGUZTCNRTGMYDMRBWGU2UMNZUGMYDKRRUHA2DOMZUGRCDGMZVIYZTEMZQGMZDGMJXIQ======`

经过观察，疑似Base32，解码得到：

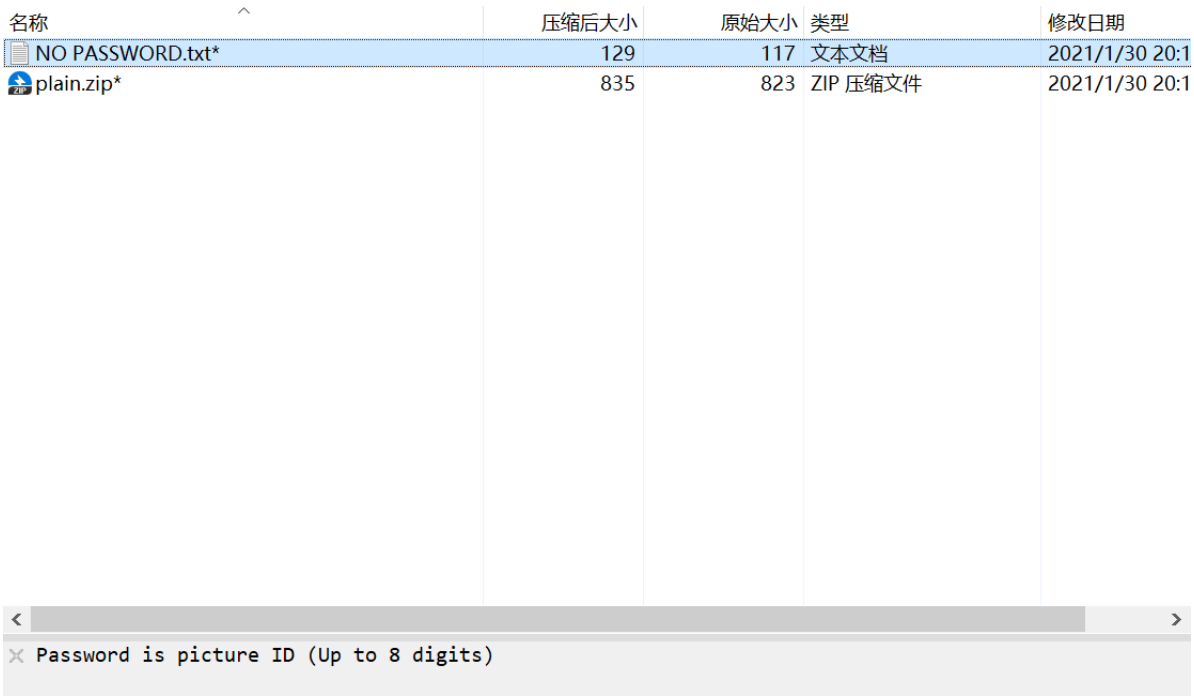
`6867616D657B57653163306D655F74305F4847344D335F323032317D`

特征疑似Base16，继续解码。

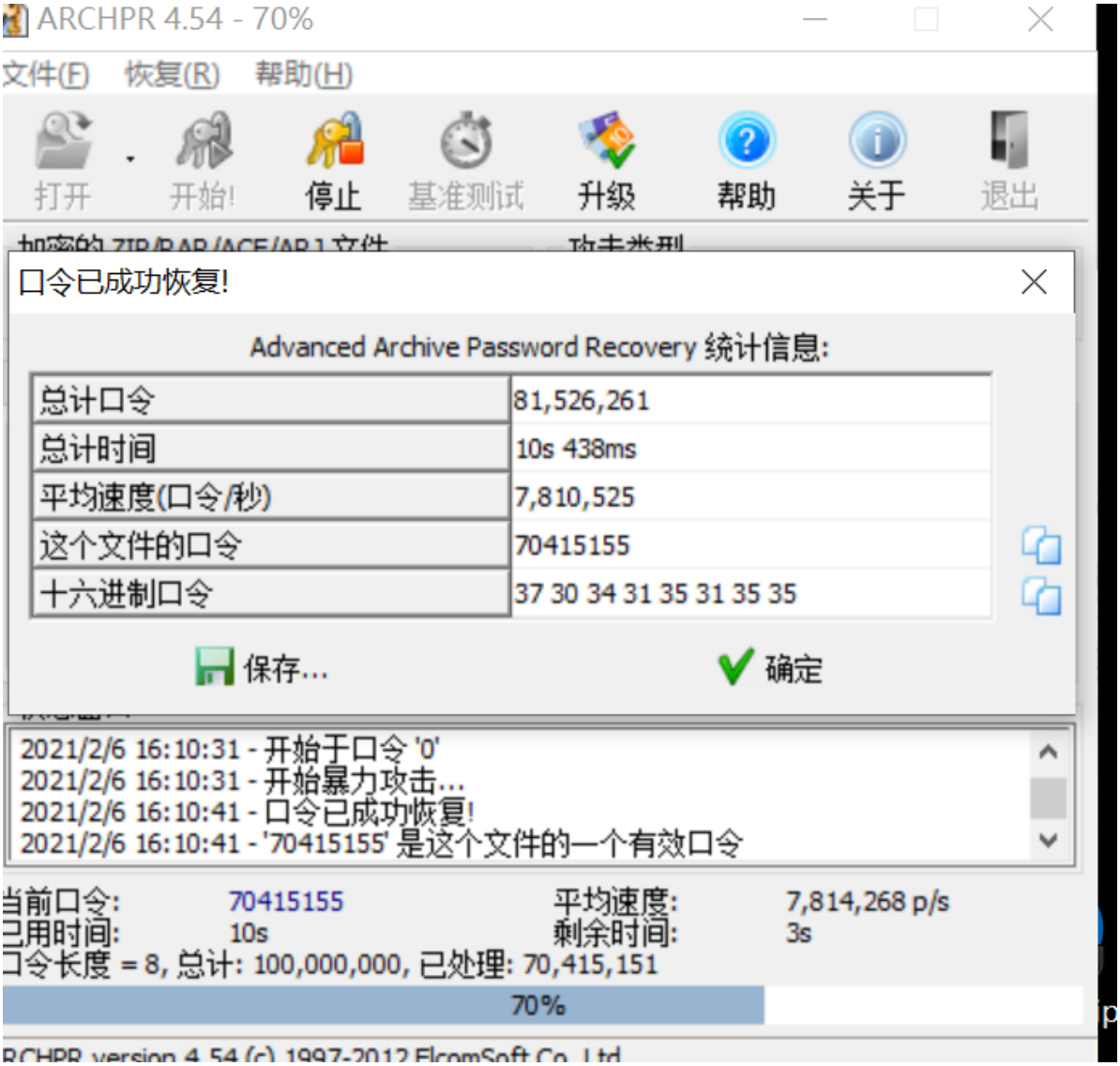
`hgame{We1cOme_tO_HG4M3_2021}`

不起眼压缩包的养成的方法

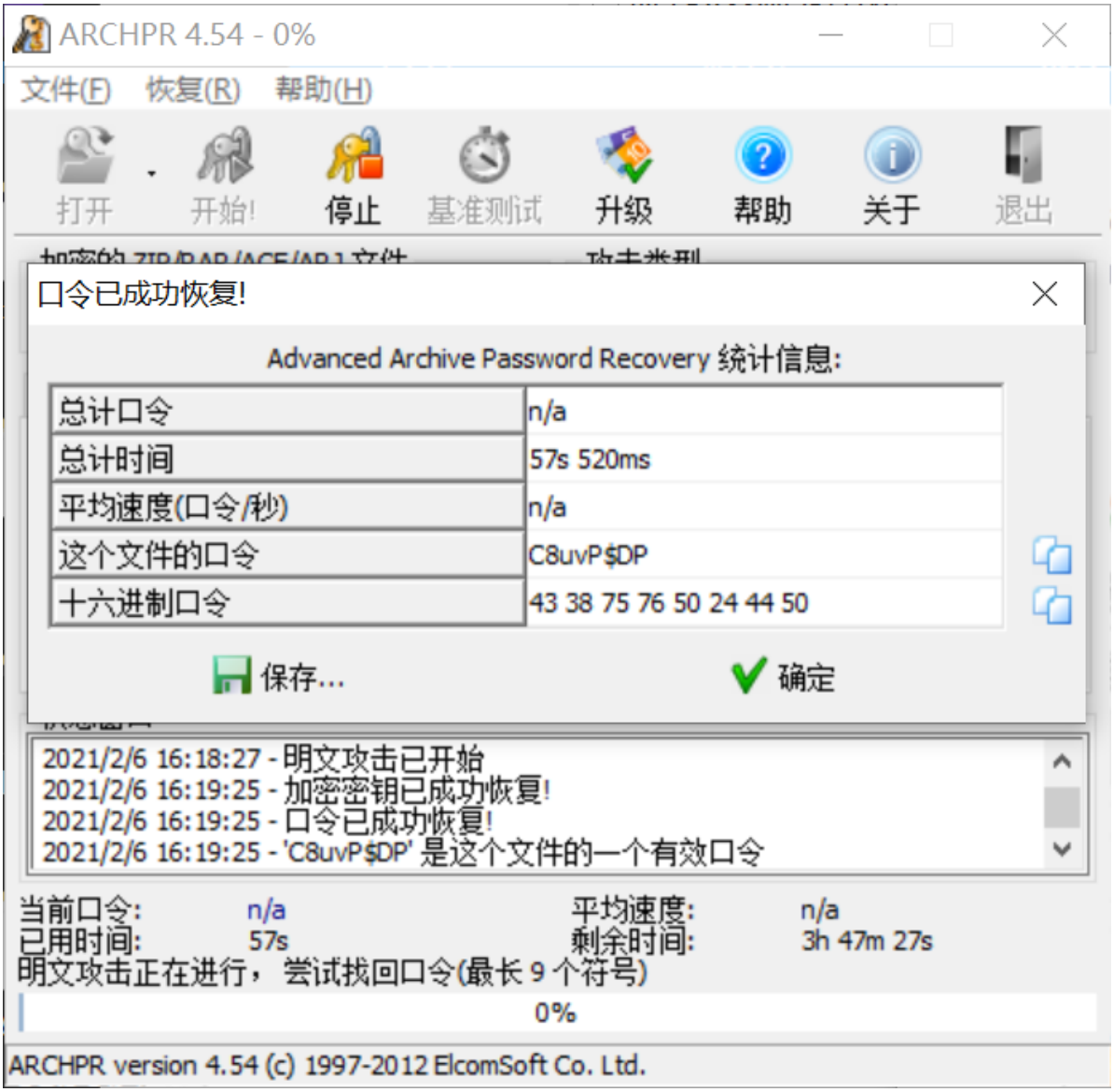
使用binwalk后如上图。get压缩包。



信息如上图。根据`8digit`提示暴力破解。



发现解压后依然为压缩包，且也有密码。发现两个nopassword的大小一致.明文攻击~



疑似伪加密。处理后进行HTML解码，得到flag。

粘贴你想在这里HTML解码的文本：

hgame{2IP_is_Usefu1_and_Me9umi_i5_W0r1d}

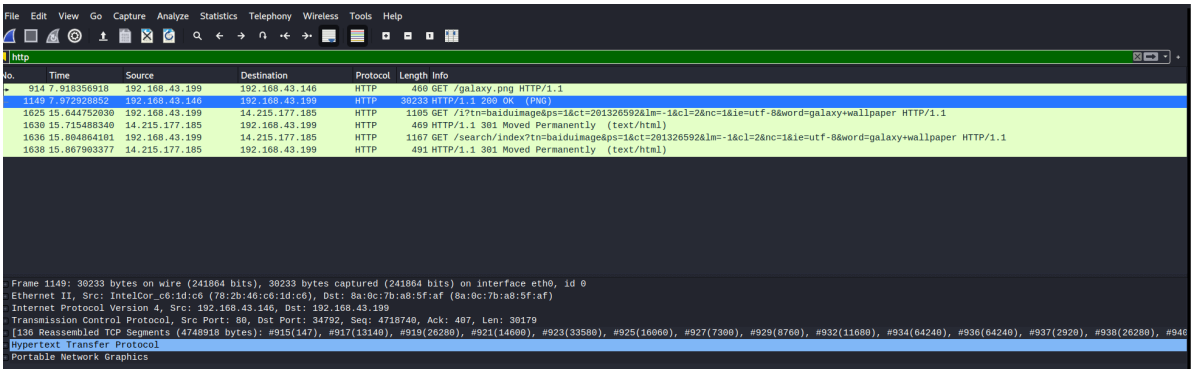
HTML解码!

复制你的HTML这里解码的文本：

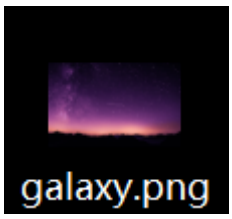
hgame{2IP_is_Usefu1_and_Me9umi_i5_W0r1d}

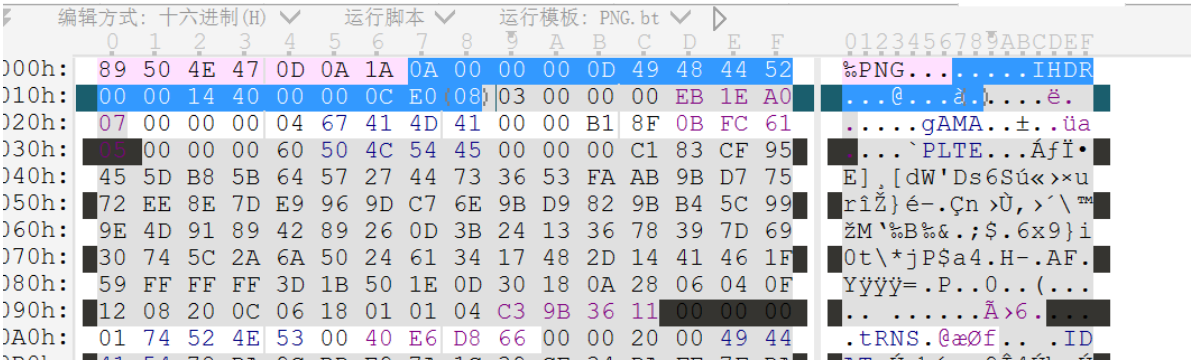
Galaxy

打开wireshark，筛选后捕捉到galaxy.png。从kali导出。

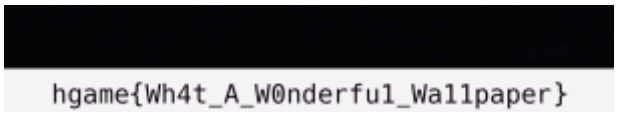


获得Akira的信物√





然后修改高度，获得flag。



~~手打对近视至残的朋友很不友好~~

Word RE:MASTER

~~想象力强大的选手以为是某种高级密码并艰难解密若干hour~~

此处隐藏了fuckbrain的提示。

Fuck！我的脑子好疼！这可能是音游瘾发作最严重的一次,躺在床上很想打交互，嘴里念叨：O-oooooooooo AAAAE-A-A-I-A-U JO-oooooooooooo AAE-O-A-A-U-U-A E-eee-ee-eee AAAAE-A-E-I-E-A JO-ooo-oo-oo-oo EEEEO-A-AAA-AAAA,不行我得在brainpower 耗尽前把密码记下来。↵

发现2021的文件√~~（连夜出题兢兢业业，膜）~~

名称	压缩后大小	原始大小	类型	修改日期
..				
_rels				
media				
theme				
document.xml	1,892	10,103	XML 文档	1980/1/1 0:00:0
fontTable.xml	595	2,415	XML 文档	1980/1/1 0:00:0
password.xml	163	284	XML 文档	2021/1/29 20:5
settings.xml	1,209	3,353	XML 文档	1980/1/1 0:00:0
styles.xml	2,925	29,326	XML 文档	1980/1/1 0:00:0
webSettings.xml	553	4,282	XML 文档	1980/1/1 0:00:0

Brainfuck

Brainfuck

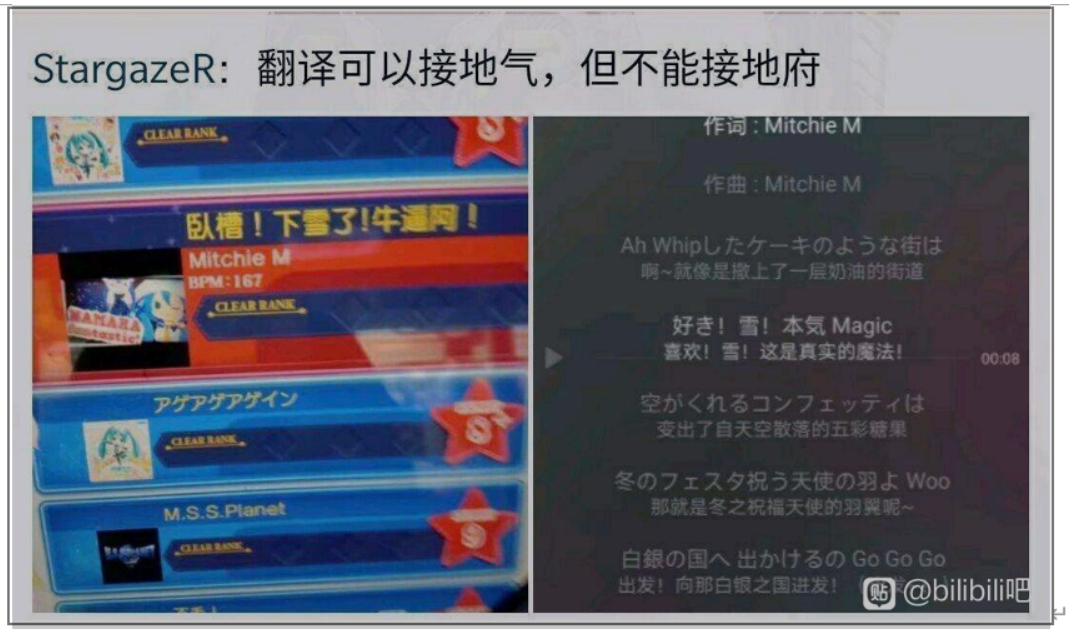
```
+++++ +++[- >++++ ++++< ]>+++ +.<++ +[->+ ++<]> ++.<+ ++[-> +++<] >+.<+ ++[-> ---<] >-.++ ++++. <+++[ ->--- <]>-. +++.+ .++++ ++++. <+++[ ->--- <]>-- ----. +.--- --.. + .++++ +++++ .<+++ [->-- -<]>- ----- .<
```

字符集 utf8(unicode编码)

加 密　　解 密

```
DOYOUKNOWHIDDEN?
```

解码。



~~建议立即接通地府（手动doge）~~

~~小东西，哪里跑~~

抓到隐藏文字*若干，经过搜索，怀疑是snow加密。

http://www.darkside.com.au/snow/index.html中获取工具，

```
. \snow. exe -C . \snow. txt
ngame{Challen9e_Whit3_P4NDOR4_P4R4DOXXX}
```

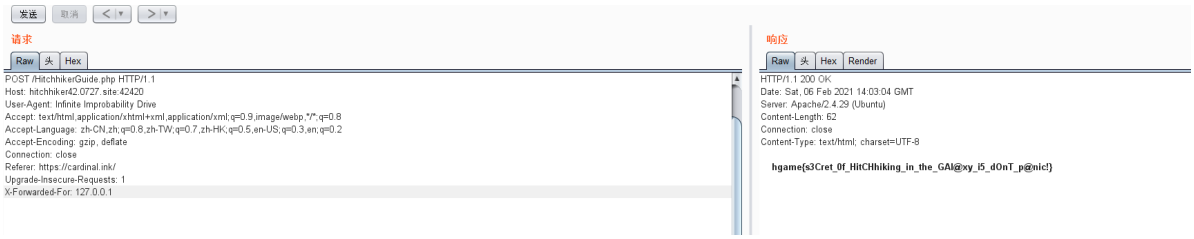
flag如图。

Web

Hitchhiking_in_the_Galaxy

浏览器Firefox代理

在burp suite里抓包，发到repeater后更改如下图。



watermelon

使用firefox打开链接，在“web开发者”——“调试器”中几个可疑的文件夹里，~~乱翻~~找到游戏结束模块处的代码。

由>1999推断下面一行为flag。

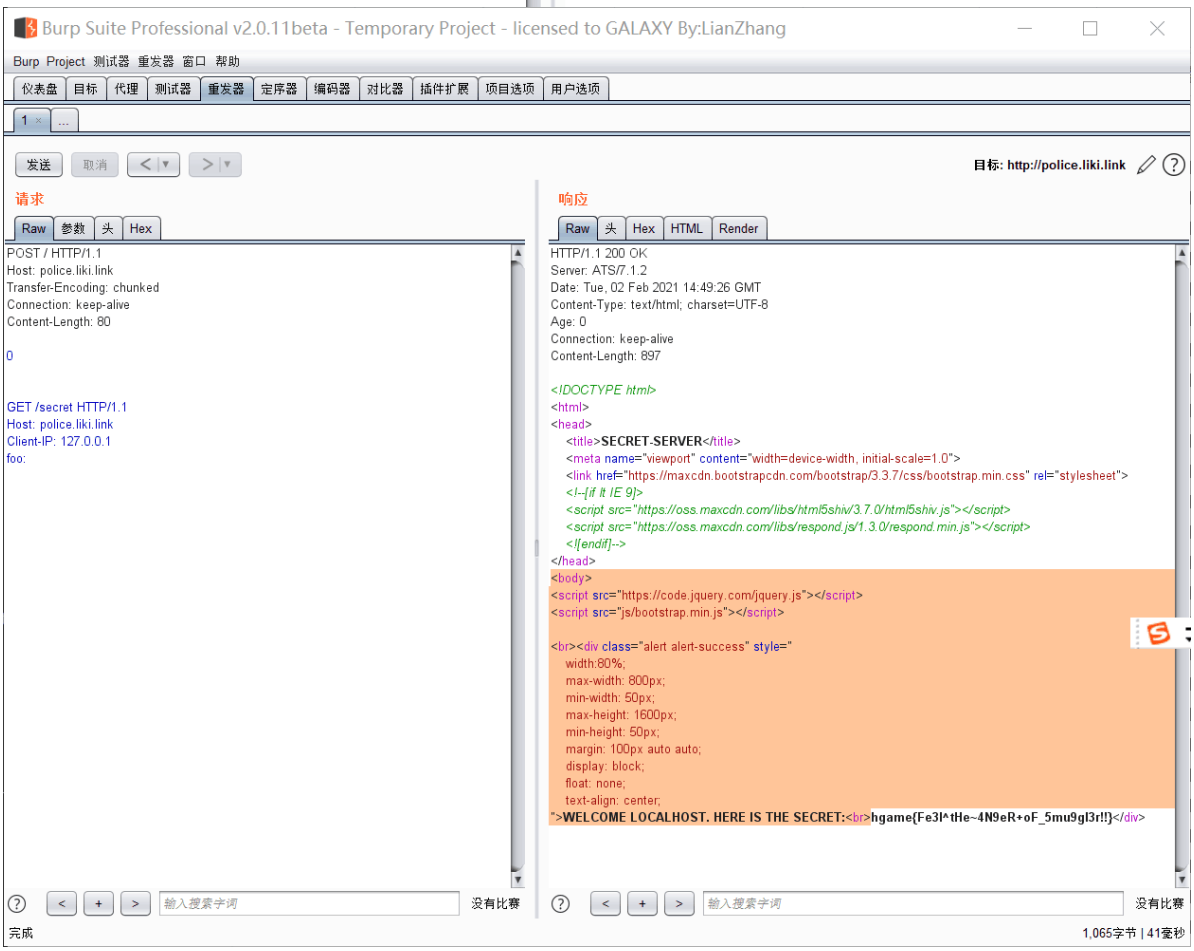
flag疑似base64，解码后得到

宝藏走私者&走私者的愤怒

参考群里关于走私的资料里的写法，在burp suite的repeater模块里进行如下操作。（由于出题人[switch前辈√]友情提示提示“get”有误，改为”post“）

```
POST / HTTP/1.1
Host: police.liki.link
Connection: keep-alive
Transfer-Encoding: chunked
Content-Length: 81

0


POST /secret HTTP/1.1
Host: police.liki.link
Client-IP: 127.0.0.1
foo:
```

```
HTTP/1.1 200 OK
Server: ATS/7.1.2
Date: Tue, 02 Feb 2021 08:20:09 GMT
Content-Type: text/html; charset=UTF-8
Age: 0
Connection: keep-alive
Content-Length: 922

<!DOCTYPE html>
<html>
<head>
  <title>SECRET-SERVER</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">
  <!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
  <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>
  <![endif]-->
</head>
<body>
<script src="https://code.jquery.com/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>

<br><div class="alert alert-danger" style="
  width:80%;
  max-width: 800px;
  min-width: 50px;
  max-height: 1600px;
  min-height: 50px;
  margin: 100px auto auto;
  display: block;
  float: none;
  text-align: center;
">ONLY LOCALHOST(127.0.0.1) CAN ACCESS THE SECRET_DATA!<br>YOUR Client-IP(115.[redacted]
IS NOT ALLOWED!</div>
```



点击若干次后发送成功。获得flag。

智商检测鸡

~~是送分题，算一百道积分就可以，我可快子~~

`（ax+b）dx` 即 `ax²/2+bx`,由于a的数值与上下限的数值都比较小，可以采用`（上限²-下限²）*（a/2)+b(上限-下限)`公式计算。

上式（a/2)、(上限-下限)两部分可口算。

Reverse

helloRe

使用IDA分析，因为没有符号表，逐个尝试找到入口函数

```
  if ( v11 != 22 )
LABEL_13:
    sub_140001480();
  v3 = v12;
  v4 = (void **)Memory;
  do
  {
    v5 = &Memory;
    if ( v3 >= 0x10 )
      v5 = v4;
    if ( (*((_BYTE *)v5 + v0) ^ (unsigned __int8)sub_140001430()) != byte_140003480[v0] )
      goto LABEL_13;
    ++v0;
  }
  while ( v0 < 22 );
  v6 = std::basic_ostream<char,std::char_traits<char>>::operator<<(std::cout, sub_140001990);
  v7 = sub_1400017C0(v6, &unk_140003470);
  std::basic_ostream<char,std::char_traits<char>>::operator<<(v7, sub_140001990);
  if ( v3 >= 0x10 )
  {
    v8 = v4;
    if ( v3 + 1 >= 0x1000 )
    {
      v4 = (void **)*(v4 - 1);
      if ( (unsigned __int64)((char *)v8 - (char *)v4 - 8) > 0x1F )
        invalid_parameter_noinfo_noreturn();
    }
    j_j_free(v4);
  }
  return 0i64;
```

发现是将输入的字符逐位和0xFF自减进行异或~（thanks to dear Mezone！）

根据 140003480 处内容

```
.rdata:0000000140003480 ; _BYTE byte_140003480[24]
.rdata:0000000140003480 byte_140003480  db 97h, 99h, 9Ch, 91h, 9Eh, 81h, 91h, 9Dh, 9Bh, 2 dup(9Ah)
.rdata:0000000140003480                                         ; DATA XREF: sub_1400014C0+A9↑o
.rdata:0000000140003480                 db 0ABh, 81h, 97h, 0AEh, 80h, 83h, 8Fh, 94h, 89h, 99h
.rdata:0000000140003480                 db 97h, 2 dup(0)
```

反着异或一次，得到flag。

pypy

python字节码尝试翻译。

```
4           0 LOAD_GLOBAL              0 (input)
            2 LOAD_CONST               1 ('give me your flag:\n')
            4 CALL_FUNCTION            1
            6 STORE_FAST               0 (raw_flag)

5           8 LOAD_GLOBAL              1 (list)
           10 LOAD_FAST                0 (raw_flag)
           12 LOAD_CONST               2 (6)
           14 LOAD_CONST               3 (-1)
           16 BUILD_SLICE              2
           18 BINARY_SUBSCR
           20 CALL_FUNCTION            1
           22 STORE_FAST               1 (cipher)

6          24 LOAD_GLOBAL              2 (len)
           26 LOAD_FAST                1 (cipher)
           28 CALL_FUNCTION            1
           30 STORE_FAST               2 (length)

8          32 LOAD_GLOBAL              3 (range)
           34 LOAD_FAST                2 (length)
           36 LOAD_CONST               4 (2)
           38 BINARY_FLOOR_DIVIDE
           40 CALL_FUNCTION            1
           42 GET_ITER
      >>   44 FOR_ITER                54 (to 100)
           46 STORE_FAST               3 (i)

9          48 LOAD_FAST                1 (cipher)
           50 LOAD_CONST               4 (2)
           52 LOAD_FAST                3 (i)
           54 BINARY_MULTIPLY
```

@善良热心的r3n0

你可以用dis转成字节码对比一下

艰难比对后~~(a few years later...)~~~~

```python
import dis

def main():
    raw_flag=input("give me your flag:\n")
    cipher=list(raw_flag[6:-1])
    length=len(cipher)

    for i in range (length//2): #100
        cipher[2*i], cipher[2*i+1] = cipher[2*i+1], cipher[2*i]


    res=[]
    for i in range (length):
        res.append(ord(cipher[i])^i)
    res=bytes(res).hex()
    print("your flag: " + res)
dis.dis(main)
```

交换上下部分并改动部分细节后如下图。

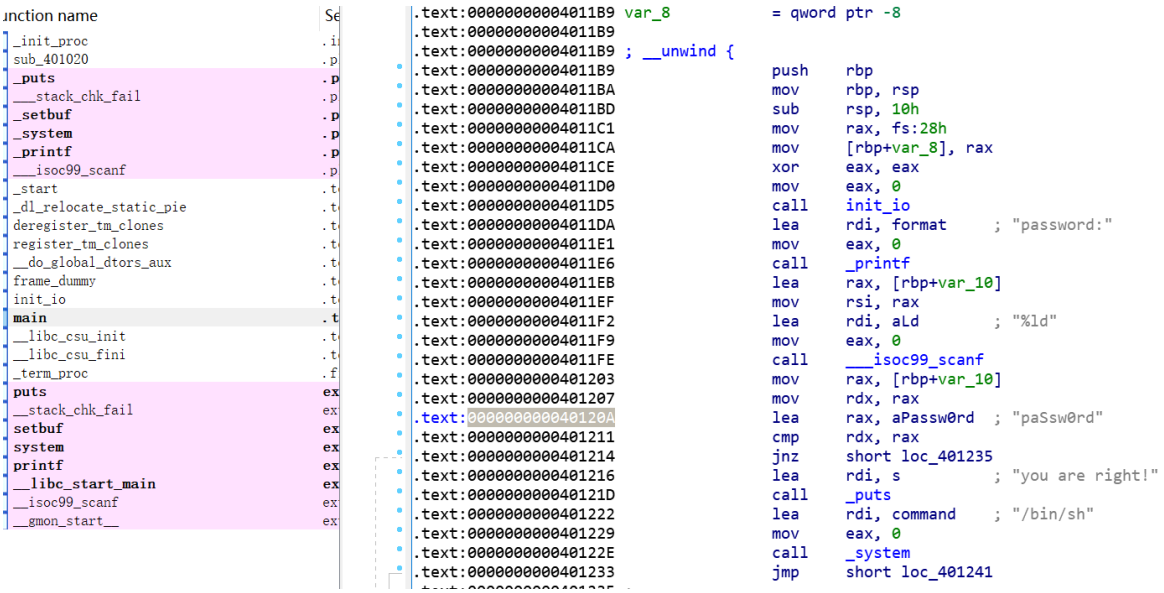
```python
    #raw_flag=input("give me your flag:\n")
    #cipher=list(raw_flag[6:-1])
    cipher=b"\x30\x46\x66\x33\x34\x6f\x59\x21\x3b\x41\x39\x79\x45\x
    length=len(cipher)
    res=[]

    for i in range (length):
        res.append(cipher[i]^i)
    cipher=res

    for i in range (length//2): #100
        cipher[2*i], cipher[2*i+1] = cipher[2*i+1], cipher[2*i]

    out=""
    for i in range (length):
        out+=chr(cipher[i])
    print(out)

main()
```

输出flag。

PWN

whitegive

使用IDA分析，在main函数找到密码。



kali出场。直接输入后发现它只吃一个字符，于是尝试地址。

```
:a:0000000000402012 aPassw0rd       db 'paSsw0rd',0          ; DATA XREF: main+51↑o
:a:000000000040201B ; char s[]
```

再次错误，于是尝试进制。

十六进制转十进制、16进制转10进制

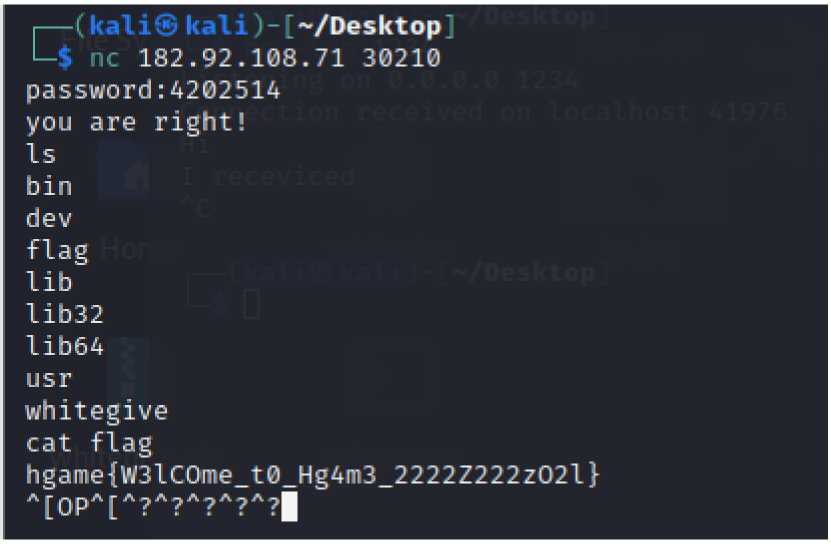
○ 2进制 ○ 8进制 ○ 10进制 ◉ 16进制 ○ 32进制 ○ 64进制 | 更多进制： 16

步骤：上面选择当前进制，然后下面输入数值，再点【转换】按钮，就能得到常见的进制数

402012 转换

进制	结果
二进制	10000000010000000010010
四进制	100002000102
八进制	20020022
十进制	4202514

kali界面输入nc以及密码后进行下图操作。



Crypto

まひと

--../-..../-..-./.----/-----/----./-..-./.----/-----/---../-..-./.----/.----/-----/-..-./----./-----/-..-./---../--.../-..-./...../...--/-..-./.----/-----/---../-..-./----./----./-..-./.----/-----/----./-..-./---../...../-..-./.----/.----/-.
/.----/-----/....-/-..-./.----/..---/-----/-..-./.----/-----/.----/-..-./.----/.----/-----/-..-./--.../....-/-..-./---../...../-..-./---../....-/-..-./---../-..../-..-./...../--.../-..-./--.../----./-..-./----./--.../-..-./.----/.----/-----/-..-./

看起来是摩斯密码。解码得到ASCII码。

```
8 6 / 1 0 9 / 1 0 8 / 1 1 0 / 9 0 / 8 7 / 5 3 / 1 0 8 / 9 9 / 1 0 9 / 8 5 / 1 1 6 /
8 4 / 7 1 / 1 0 8 / 1 1 4 / 9 7 / 8 4 / 1 1 2 / 5 7 / 8 6 / 1 0 9 / 1 1 6 / 1 1 6 /
1 0 0 / 1 0 7 / 1 1 2 / 1 0 5 / 7 3 / 8 4 / 7 0 / 8 9 / 1 0 0 / 6 9 / 7 0 / 5 2 / 9
0 / 8 3 / 7 0 / 1 1 1 / 9 9 / 6 9 / 4 8 / 1 2 0 / 1 0 1 / 4 8 / 4 8 / 1 1 4 / 7 9 /
8 8 / 1 0 4 / 1 2 0 / 1 0 1 / 1 1 0 / 7 4 / 8 5 / 8 4 / 8 6 / 5 7 / 7 9 / 9 7 / 1 1
0 / 5 3 / 1 0 6 / 8 5 / 1 0 9 / 9 9 / 4 8 / 1 0 1 / 6 5 / 6 1 / 6 1
```

记录后，base64解码，得到vigenere。

因为有}始终在第一位，栅栏后得到}xxxxxxxxxxx{xxxxx格式。

之后根据hint `格式为hgame{}`

逆序以及凯撒，得到flag。

Transformer

在kali中使用cat、命令把碎片整合起来。

ɜzko, pic oqhd hgxhnfhiehc oqh odxfxqd, cpopmpkh peehkk, khnvhn esfhc rfoq oqh nhkzuo?
ɔmkposantk fi kath rpd kqpxh an sant (rfoq oqh hgehxofanfoh as ihpnud azn hiofnh xnaczeo ufihzx, pic nzko xnaehkk pzoatpofepuud, thpifiy azn eufhio-kfch cqh mhko as uzej fi enhpofiy o

'qfuh kpsqh eaiehnik as wkai xpnkfiy avhn oqh sanfiy upiyzpyhk rfoq p uadpu sauuarfiy as chvhuaxhnk pic as oqh kxhhc pic xhnsantpqpo nzko pic epnya puka fieuzch thpi oqpo daz purpdk
'h' vh mhhi zkfiy nzko fikopnohc oqfk xnawheo. rh' vh mhhuaxthio xnaehkk, puuarfiy zk oa tavh tzeq tanh lzfeo rh eazuc opjh pcvpiopyhh pnh puka sficfiy oqh eahfyi szieofai fiohnspeh (ss

:feqphu nh ihr oa 1xpkkranc, daz epi kfyi zx oacpd rfoq eaikzth kathoqfiy fi e an snat ipn chvk oa saezk ai kauvfiy xnamuhkpshod as azn ezkoathnk' npohk pvpfupmuh san zkh, rh cfc qpvh
ɜzko qpkqhn vhnd xarhnszu (pic asohi avkranc. fs daz qpvh hvhn qpc oqh xuhpkznh oattzifepofai, pic tanh rnpxxhc fi p oqfi zf updhn oqpoxhnfthiok rqhi rh rhnh yhoofiy kopnohk mzfuo. rh'

ɔid xpnofezupn nzqhnh fk p kfyifsfepio xhnsantpieh mhihsfo oafuhnxupoh each oa eattzifepoh avhn oqh ssf.

ɟar yaa mzfuc p tpwanfod as oqh mpkh oqpo kheznfod-uposant.

'qpo rpk oqh mfyyhko eqpuuhiyh rqfuh chvhuaxxpkkranc?
ɐih as oqh tpfi oqfiyk oqpo d tpipyhthio pic arihnkqfx tachu. r mhhi chkfyihc rfoq tachni khikfmfufofhfok xnaehcznpu tpena kdkoht, rqfetanh.

: razuc ufjh oa oqpij tfeqphu san oqh fiofczpu peeaziok. fs daz' nh ranjfinaynpt eannheoihkk" pic tpid y, rficark qhuua) pic xuposant-kx, oqhi daz' vh mhhi uzejd hiazyq oaofai uhpck oa eu
ɔmkauz xnaczeofai koanfhk oqpo dapkkranck ka daz cai' o qpvh oanpofai khnvhn pk rhuu, thpifiy oqpo eqpiyhkehiohnhc pxxufepofaik nhlzfnh. oqhnhheznfod-ehiohnhc pxxufepofaik, pic r

ɔuoqazyq oqhnh pnh eaziouhkk epofaik ufjh 1xpkkranc?
ɔqhnh fk tanh oqpi hiazyq ohssfefhio, tanh saezkhc, pic qfyqhn lzpufod eacho-mpkhc kauzofai eazuc xnavfch.txpid pic dazn nauh oqhnh?
lxpkkranc fk p xpkkshd
ɐpi daz opuj pmazo oqthiok as azn xnaczeok.

'o cf ai oax as oqpo. rh npi p upnyh iztmhn as hgka xanohc oqh 1xpkkranc mnpfi – oqh hich epi mhihsfo snat nzko' k xqfuakaxqd.

's daz' iofpu uaepufbpofai ftxuhthiopofai oa thho oqh nhlzfnh zkh kathoqfiy td ohpt qp snat ya oa nzko po oqh hic as 2019 ka oqpofvh xuposant ufmnpnfhk. rh zkh oqfk oa ynhpo hsstk rfc

ɔid jhd opjhprpdk oqpo o cznfiy azn nhvfhr xnaehkk.

ɔqfk oaau qnh fk nzko ynhpo oa zkh, pic rqhnmh kznh oa sauuar zk ai thcfzt, chv, an orfoxhaxuh pic 70,000 mzkfihkkhk oa kheznhz tfyqo rpio oa eqhej azo.hieh?
's daz' nh ihr oa nzko, kopno ktpuu pic mzfuch qaxfiy rhmpkkhtmud razuc opjh zk sznoqhn fi oh kopnohc p ihr khnfhk ai puu tpwan mnarkhnk, xuzk chkjoax pic tamfueh p thtand-nhupoh
ɘh azn odtpjhk pxfk cfssfezuo oa zkh fieannheoud pic nhkzuok fhvk epi eaiofizh oa ranj fihe xnftpnfud mhfiy mpejhc md pxxuh. hnvfhr xhaxuh oqpo qpvh zkhc nzko sapnh zkfiy nzko oa enh

ɐazuc daz ohpcvpiopyhk ufjh kxhhc an odxh/thtasnat nziofth eqhejfiy as eaikonpfiok pic fivpnfpiok; th chmzyyfiy eatxpnhc oa aoqhn upiyzpyhk. fs fo eatxfuh fk purpdk oqh axofai oa cfx car

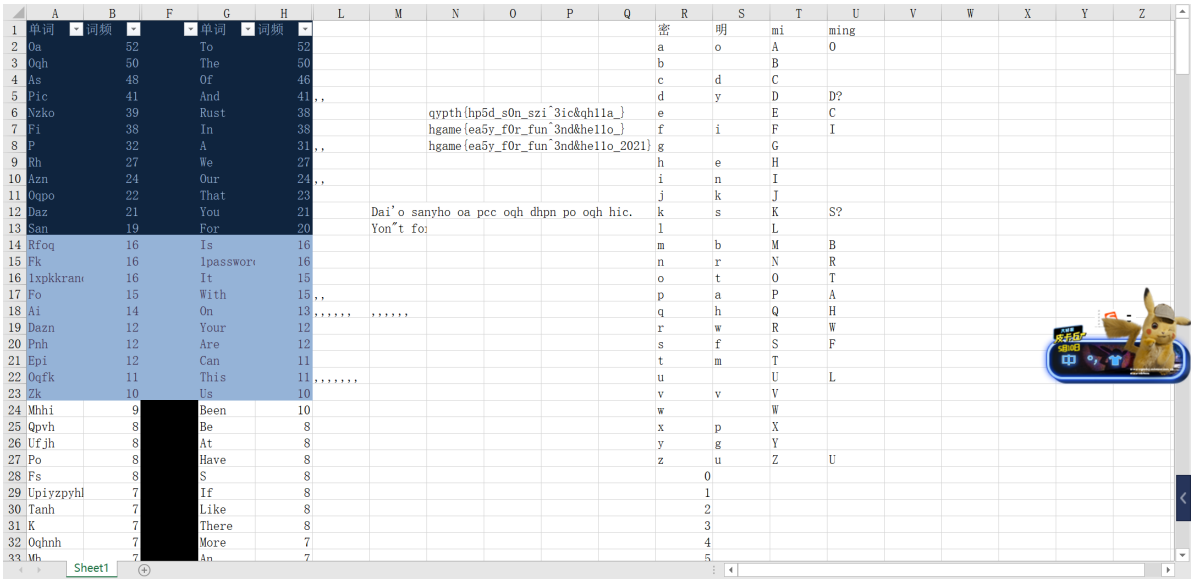
ɔqhkh qpvh mhhi fi xnaq qpk puuarhc zk oa rnfoh p oaauokhe cpopmpkh, rqfeq fk eattzifod-kazpk mhhi pi fiohynpu eatxaihio fi azn chvhhk fi oqh eufhio pxxufepofaik oqpo pnh epzyqtpipy

ɔa nhpc tanh pmazo xnayniofnh nziofth fi rpkt qpk mhhi p eqpuuhiyh. pttfiy upiyzpyhk ufjh nzko, ieh as chxuadfiy nzko oa rhmpkkhtmud fi azn mnarkhn hgoeaionfmzoank.

ɔa uhpni rqd oqfk upiyvhnd aih as azn opnyho upiyzpyhk. rh' vh fiohynpohnfiy san eufhio pxxk qhnh po 1xpkoqhfn oaauk pic caek epi mh sazic qhnh.

'qhhsanh fo hghezohk. qpvfiy oa xhnc spuu kqano san zk fi aih jhd pnhp: rh rhnxhk pnh chsfihc fi nzko, rh pnh pmuh oa ftthcfp oqhfn khikfofvh cpop. fo nhthtmhnk puu dazn xka oqpo rh'
:pid as oqh sasant uhkk nziofth kopoh vpufcp xnaczeofai po 1xpkkranc san p shrno as dazn eachmpkh fk rnfoohi fi nzkoofiy p ranjazo.

使用word和词频分析工具，发现txt最后一句话中含有提示"加上年份"，于是2021安排。



单词	词频	单词	词频
Oa	52	To	52
Oqh	50	The	50
As	48	Of	46
Pic	41	And	41
Nzko	39	Rust	38
Fi	38	In	38
P	32	A	31
Rh	27	We	27
Azn	24	Our	24
Oqpo	22	That	23
Daz	21	You	21
San	19	For	20
Rfoq	16	Is	16
Fk	16	1passwor	16
1xpkkran	16	It	15
Fo	15	With	15
Ai	14	On	13
Dazn	12	Your	12
Pnh	12	Are	12
Epi	12	Can	11
Oqfk	11	This	11
Zk	10	Us	10
Mhhi	9	Been	10
Qpvh	8	Be	8
Ufjh	8	At	8
Po	8	Have	8
Fs	8	S	8
Upiyzpyhl	7	If	8
Tanh	7	Like	8
K	7	There	8
Oqhnh	7	More	7
Mh	7	An	7

qypth{hp5d_s0n_szi^3ic&qh11a_}
hgame{ea5y_f0r_fun^3nd&he11o_}
hgame{ea5y_f0r_fun^3nd&he11o_2021}
Dai'o sanyho oa pcc oqh dhpn po oqh hic.
Yon"t fo:

~~完结散花~~